UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA APLICAÇÃO PROPOSTA PARA TROCA DE ARQUIVOS EM REDES OPORTUNISTAS SEM INFRAESTRUTURA

André Francisco Martins

Orientador

Prof. Dr. Carlos Alberto Vieira Campos

RIO DE JANEIRO, RJ - BRASIL

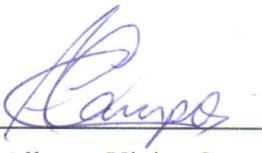
OUTUBRO DE 2014

UMA APLICAÇÃO PROPOSTA PARA TROCA DE ARQUIVOS EM REDES OPORTUNISTAS SEM INFRAESTRUTURA

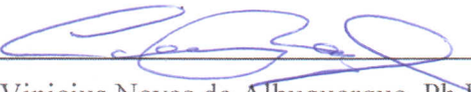
André Francisco Martins

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO), APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

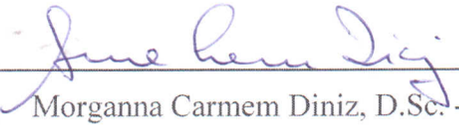
Aprovada por:



Carlos Alberto Vieira Campos, D.Sc. – UNIRIO



Célio Vinicius Neves de Albuquerque, Ph.D. – UFF



Morganna Carmem Diniz, D.Sc. - UNIRIO

Rio de Janeiro, RJ – BRASIL

OUTUBRO DE 2014

Martins, André Francisco.
M386 Uma aplicação proposta para troca de arquivos em redes oportunistas sem infraestrutura / André Francisco Martins, 2014.
106 f. ; 30 cm

Orientador: Carlos Alberto Vieira Campos.
Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2014.

1. Sistemas de transmissão de dados. 2. Redes MANETs. 3. LocalScan (Protocolo de rede de computador). 4. Peer to Peer. 5. MaxProp. 6. BUBBLE Rap. 7. ProPHET. I. Campos, Carlos Alberto Vieira. II. Universidade Federal do Estado do Rio de Janeiro. Centro de Ciências Exatas e Tecnológicas. Curso de Mestrado em Informática. III. Título.

CDD – 004.6

"Keep Walking"
Johnnie Walker

AGRADECIMENTOS

Agradeço ao meu orientador Carlos Alberto Campos pela dedicação nesses mais de dois anos.

Agradeço a todos os professores e funcionários da Unirio que me ajudaram ao longo dessa jornada, em especial, aos professores Morganna Carmem Diniz e Mariano Pimentel que participaram de minhas bancas de acompanhamento e contribuíram muito para minha formação.

Aos mestres Bruno Fernandes Guedes e Luiz Fernando Teixeira de Farias, companheiros de mestrado, que me ajudaram a não esmorecer diante das dificuldades. Agradeço ao amigo e irmão Eduardo Lima Rodrigues que forneceu apoio e recursos para realização das simulações.

Agradeço aos meus pais, Fatima e Jorge, por tudo o que fizeram por mim, permitindo que eu chegasse até aqui e à minha companheira Alessandra, primeira leitora e revisora desse texto.

Por último, um agradecimento especial à minha filha e musinha inspiradora, Mariana, que trouxe uma alegria diferente à minha vida e que em tão pouco tempo já me ensinou tanto.

MARTINS, André Francisco. **UMA APLICAÇÃO PROPOSTA PARA TROCA DE ARQUIVOS EM REDES OPORTUNISTAS SEM INFRAESTRUTURA.** UNIRIO, 2014. 105 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada - UNIRIO

RESUMO

Redes oportunistas são redes móveis de comunicação cujos nós se comunicam sem auxílio de infraestrutura e nenhuma presunção sobre a topologia de rede pode ser feita em função da grande mobilidade de seus nós. Os nós, quando em contato, têm a oportunidade de transmissão de mensagens. Durante o intervalo de desconexão, as mensagens são armazenadas até serem repassadas em uma estratégia chamada de *store, carry and forward*.

Quando os nós são representados por humanos, as estratégias de encaminhamento de mensagens procuram explorar as características temporais, espaciais e/ou sociais dos seres humanos com o intuito de prever um contato futuro entre os nós e, assim, trocar dados.

Como não dependem de infraestrutura para se comunicar e possuem uma arquitetura distribuída, as redes oportunistas são difíceis de serem controladas por uma organização ou Estado. Redes oportunistas são diferentes da Internet que possui uma estrutura hierarquizada e que, com isso, permite a indisponibilidade premeditada de serviços por agentes governamentais [1]. Por isso, há uma grande preocupação com a privacidade dos dados que são transmitidos pela Internet atualmente. Já que empresas e governos podem monitorar atividades civis e esse controle fomenta uma discussão a respeito de uma rede verdadeiramente livre.

Esta dissertação possui 3 contribuições. Apresenta a Aplicação de Troca de Arquivo - ApTA, que é uma proposta de aplicação de troca de arquivos P2P em redes oportunistas onde procura-se disseminar rapidamente partes do arquivo na rede para favorecer a distribuição do arquivo, aumentando a redundância de nós fonte das partes do arquivo, aproveitando as características das redes oportunistas. A avaliação de desempenho da aplicação ApTA mostrou que ela permite que os nós completem mais rapidamente os arquivos compartilhados, sem onerar a rede com muitas réplicas de mensagens. A aplicação ApTA consegue, totalmente livre de infraestrutura, aproximar o conteúdo do usuário e, assim, acelerar a troca de partes do arquivo.

Também é proposto um protocolo de encaminhamento de mensagens denominado LocalScan baseado no padrão de histórico de contatos e padrão de movimento. LocalScan é um protocolo de características de disseminação controlada, evitando que a rede seja inundada com mensagens e, assim, evitar que o custo de transmissão de uma mensagem seja elevado.

E por fim, é questionada a representação vazia da camada de aplicação, situação que ocorre quando é criada uma aplicação geradora de tráfego para análise de rede sem se preocupar com as semânticas de uma aplicação. Percebemos que essa falta de caracterização da aplicação pode levar a resultados diferentes na análise do comportamento de uma rede.

Palavras-chave: Redes Oportunistas, Redes MANETs, Mobilidade, Peer to Peer, MaxProp, BUBBLE Rap, ProPHET, LocalScan.

ABSTRACT

Opportunistic networks are mobile communication networks whose nodes communicate without the aid of infrastructure and no assumption on the network topology can be made due to the high mobility of its nodes. Nodes, when in contact, have the opportunity to broadcast messages. During the disconnect interval, the messages are stored until they are passed in a strategy called store, carry and forward.

When the nodes are represented by humans, strategies for routing messages seek to exploit the temporal, spatial or social characteristics of human beings in order to predict future contact between nodes and thus exchange data.

As opportunistic networks does not rely on infrastructure to communicate and has a distributed architecture, they are difficult to be controlled by an organization or state. Opportunistic networks are different from the Internet that has a hierarchical structure and easy control. There is great concern about the privacy of data being transmitted over the Internet today. Businesses and governments monitor civilian activities and this control fosters a discussion of a truly free network.

This paper presents a proposal of application of P2P file exchange in an opportunistic network where the aim is to quickly disseminate portions of the file on the network to support distribution of the file, increasing the redundancy of the sources we file parts fit application. The performance evaluation of application fit showed that it allows us to complete the shared files quickly, without burdening the network with many replicas of messages. APTA totally free application infrastructure can bring user content and thus accelerate the exchange of parts of the file.

It`s proposed a routing protocol called LocalScan messages based on the pattern of contact history and pattern of movement. LocalScan protocol is a controlled spread characteristics, preventing the network is flooded with messages and thereby avoid the cost of transmitting a message to be high.

Finally, it's questioned the empty representation of the application layer, a situation that occurs when a traffic generator application is created for network analysis without worrying about the semantics of that application. We realize that this lack of characterization of the application may lead to different results when analizing the network behavior.

Keywords: Computer Networks, Opportunists Networks, MANETS, Mobility, Peer to Peer, MaxProp, BUBBLE Rap, ProPHET, LocalScan.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

CDN - Content Delivery Network

DTN - Delay Tolerant Network

GPS - Global Positioning System

HTTP - Hypertext Transfer Protocol

MANET – Mobile Ad Hoc Network

P2P – Peer to Peer

PDA – Personal Digital Assistant

PNAD - Pesquisa Nacional por Amostra de Domicílios

QR – Quick Response

RSSI - Received signal strength indication

VANET – Vehicular Ad Hoc Network

XML - eXtensible Markup Language

1 INTRODUÇÃO

A computação vem nas últimas décadas se tornando cada vez mais presente no dia a dia de todos. Se há algumas décadas a presença do computador era restrita a universidades, aplicações na área militar e algumas funções em grandes empresas, hoje o computador é onipresente [2]. Depois da popularização dos desktops em lares e escritórios, outros segmentos foram surgindo e necessitando de outras funcionalidades. Dentre elas, é possível destacar a mobilidade que produziu desde notebooks a computadores vestíveis como o Google Glass.

Desde a criação do Macintosh Newton em 1992, considerado o primeiro PDA (Personal Digital Assistant) construído, muitos outros vieram ao mercado. Em outubro de 2013, Tim Cook, CEO da Apple, anunciou que a empresa havia vendido 170 milhões de unidades de Ipad até aquele momento [3]. Basicamente, um PDA é um computador de tamanho reduzido com funcionalidades de agenda, aplicativos de produtividade e escritório e algum meio de conexão a redes sem fio. Não demorou para que os telefones celulares evoluíssem e acabassem dominando o segmento de computação pessoal ao juntarem as funcionalidades existentes nos PDAs à funcionalidade de fazer ligações telefônicas. Tal dispositivo com esse conjunto de funcionalidades ou possibilidade de instalação é chamado de *smartphone*. E mais: devido a convergência de funcionalidades, os *smartphones* se tornaram cada vez mais necessários. Não foram só os PDAs que se tornaram obsoletos com a evolução dos *smartphones*. Players de música, câmeras do tipo *point and shoot*, calculadoras, leitor de livros eletrônicos,

gravador de voz, blocos de anotação, despertadores, todos essas ferramentas foram deixadas de lado. Segundo a Cisco [4], em 2013 foi alcançada a marca de 7 bilhões de dispositivos móveis em funcionamento e os *smartphones* são responsáveis por 21% desse volume.

Como dispositivo de uso pessoal, os *smartphones* se tornaram ferramentas sempre disponíveis ao alcance das mãos e são levados ao longo das jornadas de seus donos. Percorrem distâncias em trens, ônibus, automóveis ou mesmo a pé e são colocados ao alcance de outras centenas ou milhares de dispositivos diferentes que estiveram em contato com outras centenas ou milhares de dispositivos antes. A possibilidade de se construir uma rede de comunicação com esses dispositivos e aproveitar do movimento humano para carregar os dados até o destino chamou a atenção de pesquisadores para a área que viria a ser conhecida como redes oportunistas. Em determinados contextos, o termo redes tolerantes a atraso pode ser usado de forma equivalente, embora exista diferenças nas ideias originais.

Redes oportunistas são redes formadas por dispositivos que possuem mobilidade tamanha que durante a maior parte do tempo os nós estão desconectados e se aproveitam de oportunidades de transmissão para encaminhar os dados para outro nó [5]. Tais oportunidades acontecem quando os nós, através do deslocamento dos seus portadores, entram no alcance do rádio de outro dispositivo e podem se comunicar. De oportunidade em oportunidade, os dados vão sendo transferidos de nó para nó até que alcance o destinatário. O grande desafio nas redes oportunistas é conseguir entender e usar o movimento humano para conseguir que a mensagem consiga chegar ao destino.

Existem vários modelos que procuram explorar determinadas características do movimento humano para que esses modelos possam influenciar no encaminhamento das mensagens. Dentro desse contexto, informações sobre o movimento humano como

frequência e distância de deslocamento, localidade, etc, são modeladas com o intuito de melhorar as estimativas de um possível futuro contato entre nós que carreguem a mensagem na direção correta [6].

Essas mensagens que são encaminhadas pela rede, em geral, são oriundas das aplicações. A maioria das aplicações propostas são de utilização para busca e salvamento nos chamados cenários de emergência, onde toda ou parte da infraestrutura de comunicação é atingida e danificada por eventos que isolam uma determinada região ou em cenários onde a Internet não está disponível para determinados nós, funcionando como uma rede de acesso destes nós para a Internet, por exemplo. É provável que a razão do grande número de aplicações propostas desse tipo se deve ao fato de que o conjunto de conhecimento sobre redes oportunistas não permite ainda a execução de aplicativos diversos com o mesmo desempenho quando comparados aos similares executados em redes cabeadas ou redes móveis das companhias telefônicas.

1.1 MOTIVAÇÃO

Embora, originalmente, os principais fatores impulsionadores da pesquisa em redes oportunistas fossem os ditos cenários de emergência e levar conectividade a regiões rurais e sem infraestrutura, outros cenários parecem surgir como possíveis locais onde redes oportunistas poderiam ser usadas. Locais esses em que mesmo com a presença de redes sem fio infraestruturada ou cobertura 3G/4G não tem requisitos como liberdade de expressão ou baixo custo de acesso a Internet [7].

Comunicação em áreas sob domínio de governos opressores parece ser uma preocupação em alguns trabalhos como [7] e [8]. A Internet tem sido a principal ferramenta para comunicação nos dias de hoje, mas com sua estrutura hierarquizada é relativamente fácil indisponibilizar parte de uma determinada região ou de serviços da

Rede, vide o episódio de bloqueio imposto ao Twitter pelo governo da Turquia. Além de bloqueios e indisponibilidade de serviços, governos também monitoram atividades civis fazendo uso de programas farejadores para receptar dados de comunicação de usuários. Nesse caso, redes oportunistas pela sua característica descentralizada e sem uma maneira de identificar formalmente o usuário autor de uma mensagem, seria uma solução que protegeria e tornariam anônimos os usuários dessa rede contra programas farejadores e tornaria difícil indisponibilizar o serviço em face da rede ser construída com a adesão de muitos nós, não possuindo um ponto único de falha que poderia ser utilizado para "desligar" o sistema.

Quanto a possibilidade de redes oportunistas vir a se tornar uma solução mais barata que os planos de cobertura móvel, em grandes centros urbanos, com o adensamento de nós é possível que um grande número de conexões através das interfaces de rádio disponíveis poderia ser feito e com isso ter acesso a conteúdo fornecido dos dispositivos móveis dos usuários. É importante lembrar que, atualmente, os usuários não são mais meros consumidores de informação. Também produzem grande quantidade de material que parte não é armazenado em um servidor, ficando unicamente disponíveis em seus dispositivos móveis.

1.2 Objetivos

Esse trabalho tem como seu objetivo principal, a criação de um mecanismo para troca de arquivos usando o paradigma P2P para ser usado em redes oportunistas de forma totalmente independente de qualquer tipo de infraestrutura. Desta forma, foi proposta uma aplicação para troca de arquivos que não necessita de qualquer infraestrutura para que funcione. A aplicação proposta procura reduzir o tempo de

resposta da comunicação ao permitir que nós intermediários possam contribuir na disseminação dos arquivos.

Além da aplicação, o presente trabalho procura questionar se a falta de representatividade na camada de aplicação não pode interferir nos resultados de avaliação de uma rede oportunista.

Foi feita uma comparação envolvendo alguns protocolos de encaminhamento de mensagens em redes oportunistas com a aplicação proposta e uma aplicação P2P característica para verificar o comportamento da rede através das métricas número de mensagens criadas, atraso, sobrecarga, taxa de entrega, número de saltos para entrega de mensagem e número de nós com cópias completas do arquivo.

1.3 Estrutura do estudo

Esta dissertação está dividida em 7 capítulos. No Capítulo 2, há uma introdução aos conceitos sobre redes móveis ad hoc e redes oportunistas, a importância do entendimento da mobilidade humana e uma classificação das propriedades do movimento humano. Também são abordados os protocolos de encaminhamento em redes oportunistas. No final do capítulo são abordados conceitos sobre paradigmas de aplicações cliente-servidor, P2P e sobre o Bittorrent.

No Capítulo 3, é feita uma revisão das aplicações em redes oportunistas , em especial, das aplicações de troca de arquivo. No Capítulo 4, são apresentados e comentados alguns dos problemas referentes às aplicações de troca de arquivo em redes oportunistas.

No Capítulo 5 é apresentada a aplicação desenvolvida nesse trabalho e são abordadas suas mensagens e implementação. A avaliação da aplicação será discutida no

Capítulo 6 com a descrição dos parâmetros utilizados, cenários usados e métricas escolhidas.

E por fim, no Capítulo 7, são apresentadas as conclusões e ideias de possíveis linhas de trabalhos futuros.

2 Referencial teórico

Neste capítulo, o objetivo é dar ao leitor uma visão inicial sobre redes móveis ad hoc e redes oportunistas, cuja área de pesquisa emergiu da primeira. O capítulo está dividido em 4 partes sendo a primeira dedicada a redes ad hoc e suas características. Na segunda, veremos os conceitos por trás de redes oportunistas, a importância do entendimento da mobilidade humana e uma classificação das propriedades do movimento humano. A terceira parte será destinada aos protocolos de encaminhamento em redes oportunistas e como será visto também como eles tiram proveito do estudo da mobilidade humana. Ao final, iremos ver conceitos dos paradigmas de aplicações cliente-servidor, P2P e sobre o Bittorrent, protocolo utilizado em redes P2P para troca de arquivos.

2.1 Redes móveis *ad hoc*

Redes móveis *Ad Hoc* (Mobile *Ad hoc* Network- MANET) são redes de comunicação sem fio caracterizadas pela dificuldade em definir a topologia de rede devido a grande movimentação dos seus nós. Em razão dessa característica, a conectividade não é constante e isso pode ocasionar em desconexões e particionamentos na rede a medida que os nós saiam do alcance uns dos outros. Além da mobilidade, outra característica desse tipo de rede é a ausência de infraestrutura de comunicação que permita acesso a outros serviços. Não há nós fixos e conectados a redes cabeadas que funcionem como gateways para os demais nós. Assim, estratégias são implementadas

para resolver o problema da frequente mudança de posição dos nós e a dificuldade de estabelecimento da topologia de rede o que permitiria o estabelecimento de rotas de encaminhamento de mensagens entre um nó e outro como é feito na Internet. A Figura 2.1 mostra uma MANET em dois momentos com 5 nós. Os círculos menores representam os dispositivos e os maiores o raio de alcance do radio de seu dispositivo. Ou seja, dentro desse raio, os nós podem se comunicar. No primeiro momento Figura 2.1 (a), todos os nós estão conectados e no segundo momento Figura 2.1(b), devido a movimentação, cria-se partições na rede criando três grupos de nós conectados.

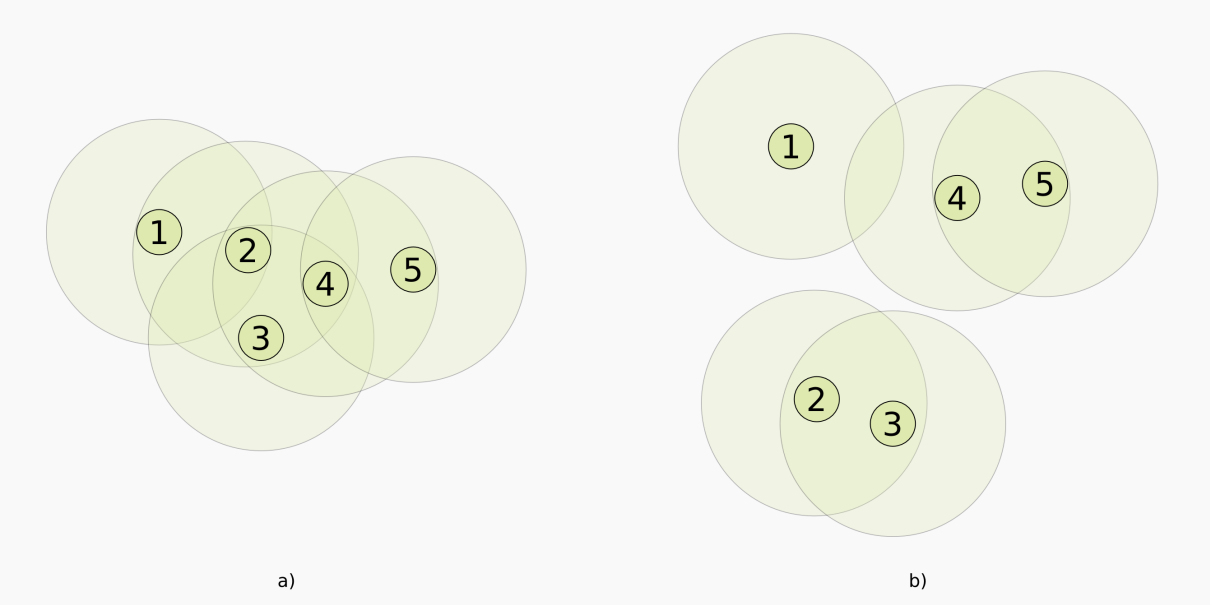


Figura 2.1 - Rede MANET em dois momentos. Com 5 nós conectados em a) e uma partição causada pela movimentação em b).

Enquanto as estratégias de roteamento em redes MANET procuram contornar o problema da desconexão, outras redes aceitam a perda de conectividade como um fato inevitável. É o caso das Redes Tolerantes a Interrupção e Atraso (Delay Tolerant Network – DTN) que tiveram sua arquitetura especificada em [9]. Este tipo de rede permite que os nós possam trocar dados entre eles eventualmente, sem a existência de

uma conexão fim-a-fim. Os nós que funcionam como gateways entre as redes são chamados de DTN Nodes e são responsáveis pela custódia das mensagens com os dados a serem transferidos nos períodos de conectividade. Projetos de ligação de áreas remotas à Internet foram propostos utilizando a rede pública de transportes que, com certa regularidade, interligava as redes [10] em um exemplo do chamado DTN previsível, pois os momentos de conexão são programados e conhecidos. Essa diferenciação se faz em razão de outras redes tolerantes a atraso onde não é conhecido o próximo momento de conexão.

2.2 Redes Oportunistas

É uma evolução do estudo das redes MANETs caracterizada pela conectividade não ser constante entre os nós e nenhuma presunção sobre a topologia de rede pode ser feita. Os nós, quando em contato, têm a oportunidade de transmissão de mensagens. Durante o intervalo de desconexão, as mensagens são armazenadas até serem repassadas em uma estratégia chamada de *store, carry and forward*. Ou seja, se não for possível repassar a mensagem, a mesma é armazenada e transportada aproveitando-se da movimentação do nó até que nova oportunidade de encaminhamento apareça ou seja feito contato com o nó destino. As mulas de dados, como são chamados os agentes que carregam os dispositivos que representam os nós, contribuem com sua movimentação para conectar nós que estejam fora do alcance, criando assim, as oportunidades de transmissão.

A dificuldade em definição da topologia dessa rede é uma quebra de paradigma onde a cada oportunidade de encaminhamento deve ser avaliada se o nó detentor da mensagem deve ou não repassá-la, baseando-se nas informações apresentadas pelo nó em contato. A cada encaminhamento da mensagem, espera-se que o nó receptor da

cópia da mensagem irá levá-la para mais perto do nó destino ou entregá-la a outros nós que o façam até que a mensagem seja entregue.

Muito do que é empregado em redes oportunistas tem origem no estudo das redes DTN. Na literatura, os termos redes tolerantes a atraso e redes oportunistas são usados quase como sinônimos em determinados trabalhos. Em [5], há uma pequena discussão sobre os dois conceitos e sobre a evolução dos mesmos ao longo do tempo.

Em 2013, mais de 400 milhões de novos *smartphones* foram ativados no mundo, totalizando 7 bilhões de dispositivos móveis [11]. Existe um enorme potencial em dispositivos e contatos todos os dias dentro dos centros urbanos, densamente povoados, que poderiam usar as redes oportunistas para trocar dados. Especificamente na cidade do Rio de Janeiro, podemos usar como exemplo os dados fornecidos pela concessionária Metrô Rio que transporta 645 mil passageiros em todos os dias úteis [12] e o resultado do PNAD 2011 [13] onde mais de 69% da população brasileira tem celular, é possível imaginar um enorme potencial de utilização dos celulares como dispositivos para formação de redes oportunistas.

Neste cenário, os nós representam os seres humanos que carregam seus dispositivos móveis ao longo de seus trajetos diários e é importante conhecer o movimento humano para poder entender como e para onde os dados são carregados. Entendendo isso, é possível presumir ou estimar um encontro futuro entre dois nós e uma eventual transferência de mensagens. Tanto traces de dados de localização de usuários, quanto medida de contatos, permite analisar o movimento e, consequentemente, criar modelos de mobilidade que possam auxiliar nas decisões de encaminhamento de mensagem. Existem vários modelos que tentam representar uma ou mais propriedades do movimento humano. Podemos citar [14] onde usando dados oriundos de 100 mil usuários de telefone celular que foram coletados por 6 meses,

permitiram criar um modelo com base nas distâncias espaciais. Neste trabalho, foi mostrado que a distribuição dos comprimentos dos deslocamentos dos usuários tinham tendência de seguir uma lei de potência truncada e que o movimento era heterogêneo. Enquanto alguns usuários tinham tendência a se movimentar em locais próximos de sua vizinhança, outros se deslocavam a locais mais distantes de suas casas. Esse modelo é um exemplo que procura explorar as propriedades espaciais do movimento humano como frequência ou distância.

Segundo [6], as propriedades da mobilidade humana de interesse podem ser classificadas em três eixos: espacial, temporal e social.

2.2.1 Propriedades Espaciais

São as propriedades que buscam padrões nos movimentos em suas distâncias físicas e frequências dos deslocamentos. Em [15], é demonstrado que o movimento humano tem alto grau de regularidade espacial, indicando que os nós têm uma grande probabilidade de retornar para poucos lugares com alto índice de visitas. Além disso, também foi exposto que a maioria dos deslocamentos é confinado a uma determinada região.

2.2.2 Propriedades Temporais

Assim como a regularidade no espaço, as propriedades temporais também têm grande regularidade. Se frequentamos determinados lugares com maior frequência que os outros, também o fazemos em intervalos regulares e a estadia tende a durar a mesma grandeza. Em [16] é medida a probabilidade de retorno para uma localização previamente visitada em t horas. É verificado que a probabilidade de retorno é caracterizada por picos de 24h, 48h e 72h, o que descreve a recorrência de nossas atividades diárias. Já [17] caracteriza o movimento humano através de saltos e estadias em cada um dos pontos visitados pelos indivíduos.

2.2.3 Propriedades Sociais

Essa é a classe de propriedades mais estudada pois as duas outras classes são consequências desta quando se trata de seres humanos carregando dados. Em geral, os movimentos que os indivíduos realizam ao longo de suas vidas, têm relação com os seus hábitos sociais. Grande parte dos seres humanos se deslocam todos os dias até os respectivos trabalhos, além de ir a outros pontos de interesse e retornam para suas moradias. A rotina da jornada humana é representada através dessa classe. Em seu trajeto, encontram-se com outros indivíduos que estejam relacionados aos mesmos hábitos, como os companheiros de trabalho e a família. Desses encontros, podemos extrair contatos e daí identificar o grafo representativo da rede social de determinado indivíduo. Nesse grafo, os vértices representam os nós participantes da rede, ou seja, outros indivíduos com os quais o nó teve contato. As arestas representam a força da relação entre os dois indivíduos.

Nessa análise, duas medidas são importantes para estimar o quão forte é a relação entre os indivíduos. O número de contatos e o tempo entre contatos permitem medir a frequência e a duração dos encontros entre os indivíduos. Com esses dados, várias métricas foram criadas para dar peso às arestas dos grafos sociais e, com isso, eleger os melhores nós para repassar a mensagem e assim melhorar as chances de entrega da mensagem ao seu destino final. Por exemplo, frequência de encontros, número total ou médio de encontros por período, período médio de separação [18], pressão social [19] que leva em consideração três características do histórico de contatos, frequência, regularidade e longevidade. Algumas dessas métricas funcionam em determinados casos, mas não em outros. Após ser construído, o grafo é utilizado

para o racional da estratégia para decidir se uma mensagem deve ou não ser repassada ao outro nó durante um encontro. As métricas listadas funcionam como guia para indicar o quais indivíduos são melhor indicados para encaminhar uma mensagem.

De acordo com a Figura 2.1,três tipos de contato podem ocorrer dentro de um intervalo de tempo t. Se utilizarmos uma métrica para representar as arestas baseada somente em tempo de contato, o cenário (a) seria escolhido pois é o que mais oferece largura de banda para troca de mensagens, mas esse cenário tende a ser improvável a medida que o intervalo de tempo analisado cresce, pois a probabilidade de desconexão aumenta. Dos três, o cenário (a) é o melhor por oferecer maior largura de banda, mas se fossem utilizadas métricas de frequência, regularidade ou tempo mínimo entre contatos, os outros cenários poderiam ser escolhidos no lugar do cenário (a).

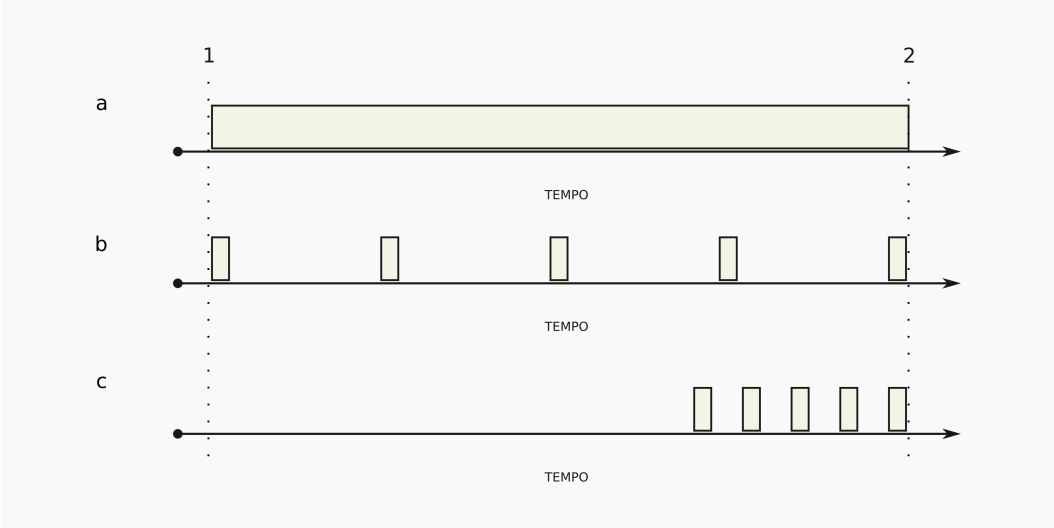


Figura 2.2 - Três possibilidades de contatos entre dois nós em um tempo t

2.3 Principais Estratégias de encaminhamento em redes oportunistas

Em redes oportunistas não é possível saber quando vai acontecer um contato entre dois nós e também não é possível fazer qualquer presunção sobre conectividade fim-a-fim [5]. O remetente e o destinatário de uma mensagem podem nunca vir a entrar em

contato. Em redes oportunistas, as estratégias são pensadas para aproveitar os encontros entre dois nós que é a oportunidade de transmissão de dados. De acordo com a estratégia empregada, a mensagem pode ou não ser repassada ao nó encontrado. Em alguns protocolos de encaminhamento de mensagens, o nó com a mensagem durante um encontro pode decidir que é melhor manter a mensagem e carregá-la mais um pouco através do deslocamento humano ou outro agente de mobilidade do que repassá-la para o nó encontrado.

A seguir, veremos algumas classes de protocolos de encaminhamento de mensagens em redes oportunistas e descreveremos as principais características de alguns de seus representantes.

2.3.1 Protocolos Epidêmicos

Nem todas as estratégias exploram uma das classes discutidas na seção 2.2. Algumas, em especial as mais antigas, simplesmente só se preocupam em espalhar a mensagem em uma estratégia de inundação. É o caso do algoritmo chamado de Epidêmico [20] e todas as suas variações. O nome advém do seu funcionamento ser parecido com a epidemia de um vírus. A cada contato entre dois nós, a mensagem é repassada para o outro nó, “infectando-o” e assim aumentando o número de cópias da mensagem dentro da rede. Após a “infecção”, o novo “infectado” passa a ter comportamento semelhante aos outros portadores da mensagem, passando a “infectar” todos os nós com os quais venha a ter contato. A única restrição na versão inicial é o tamanho do buffer que funcionava como uma fila para armazenar as mensagens com as mais antigas sendo apagadas para dar lugar as mais novas. Existem variações do protocolo Epidêmico original que tentam mitigar os problemas de buffer ao limitar o número de nós pelos quais uma mensagem pode trafegar. O 2-hop [21] é um exemplo onde a mensagem só pode dar dois saltos até chegar ao destinatário.

O protocolo Spray and Wait [22] é uma evolução do epidêmico com o intuito de não inundar a rede com mensagens e funciona da mesma forma até que um determinado número (N-1) de mensagens seja repassado. Depois que (N-1) cópias da mensagem foram feitas, só é permitido repassar a mensagem para o destinatário final. É uma forma de espalhar as mensagens sem saturar a rede de mensagens como ocorre no protocolo epidêmico original. Afinal, o grande número de mensagens polui o meio e degrada a performance da rede ao diminuir a razão entre o número de mensagens trocadas e o número de mensagens entregues.

2.3.2 Protocolos Probabilísticos

Como a maioria das pessoas não se movimentam aleatoriamente em seus percursos, o protocolo Probabilistic Routing Protocol using History of Encounters and Transitivity (ProPHET) [23] procura explorar os padrões de movimento humano existentes no mundo real através do histórico de contatos. Para tal, o ProPHET utiliza uma métrica chamada Previsibilidade de Entrega (P) usada para determinar se um nó deve repassar ou não uma mensagem para outro nó durante um encontro. Sempre que dois nós se encontram, eles trocam os valores que dispõem sobre os outros nós da rede. Assim, P(a, b) representa a probabilidade de encontro entre os nós a e b. Cada nó mantém um valor para cada outro nó conhecido e essa métrica é uma representação dos encontros entre esses nós. A cada encontro, essa métrica é recalculada para cada nó na rede. Se um nó a encontrar com o nó b, a P(a, b) será incrementada enquanto a dos outros nós será diminuída pois há uma previsão de “envelhecimento” desses valores caso dois nós não se encontrem por determinado período.

MaxProp [24] também utiliza o histórico de encontros para estimar a probabilidade de entrega da mensagem. Cada nó da rede, mantém um vetor contendo a probabilidade de entregar a mensagem para um outro nó da rede. Quando os nós se

encontram, eles trocam os seus vetores para que o outro nó possa avaliar se deve ou não repassar uma mensagem baseada nos valores do vetor recebido. Se a probabilidade de entrega for maior, a mensagem será repassada. Os valores são atualizados a cada encontro através de um método chamado *incremental averaging*. O valor referente ao nó que é encontrado é incrementado e todos são normalizados para que a soma seja 1.

2.3.3 Protocolos baseados em padrões de movimento

Após vencida a ideia de que o movimento humano não é aleatório, procurou-se estudar suas propriedades para melhorar os protocolos de encaminhamento de mensagens. Percebeu-se que o ser humano em sua jornada diária realiza determinados movimentos que são rotineiros e bem delimitados no espaço. Em [15] é apresentado um protocolo de encaminhamento geográfico em que o cenário é dividido em zonas e, a medida que os nós realizam seus trajetos e vão visitando as zonas, vão guardando as identificações das zonas por onde passaram. Essa informação é trocada a cada encontro e os nós têm a possibilidade de estimar se é mais vantajoso repassar a mensagem ou não, baseado na distância entre as zonas frequentadas pelo nó encontrado e pelo nó destinatário da mensagem. A ideia é ir conduzindo a mensagem para a região geográfica frequentada pelo destinatário.

MobySpace [25] é outro protocolo que se baseia no padrão de movimento dos nós. MobySpace define o conceito de MobyPoint que é um valor representativo do movimento realizado pelo nó. A cada encontro, os nós trocam a informação dos valores de seus MobyPoints para decidirem se têm mensagens para encaminhar através do nó encontrado. A ideia é que quanto mais próximo o valor de MobyPoint de dois nós, maior a probabilidade de que os nós tenham padrões de movimentação parecidos e possam se encontrar. Assim, a mensagem é encaminhada aos nós com valores cada vez mais próximos do MobyPoint do destinatário da mensagem.

2.3.4 Protocolos baseados no contexto social.

Com a premissa de que os movimentos humanos não são aleatórios e obedecem aos hábitos sociais, os protocolos sociais usam a teoria de redes complexas [26] para representar o grafo de contato dos nós através de seu histórico de contatos e, com isso, extrair informação que possa auxiliar na escolha de melhores nós para repassar as mensagens.

A estratégia de encaminhamento de mensagens utilizada pelo SimBet [27] é baseada na centralidade dos nós. A centralidade de um nó em redes representa a importância desse nó para a rede no que se refere a capacidade desse nó em interligar outros nós. Ela pode ser inferida através de vários indicadores, extraídos do grafo obtidos do histórico de contatos. Em [28] e [29], são propostos vários indicadores de centralidade em grafos. O indicador utilizado pelo SimBet é o *betweennees.*

Betweenness de um vértice m de um grafo V é soma da razão entre o número de caminhos mínimos que utilizam m e do número de caminhos mínimos para todos os pares de vértices em V, podendo ser expresso através da formula:

$$Betweeness = \sum_{v,w \epsilon V} \frac{C_m v, w}{C v, w} \qquad (2.1)$$

Na Equação 2.1, $\mathbf{C_m v, w}$ representa o número de caminhos mínimos entre os vértices v e w que utilizam m e $\mathbf{Cv,w}$ representa o número total de caminhos mínimos entre os vértices v e w.

Além do *betweenness,* outro conceito importante para o SimBet é o de similaridade entre os nós. Aqui a similaridade é a medida da proximidade dos nós capturada através do número de vizinhos de cada nó que mantém contato entre si. O grau de separação entre dois nós pode indicar que a difusão da informação pode ser demorada em caso de alto grau de separação. Assim, essa propriedade permite escolher

quais nós são mais propensos a entregar a mensagem aos seus vizinhos. Essa teoria foi estudada em [30] o que permitiu estimar a probabilidade de dois autores colaborarem no futuro analisando as suas publicações e os seus coautores.

Outro protocolo cuja estratégia de encaminhamento de mensagens é baseada na centralidade dos nós é o BUBBLE Rap [31] que agrupa os nós em comunidades. A decisão de repassar uma mensagem a um outro nó durante um encontro é baseada na centralidade dos nós. Todo nó pertence ao menos a duas comunidades: a global e uma comunidade local. Assim, quando uma mensagem é criada, ela é passada para nós com maior centralidade global até que ela alcance algum nó que pertença a mesma comunidade do destinatário. Essa primeira fase é chamada de *bubble-up*. Após chegar a comunidade de destino, o protocolo passa a disseminar a mensagem para nós com maior centralidade local, com o intuito de entregar a mensagem a nós com maior probabilidade de encontrar o destinatário.

Peoplerank [32] é o protocolo baseado no algoritmo Pagerank [33] utilizado pelo Google. Pagerank funciona partindo da ideia de que a probabilidade de se chegar a uma página com muitos hyperlinks que apontam para ela é maior a partir de uma outra página qualquer. Assim, Peoplerank utiliza a mesma ideia ao criar uma escala para os nós esperando que a mensagem chegue aos nós com maior índice e de lá ate o destino final. Quando dois nós se encontram, eles trocam os seus valores Peoplerank e o número de vizinhos que eles têm. Com isso, é possível estimar a importância que o nó tem na rede através de suas ligações. O valor de Peoplerank é dado pela seguinte equação:

$$\mathrm{PeR}(N_i) = (1-d) + d \sum_{N_j \in F(N_i)} \frac{PeR(N_j)}{|F(N_j)|} \tag{2.2}$$

Na equação 2.2, F(Nj) é o conjunto de nós vizinhos a Nj e d é o fator de corte que serve para controlar a aleatoriedade ao repassar mensagens. Caso o valor de d seja 0, todos os nós terão a mesma probabilidade de receber as mensagens para serem encaminhadas. Caso seja 1, dará preferência aos nós cujos valores de Peoplerank sejam maiores.

SimBetTS [34] é um protocolo proposto que utiliza as métricas similaridade, *betweenness* e força do laço para tomadas de decisão em encaminhamento de mensagens na rede. Diferentemente da forma como é calculado o *betweenness* por Freeman [28] [29], SimBetTS utiliza o conceito de Ego Networks [35] que é uma rede formada por um determinado nó (ego) e os nós diretamente conectados a ele. A justificativa para tal é que o cálculo das medidas de centralidade utilizadas pelo protocolo em suas decisões pode ser feita localmente, sem conhecimento de toda a estrutura da rede com seus nós e arestas. Assim, o valor de *betweenness* proposto é calculado através dos contatos que o nó ego tem com seus vizinhos e os contatos entre estes. Com o histórico de contatos, é criada uma matriz M(N x N) onde n é o número de vizinhos e os valores para cada elemento N_{ij} da matriz será 1 se houve contato entre os nós i e j ou 0 do contrário. Completada a matriz, o valor de *beetweenness* será dado pelo somatório dos recíprocos de $M^2[1 - M]_{ij}$.

Já a similaridade de um nó A com um nó B é calculada simplesmente através da intersecção do conjunto de nós vizinhos dos dois nós em questão.

A força do laço entre dois nós é o cálculo conjunto de vários fatores que podem influir positivamente ou negativamente a avaliação da relação entre os dois nós. No SimBetTS, são levados em conta:

- Frequência – Indicador de quantas vezes dois nós se encontraram em um determinado período em relação a todos os demais encontros do nó.
- *Closeness* – Esse indicador procura representar a duração dos encontros entre dois nós. Quanto maior for a duração, mais positivamente será a influência.
- *Recency* – Para ser valioso um enlace entre dois nós, é preciso que a última transação tenha sido recente, ou seja, que em cada momento de avaliação, o último encontro não tenha ocorrido há muito tempo atrás.

2.4 Peer to Peer e troca de arquivos

O paradigma cliente-servidor é uma arquitetura de aplicação distribuída onde um ou mais servidores oferecem um serviço que será consumido por um ou mais clientes. A comunicação entre eles é feita por mensagens de requisição enviadas pelos clientes (solicitando o serviço) e de respostas (enviadas pelo servidor com o resultado da execução do serviço). Um exemplo desse paradigma é o serviço de oferta de páginas na World Wide Web ou WWW. Nesse serviço, um servidor aguarda por requisições feitas através do protocolo HTTP de clientes que em geral são navegadores como Firefox ou Chrome. Estes programas têm como principais funções recuperar e exibir o conteúdo desejado. Embora a aplicação servidora e a aplicação cliente possam existir na mesma máquina física, os papeis cliente e servidor continuam bem definidos. Clientes solicitando e consumindo os recursos que são compartilhados pelos servidores.

Peer to Peer ou P2P é outro paradigma arquitetural onde essas duas funções, requisitar e oferecer serviços, são desempenhadas pelo mesmo componente [36]. As redes P2P são formadas pelos pares ou *peers* que são equipotentes em relação a aplicação e não desempenham uma relação assimétrica como no modelo cliente-servidor. A proposta de redes P2P é de criar uma topologia de redes onde as funções e

responsabilidades estão descentralizadas na rede. Algumas tarefas se tornam mais difíceis de serem completadas em uma arquitetura assim. Buscar informações sobre os recursos compartilhados é uma dessas tarefas. Enquanto no modelo cliente-servidor o local onde residem as informações sobre os recursos é único e localizado em um ponto só, na arquitetura P2P essa informação está distribuída. Entretanto, existem arquiteturas híbridas, onde há uma mistura de P2P com cliente-servidor. O Bittorrent tem uma arquitetura mista ao utilizar rastreadores que são servidores com endereços de outros pares.

Uma característica do modelo P2P é a sua capacidade de auto-escalar a medida que novos nós ingressem na rede pois se eles aumentarão o número de requisições aos outros nós, também disponibilizarão seus recursos para a rede, aumentando a capacidade de processamento, largura de banda e armazenamento. Outra característica é que P2P não necessita de nenhum atributo especial de hardware, sendo muitas vezes descrita como uma solução barata em comparação com o paradigma cliente-servidor onde os recursos de processamento e armazenamento estão concentrados nos servidores.

Por outro lado, P2P tem algumas particularidades que podem se tornar problemas. A questão da segurança, assimetria de *downstream* e *upstream* nos links fornecidos e colaboração [36] são desafios que devem ser considerados quando esse modelo for utilizado.

2.4.1 Bittorrent

Uma área de aplicações onde o P2P é bem sucedido é a de troca de arquivos [37] onde usuários compartilham suas bibliotecas de arquivos multimídia entre outros usuários através de programas que gerenciam as conexões e a origem do conteúdo. Bittorrent é um protocolo de troca de arquivos em redes P2P baseado em incentivos e

permite que cada par que esteja recebendo também possa enviar partes do arquivo já recebidas, mesmo que esse par ainda não tenha terminado de adquirir o arquivo por inteiro. Dessa forma, o Bittorrent aumenta a largura de banda disponível para a oferta do arquivo, pois todos os pares que estejam requisitando partes desse arquivo também estão oferecendo as partes que possuem. Arquivos mais populares oferecem muito mais banda devido à procura por esses arquivos ser maior e, por consequência, existirem mais pares requisitando e oferecendo. Cria-se assim, um interessante mecanismo para melhorar a escalabilidade da rede formada por clientes Bittorrent.

Para obter determinado arquivo através do Bittorrent, um cliente deve primeiro conseguir as informações iniciais sobre o arquivo desejado. Essas informações podem estar contidas em um meta-arquivo conhecido como *torrent* ou através de um *magnet-link*. Dentre as informações iniciais, estão os endereços (*urls*) dos rastreadores (*trackers*) que são os hospedeiros responsáveis por manter e disponibilizar as informações para se conectar aos pares pertencentes ao enxame (*swarm*) - grupo de pares que estão oferecendo e/ou recebendo

um determinado arquivo em determinado momento. Como pares se conectam e se desconectam a todo o momento, as informações sobre o enxame mudam constantemente. Em intervalos regulares de tempo, os pares enviam estatísticas para o rastreador de quanto receberam e de quanto enviaram para o enxame. Assim, o rastreador pode atualizar a lista dos pares conectados.

Uma vez que o cliente Bittorrent é iniciado e alimentado com as informações iniciais (Figura 2.3 - a), irá requerer ao rastreador uma lista de pares que estão no enxame (Figura 2.3 - b) e, depois de obtida essa lista, o cliente se conectar a esses pares e começar a requisitar partes do arquivo (Figura 2.3 - c). Ao final (Figura 2.3 - d), o arquivo estará completo no hospedeiro.

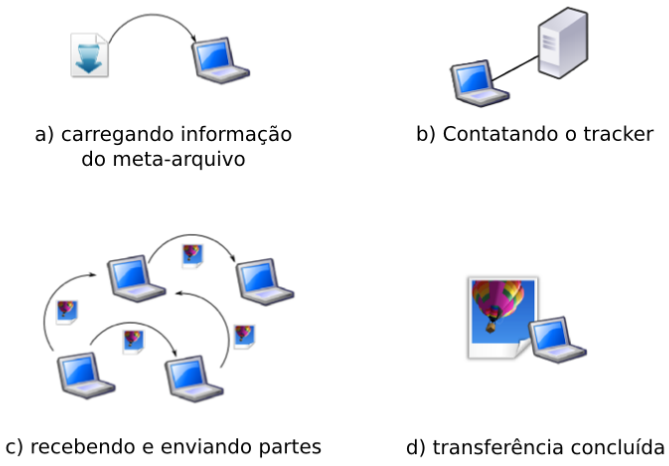


Figura 2.3 - Fases do processo de obtenção de um arquivo com o Bittorrent

A menor parte do arquivo requisitada pelo cliente se chama bloco (*block*). O bloco não tem tamanho definido, embora, em geral, tenha 2^14 bytes nas implementações mais novas e 2^15 bytes nas implementações antigas. Um ou mais blocos formam um pedaço (*piece*) que, após ser completado, é verificado através da assinatura SHA-1 contida no meta-arquivo. Uma vez completado e verificado, esse pedaço passará a ser oferecido também por esse par, aumentando a disponibilidade desse pedaço.

Os pares são divididos entre sugadores (*leechers*) e semeadores (*seeders*). A diferença básica entre esses dois tipos é que semeadores possuem uma cópia completa

do arquivo distribuído e continuam conectados ao enxame, oferecendo o arquivo mesmo após completarem o *download* do mesmo. Esse processo se chama semeadura (*seeding*). Os sugadores, ao contrário dos semeadores, não possuem o arquivo completo e tendem a se desconectar logo que finalizam o *download*, retirando essa cópia do enxame.

2.4.1.1 O meta-arquivo (*.torrent)

Esse é o arquivo de configuração que detém todas as informações necessárias para que um cliente Bittorrent possa ingressar no enxame. Todas as informações inclusas no meta-arquivo estão codificadas em *bencode* [38]. Dentro do arquivo estão as assinaturas SHA-1 de cada pedaço do arquivo para se verificar a integridade do pedaço retirado do enxame. Cada assinatura tem 20 bytes de extensão. Em geral, a distribuição desse arquivo se faz através de um servidor web, mas pode ser compartilhado através de outros meios.

Atualmente, tem se tornado popular outro sistema de distribuição das informações iniciais do enxame. Uma dessas formas é através de *magnet-links* que são hyperlinks HTML com semântica e sintaxe próprias contendo as informações de localização dos rastreadores conforme pode ser visto abaixo.

magnet:?xt=urn:btih:c49b1c95e2d1b5f42f76276e04eb47e418e2edaf&dn=descrição do arquivo&tr=udp://tracker.openbittorrent.com:80&tr=udp://tracker.publicbt.com:80&tr=udp://tracker.istole.it:6969&tr=udp://open.demonii.com:1337

Tabela 2.1 Exemplo de magnet-link.

2.4.1.2 Rastreador

O rastreador desempenha um papel importantíssimo na arquitetura do Bittorrent por fornecer o endereço dos nós participantes do enxame. É um servidor que, além de ser um facilitador na comunicação entre os pares, também recebe e disponibiliza

estatísticas globais sobre o *torrent*. Em geral, são configurados vários rastreadores para cada *torrent* como medida de segurança.

Um par para começar a solicitar partes do arquivo, inicialmente, irá solicitar ao rastreador uma lista de endereços IPs de pares que estão atualmente no enxame. Só após receber essa lista que ele irá se conectar aos pares. De tempos em tempos, à medida que o par atualiza as suas estatísticas locais - bytes recebidos (*downloaded*) e enviados (*uploaded*) - ele envia esses valores para o rastreador para que sejam disponibilizados globalmente.

Existem outras formas para se conseguir os endereços IPs dos pares participantes do enxame sem o uso de um rastreador. O DHT e o PEX são as formas utilizadas para se construir uma rede Bittorrent *trackerless* (sem rastreadores) [39]. Para sinalizar que o cliente suporta as funcionalidades de DHT, durante o *handshake,* ele deverá marcar o último bit do oitavo byte reservado. Com isso, cada cliente Bittorrent que suporte essa funcionalidade passará a desempenhar as funções do rastreador, armazenando listas de endereços IPs de outros pares que estarão disponíveis para consulta.

3 Trabalhos relacionados

Neste capítulo, será apresentado ao leitor os trabalhos anteriores que propuseram aplicações em rede oportunistas. Primeiro serão vistos trabalhos com aplicações voltadas para cenários de emergência ou desastre, um forte motivador para a utilização de redes oportunistas. Nessa primeira parte, serão vistas também aplicações voltadas para outras áreas. Depois, serão apresentadas aplicações voltadas para troca de arquivo, objetivo deste trabalho.

3.1 Aplicações em redes oportunistas.

Inicialmente, aplicações para redes oportunistas eram voltadas para cenários de emergência ou desastre. Esses cenários são caracterizados pela falha ou destruição da infraestrutura de comunicação e impossibilidade de deslocamento pelo transporte público convencional e estradas ou rodovias. Neste tipo de cenário, é necessário que os pedidos de socorro cheguem até as equipes de salvamento e, assim, as orientações para deslocamento ou socorro cheguem até a população atingida.

Em [40], é proposto um sistema de navegação para evacuação de prédios onde o cenário de emergência está em curso, em um mapa bidimensional. Uma evolução desse sistema de navegação são as propostas em [41] onde o mapa passa a ser considerado em três dimensões com escadas para alcançar outros pavimentos e em [42] no qual é considerado a dinâmica de espalhamento da ameaça como fogo ou fumaça no interior do prédio através do sistema FireGuard. Já em [43] e [44] é levada em consideração

congestionamentos e gargalos possíveis nas rotas de fuga e ajuda para as equipes de salvamento chegarem a determinado ponto crítico do prédio. A proposta apresentada em [45] é baseada em *smartphones* usando o sistema Android e chaves públicas para assinar com criptografia de chaves assimétricas para impedir tentativas de falsos chamados de emergência, onerando as equipes de busca e salvamento e se destina a áreas maiores, onde a emergência a ser contornada não está confinada a prédios ou demais espaços fechados.

Com o avanço do conhecimento na área, novas propostas apareceram como alternativas para aplicações fora da situação de emergência e desastre. Aplicações de cunho social são desenvolvidas para mostrar a viabilidade do modelo de mobilidade e/ou estratégia de encaminhamentos de mensagens dos trabalhos propostos. Em [46], é proposta uma aplicação que permite ver os seus contatos próximos ao usuário utilizando o protocolo de encaminhamento Epidêmico. Também é proposta uma arquitetura e uma API que outras aplicações possam utilizar a nova camada proposta. Mais que uma aplicação, [47] é um *framework* para desenvolvimento de aplicações em redes oportunistas. Para validar a proposta, é implementada uma aplicação de troca de mensagens e perfil social sobre a plataforma MobiClique, usando os seus serviços. Dentre os *frameworks*, destacamos também Haggle [48], que é uma iniciativa pioneira em implementar soluções para troca de conteúdo em redes oportunistas, fornecendo ao desenvolvedor uma abstração da tecnologia de rádio a ser utilizada na transmissão e uma interface para o controle da comunicação assíncrona. Uma vez que a desconexão em redes oportunistas pode levar a grandes atrasos na comunicação, aplicações que se utilizam de padrões de comunicação baseados no modelo requisição/resposta podem ser sensíveis a esse atraso e o conteúdo buscado pode estar em posse de um nó específico e não distribuído ao longo da rede.

Com o intuito de oferecer serviços a outras aplicações, mas com conceito ligeiramente diferente dos *frameworks*, temos os *middlewares* que são sistemas independentes, fornecendo integração entre redes oportunistas e outros serviços em outras redes como a Internet, por exemplo. Dentro desse contexto, é apresentado em [49] uma arquitetura que foi usada para recepcionar mensagens de visitantes ao campus da universidade para posterior publicação na aplicação Twitter residente na Internet.

3.1.1 Aplicações de troca de arquivo

A medida que o conceito de redes oportunistas foi emergindo de redes MANET, as propostas de aplicações que permitiam troca de arquivos acompanharam a formação do novo conceito. No início, assim como se pretendia manter o conhecimento sobre a topologia da rede para poder estabelecer rotas fim-a-fim, as propostas de aplicações desta natureza propunham uma estrutura sobre a camada de aplicação que manteria o conhecimento da distribuição de conteúdo ao longo da rede. Essa estratégia é conhecida como P2P estruturado em razão da criação de uma estrutura auxiliar para funcionamento das buscas pelo conteúdo na rede. Em [50] é proposto um mecanismo de *query* e recuperação de arquivos baseado em uma estrutura montada através de mensagens para busca de conteúdo. A cada mensagem *query* para descobrir a localização dos arquivos na rede, a estrutura vai sendo atualizada. Cada nó mantém uma tabela com o próximo nó da rota do arquivo. Essa tabela é mantida com as informações oriundas das respostas das mensagens *queries* para descoberta de conteúdo.

Em [51], é apresentado um esquema de persistência, busca e recuperação de arquivos em uma rede MANET. Utilizando particionamento de arquivo para melhorar as transferências, são selecionados nós com grande espaço de armazenamento para guardar as mensagens quando uma transferência for afetada em qualquer parte do

caminho. Quando a conectividade for reestabelecida, o arquivo é recuperado. A pé-configuração dos nós é um ponto crítico para a proposta, pois, além de selecionar os nós que funcionarão provendo o serviço de armazenamento, também é necessário que todos os nós conheçam o endereço de todos os demais nós existentes na rede. Esses dois requisitos aliado a presunção de rotas fim-a-fim em redes oportunistas/DTN parece reduzir bastante o cenário de atuação dessa solução. Outro fator limitante, é que necessita para uma melhor performance da existência de nós estáticos. Não faz qualquer menção ao tamanho máximo do arquivo que esse sistema poderia suportar, nem os requisitos necessários dos dispositivos móveis a serem utilizados.

Em [52], é proposto um sistema chamado 7DS (7 degrees of separation) cujo objetivo é prover conectividade à Internet através da cooperação entre os nós e um mecanismo de disseminação de dados entre nós móveis. Quando um nó tenta acessar uma página na Internet, por exemplo, e não tem conectividade com a rede, o sistema pode requerer a página de outros nós próximos que também executem o 7DS. Existem três tipos de mensagens para interação entre os nós. As *queries* que são solicitações de dados entre os nós e são identificadas pela URL do objeto solicitado e pelo endereço MAC do nó que originou a *query*. Ao receber uma query, o nó procura em seu cache pelo objeto solicitado e, caso encontre, envia um *report* para o outro nó com o objeto desejado. Outra mensagem usada pelo sistema são os *advertisements* que são utilizados para informar a presença de servidores 7DS.

Para conservar a energia, 7DS propõe um modo de operação sincronizado onde os dispositivos desligariam suas interfaces de rádio e toda a comunicação ficaria agendada para ser executada no próximo momento em que a interface estivesse ligada. Para tal modo funcionar, os dispositivos precisam sincronizar seus relógios através de um GPS ou de dispositivos que tenham GPS. Os sistema 7DS oferece suporte a criptografia

através de chaves públicas e só envia objetos marcados como privados através desse modo. Para validar o trabalho, foi usado um cenário de 1000m X 1000m contendo 5 a 25 nós com alcance de rádio entre 55 e 230m. Esses alcances foram conseguidos através de transmissões com potência entre 1.1mW e 281.8mW. Para fins de comparação, o dispositivo WAP54G (um ponto de acesso fixo, conectado a rede elétrica) da Cisco usa 13.5dBm, aproximadamente 22mW para transmissão [53]. Considerando a área de alcance do rádio dos dispositivos circular e 230m de alcance das estações, bastam 6 dispositivos para cobrir toda a área do cenário proposto. Dois pontos que podem trazer dificuldades na configuração de um dispositivo para ingressar na rede é a obtenção de chaves públicas dos outros nós para criptografia das transmissões de objetos privados e sincronização dos relógios para os momentos de inatividade de transmissões.

Com os pesquisadores definindo os conceitos por trás do oportunismo, as propostas que se seguem passam a tentar explorar o novo paradigma de rede utilizando em suas propostas técnicas e/ou meios diferentes que favoreçam a troca de dados. Alguns trabalhos sugerem a adoção de IEEE 802.11 em modo *ad hoc* por alcançar maiores taxas de transferência ou usar Bluetooth por gastar menos energia dos dispositivos móveis. Ou fazer uso de redes híbridas, com pontos de acesso ou nós fixos para melhorar a troca de dados e acelerar a fase inicial de disseminação de conteúdo.

Em Bluetorrent [54], é proposta uma solução de disseminação de conteúdo através da implementação do protocolo Bittorrent em ambiente móvel com o protocolo de acesso ao meio Bluetooth. Assim como Bittorent, Bluetorrent mantém registros de bons nós por um determinado tempo para favorecê-los em outra oportunidade, mas a escolha de bons nós é realizada em parâmetros opostos. Para o Bluetorrent, os bons nós são aqueles com registros de menor frequência de contatos e duração média de conexão baixa, pois isso favorece a descoberta de novos nós uma vez que o tempo para se

conectar a um nó é longo e similar ao tempo de se descobrir novos nós usando Bluetooth. Já o Bittorent mantém registro dos outros nós no enxame para determinar os nós que são mais vantajosos manter. Isso significa, escolher os nós menos egoístas, ou seja, os nós que transmitiram mais pedaços do arquivo. Enquanto [54] pretere os nós com maior duração de conexão, Bittorrent procura mantê-los.

Bluetorrent procura estimar a duração de contato para cada encontro. Isso depende da direção e velocidade de deslocamento dos nós. Para tal, Bluetorrent estima a distância entre os nós através do RSSI enviado durante a fase de *inquiry* do Bluetooth que foi modificado para durar 1.28 segundos. Não fica claro no trabalho se de alguma forma conseguiram modificar a implementação do Bluetooth para conseguir realizar *inquiry* e conectar a outros nós ao mesmo tempo, pois para realizar o *inquiry*, o dispositivo precisa estar em *Inquiry State* o que não permite outras operações. Segundo os autores, com duas pessoas se movendo em direções opostas a 1m/s, Bluetooth poderia manter um link máximo teórico de 10s com alcance de rádio de 10m. Descontando o tempo estimado de *inquiry* e conexão em torno de 4s, seria possível transferir até 286KB de dados nesse encontro. O trabalho foi validado através de simulação onde os nós eram dispostos em corredores, artificializando o movimento dos nós já que estão confinados em um ambiente muito específico. Não fica claro no trabalho se todas as mensagens do Bittorrent foram implementadas, pois as estratégias de *choke/unchoke* e *interested/not interested* não seriam interessantes em um ambiente oportunístico.

Um sistema de distribuição de podcasts é proposto em [55] utilizando Bluetooth para disseminação dos canais de informação. O conteúdo é obtido da Internet através de pontos de acesso instalados ao longo da área. Uma vez que um nó consiga recuperar o *podcast,* passa a distribuí-lo aos solicitantes. A área do cenário é de 300m x 300m com

um ponto de acesso ao centro. Outro trabalho que se utiliza de uma estratégia mista com nós fixos é o R-P2P [56]. A arquitetura proposta usa três tipos de dispositivos de comunicação: *throwboxes*, nós-usuários e fontes. Nós-usuários são dispositivos móveis como telefones ou PDAs. O conteúdo pode vir dos denominados fontes que são computadores com interface de rede para se conectar ou gerado pelos nós-usuários. Os *throwboxes* formam uma rede IP estática com grande capacidade de armazenamento com o intuito de disseminar e manter os dados disponíveis. Possuem uma segunda interface que permite a comunicação com os nós usuários. Quando um nó-usuário transfere o conteúdo para um *throwbox*, deleta de seu sistema de arquivos local.

Um modelo que não utiliza nós estáticos intermediários é proposto em [15] sobre os padrões de movimentação dos nós em um cenário realístico. Com esse modelo, um protocolo de encaminhamento de mensagens é criado para prover uma rede P2P de forma que não dependa de um serviço ou infraestrutura. O trabalho não faz qualquer menção à característica dos dispositivos clientes dessa proposta nem de que tipo de arquivos ou tamanhos dos mesmos seria suportado. Falta também maiores informações sobre a camada de aplicação.

Em [57] é apresentada uma proposta de um *middleware* que não necessita de infraestrutura para prover serviços a camada de aplicação. Como outras propostas, uma classificação de conteúdo é criada para organizar as entradas e facilitar a busca. Seguindo a estrutura de Atom Syndication Format que é uma especificação de um dialeto XML para descrever fluxos de arquivos distribuídos pela Web, o conteúdo é agrupado em *feeds* que são simples containers para armazenar as entradas que mantém os dados dos arquivos. Estes são transferidos em *chunks* de 16KB, valor apontado como melhor equilíbrio entre sobrecarga e probabilidade de recepção incompleta, mas o embasamento para se chegar a esse valor foi omitido do trabalho. Um modelo de

camadas de protocolo é apresentado com algumas variações ao tradicional modelo de 5 camadas. Uma camada de sessão é incluída ao modelo entre as camadas de aplicação e transporte para oferecer uma API às aplicações e acesso aos módulos de descoberta de serviço e transporte através do gerenciador de sincronização. As camadas físicas e de enlace são unidas em uma. Embora o cenário de atuação do trabalho seja redes oportunistas a validação consistiu em um conjunto de nós estacionários.

Em [58], uma aplicação é proposta para troca de arquivos de músicas entre usuários do metrô de Londres usando Bluetooth. Esse trabalho explora um sistema de categorização de conteúdo através das *tags* e meta-informações de arquivos de música obtidos a partir das listas de execução dos usuários do sistema Last.fm. A partir da lista de execução e das preferências musicais de 500 mil usuários do sistema, foram extraídos os 50 artistas mais populares e destes, as 50 *tags* mais populares associadas de forma a poder classificar e medir a popularidade destes gêneros em relação a todo o sistema. O trabalho utiliza essa caracterização para poder distribuir os arquivos entre os usuários durante a simulação. Se por um lado essa iniciativa se mostra interessante para poder aproximar da realidade a distribuição de conteúdo ao longo da rede, há uma diminuição dos estilos musicais ofertados de tal maneira que favorece o encontro de conteúdo entre os nós pois há menos opções do que no mundo real. O trabalho também procura utilizar de dados reais de mobilidade de usuários para simulação. Os dados utilizados são de duas linhas do metrô de Londres onde foi introduzido o sistema de pagamento através de cartões com RFID. Os cartões são utilizados para entrar nas estações e também para sair. Assim, o tempo de permanência do usuário no sistema pode ser determinado, faltando estimar se os usuários estão em contato ou não dentro do sistema. Para isso, o trabalho estima se dois usuários estão juntos no mesmo trem a partir de suas saídas do sistema e se estão viajando na mesma linha e direção. Se a

diferença de tempo de saída do sistema for inferior ao intervalo entre dois trens consecutivos, presume-se que os usuários viajaram no mesmo trem. Com base nessa premissa, os usuários que viajaram juntos são distribuídos uniformemente no trem durante a simulação. Essa simplificação dos encontros esconde muitos detalhes do movimento real dos seres humanos motivados por seus interesses. Nesse caso, em particular, não leva em consideração as diferentes velocidades de movimentação dos usuários por fatores como idade, acessibilidade, saídas distintas do sistema (saídas de estações) ou outros fatores que possam levar os usuários a pararem na estação final como serviços oferecidos ou pontos de encontros.

Com o grande aumento de conteúdo gerado por dispositivos móveis, uma acentuada preocupação com o volume de tráfego poder vir a gerar sobrecarga nas redes celulares [11] tem feito com que os pesquisadores proponham soluções oportunistas para disseminação de conteúdo entre dispositivos móveis, diminuindo o tráfego nas redes celulares. Uma estratégia que vem sendo bastante discutida e tem sido apresentada em muitos trabalhos é a de *offloading* que é a utilização de outras redes de comunicação de dados complementares para transmissão de dados móveis. Em [59] e [60] são apresentadas propostas neste sentido.

4 Problemas relacionados às aplicações de troca de arquivos em redes oportunistas

Neste capítulo, iremos apresentar alguns problemas relacionados a uma aplicação de troca de arquivo em redes oportunistas. Iniciaremos abordando o problema da representação vazia da camada de aplicação. Depois apresentaremos as questões sobre localização do conteúdo, de como aproximar o conteúdo do usuário e acelerar a disseminação de partes do arquivo em uma rede oportunista.

Como visto na Seção 3.1.1, existem muitas propostas para prover troca de arquivos em redes *ad hoc* móveis. Contudo, há algumas decisões nos trabalhos que levam a questionamentos sobre as propostas apresentadas nesses trabalhos e o quão efetiva são em prover a solução como alegam. Considerando o objetivo do presente trabalho em verificar a viabilidade de uma solução de troca de arquivo em redes oportunistas independente de infraestrutura e utilizando a capacidade atual dos dispositivos móveis modernos como *smartphones*, faz-se necessário não escolher cenários irreais, não propor soluções ou técnicas que não sejam triviais de implementar em redes oportunistas, não se valer de qualquer tipo de dispositivo fixo especial que possa facilitar os contatos ou manter as mensagens em determinado local de uma infraestrutura como em soluções híbridas e não ignorar que as aplicações precisam ser repensadas para funcionarem no contexto oportunístico. Dos trabalhos apresentados na seção anterior, podemos classificá-los conforme essas observações da seguinte forma.

#	Observação	Descrição	Trabalhos
1	Aplicações desenvolvidas para redes MANET	Trabalhos que se baseiam na premissa de conexão fim-a-fim.	[50] [51] [52]
2	Suposições muito restritivas no cenário oportunista.	Operações complexas de serem realizadas no contexto oportunístico. Por exemplo, conhecer todos os dispositivos existentes.	[51] [52] [54] [57] [58]
3	Falta de importância da camada de aplicação ou não-adequação do aplicativo	Ignorar a importância da camada de aplicação ou não adaptar os aplicativos ao contexto oportunista.	[50] [51] [52] [54] [55] [56] [58] [15]
4	Cenários irreais ou com parâmetros exagerados	Utilizar de premissas como nós estacionários, movimento aleatório dos nós, tamanhos de cenários pequenos ou alcance do rádio exagerado.	[52] [54] [55] [57]
5	Dependência de infraestrutura	Utilizar de qualquer infraestrutura (redes, *throwboxes*, sistemas externos, etc) para melhorar a troca de mensagens ou fomentar encontros.	[55] [56] [15]

Tabela 4.1 - Classificação dos trabalhos de acordo com não-aderência aos objetivos do presente trabalho

Em relação às propostas que foram classificadas com os itens 1 e 5 da Tabela 4.1, é entendido que são características dos trabalhos apresentados e simplesmente não se enquadram nos objetivos perseguidos no presente trabalho. Quanto aos trabalhos classificados com os itens 2 e 4, é possível que o trabalho estivesse em um momento de transição do conhecimento da área ou da tecnologia no momento de desenvolvimento e hoje pareça não razoável os conceitos aplicados. Assim, como nas simulações realizadas no início da pesquisa da área, o movimento dos nós era feito através do modelo de mobilidade Random Waypoint. Desde então, adquiriu-se maturidade ao longo do tempo para perceber que isso era inadequado para validar os trabalhos e outros conceitos

podem também ter sido tentados dentro do contexto de redes oportunistas, mas não conseguiram se solidificar e prosperar. Trataremos do caso das propostas que não adaptaram os aplicativos para a realidade oportunista ou não procuraram explorar a camada de aplicação com estratégias para fomentar o sucesso das aplicações.

4.1 Representação vazia da Camada de Aplicação

Como exposto em [61], uma maneira de tornar redes ad hoc multisalto uma realidade é eliminar a presunção de ser uma rede de propósito geral e criar redes e aplicações especializadas. A grande mobilidade dos nós no contexto oportunista faz com que seja desafiador a transmissão de dados com vários saltos. Estratégias e otimizações têm sido criadas e testadas com o intuito de mitigar a falta de conectividade ocasionada pelo deslocamento entre os nós e melhorar a performance de aplicações em um cenário com muitas restrições como, por exemplo, a carga de energia disponível em dispositivos móveis como *smartphones*.

Dentre as otimizações, há uma mudança de visão de como pensar soluções para redes oportunistas com a estratégia de desenvolvimento orientado às aplicações [62] onde seriam analisadas as necessidades das aplicações antes da construção dos demais componentes, justamente para que as soluções técnicas criadas possam atender aos requisitos enumerados antes. Assim, é inevitável que ao se pensar em uma solução para troca de arquivos em redes oportunistas, leve em consideração as estratégias que possam ser empregadas na camada de aplicação para melhorar a taxa de sucesso. Em muitos trabalhos que propõem soluções de aplicativos reais para o usuário final, as aplicações são meros geradores de tráfego. Não se preocupar com a camada de aplicação não é um erro em si, mas é desprezar um conjunto de possibilidades para se alcançar sucesso na empreitada de implementação de aplicações em redes oportunistas.

4.2 informação da Localização do conteúdo

Em geral, quando é necessário realizar uma pesquisa ou recuperar alguma conteúdo, os usuários procuram usar um sistema de busca de informação onde um conjunto de palavras-chave vão retornar localizações do conteúdo pretendido através de algum mapeamento entre termos buscados e o conteúdo. Esse é um modelo muito utilizado na Web através dos buscadores ou motores de busca. Essa forma de realizar pesquisas funciona em cenários onde a conectividade tanto com o sistema de busca quanto a conectividade do cliente com o local onde o conteúdo está armazenado são presumidas. Nada adianta localizar a fonte se não for possível recuperar o conteúdo. Assim, sistemas de busca com servidores centralizados no contexto oportunista tem a eficácia reduzida em função da falta de disponibilidade do serviço em razão da falta de conectividade. Existem propostas de sistemas de busca distribuídos, mas o esforço para manter mapeamento de palavras-chave entre os nós constituintes do sistema de busca é tão grande quanto de manter a topologia de rede. Esses mecanismos criam uma estrutura com as informações de localização do conteúdo entre os nós e por isso são chamados de estruturados.

Além disso, com as redes sociais, o usuário não é mais apenas consumidor da informação. O usuário passou a produzi-la e publicá-la, entrando em uma outra categoria que vai além da classificação consumidor e fornecedor de informação. Com efeito, existe muito conteúdo que está fora de um servidor local e que não foi mapeado e não pode ser recuperado através dos meios tradicionais, sendo desejável um mecanismo de buscas que permita a consulta local, aos nós na vizinhança.

No contexto do presente trabalho, a informação de quem tem partes do arquivo deve ser recuperada para que as chances de obtenção aumente ao se criar requisições em paralelo a mais de um semeador ao mesmo tempo. O Bittorrent mantém uma estrutura

para que todos saibam o IP dos nós participantes do enxame. Assim que um nó participante do enxame obtém uma parte do conteúdo, ele passa a ser sinalizado dentro do enxame como detentor de partes do arquivo e com isso passa a receber requisições desse conteúdo.

Com a proposta de introdução de duas mensagens na comunicação entre os nós nesse trabalho, é possível fazer com que os usuários participantes do enxame possam receber as requisições e contribuir devolvendo o conteúdo solicitado caso possuam ou encaminhando a mensagem até alcançar quem tenha.

4.3 Como aproximar o conteúdo do usuário em redes oportunistas

No cenário oportunista, o tempo de transmissão não pode ser determinado pois depende da movimentação dos nós. Esse tempo é tão duradouro quanto o tempo em que as pessoas passam juntas e, assim, os dispositivos permanecem ao alcance um do outro e podem se comunicar. Isso significa que existe um limite para o número de bytes que podem ser transferidos durante um encontro. Se quisermos uma estratégia que permita uma transferência de arquivos de tamanho arbitrário, temos que considerar a possibilidade de aproveitar ao máximo o tempo de duração do encontro para transferir a quantidade de bytes possível e recuperar o restante em outra oportunidade. Mas será possível contar com outra oportunidade no contexto oportunista? Com que frequência os nós se reencontram?

Um experimento realizado pelo autor utilizou um mecanismo de coleta com interface Bluetooth que a cada 1 minuto realizava *inquiries* para descobrir dispositivos próximos pela região central da cidade do Rio de Janeiro. O mecanismo conseguiu detectar 8.774 dispositivos diferentes ao longo de 33 dias. Foram retornados 56.268

respostas aos *inquiries*, número superior ao número de dispositivos, o que indica que alguns dispositivos foram encontrados mais de uma vez ao longo do experimento.

Números de dias observados	**Percentual**
1	93,2%
2	5,1%
3	1,7%

Tabela 4.2 Percentuais de nós encontrados classificados de acordo com o número de dias.

Na Tabela 4.2, vemos que mais de 93% dos nós encontrados foram percebidos apenas em 1 dia ao longo do experimento, não voltando a ser encontrados.

Um mecanismo simples de troca de arquivos, onde os arquivos seriam passados com apenas um salto, dependeria muito do tamanho da janela de transferência disponível entre os nós, pois se não fosse possível transferir todo o conteúdo de uma vez, teria pouca probabilidade de reencontrar com o nó que possui o arquivo, de acordo com a Tabela 4.2. Mesmo que conseguisse reencontrar, poderia não reunir condições novamente para realizar a transferência com êxito, uma vez que o movimento dos nós entre si é fator determinante para o tamanho da janela. Outro ponto é que esse mecanismo não suportaria arquivos maiores do que pudesse transferir na janela de tempo máxima, pois o tamanho da janela depende de vários fatores para ser obtida como velocidade relativa entre os nós.

Arquivos populares, com grande número relativo de nós que o possuam, teriam mais oportunidades de serem transferidos, pois ocorreriam mais contatos. Assim, para não depender da popularidade do conteúdo para conseguir que o mecanismo funcione a contento, pode-se empregar alguns princípios oriundos de Redes de Distribuição de

Conteúdo (Content Delivery Network – CDN). Uma CDN é um conjunto de servidores distribuídos e interconectados que cooperam para melhorar a experiência do usuário na recepção de conteúdo. As duas principais técnicas utilizadas pelas CDNs é replicar conteúdo em algum ponto da rede que seja mais próximo e que não ofereça congestionamento para o usuário e redirecioná-lo para este novo servidor [37]. CDNs são uma boa solução para determinados conteúdos que se tornam virais e, de forma inesperada, demandam uma grande largura de banda em função de sua popularidade se tornar muito grande em pouco tempo ou para mitigar o efeito *slashdot* [63].

Assim, iremos utilizar do conceito aplicado em CDNs para favorecer o retorno de partes do arquivo. Como exposto em [64], é vantajoso que os nós da rede conheçam a semântica da aplicação para melhorar a performance e que aplicativos de troca de arquivo só podem considerar a transferência bem sucedida se todas as partes do arquivo também forem corretamente recebidas.

4.4 Disseminação rápida do conteúdo

Como dito na Seção 4.3, é importante aproximar o conteúdo do usuário para que se possa completar o arquivo mais rapidamente. Assim, é muito importante melhorar a distribuição do arquivo na rede. Dentro desse contexto, pode-se usar os nós intermediários para oferecer partes do arquivo ao invés da solicitação percorrer um caminho mais longo até chegar ao semeador e conseguir maior número de encontros onde exista partes do arquivo que se deseja transferir.

4.5 Considerações finais

Com base no que foi apresentado, iremos propor uma aplicação P2P no próximo capítulo que possuirá como característica a disseminação rápida do conteúdo, utilização dos nós intermediários para fornecer partes do arquivo e assim diminuir o número de saltos da requisição por partes do arquivo e que seja favorecida da redundância das partes do arquivo no enxame.

5 Proposta de aplicação de distribuição de conteúdo em redes oportunistas

A aplicação proposta no presente trabalho é baseada em algumas aplicações P2P, notadamente o Bittorrent. Distribuição de arquivos segmentada, descentralizada e que permite a imediata oferta de uma parte recebida por um nó são características do Bittorrent que vêm ao encontro de alguns requisitos desejados em aplicações de troca de arquivos em redes oportunistas. O particionamento do arquivo em blocos ou partes menores é um atributo interessante uma vez que as janelas de transmissão podem ser tão pequenas quanto o tempo de duas pessoas passando uma pela outra na rua e, assim, o arquivo vai sendo transferindo em partes a cada contato do nó com os outros nós detentores de partes do arquivo. Assim que recebe uma parte do arquivo, o nó passa de imediato a disponibilizá-lo, aumentando a sua oferta dentro do enxame. Isso permite uma rápida disseminação do arquivo ao longo da rede, criando mais oportunidades de transmissão aproveitáveis, pois aumenta as chances de um nó ter o que oferecer ao outro em um contato.

Nem todas as características do Bittorrent são reproduzíveis ou mesmo desejáveis no cenário de aplicações P2P em redes oportunistas. O algoritmo *rarest first*[2] induz na aplicação a tarefa de procurar primeiro por blocos do arquivo cuja frequência dentro do enxame seja menor para evitar o problema de não conseguir completar o arquivo por não encontrar as mesmas. Isso é impraticável em redes oportunistas pelo fato de não ser possível distinguir os nós participantes do enxame a cada momento e, na prática, um nó

oportunista não poderia preterir um bloco ofertado por nenhum critério a não ser já possuir tal bloco.

Outras situações que não são interessantes nesse cenário, é o mecanismo de incentivo *tit-for-tat* [65] praticado em Bittorrent. Através das mensagens *choke* e *not-interested,* o nó sinaliza para o outro nó conectado que ele não vai responder a nenhuma requisição oriunda dessa conexão ou não está interessado em nada que o outro possa oferecer, respectivamente. Esses estados podem ser mudados com as mensagens *unchoke* e *interested.* Isso permite uma série de estratégias e jogos para melhorar a taxa de obtenção de partes do arquivo através de uma conexão em especial e economizar recursos desnecessariamente, estimulando o outro lado a desconectar. Já no cenário oportunista não é possível gastar tempo com jogos pois o tempo de contato pode ser tão curto quanto duas pessoas se encontrando na rua ou mesmo veículos em movimento e também não há necessidade de gerenciar longas conexões pois, em geral, as conexões se desfazem devido a grande movimentação dos nós.

Diante do exposto, a seguir será descrita a aplicação proposta denominada Aplicação para Troca de Arquivos, ApTA, com suas características e respectivas motivações.

5.1 Funcionamento básico da aplicação

Uma vez iniciada a requisição do arquivo, a aplicação irá gerar pacotes de requisição de partes do arquivo a todos os semeadores conhecidos. Em cada requisição, existe o identificador do arquivo que se deseja obter e o número do bloco desejado. Se o nó não possuir nenhum bloco do arquivo, irá enviar uma mensagem de requisição RQP (ver Seção 5.5). Caso contrário, irá selecionar aleatoriamente um dos blocos faltantes para reiniciar o processo. Cada vez que um nó é encontrado, é verificado se mensagens

devem ser trocadas. Caso ocorra uma troca de mensagens e a mesma seja uma requisição, o nó receptor irá procurar no cabeçalho pelo identificador do arquivo e pelo bloco solicitado. Caso possua essa parte do arquivo, ele mesmo irá responder à solicitação devolvendo o bloco e não repassando a mensagem original (ver Seção 5.6).

5.2 Particionamento dos arquivos

A ideia de dividir o arquivo em blocos menores é a implementação da velha estratégia da computação de dividir para conquistar. Ao invés de tentar transferir todo o arquivo de uma vez como é feito normalmente no protocolo HTTP, em P2P a ideia é incrementar aos poucos o arquivo, recebendo partes do mesmo. Embora seja possível realizar downloads incrementais sobre HTTP através dos header *Range*, não permitiria o mesmo efeito. No Bittorrent, o particionamento é feito para melhorar a escalabilidade e não onerar o nó detentor do arquivo. Quando um nó adquire um bloco de um arquivo, ele passa a servi-lo, criando redundância e escalando a aplicação pois a cada novo cliente que entre no enxame a procura do arquivo, ele também oferecerá blocos do arquivo já recebidos enquanto estiver a procura dos demais. Essa característica do Bittorrent faz com que a aplicação escale independente do número de nós no enxame.

No cenário oportunista, isso não só permitirá a desoneração dos nós inicialmente detentores do arquivo, como também a oferta de processamento e largura de banda já que todo nó que esteja executando a aplicação irá transportar as mensagens recebidas. Além disso, em cenários onde o tempo de contato entre os nós for pequeno, a estratégia de dividir o arquivo em partes menores parece acertada, pois permitiria a transferência do arquivo aos poucos, aproveitando o pouco tempo de transmissão durante os contatos.

O particionamento do arquivo é realizado através de uma divisão lógica do tamanho do arquivo em bytes pelo tamanho em bytes do bloco. Todos os blocos terão o

mesmo tamanho, salvo o último que poderá ter tamanho diferente dos demais, variando entre 1 e N bytes.

5.3 Inicialização de Arquivos

Para começar a recuperar partes do arquivo é necessário saber o endereço dos nós que estejam semeando esse arquivo. Um sistema de pesquisa de arquivos é um tema que está fora dos objetivos do presente trabalho. Existem soluções propostas que poderiam ser implementadas de forma que o usuário final conseguisse realizar pesquisas por nome, tipo de arquivo ou descritores de forma a conseguir a informação necessária para conseguir requisitar o arquivo. Uma forma simples de fazer isso seria armazenar informações sobre os arquivos compartilhados de forma a poder catalogá-lo. O usuário criaria regras através da combinação dessas informações para que a aplicação decidisse se deve ou não iniciar a transferência de um arquivo de acordo com essas regras. As informações dos arquivos de cada nó são trocadas de forma epidêmica a cada encontro. Se as informações de algum arquivo retornado em um encontro forem adequadas de acordo com as regras estabelecidas pelo usuário previamente, a aplicação iniciaria a transferência. De uma forma mais refinada, poderia ser utilizado um esquema de subscrição como proposto em [66]. Outra forma interessante seria a impressão de uma imagem contendo um código QR que armazenaria as informações do identificador do arquivo e endereço do nó que publicou e que seriam lidos através das câmeras dos *smartphones*. Inicialmente, as informações necessárias para o início da aplicação são o endereço do nó original e o número de partes do arquivo. Com o passar do tempo, a medida que está obtendo o arquivo, aplicação vai tomando conhecimento de outros nós para solicitar partes do arquivo.

5.4 Mensagens de requisição e resposta

A aplicação utiliza de mensagens de requisição para solicitar partes e mensagens de resposta para devolver a parte solicitada. Além dessas, existem as mensagens RQP e RespNNS que serão abordadas nas Seções 5.5 e 5.6, respectivamente.

As mensagens utilizadas pela aplicação possuem um cabeçalho composto de 3 campos.

Nome	**Descrição**	**Tamanho**
Arquivo	Identificador do arquivo solicitado	16 bytes
Parte	Identificador da parte solicitada do arquivo	4 bytes
Tipo	Tipo de mensagem.	1 byte

Tabela 5.1 - Detalhamento dos campos do cabeçalho das mensagens.

Quanto ao campo "Tipo", os valores possíveis correspondentes aos tipos de mensagem são exibidos abaixo.

Valor	**Tipo de Mensagem**
0	Request
1	Response
2	RQP
3	RespNNS

Tabela 5.2 - Valores representativos dos tipos de mensagem

5.5 Mensagens de requisição RQP

As mensagens de requisição da aplicação têm em seu cabeçalho os valores de identificação do arquivo e da parte que se deseja obter. Em cenários onde o número de

semeadores é muito pequeno em relação ao número total de nós é desejável que partes do arquivo sejam disseminadas rapidamente para que mais fontes dessas partes estejam disponíveis ao longo do enxame, diminuindo o tempo de obtenção da parte solicitada por um nó. O protocolo Bittorrent tem a mensagem HAVE para indicar que partes de um arquivo um nó já possui. Com isso, o nó anuncia essas partes para um outro nó poder solicitar. No cenário oportunista, criar tais mensagens não seria interessante, pois com as grandes latências características dessas redes, provavelmente isso não vai contribuir com a diminuição do tempo de resposta que é o objetivo.

Assim, introduzimos as mensagens Requisição de Qualquer Parte (RQP) para permitir que os nós iniciantes possam adquirir partes mais rapidamente e possam também contribuir com os outros nós oferecendo à rede redundância na oferta dessa parte. Para tal, assumimos o valor especial -5 no campo de identificação da parte do arquivo no cabeçalho da mensagem de requisição. Se o próximo nó que receber essa mensagem tiver alguma parte do arquivo solicitado, ele pode devolver alguma parte selecionada aleatoriamente e impedir que a requisição original continue a se propagar pela rede. O número de vezes que um nó envia mensagens RQP antes de solicitar nominalmente a parte desejada é parametrizado na aplicação.

5.6 Mensagens RespNNS

As mensagens Resposta de Nó Não Semeador (RespNNS) servem para os nós intermediários entre o solicitante e o semeador responderem às solicitações por partes do arquivo caso tenham a parte do arquivo solicitada ou a requisição seja uma mensagem RQP e possuam ao menos uma parte do arquivo solicitado.

Além disso, esse tipo de mensagem também serve para anunciar que o nó que a enviou faz parte do enxame daquele arquivo. Essa é uma forma sem custo de receber informações sobre os nós participantes.

5.7 Fluxo de ações

Na Figura 5.1, podemos ver as atividades desempenhadas pelos nós em um ciclo de requisição e resposta.

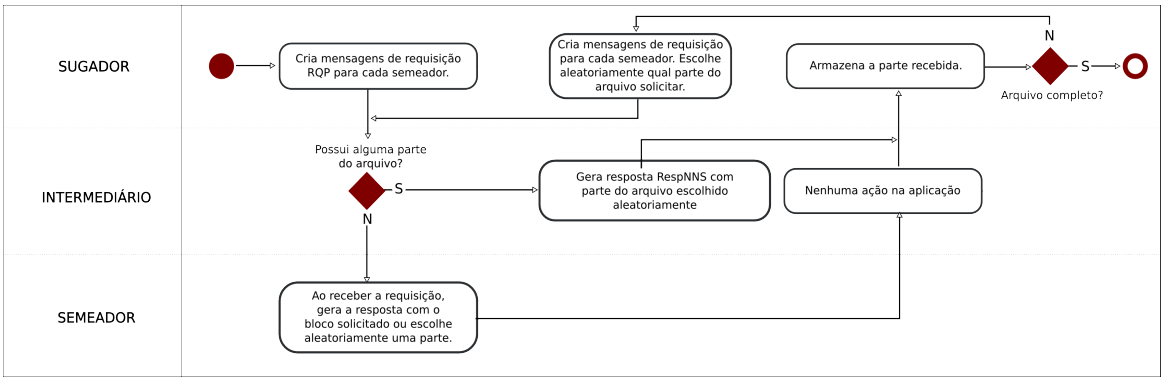


Figura 5.1 - Fluxo de ações para cada tipo de nó usando APTA.

5.8 Implementação da Aplicação no simulador ONE

A aplicação do presente trabalho foi desenvolvida no simulador ONE através da classe ApTAApplication estendendo a classe abstrata Application do próprio *framework* do simulador. A classe ApTAApplication utiliza-se das classes ajudantes P2PConfig e PartRequest para manter controle sobre os arquivos com os quais esteja ofertando ou obtendo partes. Ao iniciar, a aplicação, com o auxílio da classe P2PConfig, lê suas configurações e verifica se há algum arquivo configurado para ser obtido. Caso exista, ela cria uma requisição RQP para cada um dos semeadores conhecidos. Caso ela possua a cópia completa do arquivo configurado, ela passará a atuar como semeadora e ficará aguardando por requisições de outros nós. A implementação foi feita em Java e utilizado o JDK 1.7.

5.9 Protocolo de encaminhamento baseado em vizinhos no local

Tipicamente, em cenários urbanos há um grande número de dispositivos que podem ser utilizados para redes oportunistas. Os protocolos baseados no contexto social podem ter seu desempenho degradado em função do grande número de nós encontrados. Em [67], é mostrado que algoritmos de agregação de contatos podem conduzir os protocolos baseados em contexto social a escolherem nós de forma aleatória, degradando o encaminhamento de mensagens.

Proporemos um protocolo denominado LocalScan que usará os princípios dos protocolos baseados em padrões de movimento. O protocolo LocalScan irá, a cada encontro, tentar enviar a mensagem para as regiões frequentadas pelo destinatário através da análise das regiões frequentadas pelos nós durante o encontro. Quando os nós se encontram, eles trocam as informações de regiões frequentadas e cada nó avalia se deve ou não repassar alguma mensagem com base nessa.

Ao invés de dividirmos o cenário em zonas conforme faz o protocolo DeepWidthSearch [15], iremos mapear as regiões através da percepção dos nós à volta. De tempos em tempos, os nós enviarão *beacons* de sinalização a procura de nós na região. Isso já é uma prática no contexto oportunista pois é preciso detectar nós para se ter encontros. De acordo com a resposta, o nó cria uma entrada com uma lista de representações do endereço MAC dos nós que responderam. Essa lista de nós se chama lista de encontros. Esse processo criará um mapeamento dos nós encontrados. Essa informação será usada para que os nós encaminhem as mensagens apenas aos nós que tiveram contato direto com os nós destinatários das mensagens sendo, portanto, um protocolo de encaminhamento restritivo.

Quando ocorre um encontro, os nós trocam as suas listas de encontros. Cada nó verifica se carrega alguma mensagem para ser entregue ou repassado ao nó encontrado.

Para verificar a possibilidade de repassar alguma mensagem ao nó encontrado, o nó verifica a própria lista de encontros e a lista de encontros trocada durante o encontro. Se algum grupo da lista contiver o endereço MAC do nó encontrado e do nó destinatário da mensagem, o nó deverá repassar cópia da mensagem para o nó encontrado com o intuito de melhorar as chances de entrega da mensagem. A ideia é verificar se os nós estiveram em contato e, com essa informação, decidir se a mensagem deve ser repassada ou não para um nó que tenha probabilidade de reencontro com o destinatário da mensagem. Diante disso, o pseudo-código do LocalScan está estruturado da seguinte forma (Tabela 5.3):

```
Início
    Para cada nó conectado
        Para cada mensagem no buffer
            Se o nó conectado é o destinatário da mensagem
                Entrega a mensagem ao nó.
            Senão
                Se o nó conectado pertence ao mesmo grupo de contatos que o destinatário da mensagem
                    Repassa a mensagem ao nó conectado
Fim
```

Tabela 5.3 - Pseudo-código do LocalScan

6 análise de desempenho

Este capítulo tem como objetivo avaliar o desempenho da aplicação ApTA, bem como seu impacto nos protocolos de encaminhamento das redes oportunistas. Esse capítulo está dividido em 5 partes. Na primeira parte, serão descritos os traces reais utilizados para representar a posição dos nós durante a simulação bem como o tratamento dado aos mesmos. Na segunda parte, será abordado o simulador e as implementações necessárias para realização da simulação. Depois, serão descritos o cenário e listaremos os parâmetros utilizados na simulação e, na sequência, as métricas escolhidas para comparação entre os trabalhos. Na quinta parte, exibiremos e analisaremos os resultados obtidos

6.1 Traces reais de movimento

Para validar as ideias propostas no presente trabalho foi utilizada a simulação como método. Como exposto nos capítulos anteriores, a mobilidade dos nós é o grande fator de desafio para as soluções em redes oportunistas. Além disso, para que as simulações possam refletir os problemas relativos a mobilidade dos nós, foi utilizado dados de traces reais oriundos do CRAWDAD [68], um repositório de dados de redes sem fio mantido através de contribuições de pesquisa disponível na web.

Para o presente trabalho, foi utilizado o *dataset* da rede do campus da Universidade de Dartmouth. A coleta de dados nas redes locais sem fio de Dartmouth que compõem esse *dataset* começou em abril de 2001 e persistiu até outubro de 2006. O

dataset tem as informações de timestamp, nome do ponto de acesso, endereço MAC da estação e tipo de mensagem. O tipo de mensagem indica o evento que aconteceu entre autenticado, associado, desassociado ou desautenticado. Existem 566 pontos de acesso ao longo de 188 prédios do campus. Cada ponto de acesso tem um nome que indica o prédio onde está localizado que são divididos entre as categorias Acadêmico, Biblioteca, Social, Administrativo, Residencial ou Outros. O período do *dataset* usado no presente trabalho foi o compreendido entre 21 de setembro e 20 de outubro de 2003. Esse período foi escolhido em função de ausência de falhas na captura de dados, o que acontece em alguns períodos. Segundo [69], existiram falhas no fornecimento de energia ao campus que levaram a interrupção da coleta em alguns pontos ao longo do tempo. Outros problemas como, por exemplo, mau funcionamento de equipamentos também são relatados.

Foram escolhidos 100 nós entre os 7602 existentes no dataset. Foram selecionados nós que tivessem mais registros dentro do período selecionado. Outro critério de escolha foi o número de pontos de acesso visitados no intervalo de tempo. Como existem desktops que se conectam a rede sem fio da universidade, foi usado o limite mínimo de 10 pontos de acesso visitados para ser selecionado. A razão dessa decisão está em utilizar apenas nós móveis no presente trabalho.

Os arquivos de trace foram lidos e importados para uma base Mysql para facilitar a seleção e exportação das informações necessárias. Os eventos foram classificados em dois tipos de eventos. Quando o tipo de mensagem no trace era referente aos eventos “associação” ou “autenticação”, o registro do trace foi associado ao tipo de evento “chegada da estação ao ponto de acesso” e associado a “partida da estação do ponto de acesso” quando tinha relação com os eventos “deassociação” e “desautenticação”. O período de tempo entre os eventos “chegada da estação ao ponto de acesso” e “partida

da estação do ponto de acesso" é o período de tempo em que a posição da estação é conhecida e utilizada no trace para realizar contatos.

Após escolha das estações que iriam participar da simulação, todos os eventos relativos foram gravados em um arquivo usando os campos *timestamp*, identificação do ponto de identificação da estação, coordenadas do ponto de acesso e tipo de evento.

Abaixo, na Figura 6.1, temos uma imagem com o campus de Dartmouth. Os pontos de acesso que coletaram os traces que foram usados no presente trabalho, foram espalhados pelo campus em mais de 161 prédios.

Figura 6.1 - Mapa do campus de Dartmouth

6.2 Simulador

Para realizar a simulação, foi utilizado o simulador The ONE que possui em seu núcleo um simulador de eventos discretos baseado em agentes. As principais funções

do The ONE são a modelagem da movimentação dos nós, os contatos entre os nós, roteamento e o processamento de mensagens [70]. Além do simulador, o The ONE é um verdadeiro *framework* que permite a extensão de suas funcionalidades e configuração de cenários.

Para o presente trabalho, foram implementados os roteadores DepthWidthSearch presente no trabalho [15] e o LocalScan para a simulação através da extensão da classe ActiveRouter. Além dessas duas classes, também foi criado um gerenciador de movimento através da extensão da classe MovementModel. A função dessa classe é controlar a movimentação dos nós ao longo da simulação, alterando a localização e/ou presença dos nós no cenário de acordo com o tempo de simulação e o arquivo de trace armazenado em disco.

O ONE provê um mecanismo de coleta de dados da simulação baseado em relatórios e listeners que ficam aguardando a notificação de eventos que ocorram ao longo da simulação de acordo com o tipo de objeto de interesse:

- Listeners para mensagens onde é possível ser notificado da criação, transferência e exclusão de uma mensagem;
- Listeners para aplicações que recebe eventos customizados lançados por aplicações;
- Listeners para conexões que são notificadas quando dois nós se encontram e quando se desfaz este encontro;
- Listeners para movimento que é notificado da posição inicial do nó quando posicionado no cenário e toda vez quando recebe um novo destino de seu modelo de movimento.

Foram implementados dois relatórios para poder extrair as métricas que permitiram validar a presente proposta. O ReportTime é um relatório implementado através de um *listener* de aplicação. As aplicações ApTA e P2PSimples registraram os eventos abaixo, com exceção dos eventos P2PMSG_HAVE_SENT e P2PMSG_DUPLICADO que são exclusivos da aplicação ApTA.

Identificador do Evento	**Descrição do Evento**
P2PMSG_REQUEST_SENT	Requisição é enviada a partir de um sugador.
P2PMSG_REQUEST_RECEIVED	Recebimento da requisição por um semeador ou nó intermediário que vá responder a requisição.
P2PMSG_RESPONSE_SENT	Resposta enviada para um sugador com a parte do arquivo.
P2PMSG_RESPONSE_RECEIVED	Recebimento da resposta com a parte do arquivo.
P2PMSG_HAVE_SENT	Mensagem RespNNS enviada por um nó intermediário que respondeu a solicitação. Somente a aplicação ApTA gerou esse evento.
P2PMSG_COMPLETE	Evento gerado no momento em que o arquivo é completado em um nó.
P2PMSG_DUPLICADO	Evento de recebimento de uma resposta com uma parte do arquivo já recebida. Somente a aplicação ApTA gerou esse evento.

Tabela 6.1 - Identificadores e descrições dos eventos gerados pelas aplicações durante a simulação.

O segundo relatório é o DelayMessageReport que é um *listener* de mensagens e coleta informações de criação, repasses e entregas de mensagens na simulação.

Também foi implementada a aplicação P2PSimples que é uma aplicação geradora de tráfego simples que servirá de comparação. Para não ser tendencioso, não foram implementados mecanismos como os algoritmos *rarest first*, nem o jogo tit-for-tat na aplicação P2PSimples, pois entendemos que esses mecanismos funcionam bem

em um contexto onde a conectividade pode ser presumida, o que não é o caso das redes oportunistas.

As duas aplicações têm em comum o funcionamento inicial. Ao iniciar, as aplicações geram uma mensagem de requisição de parte do arquivo para cada um dos semeadores de acordo com o número de semeadores parametrizado. A cada requisição recebida, o semeador envia uma resposta. Ao receber a resposta, o sugador envia nova requisição solicitando uma nova parte e esse processo continua até que o arquivo se completa. A diferença entre as duas implementações reside em dois pontos:

1. Na requisição inicial, a aplicação ApTA envia uma mensagem de requisição RQP cujo valor da identificação do bloco é -5 para cada um dos semeadores conhecidos. A aplicação P2PSimples sorteia aleatoriamente o número do bloco a ser solicitado dentre os que o nó ainda precisa obter.
2. Quando um nó que não é a origem nem o destino da mensagem, chamado de nó intermediário, recebe uma requisição, verifica se possui o bloco do arquivo solicitado. Se possuir, este nó devolve o bloco através de uma mensagem de resposta RespNNS. Na aplicação P2PSimples, não há qualquer ação por parte do nó intermediário nesse caso.

A Figura 6.2 descreve o fluxo de ações para o tratamento de mensagens de acordo com o nó que recebe a requisição na aplicação P2PSimples.

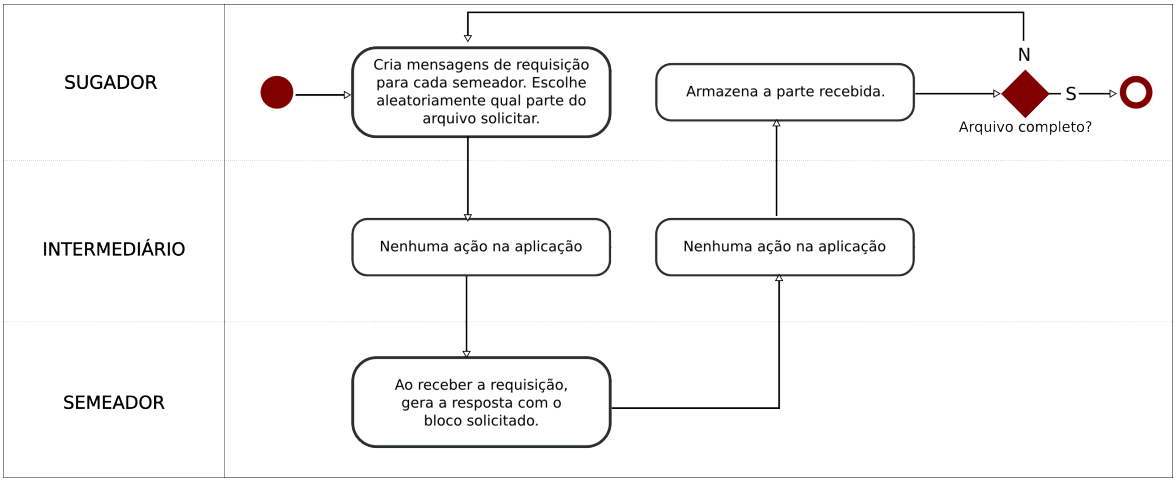


Figura 6.2 - Fluxo de ações para cada tipo de nó usando P2PSimples

6.3 Cenário de Simulação

A simulação foi realizada com os traces descritos na Seção 6.1 e foram escolhidos os protocolos de encaminhamento Spray and Wait, BUBBLE Rap, MaxProp, ProPHET, DepthWidthSearch e LocalScan. Esses protocolos foram escolhidos por serem de categorias diferentes conforme a Seção 2.3 e permitir avaliar a diferença entre essas abordagens em relação ao comportamento de rede e no uso das aplicações implementadas no presente trabalho. Todos os parâmetros tiveram valores default de implementação. Essa decisão foi necessária porque os trabalhos referentes aos protocolos de encaminhamento de mensagens em redes oportunistas, em geral, exploram valores diferentes para os parâmetros para analisar a influência dos mesmos no comportamento da rede. Como simular cada um dos muitos parâmetros e suas possibilidades exigiria um esforço que suplantaria o disponível para este trabalho, decidimos optar por valores definidos na literatura e equilibrados que não fossem tendenciosos em uma das métricas usadas para avaliar o comportamento de rede. As implementações do Spray and Wait, ProPHET e MaxProp são as implementações fornecidas juntamente com o simulador The ONE. A implementação do BUBBLE Rap foi desenvolvida por P.J. Dillon da Universidade de Pittsburgh e está disponível em

[71]. O autor implementou os protocolo DepthWidthSearch de acordo com o trabalho [15] e levou em consideração os valores para os parâmetros conforme listado abaixo.

6.3.1 Parâmetros utilizados pelos protocolos

- Spray and Wait
  Número de cópias de mensagens: 6
  Modo de operação binário: Sim

- ProPHET
  Número de segundos por unidade de tempo: 30

- BUBBLE Rap
  Algoritmo de detecção de comunidade: K-Clique
  Valor de K: 5
  Valor limite do conjunto familiar: 700

- DepthWidthSearch
  L: 0
  Número de ativistas: 30
  Número de cópias de mensagens: 20
  Centro das zonas: Para a divisão do cenário em zonas, foi considerada a localização de cada um dos pontos de acesso como identificador e centro das zonas, conforme o trabalho que propôs o protocolo [15].

- LocalScan
  Número de ativistas: 30
  Número de cópias de mensagens: 10

Como dito na Seção 6.2, foi desenvolvido um gerenciador de movimento para ler os registros de posição que foram compilados do trace e refletir a posição e estado do nó a cada passo da simulação. Assim, a medida que a simulação vai sendo executada, nós aparecem no cenário e nós são retirados do cenário de acordo com os tipos de evento lidos.

Foram configurados diferentes números de semeadores para observar a disseminação de conteúdo. Foram utilizados 1, 5, 10 e 20 semeadores que foram escolhidos aleatoriamente. Os tamanhos dos arquivos utilizados foram de 512KB que representaria uma foto tirada com uma câmera de um celular, por exemplo, e 4MB que

é o tamanho médio para uma música codificada com o padrão MP3 ou um curto clipe de vídeo de aproximadamente 30s de duração codificado com o codec H.264 e tamanho 480X480 pixels. Foi adotado o padrão 802.11b em modo ad hoc a taxa de 11Mbps e alcance dos rádios de 50m.

As mensagens usadas pelas aplicações têm tamanho de 21 bytes para as requisições e 21 bytes de cabeçalho e 64KB de dados para as respostas, totalizando 65.557 bytes, o tamanho do buffer dos nós foi configurado para 100MB e o TTL das mensagens foi configurado para todo o tempo de simulação.

O tempo de simulação foi de 864.000 segundos que equivale a 10 dias.

6.4 Métricas de avaliação

A seguir, serão apresentadas as métricas que utilizaremos para avaliar o comportamento da rede e depois a métrica utilizada para avaliar as aplicações implementadas no presente trabalho.

Taxa de entrega – É a razão de mensagens entregues em relação às mensagens criadas. É uma métrica importantes pois ela mede a eficiência do protocolo em entregar as mensagens ou indicar qual a probabilidade de uma mensagem ser entregue.

$$T = \frac{E}{C}, \qquad C > 0 \tag{6.1}$$

Onde T é a taxa de entrega, E é o número de mensagens entregues e C o número de mensagens criadas.

Sobrecarga – É a razão entre a diferença entre o número de mensagens repassadas e mensagens entregues pelo número de mensagens entregues. Indica o número médio de transmissões necessárias para que uma mensagem seja entregue.

$$S = \frac{R - E}{E}, E > 0 \quad 6.2$$

Onde S é a sobrecarga, R é o número de mensagens repassadas e E o número de mensagens entregues.

Latência – Diferença de tempo entre o momento de criação da mensagem e o momento da entrega da mensagem ao destinatário. É o tempo total utilizado para se entregar a mensagem.

Número de saltos – Em redes oportunistas, a mensagem é repassada a outros nós até que chegue ao destinatário. Cada transferência da mensagem ou salto de um nó para outro, incrementa essa métrica. Assim, número de saltos é o número total de transferências realizadas nó a nó para entregar a mensagem do criador da mensagem ao destinatário da mensagem.

Número de nós que completaram a transferência do arquivo – É o número de nós que conseguiram reunir todas as partes do arquivo solicitado durante o experimento.

As métricas taxa de entrega, latência e atraso nos permitem observar como foi o fluxo de dados através da rede. Quanto às métricas número de mensagens, número de saltos e número de nós que completaram a transferência do arquivo nos permitirão avaliar o comportamento das aplicações e verificar se os mecanismos propostos para a aplicação ApTA lograram êxito em aproximar o conteúdo do usuário e acelerar a disseminação das partes do arquivo no enxame.

6.5 Resultados obtidos

Começando a análise, é importante observar que o número de mensagens geradas foram diferentes entre as aplicações. Tanto a aplicação ApTA quanto a aplicação P2PSimples seguiam um ciclo de requisição e resposta. Quando o nó sugador iniciava

seu funcionamento pela primeira vez na simulação, enviava requisições para todos os semeadores conhecidos e aguardava por respostas para enviar novas requisições. Assim, na Figura 6.3 e Figura 6.4, podemos ver que a aplicação ApTA gerou mais mensagens do que a aplicação P2PSimples, o que é um sinal de que realizou ciclos de requisição e resposta de forma mais rápida.

Quanto a Figura 6.3, especificamente, podemos ver que o comportamento de todos os roteadores é muito parecido. Há um crescimento do número de mensagens em função do aumento de semeadores, pois mais semeadores significam mais requisições a cada ciclo.

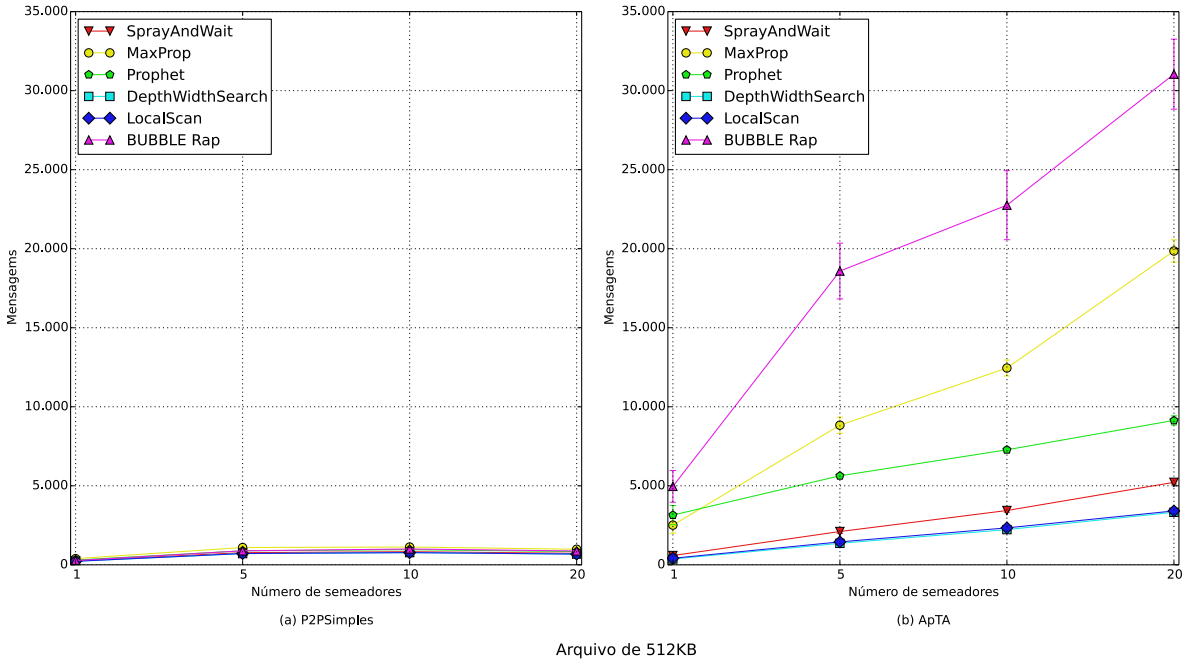


Figura 6.3 - Número de mensagens criadas com o tamanho de arquivo de 512KB

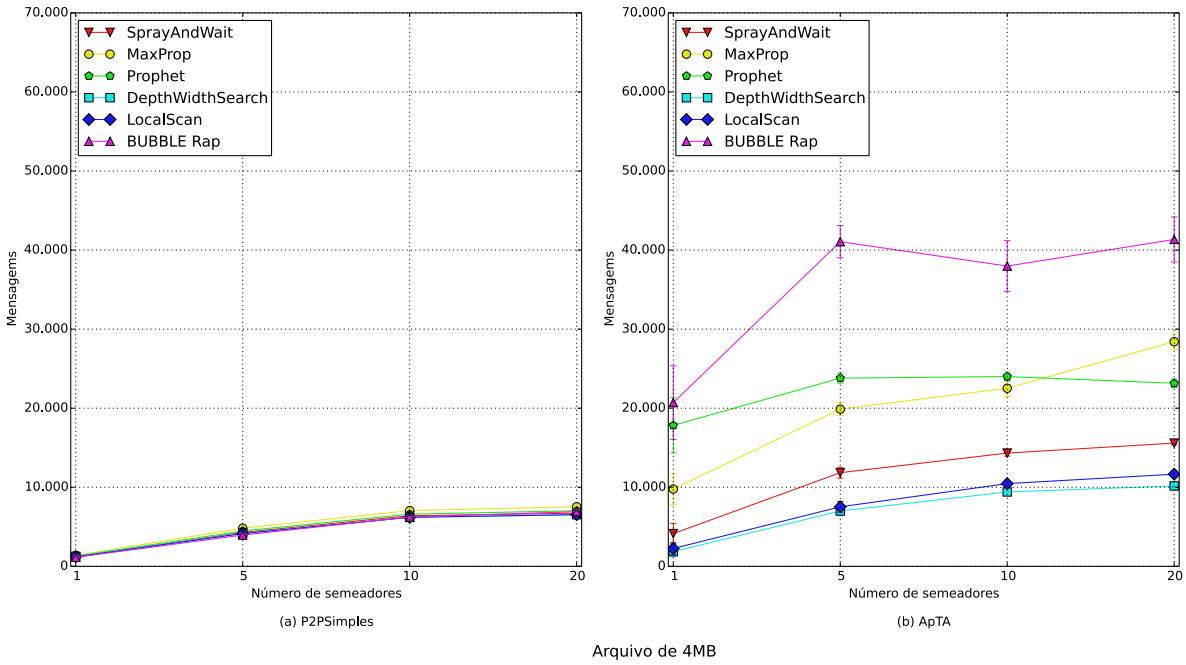


Figura 6.4 - Número de mensagens criadas com o tamanho de arquivo de 4MB

Seguindo com as métricas de avaliação da camada de redes, percebemos pela Figura 6.5 e Figura 6.6 que o atraso das mensagens, em geral, segue a tendência de acompanhar o crescimento do número de semeadores no caso da aplicação P2PSimples. Esse fato se deve, mais uma vez, em grande parte ao aumento do número de mensagens na rede, uma vez que as aplicações solicitavam partes a todos os semeadores. Já com a utilização da aplicação ApTA, o atraso se mantém praticamente constante mesmo com o aumento do número de semeadores, pois, a medida que o tempo passa, a distribuição do conteúdo permite que o mesmo seja recuperado a partir de nós mais próximos o que mitiga o atraso. Os protocolos com maior controle de disseminação tiveram menores atrasos nas duas situações com arquivos de 512KB e 4MB. Spray and Wait, DepthWidthSearch e LocalScan têm comportamentos muito parecidos em função do crescimento dos semeadores. Os protocolos BUBBLE Rap, ProPHET e MaxProp tiveram um comportamento parecido e apresentaram maior atraso que os outros três.

O protocolo LocalScan foi o que apresentou o menor atraso, tanto com o tamanho de arquivo de 512KB quanto com o tamanho de 4MB usando a aplicação P2PSimples.

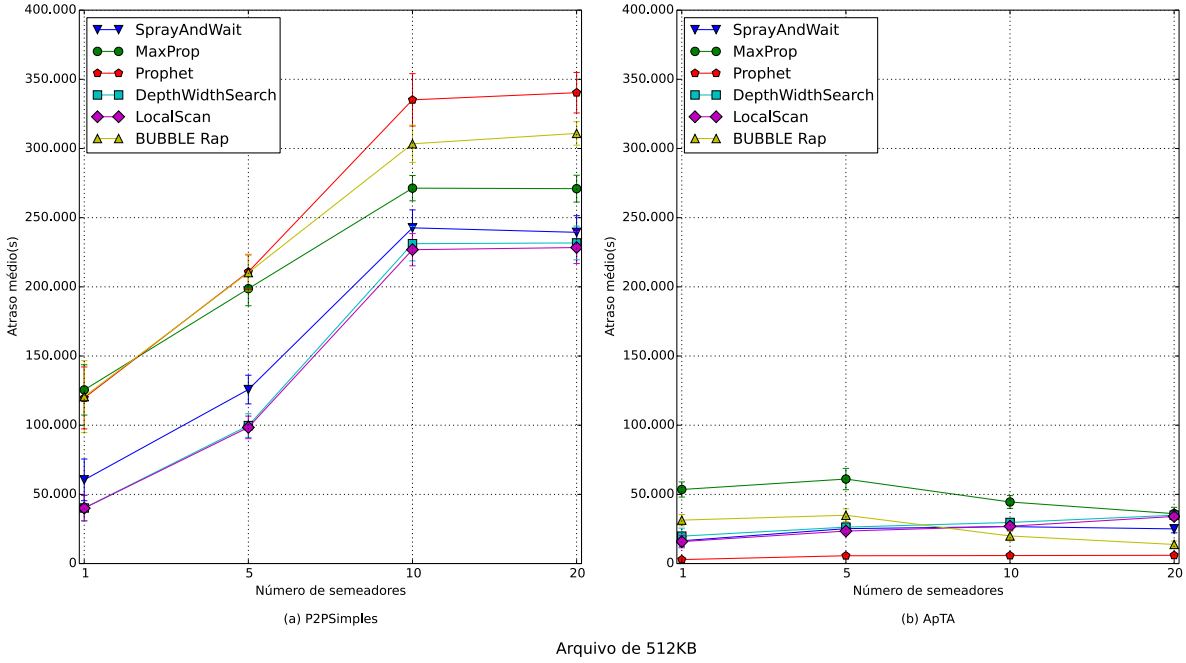


Figura 6.5 - Atraso com arquivo de tamanho 512KB

Já na aplicação ApTA, os protocolos Spray and Wait, DepthWidthSearch e LocalScan também apresentaram comportamentos muito parecidos e têm valores de atraso medianos em relação aos demais protocolos. Os protocolos BUBBLE Rap e MaxProp têm o mesmo comportamento com o valor máximo de 5 semeadores para os dois tamanhos de arquivo. O ProPHET foi o que teve o melhor desempenho tanto com o arquivo de 521KB quanto com o arquivo de 4MB.

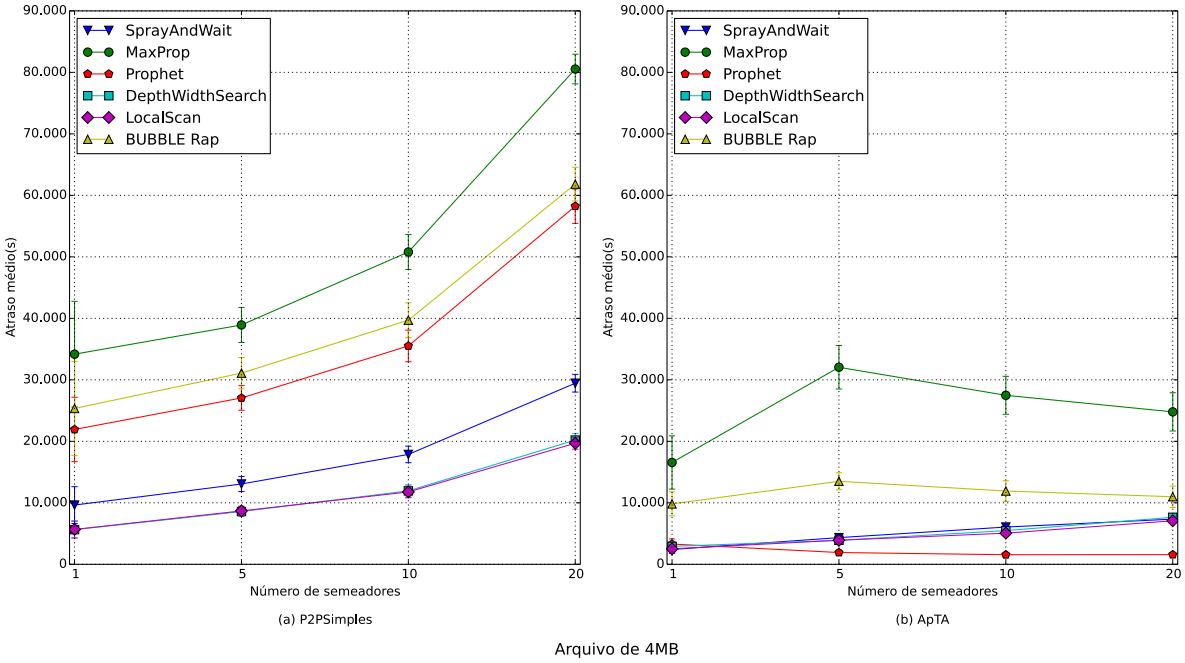


Figura 6.6 - Atraso com arquivo de tamanho 4MB

Comparando as duas aplicações, ApTA teve um atraso muito menor em todos os protocolos com todos os tamanhos de arquivo. Chegando a ter uma melhoria de 34 vezes como é o caso do ProPHET com 20 semeadores e com arquivo de 512KB. É interessante notar que tanto o MaxProp quanto o ProPHET são protocolos de encaminhamento baseados na probabilidade de entrega, mas apresentaram grande diferença entre os atrasos quando utilizada a aplicação ApTA. Aparentemente, o encurtamento de caminhos através das mensagens RQP e RespNNS não beneficiaram o MaxProp que prioriza mensagens com baixo número de saltos e descarta as mensagens com baixa probabilidade de entrega. Com a diminuição do número de saltos através das mensagens RQP e RespNNS, o ProPHET terminou sendo o protocolo com o menor atraso com o arquivo de 512KB e com os números de semeadores 5, 10 e 20 para os arquivos de 4MB.

Em relação a taxa de entrega, as duas aplicações apresentaram resultados mais equilibrados. Na Figura 6.7 e na Figura 6.8, podemos observar que os protocolos

tiveram um comportamento semelhante quando é utilizada a aplicação P2PSimples, diminuindo suas taxas de entrega em razão do aumento de semeadores que, como já observamos antes, gera muito mais mensagens uma vez que cada sugador envia requisições para cada um dos semeadores.

A aplicação ApTA não é favorecida pelo uso do ProPHET e tem desempenho melhor do que a aplicação P2PSimples em todos os demais protocolos, salvo o BUBBLE Rap que oferece melhor resultado para 10 e 20 semeadores para o tamanho de 512KB. Para o arquivo de 4MB, a aplicação ApTA consegue resultados melhores com o Spray and Wait, LocalScan e DepthWidthSearch. O protocolo MaxProp oferece resultado semelhante com as duas aplicações. Aqui cabe ressaltar que a taxa de entrega é uma medida relativa das mensagens entregues em função das mensagens criadas. As aplicações seguem ciclos de requisição e resposta, ou seja, salvo as requisições iniciais, novas requisições só são criadas se os nós receberem respostas.

Cabe ressaltar que o protocolo LocalScan fica em uma posição mediana em relação aos outros onde o tamanho do arquivo é de 512KB e é um dos melhore juntamente com o Spray and Wait e o DepthWidthSearch quando é utilizado o arquivo de 4MB.

Observando as taxas de entrega, percebe-se que, ao contrário do que ocorria com o atraso na entrega das mensagens que acompanhava o aumento do número de semeadores, a taxa de entrega diminui a medida em que o número de semeadores aumenta, o que é explicado pelo aumento do número de mensagens geradas uma vez que cada sugador gera uma requisição para cada um dos semeadores. Com mais mensagens na rede, a tendência é de aumento de utilização dos buffers e congestionamento na troca de mensagens, diminuindo as taxas de entrega. Quanto a

aplicação P2PSimples, o protocolo de encaminhamento MaxProp se sobressai na taxa de entrega de mensagens, sobretudo quando o arquivo a ser distribuído é o com tamanho de 512KB.

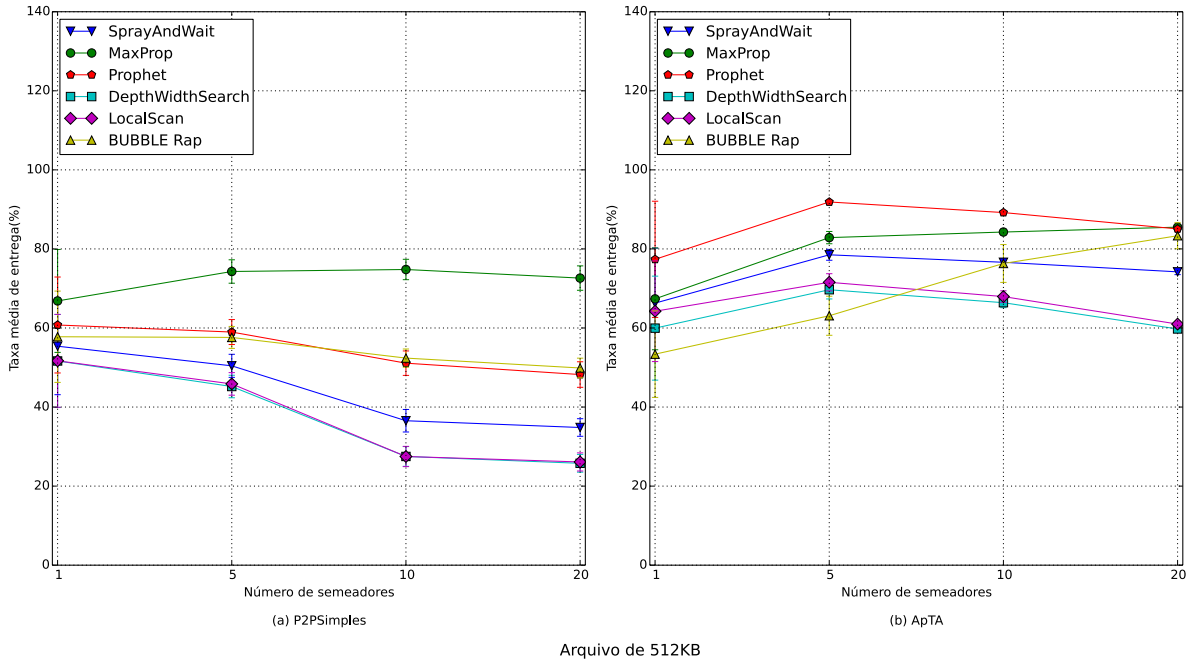


Figura 6.7 - Taxa de entrega com tamanho de arquivo de 512KB

Percebe-se que a taxa de entrega dos protocolos de encaminhamento quando foi utilizada a aplicação ApTA teve comportamento parecido com a taxa de entrega dos protocolos com a execução da aplicação P2PSimples. A única diferença foi o BUBBLE Rap que ao invés de diminuir, melhorou sua taxa de entrega em função do aumento dos semeadores. A notar a diferença entre os valores para 1 semeador na distribuição do arquivo de 4MB, vemos que o BUBBLE Rap não se sai tão bem uma vez que também teve uma taxa de entrega mais baixa que os demais protocolos na distribuição do arquivo de 512KB.

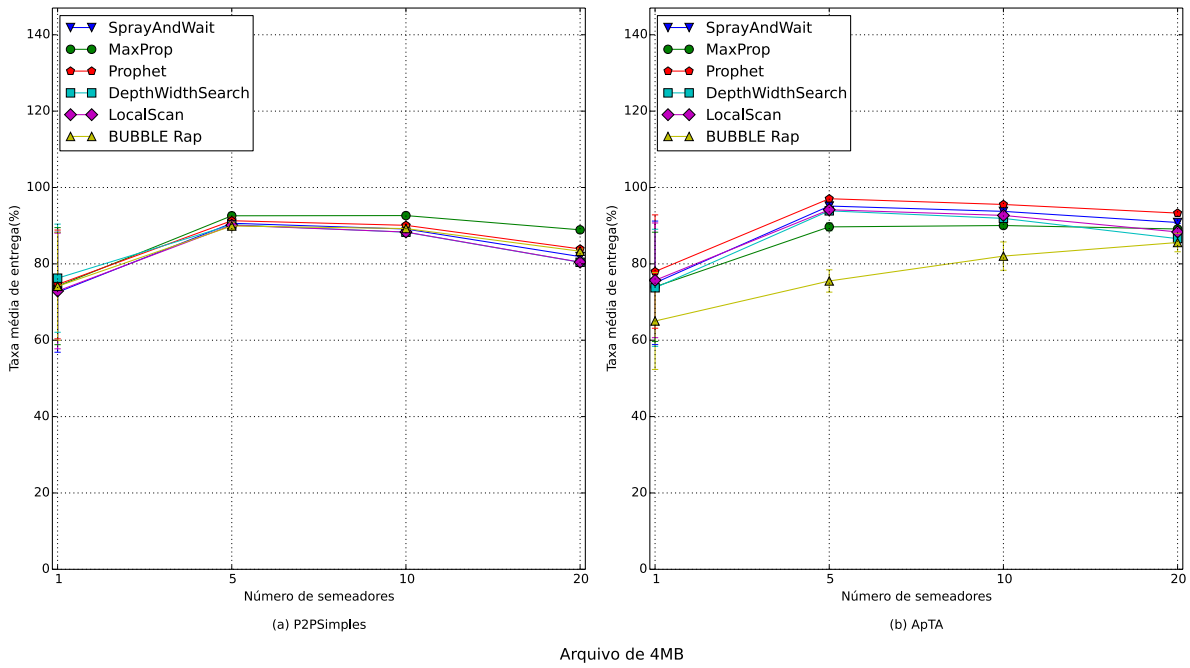


Figura 6.8 - Taxa de entrega com tamanho de arquivo de 4MB

As duas aplicações tiveram resultados parecidos em relação às taxas de entrega. Há uma sensível melhora nas taxas de entrega na aplicação ApTA em relação à aplicação P2PSimples quando o arquivo distribuído é o com tamanho de 512KB e as aplicações praticamente empatam quando distribuem o arquivo de 4MB, embora haja uma pequena diferença entre as duas aplicações a favor da aplicação ApTA.

Uma vez que apresentamos os resultados referentes às métricas de atraso e taxa de entrega, vamos observar os resultados referentes à sobrecarga das mensagens para poder avaliar se os protocolos precisaram replicar muitas mensagens na rede para entregar as mensagens.

De acordo com a Figura 6.9 e Figura 6.10, quando é distribuído o arquivo de 512KB utilizando a aplicação P2PSimples, os protocolos de encaminhamento de disseminação controlada conseguem praticamente manter a mesma sobrecarga apesar do aumento do número de semeadores. Esse comportamento é melhorado quando o arquivo a ser distribuído é o arquivo com 4MB de tamanho, ou seja, os protocolos Spray

and Wait, DepthWidthSearch e o LocalScan conseguem manter constante e baixo o número de réplicas de mensagens necessárias durante a distribuição do arquivo. Já os outros protocolos tiveram maior custo para entregar as mensagens. Comparando a distribuição do arquivo de 512KB com a do arquivo de 4MB, os protocolos MaxProp e BUBBLE Rap diminuíram o número de réplicas na distribuição do arquivo de 4MB, enquanto que o ProPHET aumentou o número de réplicas de mensagens.

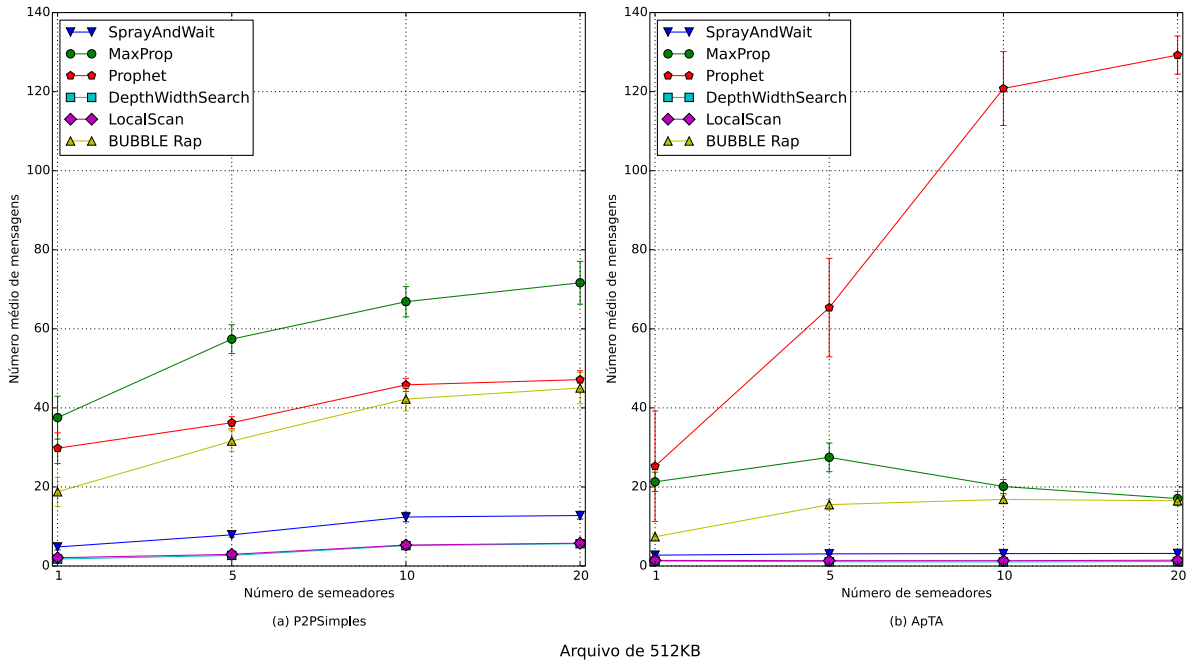


Figura 6.9 - Sobrecarga com tamanho de arquivo de 512KB

Com a aplicação ApTA, o comportamento foi semelhante. Vemos que os protocolos Spray and Wait, DepthWidthSearch e o LocalScan conseguiram manter baixo e constante o valor de réplicas de mensagens, acompanhados pelos protocolos BUBBLE Rap e MaxProp que conseguiram baixar o número de réplicas ao longo da distribuição. A exceção foi o ProPHET que precisou criar muitas réplicas para conseguir entregar as mensagens. É possível que em cenários onde mais arquivos estejam sendo distribuídos na rede esse comportamento do ProPHET possa degradar o número de cópias do arquivo distribuídas.

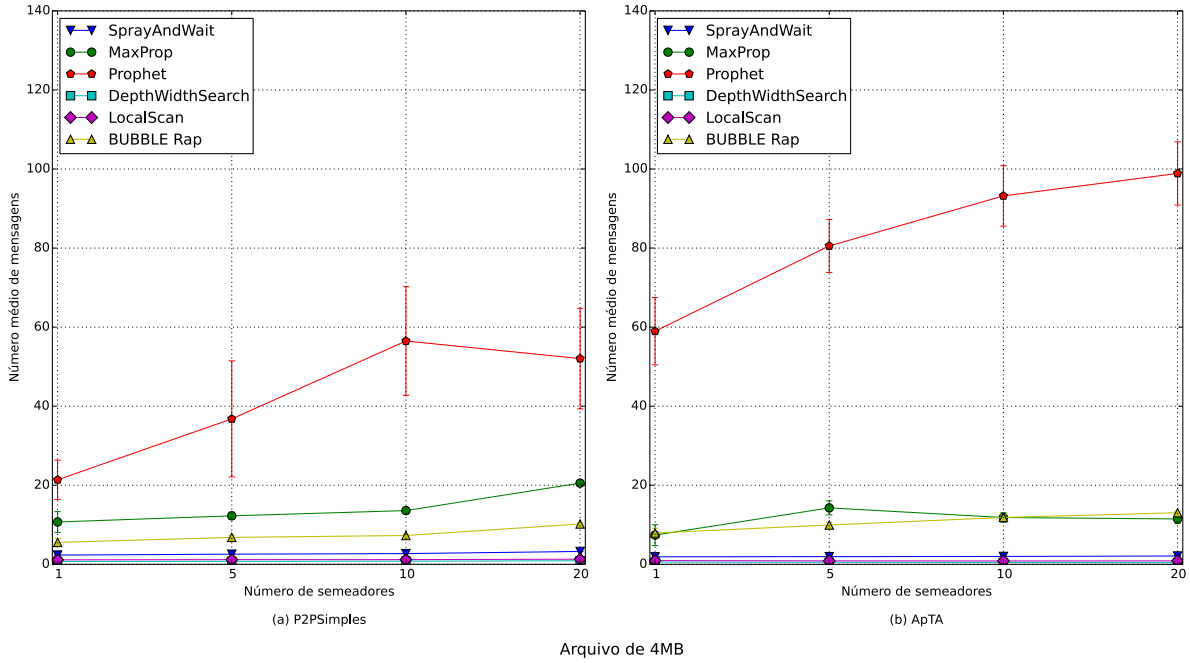


Figura 6.10 - Sobrecarga com tamanho de arquivo de 4MB

De acordo com a Figura 6.11 e Figura 6.12, podemos perceber que o número de saltos aumentou a medida que o número de semeadores foi incrementado. Essa tendência é melhor percebida na distribuição do arquivo com tamanho de 512KB do que na distribuição do arquivo de 4MB. Em ambos os casos, o MaxProp teve que realizar mais transferências até conseguir entregar ao destinatário a mensagem. O LocalScan foi o que teve menor número de saltos uma vez que suas características o tornam mais controlado, fazendo com que as mensagens sejam trocadas com menor salto.

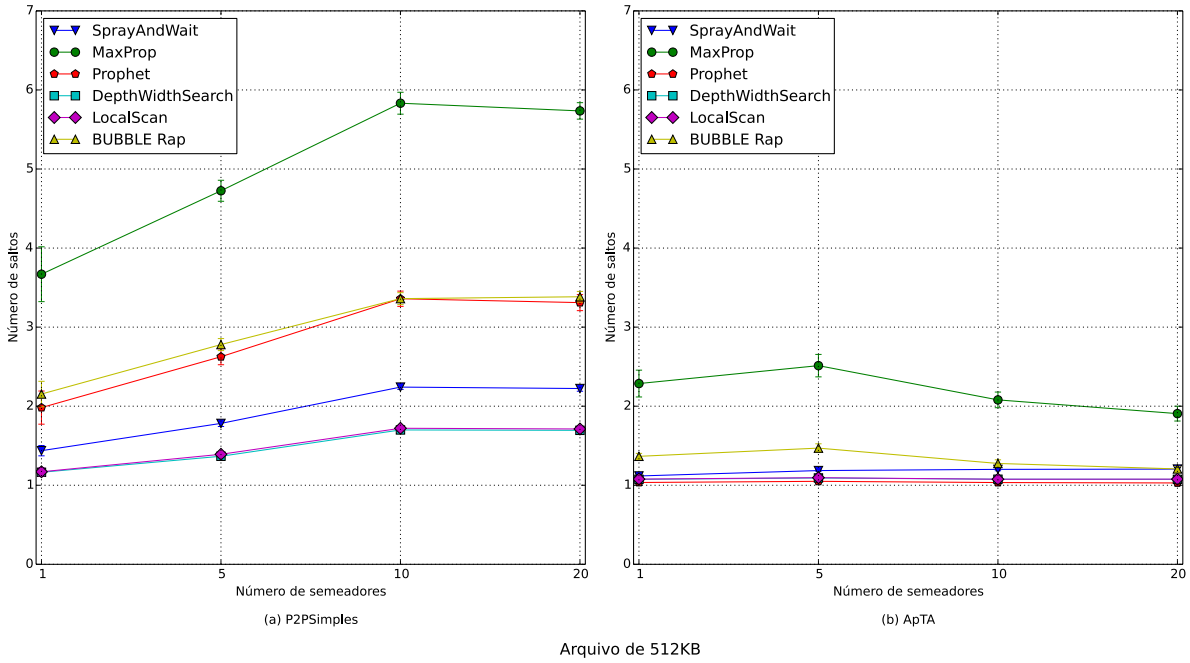


Figura 6.11 – Número médio de saltos na entrega de mensagens com tamanhos de arquivo de 512KB

Comparando as aplicações, percebemos que todos os protocolos tiveram diminuição do número de saltos para entregar mensagens com a utilização da aplicação ApTA. O ProPHET que necessitava de ao menos 2 saltos para entrega com a aplicação P2PSimples, passou a praticamente 1, o que significa que passou a entregar diretamente as mensagens. Essa diminuição do número de saltos, indica que as estratégias inclusas na aplicação para aproveitar os nós mais próximos para solicitar o conteúdo surtiu efeito.

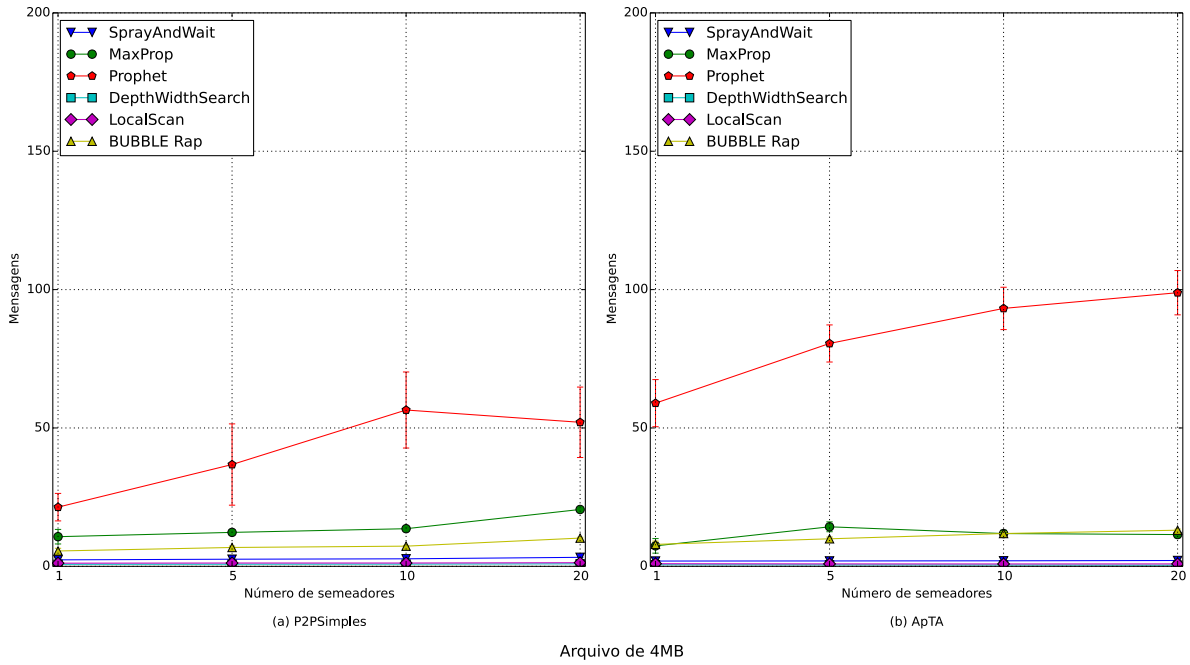


Figura 6.12 - Número médio de saltos na entrega de mensagens com tamanho de arquivo de 4MB

Após apresentadas as métricas de número de mensagens criadas, atraso, sobrecarga, taxa de entrega e número de salto, vimos que quando a aplicação ApTA é utilizada para disseminar arquivos, a rede conseguiu entregar mais mensagens proporcional e absolutamente, em menor tempo e com menor esforço, salvo em alguns poucos pontos com o uso de alguns protocolos, notadamente o ProPHET.

O protocolo proposto LocalScan foi o que menos onerou a rede, pois funcionou com sobrecarga baixa e constante mesmo em razão do aumento de mensagens proporcionado pelo acréscimo de semeadores e teve o menor tempo de atraso com a distribuição dos arquivos de tamanho 512KB e de tamanho 4MB, tanto com a aplicação P2PQRP quanto com a aplicação P2PSimples. Contudo, a taxa de entrega do protocolo ficou entre as mais baixas quando o arquivo a ser distribuído tem tamanho de 512KB, mas a taxa de entrega ao distribuir o arquivo de 4MB foi mediana em relação aos outros protocolos.

Muitos trabalhos tentam contribuir com aplicações para o cenário oportunista, persistindo na ideia de que uma rede oportunista seja uma rede de uso geral. Então, realizam experimentos baseados no encaminhamento de mensagens em rede, argumentando terem conseguido entregar mais mensagens, com menor custo, no menor tempo. Mas será que isso é o suficiente para que uma aplicação de troca de arquivos no contexto oportunista tenha sucesso?

Iniciaremos, agora, a exposição das métricas de avaliação pertinente ao desempenho das aplicações e que usaremos para avaliar as aplicações aqui propostas.

A primeira métrica que iremos discutir é o número de nós sugadores que conseguiram reunir todas as partes do arquivo e possuem ao final do tempo do experimento uma cópia completa do arquivo em seu buffer. Analisando a Figura 6.13(a) , vemos que todos os protocolos tiveram comportamento parecido. Ao longo do tempo de simulação que compreendeu 10 dias, ao menos 8% dos nós conseguiram completar o arquivo com apenas 1 nó fazendo o papel de semeador. Como vimos nas métricas anteriores, o ProPHET replica mais mensagens durante o seu funcionamento e consegue uma pequena vantagem em relação aos outros protocolos, salvo o MaxProp que o acompanha conseguindo os mesmos resultados. A medida que o número de semeadores aumenta e, por consequência, o número de mensagens também aumenta, o número de nós que conseguiram completar o arquivo vai diminuindo devido a congestionamentos e utilização maior do buffer. Observando a Figura 6.13(c), vemos que apenas o MaxProp consegue entregar partes do arquivo o suficiente para completar alguns poucos usuários, corroborando com a Figura 6.7(a) que mostra o MaxProp ser o protocolo de encaminhamento que mais entregou mensagens.

Na Figura 6.14(a) e (c), podemos perceber que, com o acréscimo do número de mensagens em função do aumento do tamanho do arquivo, menos nós conseguem

completar o arquivo, pois há mais partes a serem reunidas para completar o arquivo. De acordo com a Figura 6.6(a) o atraso chega , com a utilização do MaxProp, a 80.000 segundos em média, mais de 22h para que uma mensagem seja entregue quando temos 20 semeadores na distribuição do arquivo com tamanho de 4MB. Como cada parte tem 64KB, são necessárias 64 partes para que um nó consiga receber todo o arquivo. Como são ciclos de requisição e resposta, seriam 64 requisições e 64 respostas, totalizando 128 mensagens. Seriam necessárias, em média, 2.816 horas ou aproximadamente, 117 dias para completar a transferência em média. Mesmo o LocalScan que teve o menor atraso com 20 semeadores ao distribuir o arquivo de 4MB levaria, em média, cerca de 29 dias.

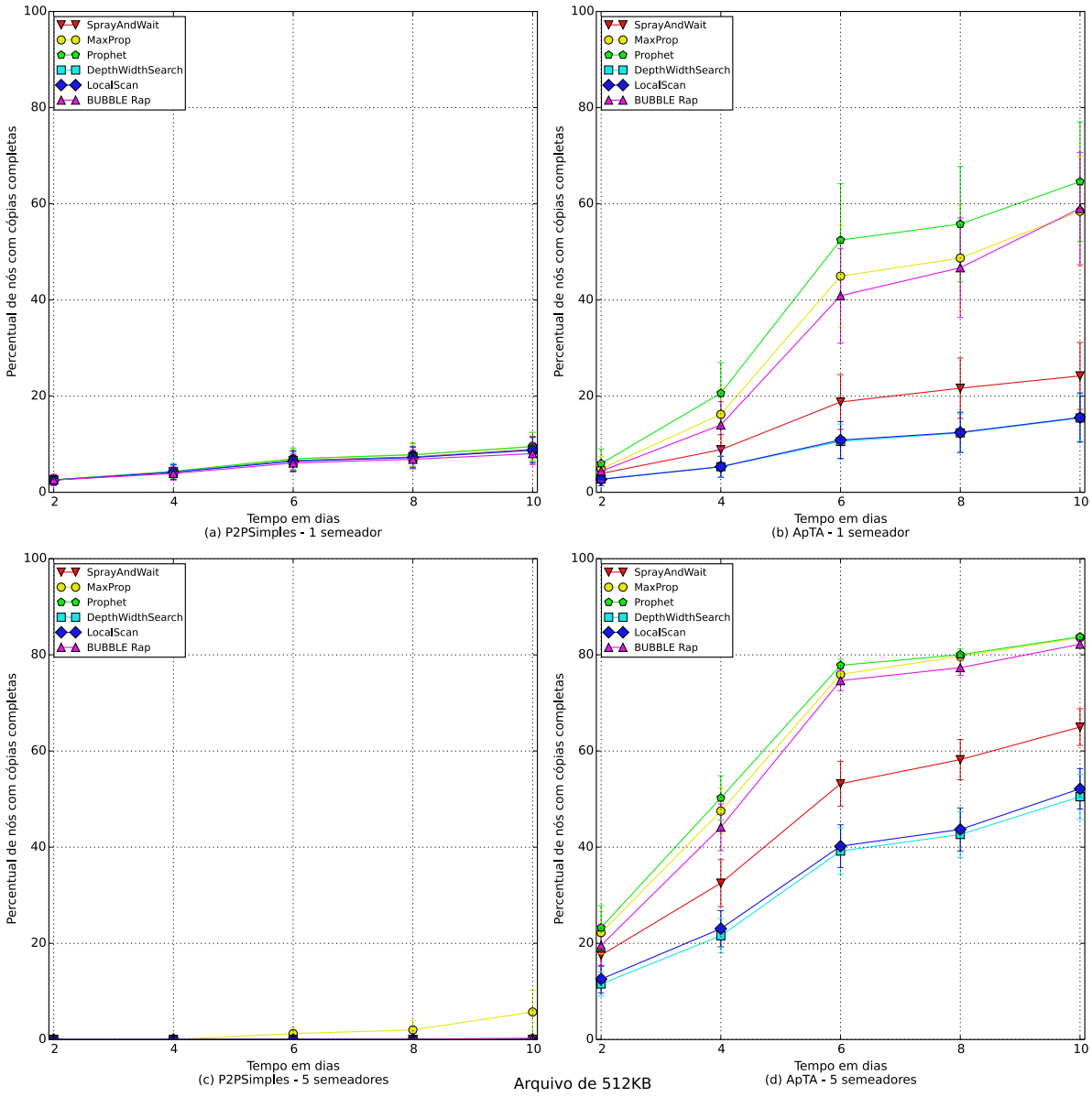


Figura 6.13 - Percentual de nós com cópias completas com arquivo de 512KB para 1 e 5 semeadores

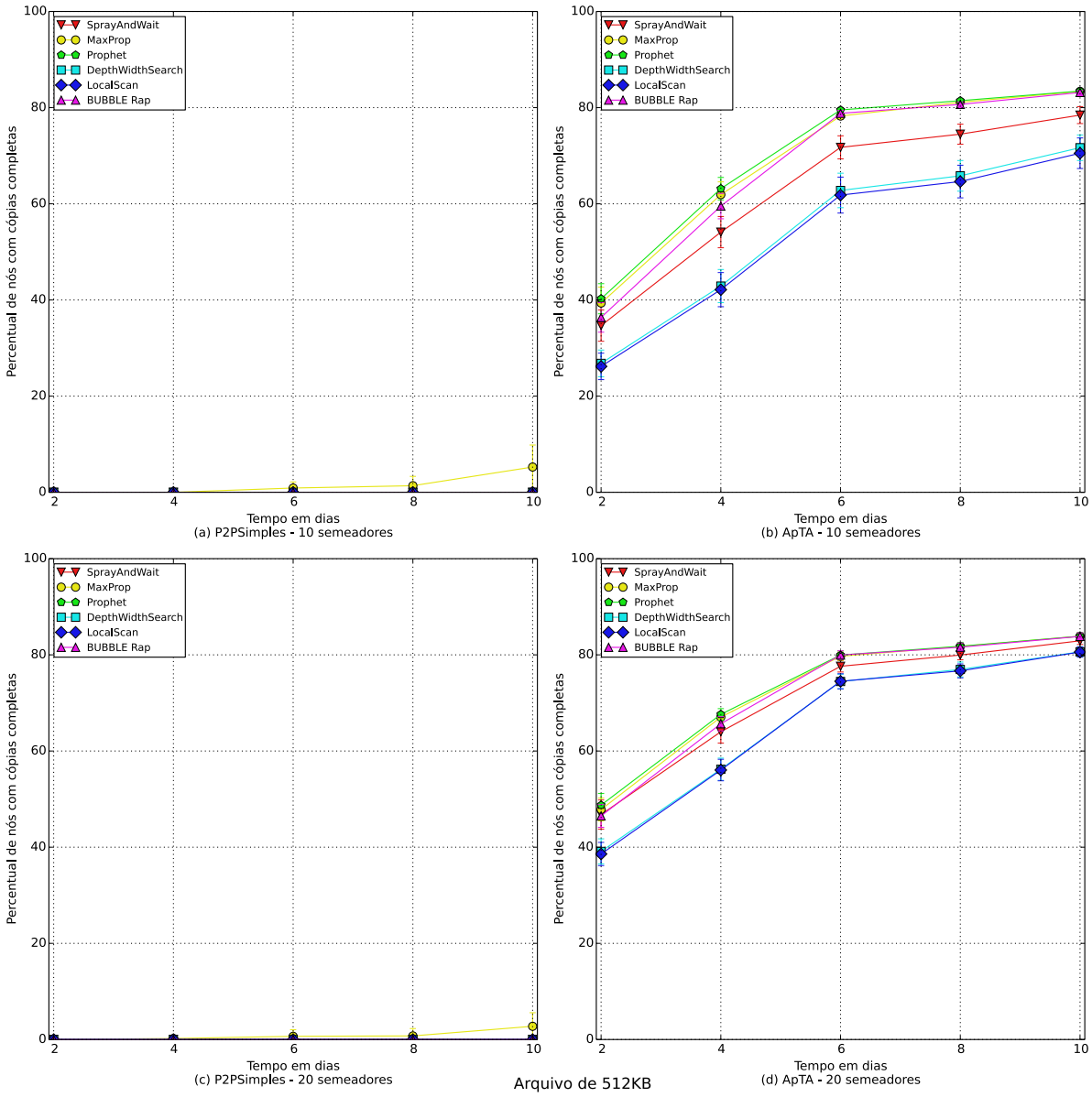


Figura 6.14 - Percentual de nós com cópias completas com arquivo de 512KB para 10 e 20 semeadores.

Quanto à aplicação ApTA, as figuras Figura 6.13, Figura 6.14, Figura 6.15 e Figura 6.16 mostram de imediato que os resultados foram muito melhores que na aplicação P2PSimples. Em aproximadamente 3 dias, a aplicação ApTA consegue distribuir o arquivo de 512K com 1 semeador para o mesmo número de nós que a aplicação P2PSimples em todo o período de simulação. Ao final da simulação, usando o protocolo de encaminhamento ProPHET, aproximadamente 65% dos usuários conseguiram completar o arquivo com apenas 1 semeador. Também teve bom

desempenho a aplicação ApTA com a utilização dos protocolos BUBBLE Rap e MaxProp com aproximadamente 60% de nós completos. Isso é mais do que 5 vezes o número de nós completos com a utilização da aplicação P2PSimples.

Outro ponto importante é que, com o aumento do número de mensagens em função do acréscimo de semeadores, a aplicação ApTA conseguiu que mais nós conseguisse completar o arquivo, ao contrário do que aconteceu com a aplicação P2PSimples, chegando a mais de 80% de nós com cópia completa do arquivo em seus buffers com todos os protocolos de encaminhamento.

Na Figura 6.16, percebemos que na distribuição do arquivo com tamanho de 4MB a análise é a mesma para o arquivo de 512KB, salvo que com 20 semeadores, a aplicação ApTA teve um desempenho ligeiramente menor. Com o uso dos protocolos DepthWidthSearch e LocalScan, o percentual de nós com o arquivo completo ficou em torno de 70% e com o protocolo Spray and Wait o percentual de nós que completaram o arquivo ficou um pouco abaixo de 80%, com os demais passando desse valor.

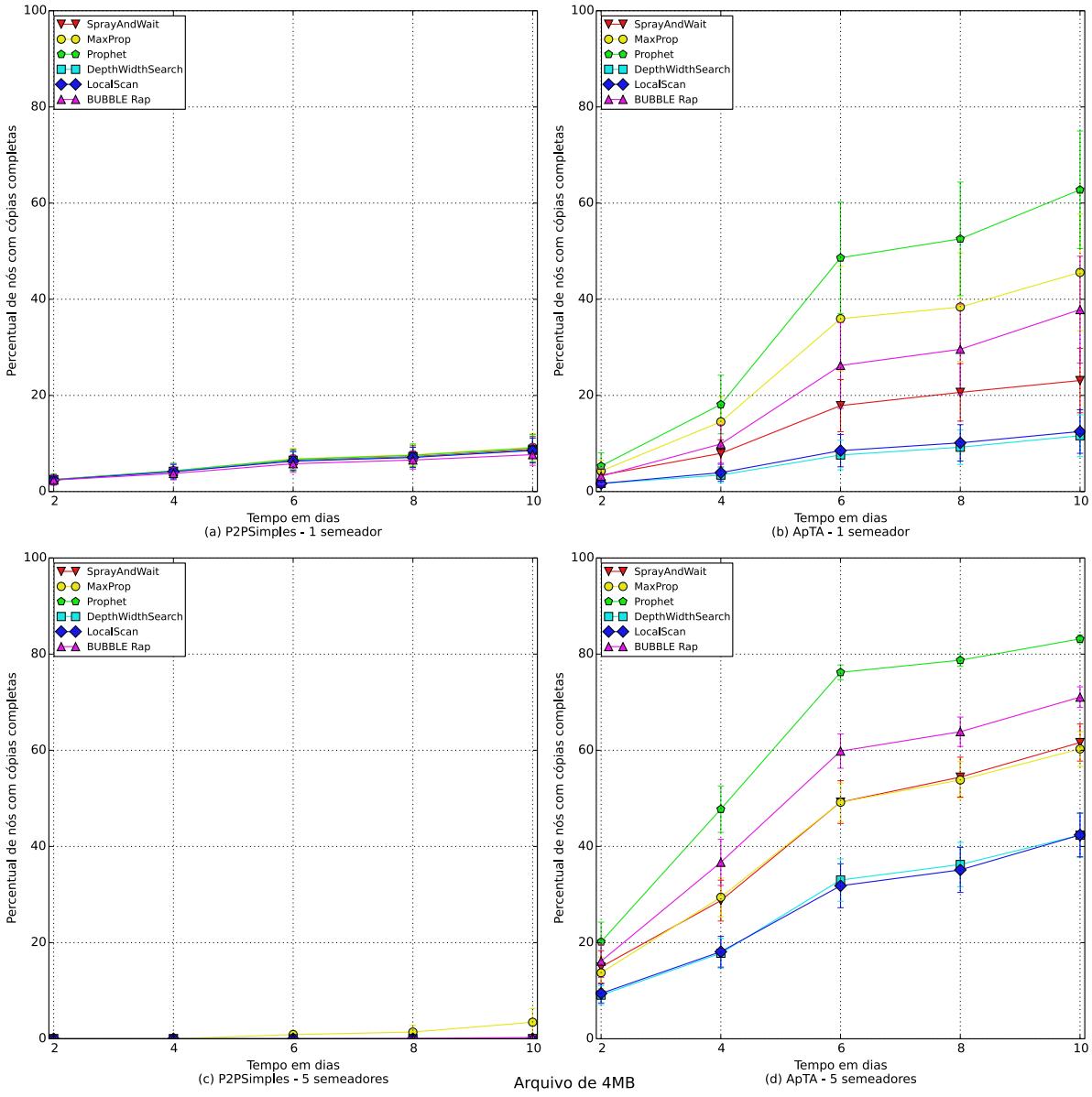


Figura 6.15 - Percentual de nós com cópias completas com arquivo de 4MB para 1 e 5 semeadores

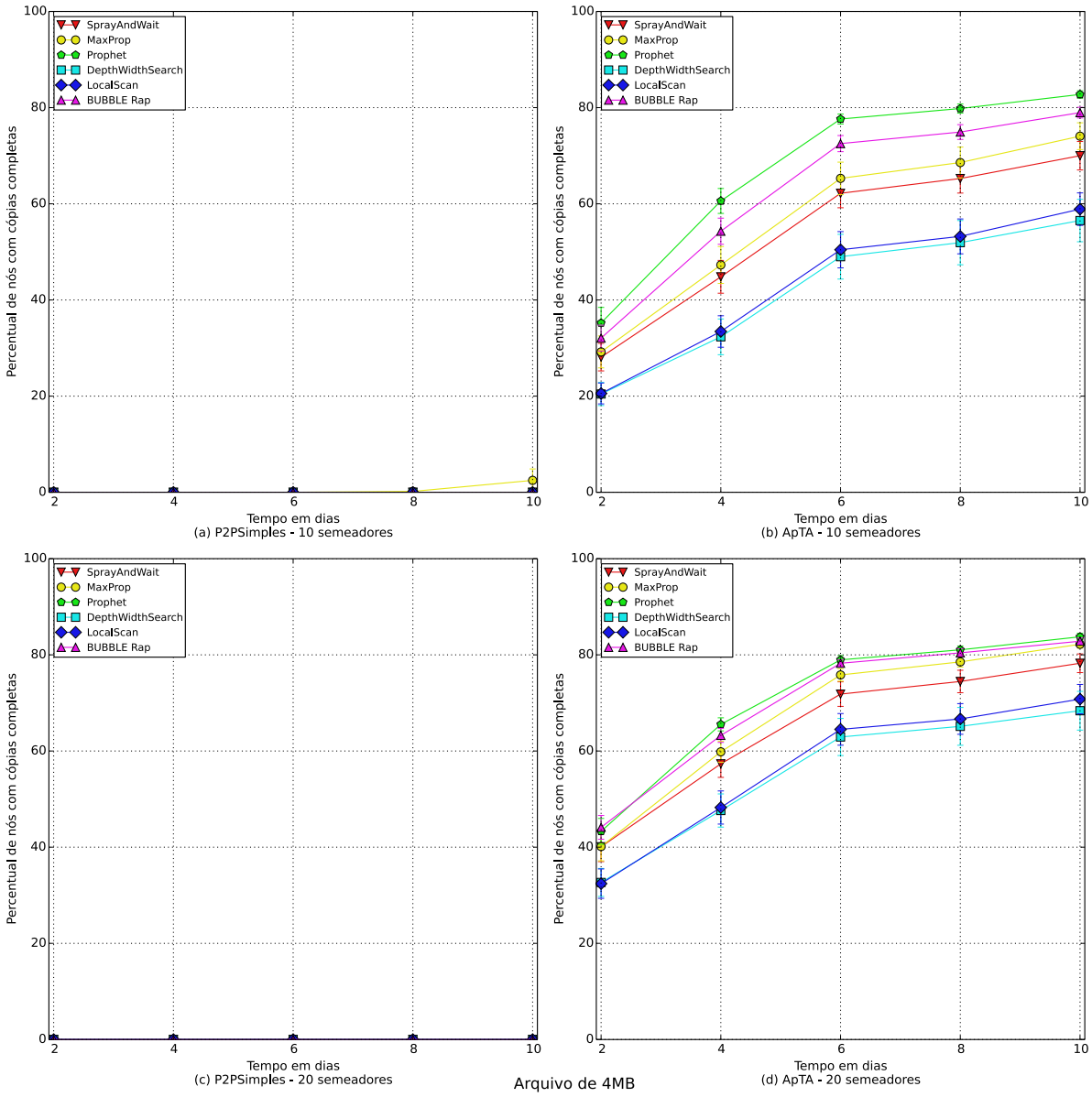


Figura 6.16 - Percentual de nós com cópias completas arquivo de 4MB para 10 e 20 semeadores.

6.6 Discussão dos resultados

Com todos os resultados expostos na seção anterior, é possível ver que a introdução das semânticas das aplicações nas simulações acarreta em uma melhora significativa nos resultados. Vimos também que é pertinente validar um protocolo de roteamento com métricas que façam sentido no âmbito da aplicação como fizemos com o LocalScan e não apenas com as métricas de rede. Este protocolo teve os melhores

valores referentes às métricas de rede atraso e sobrecarga e um valor mediano em relação à taxa de entrega, mas devido ao seu comportamento mais comedido em replicar mensagens, dividiu o último lugar em todas as situações com o protocolo DepthWidthSearch quando medimos o número de nós que conseguiram completar o arquivo. As tabelas abaixo exibem todos os resultados.

O comportamento da rede com o uso da aplicação ApTA permitiu criar mais mensagens, indicando que conseguiu realizar mais ciclos requisição-resposta. A latência foi menor, indicando que os mecanismos presentes na aplicação ApTA para aproximar o conteúdo do usuário ao utilizar os nós intermediários foi bem sucedida o que também é sustentado pelo número médio de saltos que os protocolos necessitaram para encaminhar as mensagens. A sobrecarga de mensagens foi significativamente menor que em comparação com a aplicação P2PSimples e a taxa de entrega se manteve alta com muito mais mensagens entregues.

7 Conclusão

Neste capítulo, serão apresentadas as considerações finais sobre todo o trabalho e algumas propostas de continuação do estudo.

7.1 Considerações Finais

Neste trabalho foi abordado o tema da troca de arquivos no cenário oportunista que é um cenário muito desafiador em função da elevada mobilidade do nós. Discorremos sobre a mobilidade humana e apresentamos o quão importante é compreender e aplicar os conhecimentos adquiridos sobre o movimento humano como frequência e distância de deslocamento, localidade, etc, para criar modelos que possam melhorar as estimativas de um possível futuro contato entre nós e com isso realizar a troca de uma eventual mensagem. Apresentamos também, algumas categorias de protocolos de encaminhamento de mensagens em redes oportunistas e abordamos as características de alguns dos principais representantes de cada categoria.

Apresentamos alguns problemas relacionados a aplicações de troca de arquivos em redes oportunistas como a representação vazia da camada de aplicação que pode interferir nos resultados de trabalhos. Analisamos a dificuldade em localizar conteúdo em uma rede oportunista livre de suporte de infraestrutura e, assim, não ser factível recuperar a informação de um servidor, pois para esse tipo de mecanismo funcionar, presume-se conectividade com o servidor e com o nó onde está armazenado o conteúdo. Abordamos também como prover com que os nós intermediários possam fornecer partes

do arquivo e, assim, diminuir o número de saltos na comunicação para se obter as partes do arquivo e como isso favorece a rápida disseminação do arquivo no enxame.

Assim, foi proposta a aplicação ApTA para atender a esses requisitos. Na análise de desempenho realizada, verificamos o comportamento da rede através das métricas número de mensagens criadas, atraso, latência, taxa de entrega, número de saltos. Em todas essas métricas, a configuração que utilizava a aplicação ApTA se saiu melhor, entregando mais mensagens, com menor atraso, sobrecarga e número de saltos. A última métrica mostrou que é factível o objetivo do presente trabalho além de mostrar que os mecanismos propostos para aproximar o conteúdo do usuário e acelerar a disseminação do conteúdo dentro do enxame conseguiram ter êxito e a aplicação conseguiu aproveitar o acréscimo de nós semeadores, fato que não foi percebido com a aplicação P2PSimples.

Verificamos que existe diferença entre trabalhos de avaliação de desempenho da camada de rede através de uma aplicação real e uma aplicação vazia, condição que no presente trabalho chamamos de representação vazia da camada de aplicação. É importante avaliar o impacto dessa condição no estudo de redes.

E por fim, apresentamos o protocolo de encaminhamento de mensagens em redes oportunistas LocalScan que, devido às suas características, foi o protocolo menos oneroso à rede.

7.2 Trabalhos Futuros

Como trabalhos futuros, seria interessante executar a simulação utilizando outros traces com dados de contatos reais para verificar se as características das aplicações se mantém. De preferência, traces que representem melhor um grande centro urbano, com grande número de nós e com menor número de reencontros.

Além disso, verificar variações nas estratégias de solicitação de partes aos semeadores. No presente trabalho, foi usado um ciclo requisição-resposta, mas talvez outras estratégias sejam mais vantajosas como, por exemplo, solicitar todas as partes do arquivo de uma vez ou solicitar partes distintas em um mesmo ciclo.

Verificar os efeitos da variação da população de nós que utilizam a aplicação. No presente trabalho, toda a população de nós utilizada na simulação executava a aplicação. É interessante verificar os atrasos em decorrência de um percentual menor de nós usando a aplicação, assim como verificar a aplicação com mais de um arquivo sendo distribuído ao mesmo tempo.

Verificar o comportamento do protocolo de encaminhamento LocalScan na distribuição de arquivos maiores como, por exemplo, 10MB, 20MB, 50MB e 100MB.

8 REFERÊNCIAS

[1] DAINOTTI, A. et al. **Analysis of country-wide internet outages caused by censorship**. Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. [S.l.]: ACM. 2011. p. 1--18.

[2] GREENFIELD, A. **Everyware:** The dawning age of ubiquitous computing. [S.l.]: New Riders, 2010.

[3] TAM, D. cnet.com. **CNET**, 2014. Disponivel em: <http://www.cnet.com/news/apple-170-million-ipads-sold/>.

[4] CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018. **Cisco.com**, 2014. Disponivel em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html>. Acesso em: jan. 2014.

[5] PELUSI, ; PASSARELLA, ; CONTI,. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. **Communications Magazine, IEEE**, v. 44, n. 11, p. 134--141, 2006.

[6] KARAMSHUK, D. et al. Human mobility models for opportunistic networks. **Communications Magazine, IEEE**, v. 49, n. 12, p. 157-165, 2011.

[7] LINDGREN, A.; HUI, P. **The quest for a killer app for opportunistic and delay tolerant networks**. Proceedings of the 4th ACM workshop on Challenged networks. [S.l.]: ACM. 2009. p. 59--66.

[8] SAAB, et al. **Secure delay-tolerant communications in the presence of oppressive governments**. Internet Technology and Secured Transactions (ICITST), 2011 International Conference for. [S.l.]: IEEE. 2011. p. 302--307.

[9] FALL, K. **A delay-tolerant network architecture for challenged internets**. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. [S.l.]: [s.n.]. 2003. p. 27-34.

[10] CHIOU, I. Y. A. L. M. T. **CAMPUSNET:** A Gateway-Network Approach to Interconnecting a Campus-wide Internet. Proceedings of the IEEE INFOCOM. [S.l.]: IEEE. 1985. p. 168--177.

[11] CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013‚Äì2018. **URL http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visualnetworking-index-vni/white_paper_c11-520862.html**.

[12] RIO, M. Metrô em Números. **Metrô em Números**. Disponivel em: <http://saladeimprensa.metrorio.com.br/sobre-o-metro-rio/metro-em-numeros/>. Acesso em: jan. 2014.

[13] IBGE. Acesso à Internet e posse de telefone móvel celular para uso pessoal 2011 - PNAD. **Acesso à Internet e posse de telefone móvel celular para uso pessoal 2011 - PNAD**. Disponivel em: <http://www.ibge.gov.br/home/estatistica/populacao/acessoainternet2011/default_pdf_celular.shtm>. Acesso em: jan. 2014.

[14] GONZALEZ, M. C.; HIDALGO, C. A.; BARABASI, A.-L. Understanding Individual Human Mobility Patterns. **Nature**, v. 453, n. 7196, p. 779--782, 2008.

[15] WANG, S. et al. Opportunistic routing in intermittently connected mobile p2p networks. **Selected Areas in Communications, IEEE Journal on**, v. 31, n. 9, p. 369--378, 2013.

[16] GONZALEZ, M. C.; HIDALGO, C. A.; BARABASI, A.-L. Understanding individual human mobility patterns. **Nature**, v. 453, n. 7196, p. 779-782, 2008.

[17] SONG, C. et al. Modelling the scaling properties of human mobility. **Nature Physics**, v. 6, n. 10, p. 818-823, 2010.

[18] LI, F.; WU, J. **LocalCom:** a community-based epidemic forwarding scheme in disruption-tolerant networks. Sensor, Mesh and Ad Hoc Communications and Networks. [S.l.]: IEEE. 2009.

[19] BULUT, E.; SZYMANSKI, B. K. **Friendship based routing in delay tolerant mobile social networks**. Global Telecommunications Conference (GLOBECOM 2010). [S.l.]: IEEE. 2010.

[20] VAHDAT, ; BECKER, D.; ET AL. **Epidemic routing for partially connected ad hoc networks**. Duke University. [S.l.]. 2000.

[21] SHARMA, G.; MAZUMDAR, R.; SHROFF, N. Delay and Capacity Trade-offs in Mobile Ad Hoc Networks: A Global Perspective. **IEEE/ACM Transactions on Networking (ToN)**, v. 15, n. 5, p. 981--992, 2007.

[22] SPYROPOULOS, T.; PSOUNIS, K. **Spray and wait:** an efficient routing scheme for intermittently connected mobile networks. ACM SIGCOMM workshop on Delay-tolerant networking. [S.l.]: ACM. 2005.

[23] LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. **ACM SIGMOBILE mobile computing and communications review**, v. 7, n. 3, p. 19--20, 2003.

[24] BURGESS, J. et al. **MaxProp:** Routing for Vehicle-Based Disruption-Tolerant Networks. INFOCOM. [S.l.]: IEEE. 2006. p. 1--11.

[25] LEGUAY, J.; FRIEDMAN, T.; CONAN, V. Evaluating mobility pattern space routing for DTNs. **INFOCOM**, 2006.

[26] BENEVENUTO, F.; JUSSARA, A. M.; ALTIGRAN, S. S. Explorando redes sociais online: Da coleta e análise de grandes bases de dados às aplicações. **Mini-cursos do Simpósio Brasileiro de Redes de Computadores (SBRC)**, 2011.

[27] DALY, E. M. A. H. M. **Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs**. Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing. Montreal: ACM. 2007. p. 32--40.

[28] FREEMAN, L. C. A set of measures of centrality based on betweenness. **Sociometry**, p. 35--41, 1977.

[29] FREEMAN, L. C. Centrality in social networks conceptual clarification. **Social networks**, v. 1, n. 3, p. 215--239, 1979.

[30] NEWMAN, M. E. Clustering and preferential attachment in growing networks. **Physical Review E**, v. 62, n. 2, p. 025102, 2001.

[31] HUI, P. et al. **Pocket switched networks and human mobility in conference environments**. Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking. [S.l.]: [s.n.]. 2005. p. 244-251.

[32] MTIBAA, A. et al. **Peoplerank:** Social opportunistic forwarding. INFOCOM, 2010 Proceedings IEEE. [S.l.]: IEEE. 2010. p. 1--5.

[33] PAGE, L.; BRIN, S.; MOTWANI, R. The PageRank citation ranking: Bringing order to the web, 1999.

[34] DALY, E. M.; HAAHR, M. Social network analysis for information flow in disconnected delay-tolerant MANETs. **Mobile Computing, IEEE Transactions on**, v. 8, n. 5, p. 606--621, 2009.

[35] EVERETT, M.; BORGATTI, S. P. Ego network betweenness. **Social networks**, v. 27, n. 1, p. 31--38, 2005.

[36] KUROSE, J. F. **Computer networking:** a top-down approach - 6th ed. [S.l.]: Pearson Education India, 2013.

[37] PASSARELLA, A. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. **Computer Communications**, 35, n. 1, 2012. 1--32.

[38] COHEN, B. The BitTorrent protocol specification, 2008. **URL http://bittorrent.org/beps/bep_0003.html**, 2008.

[39] BITTORRENT.ORG. DHT Protocol. **Bittorrent.org**, 2014. Disponivel em: <http://www.bittorrent.org/beps/bep_0005.html>.

[40] TSENG, Y.-C.; PAN, M.-S.; TSAI, Y. Wireless sensor networks for emergency navigation, v. 39, n. 7, p. 55--62, 2006.

[41] PAN, M.-S.; TSAI, C.-H.; TSENG, Y. Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks. **International Journal of Sensor Networks**, v. 1, n. 1, p. 2-10, 2006.

[42] BARNES, ; LEATHER, ; ARVIND, D. **Emergency evacuation using wireless sensor networks**. Local Computer Networks. [S.l.]: IEEE. 2007. p. 851--857.

[43] LI, S. et al. **ERN:** emergence rescue navigation with wireless sensor networks. Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on. [S.l.]: IEEE. 2009. p. 361--368.

[44] VAN WILLIGEN, W. et al. **WILLEM:** A wireless intelligent evacuation method. Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on. [S.l.]: IEEE. 2009. p. 382--387.

[45] JIANG, ; BIGHAM, ; BODANESE, E. **Adaptive service provisioning for emergency communications with DTN**. Wireless Communications and Networking Conference. [S.l.]: IEEE. 2011. p. 2125--2130.

[46] MERONI, P.; PAGANI, E.; ROSSI, G. P. **An opportunistic platform for android-based mobile devices**. International Workshop on Mobile Opportunistic Networking. [S.l.]: ACM. 2010. p. 191--193.

[47] PIETILÄINEN, A.-K. et al. **MobiClique:** middleware for mobile social networking. Proceedings of the 2nd ACM workshop on Online social networks. [S.l.]: ACM. 2009. p. 49--54.

[48] SCOTT, et al. **Haggle:** A networking architecture designed around mobile users. WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services. [S.l.]: [s.n.]. 2006. p. 78--86.

[49] RISTANOVIC, N. A. T. G. A. L. B. J.-Y. **Traps and pitfalls of using contact traces in performance studies of opportunistic networks**. INFOCOM, 2012 Proceedings IEEE. [S.l.]: IEEE. 2012.

[50] KLEMM, ; LINDEMANN, C.; WALDHORST., O. P. **A special-purpose peer-to-peer file sharing system for mobile ad hoc networks.** Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th. [S.l.]: IEEE. 2003.

[51] YANG, G. et al. **Ad-hoc storage overlay system (asos):** A delay-tolerant approach in manets. Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on. [S.l.]: IEEE. 2006. p. Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on.

[52] PAPADOPOULI , M.; SCHULZRINNE, H. **Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices**. Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing. [S.l.]: ACM. 2001. p. 117--127.

[53] CISCO. Linksys WAP54G Wireless-G Access Point. Disponivel em: <http://downloads.linksys.com/downloads/datasheet/WAP54G-EU_V10_DS_A-WEB,0.pdf>. Acesso em: jun. 2014.

[54] JUNG, S. et al. Bluetorrent: Cooperative content sharing for bluetooth users. **Pervasive and Mobile Computing**, v. 3, n. 6, p. 609--634, 2007.

[55] LENDERS, V.; KARLSSON , G.; MAY, M. **Wireless ad hoc podcasting**. Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on. [S.l.]: IEEE. 2007. p. 273--283.

[56] DE PELLEGRINI , F. et al. **R-P2P:** a data centric DTN middleware with interconnected throwboxes. Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems. [S.l.]: ICST. 2008. p. 2.

[57] HELGASON, Ó. R. et al. **A mobile peer-to-peer system for opportunistic content-centric networking**. Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds. [S.l.]: ACM. 2010. p. 21--26.

[58] MCNAMARA, ; MASCOLO, C.; CAPRA, L. **Media sharing based on colocation prediction in urban transport**. Proceedings of the 14th ACM international conference on Mobile computing and networking. [S.l.]: ACM. 2008. p. 58--69.

[59] VALERIO, L.; BRUNO, ; PASSARELLA, A. **Adaptive data offloading in opportunistic networks through an actor-critic learning method**. Proceedings of the 9th ACM MobiCom workshop on Challenged networks. [S.l.]: ACM. 2014. p. 31--36.

[60] HAN, B. et al. **Cellular traffic offloading through opportunistic communications:** a case study. Proceedings of the 5th ACM workshop on Challenged networks. [S.l.]: ACM. 2010. p. 31--38.

[61] CONTI, M.; GIORDANO, S. Multihop ad hoc networking: The reality. **Communications Magazine, IEEE**, 45, n. 4, 2007. 88--95.

[62] CONTI, M.; GIORDANO, S. Mobile ad hoc networking: milestones, challenges, and new research directions. **Communications Magazine, IEEE**, 52, n. 1, 2014. 85--96.

[63] ADLER, S. The Slashdot effect: an analysis of three Internet publications. **Linux Gazette**, 38, 1999. 2.

[64] CHOO, C.; SESHADRI, P. V.; CHAN , M. C. Application-aware disruption tolerant network. **Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on**, 2011. 1--6.

[65] THEORY.ORG. Bittorrent Protocol Specification v1.0. **theory.org**. Disponivel em: <https://wiki.theory.org/BitTorrentSpecification>. Acesso em: jan. 2014.

[66] EUGSTER, T. et al. The many faces of publish/subscribe. **ACM Computing Surveys (CSUR)**, v. 35, n. 2, p. 114--131, 2003.

[67] HOSSMANN, T.; SPYROPOULOS, ; LEGENDRE, F. **Know thy neighbor:** Towards optimal mapping of contacts to social graphs for dtn routing. INFOCOM, 2010 Proceedings IEEE. [S.l.]: IEEE. 2010. p. 1--9.

[68] CRAWDAD. Crawdad. **Crawdad.org**, 2014. Disponivel em: <http://crawdad.org/>.

[69] KOTZ, D.; ESSIEN, K. Analysis of a campus-wide wireless network. **Wireless Networks**, v. 11, n. 1-2, p. 115--133.

[70] KERÄNEN, A.; OTT, ; KÄRKKÄINEN, T. **The ONE simulator for DTN protocol evaluation**. Proceedings of the 2nd international conference on simulation tools and techniques. [S.l.]: ICST. 2009. p. 55.

[71] DILLON, P. J. **University of Pittsburgh**, 2014. Disponivel em: <http://people.cs.pitt.edu/~pdillon/one/routing/>.

[72] LEGOUT; URVOY-KELLER; MICHIARDI. **Rarest First and Choke Algorithms Are Enough**. ACM SIGCOMM Internet Measurement Conference. [S.l.]: [s.n.]. 2006.

[73] JUANG, P. et al. **Energy-efficient computing for wildlife tracking:** Design tradeoffs and early experiences with ZebraNet. ACM Sigplan Notices. [S.l.]: [s.n.]. 2002. p. 96-107.

[74] HARTENSTEIN, H.; LABERTEAUX, K. P. A tutorial survey on vehicular ad hoc networks. **Communications Magazine, IEEE**, v. 46, n. 6, p. 164--171, 2008.

[75] OTT, J.; PITKANEN, J. **Dtn-based content storage and retrieval**. World of Wireless, Mobile and Multimedia Networks, 2007. [S.l.]: IEEE. 2007. p. 1--7.

[76] PASSARELLA, A. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. **Computer Communications**, 35, n. 1, 2012. 1--32.

[77] VALERIO, L.; BRUNO, R.; PASSARELLA, A. **Adaptive data offloading in opportunistic networks through an actor-critic learning method**. Proceedings of the 9th ACM MobiCom workshop on Challenged networks. [S.l.]: ACM. 2014. p. 31--36.

[78] THE INTERNET ENGINEERING TASK FORCE. RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1. **URL http://www.ietf.org/rfc/rfc2616.txt**.