



HEWERTON ENES DE OLIVEIRA

**GRUBIROUTER: UM PROTOCOLO DE
ROTEAMENTO PARA REDES
OPORTUNISTAS COM PADRÃO DE
MOVIMENTO SOCIOLÓGICO**

LAVRAS - MG

2014

HEWERTON ENES DE OLIVEIRA

**GRUBIROUTER: UM PROTOCOLO DE ROTEAMENTO
PARA REDES OPORTUNISTAS COM PADRÃO DE
MOVIMENTO SOCIOLÓGICO**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

Orientador

Dr. Tales Heimfarth

LAVRAS - MG

2014

**Ficha Catalográfica Elaborada pela Coordenadoria de Produtos
e Serviços da Biblioteca Universitária da UFLA**

Oliveira, Hewerton Enes de.

GrubiRouter: Um protocolo de roteamento para redes
oportunistas com padrão de movimento sociológico/ Hewerton Enes
de Oliveira. – Lavras : UFLA, 2014.

87 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2014.

Orientador: Tales Heimfarth.

Bibliografia.

1. Redes oportunistas. 2. Roteamento. 3. Redes tolerantes a
falhas. 4. Ad Hoc. I. Universidade Federal de Lavras. II. Título.

CDD – 004.6

HEWERTON ENES DE OLIVEIRA

**GRUBIROUTER: UM PROTOCOLO DE ROTEAMENTO
PARA REDES OPORTUNISTAS COM PADRÃO DE
MOVIMENTO SOCIOLÓGICO**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

APROVADA em 19 de março de 2014.

Dr. Wilian Soares Lacerda UFLA

Dr. Edison Pignaton de Freitas UFSM

Dr. Tales Heimfarth

Orientador

LAVRAS - MG

2014

*Dedico esta dissertação aos meus pais, José Leôncio de Oliveira e
Vicentina Enes de Oliveira, e à minha irmã Hewellin Enes de Oliveira que
me apoiaram incondicionalmente durante toda essa jornada.*

AGRADECIMENTOS

Agradeço ao meu orientador professor Tales pelas dicas, conselhos e a plena disponibilidade.

Agradeço aos meus amigos Renato Oliveira e Ana França pelo total apoio e amizade desde a formatura da graduação.

Agradeço a minha namorada Alessandra Millezi pelo companheirismo e compreensão durante todo o mestrado.

Agradeço ao Grupo de Redes Ubíquas pelas amizades proporcionadas durante todo o mestrado e parte da graduação.

Agradeço aos meus colegas da primeira turma do mestrado do PPGCC da UFLA pela amizade e pela troca de experiências nesses dois anos de luta.

Agradeço a todos os meus colegas do PPGCC da UFLA pela ajuda durante as disciplinas do mestrado.

Agradeço a todos os professores do PPGCC da UFLA pelas instruções e conhecimentos passados.

Agradeço à secretaria do PPGCC da UFLA pela pronta disponibilidade.

Agradeço à CAPES pelo apoio financeiro concedido durante o desenvolvimento deste trabalho.

RESUMO

Este trabalho apresenta um protocolo de roteamento para redes oportunistas que se baseia no modelo de movimento sociológico. Neste modelo os nós móveis tendem a frequentar regiões sociais, como restaurantes e universidades, de forma a realizar os mesmos movimentos entre as regiões repetidamente. Um desafio para este ambiente é realizar o roteamento de mensagens, uma vez que a topologia da rede muda frequentemente. O protocolo de roteamento apresentado mostra uma maneira de resolver este problema estabelecendo pontos de acesso no ambiente da rede. Estes pontos de acesso são denominados de *hubs*, formado pela associação de vários nós móveis da rede que estão presentes nas regiões sociais. Por meio das informações de contexto, como a posição e histórico de encontros com outros nós, é possível identificar quais nós realizam rotas frequentes entre *hubs*. Dessa forma eles são utilizados como arestas para interligar os *hubs* da rede e realizar a transferência de mensagens entre os mesmos. Assim, rotas são formadas na rede e utilizadas para o roteamento de mensagens. As mensagens são encaminhadas para o *hub* que é frequentado pelo destinatário. A entrega é efetuada quando o destinatário entra em contato com o *hub* que possui relação. O protocolo foi testado por meio de simulações e comparado com outros protocolos de roteamento para redes oportunistas. Os resultados mostram que o protocolo desenvolvido obteve desempenho satisfatório comparado a outros métodos tradicionais descritos na literatura.

Palavras-chave: Redes Oportunistas; Roteamento; Rede Tolerantes a Falhas; Ad Hoc.

ABSTRACT

This paper presents a routing protocol for opportunistic networks that is based on a sociological movement model. In this model the mobile nodes tend to attend social areas such as restaurants and universities, in order to perform the same moves repeatedly between regions. One challenge for this environment is to perform message routing, since the network topology changes frequently. The routing protocol developed shows one way to solve this problem by establishing access points in the network environment. These access points are called hubs, formed by the association of multiple mobile network nodes that are present in social areas. By means of the context information, such as position and history of encounters with other nodes, it is possible to identify which nodes perform frequent routes between hubs. Thus they are used as edges to interconnect the hubs of the network and perform the transfer of messages therebetween. Thus, routes are formed in the network and used for routing messages. Messages are gathered for the hub which is frequented by the recipient. Delivery is made when the recipient contacts the hub that has a relationship. The protocol was tested through simulations and compared with other routing protocols for opportunistic networks. The results show that the protocol developed got satisfactory performance compared to other traditional methods described in the literature.

Keywords: Opportunistic Networks; Routing; Delay Tolerant Networks; Ad Hoc.

LISTA DE FIGURAS

Figura 1	Cenário considerado para o problema. Observa-se a movimentação dos nós e os contatos feitos na universidade, escritório, clube, restaurante e em casa.....	17
Figura 2	Estrutura de uma aplicação para Redes Oportunistas.....	23
Figura 3	Modelo da rede para disseminação de informações.....	24
Figura 4	Algoritmo de crescimento de uma OppNet.....	25
Figura 5	Visão geral do simulador <i>The ONE</i> (KERÄNEN; OTT; KÄRKKÄINEN, 2009).....	29
Figura 6	Interface gráfica do simulador <i>The ONE</i>	31
Figura 7	Janela <i>pop-up</i> com informações detalhadas <i>The ONE</i>	32
Figura 8	Mapa da área central de <i>Helsinki</i> utilizado nas simulações (KERÄNEN; OTT; KÄRKKÄINEN, 2009).	33
Figura 9	Exemplo de um ambiente par ao protocolo <i>Solar-Hub</i> . Fonte: (GHOSH; PHILIP; QIAO, 2005).....	39
Figura 10	Taxonomia proposto por Cao e Sun (2012),.....	47
Figura 11	Visão geral do protocolo desenvolvido.....	49
Figura 12	Zona âncora de um item, nós móveis e seus alcances de comunicação: o conteúdo é replicado entre os nós dentro da zona e deletados dos nós que estão a uma distância maior que a do centro da região. (KANGASHARJU; OTT; KARKULAHTI, 2010)	50
Figura 13	Cenário para exemplificação do protocolo.	51
Figura 14	Exemplo da coleta das informações de contexto pelos nós N1 e N2 e transferência das mesmas para os <i>hubs</i> com os quais os nós mantêm contato.	54
Figura 15	Exemplo do compartilhamento das tabelas de roteamento entre os <i>hubs</i> da rede.	59
Figura 16	Fluxograma detalhando o funcionamento do processo de roteamento.	63
Figura 17	Exemplo do roteamento de mensagens.	64
Figura 18	Exemplo de uma rede criada pelo gerador. Notam-se os pontos fixos denominados <i>hubs</i> e as arestas que os conectam.....	68
Figura 19	Tempo médio para propagação completa das tabelas de roteamento em cada cenário.	71
Figura 20	Taxa de entrega calculada para cada protocolo. Os dados foram obtidos do experimento 1.....	73

Figura 21	Taxa de entrega calculada para cada protocolo. Os dados foram obtidos do experimento 2.....	73
Figura 22	Número médio de mensagens de controle para cada cenário. Os dados foram obtidos do experimento 1.	76
Figura 23	Número médio de mensagens de controle para cada cenário. Os dados foram obtidos do experimento 2.	76
Figura 24	Latência média para entrega de mensagens. Os dados foram obtidos do experimento 1.....	78
Figura 25	Latência média para entrega de mensagens. Os dados foram obtidos do experimento 2.....	80

LISTA DE TABELAS

Tabela 1	Exemplo de tabela gerada após o processamento das informações de contexto para nó móvel N1.....	54
Tabela 2	Exemplo de tabela gerada após o processamento das informações de contexto para o nó móvel N2.	55
Tabela 3	Exemplo de tabela de roteamento gerada no <i>hub</i> H1 após o processamento das informações transferidas a partir do nó N1.....	57
Tabela 4	Exemplo de tabela de roteamento gerada no <i>hub</i> H2 após o processamento das informações transferidas a partir do nó N1 e N2.	57
Tabela 5	Exemplo de tabela de roteamento gerada no <i>hub</i> H3 após o processamento das informações transferidas a partir do nó N2.....	57
Tabela 6	Exemplo de tabela gerada após o processamento das informações de contexto coletadas pelos nós móveis.	60
Tabela 7	Parâmetros para o experimento 1. Nesse experimento variou-se o número de <i>hubs</i> na rede.	69
Tabela 8	Parâmetros para o experimento 2. Nesse experimento variou-se o número de nós aleatórios e visitantes para observar o comportamento dos protocolos quanto a quantidade de nós.	69
Tabela 9	Configuração dos computadores utilizados na execução das simulações.	70
Tabela 10a	Estatísticas sobre a taxa de entrega. Resultados obtidos do experimento 1.	72
Tabela 10b	Estatísticas sobre o <i>overhead</i> . Resultados obtidos do experimento 2.	72
Tabela 11a	Estatísticas sobre o <i>overhead</i> . Resultados obtidos do experimento 1.	75
Tabela 11b	Estatísticas sobre o <i>overhead</i> . Resultados obtidos do experimento 2.	75
Tabela 12a	Estatísticas sobre a latência para entrega de mensagens. Resultados obtidos do experimento 1.....	79
Tabela 12b	Estatísticas sobre a latência para entrega de mensagens. Resultados obtidos do experimento 2.....	79

LISTA DE SIGLAS

UFLA	Universidade Federal de Lavras
GRUBI	Grupo de Redes Ubíquas
OppNet	Opportunistic Nertwork
Manet	Mobile Ad Hoc Network
The ONE	The Opportunistic Network Environment

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização e motivação	14
1.2	O problema	15
1.3	Objetivos gerais e específicos	18
1.4	Organização do trabalho	18
2	REFERENCIAL TEÓRICO	20
2.1	Redes Oportunistas	20
2.1.1	Conceitos básicos	21
2.1.1.1	Tipos de nós	22
2.1.1.2	Projetos de Redes Oportunistas	22
2.1.2	Ciclo de vida	24
2.1.3	Diferenças entre MANETs e OppNets	26
2.1.4	Diferenças entre Redes <i>Peer-to-Peer</i> e OppNets	27
2.2	O simulador <i>The ONE</i> para Redes Oportunistas	28
2.2.1	Configuração	29
2.2.2	Saída das simulações	31
2.2.3	O mapa utilizado	33
2.3	Estado da Arte sobre Roteamento em Redes Oportunistas	34
2.3.1	Taxinomia dos métodos de roteamento	35
2.3.2	Roteamento com contexto óbvio	35
2.3.3	Roteamento baseado na mobilidade dos nós	37
2.3.4	Roteamento baseado nas informações sociais	42
3	METODOLOGIA	48
3.1	Visão geral do protocolo <i>GrubiRouter</i>	49
3.2	Fase 1 - Reconhecimento	52
3.3	Fase 2 - Compartilhamento	56
3.4	Controle do <i>buffer</i>	61
3.5	Roteamento	62
4	RESULTADOS E DISCUSSÕES	66
4.1	Implementação do padrão de movimento	66
4.2	Gerador de cenários	67
4.3	Configurações e execução das simulações	68
4.4	Resultados obtidos	70
4.4.1	Análise sobre o compartilhamento das rotas	70
4.4.2	Análise sobre a taxa de entrega	71
4.4.3	Análise sobre o <i>overhead</i>	74

4.4.4	Análise sobre a latência	78
5	CONCLUSÃO	81
5.1	Trabalhos futuros	82
	REFERÊNCIAS.....	84

1 INTRODUÇÃO

1.1 Contextualização e motivação

O rápido desenvolvimento de novos e baratos dispositivos para redes sem fio tem criado grandes oportunidades para redes de comunicação em novas situações. Dispositivos equipados com tecnologia de comunicação sem fio são usados pervasivamente na vida diária. A ubiquidade desses dispositivos pode facilitar a comunicação em situações extremas nas quais não há uma rede estruturada, por exemplo, em áreas afetadas por desastres naturais como terremotos.

Dispositivos móveis podem formar redes isoladas, conectadas por um *link* sem fio como ocorre com as *Mobile Ad Hoc Networks* (MANETs) (SINGH; SINGH; VATS, 2011). As Redes Oportunistas (OppNets) reforçam esta visão considerando as características das redes móveis como, por exemplo, a mobilidade nos nós, a desconexão e as partes isoladas da rede. Os nós em uma OppNet exploram oportunidades de comunicação que a mobilidade lhe proporciona.

Um problema fundamental em OppNets é como realizar o roteamento de mensagens da origem até o destinatário, considerando que o caminho entre os nós (caminho fim-a-fim) pode ser indeterminado durante o tempo de vida da mensagem (LILIEEN et al., 2007). Como não há um caminho entre a origem e o destino, os nós da rede necessitam comunicar-se entre si através de conexões oportunistas e utilizar um método *store-carry-forward* (armazena-carrega-encaminha) para realizar o encaminhamento das mensagens (NGUYEN; GIORDANO, 2009).

O roteamento em OppNets pode ser melhorado obtendo conheci-

mento sobre o ambiente em que o nó da rede se encontra, isto é, obtendo as informações sobre o contexto no qual o nó se encontra. As informações de contexto podem ser definidas como qualquer informação que pode ser usada para proporcionar um roteamento eficiente e efetivo (NGUYEN; GIORDANO, 2012).

Dessa forma é possível desenvolver métodos para realizar o roteamento em uma OppNet que utilizem essas informações e estabeleçam rotas entre transmissor e receptor, possibilitando a transferência de mensagens entre os nós da rede. Os protocolos presentes na literatura que utilizam essa metodologia mostraram-se promissores, porém os mesmos enfrentam problemas de eficiência como altas taxas de *overhead*. Assim, foi desenvolvido neste trabalho um protocolo de roteamento que utiliza as informações de contexto para realizar o roteamento na rede que minimiza os recursos utilizados e que efetua um roteamento de forma eficiente.

1.2 O problema

O roteamento de mensagens entre dois nós pertencentes a uma rede oportunistas de nós móveis é o problema estudado neste trabalho.

O problema ambienta-se em uma rede constituída por nós móveis que possuem a capacidade de efetuar conexões sem fio entre si e armazenar informações sobre o contexto no qual está envolvido, como o histórico de contatos. Esses nós móveis se comunicam por meio de uma Rede Oportunista, ou seja, utilizando conexões ocasionais em momentos de proximidade entre diferentes participantes da rede.

Nesse ambiente estão presente também pontos fixos que representam localidades como um escritório ou universidade, os quais são visitados

periodicamente por um subconjunto dos nós móveis da rede. Além disso alguns nós móveis realizam movimentos entre os pontos fixos do ambiente. Assim, os nós realizam um padrão de movimento baseado na movimentação rotineira de pessoas em seu cotidiano, denominado padrão de movimento sociológico.

A Figura 1 exemplifica o ambiente utilizado para definir o problema analisado nesse projeto. Nesse cenário existem nós móveis que possuem acesso a um determinado número de recursos de *hardware*, como a comunicação sem fio, essencial para formar uma rede oportunista. Os nós desse cenário podem mover-se livremente em qualquer direção, pois os mesmos estão em posse de seu portador. Assim, em determinados momentos, um grupo de nós pode aproximar-se e afastar-se dependendo da direção do movimento de seus portadores. No cenário apresentado não é utilizada uma rede infraestruturada, por isso torna-se necessária a interação entre os nós para a realização da comunicação.

Por exemplo, um determinado grupo de nós pode pertencer a estudantes de uma determinada disciplina que possui um horário definido. Isso implica que em alguns momentos esses nós podem ficar próximos por um determinado intervalo de tempo. Após esse intervalo os nós podem novamente entrar em movimento. Além disso, os portadores podem ter relações sociais com colegas de trabalhos no escritório, com a sua família em casa e com amigos em um determinado clube. Isso faz com que os nós tenham contato com um número diversificado de nós.

Nesse cenário deseja-se enviar uma mensagem a partir do nó S que frequenta a universidade para o nó R , o qual frequenta o escritório. O problema analisado nesse projeto trata-se de como efetuar o roteamento de

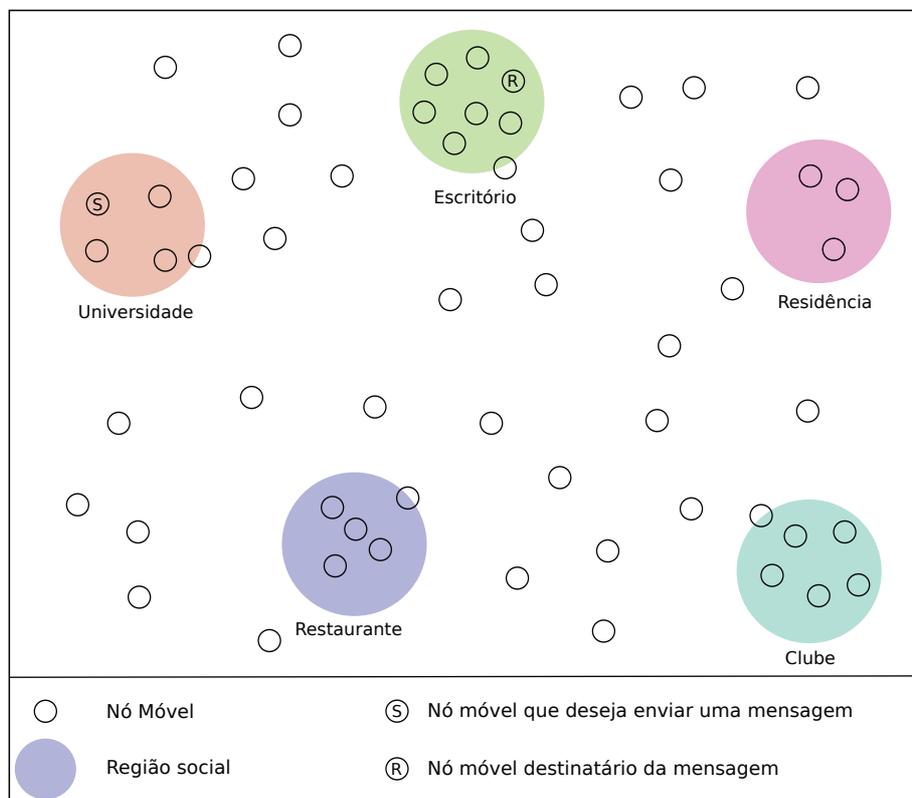


Figura 1 Cenário considerado para o problema. Observa-se a movimentação dos nós e os contatos feitos na universidade, escritório, clube, restaurante e em casa.

algum nó móvel qualquer para outro nó móvel qualquer, considerando que os nós podem interagir entre si para compor uma OppNet e realizar a troca de informações, pressupondo que os nós realizam um padrão de movimento sociológico. Um ponto específico do problema é como determinar o próximo nó para o qual a mensagem deve ser encaminhada.

1.3 Objetivos gerais e específicos

O objetivo geral deste trabalho é desenvolver um protocolo de roteamento para Redes Oportunistas que utilize informações de contexto para estabelecer uma rota entre transmissor e receptor a fim de realizar a transferência de mensagens de forma eficiente, minimizando os recursos utilizados. Os protocolos descritos na literatura apresentam altas taxas de *overhead* comprometendo sua eficiência quanto ao uso de recursos e entrega de mensagens. O roteamento deve ser realizado pressupondo que os nós da rede realizam o padrão de movimento sociológico apresentado anteriormente.

Os seguintes objetivos específicos foram estabelecidos:

- Extrair dados de contexto, como encontros e posições geográficas do dispositivo e criar uma base de dados de alto nível.
- Criar um método para roteamento que utilizará a base de dados para estabelecer qual é o próximo nó a ser utilizado no roteamento.
- Desenvolver uma aplicação para testar o protocolo desenvolvido.
- Desenvolver um gerador de cenários aleatórios para serem utilizados nos testes.
- Avaliar o desempenho dos métodos propostos utilizando simulação e comparar com métodos presentes na literatura.

1.4 Organização do trabalho

O trabalho está organizado da forma apresentada a seguir. Na Seção 2 será apresentado um referencial teórico acerca dos conceitos utilizados. Na Seção 3 será exposta a metodologia do trabalho desenvolvido. Na Seção 4

serão apresentadas as simulações computacionais realizadas e os resultados obtidos. As conclusões e os trabalhos futuros podem ser observados na Seção 5.

2 REFERENCIAL TEÓRICO

Os pontos relevantes para o entendimento deste trabalho estão descritos neste capítulo. Uma revisão bibliográfica foi feita com o objetivo de estabelecer as bases para seu desenvolvimento e para obter os recentes avanços na área.

2.1 Redes Oportunistas

As Redes Oportunistas, ou OppNets, são redes que correspondem a conexão oportunista de redes *Ad Hoc*, *Peer-to-Peer* (P2P) e Redes de Sensores Sem Fio (RSSFs). Elas aumentam a eficiência das aplicações sem fio existentes e abrem novas vertentes para o desenvolvimento de soluções e aplicações. São formadas por conexões *Ad Hoc* entre vários nós sem fio e móveis, sendo que o tempo de conexão entre eles é curto, ou seja, a conexão é temporária. Os nós podem juntar-se a uma rede ou simplesmente abandoná-la levando em consideração a sua utilidade para os serviços prestados pela rede em um determinado momento.

De acordo com Pelusi, Passarella e Conti (2006) os conceitos das Redes Oportunistas vieram de estudos sobre as Redes Tolerantes a Falhas (*Delay Tolerant Networks* - DTN) que foram conduzidos pelo *Internet Research Task Force* (Força Tarefa para Pesquisas com a Internet) que são responsáveis pelo desenvolvimento das especificações das redes DTN. A arquitetura de uma DTN consiste em uma rede de nós que realizam uma conexão similar à conexão com a internet, havendo somente conexões ocasionais realizadas quando ocorre uma aproximação entre eles, ou seja, quando ocorre oportunidades de conexão.

Pelusi, Passarella e Conti (2006) descrevem também que ainda não

há uma clara separação de conceitos entre redes DTN e Redes Oportunistas. Os dois termos são utilizados de forma frequente e intermitente. Em sua visão as Redes Oportunistas representam um conceito geral que abrange as redes DTN.

A topologia das OppNets varia de acordo com a mobilidade dos nós, inserção e remoção dos mesmos por alguma falha como o fim da carga da bateria. Uma OppNet deve possuir no mínimo duas funcionalidades básicas que são a *Node Discovery* e a *One-Hop Exchange*. A funcionalidade *Node Discovery* permite que os nós da rede pesquisem outros nós, dentro do alcance de seu rádio, para estabelecer conexões. Já a funcionalidade *One-Hop Exchange* permite a troca de informações entre os nós conectados.

As OppNets diferem das redes tradicionais, como as MANETs (Mobile *Ad Hoc* Networks), na mobilidade dos nós e na topologia dinâmica da rede que se justifica no contexto da aplicação. A comunicação entre os nós da rede é possível somente quando os nós se encontram dentro do mesmo alcance de transmissão. As OppNets podem se auto-organizar para executar uma tarefa em conjunto e permitem detectar dispositivos vizinhos utilizando vários meios de comunicação como o *ZigBee* (FARAHANI, 2008), *Bluetooth*, *WiFi*, etc (LEE; SU; SHEN, 2007). Segundo (MAKKI, 2007), as OppNets podem ser empregadas em situações de emergência, por exemplo catástrofes naturais.

2.1.1 Conceitos básicos

Nesta seção definem-se alguns conceitos relacionados às redes oportunistas e são resumidos alguns aspectos relacionados ao projeto de uma rede oportunista.

2.1.1.1 Tipos de nós

Nó: o nó de uma OppNet é um dispositivo que tem a capacidade de realizar comunicação sem fio utilizando protocolos de compartilhamento de disseminação de informações. O nó é chamado de nó móvel se o mesmo possuir mobilidade dentro da rede.

Nó *helper*: Os nós *helpers* são nós que não têm qualquer capacidade de criar informações para rede, mas que podem contribuir positivamente quanto ao processamento e capacidade de comunicação.

2.1.1.2 Projetos de Redes Oportunistas

O projeto de OppNets, para uma determinada aplicação, deve considerar alguns serviços dos quais destacam-se:

Presence Awareness Service: fornece à aplicação a informação dos nós e dos utilizadores que estão ativos no mesmo alcance do raio de transmissão. Esse serviço é responsável por detectar a presença de nós vizinhos que podem agregar-se à rede para oferecer serviços e recursos.

Message Exchange Service: permite que as mensagens sejam trocadas entre os nós que estão no mesmo alcance do raio de transmissão. Esse serviço não verifica o sucesso ou insucesso do envio ou recebimento das mensagens deixando essa tarefa para outro serviço chamado *Acknowledgment Service* que se encontra na camada de aplicação.

Information Filtering Service: este serviço é responsável por filtrar as

informações que trafegam na rede, evitando que mensagens desnecessárias ou maliciosas trafeguem na rede.

Security Service: serviço responsável por manter a integridade e autenticidade das comunicações; ele oferece serviços de encriptação e assinatura das informações.

Identity Management Service: gerencia como os utilizadores são identificados e seus status na rede.

O projeto de uma aplicação que utiliza uma OppNet deve seguir a estrutura apresentada na Figura 2. A Colaboração Ativa é responsável por coletar dados de contexto do ambiente e dos nós que estão próximos entre si, reconhecendo e definindo as responsabilidades dos mesmos na rede. A Colaboração Passiva é um estado do nó responsável por permitir a troca de mensagens entre ele e outros nós da rede, onde um nó servirá de intermediário para o envio e recebimento de mensagens (DATTA; QUARTERONI; ABERER, 2004).

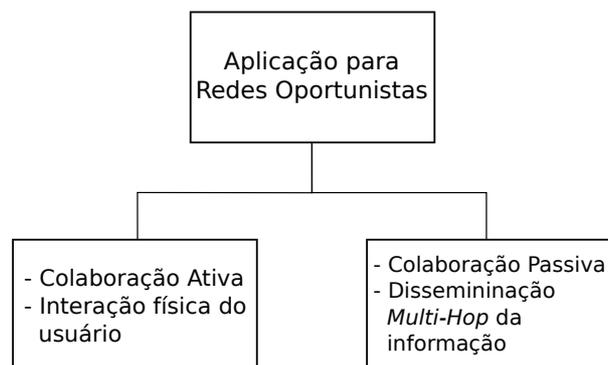


Figura 2 Estrutura de uma aplicação para Redes Oportunistas.

A disseminação de informação é o processo no qual a informação é difundida na OppNet. Ela é definida por protocolos de compartilhamento de informações e seguem dois passos explicados anteriormente: *Node Discovery* e *One-Hop Exchange*. Os nós presentes na OppNet definem seu modelo para distribuição da informação. Na Figura 3 é possível observar um modelo com nós móveis e fixos. Os nós representados por pequenos quadrados são pontos fixos de uma rede estruturada, eles possuem um raio de alcance maior para efetuar uma comunicação sem fio. Os nós representados por círculos são nós móveis com menos recursos de energia e comunicação, por isso seu raio de alcance sem fio é reduzido. Observa-se que os nós móveis podem conectar entre si e criar uma rede temporária para a disseminação de informações.

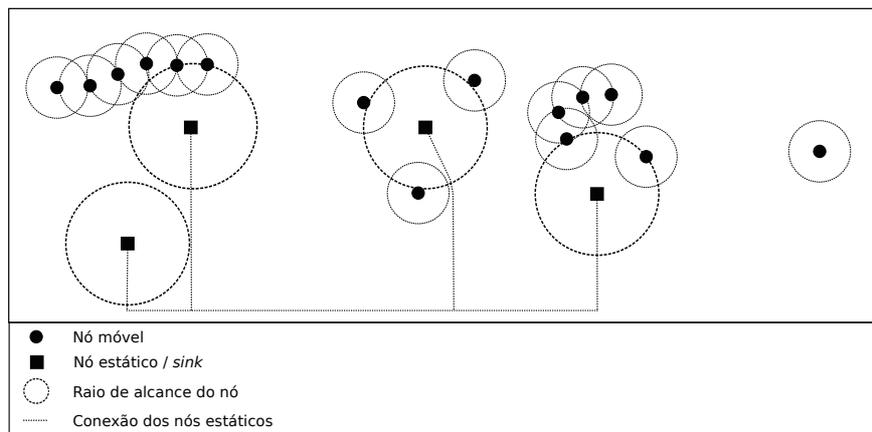


Figura 3 Modelo da rede para disseminação de informações.

2.1.2 Ciclo de vida

Uma OppNet inicia-se a partir de um conjunto de nós pré-definidos no momento de sua implementação. Após isso a OppNet começa a crescer ao redor desses nós alterando o número de indivíduos e sua topologia. O

crescimento ocorre através da detecção de nós vizinhos e sua integração à OppNet. Na Figura 4 é possível observar o algoritmo que a OppNet utiliza para realizar o crescimento da rede.

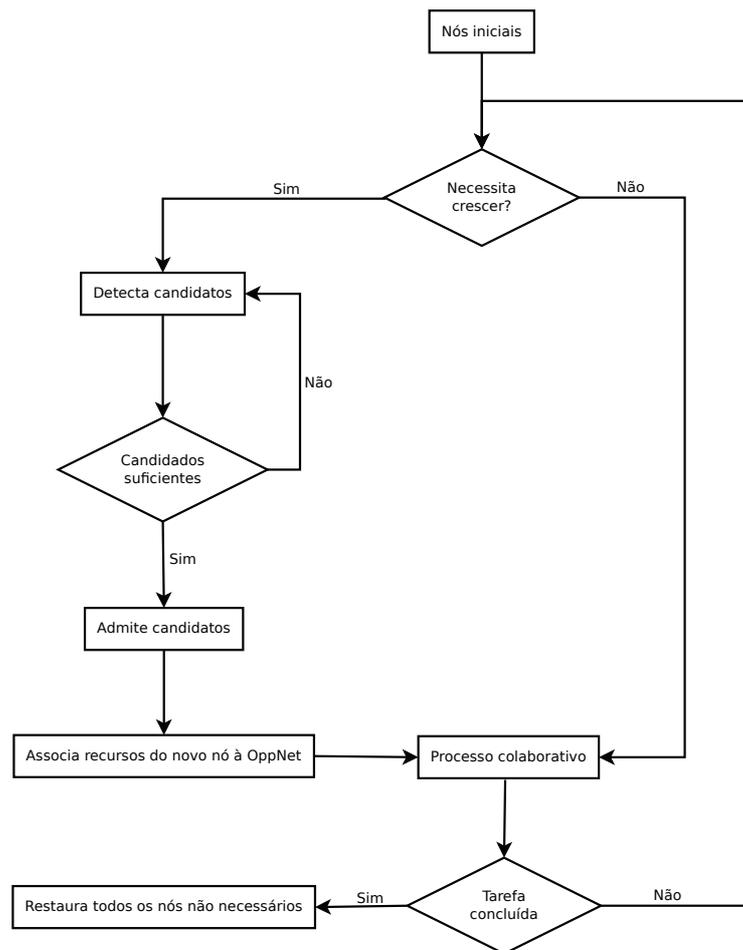


Figura 4 Algoritmo de crescimento de uma OppNet.

Os nós detectados são identificados e avaliados para verificar a sua utilidade na rede. Em seguida, os possíveis candidatos são convidados a participar podendo os mesmos aceitar ou rejeitar o convite. Caso algum nó

aceite o convite, seus recursos passam a fazer parte da rede podendo, assim, serem utilizados por qualquer outro nó da OppNet.

A partir do momento em que um nó deixa de ser necessário à OppNet, o mesmo deve ser liberado e restaurado ao estado em que se encontrava antes de ser inserido na rede. Esta característica minimiza o crescimento desordenado da OppNet e revela o caráter oportunista da mesma, já que apenas utiliza os recursos de um dispositivo enquanto ele pode ser útil, caso contrário o mesmo é liberado.

Os dispositivos *wireless* da OppNet fornecem acesso a estações de comunicações distintas, como estações fixas para acesso a internet, o que lhe proporciona adquirir e disseminar informações e efetuar uma melhor análise de dados, já que possui acesso a um poder de processamento e armazenamento maior. Por causa dessa característica as OppNets podem estabelecer pontes entre diferentes recursos e serviços disponíveis (LILIEEN et al., 2007).

2.1.3 Diferenças entre MANETs e OppNets

As OppNets utilizam a vantagem do movimento de seus nós para realizar suas conexões, o que a aproxima muito das redes MANETs. Porém, as MANETs são orientadas à camada de rede enquanto as OppNets estão orientadas à aplicação. Uma importante diferença entre ambos tipos de redes consiste no roteamento. As MANETs focam em algoritmos de roteamentos eficientes como o modo proativo e reativo proposto por Perkins e Bhagwat (1994), além disso, alguns algoritmos podem utilizar informações de posição geográfica para aumentar a eficiência do algoritmo. Em uma MANET todos os nós se relacionam entre si com o propósito de resolver um objetivo comum na rede, cada nó da rede tem sua parte de responsabilidade

para resolver um determinado problema.

Em redes MANETs os protocolos de roteamento tendem a manter rotas fim-a-fim entre todos os nós da rede conforme a topologia evolui. Já as OppNets são formadas por grupos anônimos de dispositivos, o que altera consideravelmente o roteamento de informações, já que novos nós podem aderir ou serem removidos da rede. Isso impossibilita manter rotas fim-a-fim para realizar o roteamento.

2.1.4 Diferenças entre Redes *Peer-to-Peer* e OppNets

Segundo Schollmeier (2001), as entidades de comunicações têm aumentado seu interesse nas Redes *Peer-to-Peer* (P2P). Elas possuem arquitetura distribuída onde os nós da rede compartilham seus próprios recursos de *hardware*. Isso é necessário para que os dispositivos possam se comunicar sem a necessidade de um intermediário. Outra característica das redes *Peer-to-Peer* é que elas são criadas como uma nova camada sobre uma rede existente. Esse tipo de rede é denominada rede *overlay*.

Assim como as Redes *Peer-to-Peer*, as OppNets possuem em seus nós as funcionalidades de cliente e servidor. Os nós de uma OppNet podem necessitar ou gerar dados para outros serviços da rede. Porém, a mobilidade dos nós não é levada em consideração nas redes P2P, pois o objetivo dessas redes é o compartilhamento de recursos. Já nas redes MANETs e OppNets, as conexões entre os diversos dispositivos são definidas automaticamente, ou seja, são imprevisíveis. Identificar e alocar recursos são serviços compartilhados entre dois nós o que significa um comportamento cooperativo entre ambos.

2.2 O simulador *The ONE* para Redes Oportunistas

O desenvolvimento de protocolos de roteamento para Redes Oportunistas dá-se, na maioria das vezes, por simulação. Esta técnica possibilita que o novo protocolo possa ser avaliado antes de ser testado em dispositivos reais, diminuindo o trabalho de programar e reprogramar os dispositivos em caso de falhas e novos testes.

Neste trabalho foi utilizado o simulador *The ONE (Opportunistic Network Environment)* (KERÄNEN; OTT; KÄRKKÄINEN, 2009) desenvolvido para avaliar protocolos de roteamento para Redes Tolerantes a Falhas. O simulador foi implementado em *Java* e é completamente configurável. Seu motor de simulação consegue simular todo o ambiente onde os nós da rede estão imersos, incluindo o movimento dos nós e conexões. O *The ONE* possibilita também a simulação de diversos protocolos de roteamento, permitindo a comparação dos protocolos implementados.

Uma característica do simulador *ONE* é seu auto nível de modularidade, como mostrado na Figura 5. Cada modelo de movimento e protocolo de roteamento são implementados em módulos independentes, que podem ser carregados dinamicamente desde que a configuração da simulação seja feita de forma correta. Esta característica facilita a implementação de novos protocolos para o simulador.

Ele permite também a criação de módulos de relatórios que facilitam a coleta dos resultados das simulações, salvando-os em arquivos de relatório.

O simulador possui implementado diversos algoritmos de roteamento bem conhecidos na área de Redes Tolerantes a Falhas, incluindo o PROPHET e o *Spray and Wait* o que justifica o seu uso neste trabalho. Além disso, ele possui uma interface gráfica que permite a visualização

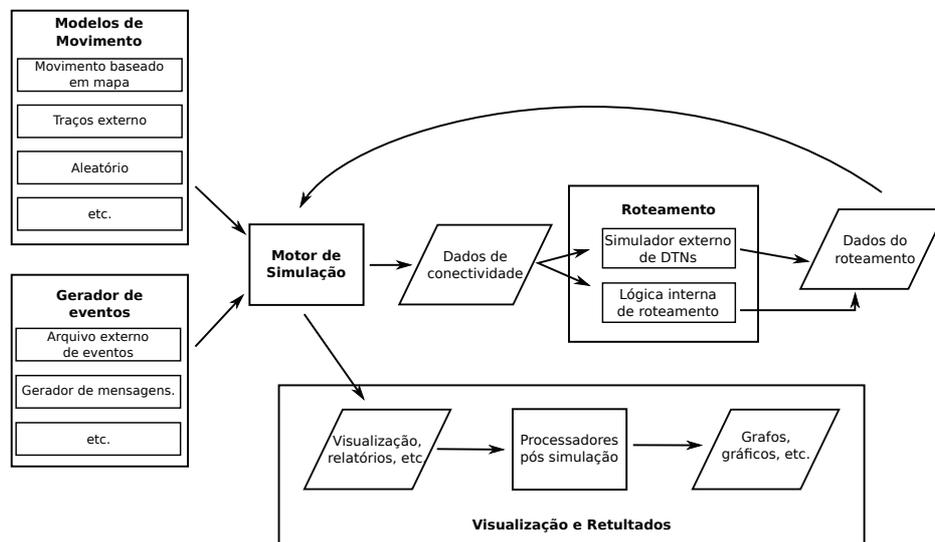


Figura 5 Visão geral do simulador *The ONE* (KERÄNEN; OTT; KÄRKKÄINEN, 2009).

da simulação em tempo real, permitindo que determinados dados da rede possam ser observados durante a simulação.

2.2.1 Configuração

As simulações são configuradas antes de serem executadas. Elas são configuradas por meio de arquivos textos que descrevem o ambiente de simulação, desde o número de nós até a quantidade de tempo que a simulação levará. Os arquivos contêm parâmetros sobre a interface gráfica, gerador de eventos e até sobre os módulos de relatórios. Todos esses parâmetros são lidos e configurados antes das simulações ajustando o ambiente para a execução.

Os arquivos de configuração são compostos por parâmetros que são

salvos em pares chave-valor como o seguinte:

$$\text{Namespace.chave} = \text{valor}$$

Os *namespaces* representam partes do ambiente de simulação sobre o qual a configuração tem efeito, geralmente são utilizados os nomes das classes como *namespaces*. Os valores de cada par podem ser numéricos (com ou sem ponto flutuante e com os sufixos “k”, “M” ou “G” para representar sua grandeza), *booleanos* (*true* ou *false*) ou texto. Também é possível realizar comentários de linha nos arquivos de configuração utilizando o símbolo “#”.

As configurações podem ser divididas em diversos arquivos o que permite a modularização da configuração de cada parte do ambiente como o cenário, nós e o protocolo de roteamento. O primeiro arquivo de configuração carregado será o “default_settings.txt”, caso ele exista.

Segundo Bujari, Palazzi e Bonaldo (2011), os agentes básicos do simulador são denominados nós que modelam um objeto móvel capaz de atuar como um roteador *store-and-forward* (isto é, um pedestre, um carro ou qualquer objeto com o *hardware* requerido). Os nós com características iguais podem ser divididos em grupos, onde cada grupo pode ser configurado independentemente. Em um mesmo grupo todos os nós compartilham as mesmas características como a interface de rádio, movimento, armazenamento e protocolo de roteamento. Com esse modelo de configuração é possível criar um ambiente de simulação heterogêneo com diferentes tipos de nós, agregando mais realidade e confiabilidade na simulação.

2.2.2 Saída das simulações

O simulador oferece uma interface gráfica para acompanhar a execução da simulação. Na interface, mostrada na Figura 6, é possível observar o movimento dos nós e detalhes como o caminho percorrido durante a simulação. Um painel com o *log* dos eventos também está presente na parte inferior da interface, possibilitando ver as ocorrências de trocas de mensagens na simulação. Os eventos podem ser filtrados pelo tipo reduzindo a quantidade de informações no painel.

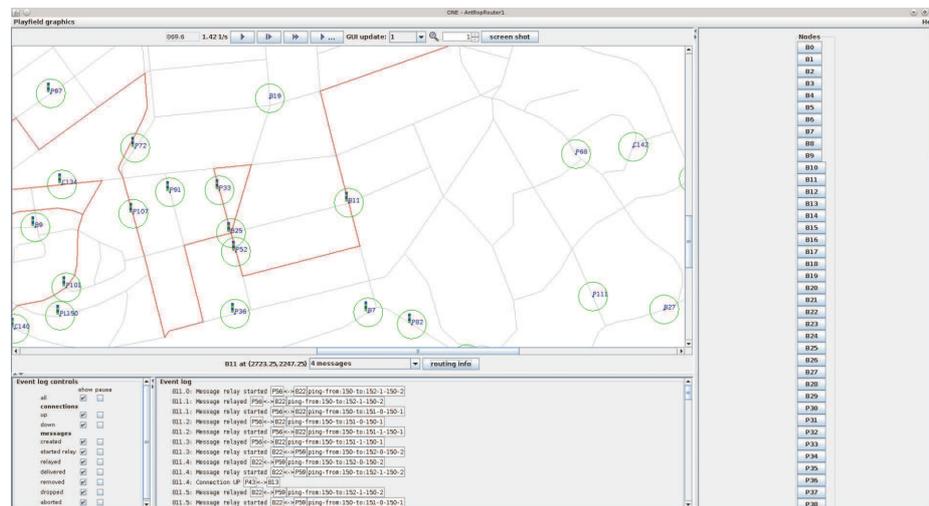


Figura 6 Interface gráfica do simulador *The ONE*.

Clicando sobre um nó é possível ver informações avançadas como as conexões ativas, mensagens que estão sendo recebidas e mensagens que foram entregues ao destinatário. Para cada mensagem entregue são mostrados os saltos que foram necessários para chegar ao destinatário. Esta janela *pop-up* pode ser vista na Figura 7.

As saídas das simulações também podem ser vistas nos arquivos



Figura 7 Janela *pop-up* com informações detalhadas *The ONE*.

de relatório. Os arquivos são gerados pelos módulos de relatório. Cada módulo é carregado dinamicamente no início, de acordo com os parâmetros presentes nos arquivos de configurações. Os relatórios são úteis quando as simulações são executadas sem a interface gráfica, permitindo que várias simulações possam ocorrer em sequência sem a necessidade de executá-las individualmente. Os parâmetros das simulações podem ser escritos como um arranjo nos arquivos de configuração, o que permite configurar cada parâmetro com valores diferentes para cada simulação.

O *The ONE* já possui alguns módulos de relatório prontos para uso, possibilitando a geração de relatórios relacionados a:

Mensagens: número de mensagens criadas, tempo de entrega, mensagens expiradas, etc...

Contatos: mostra os contatos entre os nós durante a simulação.

Conexões: mostra as mudanças no estado das conexões.

Com a modularização do simulador novos módulos de relatório po-

dem ser implementados permitindo que novos dados sejam coletados.

2.2.3 O mapa utilizado

O mapa define o espaço e rotas nos quais os nós podem se mover na simulação. A Figura 8 mostra o mapa distribuído juntamente com o simulador. Trata-se do mapa do centro da cidade de *Helsinki* na Finlândia. O mapa é separado em quatro partes e são descritas em arquivos no formato WKT (*Well Known Text*) que é utilizado para representar vetorialmente objetos em mapas.



Figura 8 Mapa da área central de *Helsinki* utilizado nas simulações (KERÄNEN; OTT; KÄRKKÄINEN, 2009).

2.3 Estado da Arte sobre Roteamento em Redes Oportunistas

Segundo Nguyen e Giordano (2009), nas redes oportunistas, dispositivos móveis transmitem mensagens explorando os contatos diretos, sem a necessidade de uma estrutura fim-a-fim como nas redes tradicionais. As desconexões e a alta taxa de rotatividade dos nós são características normais em redes oportunistas. Por isso, o roteamento é o principal desafio neste ambiente.

A eliminação da necessidade de construir caminhos simplifica muito o roteamento em redes oportunistas. Contudo, os desafios que surgem desta simplificação são diferentes dos que ocorrem em métodos tradicionais de roteamento em redes convencionais. Um esquema de roteamento em Opp-Nets deve prover os dados com alguma confiabilidade (o ideal seria prover dados com total confiabilidade) mesmo quando a conectividade da rede é intermitente ou quando um caminho fim-a-fim está temporariamente não disponível.

Em tais ambientes, protocolos de roteamento baseados em inundação (*flooding*) dominaram por algum tempo os estudos de novos métodos. Porém, essa abordagem tende ter altíssimo custo de consumo de energia e *overhead* de tráfego. Um fator que contribui para isso é que o *buffer* dos dispositivos são limitados. Assim, utilizar um método que utilize a técnica de inundação causaria uma alta taxa de perda de dados.

O desempenho do roteamento é melhorado quando informações sobre a topologia da rede, ambiente e dos participantes são explorados, isto é, as informações de contexto da rede.

Nesta seção será apresentada a taxinomia dos métodos propostos na literatura assim como uma revisão bibliográfica desses métodos.

2.3.1 Taxinomia dos métodos de roteamento

Nesse trabalho será utilizada uma taxonomia dos métodos de roteamento mais simplificada utilizada por Nguyen e Giordano (2009). Essa taxonomia separa os métodos em classes descritas a seguir.

Roteamento com contexto óbvio: são protocolos de roteamento que utilizam técnicas de *flooding* para efetuar o encaminhamento das mensagens.

Roteamento baseado na mobilidade dos nós: as informações sobre os movimentos dos nós são utilizadas para determinar por qual nó serão encaminhadas as mensagens.

Roteamento baseado nas informações sociais: os contextos sociais dos portadores dos dispositivos são recuperados e utilizados para determinar por qual nó as mensagens serão encaminhadas.

Nas seções a seguir serão discutidos os protocolos recentes encontrados na literatura. Para mais detalhes sobre esses protocolos o material apresentado por Cao e Sun (2012) pode ser consultado.

2.3.2 Roteamento com contexto óbvio

Métodos de roteamento baseados em *flooding* (inundação da rede por meio da duplicação de mensagens) pertencem ao grupo de algoritmos denominado contexto óbvio. Esse grupo utiliza técnicas para controlar o *flooding* da solução. Utilizando as técnicas de *flooding*, um nó envia uma mensagem para todos os outros nós vizinhos a ele. Cada nó que recebe a mensagem, repassa a mesma mensagem para todos os seus vizinhos, exceto

para aquele que lhe enviou a mensagem. Cada mensagem repassada pode chegar ao seu destino por rotas e tempos diferentes. A vantagem desse método é a simplicidade para encontrar uma rota, não é necessária nenhuma informação global sobre a topologia da rede ou qualquer informação de contexto. Porém, essa estratégia pode resultar em um congestionamento na rede pela quantidade de mensagens disseminadas. Além disso, o consumo de energia é alto, pois as mensagens são transmitidas a todos os nós da rede.

O desempenho da rede é particularmente importante em OppNets por causa das limitações dos dispositivos. Uma técnica utilizada para melhorar o desempenho do *flooding* é utilizar um contador para limitar o número de mensagens disseminadas na rede.

Um exemplo dessa abordagem é apresentada por Vahdat e Becker (2000). Esta abordagem utiliza um campo *Time-To-Live* (TTL) para cada pacote. O algoritmo de roteamento *Epidemic* proposto, modifica o *flooding* tradicional. Cada nó mantém uma lista de todas as mensagens que ele está portando no momento, as quais as entregas ainda não foram realizadas. Quando ele encontra algum outro nó, ambos trocam todas as mensagens que eles não têm em comum e, desta forma, todas as mensagens são espalhadas para todos os nós da rede. A mensagem é entregue quando o primeiro nó que porta uma cópia da mensagem encontra o destinatário. A mensagem continuará a ser copiada até que seu TTL expire.

Outra abordagem para rotear pacotes em redes esparsas é controlar a replicação utilizando a técnica de *spraying*. No algoritmo *Spray and Wait* (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2005), um número pequeno e fixo de cópias da mensagem é distribuído para um número de nós distintos (*spraying*). A partir desse momento, cada nó carrega a mensa-

gem até que o destino seja encontrado ou até que o TTL do pacote expire. Por ter múltiplos nós procurando pelo destinatário independentemente e em paralelo, esse protocolo cria uma diversidade suficiente para explorar a rede, reduzindo assim a quantidade de recursos utilizados para transportar os pacotes.

Utilizar técnicas para controlar o *flooding* torna esse método mais eficiente, mas não remove completamente a necessidade de replicação das mensagens. Esse ponto faz com que a eficiência desse método seja prejudicada, pois aumenta consideravelmente o consumo de recursos e pode causar congestionamento no tráfego da rede. O método proposto neste trabalho não utilizará técnicas de *flooding* o que evitará um alto consumo de recursos da rede. As mensagens serão encaminhadas nó a nó até que o destinatário seja alcançado, evitando assim a replicação desnecessária das mensagens.

2.3.3 Roteamento baseado na mobilidade dos nós

Protocolos de roteamento que se baseiam na mobilidade dos nós exploram as informações de contexto para determinar qual é o melhor nó para o qual um pacote deve ser encaminhado. A mobilidade dos nós impacta na eficiência do roteamento em OppNets. Grossglauser e Tse (2002) provaram que a mobilidade incrementa o desempenho de redes *Ad Hoc*, especialmente no roteamento de mensagens quando técnicas eficientes são aplicadas.

O método *Probabilistic Routing* (PRoPHET) proposto por Lindgren, Doria e Schelén (2003) calcula a previsibilidade de entrega de uma mensagem de um nó para um destinatário particular observando o histórico de contatos, e encaminha a mensagem se e somente se aquele nó vizinho tem uma alta previsibilidade de entrega. O PRoPHET utiliza a “história dos

encontros e transições passadas” para estimar a probabilidade de entrega de um determinado nó para outro nó. A previsibilidade de entrega é a probabilidade de um nó encontrar um determinado destinatário. Ela aumenta quando o nó encontra frequentemente o destinatário e diminui caso contrário.

A informação de contexto utilizada pelo PRoPHET é a frequência de encontros entre os nós, como é também visto nos protocolos MV (*Meeting and Visits*) e MaxProp (BURNS; BROCK; LEVINE, 2005; BURGESS et al., 2006). Um nó utiliza o MaxProp para efetuar a transmissão de pacotes para seus pares e determinar quais pacotes deverão ser removidos quando seu *buffer* estiver quase cheio. Os pacotes são enviados para seus pares de acordo com os históricos de dados que permitem calcular a probabilidade de um par conter um caminho para o destinatário.

No protocolo MobySpace proposto por Leguay et al. (2006), os padrões de movimento dos nós são as informações de contexto que o nó fonte utiliza para efetuar o roteamento das mensagens. Ele confia na suposição que um nó é um bom candidato para repassar a mensagem se ele possuir um padrão de movimento similar ao padrão do destinatário. O roteamento é feito encaminhando as mensagens para nós que possuem um padrão cada vez mais similar ao padrão do destinatário. Para trabalhar com os padrões de movimento dos nós, um espaço Euclidiano virtual é usado, onde cada eixo representa um contato possível entre dois nós e a distância ao longo de um eixo mede a probabilidade em que o contato pode ocorrer.

Ghosh, Philip e Qiao (2005) propõem um protocolo baseado em uma infraestrutura pré-definida, como lugares em que os portadores de dispositivos frequentam constantemente. Eles são chamados de *hubs*. Esse método

tem a vantagem de utilizar os perfis de mobilidade do usuário para executar um roteamento baseado em *hub-level*. Com o protocolo *Solar-Hub*, a mensagem de um nó para outro é roteada para um ou mais *hubs* frequentemente visitados pelo destinatário, onde o destinatário poderá obter a mensagem quando visitar os mesmos *hubs* novamente. Nesse cenário, é necessário que o nó conheça sobre os lugares visitados pelo destinatário. A Figura 9 exemplifica um ambiente para o protocolo. Esse protocolo é similar ao desenvolvido neste trabalho, uma análise das diferenças entre eles será realizada posteriormente.

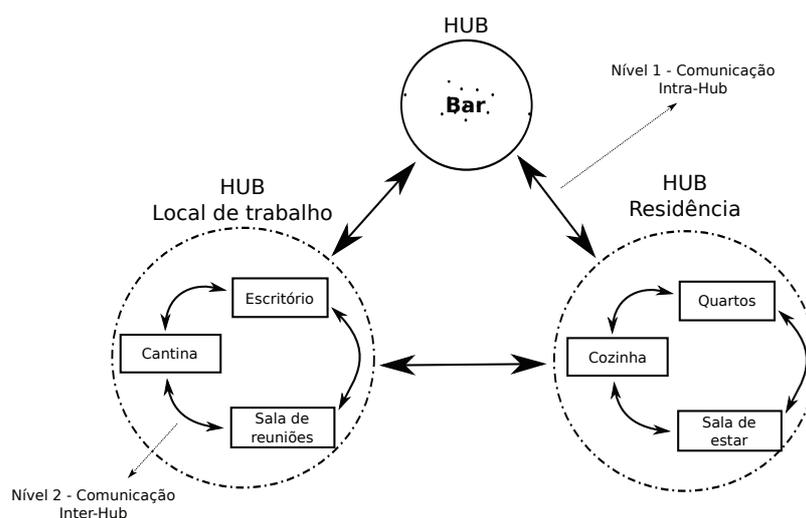


Figura 9 Exemplo de um ambiente par ao protocolo *Solar-Hub*. Fonte: (GHOSH; PHILIP; QIAO, 2005)

Diferentemente do *Spray and Wait* (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2005), na fase *Focus* do protocolo *Spray and Focus* (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2007), além de aguardar para que o destinatário seja encontrado, cada nó pode encaminhar sua cópia da mensagem para um nó potencialmente mais apropriado, utilizando

para isso um método denominado *utility-based*. Os potenciais nós são selecionados sobre um conjunto de relógios que armazenam o intervalo de tempo desde que dois nós se encontraram pela última vez. Utilizar o último encontro como critério para efetuar o encaminhamento de mensagens também está presente no método proposto por Grossglauser e Vetterli (2003). Nesse método, a mobilidade é explorada para disseminar informações sobre a localização do destinatário. Cada nó mantém um banco de dados com os tempos e locais de seus últimos encontros. Esse banco de dados é consultado para obter estimativas da localização dos destinatários das mensagens. Como o pacote é transportado pela rede, é possível refinar a busca sucessivamente para obter uma melhor estimativa da localização.

As informações de contexto utilizadas no protocolo *Bubble Rap* (HUI; CROWCROFT; YONEKI, 2011) são as comunidades sociais as quais os nós pertencem. As comunidades são automaticamente definidas e rotuladas baseando-se nos padrões de contatos entre os nós. Quando um nó deseja enviar um mensagem a outro nó ele procura por nós que pertencem a mesma comunidade do destinatário. Caso tais nós não sejam encontrados, a mensagem é encaminhada para incrementar os nós sociáveis, os quais têm maiores chances de entrar em contato com a comunidade do destinatário. O nível de socialização de um nó é definido por um conjunto de pares que o nó entra usualmente em contato.

CAR (*Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks*) proposto por Musolesi, Hailes e Mascolo (2005) utiliza filtros de Kalman para combinar e avaliar as múltiplas dimensões do contexto para efetuar decisões sobre o roteamento. O contexto é composto por mensurações realizadas frequentemente pelos nós, as quais podem estar relacionadas

com a conectividade, mas não necessariamente. As conexões acontecem dentro de redes MANETs (Mobile Ad Hoc Networks), para alcançar nós fora de uma rede MANET um nó interessado em enviar uma mensagem irá procurar pelo nó com maior probabilidade de sucesso para entregar a mensagem ao destinatário, ou seja, o nó que tem um padrão de movimento no qual o nó destinatário é frequentemente encontrado. Então o nó escolhido armazena temporariamente a mensagem, aguardando entrar em contato com o destinatário ou entrar em uma nuvem com outros nós com maior probabilidade de encontrar o destinatário. Além disso, os nós calculam a probabilidade de entrega proativamente, e as disseminam em suas próprias nuvens.

A utilização dos padrões de movimento para determinar os próximos saltos de uma mensagem torna-se importante, já que em redes oportunistas a movimentação dos nós é inevitável. Os métodos apresentados nessa seção utilizam esses padrões para efetuar a seleção do próximo nó a ser utilizado para o repasse da mensagem. Dentre esses métodos dois destacam-se: *Solar-Hub* proposto por Ghosh, Philip e Qiao (2005) e *Bubble Rap* proposto por Hui, Crowcroft e Yoneki (2011). Esses métodos possuem uma similaridade que é o estabelecimento de pontos para efetuar o encaminhamento das mensagens. No *Solar-Hub* não são utilizados *hubs* pré-definidos enquanto no *Bubble Rap* esses pontos são definidos automaticamente, mas possui a desvantagem de serem fixos.

O método proposto neste trabalho utilizará a junção do *Solar-Hub* e do *Bubble Rap* que permite que *hubs* sejam estabelecidos automaticamente para troca de informações de contexto entre os nós. Assim, não é necessário que os *hubs* sejam préfixados como é feito no protocolo *Solar-Hub*. As comunidades sociais apresentadas pelo protocolo *Bubble Rap* serão estabele-

cidas no momento em que um novo *hub* é criado e desfeitas quando o *hub* é destruído. Essa técnica permite que o protocolo se adapte ao ambiente caso haja alterações no comportamento da rede, como o surgimento de outro *hub*.

2.3.4 Roteamento baseado nas informações sociais

A principal diferença entre os protocolos baseados em informações de contexto social e os baseados na mobilidade dos nós é que esse último utiliza os padrões de movimento dos nós, as informações sobre os dispositivos ou o histórico de contatos entre os nós. Protocolos que utilizam o contexto social não exploram somente as informações sobre a mobilidade dos nós, mas também os aspectos sociais nos quais os nós estão envolvidos.

Assim, o relacionamento social dos portadores dos dispositivos tem um papel importante na forma em que os nós podem encontrar-se. A vantagem dessa abordagem é que ela é mais geral do que as abordagens que utilizam as informações sobre mobilidade. Esses protocolos de roteamento podem ser utilizados com qualquer conjunto de informações de contexto, e assim eles podem facilmente serem personalizados para oferecer um serviço melhor para determinado ambiente.

O protocolo *HiBOp* proposto por Boldrini et al. (2007) utiliza um conjunto de informações que descreve o perfil do nó e o histórico dos relacionamentos sociais com outros nós. Em cada nó, as informações utilizadas para construir o contexto podem ser pessoais como o nome e endereço do usuário. Os nós compartilham seus próprios dados durante os contatos e assim aprendem o contexto em que estão interessados. O protocolo assume que cada nó armazena localmente uma tabela denominada *Identity Table*

(IT) que contém as informações pessoais do usuário portador do dispositivo. As ITs são trocadas entre os nós quando eles entram em contato. Assim, cada nó possui sua própria tabela e um conjunto de tabelas de seus vizinhos que representa o contexto atual que proverá as informações para efetuar o roteamento.

A segunda informação de contexto utilizada pelo *HiBOP* é o histórico de contatos. Mesmo se um nó não é um bom candidato para encaminhar mensagens por causa de seu contexto atual, ele ainda pode ser um bom candidato por causa de seus hábitos e experiências passadas. Baseando-se na suposição de que os humanos são, em sua maior parte do tempo, “previsíveis”, é importante coletar informações históricas sobre cada nó e a recorrência das mesmas no contexto atual. Cada atributo do contexto atual é armazenado na *History Table*, em conjunto com a probabilidade de encontrar o mesmo atributo no futuro.

A principal ideia do *HiBOP* para encaminhar mensagens é procurar por nós que mostram uma maior similaridade com os atributos do contexto do destinatário. Uma alta similaridade entre o contexto do nó e do destinatário significa uma alta probabilidade que o nó tem para carregar a mensagem até a comunidade do destinatário. Além disso, um nó que deseja enviar um mensagem utilizando o *HiBOP* especifica qualquer conjunto da IT do destinatário no cabeçalho da mensagem. Qualquer nó entre o nó fonte e o destino é questionado sobre a sua similaridade dos atributos, e encaminha a mensagem se algum nó possuir uma similaridade maior que a do portador da mensagem.

No protocolo *Propicman* proposto por Nguyen, Giordano e Puiatti (2007), a informação de contexto de cada nó é representada por um conjunto

de pares ordenados chamado de *node profile*, que contém pares evidência - valor. Para cada evidência é associado um peso que representa a sua importância na rede. Por razões de segurança, cada *node profile* contém o *hash* de cada par ordenado de evidência e valores.

Quando um nó S deseja enviar uma mensagem M para um nó D , ele enviará para seus vizinhos o cabeçalho da mensagem denominado h_M . Esse cabeçalho contém informações sobre o nó destinatário, as quais foram informadas pelo nó que enviou a mensagem. h_M é encaminhado aos nós vizinhos que estão a dois saltos de distância do nó S . Baseado nessa informação e em seus *node profiles*, os nós vizinhos calculam sua probabilidade de entrega (DP). S então enviará a mensagem M somente ao vizinho que está a dois saltos de distância com um valor de DP maior que o valor do portador da mensagem. Após enviar o conteúdo da mensagem, o nó S mantém uma cópia da mensagem para o próximo eventual encontro.

O protocolo *SpatioTempo* (NGUYEN; GIORDANO, 2008) é derivado do protocolo *Propicman* e também utiliza informações de contexto como os *node profiles* para calcular o DP de nós para destinatários. O encaminhamento de mensagens é efetuado como no *Propicman*: o nó que deseja enviar uma mensagem primeiramente envia o cabeçalho da mensagem que contém algumas informações sobre o destinatário. Os vizinhos que estão a dois saltos de distância calculam seus próprios DPs baseados em sua similaridade com as informações de contexto do destinatário e retorna os resultados obtidos para o nó portador da mensagem. O nó portador irá selecionar o vizinho a dois saltos de distância que possui um DP maior que o seu próprio para efetuar o encaminhamento da mensagem.

O método *SpatioTempo* é o primeiro a utilizar informações de con-

texto temporais. Nesse método, as atividades humanas são divididas em duas categorias principais: comportamentos periódicos e não periódicos. Os comportamentos periódicos incluem atividades que ocorrem frequentemente, são atividades rotineiras de um indivíduo, por exemplo, ir à escola toda manhã. Os comportamentos não periódicos são todas as outras atividades que acontecem raramente, por exemplo, trabalhar em um feriado. Além disso, os autores consideram profundamente a relação social entre os usuários. Eles observam o frequente contato entre pessoas e outros contatos ocasionais.

Com essas informações o protocolo *SpatioTempo* pode classificar os destinos em dois tipos: destinos frequentes e destinos ocasionais. Para destinos ocasionais, os autores propõem o uso do protocolo *Propicman*, já que seu esquema é eficiente em casos onde não há suporte para informações sobre quando e para onde enviar a mensagem para um tipo específico de destino. Para destinos frequentes, o protocolo *SpatioTempo* introduz os conceitos de ciclos e períodos das atividades que o portador do dispositivo pode realizar. Em certos períodos, os portadores podem encontrar indivíduos específicos, por exemplo, colegas de classe, em outro período ele raramente encontra outros indivíduos. Essa regularidade pode ajudar os dispositivos portadores de mensagens a encontrar a melhor forma para enviar uma mensagem para um destino específico.

O protocolo desenvolvido utiliza o histórico de contato, armazenando a posição geográfica, tempo e identificador do nó para realizar o roteamento. Esse método assemelha-se ao método utilizado pelo protocolo *HiBOp*, porém as tabelas geradas são transferidas apenas para nós que estão em regiões geográficas denominadas *hubs*. Cada nó mantém uma tabela que é proces-

sada periodicamente excluindo as informações antigas. Diferentemente dos protocolos *Propicman* e *SpatioTempo*, as informações coletadas não possuem apenas um par evidência - valor. Também são considerados o tempo em que a informação foi coletada e também sua recorrência. As tabelas com as informações são trocadas quando um nó encontra-se com um *hub* que é responsável por gerar uma tabela de roteamento vinculada àquele *hub*.

O roteamento é efetuado somente após todos os *hubs* da rede possuírem em sua tabela de roteamento uma entrada para o destinatário de uma mensagem. Baseando-se no padrão de movimento sociológico é possível realizar rotas entre *hubs* para realizar o encaminhamento de mensagens. O roteamento poderá ser realizado *hub a hub* até que a mensagem seja entregue ao destinatário. Cada entrada das tabelas de roteamento presentes nos *hubs* possuem uma coluna que representa a qualidade de uma rota (*fitness*), esse valor é atualizado periodicamente favorecendo as rotas que são constantemente utilizadas e torna inutilizável as demais.

Para uma taxonomia com mais detalhes o trabalho de Cao e Sun (2012) poderá ser consultado. A Figura 10 representa a árvore de classificação dos protocolos, porém essa classificação é muito extensa e detalhada e por isso não foi utilizada neste trabalho.

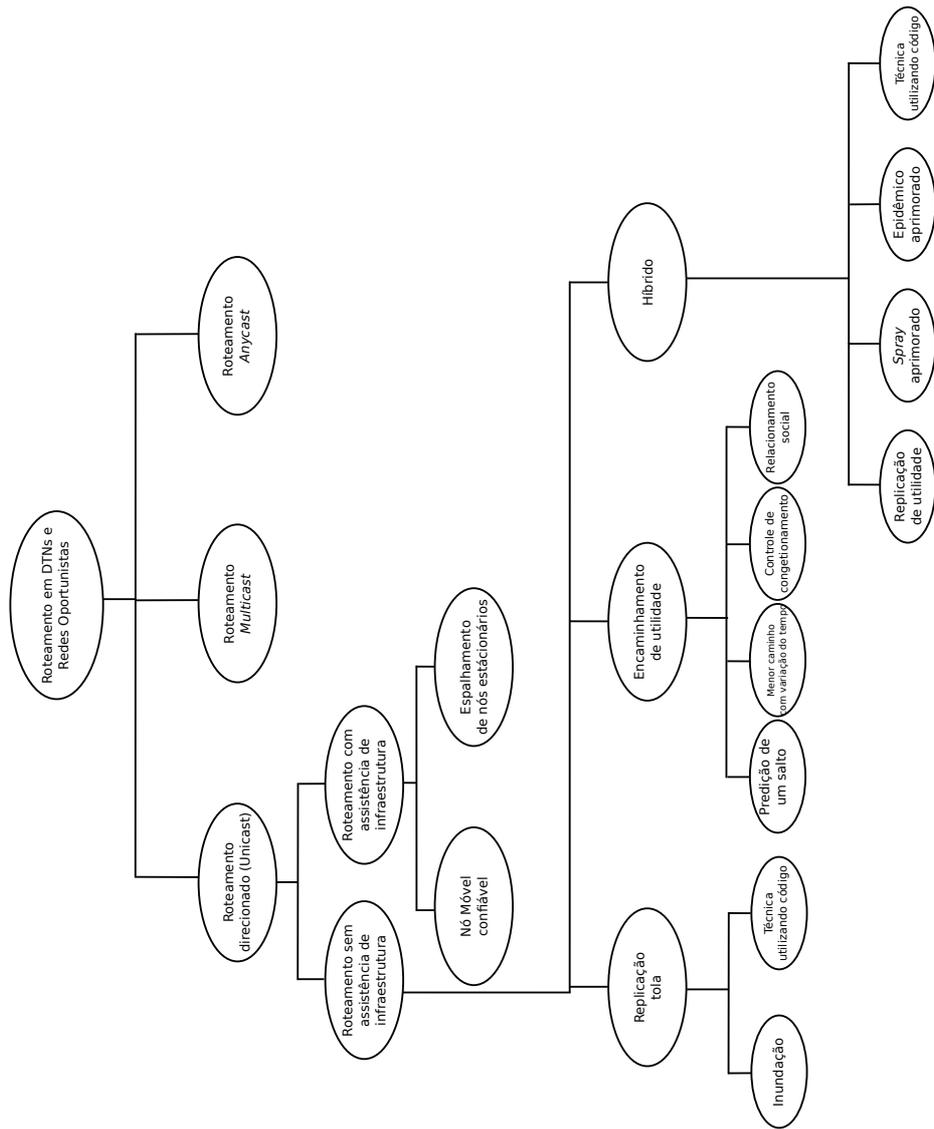


Figura 10 Taxonomia proposto por Cao e Sun (2012),

3 METODOLOGIA

Para efetuar o roteamento em uma OppNet foi implementado um protocolo que utiliza as características da rede para realizar a troca de mensagens. O protocolo utiliza a abordagem *store-carry-forward* (NGUYEN; GIORDANO, 2009) para efetuar a entrega de mensagens na rede. Além disso, as informações de contexto como histórico de encontros e movimentação dos nós são utilizadas para formar a rota correta ao destinatário.

O método apresenta diferenças significativas sobre os métodos presentes na literatura. A principal diferença e mais importante é o estabelecimento de pontos de acessos automaticamente. Um ponto de acesso é alguma região geográfica onde há uma grande concentração de nós que podem agregar-se e formar um *hub*. Com essa pequena rede local é possível realizar o compartilhamento de informações entre os nós da rede, permitindo ter acesso a informações essenciais de contexto para efetuar o roteamento de mensagens. Além dessas informações, os nós mantêm um histórico que conterá informações sobre localidades e contatos com outros nós que ocorreram recentemente. Essa informação é necessária para formar rotas, pois contém dados sobre os últimos *hubs* em que o nó esteve presente.

A troca de informações de contexto entre os nós é feita quando os mesmos encontram-se com *hubs*, que coletará as informações necessárias e formará tabelas de roteamento. Quando algum nó da rede necessita enviar alguma mensagem, ele a entregará a algum *hub* que efetuará o roteamento baseado em suas tabelas de roteamento.

Nesta seção será apresentada a visão geral do protocolo e detalhes de sua implementação.

3.1 Visão geral do protocolo *GrubiRouter*

O protocolo de roteamento desenvolvido e denominado *GrubiRouter* utiliza os padrões de movimento sociológico para efetuar o roteamento de mensagens. Em meio a esse modelo de movimento existem nós móveis movimentando-se periodicamente entre pontos fixos específicos (escritórios, casas, restaurantes, etc.). Utilizando estas informações é possível reconhecer as rotas e determinar quais nós realizam as mesmas rotas frequentemente e desta forma realizar o roteamento de mensagens. A Figura 11 mostra a visão geral do protocolo de roteamento desenvolvido, o qual será explicado com mais detalhes a seguir.

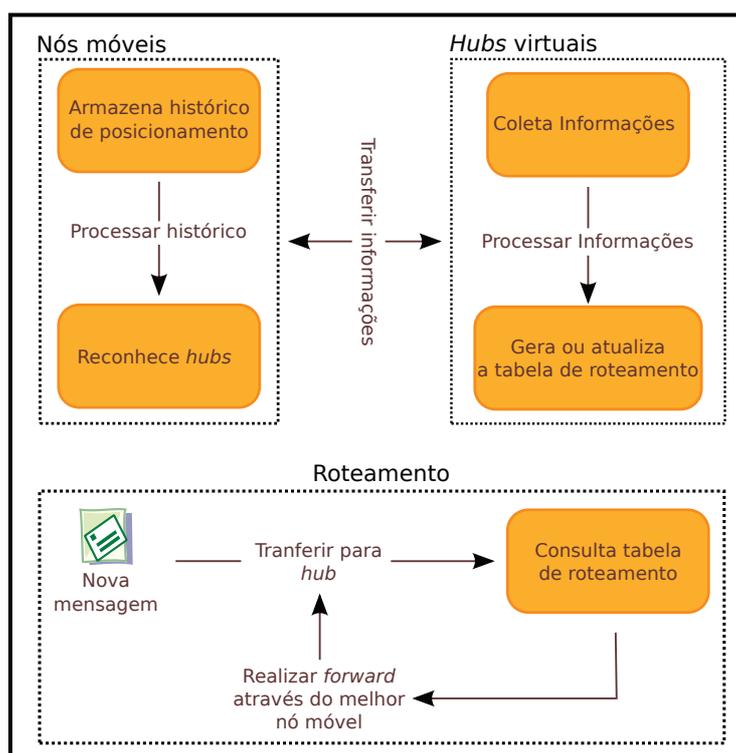


Figura 11 Visão geral do protocolo desenvolvido.

Com essas informações é possível fixar *hubs* cujo objetivo é servir como pontos de acessos aos demais dispositivos móveis para efetuar a troca de mensagens. Para compartilhar informações em um *hub* utilizou-se uma técnica denominada *Floating Content* (conteúdo flutuante, em tradução livre) descrita por Kangasharju, Ott e Karkulahti (2010) e Hyytia et al. (2012). Nessa técnica uma área denominada zona âncora é estabelecida em uma região de interesse. Todos os nós que entrarem nessa região devem compartilhar as informações de interesse, e ao saírem da região as mesmas podem ser descartadas. A Figura 12 mostra o funcionamento da técnica de *Floating Content*. Essa técnica foi utilizada para que seja possível compartilhar as informações em um ambiente dinâmico como nas Redes Oportunistas.

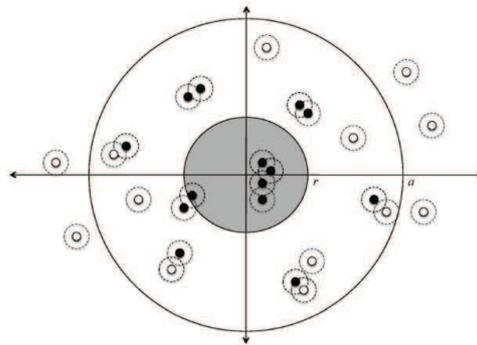


Figura 12 Zona âncora de um item, nós móveis e seus alcances de comunicação: o conteúdo é replicado entre os nós dentro da zona e deletados dos nós que estão a uma distância maior que a do centro da região. (KANGASHARJU; OTT; KARKULAHTI, 2010)

Para melhor exemplificação do protocolo considere a configuração de uma rede como representada na Figura 13. Nela é possível observar que existem nós móveis movimentando-se aleatoriamente no ambiente, alguns nós móveis que estão em contato com pontos fixos como escritórios e su-

permercados e alguns nós que movimentam-se repetidamente entre regiões sociais como um escritório, clube e universidade. Neste contexto vários outros nós vindos de outras direções podem se movimentar até os pontos fixos e novamente voltar ao seu local de origem. Esse padrão de movimento assemelha-se ao movimento rotineiro de pessoas e veículos no seu dia a dia, por isso é conhecido como um padrão de movimento sociológico (GHOSH; PHILIP; QIAO, 2005).

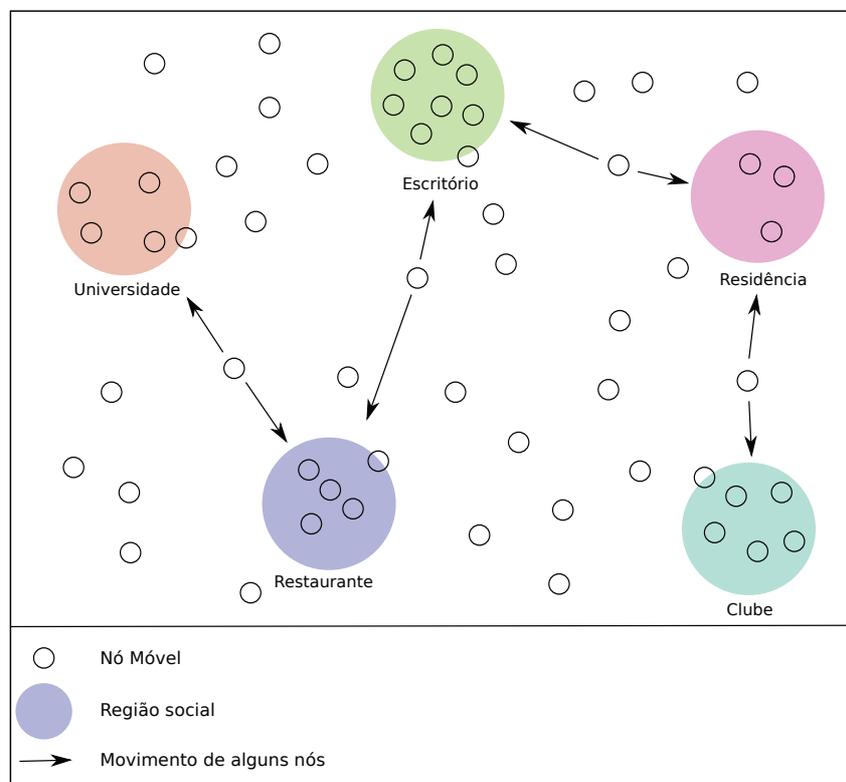


Figura 13 Cenário para exemplificação do protocolo.

O protocolo de roteamento é dividido em duas fases, reconhecimento e compartilhamento, que são responsáveis por gerar e processar a base de

dados utilizadas para criar e manter as tabelas de roteamento. Nas seções a seguir será detalhada cada uma das fases.

3.2 Fase 1 - Reconhecimento

Na fase 1 ocorre o reconhecimento dos nós móveis que realizam rotas periódicas entre dois ou mais pontos fixos. O reconhecimento é realizado armazenando o histórico de movimentação do nó e de encontros. Uma tabela é criada para armazenar uma métrica de confiabilidade denominada de *fitness* e a quantidade de visitas feitas a um determinado ponto fixo designado pela sua posição geográfica. O *fitness* é incrementado cada vez que um encontro ocorre dentro de um determinado intervalo de tempo e decrementado quando o mesmo não ocorre.

O Algoritmo 1 é responsável por incrementar o *fitness* da tabela a cada encontro que ocorre. É importante observar que a forma e os valores utilizados podem ser alterados a fim de melhorar o processo. Quando algum ponto fixo obtém um *fitness* superior a 0.9 ele é adicionado à lista de *hubs* com os quais aquele nó se encontra, como pode ser observado na linha 10.

Para realizar o envelhecimento da tabela de encontros e ajuste do *fitness* o Algoritmo 2 é executado periodicamente. Ele favorece as entradas da tabela que são atualizadas frequentemente. Caso alguma entrada chegue ao valor de *fitness* 0 o *hub* daquela entrada é retirado da lista de *hubs* como é possível observar na linha 10.

A Figura 14 demonstra parte do cenário apresentado na Figura 13. A partir do encontro com algum ponto fixo, o nó móvel registra as informações em uma tabela que será utilizada posteriormente.

As Tabelas 1 e 2 mostram exemplos das tabelas geradas pelo Algo-

```

1 início
2   se encontrou um ponto fixo então
3     Fitness fit = getFitness(ponto_fixo);
4     se fit.valor == 0 então
5       fit.valor = 0.1;
6     senão se fit.valor > 0 então
7       fit.valor = fit.valor + 0.2;
8     fit.contaEncontros++;
9     se fit.valor > 0.9 então
10      addListaHubs(ponto_fixo);

```

Algoritmo 1: Algoritmo responsável por gerar e atualizar a tabela de encontros realizando a formação dos *hubs* virtuais.

```

1 início
2   para cada PontoFixo pontoFixo : tabelaFitness faça
3     Fitness fit = getFitness(pontoFixo);
4     se (tempoAtual - fit.ultimaAtualizacao) >= limiar
5       então
6         double novoValor = fit.valor - 0.1;
7         se novoValor <= 0 então
8           novoValor = 0;
9         fit.valor = novoValor;
10        se estaListaHubs(pontoFixo) então
11          removeDaListaHubs(pontoFixo);

```

Algoritmo 2: Algoritmo responsável por decrementar o *fitness* da tabela de encontro após determinado limiar de tempo.

ritmo 1 para os nós N1 e N2 representados na Figura 14. É possível observar que o *fitness* possui um limite superior de valor 1 e um limite inferior de valor 0 determinados pelos algoritmos. Quando um *fitness* alcança o valor 1 os incrementos não terão efeito sobre o mesmo, dessa forma o *fitness* permanece em seu limite superior, podendo ser decrementado com o tempo. O mesmo acontece quando o *fitness* alcança o limite inferior, os decrementos

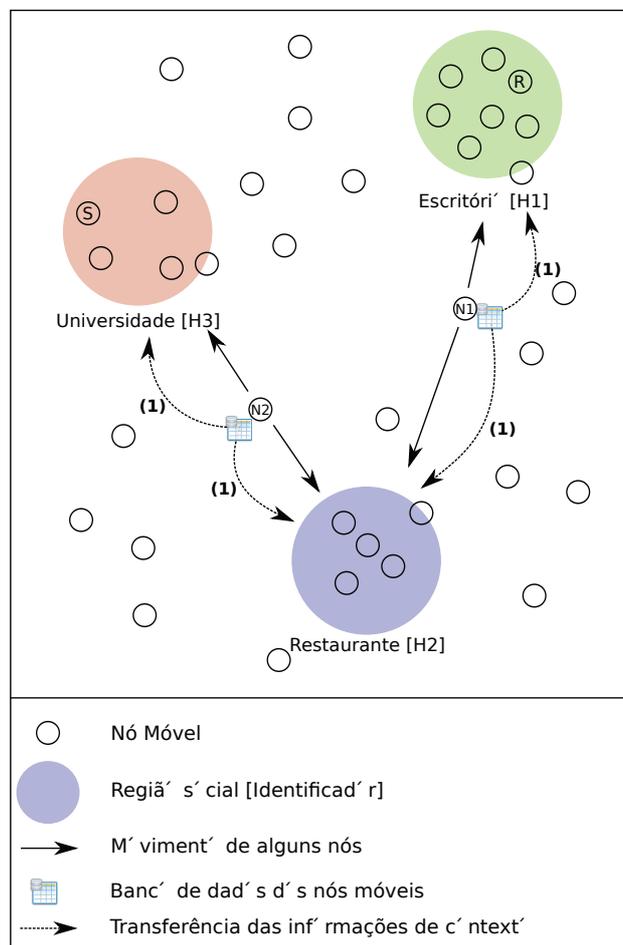


Figura 14 Exemplo da coleta das informações de contexto pelos nós N1 e N2 e transferência das mesmas para os *hubs* com os quais os nós mantêm contato.

não terão efeito sobre o valor do *fitness* que permanecerá em 0.

Tabela 1 Exemplo de tabela gerada após o processamento das informações de contexto para nó móvel N1.

Hub Virtual	<i>Fitness</i>	Qtd. encontros	Última atualização (segundos)
H1	0.900	7	15413
H2	1.00	8	13228

Tabela 2 Exemplo de tabela gerada após o processamento das informações de contexto para o nó móvel N2.

<i>Hub</i> Virtual	<i>Fitness</i>	Qtd. encontros	Última atualização (segundos)
H2	0.300	2	15413
H3	1.00	8	13228

Na Tabela 1 é possível observar que o nó N1 encontra-se periodicamente com os pontos H1 e H2 que são, respectivamente, o escritório e o restaurante. Desta forma, caso o movimento continue repentinamente, o *fitness* dos dois pontos fixos permaneceram em 1 representando que o encontro com esses pontos é frequente. Caso o movimento pare o *fitness* será decrementado a cada intervalo de tempo t até que atinja o valor 0.

Já a Tabela 2 apresenta os encontros do nó N2. Pela análise da tabela é possível determinar, baseado no valor do *fitness*, que o nó N2 possui uma relação fraca com o ponto fixo H2. Isso deve-se ao fato de que o nó realiza movimentos aleatórios e entra em contato com maior frequência com o ponto fixo H3. Caso o nó continue a encontrar o ponto fixo H2 seu *fitness* subirá indicando que o encontro entre N2 e H2 é frequente. Com o *fitness* é possível determinar quais nós são mais aptos a servirem de transporte para a troca de mensagens entre os pontos fixos.

O método descrito possibilita que os nós móveis que frequentemente encontram um ponto fixo sejam reconhecidos e utilizados, posteriormente, para o transporte de mensagens servindo como ponte entre os pontos fixos. A métrica utilizada para reconhecer uma rota permite que a rede se adapte com o tempo e favoreça os nós que efetuam maior número de encontro com determinado ponto fixo e desfavorecendo aqueles com menor quantidade de encontros ou os que encontram algum ponto fixo esporadicamente.

Nessa fase os pontos fixos tornam-se *hubs* responsáveis por efetuar a

troca de mensagens entre os nós móveis que se conectam a outros *hubs* e são também responsáveis por registrar quais nós móveis serão utilizados como transporte. As informações de contexto dos nós móveis são transferidas periodicamente para os *hubs* virtuais com os quais o nó móvel tem contato. Esse processo está exemplificado na Figura 14 indicado pela marcação (1).

3.3 Fase 2 - Compartilhamento

Utilizando a tabela de encontros registrados pelos nós móveis os *hubs* obtêm o conhecimento de quais outros *hubs* um determinado nó móvel entra em contato periodicamente. Dessa forma é possível criar uma tabela de roteamento onde cada entrada contém o *hub* destinatário e o nó móvel utilizado para alcançá-lo.

O Algoritmo 3 gera a tabela de roteamento a partir de uma tabela de encontros transmitida por um nó móvel, denominada no algoritmo como TABELAENCONTROS. Cada entrada da tabela de encontros, denominada de INFO, é processada podendo ser inserida na tabela de roteamento como uma nova informação de roteamento, denominada ROUTERINFO, ou utilizada para atualizar uma entrada existente. A tabela de roteamento contém para cada *hub* destinatário uma lista com os nós móveis que podem alcançá-los. Além disso informações sobre o número de saltos e de encontros são importantes para determinar a rota posteriormente.

As Tabelas 3, 4 e 5 exemplificam as tabelas de roteamento inicialmente geradas a partir do Algoritmo 3 em cada um dos *hubs* mostrados na Figura 14.

Observando as Tabelas 3, 4 e 5 é possível verificar que a primeira coluna da tabela de título “Para” contém o *hub* destinatário daquela entrada

```

1 início
2   Node passandoPor = tabelaEncontro.de();
3   para cada Info info : tabelaEncontros faça
4     se info.node != this.node então
5       RouterInfo routerInfo =
6         criaOuRecuperaRouterInfo(info.node, passandoPor);
7       routerInfo.hops = 1;
8       routerInfo.time = tempoAtual();
9       routerInfo.fitness = info.numEncontros;
       addRouterTable(routerInfo);

```

Algoritmo 3: Algoritmo para criar ou atualizar entradas na tabela de roteamento dos *hubs*.

Tabela 3 Exemplo de tabela de roteamento gerada no *hub* H1 após o processamento das informações transferidas a partir do nó N1.

Para	Passando por	<i>Fitness</i>	Número de Saltos	Última atualização (segundos)
H2	N1	10	1	3789

Tabela 4 Exemplo de tabela de roteamento gerada no *hub* H2 após o processamento das informações transferidas a partir do nó N1 e N2.

Para	Passando por	<i>Fitness</i>	Número de Saltos	Última atualização (segundos)
H1	N1	10	1	3769
H3	N2	10	1	3729

Tabela 5 Exemplo de tabela de roteamento gerada no *hub* H3 após o processamento das informações transferidas a partir do nó N2.

Para	Passando por	<i>Fitness</i>	Número de Saltos	Última atualização (segundos)
H3	N2	10	1	3589

na tabela de roteamento. Na segunda coluna, intitulada “Passando por”, está o identificador do nó utilizado para alcançar o *hub* destinatário daquela entrada. O *fitness* utilizado nessa tabela representa a quantidade de vezes

que o nó usado para transporte encontrou com o *hub* destinatário. Dessa forma é possível selecionar qual nó é mais confiável para transportar uma mensagem e entregá-la ao *hub* desejado.

Porém para efetuar o roteamento é fundamental que cada *hub* saiba da existência dos outros *hubs* na rede e que também saibam como alcançar esses *hubs*. Por exemplo, na Figura 13 o *hub* H1 deve ter conhecimento de que é possível alcançar o *hub* H3 passando pelo *hub* H2 e vice-versa.

Por isso torna-se necessário que cada *hub* compartilhe sua tabela de roteamento com os demais a fim de que os outros possam utilizá-la para completar sua própria tabela de roteamento. Com isso, cada *hub* obtém o conhecimento de toda a rede com as informações transmitidas periodicamente.

A Figura 15 exemplifica o compartilhamento das tabelas de roteamento entre os *hubs* da rede. Em (1) o *hub* H1 compartilha sua tabela de roteamento com os *hubs* H1 e H3 utilizando para isso os nós móveis N1 e N2. Através do nó móvel N2 o *hub* H3 compartilha sua tabela de roteamento com o *hub* H2, como exemplificado em (2). O *hub* H1 realiza o mesmo processo, mas utiliza o nó móvel N1; esse processo está ilustrado em (3). O processo é repetido periodicamente, assim todos os *hubs* da rede conseguem ter suas tabelas de roteamento atualizadas e rotas formadas para os outros *hubs* da rede.

O Algoritmo 4 é executado em um *hub* toda vez que é recebida uma mensagem contendo a tabela de roteamento de outro *hub*. Cada entrada da tabela de roteamento recebida é processada, durante o qual é verificado se a entrada não possui como *hub* destinatário o mesmo *hub* que o está processando; essa instrução é vista na linha 3. Na linha 4 o algoritmo busca

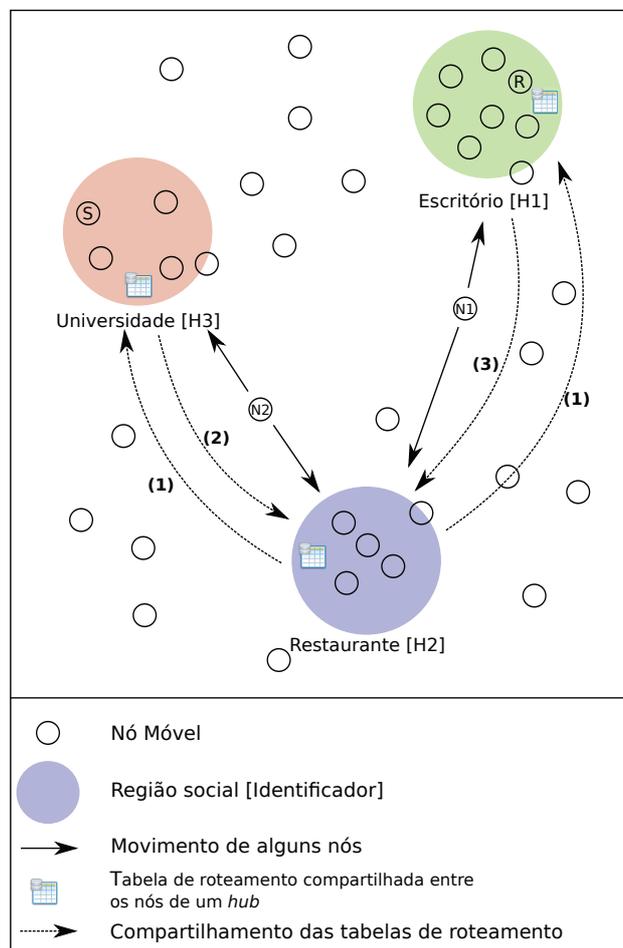


Figura 15 Exemplo do compartilhamento das tabelas de roteamento entre os *hubs* da rede.

o melhor nó móvel presente na sua tabela para alcançar o *hub* da entrada. O melhor nó móvel é aquele que possui maior *fitness* e menor número de saltos. Entre as linhas 5 e 9 a entrada é comparada com a melhor disponível, caso ela possua maior *fitness* a mesma é adicionada à tabela de roteamento e o número de saltos incrementado.

A Tabela 6 mostra a tabela de roteamento do *hub* H1 completamente

```

1 início
2   para cada RouterInfo routerInfo : tabelaRoteamento faça
3     se routerInfo.destinatario != this.node então
4       RouterInfo melhor =
5         recuperaMelhor(routerInfo.destinatario);
6       se (routerInfo.hops < melhor.hops) &&
7         (routerInfo.fitness > melhor.fitness) então
8         RouterInfo novaEntrada =
9           criaRouterInfo(routerInfo);
           novaEntrada.hops = novaEntrada.hops + 1;
           novaEntrada.ultimaAtualizacao = tempoAtual();
           addRouterInfo(novaEntrada);

```

Algoritmo 4: Algoritmo processar tabelas de roteamento recebidas de outros *hubs*.

atualizada após a execução do Algoritmo 4. Nela é possível observar que a partir do nó móvel N1 é possível alcançar os *hubs* H2 e H3, isso é possibilitado por consequência do contato que o H2 tem com o nó N2, que, por sua vez, conecta-se ao *hub* H3. Outra informação importante é quantidade de saltos necessários para alcançar o *hub* H3 que são 2, H1 para H2 através de N1 e H2 para H3 através de N2.

Tabela 6 Exemplo de tabela gerada após o processamento das informações de contexto coletadas pelos nós móveis.

Para	Passando por	<i>Fitness</i>	Número de Saltos	Última atualização (segundos)
H2	N1	10	1	7360
H3	N1	10	2	7830

Há também nos *hubs* um mecanismo que possibilita o envelhecimento das tabelas de roteamento. Caso as mesmas não sejam atualizadas em um determinado intervalo de tempo as mesmas são descartadas e retiradas da tabela. O Algoritmo 5 é executado periodicamente para manter a tabela de

roteamento atualizada.

```

1 início
2   para cada RouterInfo routerInfo : tabelaRoteamento faça
3     se (tempoAtual() - routerInfo.ultimaAtualizacao) >=
4       limiar então
5         removeDaTabelaRoteamento(routerInfo);

```

Algoritmo 5: Algoritmo para remover entradas antigas da tabela de roteamento.

É importante observar que o Algoritmo 4 favorece as entradas da tabela de roteamento compartilhadas que são melhores que as contidas em sua tabela de roteamento. Isso possibilita que sempre seja utilizada a que oferece melhor desempenho em relação às outras. Nesse caso são rotas que possuem maior *fitness* e menor número de saltos necessários para alcançar o *hub* destinatário.

Outro ponto importante é o tempo necessário para que os *hubs* tenham conhecimento de todas as rotas disponíveis, que é o tempo de propagação das tabelas de roteamento compartilhadas. Esse item será averiguado com mais detalhes na seção 4.

3.4 Controle do *buffer*

Para diminuir o *overhead* do protocolo foi implementado um método de controle do *buffer* de mensagens. As mensagens de controle dos nós arestas e *hubs* são filtrados para que sempre haja a versão mais atual da informação. Assim, as mensagens com informações mais antigas são descartadas permitindo que o *buffer* mantenha-se vazio.

Outro controle efetuado é a reordenação das mensagens do *buffer*. A reordenação é feita dando-se prioridades aos diferentes tipos de mensagens.

As mensagens de aplicações têm a prioridade (P_{app}) mais alta e ficam no início da fila, pois as mesmas devem ser transferidas com o menor tempo possível para diminuir a latência. Em seguida estão as mensagens com as informações para construção das rotas que são geradas pelos *hubs*. A prioridade (P_{hub}) desse tipo de mensagem é maior do que a prioridade (P_{art}) das mensagens geradas pelos nós arestas, pois são fundamentais para efetuar um roteamento eficiente através do *hubs*. A Inequação 1 mostra a ordem de prioridade das mensagens utilizada para a reordenação da fila.

$$P_{app} > P_{hub} > P_{art} \quad (1)$$

3.5 Roteamento

O processo de roteamento de mensagens é executado com facilidade após as tabelas de roteamento estarem completas. Cada *hub* mantém um histórico com informações de cada nó que se mantém em contato. Esta informação é também compartilhada entre os *hubs* para que se possa determinar para qual *hub* uma mensagem deve ser encaminhada para que chegue a seu destinatário.

Com essas informações disponíveis o processo de *forwarding* dar-se-á como em um algoritmo comum do tipo vetor de distância. Neste tipo de roteamento os *hubs* mantêm uma tabela que fornece a melhor distância conhecida até cada destinatário. O protocolo determina qual linha da tabela deverá ser utilizada para alcançar o destinatário eficientemente. As tabelas são atualizadas através da troca de informações com os nós *hubs* vizinhos. Esse processo é mostrado no fluxograma presente na Figura 16

O roteamento é efetuado *hub a hub*, sempre determinando qual é o

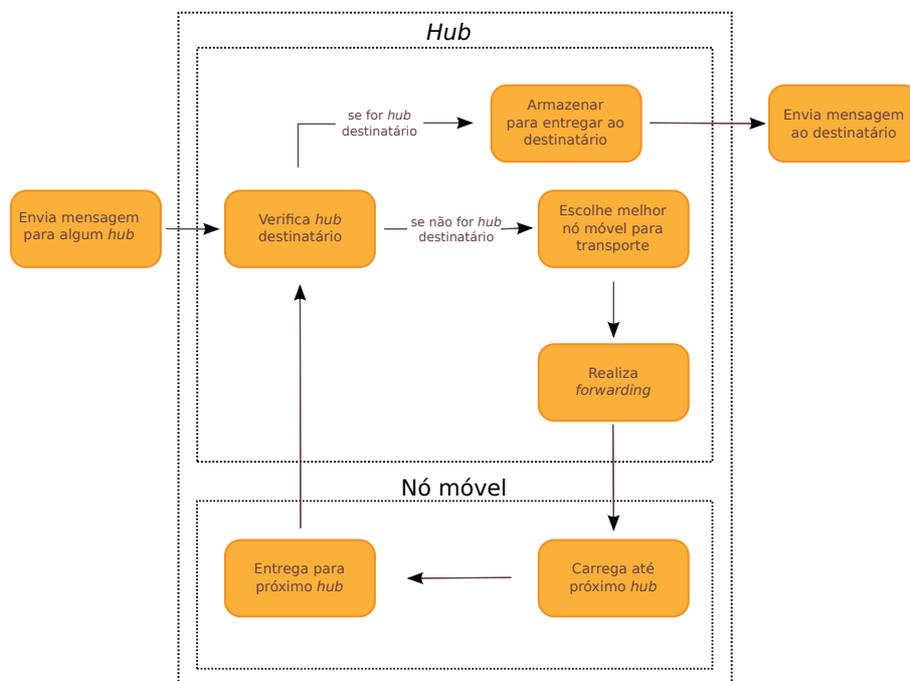


Figura 16 Fluxograma detalhando o funcionamento do processo de roteamento.

próximo salto para que a mensagem chegue ao *hub* frequentado pelo destinatário. Para determinar o próximo salto é considerada a entrada da tabela de roteamento que possui o maior *fitness* e o menor número de saltos necessários para que a mensagem chegue ao último *hub* da rota.

A Figura 17 ilustra o roteamento efetuado para uma mensagem transmitida do nó R, que encontra-se no escritório, para o nó destinatário S, que está na universidade.

Observando a Figura 17 verifica-se a ordem de execução a seguir. Primeiramente a tabela de roteamento do *hub* de origem é consultada para determinar o próximo salto necessário para alcançar o *hub* destinatário (1). Verifica-se que, através de N1, é possível alcançar o *hub* de destino com

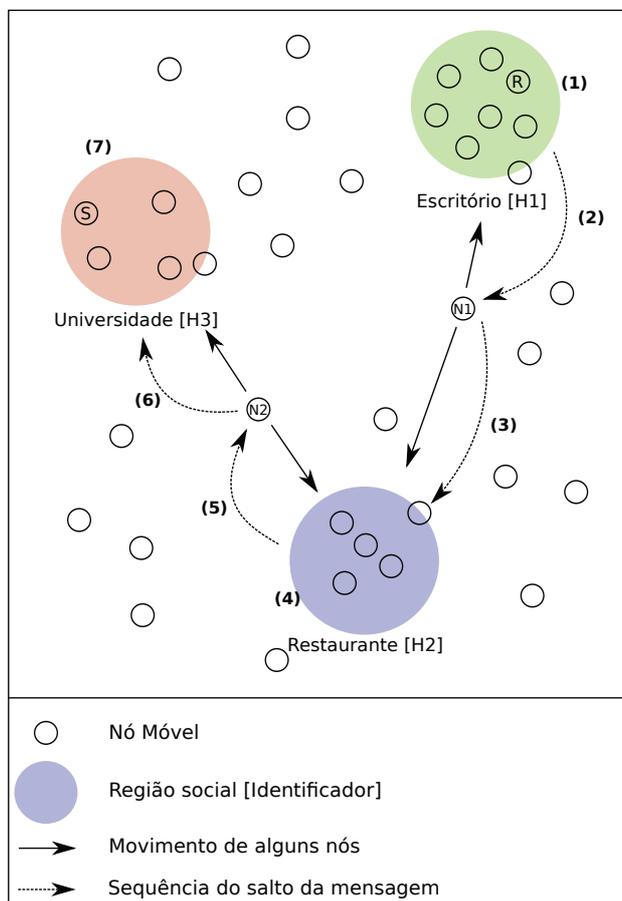


Figura 17 Exemplo do roteamento de mensagens.

apenas dois saltos. Com essa informação a mensagem é transferida para o nó N1 quando o mesmo entra em contato com o *hub* H1 (2). N1 ao encontrar com o H2 realiza a transferência da mesma (3). Novamente em um *hub* o processo é repetido obtendo a informação que através de N2 é possível alcançar o *hub* destinatário com apenas um salto (4). Assim a mensagem é transferida para o nó N2 (5) e posteriormente ao *hub* H3 (6). No *hub* final a mensagem é replicada entre os nós e permanecerá no mesmo

até que o nó S entre em contato com o *hub* e a mensagem seja entregue ao destinatário final.

O protocolo implementado permite que haja uma forma determinística para realizar o roteamento. Isso reduz a complexidade da movimentação dos nós da rede e utiliza essa característica para favorecer o roteamento. O método desenvolvido assemelha-se a uma rede *overlay* que é formada sobre a estrutura de uma outra rede. Neste trabalho utilizou-se as características e os serviços de uma OppNet para, sobre essa estrutura, criar uma rede virtual capaz de diminuir a complexidade do roteamento de mensagens em uma rede com topologia dinâmica.

4 RESULTADOS E DISCUSSÕES

Os resultados deste trabalho foram obras de análises feitas de dados retirados das simulações executadas no simulador *The ONE*. Os detalhes sobre os cenários utilizados, quantidade de simulações e significados dos resultados estão descritos nesta seção. Para a análise e discussão foram utilizadas as recomendações propostas por (GRASIC; LINDGREN, 2012) para avaliações de protocolos de roteamento para Redes Tolerante a Falhas.

4.1 Implementação do padrão de movimento

Implementar um padrão de movimento sociológico foi um requisito necessário para executar as simulações. O padrão desenvolvido permite simular um ambiente social. Devido a complexidade de desenvolver um protocolo de compartilhamento baseado em *Floating Content* juntamente com o protocolo de roteamento utilizou-se nós fixos para representar regiões sociais no simulador.

Foram implementados quatro tipos diferentes de nós para que o padrão funcionasse como desejado. Os tipos de nós estão descritos a seguir.

Hubs: nós que representam regiões sociais.

Nós aleatórios: realiza movimentos aleatórios, esses nós não respeitam o padrão sociológico, mas foram utilizados para inserir interferência na rede.

Nós arestas: realiza movimentos entre dois *hubs* e possuem uma probabilidade de sair da rota e realizar algum movimento aleatório. Após o movimento aleatório retornam a executar o rota normalmente.

Nós Visitantes: estes nós visitam um determinado *hub* inúmeras vezes, representam os nós que têm alguma relação com algum *hub*. Eles realizam movimentos aleatórios, mas retornam a algum *hub* com o qual possuem uma relação.

Com o padrão de movimento implementado um gerador de cenários também foi desenvolvido e será detalhado a seguir.

4.2 Gerador de cenários

O gerador desenvolvido é capaz de criar cenários aleatórios baseando em um arquivo de configuração que define os parâmetros para gerar cada cenário. Foi utilizado um *namespace* (NS) para representar as configurações de cada tipo de cenário que queremos gerar. O arquivo de configuração deve possuir os seguintes parâmetros para funcionar:

NumOfTypes: número de tipos diferentes de cenários a serem gerados.

NS.numOfScenarios: número de cenários de cada tipo a serem gerados.

NS.numOfFixedPoints: número de *hubs* de um determinado cenário.

NS.numOfRandomNodes: número de nós que realizam movimentos aleatórios.

NS.numOfVisitorsNodes: número de nós visitantes, visitam aleatoriamente algum *hub* da rede.

NS.endTime: tempo em que a simulação irá terminar. Para todas as simulações foi utilizado um tempo de 345600 segundos que foi suficiente para obter os resultados.

Na Figura 18 está um exemplo da estrutura principal de uma rede criada pelo gerador. Estão representados apenas os *hubs* e os nós arestas da rede. Cada *hub* é representado por um vértice do grafo e seus identificadores possuem o prefixo “FIX”. As arestas mostram por quais nós móveis os *hubs* estão conectados e possuem o prefixo “LINK”.

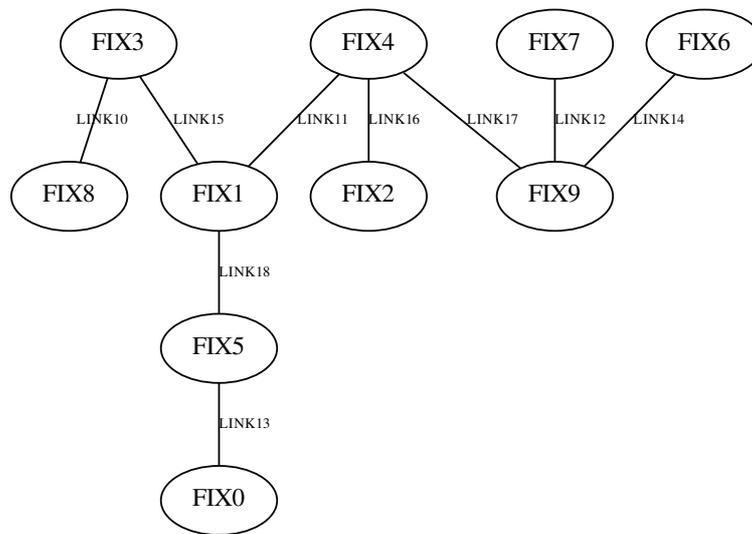


Figura 18 Exemplo de uma rede criada pelo gerador. Notam-se os pontos fixos denominados *hubs* e as arestas que os conectam.

4.3 Configurações e execução das simulações

Para realizar as simulações foram gerados 20 cenários de cada tipo especificado nas Tabelas 7 e 8. Os protocolos de roteamento *Spray and Wait*, *Spray and Focus* e o protocolo desenvolvido *GrubiRouter* foram aplicados sobre os mesmos cenários gerados. Dessa forma pôde-se avaliar os protocolos sobre um mesmo cenário, garantindo a confiabilidade dos dados obtidos. Foram realizadas ao todo 360 simulações. Foi utilizada a parte A do mapa representado na Figura 8 da seção 2.2.3. Seu tamanho mostrou-se

satisfatório para a execução das simulações.

Os protocolos *Spray and Wait* e *Spray and Focus* foram utilizados por serem métodos clássicos e apresentarem resultados relevantes na literatura. Além disso, os protocolos estão disponíveis no pacote do simulador utilizado.

Foram criados dois experimentos variando o número de *hubs* e o número de nós aleatórios e visitantes. Os parâmetros dos cenários gerados podem ser vistos nas Tabelas 7 e 8.

Tabela 7 Parâmetros para o experimento 1. Nesse experimento variou-se o número de *hubs* na rede.

Experimento 1						
	C1	C2	C3	C4	C5	C6
Qtd. <i>Hubs</i>	5	6	7	8	9	10
Qtd. Nós Aleatórios	15	15	15	15	15	15
Qtd. Nós Visitantes	15	15	15	15	15	15

Tabela 8 Parâmetros para o experimento 2. Nesse experimento variou-se o número de nós aleatórios e visitantes para observar o comportamento dos protocolos quanto a quantidade de nós.

Experimento 2						
	C1	C2	C3	C4	C5	C6
Qtd. <i>Hubs</i>	7	7	7	7	7	7
Qtd. Nós Aleatórios	15	20	25	30	35	40
Qtd. Nós Visitantes	15	20	25	30	35	40

Para realizar a execução das simulações os cenários foram divididos em quatro computadores idênticos que executaram as simulações paralelamente. A configuração de cada computador está descrita na Tabela 9 e localizam-se no laboratório do Grupo de Redes Ubíquas (Grubi) pertencente ao Departamento de Ciência da Computação da Universidade Federal de Lavras.

Tabela 9 Configuração dos computadores utilizados na execução das simulações.

Configuração do cluster	
Processador	Intel(R) Core(TM)2 Quad CPU Q8400
Frequência do processador	2.66GHz
Arquitetura do processador	64-bit
L2 Cache	4 MB
Memória RAM	4 GB
Capacidade do HD	320 GB
Sistema Operacional (SO)	Ubuntu 13.04 3.8.0-31-generic
Arquitetura do SO	32-bit

4.4 Resultados obtidos

A seguir serão descritos os dados analisados e obtidos das simulações executadas. Para este trabalho foram considerados os resultados sobre o tempo de propagação das rotas, taxa de entrega das mensagens, *overhead* e latência.

4.4.1 Análise sobre o compartilhamento das rotas

A Figura 19 apresenta o tempo médio para propagação e atualização completa da tabela de roteamento entre os *hubs* da rede.

É possível observar uma ligeira elevação quanto ao tempo entre o cenário com 5 *hubs* e o cenário com 10 *hubs*. Essa elevação é explicada pela quantidade necessária de saltos para que a informação sobre as rotas chegue a todos os *hubs* da rede. A topologia gerada também influencia no compartilhamento da informação entre os *hubs*, pois os mesmos podem estar a um salto de distância ou de 4 a 9 saltos dependendo da topologia da rede.

O resultado apresentado é importante para determinar o tempo de simulação que será utilizado para a análise sobre outros dados significativos, pois o roteamento é totalmente efetuado após o compartilhamento completo

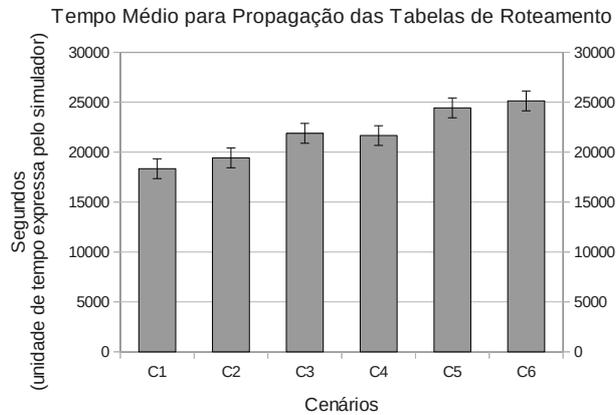


Figura 19 Tempo médio para propagação completa das tabelas de roteamento em cada cenário.

das rotas entre todos os *hubs* da rede.

4.4.2 Análise sobre a taxa de entrega

A taxa de entrega de mensagens foi comparada com os protocolos *Spray and Wait* e *Spray and Focus* e é mostrado no gráfico da Figura 20 para o experimento 1 e no gráfico da Figura 21 para o experimento 2, as Tabelas 10a e 10b apresentam as estatísticas sobre os resultados obtidos.

Observa-se que o protocolo *Spray and Wait* apresenta taxas menores que o protocolo desenvolvido e que o protocolo *Spray and Focus*. O protocolo *GrubiRouter* diminui a complexidade do problema de roteamento em uma rede oportunista, facilitando o roteamento de mensagens como em uma rede tradicional. Esse comportamento é expressado pela alta taxa de entrega apresentada no gráfico.

O comportamento do protocolo *Spray and Wait* compromete a taxa de entrega, já que a duplicação e disseminação de mensagens é limitada. Outro fator importante é a topologia da rede. O protocolo *GrubiRouter*

Tabela 10a Estatísticas sobre a taxa de entrega. Resultados obtidos do experimento 1.

	GrubiRouter			SprayAndWait			SprayAndFocus		
	\bar{x} ^a	σ ^b	Int. Confiança	\bar{x}	σ	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	0,9977	0,0040	0,0017	0,8261	0,1932	0,0847	0,9908	0,0157	0,0069
C2	0,9945	0,0057	0,0025	0,5029	0,3649	0,1599	0,9793	0,0219	0,0096
C3	0,9942	0,0062	0,0027	0,5481	0,2785	0,1221	0,9811	0,0183	0,0080
C4	0,9950	0,0070	0,0031	0,5828	0,3650	0,1600	0,9781	0,0279	0,0122
C5	0,9801	0,0657	0,0288	0,3851	0,3293	0,1443	0,9558	0,0517	0,0227
C6	0,9898	0,0092	0,0025	0,4381	0,3592	0,1574	0,9385	0,0697	0,0305

Tabela 10b Estatísticas sobre o *overhead*. Resultados obtidos do experimento 2.

	GrubiRouter			SprayAndWait			SprayAndFocus		
	\bar{x} ^a	σ ^b	Int. Confiança	\bar{x}	σ	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	0,9924	0,0097	0,0042	0,5532	0,3720	0,1630	0,9836	0,0242	0,0106
C2	0,9943	0,0059	0,0026	0,5306	0,3372	0,1478	0,9660	0,0294	0,0129
C3	0,9779	0,0578	0,0253	0,4472	0,3799	0,1665	0,9325	0,0908	0,0398
C4	0,9927	0,0086	0,0038	0,4578	0,4016	0,1760	0,9166	0,0797	0,0349
C5	0,9940	0,0069	0,0030	0,5935	0,3519	0,1542	0,9290	0,0833	0,0365
C6	0,9945	0,0057	0,0025	0,5144	0,3347	0,1467	0,9189	0,1221	0,0535

^aMédia dos resultados.

^bDesvio padrão dos resultados.

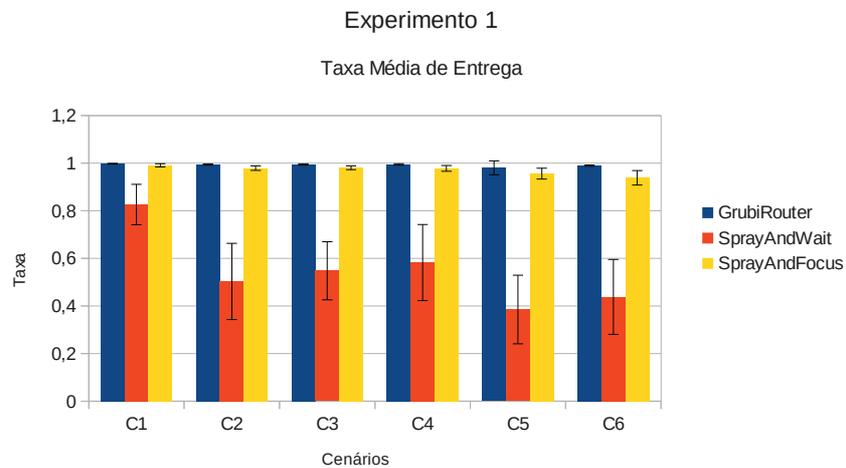


Figura 20 Taxa de entrega calculada para cada protocolo. Os dados foram obtidos do experimento 1.

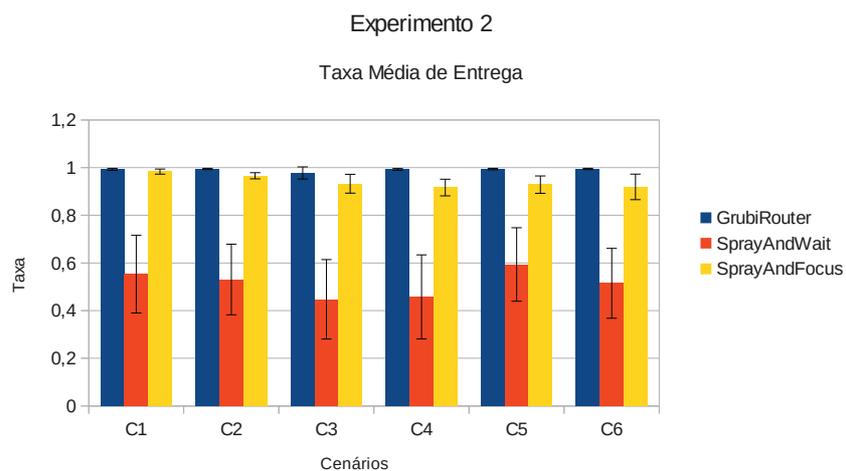


Figura 21 Taxa de entrega calculada para cada protocolo. Os dados foram obtidos do experimento 2.

foi projetado para rotear mensagens em uma topologia que atenda aos requisitos de um padrão de movimento sociológico, já o *Spray and Wait* foi projetado para cenários onde a rede possui uma topologia totalmente dinâmica, a qual não ocorre em determinadas situações. Outra desvantagem do

Spray and Wait é a duplicação de mensagens que ocorre de forma controlada, mas ainda influencia no consumo de recursos da rede.

Utilizando uma técnica de mobilidade assistida o protocolo *Spray and Focus* equipara-se ao protocolo *GrubiRouter*. Com essa técnica ele consegue reconhecer os nós que possuem contato com os destinatários das mensagens e efetuar um roteamento eficiente. Mas no gráfico da Figura 21 nota-se que a taxa de entrega do protocolo *Spray and Focus* decai conforme o número de nós eleva-se enquanto a taxa de entrega do protocolo *GrubiRouter* permanece em um patamar constante. Esse resultado mostra a sensibilidade do protocolo *Spray and Focus* quanto ao número de nós da rede, que pode ter seu funcionamento comprometido caso o número de nós aumente significativamente.

4.4.3 Análise sobre o *overhead*

Os gráficos presentes nas Figuras 22 e 23 mostram a quantidade de mensagens de controle criadas nos cenários simulados, as estatísticas sobre os resultados obtidos estão nas Tabelas 11a e 11b respectivamente para os experimentos 1 e 2.

Os valores apresentados no gráfico da Figura 23 apresentam um elevado nível de *overhead* criado pelo protocolo *GrubiRouter*. É possível observar que esse nível eleva-se quando o número de *hubs* virtuais aumenta, pois é necessário também um maior número de nós arestas. Os *hubs* e os nós arestas são responsáveis por gerar as mensagens de controle para formar e compartilhar as informações sobre as rotas da rede. No entanto, o resultado apresentado é significativamente baixo comparado ao protocolo *Spray and Focus*, pois o mesmo atingiu um *overhead* mínimo de 38972 mensagens para

Tabela 11a Estatísticas sobre o *overhead*. Resultados obtidos do experimento 1.

	GrubiRouter		SprayAndWait		SprayAndFocus	
	\bar{x}^a	σ^b	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	2347,750	300,344	131,629	928,050	117,894	51,668
C2	2704,000	337,840	148,062	758,150	209,784	91,940
C3	3117,500	359,062	157,363	776,450	159,672	69,978
C4	3481,100	243,808	106,852	801,550	208,098	91,201
C5	4129,200	345,100	151,244	687,550	195,931	85,869
C6	4446,300	414,420	181,624	734,250	219,326	96,122
				\bar{x}	σ	Int. Confiança
				38972,400	5252,863	2302,126
				42527,000	5071,195	2222,509
				45544,900	4401,987	1929,220
				52060,200	4159,620	1823,000
				57521,600	6384,766	2798,196
				60109,400	4504,543	1974,167

Tabela 11b Estatísticas sobre o *overhead*. Resultados obtidos do experimento 2.

	GrubiRouter		SprayAndWait		SprayAndFocus	
	\bar{x}^a	σ^b	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	2893,150	105,707	46,327	757,650	190,179	83,348
C2	3548,350	397,964	174,412	791,750	224,946	98,585
C3	3955,900	623,662	273,327	757,200	248,684	108,988
C4	4315,800	753,288	330,137	732,900	239,389	104,915
C5	4934,500	1004,044	440,033	826,900	228,278	100,046
C6	5330,900	912,575	399,946	786,000	226,416	99,229
				\bar{x}	σ	Int. Confiança
				29938,500	2435,035	1067,182
				66837,400	6909,753	3028,277
				89077,400	7696,528	3373,090
				116028,300	10093,974	4423,798
				147734,900	13303,873	5830,572
				182784,600	11742,524	5146,293

^aMédia dos resultados.

^bDesvio padrão dos resultados.

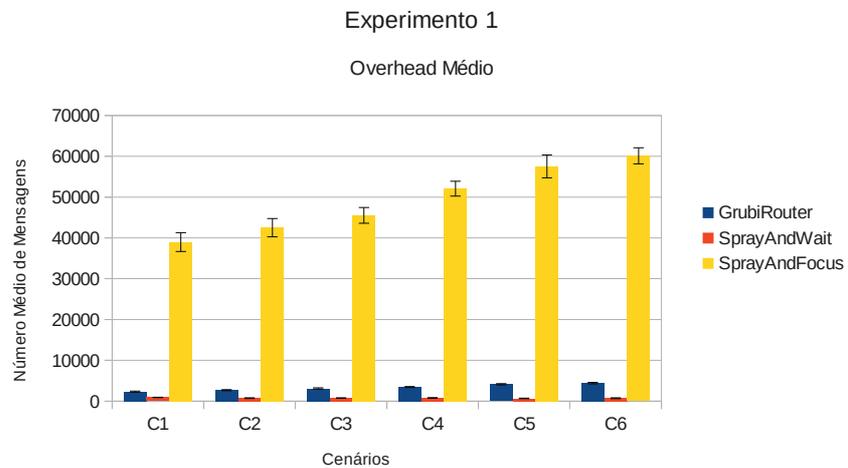


Figura 22 Número médio de mensagens de controle para cada cenário. Os dados foram obtidos do experimento 1.

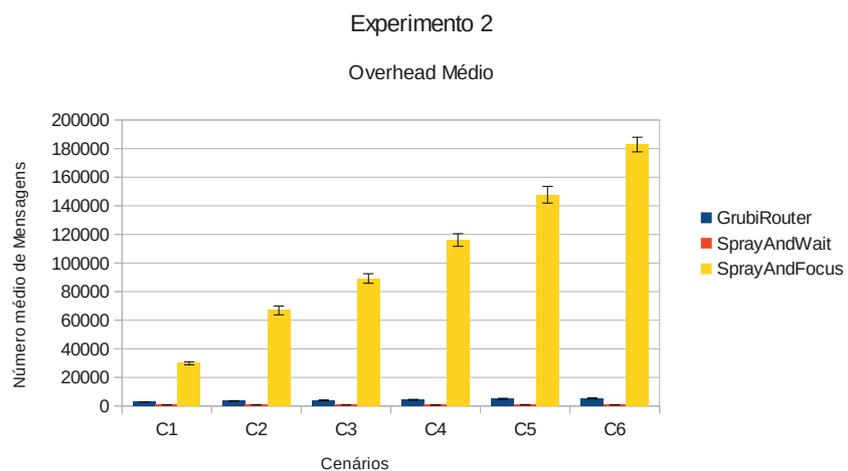


Figura 23 Número médio de mensagens de controle para cada cenário. Os dados foram obtidos do experimento 2.

o cenário com menor número de *hubs* do experimento 1 e um máximo de 60109 mensagens para o cenário com maior número de *hubs* do mesmo experimento. Para os mesmos cenários o protocolo *GrubiRouter* apresentou médias de 2347, em seu mínimo, e 4446 em seu máximo.

No experimento 2 os resultados do protocolo *GrubiRouter* mostraram-se compatíveis como os resultados do experimento 1, mantendo o mesmo patamar de *overhead*, já o protocolo *Spray and Focus* apresentou valores superiores de *overhead* comparado ao primeiro experimento. Fica clara a influência do número de nós sobre o funcionamento do *Spray and Focus*. O protocolo *GrubiRouter* mostrou-se eficiente quanto ao *overhead* por apresentar uma técnica que limita o número de mensagens de controle, utilizados apenas pelos *hubs* e nós arestas.

O protocolo *Spray and Wait* não utiliza mensagens de controle para realizar o roteamento, porém ele utiliza uma técnica para replicar as mensagens para alguns nós da rede. Por isso foi contabilizado nos resultados o número de mensagens replicadas. Os resultados mostram que o nível de *overhead*, tanto no experimento 1 quanto no experimento 2, são muito parecidos e possuem em média 770 mensagens replicadas. No entanto, como mostrado na seção 4.4.2, a taxa de entrega ficou comprometida.

A técnica utilizada pelo protocolo *GrubiRouter* para liberar o *buffer* dos nós favorece no fluxo do controle das tabelas de roteamento, permitindo que as mensagens trafeguem mais rapidamente e que os recursos sejam utilizados com eficiência.

A quantidade de mensagens de controle pode ser controlada de forma a manter o valor mínimo possível para que as informações de rotas sejam mantidas corretamente. Essa quantidade depende do número de *hubs* virtuais e deve ser ajustada para cada cenário.

4.4.4 Análise sobre a latência

Neste trabalho foi avaliada também a latência de cada protocolo de roteamento necessária para efetuar a entrega de mensagens aos seus destinatários. Nos gráficos das Figuras 24 e 25 e nas Tabelas 12a e 12b estão os resultados obtidos das simulações. Apesar de a latência não ser um requisito fundamental em Redes Oportunistas e em Redes Tolerantes a Falhas, seu valor é fundamental para que algumas aplicações funcionem de forma eficiente. Em redes de emergência é necessário que mensagens fluam com rapidez na rede para que urgências sejam respondidas em tempo hábil.

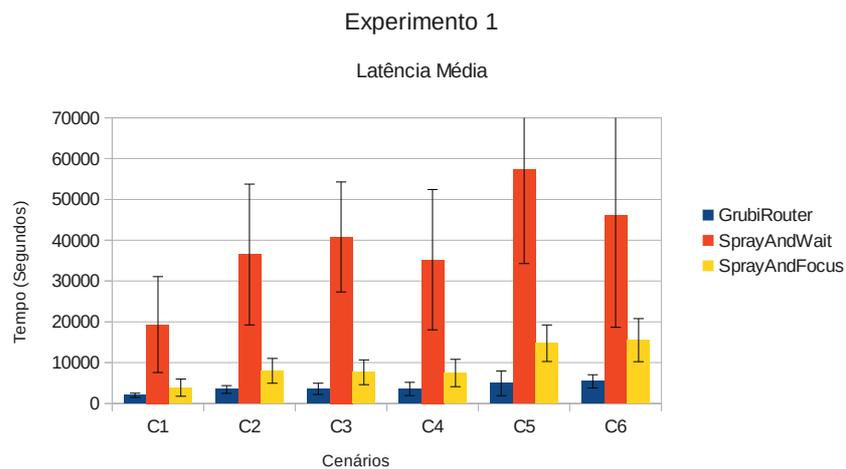


Figura 24 Latência média para entrega de mensagens. Os dados foram obtidos do experimento 1.

Nos dois experimentos realizados ficou claro que o protocolo *GrubiRouter* obteve os níveis de latência mais baixos em todos os cenários simulados. A formação de rotas, a partir de *hubs* e nós arestas, para efetuar a entrega de mensagens é a característica fundamental que contribuiu para este resultado. Com isso a latência fica limitada ao somatório do tempo que permanece armazenada nos *hubs* e ao somatório do tempo gasto para

Tabela 12a Estatísticas sobre a latência para entrega de mensagens. Resultados obtidos do experimento 1.

	GrubiRouter			SprayAndWait			SprayAndFocus		
	\bar{x}^a	σ^b	Int. Confiança	\bar{x}	σ	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	1993,806	1185,819	519,698	19306,786	26826,723	11757,114	3863,836	4835,918	2119,395
C2	3423,222	2149,772	942,162	36486,489	37368,326	17262,969	8010,516	6924,362	3034,680
C3	3590,148	3162,334	1385,928	40791,289	30810,221	13502,927	7626,512	6915,032	3030,591
C4	3554,280	3685,872	1615,375	35216,410	38244,072	17196,316	7462,094	7613,791	3336,830
C5	4912,853	6923,052	3034,105	57393,897	51369,862	23098,283	14750,949	10187,150	4464,634
C6	5412,915	3667,998	1607,541	46148,386	57853,223	27501,171	15507,314	12024,503	5269,874

Tabela 12b Estatísticas sobre a latência para entrega de mensagens. Resultados obtidos do experimento 2.

	GrubiRouter			SprayAndWait			SprayAndFocus		
	\bar{x}^a	σ^b	Int. Confiança	\bar{x}	σ	Int. Confiança	\bar{x}	σ	Int. Confiança
C1	3825,483	3490,246	1529,640	36359,660	38582,770	17348,610	7577,401	7836,074	3434,248
C2	3961,264	2504,479	1097,616	35659,787	34248,259	15009,686	9057,081	7067,614	3097,462
C3	4934,269	3864,229	1693,542	45774,884	47790,389	21488,785	18363,994	17213,773	7544,130
C4	3811,485	3263,018	1430,055	39834,315	48971,307	21462,227	21396,660	19108,305	8374,430
C5	3394,661	2600,995	1139,916	15198,340	27928,427	12239,948	17822,718	22438,138	9833,767
C6	3336,402	2205,879	966,751	25717,600	31303,587	13719,150	17050,877	15408,963	6753,152

^aMédia dos resultados.

^bDesvio padrão dos resultados.

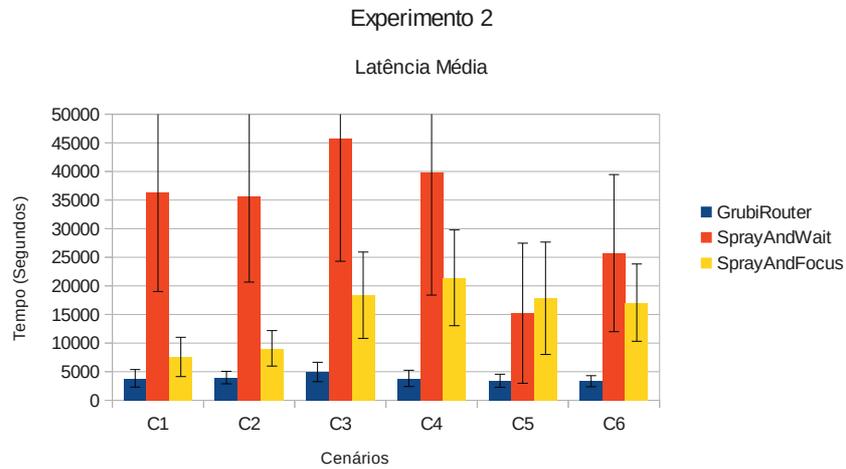


Figura 25 Latência média para entrega de mensagens. Os dados foram obtidos do experimento 2.

transfê-la através de nós arestas.

O protocolo *Spray and Wait* demonstrou os maiores níveis de latência entre os protocolos simulados. Isso ocorre por causa da disseminação de mensagens efetuada pelo protocolo. As mensagens podem percorrer caminhos longos e, em alguma parte dos casos, podem não alcançar os destinatários. Dessa forma a latência fica comprometida.

Quando o padrão de movimento é utilizado para realizar o roteamento nota-se que a latência diminui significativamente. Isso ocorreu tanto com os protocolos *GrubiRouter* e *Spray and Focus* que apresentaram menores taxas de latência. Mesmo assim o protocolo *GrubiRouter* apresentou melhores resultados comparado ao protocolo *Spray and Focus* nos dois experimentos. Observa-se também que o número de nós influencia na latência do protocolo *Spray and Focus* como é possível notar nos dois experimentos, no primeiro onde há um aumento do número de *hubs* e nós arestas e no segundo onde há um aumento de nós aleatórios e visitantes.

5 CONCLUSÃO

Este trabalho mostrou o desenvolvimento de um novo protocolo de roteamento para redes oportunistas. O método desenvolvido utiliza as vantagens do modelo de movimento sociológico para estabelecer pontos fixos denominados *hubs* virtuais e determinar quais nós móveis podem ser utilizados como arestas para interligar os *hubs*. Dessa forma o roteamento pode ser efetuado de forma menos complexa e utilizando menos recursos, estabelecendo rotas entre os *hubs*.

Os resultados apresentados mostram que o desempenho do novo protocolo é superior ao de outros protocolos para cenários que apresentam características do padrão de movimento sociológico. Entretanto os protocolos clássicos como o *Spray and Focus* e *Spray and Wait* podem apresentar melhores resultados do que o protocolo desenvolvido em outros cenários que não apresentam o padrão de movimento sociológico, visto que o *GrubiRouter* não foi desenvolvido para eles.

O *overhead* apresentado pelo protocolo não influenciou em seu desempenho, já que o mesmo é controlado para não haver uso excessivo de recursos da rede. O protocolo *Spray and Wait* apresentou o *overhead* mais baixo entre os três protocolos testados, porém seu desempenho quanto a taxa de entrega e latência ficaram comprometidos. Comparado com o protocolo *Spray and Focus*, o protocolo *GrubiRouter* apresentou o *overhead* de apenas 6% do total apresentado pelo *Spray and Focus* enquanto que, para o mesmo cenário, o protocolo *Spray and Wait* apresentou a menor taxa alcançando 2.38% do total de *overhead* do protocolo *Spray and Focus*.

A alta taxa de entrega de mensagens comprova que o protocolo desenvolvido é eficiente em cenários que apresentam um padrão de movimento

sociológico. Em seu melhor cenário o protocolo atingiu uma taxa de entrega de 0.99 enquanto o protocolo *Spray And Wait* conseguiu uma taxa de 0.82 no seu melhor cenário. Já o protocolo *Spray And Focus* também apresentou uma taxa de 0.99, porém ele também apresentou um alto *overhead* que comprometeu sua eficiência.

Os resultados também apresentaram que o protocolo *GrubiRouter* obteve a menor latência para transferência de mensagens, comprovando que ele pode ser utilizado para aplicações de emergência que exigem baixa taxa de latência.

Com o protocolo implementado atingiram-se os objetivos pretendidos neste projeto. Foi apresentada uma solução para o roteamento de mensagens em Redes Oportunistas utilizando informações de contexto que obteve êxito em seu propósito além de expor resultados promissores para a área de Redes Oportunistas.

5.1 Trabalhos futuros

Como trabalho futuro pretende-se implementar efetivamente a técnica para compartilhamento de mensagens denominada *Floating Content* para que seja possível criar *hubs* apenas utilizando nós móveis. A implementação deve ser realizada juntamente com o protocolo de roteamento. O nó poderá operar em dois modos que permitem efetuar o compartilhamento de rotas, quando o mesmo se encontra dentro dos limites de um *hub*, e realizar o roteamento de mensagens quando encontram-se fora do *hub*.

Outra técnica pretendida é o cálculo dinâmico do intervalo de tempo utilizado para criar e espalhar as mensagens de controle dos *hubs* e nós arestas. Isso permitirá que o número de mensagens de controle seja reduzido

a uma quantidade que satisfaça a conexão e o tempo que um determinado nó aresta gasta para efetuar o movimento entre dois *hubs* da rede.

Por fim pretende-se implementar o protocolo em *smartphones* para realizar experimentos reais. Será utilizado para isso o *framework* da plataforma *Android* para desenvolver e aplicar o protocolo aos dispositivos móveis.

REFERÊNCIAS

BOLDRINI, C.; CONTI, M.; JACOPINI, J.; PASSARELLA, A. Hibop: a history based routing protocol for opportunistic networks. In: **World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a**. [S.l.: s.n.], 2007. p. 1–12.

BUJARI, A.; PALAZZI, C. E.; BONALDO, D. Performance evaluation of a file sharing dtn protocol with realistic mobility. In: **Wireless Days (WD), 2011 IFIP**. [S.l.: s.n.], 2011. p. 1–6. ISSN 2156-9711.

BURGESS, J.; GALLAGHER, B.; JENSEN, D.; LEVINE, B. N. Maxprop: Routing for vehicle-based disruption-tolerant networks. In: **INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings**. [S.l.: s.n.], 2006. p. 1–11. ISSN 0743-166X.

BURNS, B.; BROCK, O.; LEVINE, B. MV routing and capacity building in disruption tolerant networks. In: **INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE**. [S.l.: s.n.], 2005. v. 1, p. 398–408 vol. 1. ISSN 0743-166X.

CAO, Y.; SUN, Z. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. **Communications Surveys Tutorials, IEEE**, PP, n. 99, p. 1–24, 2012. ISSN 1553-877X.

DATTA, A.; QUARTERONI, S.; ABERER, K. Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks. In: BOUZEGHOUB, M.; GOBLE, C.; KASHYAP, V.; SPACCAPIETRA, S. (Ed.). **Semantics of a Networked World. Semantics for Grid Databases**. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3226). p. 126–143. ISBN 978-3-540-23609-2.

FARAHANI, S. **ZigBee Wireless Networks and Transceivers**. Newton, MA, USA: Newnes, 2008. ISBN 0750683937, 9780750683937.

GHOSH, J.; PHILIP, S.; QIAO, C. Sociological orbit aware location approximation and routing in manet. In: **Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on**. [S.l.: s.n.], 2005. p. 641 – 650 Vol. 1.

GRASIC, S.; LINDGREN, A. An analysis of evaluation practices for dtn routing protocols. In: **Proceedings of the Seventh ACM International Workshop on Challenged Networks**. New York, NY, USA: ACM, 2012. (CHANTS '12), p. 57–64. ISBN 978-1-4503-1284-4.

GROSSGLAUSER, M.; TSE, D. Mobility increases the capacity of ad hoc wireless networks. **Networking, IEEE/ACM Transactions on**, v. 10, n. 4, p. 477 – 486, aug 2002. ISSN 1063-6692.

GROSSGLAUSER, M.; VETTERLI, M. Locating nodes with ease: last encounter routing in ad hoc networks through mobility diffusion. In: **INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies**. [S.l.: s.n.], 2003. v. 3, p. 1954 – 1964 vol.3. ISSN 0743-166X.

HUI, P.; CROWCROFT, J.; YONEKI, E. Bubble rap: Social-based forwarding in delay-tolerant networks. **Mobile Computing, IEEE Transactions on**, v. 10, n. 11, p. 1576 –1589, nov. 2011. ISSN 1536-1233.

HYYTIA, E.; LASSILA, P.; OTT, J.; KANGASHARJU, J. Floating information with stationary nodes. In: **Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on**. [S.l.: s.n.], 2012. p. 361–366.

KANGASHARJU, J.; OTT, J.; KARKULAHTI, O. Floating content: Information availability in urban environments. In: **Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on**. [S.l.: s.n.], 2010. p. 804–808.

KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. The ONE Simulator for DTN Protocol Evaluation. In: **SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques**. New York, NY, USA: ICST, 2009. ISBN 978-963-9799-45-5.

LEE, J.-S.; SU, Y.-W.; SHEN, C.-C. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In: **Industrial**

Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE. [S.l.: s.n.], 2007. p. 46–51.

LEGUAY, J.; LINDGREN, A.; SCOTT, J.; FRIEDMAN, T.; CROWCROFT, J. Opportunistic content distribution in an urban setting. In: **Proceedings of the 2006 SIGCOMM workshop on Challenged networks.** New York, NY, USA: ACM, 2006. (CHANTS '06), p. 205–212. ISBN 1-59593-572-X.

LILIEN, L.; KAMAL, Z.; BHUSE, V.; GUPTA, A. The concept of opportunistic networks and their research challenges in privacy and security. In: MAKKI, S.; REIHER, P.; MAKKI, K.; PISSINOU, N.; MAKKI, S. (Ed.). **Mobile and Wireless Network Security and Privacy.** [S.l.]: Springer US, 2007. p. 85–117. ISBN 978-0-387-71057-0.

LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. **SIGMOBILE Mob. Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 7, n. 3, p. 19–20, jul. 2003. ISSN 1559-1662.

MAKKI, S. **Mobile and Wireless Network Security and Privacy.** [S.l.]: Springer Science+Business Media, LLC, 2007. ISBN 9780387710587.

MUSOLESI, M.; HAILES, S.; MASCOLO, C. Adaptive routing for intermittently connected mobile ad hoc networks. In: **World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a.** [S.l.: s.n.], 2005. p. 183 – 189.

NGUYEN, H. A.; GIORDANO, S. Spatiotemporal routing algorithm in opportunistic networks. In: **World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a.** [S.l.: s.n.], 2008. p. 1 –6.

_____. Routing in opportunistic networks. **IJACI**, v. 1, n. 3, p. 19–38, 2009.

_____. Context information prediction for social-based routing in opportunistic networks. **Ad Hoc Networks**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 10, n. 8, p. 1557–1569, nov. 2012. ISSN 1570-8705.

NGUYEN, H. A.; GIORDANO, S.; PUIATTI, A. Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman).

In: **World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a.** [S.l.: s.n.], 2007. p. 1 –6.

PELUSI, L.; PASSARELLA, A.; CONTI, M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. **Communications Magazine, IEEE**, v. 44, n. 11, p. 134–141, November 2006. ISSN 0163-6804.

PERKINS, C. E.; BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 24, n. 4, p. 234–244, out. 1994. ISSN 0146-4833.

SCHOLLMEIER, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: **Peer-to-Peer Computing, 2001. Proceedings. First International Conference on.** [S.l.]: IEEE Computer Society, 2001. p. 101 –102.

SINGH, T. P.; SINGH, D. R.; VATS, J. Article: Routing protocols in ad hoc networks: A review. **International Journal of Computer Applications**, v. 25, n. 4, p. 30–35, July 2011. Published by Foundation of Computer Science, New York, USA.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking.** New York, NY, USA: ACM, 2005. (WDTN '05), p. 252–259. ISBN 1-59593-026-4.

_____. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In: **Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on.** [S.l.: s.n.], 2007. p. 79–85.

VAHDAT, A.; BECKER, D. **Epidemic routing for partially-connected ad hoc networks.** [S.l.], 2000.