



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

AVALIAÇÃO DE PROTOCOLOS DE ROTEAMENTO DE MOBILIDADE SOCIAL EM
UM CENÁRIO DE UMA CONFERÊNCIA CIENTÍFICA

Alan da Silva Sant' Anna
Marcelo de Castro Endson

Orientador
Carlos Alberto Vieira Campos
Coorientador
Nelson Machado Junior

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2017

Sant'Anna, Alan da Silva.

S231 Avaliação de protocolos de roteamento de mobilidade social de um cenário de uma conferência científica / Alan da Silva Sant'Anna, Marcelo de Castro Endson. -- Rio de Janeiro, 2017.
90 f.

Orientador: Carlos Alberto Vieira Campos.

Coorientador: Nelson Machado Junior.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2017.

AVALIAÇÃO DE PROTOCOLOS DE ROTEAMENTO DE MOBILIDADE SOCIAL EM
UM CENÁRIO DE UMA CONFERÊNCIA CIENTÍFICA

Alan da Silva Sant' Anna
Marcelo de Castro Endson

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do Estado
do Rio de Janeiro (UNIRIO) para obtenção do título de
Bacharel em Sistemas de Informação.

Aprovado por:

Carlos Alberto Vieira Campos (UNIRIO)

Sidney Cunha de Lucena (UNIRIO)

Nelson Machado Junior (UNIRIO)

Cláudio Diego Teixeira de Souza (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2017

Agradecimentos

Alan da Silva Sant'Anna

A Deus, por todas as graças alcançadas.

Ao professor Carlos Alberto Vieira Campos, pela proposta do tema e pela amigável orientação que tornaram possível concluir esta monografia.

Ao aluno de doutorado Nelson Machado Junior, pelo suporte durante a realização deste trabalho. Aos professores do CCET que participaram da minha jornada acadêmica.

Aos meus pais, Eli e Lúcia, e também a minha família por tudo que fizeram por mim.

Aos meus amigos de graduação, pela amizade e ajuda diante das dificuldades, dando especial atenção ao Marcelo C. Endson, pela confiança e consentimento em compartilhar este projeto.

Por último, um agradecimento especial à minha esposa, Marianna, pela paciência durante todos esses anos.

Agradecimentos

Marcelo de Castro Endson

A Deus, por todas as bênçãos concedidas.

A minha companheira, Amanda Cristina Pereira do Nascimento, aos meus pais, Jurandy Brandão Endson e Heloisa de Castro Endson, por todo apoio, inclusive nos momentos difíceis.

Ao professor Carlos Alberto Vieira Campos, pela orientação e por todo esforço em me atender para esclarecer dúvidas sempre que preciso.

Ao aluno de doutorado Nelson Machado Junior, pela paciência e coorientação, elucidando minhas dúvidas referentes ao seu trabalho.

Aos amigos de graduação por proporcionarem trocas de experiências vividas, incluindo aprendizados adquiridos de formação

Aos professores da UNIRIO, com destaque aos que integram o Departamento de Informática Aplicada.

Aos colegas de trabalho da DTIC que me apoiaram sempre. Em especial ao Leonardo de Salles Santos que me ajudou a otimizar o processo de tratamento dos dados, e ao Nail Mohamed Saber Abdo Bekhit que disponibilizou um ambiente nos servidores de testes da DTIC para que eu executasse a maior parte das simulações deste trabalho.

RESUMO

Redes oportunistas são redes móveis intermitentes e pertencem à categoria geral de redes tolerantes a atrasos e desconexões (DTN). Neste tipo de rede, as rotas são construídas de forma dinâmica, e os dados são encaminhados de modo oportunista para qualquer dispositivo que seja provável que a mensagem possa ser entregue mais perto do destinatário. Para isso, os protocolos de roteamento fazem uso da mobilidade destes nós na rede. Este trabalho analisa o desempenho de uma proposta de protocolo oportunista com base social, denominado SOCLEER, em um cenário de mobilidade real, através do uso de um *trace* de contato no simulador de ambiente de rede oportunista, ONE. Esse protocolo foi desenvolvido como uma proposta para resolver o problema dos nós preferidos no envio de mensagens em protocolos baseados em contexto social, com o objetivo de reduzir o consumo de energia desses dispositivos. A simulação foi realizada com o *trace* de conectividade INFOCOM2006 para os protocolos SOCLEER, BUBBLE Rap, PRoPHET e Epidêmico. A rede oportunista criada na simulação foi testada com diferentes cargas de mensagens para a medição da performance dos protocolos. Os valores das cargas foram 1000, 10000, 20000 e 30000 mensagens. Para tornar a simulação mais realista, foram realizadas dez rodadas de simulação para cada protocolo com cada uma das cargas de mensagens propostas. Após o processamento dos dados, foram calculadas a média, a margem de erro e o intervalo de confiança para cada carga de mensagens, utilizando a distribuição *T-Student* para o tratamento estatístico. A análise da performance foi realizada através das métricas: fração de mensagens entregues, *overhead*, latência, número de saltos por rota, participação dos nós no envio de mensagens e consumo de bateria dos dispositivos móveis.

Palavras-chave: Redes Oportunistas, DTN, SOCLEER, ONE e Protocolos de Roteamento.

ABSTRACT

Opportunistic networks are intermittent mobile networks and belong to the general category of Delay/Disruption tolerant networks (DTN). In this type of network, routes are constructed dynamically, and data is routed opportunistically to any device that is likely to deliver the message closer to the recipient. For this, the routing protocols make use of the mobility of these nodes in the network. This work analyzes the performance of a social context-aware opportunistic protocol proposal, called SOCLEER, in a real mobility scenario, through the use of a contact trace in the opportunistic network environment simulator, ONE. This protocol was developed as a proposal to solve the problem of preferred nodes in sending messages in protocols based on social context, with the purpose of reducing the energy consumption of these devices. The simulation was performed with the INFOCOM2006 connectivity trace for the SOCLEER, BUBBLE Rap, PROPHET and Epidemic protocols. The opportunistic network created in the simulation was tested with different message loads to measure protocol performance. The values of the charges were 1000, 10000, 20000 and 30000 messages. To make the simulation more realistic, ten simulation rounds were performed for each protocol with each of the proposed message loads. After the data processing, the average, the margin of error and the confidence interval for each message load were calculated using the T-Student distribution for the statistical treatment. Performance analysis was performed through the metrics: fraction of messages delivered, overhead, latency, number of jumps per route, participation of nodes in sending messages and battery consumption of mobile devices.

Keywords: Opportunistic Networks, DTN, SOCLEER, ONE and Routing Protocols.

LISTA DE FIGURAS

Figura 1 - Elementos de uma rede local sem fio.	15
Figura 2 - Exemplo de uma rede Ad hoc IEEE 802.11 e uma LAN sem fio.	16
Figura 3 - Exemplo de uma transmissão de dados em uma rede MANET.....	17
Figura 4 - Comunicação transitiva.....	18
Figura 5 - Troca de mensagens entre dois nós usando protocolo Epidêmico.....	23
Figura 6 - Exemplo de encaminhamento do protocolo Epidêmico.	24
Figura 7 - Exemplo do algoritmo de roteamento PRoPHET.....	27
Figura 8 - Ilustração do algoritmo de encaminhamento do <i>BUBBLE</i>	30
Figura 9 - Algoritmo <i>BUBBLE</i> em pseudocódigo.	31
Figura 10 - Pseudocódigo do SOCLEER.	34
Figura 11 - Exemplo da métrica <i>Degree Centrality</i>	36
Figura 12 - Exemplo da métrica <i>Closeness Centrality</i>	37
Figura 13 - Exemplo da métrica <i>Betweenness Centrality</i>	38
Figura 14 - Interface Gráfica do simulador ONE v1.4.1.	39
Figura 15 - Estrutura de funcionamento do Simulador ONE.	40
Figura 16 - Dispositivos iMotes	44
Figura 17 - Parte do conteúdo do arquivo <i>haggle-one-infocom2006-complete.tsv</i>	45
Figura 18 – Grafo de contatos do trace INFOCOM2006	47
Figura 19 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 1000 mensagens.....	53
Figura 20 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 1000 mensagens.....	54
Figura 21- Média de energia residual de todos os nós em mAh para uma carga de 1000 mensagens.....	55
Figura 22 - Visualização aumentada da parte final da simulação da Figura 21 – Média de energia residual de todos os nós em mAh para uma carga de 1000 mensagens.....	55
Figura 23 – Média de energia residual dos nós preferidos identificados no corte de 2% em mAh para uma carga de 1000 mensagens.....	56
Figura 24 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 1000 mensagens.....	56
Figura 25 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 10000 mensagens.....	57

Figura 26 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 10000 mensagens.....	58
Figura 27 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 10000 mensagens.....	58
Figura 28 - Média de energia residual de todos os nós em mAh para uma carga de 10000 mensagens.....	59
Figura 29 - Média de energia residual de todos os nós em mAh para uma carga de 10000 mensagens.....	60
Figura 30 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 10000 mensagens.....	60
Figura 31 - Visualização aumentada da Figura 30 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 10000 mensagens.....	61
Figura 32 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 5% e uma carga de 20000 mensagens.....	61
Figura 33 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 5% e uma carga de 20000 mensagens.....	62
Figura 34 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 20000 mensagens.....	62
Figura 35 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 20000 mensagens.....	63
Figura 36 - Média de energia residual de todos os nós em mAh para uma carga de 20000 mensagens.....	64
Figura 37 – Visualização aumentada da parte final da Figura 36 - Média de energia residual de todos os nós em mAh para uma carga de 20000 mensagens	64
Figura 38 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 20000 mensagens.....	65
Figura 39 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 30000 mensagens.....	65
Figura 40 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 30000 mensagens.....	66
Figura 41 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 30000 mensagens.....	66
Figura 42 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 30000 mensagens.....	67

Figura 43 - Média de energia residual de todos os nós em mAh para uma carga de 30000 mensagens.....	67
Figura 44 – Visualização aumentada da parte final da Figura 43 -Média de energia residual de todos os nós em mAh para uma carga de 30000 mensagens	68
Figura 45 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 30000 mensagens.....	68
Figura 46 - Média de energia residual dos nós preferidos identificados no corte de 2% em mAh para uma carga de 30000 mensagens.....	69
Figura 47 - Fração média de mensagens entregues por mensagens criadas	69
Figura 48 - Latência média por mensagens criadas.....	70
Figura 49 - Overhead médio por mensagens criadas.....	71
Figura 50 - Média de saltos por rota por mensagens criadas.....	72

LISTA DE QUADROS

Quadro 1 - Algumas das aplicações para redes Ad hoc.....	19
Quadro 2 - Uma possível classificação dos principais protocolos de encaminhamentos oportunistas.....	21
Quadro 3 -- Esquema de roteamento Spray and Wait binário.	25

LISTA DE TABELAS

Tabela 1 - Distância geodésica entre os nós da rede.	37
Tabela 2 - Cálculo da métrica <i>Betweness Centrality</i>	38
Tabela 3 - Parâmetros gerais da simulação.....	48
Tabela 4 - Valores dos parâmetros usados para o protocolo SOCCLER.	50
Tabela 5 - Valores dos parâmetros usados para o protocolo <i>BUBBLE Rap</i>	51

LISTA DE ABREVIATURAS E SIGLAS

AP - Access point

CRAWDAD - Community Resource for Archiving Wireless Data At Dartmouth

CLI - Command Line Interface

DNS - Domain Name System

DTN - Delay and disruption Tolerant Networking

GPL - GNU General Public License

IEEE - Institute of Electrical and Electronics Engineers

JDK - Java SE Development Kit

LAN - Local Area Network

MANET - Mobile Ad hoc network

OppNets - Opportunistic Networks

ONE - Opportunistic Network Environment

ProPHet - Probabilistic Routing Protocol using History of Encounters and Transitivity

PSN - Pocket Switched Networks

SOCLEER - Social-based Energy-Efficient Routing Protocol

TTL - Time-to-live

Wi-Fi - Wireless Fidelity

WLAN - Wireless Local Area Network

WPAN - Wireless Personal Area Network

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Motivação	11
1.2	Objetivo	11
1.3	Organização do texto	12
2	REFERENCIAL TEÓRICO	14
2.1	Rede infraestruturada.....	14
2.2	Rede Ad hoc	15
2.2.1	MANET.....	16
2.2.2	Redes Oportunistas.....	18
2.3	Aplicações em Redes Oportunistas	19
3	PROTOCOLOS DE ROTEAMENTO EM REDES OPORTUNISTAS.....	21
3.1	Classificação dos protocolos oportunistas.....	21
3.1.1	Protocolos Não-Sociais	22
3.1.1.1	Protocolo Epidêmico	22
3.1.1.2	Protocolo Spray and Wait.....	24
3.1.1.3	Protocolo PRoPHET.....	26
3.1.2	Protocolos com base social.....	28
3.1.2.1	Protocolo BUBBLE Rap	29
3.1.2.2	Protocolo SOCLEER.....	31
3.2	O Problema dos Nós Preferidos	33
4	TRABALHOS RELACIONADOS	35
4.1	Classificação das métricas para redes sociais.....	35
4.1.1	Comunidade (<i>Community</i>)	35
4.1.2	Centralidade do nó (<i>Node Centrality</i>)	35
4.1.2.1	Centralidade de grau ou grau de conexão (<i>Degree Centrality</i>).....	35
4.1.2.2	Centralidade de proximidade (<i>Closeness Centrality</i>).....	36
4.1.2.3	Centralidade de intermediação (<i>Betweenness Centrality</i>).....	37
4.2	O simulador para ambiente de rede oportunista	38
4.2.1	Principais funções do simulador ONE	39
4.2.2	Configuração do simulador	40
4.2.3	Módulo de Relatórios	41

4.3 Ferramenta externa de pós-processamento.....	42
4.3.1 Gnuplot.....	42
4.3.2 Graphviz	43
5 PARÂMETROS GERAIS DA SIMULAÇÃO.....	44
5.1 Visão geral.....	44
5.2 Descrição dos parâmetros gerais da simulação	47
5.3 Métricas de Avaliação de desempenho	49
6 AVALIAÇÃO DE DESEMPENHO	50
6.1 Cenários de simulações para o SOCLEER.....	50
6.2 Definindo os cortes.....	52
6.2.1 Cenário com Carga de 1000 Mensagens Criadas	53
6.2.2 Cenário com Carga de 10000 Mensagens Criadas	56
6.2.3 Cenário com Carga de 20.000 Mensagens Criadas	61
6.2.4 Cenário com Carga de 30.000 Mensagens Criadas	65
6.2.5 Avaliação do SOCLEER em Relação às Métricas de Desempenho de Rede ...	69
7 CONCLUSÕES	73
7.1 Dificuldades.....	73
7.2 Considerações Finais	73
7.3 Trabalhos Futuros	74
REFERÊNCIAS BIBLIOGRÁFICAS	75

1 Introdução

Este capítulo apresenta a motivação para a escolha do tema deste projeto de graduação, o objetivo do trabalho e a estrutura da monografia.

1.1 Motivação

Com o advento e a recente popularização dos dispositivos móveis, é possível aproveitar a mobilidade dos usuários para a troca de dados utilizando para isso as redes oportunistas, que são uma alternativa para as redes de comunicações convencionais. Este tema é importante pelo fato que as pesquisas recentes em redes oportunistas tentam resolver alguns problemas, como por exemplo, o consumo de energia dos dispositivos móveis, altas taxas de latência, de sobrecarga de mensagens na rede e a participações de determinados dispositivos como preferidos para a entrega das mensagens neste tipo de rede com conexões intermitentes. Portanto, o problema de roteamento em redes DTN ainda está em aberto e formas mais eficientes têm sido pesquisadas.

1.2 Objetivo

O objetivo deste trabalho é avaliar o desempenho de um conjunto de protocolos de encaminhamento para redes oportunistas em um cenário realístico implementado no consagrado simulador para este tipo de rede, o ONE (*Opportunistic Network Environment*).

Para atingir este objetivo, foram usados quatro protocolos de roteamento no simulador: o *Epidemic* (VAHDAT *et al.*, 2000), um protocolo não probabilístico, que propaga diversas cópias da mesma mensagem para todos os nós da rede de forma que esta alcance seu destino final, o PROPHET (LINDGREN *et al.*, 2003), protocolo probabilístico, que usa uma métrica (previsibilidade de entrega) que indica a probabilidade de um nó entregar uma mensagem a um destinatário, o *BUBBLE Rap* (HUI *et al.*, 2008), um protocolo de contexto social, que utiliza a métrica de centralidade no grafo da rede para encontrar os nós que interligam as comunidades, e por último, o SOCLEER (MACHADO, 2013), uma versão modificada do BUBBLE

Rap, que implementa uma melhoria na questão do nó preferido para indicar o melhor caminho entre a origem e o destino de uma mensagem.

Para a realização deste trabalho foi utilizado um cenário (*trace* de conectividade) realístico, obtido da base pública de dados (*dataset*) CRAWDAD¹, conhecido como Infocom2006, que foi uma conferência em redes de computadores realizada em um hotel na cidade de Barcelona, naquele ano. Este *trace* de mobilidade, ou experimento, teve a duração de aproximadamente 4 dias (337.418 segundos, ou ~3.91 dias), onde um grupo de pessoas portavam um pequeno dispositivo de rede sem fio com interface *Bluetooth* denominado *iMote* (98 dispositivos no total) que foram distribuídos para coletar dados de contato (170.601 contatos no total) dos participantes dessa conferência. Este *dataset* foi utilizado como entrada no simulador ONE. Estes dados foram obtidos da página do crawdad.org².

Como resultados obtidos, foi apresentada uma avaliação de desempenho do protocolo SOCLEER em relação aos outros protocolos oportunistas referenciados nesse texto no *trace* de contato do Infocom2006 através de simulações no ONE. Este protocolo foi desenvolvido pelo aluno Nelson Machado Junior (PPGI - UNIRIO) como proposta de sua Dissertação de Mestrado em 2013, intitulado "SOCLEER: Uma proposta de Disseminação de Dados para Redes Oportunistas com Redistribuição de Carga em Nós Preferidos".

1.3 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Apresenta o referencial teórico sobre redes móveis sem fio, redes tolerantes a atrasos e as principais aplicações em redes oportunistas.
- Capítulo III: Descreve os tipos de protocolos de roteamento em redes oportunistas. Este capítulo mostra, de forma resumida, as principais características e o

¹ CRAWDAD - A Community Resource for Archiving Wireless Data At Dartmouth

² <https://crawdad.org/uo/hagg/20160828/one/>

funcionamento de alguns protocolos de redes oportunistas não sociais e os baseados em contexto social.

- Capítulo IV: Apresenta os trabalhos relacionados. Este capítulo descreve as principais métricas para redes sociais mais utilizadas nos protocolos baseados em contexto social e apresenta algumas características do simulador ONE.
- Capítulo V: Apresenta os parâmetros gerais da simulação, a configuração do cenário no simulador e as métricas de avaliação.
- Capítulo VI: Apresenta a avaliação de desempenho do protocolo SOCLEER e os resultados obtidos na simulação.
- Capítulo VII: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2 Referencial Teórico

Este capítulo apresenta de forma resumida os conceitos essenciais sobre as redes sem fio com infraestrutura e as redes sem fio *Ad hoc*, ou sem infraestrutura. Descreve os tipos de redes sem fio móveis e as principais aplicações em redes oportunistas.

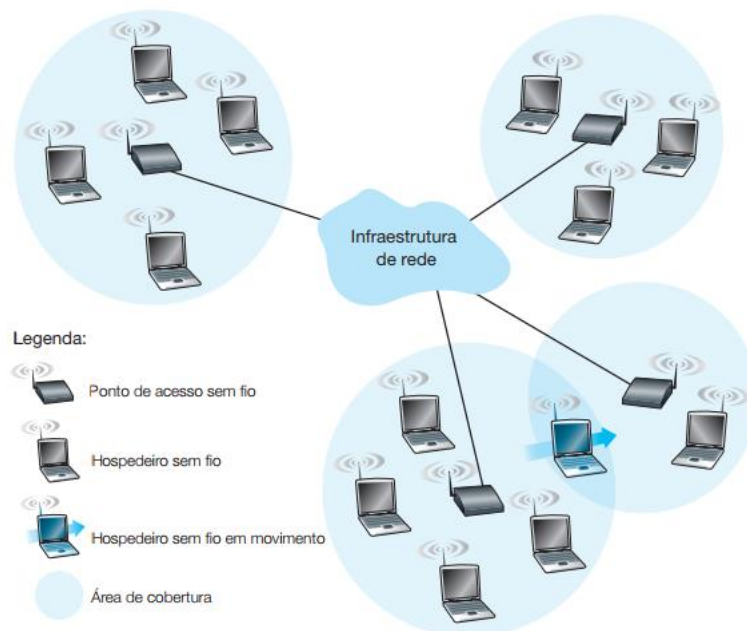
2.1 Rede infraestruturada

Uma rede sem fio não caracteriza, necessariamente, que os nós que fazem parte da rede sem fio são móveis (por exemplo, redes domésticas sem fio ou redes locais de empresas compostas por estações de trabalho e computadores portáteis), visto que os dispositivos móveis dependem da área de cobertura da rede sem fio, fato este que limita a mobilidade dos nós. Esta distinção entre sem fio e mobilidade é importante para entender os conceitos fundamentais em cada área (KUROSE; ROSS, 2007).

A Figura 1 ilustra uma infraestrutura de rede³ local sem fio (*Wireless Local Area Network* — WLAN) utilizando o enlace de comunicação sem fio IEEE 802.11 (Wi-Fi), mostrando a área de cobertura de cada estação-base (ponto de acesso sem fio), os nós da rede (hospedeiro sem fio ou dispositivo móvel). Um ponto de acesso sem fio (*Access Point* — AP) é responsável pela transmissão e recepção de dados de e para um dispositivo móvel que está associado a ele. O AP também coordena a transmissão de vários nós sem fio com os quais está associado. Um nó está associado a um ponto de acesso quando está dentro do alcance de comunicação sem fio dele e o usa para retransmitir dados entre ele e a rede de maior porte. Neste exemplo, os enlaces sem fio conectam os nós com a infraestrutura da rede de maior porte (cabeada), onde estão localizados os equipamentos de conectividade da infraestrutura de rede, como roteadores, comutadores (*switches*) dentre outros. Nesta figura, um nó se desloca para fora do alcance de um ponto de acesso e entra na faixa de um outro ponto, mudando para este a sua associação — um processo denominado transferência (*handoff*) (KUROSE; ROSS, 2007).

³ Infraestrutura de rede. É a rede maior com a qual um nó sem fio pode querer se comunicar.

Figura 1 - Elementos de uma rede local sem fio.



Fonte: KUROSE; ROSS, 2007, p. 383.

2.2 Rede Ad hoc

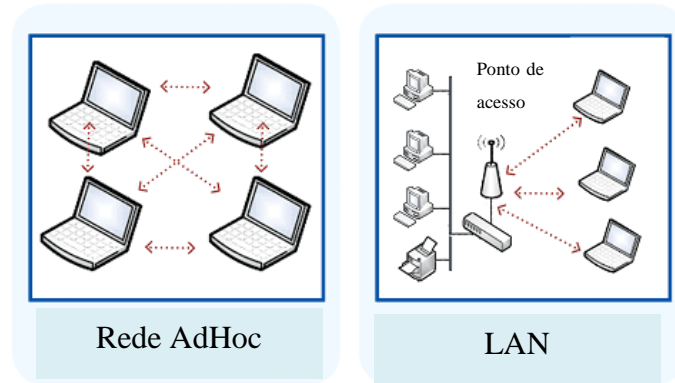
O termo *Ad hoc* é uma expressão em latim que significa "para isto", sendo geralmente usada para designar uma solução concebida para um propósito específico, sem a intenção de que ela seja generalizada.

Uma rede *Ad hoc* é uma rede temporária. Neste tipo de rede, os nós sem fio não dispõem de nenhuma infraestrutura de rede para a qual se conectar. Na ausência de tal infraestrutura, os próprios nós devem prover serviços como roteamento, atribuição de endereço, tradução de endereços semelhante ao DNS⁴ e outros. Uma rede *Ad hoc* é formada conforme a necessidade, por dispositivos móveis que, por acaso, estão próximos uns dos outros, têm necessidades de se comunicar e não dispõem de infraestrutura de rede no ambiente em que se encontram (KUROSE; ROSS, 2007).

⁴ DNS. *Domain Name System*. É um sistema de gerenciamento de nomes hierárquico e distribuído. Sua tarefa é resolver o nome de hospedeiro para o endereço IP correspondente.

A Figura 2 ilustra dispositivos móveis que usam o padrão para LAN (*Local Area Network*) sem fio 802.11 que também podem se agrupar e formar uma rede *Ad hoc*.

Figura 2 - Exemplo de uma rede Ad hoc IEEE 802.11 e uma LAN sem fio.



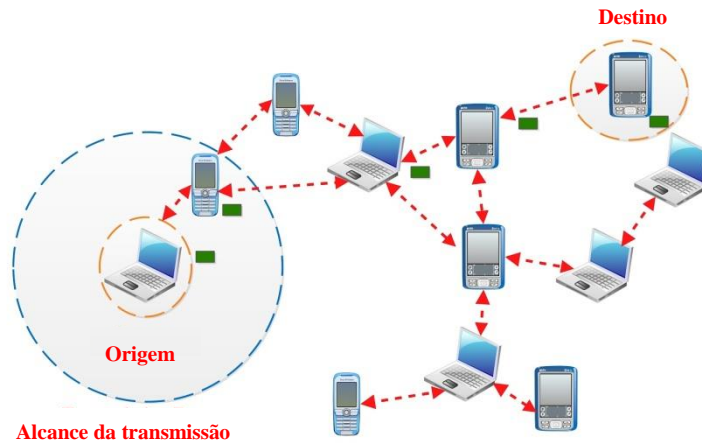
Fonte: How to make devices communicate in a wireless world. Disponível em:

<<http://bit.ly/2y13f3W>>.

2.2.1 MANET

Uma rede *Ad hoc* móvel (*Mobile Ad hoc network* — MANET) é uma coleção de dois ou mais dispositivos móveis que se comunicam através de uma rede sem fio temporária desprovida de qualquer tipo de administração e infraestrutura. Este tipo de rede possui como uma de suas características o fato de ser auto reconfigurável, ou seja, os nós entram e saem desta rede continuamente, movendo-se randomicamente em diferentes direções e velocidades, tornando a topologia da rede dinâmica. Cada nó possui conectividade apenas com os nós mais próximos e a transmissão da mensagem é por broadcast, porém o destinatário (e *next hop*) é determinado pelo protocolo. A Figura 3 ilustra uma MANET formada por dispositivos sem fio e a transferência da mensagem (ou pacote) entre o nó origem e destino através do contato entre os dispositivos móveis intermediários que estão ao alcance da transmissão.

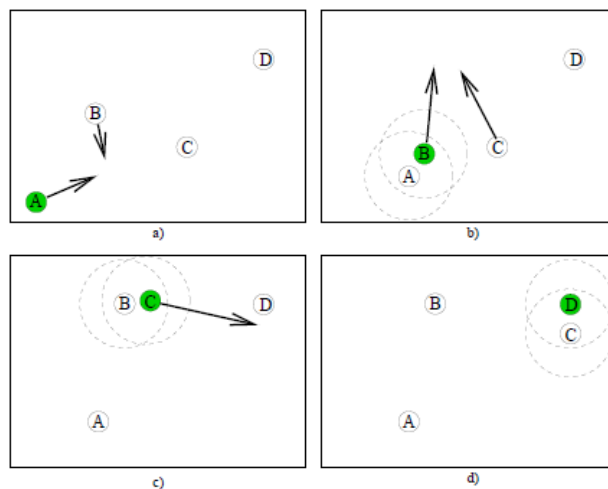
Figura 3 - Exemplo de uma transmissão de dados em uma rede MANET.



Fonte: <http://vaibhav-godbole.blogspot.com.br/2012/07/mobility-models-and-traffic-pattern.html>

Um modo de habilitar a comunicação entre dispositivos móveis quando nenhuma infraestrutura de rede está disponível, é permitindo que as mensagens sejam armazenadas em *buffer* durante um longo tempo em nós intermediários com o intuito de explorar a mobilidade desses nós para levar as mensagens até o destino, transferindo mensagens para outros nós, caso ocorra um encontro. A Figura 4 mostra como a mobilidade dos nós em tais cenários pode ser usada para eventualmente entregar uma mensagem para o seu destino. Nesta figura, o nó A tem uma mensagem (indicada pelo nó sombreado) para ser entregue ao nó D, mas não existe um caminho entre eles. Como mostrado nas subfiguras, a mobilidade dos nós permite que a mensagem primeiro seja transferida para o nó B, em seguida para o nó C, e finalmente o nó C se move dentro do alcance do nó D, podendo entregar a mensagem para o seu destino final (LINDGREN *et al*, 2003).

Figura 4 - Comunicação transitiva



Fonte: LINDGREN *et al.* 2003.

2.2.2 Redes Oportunistas

As redes oportunistas (*Opportunistic Networks* — OppNets) enfrentam grandes desafios no estabelecimento de processos de comunicação eficientes. Estas redes são aplicadas em situações onde não existe um caminho fim-a-fim persistente entre o emissor e o receptor. Redes oportunistas são chamadas de Redes Tolerantes a Atrasos e Desconexões (*Delay and disruption Tolerant Networking* — DTN).

Redes oportunistas estabelecem um cenário idêntico a uma MANET, no qual os nós existentes são capazes de comunicar uns com os outros, mesmo que nunca exista uma rota que os una diretamente. Apesar disso, também se diferenciam da MANET na medida em que não assumem a necessidade de cada nó possuir ou adquirir conhecimento sobre a topologia da rede, o que é necessário nos protocolos de roteamento para redes MANET (PELUSI *apud* QUELHAS *et al.*, 2011).

Em redes oportunistas, as rotas são construídas dinamicamente, enquanto as mensagens são enviadas entre o remetente e o destino, e qualquer possível nó pode, de modo oportunista, ser usado como o próximo salto, desde que seja o mais provável a entregar a mensagem ao destino final. As mensagens são armazenadas enquanto não surge uma oportunidade de entrega, estratégia conhecida por *store-and-forward*. Esses requisitos tornam as redes oportunistas um campo de pesquisa desafiador e promissor (PELUSI *et al.* 2006).

2.3 Aplicações em Redes Oportunistas

Redes móveis conectadas intermitentemente pertencem à categoria geral de redes tolerantes a atrasos e desconexões (DTN). Diversas redes reais se encaixam nesse paradigma. Exemplos incluem rastreamento de vida selvagem, redes de sensores de monitoramento de habitat, redes militares, redes interplanetárias, redes em comunidades nômades, etc. (SPYROPOULOS *et al* 2005).

O Quadro 1 mostra algumas das aplicações para redes *Ad hoc*:

Quadro 1 - Algumas das aplicações para redes Ad hoc.

Aplicações	Possíveis serviços de redes Ad hoc
Redes Táticas	Comunicação militar, operações militares e conflitos.
Serviços de Emergências	Operações de busca e resgate em ambientes inóspitos, substituição da rede infraestruturada em caso de desastres, policiamento, combate a incêndios e apoio hospitalar.
Extensão de Cobertura	Extensão do acesso à rede de telefonia celular, vinculação com a internet, intranets e outras.
Redes de Sensores	Dentro de casa: sensores inteligentes e atuadores embutidos em dispositivos eletrônicos de consumo, redes de área do corpo (BAN - <i>Body area networks</i>), acompanhamento de dados de condições ambientais, rastreamento de animais selvagens e detecção química e biológica.
Educação	Configurações de rede em campus universitários, salas de aula, implantação de rede ad hoc em reuniões ou palestras, jogos multiusuários, rede P2P sem fio, acesso à internet em ambientes externos, animais robóticos e parques temáticos.
Redes domésticas e empresariais	Uso de rede sem fio em casa ou escritório, conferências, salas de reuniões, rede de área pessoal (PAN), e redes pessoais.
Serviços com contexto social	Informações turísticas, encaminhamento de chamadas e espaço de trabalho móvel.
Ambientes comerciais e civis	E-commerce: pagamentos eletrônicos a qualquer hora e em qualquer lugar.

	<p>Negócios: acesso dinâmico ao banco de dados, escritórios móveis.</p> <p>Serviços de veículos: orientação rodoviária ou acidental, transmissão de condições rodoviárias e climáticas, rede de táxi, redes interveiculares.</p> <p>Estádios de esportes, feiras, shopping centers e assim por diante.</p> <p>Redes de visitantes dentro dos aeroportos.</p>
--	--

Fonte: AL-OMARI; SUMARI, 2010.

3 Protocolos de Roteamento em Redes Oportunistas

Este capítulo apresenta alguns consagrados protocolos de roteamento oportunistas da abordagem clássica, os protocolos baseados em contexto social usados neste trabalho e o problema dos nós preferidos.

3.1 Classificação dos protocolos oportunistas

Os protocolos de roteamento *Ad hoc* móveis permitem que os nós com adaptadores sem fio se comuniquem uns com ou outros sem qualquer infraestrutura de rede pré-existente, ou seja, quando não existe um caminho físico associado entre a origem e o destino (VAHDAT; BECKER, 2000).

Os protocolos oportunistas podem ser classificados de acordo com o tipo de informação que eles exploram ao tomar decisões de encaminhamento. Nesse contexto, o roteamento pode ser dividido em dois grupos: protocolos de roteamento não-sociais (*no context*) e protocolos de roteamento sensíveis ao contexto social (*social context-aware*). O Quadro 2 sumariza os principais protocolos baseado nessa possível classificação.

Quadro 2 - Uma possível classificação dos principais protocolos de encaminhamentos oportunistas.

Protocolos não-sociais		Protocolos sensíveis ao contexto social (<i>full context, context-based or social context-aware</i>)
Não probabilísticos (<i>no context or dissemination-based</i>)	Probabilísticos (<i>partial context or Partially context-aware</i>)	HiBOP SimBet SSAR
Epidemic Single-copy algorithms Spray and wait Spray and focus	PRoPHET MaxProp CAR HYMAD	People rank SREP 3R BUBBLE Rap SOCLEER

O grupo de protocolos não-sociais pode ser subdividido em protocolos probabilísticos e não probabilísticos. Dentre os existentes, serão abordados nesta seção

os protocolos Epidêmico e Spray and Wait (não probabilísticos), PRoPHET (probabilístico) e os protocolos BUBBLE Rap e o recém-proposto SOCLEER (com base social).

Os recentes trabalhos de pesquisa demonstraram que os protocolos de roteamento probabilísticos e os baseados em contexto social se destacaram como estratégias de maior sucesso.

3.1.1 Protocolos Não-Sociais

Os protocolos não sociais (*dissemination-based*) foram a primeira tentativa de encaminhamento em redes oportunistas. Estes protocolos não exploram qualquer informação contextual sobre o estado ou o comportamento dos dispositivos, usuários e ambiente. Consiste em uma estratégia de encaminhamento básica (*broadcast*⁵), onde as mensagens são espalhadas pela rede a cada novo contato. (VAHDAT; BECKER, 2000). Os protocolos Epidêmico e Spray and Wait enquadram-se nessa categoria.

A categoria *Partially context-aware* explora as informações de contexto, mas assumem um modelo específico. Quando o ambiente combina com estes pressupostos, o desempenho desses protocolos é muito bom, mas se o ambiente passa a ser diferente do que eles supõem, a operação pode não ocorrer da forma certa (MACHADO, 2013). O protocolo PRoPHET enquadra-se nesta categoria.

3.1.1.1 Protocolo Epidêmico

O protocolo de encaminhamento Epidêmico (*epidemic routing*) tem como objetivos maximizar a taxa de entrega de mensagens, minimizar a latência (atraso) das mensagens e minimizar os recursos totais consumidos na entrega de mensagens (VAHDAT; BECKER, 2000).

O funcionamento desse protocolo pode ser comparado com uma doença epidêmica. Em uma epidemia, uma pessoa infectada por um vírus, por exemplo, dissemina esse vírus para outras pessoas com quem ela entra em contato. Sucessivamente, as pessoas que foram infectadas repassarão o vírus para outras

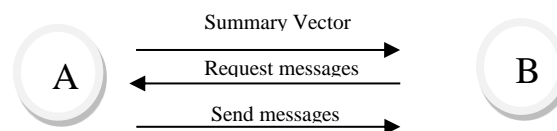
⁵ Processo pelo qual se transmite determinada informação, para muitos receptores ao mesmo tempo

pessoas que se aproximarem delas. Nesta analogia, o "vírus" pode ser entendido como a informação que se deseja propagar (ALBINI, 2013).

Uma conexão é estabelecida quando um nó entra no alcance de outro. Posteriormente, os dispositivos trocam suas listas de mensagens armazenadas. Consequentemente, a lista recebida é comparada com as mensagens presentes no nó, para determinar quais entradas da lista ele não possui. Após, o dispositivo solicita o encaminhamento de cópias dessas mensagens. Esse processo de troca é repetido toda vez que um nó estabelece contato com outro, logo permite que os dados sejam compartilhados na rede de forma rápida. Em vista disso, quanto mais cópias de uma mensagem forem encaminhadas na rede, maior será a probabilidade desses dados serem entregues ao destino, e menor será a latência (BRANCO *et al.* 2010).

Este processo de troca de mensagens chamado de *summary vector* é ilustrado na Figura 5, onde uma lista de mensagens armazenadas é mantida pelos nós, e quando dois nós se encontram, eles trocam suas listas de mensagens armazenadas. Após a troca, cada nó pode determinar se o outro nó tem alguma mensagem que ainda não foi vista nesse nó. Nesse caso, o nó solicita as mensagens (*request messages*) do outro nó. Isso significa que enquanto houver espaço em *buffer* as mensagens são repassadas a cada contato entre os nós (LINDGREN *et al.*).

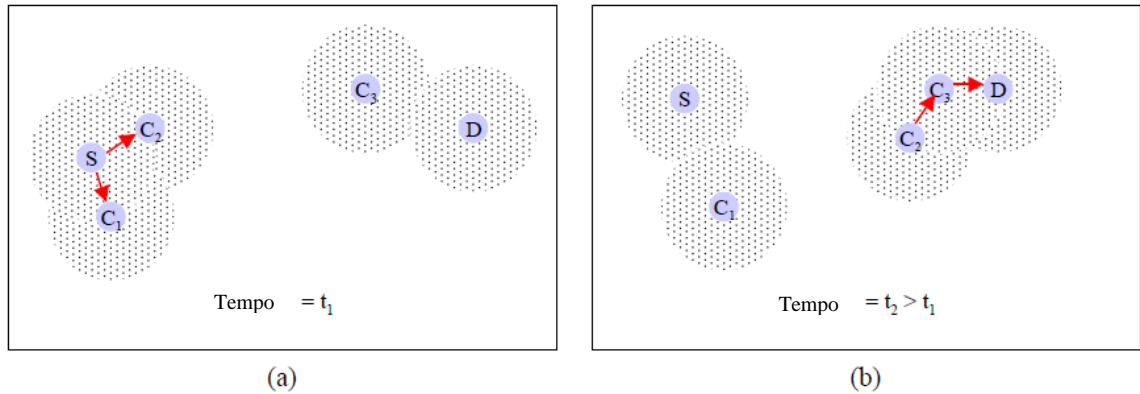
Figura 5 - Troca de mensagens entre dois nós usando protocolo Epidêmico.



Fonte: LINDGREN *et al.* 2003.

Na Figura 6 é mostrado o processo de disseminação de mensagens do Epidêmico, onde no tempo t_1 , a origem S pretende enviar uma mensagem para o destino D e para isso envia essa mensagem para os dois nós que estão dentro do seu alcance de transmissão, sendo eles o nó C_1 e o nó C_2 . No tempo t_2 , o nó C_2 , que recebeu uma cópia da mensagem, entra no raio de transmissão de C_3 e este no raio de D , de modo que uma cópia da mensagem é enviada do primeiro para o segundo e, enfim, para o terceiro, que é o destinatário.

Figura 6 - Exemplo de encaminhamento do protocolo Epidêmico.



Fonte: VAHDAT; BECKER, 2000, p. 3.

O protocolo Epidêmico possui um tempo ótimo de propagação de dados, pois explora todos os caminhos ao mesmo tempo. Além disso, apesar do protocolo de roteamento Epidêmico não detectar falhas, a redundância e aleatoriedade na disseminação de mensagens contornam potenciais falhas de nós ou enlaces. Entretanto, o protocolo Epidêmico gera uma alta quantidade de mensagens e rapidamente ocupa todo o *buffer* dos nós (GUPTA *et al.* 2002 *apud* CORREIA *et al.* 2011).

3.1.1.2 Protocolo Spray and Wait

O protocolo Spray and Wait utiliza um esquema de roteamento baseado em inundação de mensagens, assim como o Epidêmico. Esquemas baseados em inundações têm uma alta probabilidade de entrega, mas consomem muita energia dos dispositivos móveis, além de gerar sobrecarga na rede. Para minimizar esses problemas, o protocolo Spray and Wait reduz e controla a quantidade de cópias das mensagens transmitidas na rede, diminuindo a probabilidade de ocorrência de estouros de buffer e congestionamento.

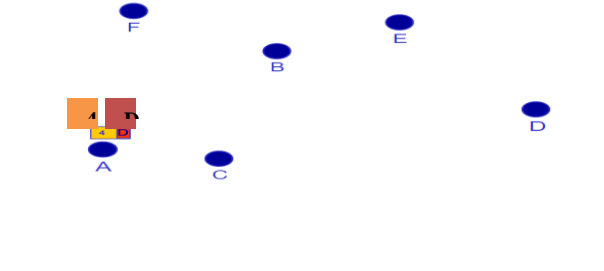
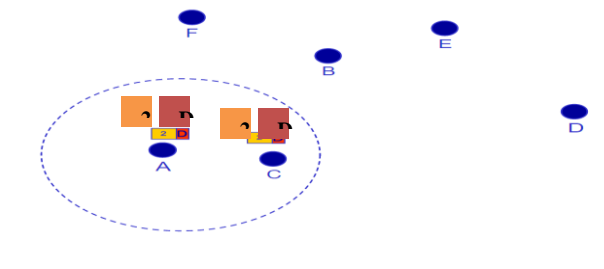
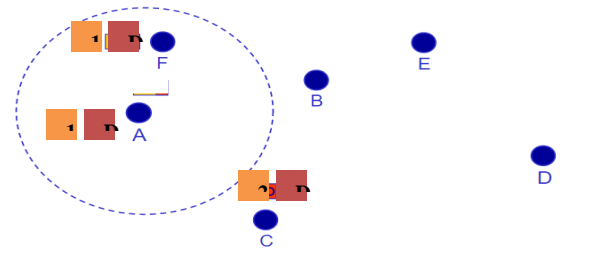
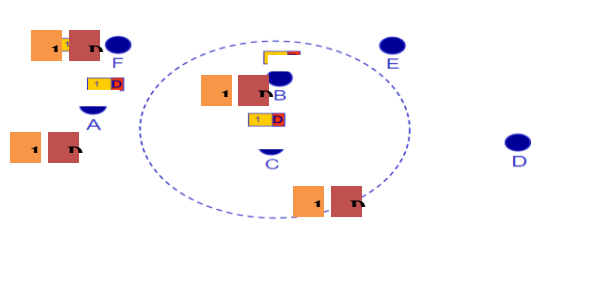
Este esquema de roteamento consiste em duas fases: na primeira fase, chamada *Spray phase*, o protocolo "pulveriza" uma série de cópias na rede e na segunda fase, chamada *Wait phase*, aguarda até que um desses nós encontre o destino.

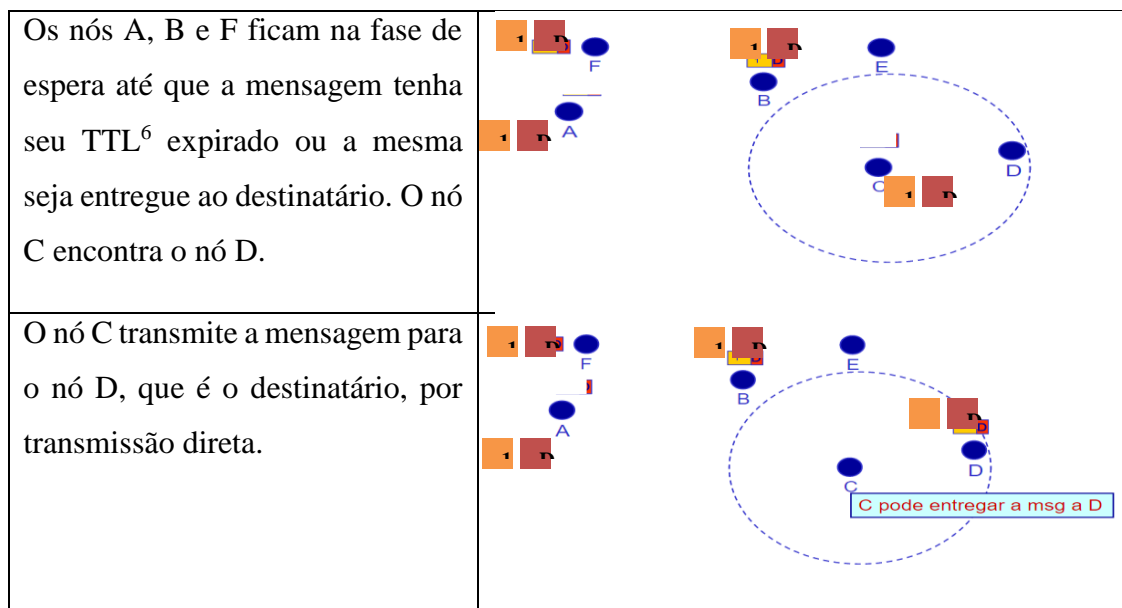
Existem dois principais métodos para a *Spray phase*, o método *source spraying* e o método *binary spraying*. No primeiro método, o nó de origem da mensagem fica responsável por distribuir todas as L cópias; já no segundo método, o nó de origem transmite $L/2$ cópias para o nó seguinte e decrementa esse valor da totalidade das cópias que possui. O nó retransmissor repete esse procedimento até que a quantidade

de L cópias da mensagem no nó seja igual a 1, entrando na segunda fase, a *wait phase*. O método *binary spraying* é considerado mais eficiente, porque usa um número limitado de cópias (SPYROPOULOS *et al* 2005).

O Quadro 3 exemplifica o encaminhamento das cópias através do referido protocolo usando o método *binary spraying*:

Quadro 3 -- Esquema de roteamento Spray and Wait binário.

<p>Nó origem A gera 4 cópias da mensagem ($L=4$), tendo como nó destino D, que está fora do seu alcance.</p>	
<p>O nó A entra em contato com o nó C e transmite metade das cópias que possui, ficando cada nó com $L=2$.</p>	
<p>O nó A encontra o nó F, e transmite metade das cópias que possui, ficando cada nó com $L=1$. Esses nós entram na fase de espera.</p>	
<p>O nó C encontra o nó B, e transmite metade das cópias que possui, ficando cada nó com $L=1$. Esses nós entram na fase de espera.</p>	



Fonte: <http://slideplayer.com.br/slide/1604823/>.

O roteamento utilizando este protocolo necessita de uma rede com alta mobilidade, pois seu desempenho depende da mobilidade dos nós. Em uma rede com baixa taxa de mobilidade, é provável que o nó destino permaneça isolado por muito tempo. (SPYROPOULOS *et al* 2005).

3.1.1.3 Protocolo PRoPHET

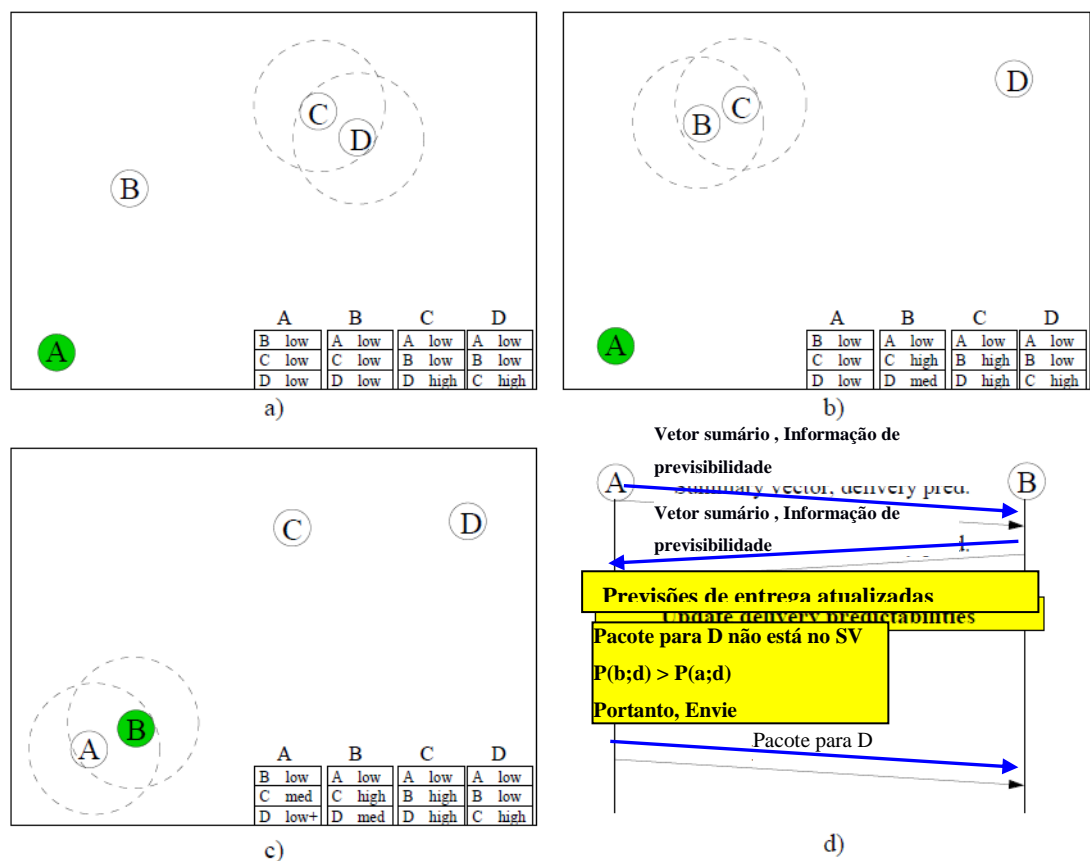
O protocolo de encaminhamento probabilístico (*Probabilistic Routing Protocol using History of Encounters and Transitivity* - PRoPHET), usa uma métrica que indica a probabilidade de um nó entregar uma mensagem a um destinatário, denominada previsibilidade de entrega. Segundo Lindgren (2003), quando dois nós se encontram, uma mensagem é enviada para o outro nó se a previsibilidade de entrega ao destino for mais alta que a do dispositivo origem. Contudo, como o nó que repassou a mensagem pode encontrar um nó melhor ou o próprio destino final no futuro, a mensagem repassada não é removida do nó, mas fica armazenada em buffer, desde que haja espaço disponível. Além disso, uma função de transitividade também é usada.

⁶ TTL (*Time to Live*) é um tempo máximo que a mensagem (pacote) tem de vida na rede. Se expirado, a mensagem é descartada para evitar que fique perdida em *looping* na rede sem achar o seu destino.

O PROPHET é um protocolo classificado como *partially context-aware* e é, provavelmente, o esquema de referência em redes oportunísticas. É também o exemplo mais conhecido dentre os protocolos baseados em histórico de contatos.

A Figura 7 ilustra um exemplo da estratégia de encaminhamento do PROPHET para ajudar a entender a propriedade transitiva e a previsibilidade de entrega. As tabelas de previsibilidade de entrega para os nós são mostradas no canto inferior direito das subfiguras (LINDGREN *et al*, 2003).

Figura 7 - Exemplo do algoritmo de roteamento PROPHET.



Fonte: Lindgren *et al*, 2003.

a) O nó A tem uma mensagem que deseja enviar para o nó D. Suponha que os nós C e D se encontrem com frequência. O contato entre eles faz os valores de previsibilidade de entrega que eles têm entre si alto (*high*) na tabela.

b) Suponha que o nó *C* também encontre frequentemente o nó *B*. Os nós *B* e *C* terão os valores de previsibilidade de entrega que eles têm entre si alto na tabela e a propriedade transitiva também aumentará o valor *B* para *D* para um nível médio (*med*).

c) Finalmente, o nó *B* chega ao alcance de transmissão do nó *A* que possui uma mensagem para o nó *D*.

d) Este quadro mostra a troca de mensagem entre o nó *A* e o nó *B*. Os vetores sumários e a informação de previsibilidade de entrega são trocados, as previsões de entrega são atualizadas e o nó *A* então percebeu que $P(b;d) > P(a;d)$, e assim encaminha a mensagem para *D* através do nó *B*.

O funcionamento subjacente é similar aquele que é proposto pelo protocolo Epidêmico distinguindo-se apenas pela adição de um mecanismo de controle de envio baseado no conceito de probabilidade de entrega. Conceitualmente, durante um encontro, os nós em contato trocam, para além do respectivo vetor sumário, um novo vetor contendo uma probabilidade de entrega para cada um dos nós com os quais já se encontraram. Como resultado disto, os nós apenas obtêm um do outro, as mensagens para os quais possuem uma maior probabilidade de entrega ao destino final. Esta probabilidade de entrega é determinada essencialmente pela frequência de contatos que cada possível nó intermediário tem com o destino final ou com outros nós que eventualmente ofereçam melhores condições de entrega.

3.1.2 Protocolos com base social

Os protocolos baseados em comportamentos sociais tomam decisões de encaminhamento usando padrões (conceitos) observados sobre as ligações entre os nós para prever futuras oportunidades de contato. Nesta abordagem, os protocolos exploram as informações referentes ao comportamento dos nós para decidir se encaminham a mensagem ou armazenam em *buffer* até que seja encontrado um nó com maior probabilidade de entrega da mensagem para o destinatário.

Segundo Hui *et al.* (2008) as relações sociais das pessoas podem variar muito mais lentamente do que a topologia, portanto pode ser usada para melhorar decisões de encaminhamento. No entanto, o comportamento social dos portadores dos dispositivos móveis tem um papel importante na forma em que os nós podem se encontrar. Esta abordagem é aplicada em redes PSN (*Pocket Switched Networks*), um

tipo de rede tolerante à atraso, para permitir que as pessoas se comuniquem na ausência de uma infraestrutura de rede utilizando as oportunidades de contato entre os dispositivos móveis transportados por elas. Para permitir este tipo de comunicação, os protocolos utilizam métricas sociais derivadas dos contatos entre os dispositivos para realizar as tomadas de decisões de encaminhamento das mensagens.

3.1.2.1 Protocolo BUBBLE Rap

O protocolo de roteamento *BUBBLE Rap* (HUI *et al.* 2008) usa métricas sociais para a tomada de decisão em seu algoritmo de encaminhamento de mensagens. Métricas sociais, ou conceitos sociais, são indicadores usados para avaliar o desempenho de um grupo de pessoas em uma rede social.

Dois aspectos importantes que estão presentes em toda sociedade são:

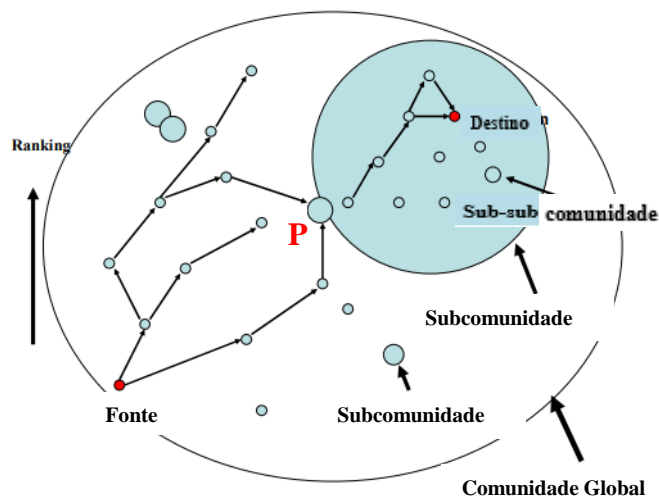
1. Comunidade (*Community*) — a sociedade é dividida estruturalmente em comunidades e subcomunidades.
2. Centralidade (*Centrality*) — dentro dessas comunidades, pessoas interagirão mais que outras, serão mais populares naquele meio. Pessoas com alta centralidade (popularidade) são classificadas como *hubs*, ou pontos centrais.

O algoritmo do *BUBBLE Rap* é baseado nos aspectos de comunidade e centralidade. Uma comunidade é definida como um conjunto de dispositivos densamente conectados, ou seja, um grupo de nós que tem mais contatos entre si do que com outros nós. Cada dispositivo pertence a pelo menos uma comunidade, ou seja, pode existir comunidade de apenas um dispositivo, considerada como pseudo-comunidade de apenas um nó, para fins de encaminhamento de mensagens. A centralidade de um dispositivo é calculada a partir do número de contatos distintos que um determinado nó teve ao longo do tempo, sendo atribuído uma pontuação (*ranking*) para o dispositivo. Essa medida de popularidade é atribuída para cada nó em relação a todos os nós do trace, chamado de *GlobalRank*, e em relação somente aos nós membros de sua comunidade, chamado de *LocalRank*.

A Figura 8 demonstra o processo de encaminhamento da mensagem de um nó origem (*source*) para o nó destino (*destination*). O nó origem transmite a mensagem a partir do grafo de contatos agregados de todo o trace, caso o nó encontrado possua um nível de centralidade maior que o seu dentro da comunidade global (*global*

Community). Este processo continua a cada encontro até um nó intermediário encontrar um nó que pertença a subcomunidade do nó destino (nó "P", na figura). A partir desse momento, o algoritmo *Bubble* (bolha) usa a centralidade de grau local do nó que recebeu a mensagem para transmitir a mensagem novamente, somente dentro da subcomunidade até encontrar o nó destino ou a mensagem expirar. O processo de entregar a mensagem de um nó de menor centralidade para um nó de maior centralidade é entendido como "borbulhar" (*bubble up*) a mensagem. (HUI *et al.* 2008).

Figura 8 - Ilustração do algoritmo de encaminhamento do *BUBBLE*.



Fonte: HUI *et al* 2008 (Adaptado).

Conforme explicado, este esquema de roteamento identifica comunidades sociais densamente interconectadas a partir do grafo de contatos agregados de todo o trace. O encaminhamento das mensagens da origem até o destinatário final se dá de acordo com a seguinte política:

- A cada encontro, um dado nó transmite a mensagem se o nó encontrado tiver um *GlobalRank* (popularidade global) maior que o seu próprio ou até que a mensagem seja alcançada por um membro da comunidade destino (uma das comunidades associadas ao nó destino).
- A partir deste momento, a mensagem é então encaminhada somente entre membros da comunidade destino caso o *LocalRank* (popularidade local) do nó

encontrado seja maior que o do nó que possui a mensagem ou caso o nó encontrado seja o destinatário final da mensagem.

- Uma vez que a mensagem chega a algum nó membro da comunidade a qual o destinatário final pertence, o encaminhamento fora desta comunidade deixa de ocorrer.

A Figura 9 descreve o algoritmo do protocolo em pseudocódigo:

Figura 9 - Algoritmo *BUBBLE* em pseudocódigo.

```

begin
  foreach EncounteredNode_i do
    if ( LabelOf (currentNode) == LabelOf (destination) ) then
      if ( LabelOf (EncounteredNode_i) == LabelOf (destination) )
        and
        ( LocalRankOf (EncounteredNode_i) > (LocalRankOf (currentNode) )
      then
        EncounteredNode_i.addMessageToBuffer ( message )
      else
        if ( LabelOf (EncounteredNode_i) == LabelOf (destination) )
          or
          ( GlobalRankOf (EncounteredNode_i) > GlobalRankOf (currentNode) )
        then
          EncounteredNode_i.addMessageToBuffer (message)
    end
  end

```

Fonte: HUI *et al* 2008.

O *BUBBLE Rap* usa os algoritmos de K-clique e *cumulative Windows* (C-Window) para identificar as comunidades (*community detection*) que os nós pertencem (quando o nó intermediário pertence a mesma comunidade do destino) e determinar as centralidades local e global dos nós (quando o nó intermediário tem centralidade maior que o nó que detém a mensagem). Com base nestas informações de comunidades e centralidades, o algoritmo decide quando criar uma cópia da mensagem.

3.1.2.2 Protocolo SOCLEER

Segundo Machado (2013), o protocolo SOCLEER (*Social-based Energy-Efficient Routing Protocol*) visa aliviar a carga dos nós mais requisitados, através de uma alteração do decisor de roteamento do consagrado protocolo oportunista com base social, o *BUBBLE Rap*, no qual foi aplicado um novo mecanismo baseado em um sorteio simples para escolha dos nós retransmissores da mensagem. A proposta deste

protocolo foi distribuir a carga de participação dos nós preferidos no envio de mensagens com outros de características sociais semelhantes, bem como reduzir seu consumo de energia.

Ainda segundo Machado (2013), o novo decisor, que foi implementado por meio de um sorteio simples de n nós dentre os melhores elencados em uma lista ordenada de maiores centralidades para efeito de encaminhamento de mensagens, reduziu o consumo de energia dos nós quando comparado com o *BUBBLE Rap*.

De acordo com Machado *et al* (2014), o mecanismo decisor do SOCLEER consiste no seguinte:

Ao invés da mensagem ser enviada imediatamente do nó atual ao nó conectado que possua maior centralidade do que a sua, durante o tempo t em que o grafo de contatos é formado, como faz o *BUBBLE Rap*, as centralidades são calculadas e armazenadas em uma lista de centralidades. Isso vale tanto para a centralidade global, quanto para a local, que continuam sendo utilizadas da mesma forma que o algoritmo original. Então, essa lista é ordenada decrescentemente do nó com maior centralidade para o nó de menor centralidade, onde então um método criado para efetuar o sorteio é invocado. Este método efetua um sorteio de n nós, sendo o mínimo de 1 e o máximo de 10, dentre os maiores em centralidade contidos na lista para encaminhar a mensagem (MACHADO, 2014, p. 5 e 6).

O objetivo do sorteio é reduzir as réplicas de mensagens disseminadas na rede e desse modo reduzir a carga de participação dos nós no envio de mensagens, pois somente os nós sorteados de acordo com a regra poderão retransmitir. Dessa forma, o sorteio possibilita que um nó mais conectado não seja sempre o mais escolhido para se encaminhar a mensagem. (MACHADO *et al*, 2014).

Machado (2013), em sua Dissertação, relata que este novo protocolo consegue obter melhores resultados em redes com alta densidade de conexões e quando há aumento na carga de mensagens, chegando a superar o *BUBBLE Rap* em tais cenários simulados. A Figura 10 ilustra o pseudocódigo do protocolo SOCLEER.

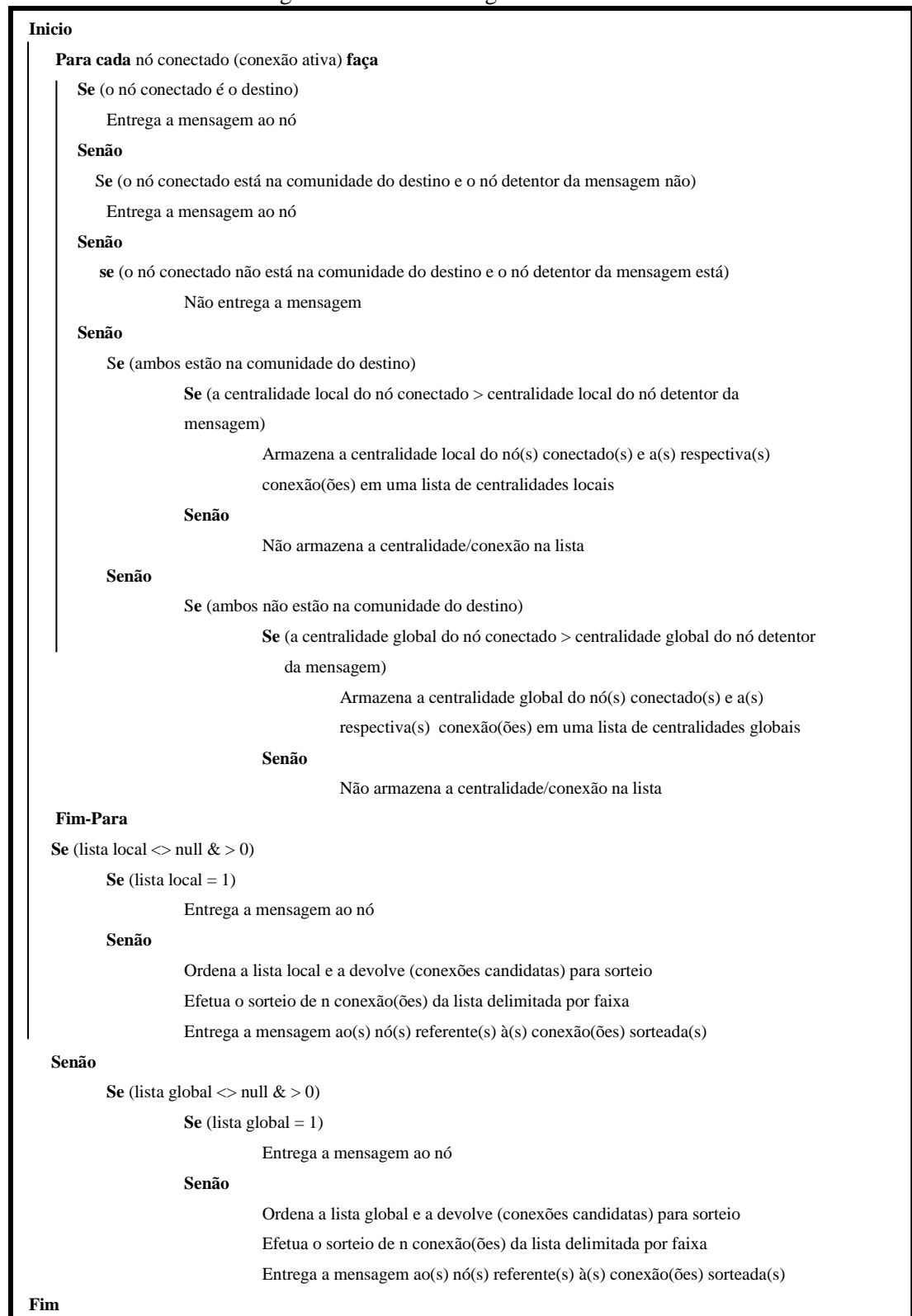
O SOCLEER utiliza os algoritmos K-clique para detecção de comunidades e o algoritmo C-Window para o cálculo das centralidades, assim como no *BUBBLE Rap* (MACHADO, 2013).

3.2 O Problema dos Nós Preferidos

As abordagens mais recentes em redes DTN são os protocolos baseados em contexto social, no entanto esta abordagem tem uma questão que ainda não possui uma solução ideal, que é o problema da frequência de utilização dos nós preferidos.

A solução para a questão dos nós preferidos em protocolos que exploram o contexto social para indicar o melhor caminho entre a origem e o destino de uma mensagem tem sido utilizada na proposta de novos protocolos de roteamento oportunistas. O uso constante de repetidos nós tidos como os mais propensos a entregar a mensagem gera sobrecarga destes, consumindo mais rápido o limitado recurso de bateria destes dispositivos móveis (MACHADO, 2013).

Figura 10 - Pseudocódigo do SOCLEER.



Fonte: Machado 2013.

4 Trabalhos Relacionados

Neste capítulo serão apresentadas as principais métricas para mensurar propriedades sociais identificadas nas relações humanas, também será apresentado o simulador ONE e algumas de suas funcionalidades.

4.1 Classificação das métricas para redes sociais

Os protocolos com base social fazem uso do comportamento humano como fator de decisão na escolha da melhor rota para a entrega de uma mensagem. Entre as métricas sociais, as medidas de comunidade e centralidade são as principais métricas utilizadas para mensurar propriedades sociais no projeto destes protocolos (MACHADO, 2013).

4.1.1 Comunidade (*Community*)

Uma Comunidade é definida como um subconjunto de nós com conexões mais fortes entre eles do que com outros nós (MACHADO, 2013). Segundo Machado (2013), a existência de fortes comunidades no grafo de contato tem diversas implicações para as redes oportunistas. Por um lado, implica um alto potencial para mecanismos de cooperação de nós e de confiança baseados na comunidade. Por outro, pode também implicar elevados tempos de convergência para algoritmos distribuídos, uma vez que podem haver gargalos fortes entre as comunidades.

4.1.2 Centralidade do nó (*Node Centrality*)

O grau de centralidade é uma medida que verifica o quão importante é um determinado nó para a rede. Entende-se que o nó central é aquele que possui uma posição mais privilegiada e cuja eliminação causaria uma grande desestabilização na rede (RECUERO, 2009).

4.1.2.1 Centralidade de grau ou grau de conexão (*Degree Centrality*)

O grau de conexão é a quantidade de adjacências (conexões) que um determinado nó possui em uma rede. Nós adjacentes são nós conectados entre si e os nós aos quais um determinado nó é adjacente são denominados vizinhança. Quanto maior o grau de conexão, mais popular e mais central é o nó na rede (RECUERO, 2009).

A fórmula da *Degree Centrality* (C_D) é definida por (FREEMAN, 1979):

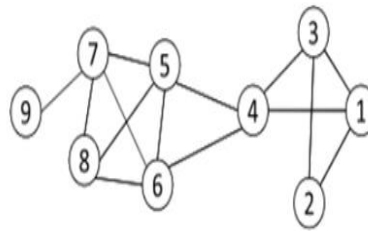
$$C_D(v_k) = \sum_{i=1}^n a(v_i, v_k)$$

Onde: n = número de nós em uma rede

$a(v_i, v_k) = 1$ se e somente se v_i e v_k estão conectados por uma aresta (adjacentes) e $a(v_i, v_k) = 0$ caso contrário.

O grafo na Figura 11 ilustra um exemplo da métrica *Degree Centrality*. Para o nó 1, a centralidade do grau é 3.

Figura 11 - Exemplo da métrica *Degree Centrality*



Fonte: Disponível em <<http://slideplayer.com/slide/7841792/>>. (Adaptado)

4.1.2.2 Centralidade de proximidade (*Closeness Centrality*)

A centralidade do tipo grau de proximidade pode ser compreendida a partir dos graus de distância entre os nós. A partir desta perspectiva, a soma das distâncias geodésicas⁷ entre todos os outros nós do grafo em relação a um específico seria o grau de proximidade entre eles. Essa medida é interessante para redes bastantes conectadas e para a identificação de grupos sociais mais fechados (RECUERO, 2009).

A fórmula da *Closeness Centrality* (C_C) de um nó n_i a um nó n_j é definida por (FREEMAN, 1979):

$$C_C(n_i) = \left[\frac{\sum_{j=1}^g d(n_i, n_j)}{n - 1} \right]^{-1}$$

Onde: $d(n_1, n_j)$ = distância geodésica entre dois nós, e

⁷ A distância geodésica entre dois nós é a menor distância possível.

g = número de nós em uma rede

O grafo na Figura 12 ilustra um exemplo da métrica *Closeness Centrality* e a Tabela 1 mostra a distância geodésica entre os nós da rede. Neste grafo, o nó 4 é mais central que o nó 3.

Figura 12 - Exemplo da métrica *Closeness Centrality*.

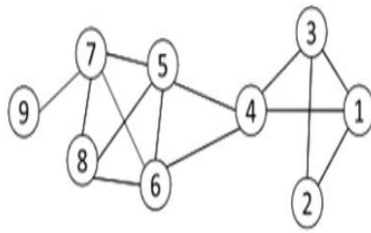


Tabela 1 - Distância geodésica entre os nós da rede.

Nó	1	2	3	4	5	6	7	8	9
1	0	1	1	1	2	2	3	3	4
2	1	0	1	2	3	3	4	4	5
3	1	1	0	1	2	2	3	3	4
4	1	2	1	0	1	1	2	2	3
5	2	3	2	1	0	1	1	1	2
6	2	3	2	1	1	0	1	1	2
7	3	4	3	2	1	1	0	1	1
8	3	4	3	2	1	1	1	0	2
9	4	5	4	3	2	2	1	2	0

Fonte: Disponível em <<http://slideplayer.com/slide/7841792/>>. (Adaptado)

$$C_c(3) = \frac{9 - 1}{1 + 1 + 1 + 2 + 2 + 2 + 3 + 3 + 4} = \frac{8}{17} = 0.47$$

$$C_c(4) = \frac{9 - 1}{1 + 2 + 1 + 1 + 1 + 1 + 2 + 2 + 3} = \frac{8}{13} = 0.62$$

4.1.2.3 Centralidade de intermediação (*Betweness Centrality*)

O grau de intermediação é a terceira forma de ver a centralidade (RECUERO *apud* FREEMAN, 2009). Esse grau é uma medida do quanto um nó possui valor de intermediação em um grafo, ou seja, o quanto ele aparece em "meio" a outros (RECUERO *apud* SCOTT, 2009). Esta medida indica o quanto um nó é essencial para que uma determinada informação circule na rede (quanto maior o grau, mais central o nó está na rede). O grau de intermediação é medido a partir da proporção de geodésicas

que conectam cada par de nós da rede e que passam pelo nó analisado (RECUERO, 2009).

A fórmula da *Betweness Centrality* (C_B) entre um nó n_i e um nó n_j é definida por (FREEMAN, 1979):

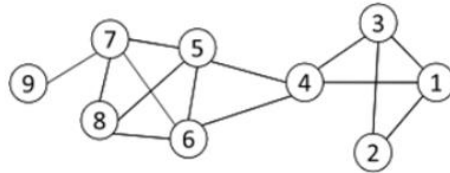
$$C_B(n_i) = \sum_{j < k} \frac{g_{jk}(n_i)}{g_{jk}}$$

Onde: g_{jk} = número total de caminhos geodésicos conectando j e k , e

$g_{jk}(n_i)$ = número de caminhos geodésicos que inclui n_i

O grafo na Figura 13 ilustra um exemplo da métrica *Betweness Centrality* e a Tabela 2 mostra o número de caminhos mais curtos entre um nó j e k que passa pelo nó n_i dividido pelo número de caminhos mais curtos entre j e k . Considere n_i o nó 4.

Figura 13 - Exemplo da métrica *Betweness Centrality*.



$$C_B(4) = 15$$

$$C_B(n_i) = \sum_{k \neq n_i \neq j \in V, k < j} \frac{g_{jk}(n_i)}{g_{jk}}$$

Tabela 2 - Cálculo da métrica *Betweness Centrality*.

$\frac{g_{jk}(n_i)}{g_{jk}}$	k=1	k=2	k=3
J=5	1/1	2/2	1/1
J=6	1/1	2/2	1/1
J=7	2/2	4/4	2/2
J=8	2/2	4/4	2/2
J=9	2/2	4/4	2/2

Fonte: Disponível em <<http://slideplayer.com/slide/7841792/>>. (Adaptado)

4.2 O simulador para ambiente de rede oportunista

Para simular a utilização dos protocolos de roteamento, foi utilizado o simulador para ambiente de rede oportunista, ONE⁸ (*The Opportunistic Network Environment*

⁸ <https://akeranen.github.io/the-one/>

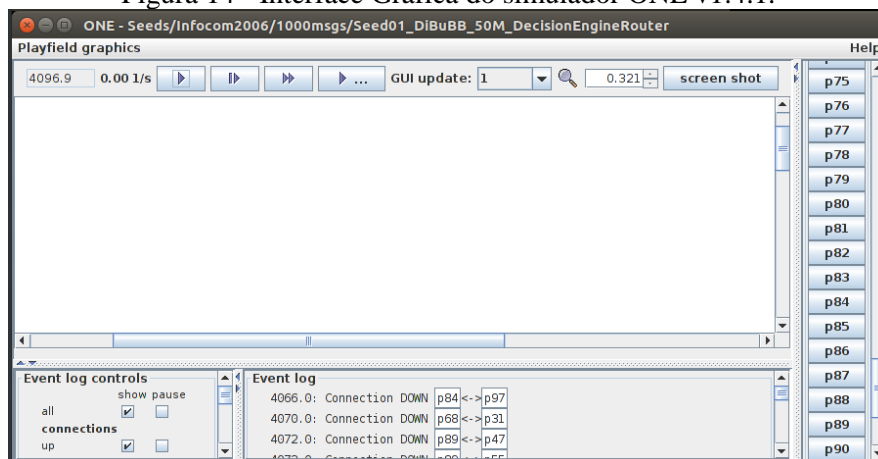
simulator). O ONE é um simulador desenvolvido em Java (1.6) por KERÄNEN *et al* (2009) na Universidade de Aalto⁹ (Finlândia) e apoiado pelo *Nokia Research Center*, em Cambridge (Reino Unido). Foi disponibilizado para uso sob uma licença GPL e está, atualmente, na versão 1.6.0 (ONE, 2017a).

O ONE é utilizado pela comunidade acadêmica internacional para realizar comparações e avaliações de performance de novos protocolos em redes oportunistas com os principais protocolos que já se encontram implementados nele. Muitas pesquisas em cenários simulados têm obtidos resultados através dele (MACHADO *et al*, 2014).

O simulador pode ser usado em ambientes Unix/Linux e Windows com o Java 6 JDK ou posterior previamente instalado. O ONE pode ser compilado do código-fonte usando o *script compile.bat* incluído nos arquivos de instalação (ONE, 2017b).

A interface gráfica do usuário (GUI) exibe uma visualização do estado de simulação mostrando os locais, os contatos ativos e as trocas de mensagens realizadas pelos nós (KERÄNEN *et al*, 2009). A Figura 14 ilustra a GUI do simulador.

Figura 14 - Interface Gráfica do simulador ONE v1.4.1.



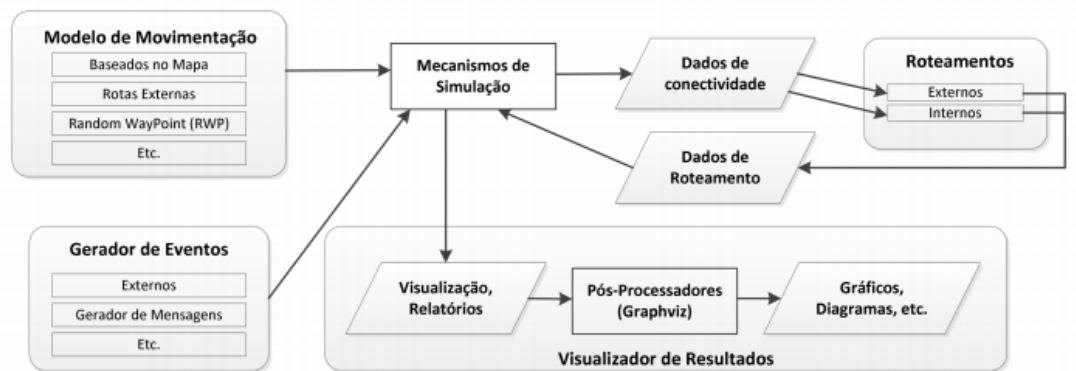
4.2.1 Principais funções do simulador ONE

Conforme ilustrado na Figura 15, o simulador é dividido em módulos. O ambiente de simulação do ONE é capaz de gerar os movimentos dos nós usando diferentes modelos

⁹ <http://www.aalto.fi/en/>

de movimentos, rotear mensagens entre os nós com vários algoritmos de roteamento DTN, visualizar a mobilidade e a entrega de mensagens em tempo real na interface gráfica do usuário e importar dados de mobilidade de *traces* reais ou de outros geradores de mobilidade. Também pode produzir uma variedade de relatórios do movimento do nó para a entrega de mensagens e estatísticas gerais. (ONE, 2017a).

Figura 15 - Estrutura de funcionamento do Simulador ONE.



Fonte: Adaptado de KERÄNEN *et al.* (2009).

4.2.2 Configuração do simulador

Para configurar o ambiente de rede e os protocolos de encaminhamento no ONE, é preciso editar os parâmetros de simulação no arquivo de configuração (*default_settings*), que é um arquivo de texto que contém os pares chave-valor. A sintaxe para a maioria das variáveis é: *Namespace.key = value*. Ou seja, a chave é (geralmente) prefixada por um espaço para nome, seguido de um ponto e, em seguida, do nome da chave. A chave e o valor são separados pelo sinal igual. O *Namespace* define (vagamente) a parte do ambiente de simulação em que a configuração tem efeito. Alguns *Namespace* são iguais ao nome da classe onde são lidos. Especialmente modelos de movimento, módulos de relatório e módulos de roteamento seguem esta convenção. Em alguns casos, o *Namespace* é definido pelo usuário. O arquivo "*default_settings.txt*" é sempre lido pelo programa, mas outros arquivos de configurações podem ser fornecidos como parâmetros. A ideia é que o usuário possa definir no arquivo "*default_settings.txt*" todas as configurações comuns para todas as simulações e executar simulações diferentes, específicas, usando diferentes arquivos de configuração (ONE, 2017b).

No ONE, por padrão, já estão implementados alguns dos protocolos utilizados neste trabalho, como o Epidêmico e o PRoPHET. Os protocolos *BUBBLE Rap* e a sua modificação, o *SOCLEER*, foram configurados manualmente. Os protocolos de roteamento podem ser aplicados de maneira global ou local no ambiente de simulação, ou seja, cada grupo distinto de nós pode possuir seu próprio protocolo de roteamento. Porém, neste trabalho, todos os nós executaram o mesmo protocolo em cada simulação para fins de comparação das métricas.

4.2.3 Módulo de Relatórios

Os resultados da simulação são coletados principalmente através de relatórios gerados por módulos de relatório (*report modules*) durante a execução da simulação. Os módulos de relatórios recebem eventos (por exemplo, troca de mensagens ou conectividade entre os nós) a partir do "motor" (*engine*) da simulação e geram resultados com base neles. Os resultados gerados podem ser registros (*logs*) de eventos que são posteriormente processados por ferramentas externas de pós-processamento, ou podem ser estatísticas agregadas calculado no simulador. (KERÄNEN *et al*, 2009).

O relatório padrão pré-configurado no simulador é o *MessageStatsReport*, que gera diferentes tipos de estatísticas totais sobre as mensagens, a saber: Probabilidade de Entrega das Mensagens (*delivery_prob*), Tempo Médio de Atraso na Entrega das Mensagens (*latency_avg*), Total de Transmissão de Mensagens Iniciadas (*started*), Total de Transmissão de Mensagens Abortadas (*aborted*), Número Médio de Saltos das Mensagens (*hopcount_avg*), Taxa de sobrecarga (*overhead_ratio*), entre outras. Se alguma estatística não puder ser criada, ela recebe o valor "NaN".

O simulador foi configurado para gerar outros arquivos de relatórios com os resultados baseados nos eventos da simulação para serem tratados por ferramentas de pós-processamento e posterior construção dos gráficos, a saber:

- *AdjacencyGraphvizReport*: Gera o gráfico compatível com a ferramenta *Graphviz* a partir das conexões.
- *DeliveredMessagesReport*: Informa sobre todas as mensagens entregues. Mostra a rota que a mensagem fez da origem até o nó destino ou se a mensagem foi descartada.

- *EnergyLevelReport*: Relatório de nível de energia do nó. Mostra o nível de energia por granularidade de tempo em segundos.

Os dois próximos relatórios geram dados somente para os protocolos baseados em contexto-social:

- *CommunityDetectionReport*¹⁰: Informa as comunidades locais em cada nó para os algoritmos de roteamento que utilizam o *DecisionEngineRouter* como roteador e cujo *RoutingDecisionEngine* implementa o *routing.community.CommunityDetectionEngine*. Desta forma, o relatório é capaz de produzir o resultado de qualquer algoritmo de detecção de comunidades.
- *DeliveryCentralityReport*¹¹: Informa a centralidade de entrega de cada nó na simulação, onde a centralidade de entrega é definida como um número inteiro contando o número de vezes em que um dado nó está no caminho mais curto entre dois outros nós comunicantes. Para cada mensagem entregue (na sua primeira entrega), o contador é incrementado para cada nó na lista de saltos (*hops*) da mensagem. O relatório gera uma lista onde cada linha identifica um nó seguido do contador de quantas vezes ele atuou como um nó intermediário no caminho para entregar a mensagem.

4.3 Ferramenta externa de pós-processamento

4.3.1 Gnuplot

Conforme descrito anteriormente, após as simulações, os módulos de relatório do ONE geram os resultados em forma de arquivos de texto (*reports*). Foram utilizados *scripts* em linguagem AWK, que contém os comandos para tratar os *reports* gerados pela simulação. Os arquivos que foram gerados pelos *scripts* foram usados como entrada na ferramenta Gnuplot¹² para a plotagem dos gráficos.

¹⁰ author PJ Dillon, University of Pittsburgh

¹¹ author PJ Dillon, University of Pittsburgh

¹² <http://www.gnuplot.info/>

4.3.2 Graphviz

A ferramenta Graphviz¹³ foi usada para construir o grafo de contatos das conexões a partir do relatório "*AdjacencyGraphvizReport*" gerado pela simulação. O grafo de contatos do *trace* INFOCOM2006 gerado por este *software* será apresentado no capítulo 5.

¹³ <http://www.graphviz.org/>

5 Parâmetros Gerais da Simulação

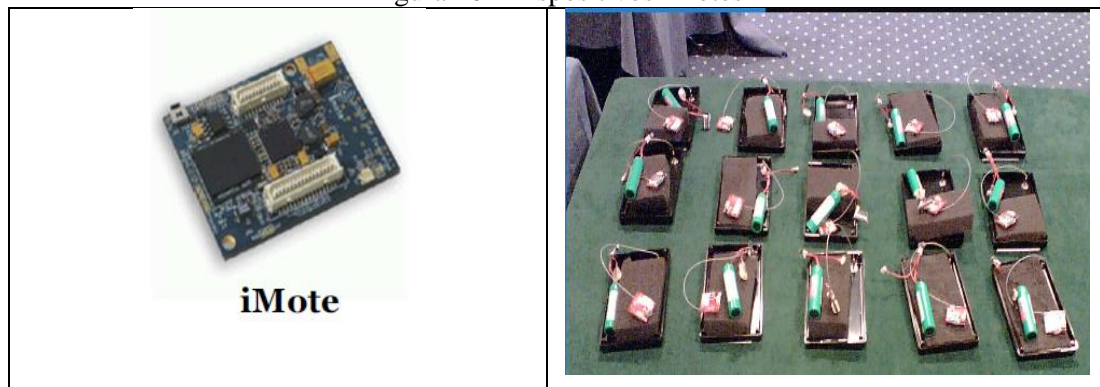
Neste capítulo será apresentado uma visão geral do trace de mobilidade real utilizado na simulação, os principais parâmetros de configuração do simulador ONE e as métricas de avaliação utilizadas.

5.1 Visão geral

O cenário proposto para este experimento consiste em usar um *dataset* denominado INFOCOM2006 como cenário real de mobilidade disponibilizado pela CRAWDAD, uma base pública de dados para futuras pesquisas em mobilidade.

Essa coleta de dados (experimento 6) foi realizada na edição de 2006, em Barcelona, na 25ª Conferência IEEE INFOCOM'06¹⁴. Foram distribuídos 98 pequenos dispositivos com interface de comunicação sem fio *Bluetooth*, denominado iMote (Figura 16), a estudantes presentes no evento para a realização do *trace* de contato e geração do tráfego de dados. O *trace* teve uma duração de 3,9 dias com 20 nós estáticos e 78 voluntários usando iMotes.

Figura 16 - Dispositivos iMotes



Fonte: Arquivo *Exp6.tar.gz*. Disponível em:

<https://crawdad.org/cambridge/haggle/20090529/imote/>.

Esses dados foram convertidos no formato *StandardEventsReader*, através de um *script* escrito na linguagem Python denominado "*generate_haggle_one_infocom*

¹⁴ <http://infocom2006.ieee-infocom.org/>

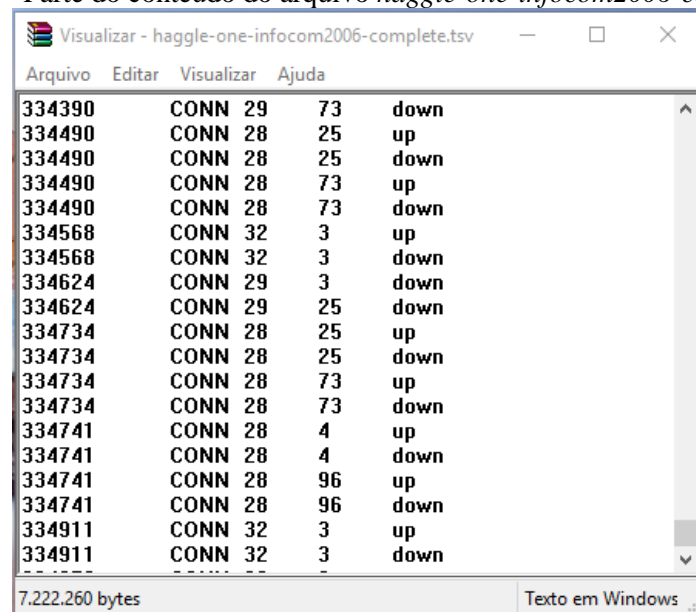
2006_complete.py" e estão disponíveis para uso no simulador ONE na base de dados CRAWDAD. Este *script* requer como entrada o caminho para o arquivo "*Exp6.tar.gz*"¹⁵, que faz parte do conjunto de dados *cambridge / haggles* (v. 2009-05-29), para gerar um *trace* de conectividade que pode ser processado pelo simulador ONE (AKESTORIDIS, 2016).

Os eventos de conexão foram então classificados de acordo com o tempo de simulação. Foi usada uma lista ordenada de eventos de conexão para criar o arquivo "*haggles-one-infocom2006-complete.tsv*", que armazena valores separados por tabulação para cada evento de conexão. Cada linha desse arquivo descreve um evento de conexão e tem os seguintes cinco campos:

[time] [action] [first_node] [second_node] [type].

A Figura 17 ilustra uma parte desse arquivo:

Figura 17 - Parte do conteúdo do arquivo *haggles-one-infocom2006-complete.tsv*



334390	CONN	29	73	down
334490	CONN	28	25	up
334490	CONN	28	25	down
334490	CONN	28	73	up
334490	CONN	28	73	down
334568	CONN	32	3	up
334568	CONN	32	3	down
334624	CONN	29	3	down
334624	CONN	29	25	down
334734	CONN	28	25	up
334734	CONN	28	25	down
334734	CONN	28	73	up
334734	CONN	28	73	down
334741	CONN	28	4	up
334741	CONN	28	4	down
334741	CONN	28	96	up
334741	CONN	28	96	down
334911	CONN	32	3	up
334911	CONN	32	3	down
---	---	---	---	---

Fonte: Arquivo *haggles-one-infocom2006-complete.tsv*. Disponível em:

< <https://crawdad.org/uoi/haggles/20160828/one/> >.

O primeiro campo corresponde ao tempo de simulação no qual o evento ocorreu. O segundo campo é sempre igual a "CONN", pois todos os eventos no *trace* de

¹⁵ <https://crawdad.org/cambridge/haggles/20090529/imote/>

conectividade são eventos de conexão (*connection-up*) ou desconexão (*connection-down*). Os valores dos terceiro e quarto campos correspondem aos IDs de dois nós. O quinto campo é "*up*" quando dois nós se conectam uns com os outros ou "*down*" quando dois nós se desconectam (AKESTORIDIS, 2016).

As principais características¹⁶ desse trace de conectividade podem ser resumidas da seguinte forma:

- Número de nós: 98
- Número de contatos: 170.601
- Duração: 335.600 segundos, ou aproximadamente 3,91 dias.
- Período: Segunda, 24 de abril até Quinta, 27 de abril de 2006.
- Os nós com ID #1 até #17 são iMotes estáticos de longo alcance implantados em toda a área do hotel.
- Os três nós com ID #18, #19 e #20 são iMotes de longo alcance que foram colocados no elevador do hotel.
- Os nós com ID #21 até #98 são participantes da oficina de estudantes da Infocom.
- Os nós com ID maiores ou iguais a #100 são dispositivos externos.

Os 20 iMotes estacionários (de longo alcance) possuem bateria mais potente e faixa de rádio estendida (cerca de 100 metros). Os 78 iMotes móveis têm uma faixa sem fio em torno de 30 metros. A Figura 18 apresenta o grafo de contatos do *trace* INFOCOM2006 para o tempo total de simulação do cenário. Observa-se que o grafo possui muitas conexões entre os nós, o que demonstra a alta densidade desse *trace*.

¹⁶ Fonte: README for the Haggle data sets collected at Infocom 2006.

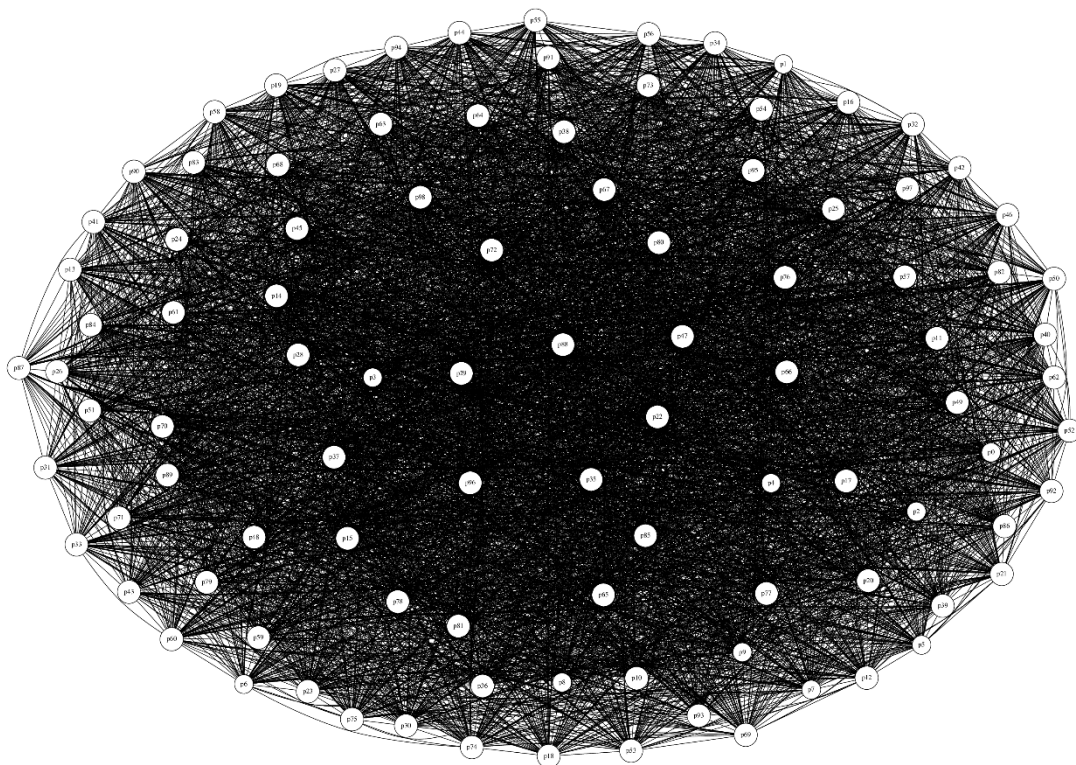


Figura 18 – Grafo de contatos do trace INFOCOM2006

5.2 Descrição dos parâmetros gerais da simulação

O simulador ONE foi configurado para rodar o *trace* de mobilidade real, INFOCOM2006, com os seguintes protocolos de roteamento oportunistas: Epidêmico, PRoPHET, Bubble Rap e sua modificação, o SOCLEER. As simulações foram feitas na interface de linha de comando (CLI) do Linux. A simulação de cada um dos protocolos foi realizada 10 vezes considerando cada carga de mensagens, isto se deve ao fato que para se obter dados mais próximos da realidade é necessário realizar um tratamento estatístico destes dados. Para o tratamento estatístico, foi usado a distribuição *T-Student* para calcular o intervalo de confiança para um nível de significância de 95%. Os parâmetros gerais de configuração do simulador ONE para esse *trace* estão mostrados na Tabela 3.

Tabela 3 - Parâmetros gerais da simulação.

Parâmetro	Chave	Valor
Tempo total da simulação (segundos)	Scenario.endTime	335600
Conexões (arquivos de eventos)	Scenario.simulateConnections	false
Tipo de interface de rede (bluetooth)	btInterface.type	SimpleBroadcastInterface
Velocidade de transmissão (kBytes)	btInterface.transmitSpeed	250k
Alcance da comunicação (m).	btInterface.transmitRange	10
Número de tipos diferentes de hosts	Scenario.nrofHostGroups	1
Modelo de movimento do grupo	Group.movementModel	StationaryMovement
Configuração obrigatória	Group.nodeLocation	10,10
Protocolo de roteamento	Group.router	DecisionEngineRouter, EpidemicRouter, ProphetRouter
Velocidades mín. e máx. (m / s)	Group.speed	0.5, 1.5
Tamanho do <i>buffer</i> (bytes)	Group.bufferSize	50M, 250M, 500M, 1G e 2G
TTL (segundos)	Group.msgTtl	360
Número de hosts do cenário	Group.nrofHosts	98
Número de eventos	Events.nrof	2
Uso de <i>trace</i> real de mobilidade.	Events1.class	ExternalEventsQueue
Caminho do(s) arquivo(s) de eventos	Events1.filePath	GeneratedFile.txt
Uso de <i>trace</i> real de mobilidade.	Events2.class	ExternalEventsQueue
Caminho do arquivo do <i>trace</i>	Events2.filePath	/mytracefile.txt
Tamanho da mensagem (kB)	Events1.size	50k
Carga inicial da bateria (mAh).	Group.intialEnergy	2300
mAh por varrer (<i>scan</i>) a rede	Group.scanEnergy	0.44
mAh por segundo ao enviar dados	Group.transmitEnergy	0.038
mAh por segundo ao receber dados	Group.receiveEnergy	0.038
Não houve recarga de energia	Group.rechargeEnergy	335600
Número de tipos de relatórios.	Report.nrofReports	6
Granularidade dos relatórios (min)	Report.granularity	900

Para gerar as cargas de mensagens no simulador ONE, foi implementado o parâmetro semente (*seed*). Esse parâmetro é usado para gerar eventos de mensagem

distintos. Toda vez que o mesmo *seed* é usado, eventos idênticos serão gerados. Para usar os *seeds* gerados, foi configurado a classe *ExternalEventsQueue* para ler o(s) arquivo(s) de geração de cargas de mensagens.

5.3 Métricas de Avaliação de desempenho

Com o objetivo de analisar comparativamente o desempenho do protocolo SOCLEER com relação aos outros protocolos, propõe-se a utilização das seguintes métricas de redes oportunistas disponibilizadas pelo simulador (MACHADO, 2013):

- Fração de Mensagens Entregues (*Fme*): É a quantidade de mensagens entregues (*Qme*) dividido pela quantidade de mensagens criadas na origem (*Qmc*).

$$Fme = Qme / Qmc$$

- (O): É a diferença entre a quantidade de mensagens retransmitidas (*Qmr*) e a quantidade de mensagens entregues (*Qme*) dividido pela quantidade de mensagens entregues (*Qme*).

$$O = (Qmr - Qme) / Qme$$

- Latência (L): É o tempo que a mensagem leva desde sua criação até a chegada ao destino.

$$L = \Delta t(m_m - m_{t_0})$$

- Número médio de saltos: É o número médio de saltos realizados pelas mensagens desde a origem até o destino.
- Participação dos nós no envio de mensagens: É a quantidade média de vezes em que um nó é requisitado para encaminhar uma mensagem (somente as mensagens entregues foram incluídas nesta métrica).
- Consumo de bateria dos dispositivos móveis: É a quantidade média de energia gasta pelos nós durante o envio e recepção de mensagens e a cada *scan* na rede.

6 Avaliação de Desempenho

Neste capítulo serão apresentados os resultados das simulações com uma análise, de acordo com o cenário proposto, o trace real de mobilidade, e as métricas de desempenho utilizadas.

6.1 Cenários de simulações para o SOCLEER.

Para os protocolos baseados em contexto social, *BUBBLE Rap* e SOCLEER, definiu-se nas simulações o valor 5 como o número máximo de distribuições de mensagens (k), este valor foi configurado na chave *DecisionEngineRouter.K* no arquivo de configuração do ONE. No cálculo das centralidades através do C-Window foi definido, em ambos os protocolos, o tempo de 10 minutos como janela de tempo de espera antes de se recalcular novos valores.

A Tabela 4 contém os parâmetros de simulação para o protocolo SOCLEER com o algoritmo de detecção de comunidade *KClique community detection* e o algoritmo de centralidade *CWindow centrality*.

Tabela 4 - Valores dos parâmetros usados para o protocolo SOCCLER.

Chave	Valor
Group.router	DecisionEngineRouter
DecisionEngineRouter.decisionEngine	community.Socleer
DecisionEngineRouter.communityDetectAlg	routing.community.KCliqueCommunity Detection
DecisionEngineRouter.K	5
DecisionEngineRouter.familiarThreshold	700
DecisionEngineRouter.centralityAlg	routing.community.CWindowCentrality
DecisionEngineRouter.numberOfElementsToChoose	2, 3, 5, todos
DecisionEngineRouter.quantityOfElementsToChoose	1, 2, 3, 5, 10

A Tabela 5 contém os parâmetros de simulação para o protocolo *BUBBLE Rap* com o algoritmo de detecção de comunidade *KClique community detection* e o algoritmo de centralidade *CWindow centrality*.

Tabela 5 - Valores dos parâmetros usados para o protocolo *BUBBLE Rap*.

Chave	Valor
Group.router	DecisionEngineRouter
DecisionEngineRouter.decisionEngine	community.DistributedBubbleRap
DecisionEngineRouter.communityDetectAlg	routing.community.KCliqueCommunity Detection
DecisionEngineRouter.K	5
DecisionEngineRouter.familiarThreshold	700
DecisionEngineRouter.centralidadeAlg	routing.community.CWindowCentrality

Na construção dessa fase do estudo foram utilizadas cinco cargas de mensagens para o *trace* de mobilidade INFOCOM 2006 com o objetivo de analisar o desempenho do protocolo SOCLEER em comparação com outros protocolos de roteamento oportunistas. As cargas geradas foram de 1000, 5000, 10000, 20000 e 30000 mensagens. Essas diferentes cargas de mensagens tem o objetivo de verificar o comportamento dos protocolos em cenários distintos de trocas de mensagens em uma DTN.

Para a carga inicial de 1000 mensagens, foram utilizados 6 cenários de sorteio para o protocolo SOCLEER com o intuito de verificar o comportamento do seu algoritmo decisor. Desse modo, foi possível testar a eficiência do método de sorteio simples de nós como intermediários no envio da mensagem dentre os de maiores centralidades guardadas, considerando as métricas avaliadas (MACHADO, 2013). Os cenários de sorteio foram os seguintes:

- Sorteio de 1 nó dentre 2 de maior centralidade;
- Sorteio de 2 nós dentre 3 de maior centralidade;
- Sorteio de 3 nós dentre 5 de maior centralidade;
- Sorteio de 3 nós dentre todos os elementos da lista (3 nós sem faixa);
- Sorteio de 5 nós dentre todos os elementos da lista (5 nós sem faixa);

- Sorteio de 10 nós dentre todos os elementos da lista (10 nós sem faixa);

Analisando os resultados obtidos dos cenários de sorteio do SOCLEER com a carga de 1000 mensagens geradas, observou-se através dos gráficos de consumo de energia e o de participações dos nós no envio de mensagens valores aproximados, considerando a margem de erro, ocorreram empate.

Posteriormente, novas simulações foram feitas utilizando uma carga de 10000 mensagens geradas para testar se o protocolo SOCLEER apresentaria um melhor desempenho nesse cenário. Assim, foram considerados os seguintes cenários de sorteio (3 cenários):

- Sorteio de 1 nó dentre 2 de maior centralidade;
- Sorteio de 2 nós dentre 3 de maior centralidade;
- Sorteio de 10 nós dentre todos os elementos da lista;

6.2 Definindo os cortes

Foram utilizados cortes percentuais para determinar um subconjunto de nós considerados preferidos. Estes cortes foram aplicados nas análises de participação dos nós no envio de mensagens e no consumo de bateria para os protocolos SOCLEER e BUBBLE Rap. Portanto, os nós considerados preferidos estão acima da margem de corte proposta no cenário simulado. Neste trabalho, foram definidos os seguintes percentuais de corte baseado nos resultados obtidos com a simulação:

- Corte de 5%: Em cada rodada de simulação, ocorreu a participação média de n nós no envio de mensagens de no mínimo 5% da média de todas as mensagens criadas.
- Corte de 3%: Em cada rodada de simulação, ocorreu a participação média de n nós no envio de mensagens de no mínimo 3% da média de todas as mensagens criadas.
- Corte de 2%: Em cada rodada de simulação, ocorreu a participação média de n nós no envio de mensagens de no mínimo 2% da média de todas as mensagens criadas.

Foram desconsiderados para análise de ocorrência de nós preferidos os cortes acima de 3%, pois a quantidade de nós considerados preferidos foi muito pequena, quando aplicado ao SOCLEER. Contudo, para o cenário de 20000 mensagens, foi aplicado um corte de 5% para diminuir a quantidade de nós preferidos para o BUBBLE Rap. Na maior parte dos cenários, cortes abaixo de 2% foram muito abrangentes na seleção dos nós para o BUBBLE Rap, não evidenciando a ocorrência de nós preferidos, pois compreendia quase todos os nós contidos no trace para efeito desta análise.

Os percentuais escolhidos para os cortes foram definidos através de comparações dos resultados das simulações do SOCLEER com o BUBBLE Rap, visto que a média da participação dos nós no envio de mensagens para o BUBBLE Rap foi maior que para o SOCLEER, o que dificultou encontrar uma margem de corte para identificar a ocorrência de nós preferidos.

6.2.1 Cenário com Carga de 1000 Mensagens Criadas

- **Corte médio de 2%:**

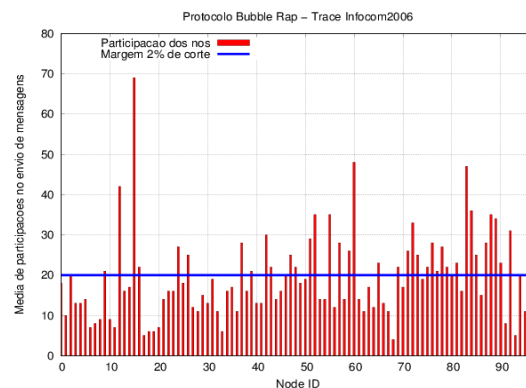


Figura 19 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 1000 mensagens.

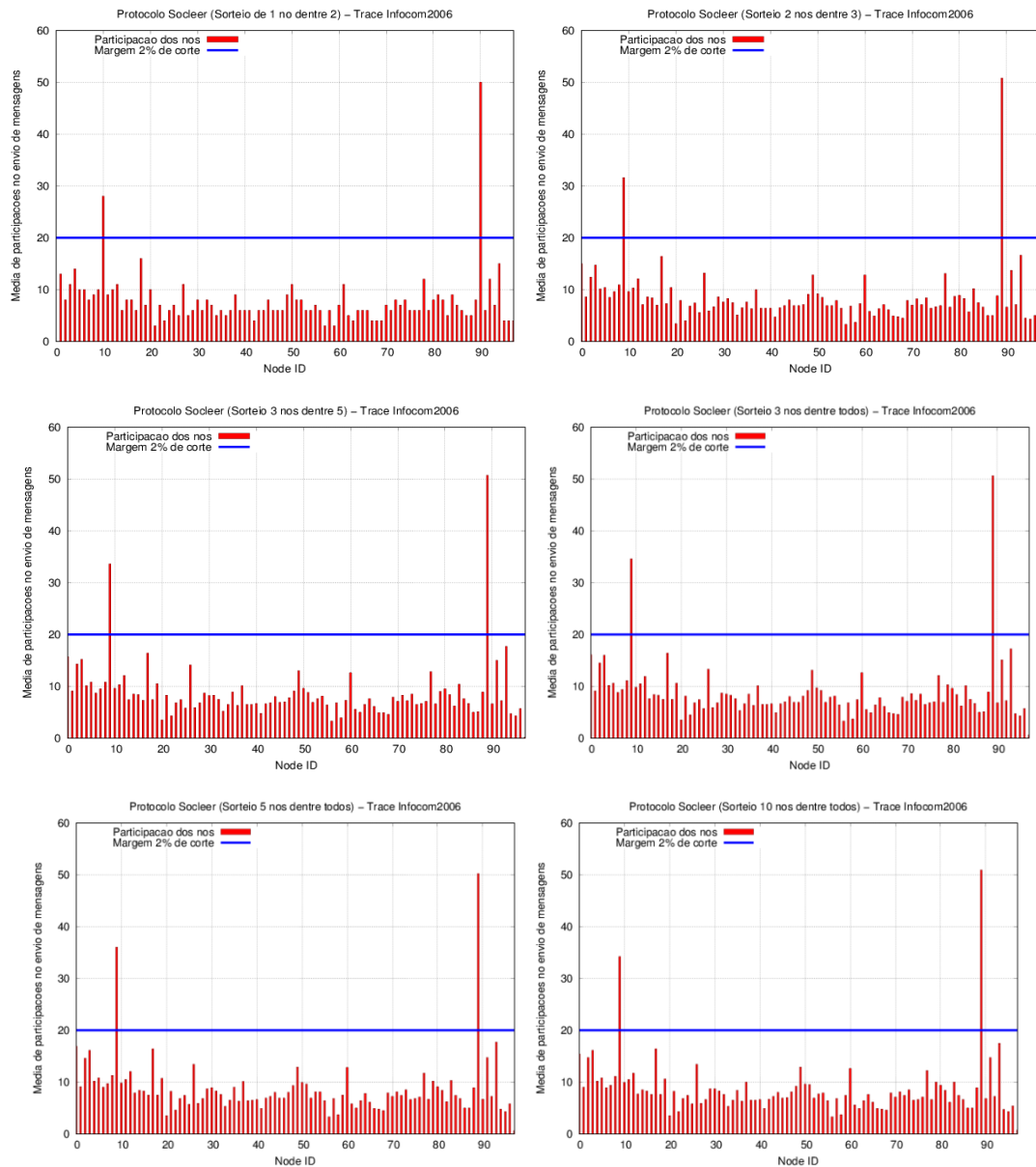


Figura 20 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 1000 mensagens.

O corte de 2% apresentou uma redução de ocorrência de nós preferidos e redistribuição na carga de participação dos nós no envio de mensagens em confronto com o BUBBLE Rap da ordem de 37,06%, não importando o cenário de sorteio – foram 2 ocorrências em 98 nós (2,0%) para o SOCLEER contra 39 ocorrências em 98 nós (39,8%) do BUBBLE Rap.

- **Desempenho do consumo de bateria para os protocolos e cenários estudados**

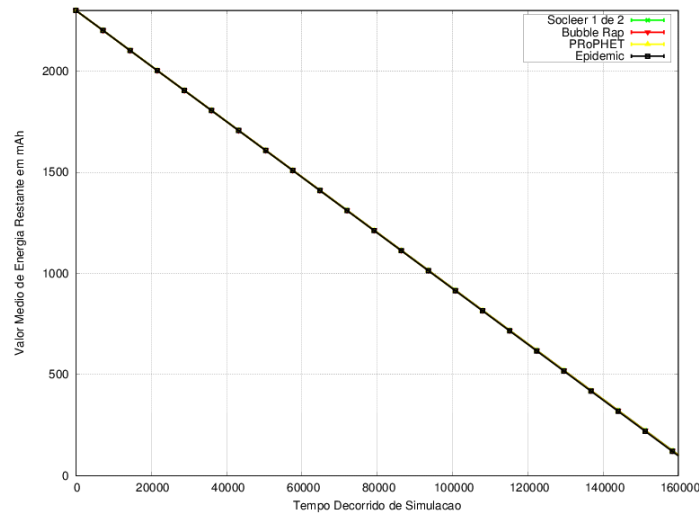


Figura 21- Média de energia residual de todos os nós em mAh para uma carga de 1000 mensagens

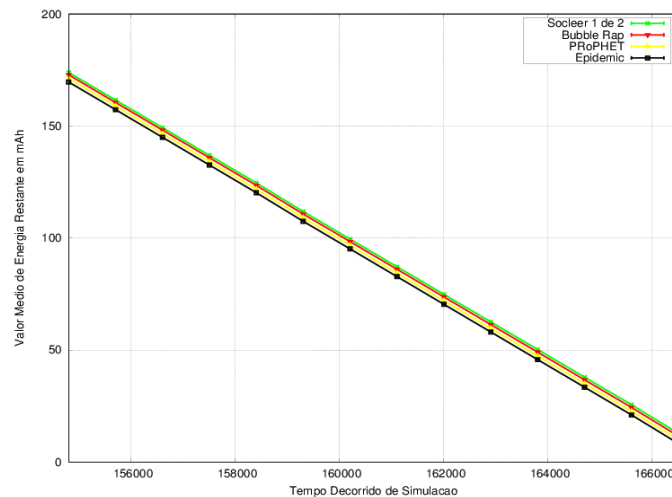


Figura 22 - Visualização aumentada da parte final da simulação da Figura 21 – Média de energia residual de todos os nós em mAh para uma carga de 1000 mensagens

A Figura 21 compara os resultados da medição de consumo de bateria de cada um dos protocolos simulados para todos os nós participantes, onde todos os protocolos apresentaram valores aproximados. Em relação ao SOCLEER, não houve diferença significativa do valor médio de energia restante para os cenários dos sorteios, portanto foi considerado, na plotagem, o resultado da simulação do sorteio de 1 nó dentre 2 de maior centralidade. A Figura 22 apresenta o intervalo de tempo entre 154800 segundos e 167400 segundos para mostrar a pequena diferença de energia residual entre os protocolos comparados. Ainda assim, o SOCLEER leva uma pequena vantagem neste intervalo.

- **Desempenho do consumo de bateria para protocolos de contexto social**

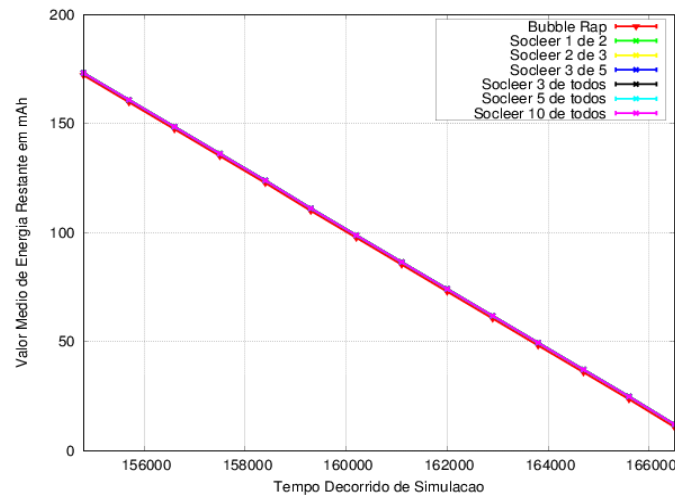


Figura 23 – Média de energia residual dos nós preferidos identificados no corte de 2% em mAh para uma carga de 1000 mensagens.

A Figura 23 apresenta um resultado para o trecho 154600 até 166500 segundos do trace. Pois, como não houve recarga de bateria, só foi considerado a parte das simulações em que os nós possuíam energia residual. Esse intervalo mostra a média de energia residual dos nós preferidos antes que a carga de bateria dos dispositivos acabe. Ocorreu um empate quando se compara o BUBBLE Rap com os cenários de sorteio do SOCLEER em consumo de bateria dos nós para uma carga de 1000 mensagens criadas.

6.2.2 Cenário com Carga de 10000 Mensagens Criadas

- **Corte médio de 2%:**

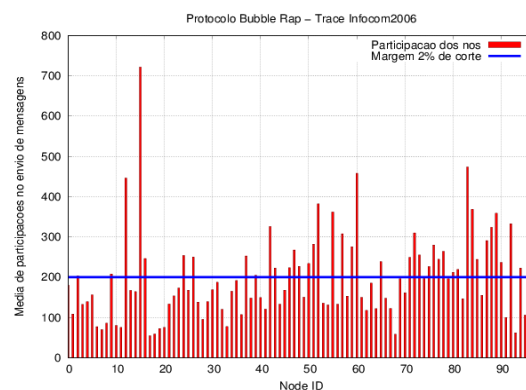


Figura 24 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 1000 mensagens.

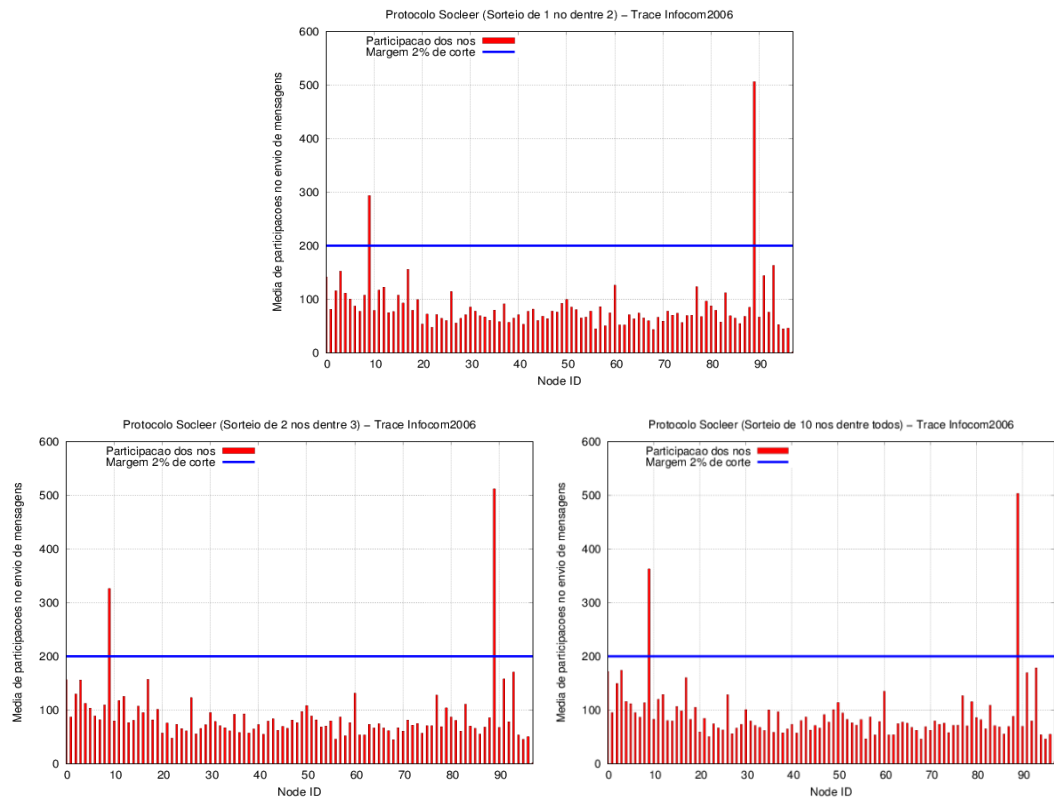


Figura 25 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 10000 mensagens.

O corte de 2% apresentou uma redução de ocorrência de nós preferidos e redistribuição na carga de participação dos nós no envio de mensagens em confronto com o BUBBLE Rap da ordem de 38,82%, não importando o cenário de sorteio – foram 2 ocorrências em 98 nós (2,0%) para o SOCLEER contra 40 ocorrências em 98 nós (40,82%) do *BUBBLE Rap*.

- **Corte médio de 3%:**

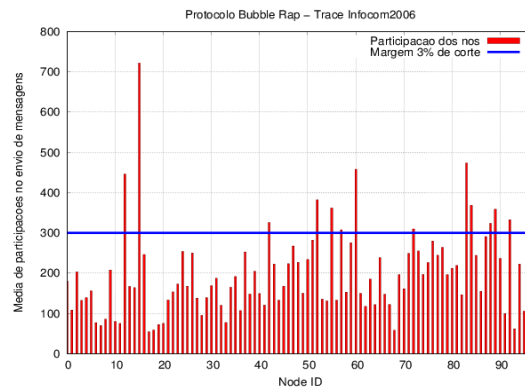


Figura 26 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 10000 mensagens.

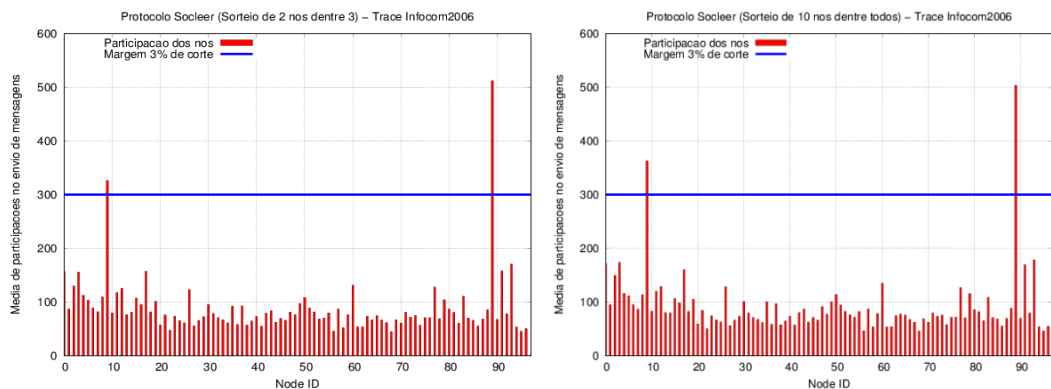
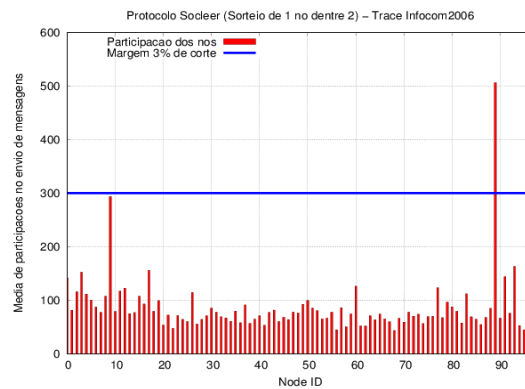


Figura 27 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 10000 mensagens.

O corte de 3% apresentou uma redução de ocorrência de nós preferidos e redistribuição na carga de participação dos nós no envio de mensagens em confronto com o BUBBLE Rap da ordem de 12,24% quando comparado com o SOCLEER utilizando o sorteio de 1 nó dentre 2. Nesse cenário do SOCLEER, a ocorrência foi de 1 nó preferido dentre 98 (1,02%). Já nos demais cenários, 2 nós dentre 3 e de 10 nós sem faixa, a variação foi de 11,22%. Quando considerados os cenários de 2 nós dentre

3 e de 10 nós sem faixa, a ocorrência foi de 2 nós preferidos em 98 (2,04%) contra 13 ocorrências em 98 (13,26%) nós do *BUBBLE Rap*

- **Desempenho do consumo de bateria para os protocolos e cenários estudados**

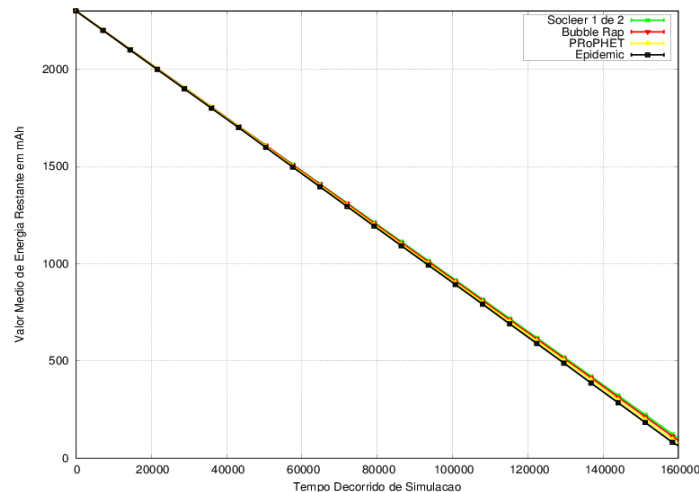


Figura 28 - Média de energia residual de todos os nós em mAh para uma carga de 10000 mensagens.

A Figura 28 compara os resultados da medição de consumo de bateria de cada um dos protocolos simulados para todos os nós participantes. No entanto, devido a configuração dos cenários sem a ocorrência de recarga de energia, a bateria dos dispositivos móveis foi totalmente utilizada antes do término do trace de mobilidade. Portanto, o gráfico apresenta o consumo médio de carga de bateria do início do trace até o instante de tempo onde a carga da bateria termina. A Figura 29 apresenta o intervalo de tempo entre 154800 e 167400 segundos para mostrar a pequena diferença de energia residual entre os protocolos comparados. Nota-se que o SOCLEER (sorteio de 1 nó dentre 2 de maior centralidade) apresentou um melhor desempenho em se tratando de energia residual neste cenário.

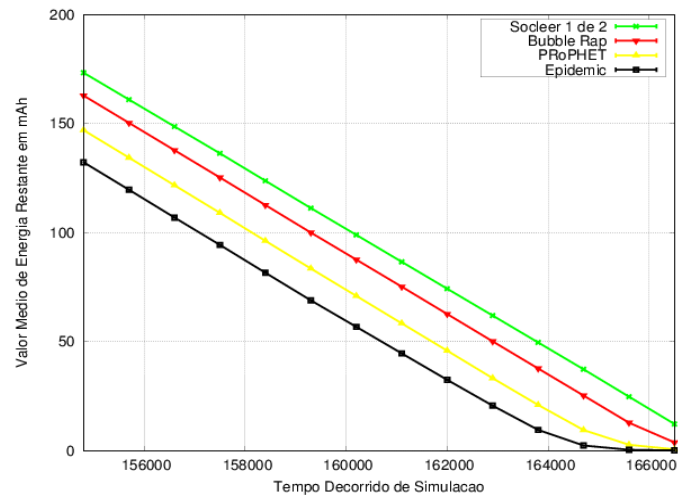


Figura 29 - Média de energia residual de todos os nós em mAh para uma carga de 10000 mensagens.

A Figura 30 compara o SOCLEER em relação ao *BUBBLE Rap* do início do *trace* até o instante de tempo onde a carga da bateria termina considerando apenas os nós preferidos para o corte de 3%. Como não houve diferença significativa entre os cenários de sorteio do SOCLEER comparado com *BUBBLE Rap*, foi escolhido o cenário de sorteio de 1 nó dentre 2 de maior centralidade no intervalo de tempo entre 154800 e 167400 segundos para evidenciar o melhor desempenho do SOCLEER, conforme mostra a Figura 31.

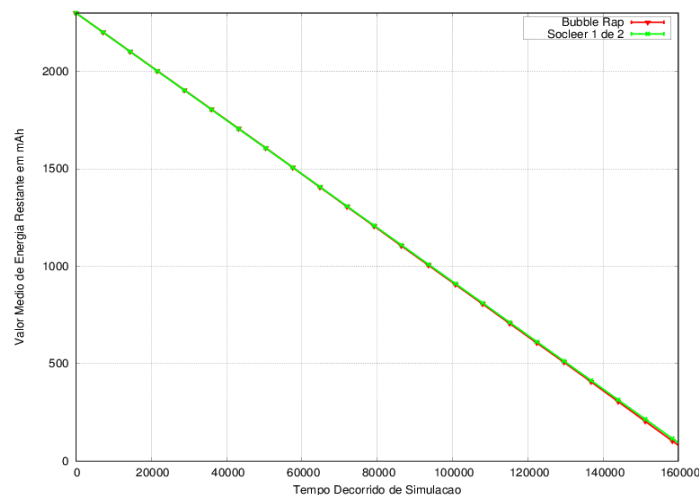


Figura 30 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 10000 mensagens

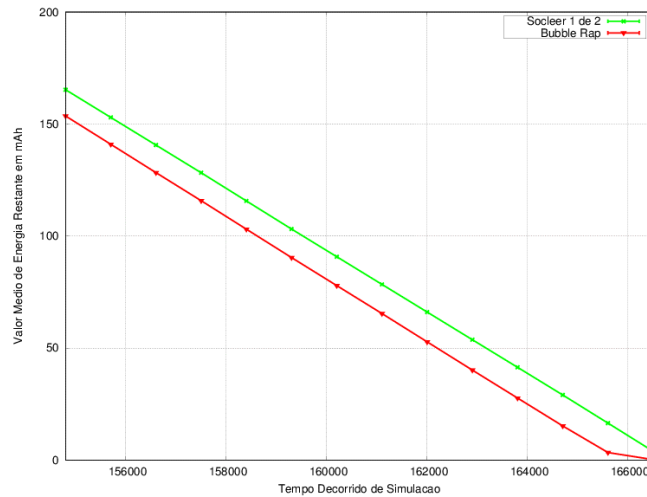


Figura 31 - Visualização aumentada da Figura 30 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 10000 mensagens

Segundo Machado (2013), em um cenário com uma carga maior de mensagens criadas na rede, o número de réplicas de mensagens aumenta e isto favorece o mecanismo de sorteio simples do SOCLEER, que passa a funcionar um pouco melhor em relação ao BUBBLE Rap no que diz respeito ao consumo de bateria, pois este gera mais réplicas que o SOCLEER em virtude da particularidade de seu mecanismo decisor de encaminhamento de mensagens.

6.2.3 Cenário com Carga de 20.000 Mensagens Criadas

- **Corte médio de 5%:**

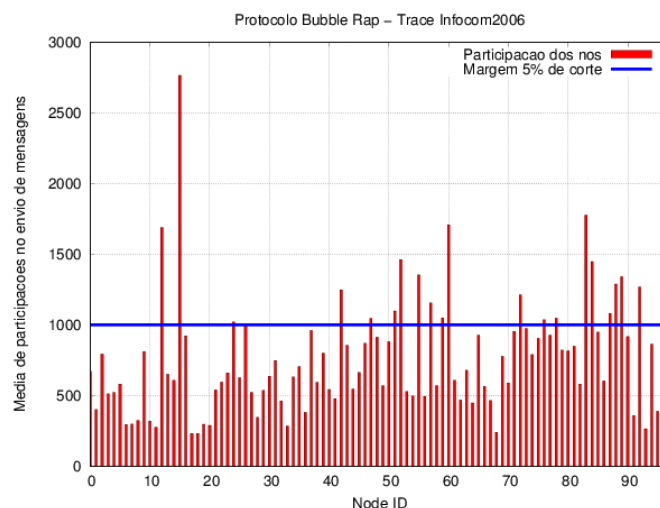


Figura 32 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 5% e uma carga de 20000 mensagens

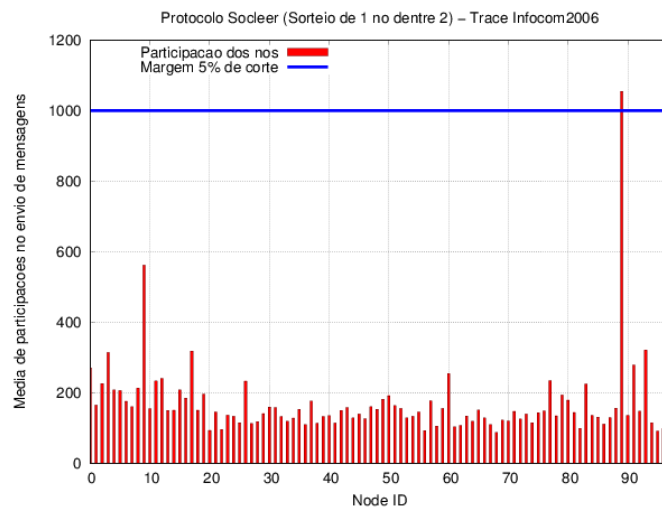


Figura 33 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 5% e uma carga de 20000 mensagens

- **Corte médio de 3%:**

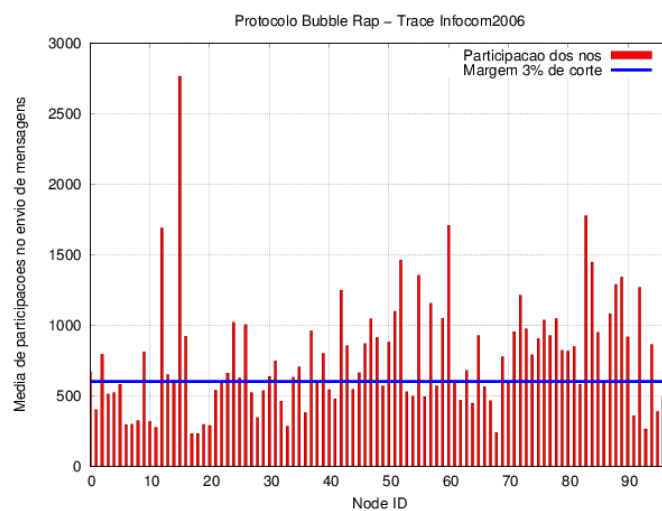


Figura 34 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 20000 mensagens

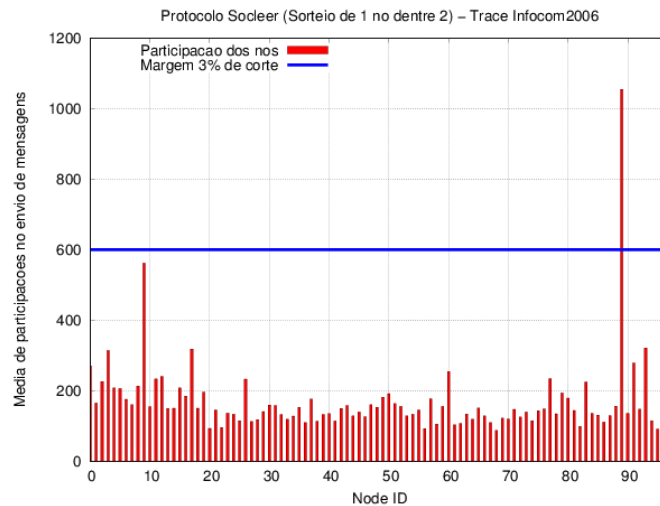


Figura 35 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 20000 mensagens

Observa-se nos gráficos que o SOCLEER, na maior parte dos nós, apresentou uma participação média no envio de mensagens menor que a maioria dos nós do *BUBBLE Rap*. Com isso, o total de nós preferidos para os cortes de 5% e 3% foi de apenas 1 nó para o SOCLEER, contra 21 e 56 nós respectivamente para o *BUBBLE Rap*. Uma possibilidade seria que devido o SOCLEER implementar um sorteio na lista com os nós de maior centralidade, pode ocorrer uma discrepância entre a taxa de centralidade destes nós. Assim, se o SOCLEER em seu sorteio, selecionar o segundo elemento da lista e este possuir uma taxa de centralidade muito inferior ao primeiro elemento, considerando ainda que o experimento utilizou TTL finito de 360 minutos, o caminho da mensagem até o destino pode ser comprometido.

As Figura 36 compara os resultados da medição de consumo de bateria de cada um dos protocolos simulados para todos os nós participantes, onde observa-se que o SOCLEER teve um desempenho superior aos outros protocolos em economia de energia.

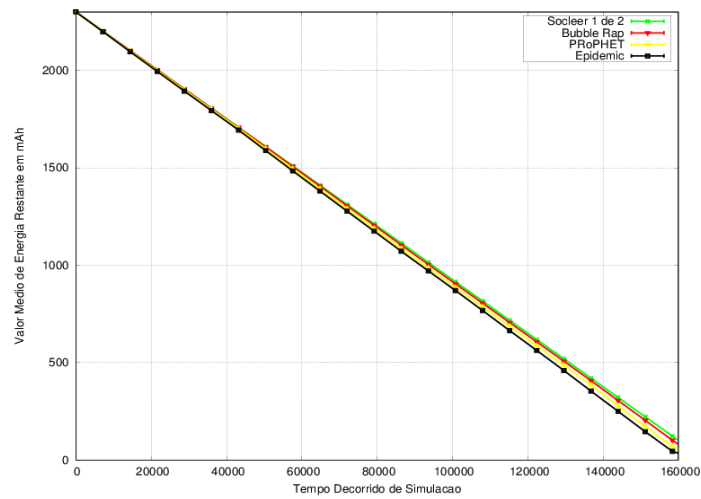


Figura 36 - Média de energia residual de todos os nós em mAh para uma carga de 20000 mensagens

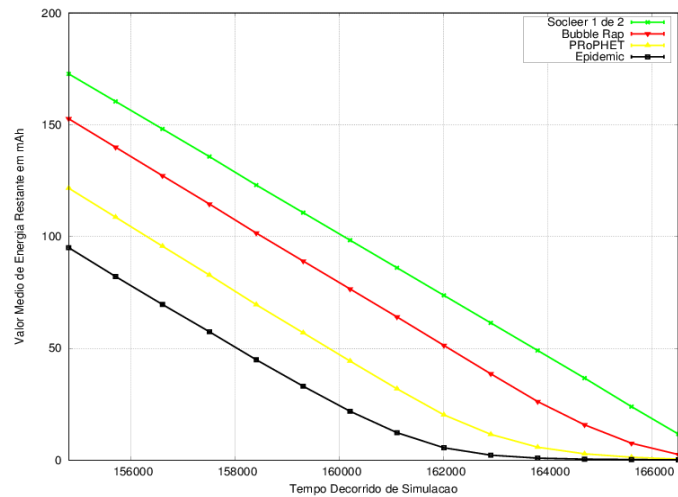


Figura 37 – Visualização aumentada da parte final da Figura 36 - Média de energia residual de todos os nós em mAh para uma carga de 20000 mensagens

No corte médio de 3% para os nós preferidos, o SOCLEER apresentou um melhor desempenho em economia de energia que o *BUBBLE Rap*, considerando, porém, que o SOCLEER teve apenas 1 nó preferido.

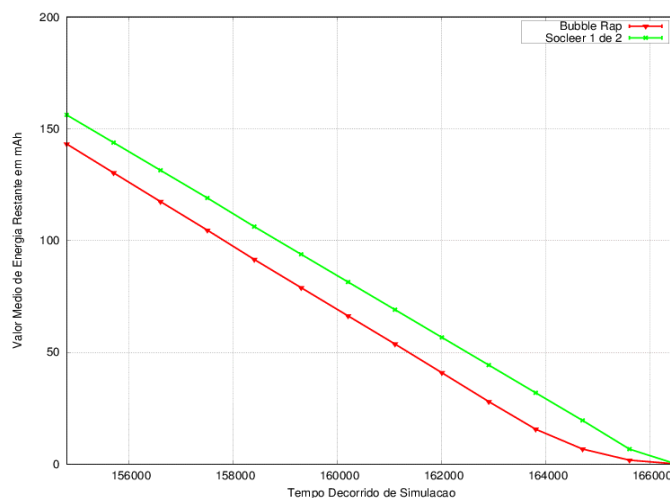


Figura 38 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 20000 mensagens

6.2.4 Cenário com Carga de 30.000 Mensagens Criadas

- Corte médio de 3%:

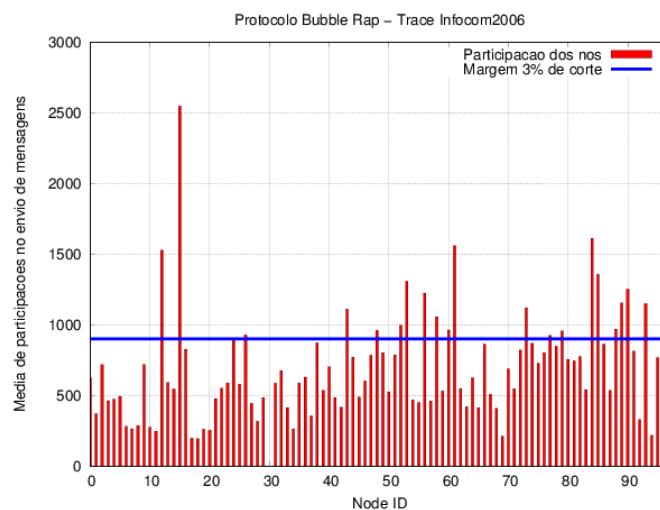


Figura 39 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 3% e uma carga de 30000 mensagens

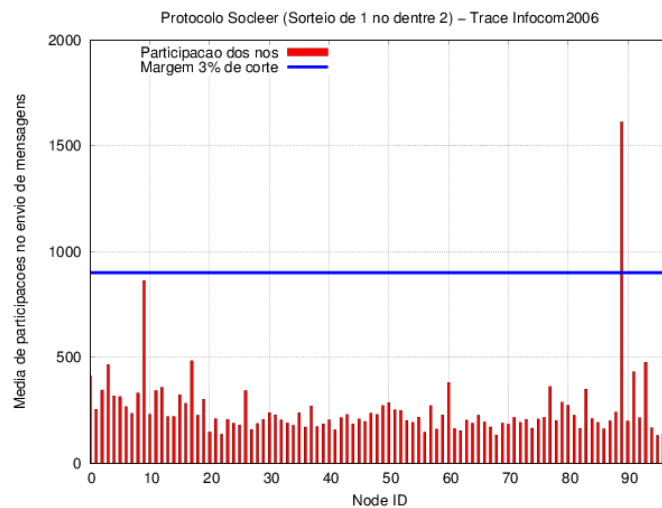


Figura 40 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 3% e uma carga de 30000 mensagens

- **Corte médio de 2%:**

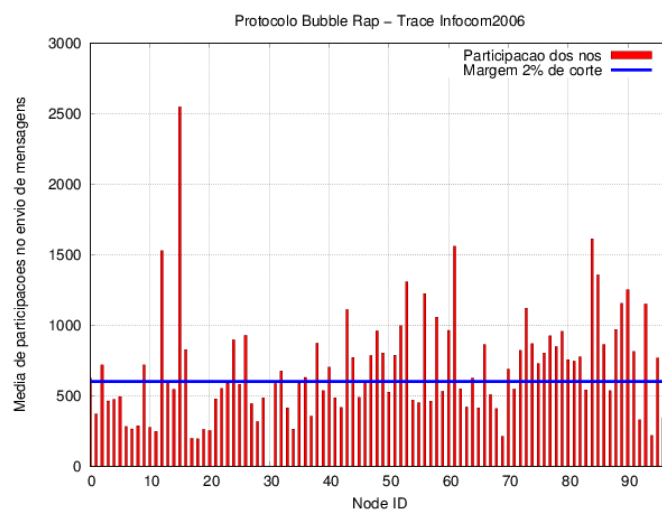


Figura 41 - Participação dos nós no envio de mensagens simulando o BUBBLE Rap para um corte de 2% e uma carga de 30000 mensagens

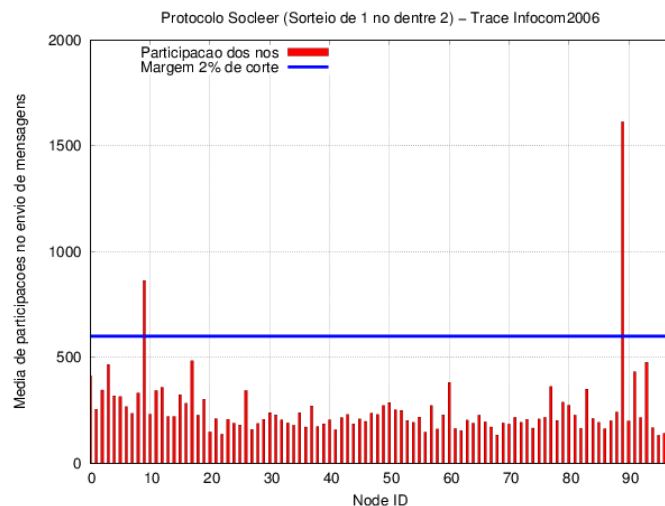


Figura 42 - Participação dos nós no envio de mensagens simulando o SOCLEER para um corte de 2% e uma carga de 30000 mensagens

O total de nós preferidos para o corte de 3% foi de apenas 1 nó (1,02%) para o SOCLEER contra 20 nós (20,41%) do *BUBBLE Rap*, e para o corte de 2% foram de apenas 2 nós (2,04%) para o SOCLEER contra 48 nós (48,98%) para o *BUBBLE Rap*.

A Figura 43 compara os resultados da medição de consumo de bateria de cada um dos protocolos simulados para todos os nós participantes, onde observa-se que o SOCLEER teve um desempenho superior aos outros protocolos em economia de energia. A Figura 44 apresenta este mesmo gráfico no intervalo de 154800 a 167400 segundos para uma melhor visualização deste resultado.

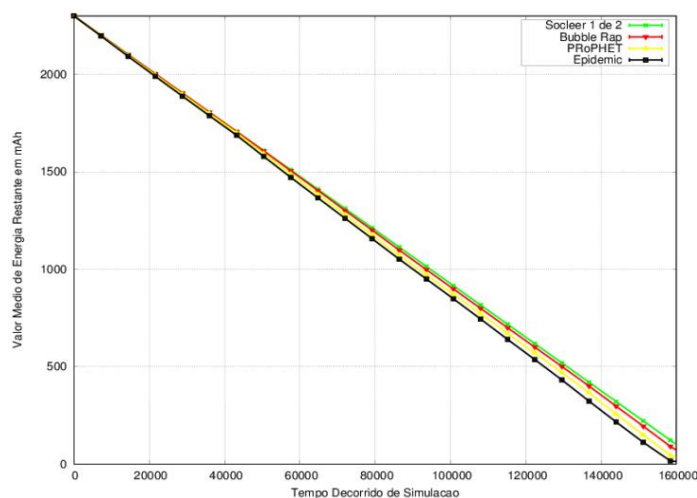


Figura 43 - Média de energia residual de todos os nós em mAh para uma carga de 30000 mensagens

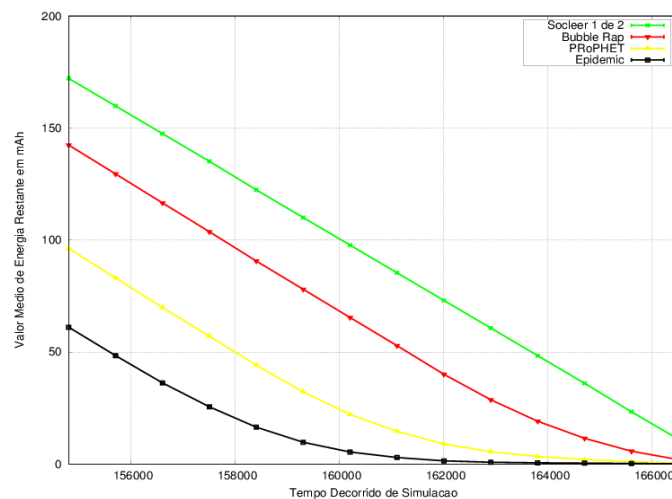


Figura 44 – Visualização aumentada da parte final da Figura 43 -Média de energia residual de todos os nós em mAh para uma carga de 30000 mensagens

Na análise de economia de energia dos nós preferidos, ambos os cortes de 3% e 2% apresentaram um desempenho superior para o SOCLEER comparado ao *BUBBLE Rap*, conforme apresentado nas Figuras 45 e 46.

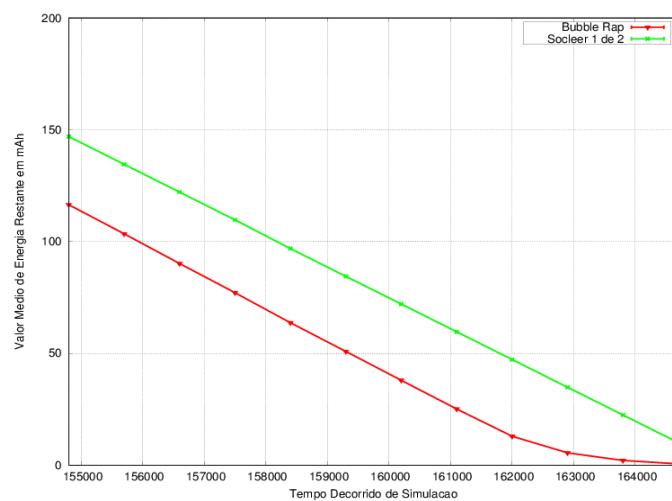


Figura 45 - Média de energia residual dos nós preferidos identificados no corte de 3% em mAh para uma carga de 30000 mensagens

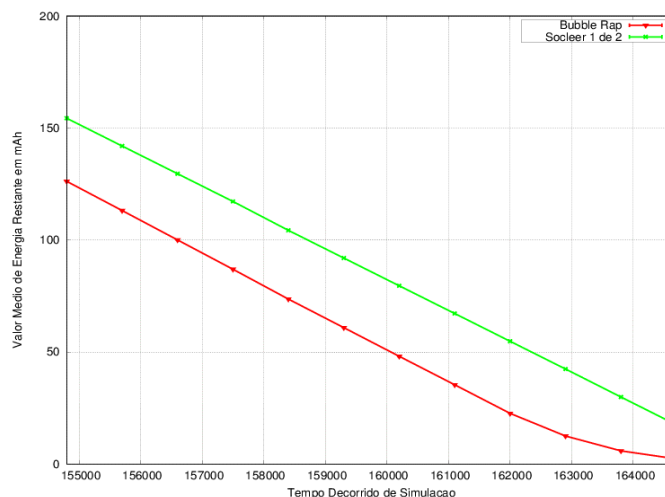


Figura 46 - Média de energia residual dos nós preferidos identificados no corte de 2% em mAh para uma carga de 30000 mensagens

6.2.5 Avaliação do SOCLEER em Relação às Métricas de Desempenho de Rede

- Fração de Mensagens Entregues:**

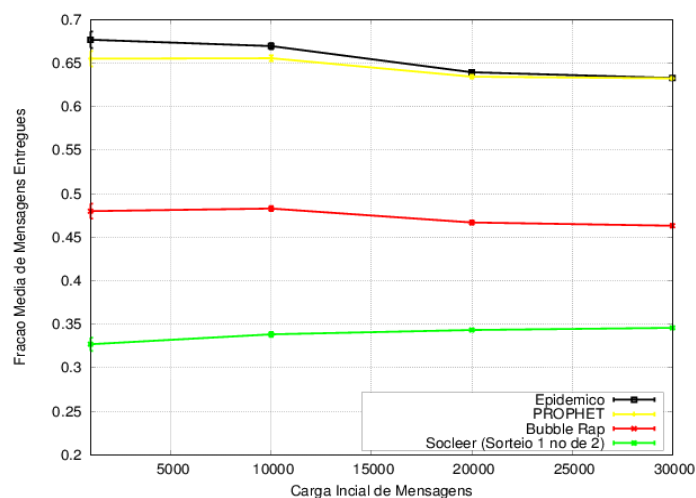


Figura 47 - Fração média de mensagens entregues por mensagens criadas

Utilizando-se o trace INFOCOM2006 nas simulações, pode-se perceber que o SOCLEER obteve a menor fração média de mensagens entregues se comparado com os demais protocolos, independente das mensagens criadas para os cenários de simulação. O Epidêmico e o PROPHET apresentaram as maiores frações médias de mensagens entregues, apresentando pequena vantagem para o Epidêmico, nas cargas de 1000 e 10000 mensagens. Nas cargas de 20000 e 30000 mensagens geradas na rede, se considerar a margem de erro, o Epidêmico e o PROPHET empataram. O *BUBBLE*

Rap apresentou uma fração média de mensagens entregues maior que o SOCLEER, porém inferior aos demais protocolos. Assim, pode-se dizer que o SOCLEER não apresentou um bom desempenho nesta métrica se comparado com os demais protocolos neste cenário de simulação.

- **Latência:**

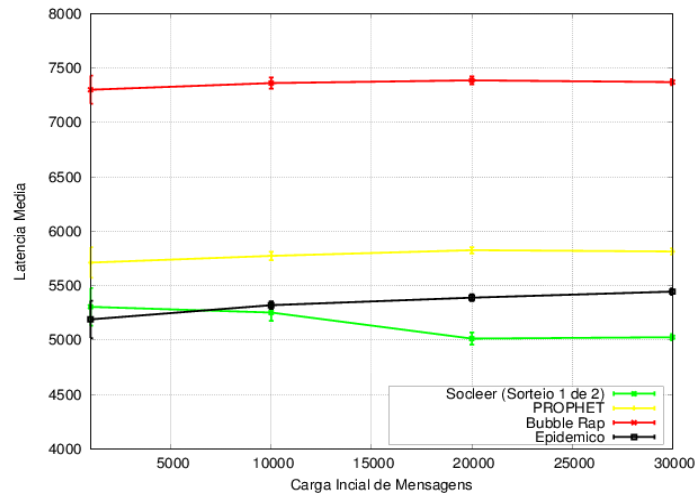


Figura 48 - Latência média por mensagens criadas

Considerando o trace do INFOCOM2006 nas simulações, verificou-se que o SOCLEER obteve uma latência média maior que o Epidêmico na carga de 1000 mensagens geradas na rede, mas se considerar a margem de erro percebe-se que ocorreu um empate entre esses protocolos. Na carga de 10000 mensagens criadas, o SOCLEER permaneceu com a menor latência, porém, se incluir a margem de erro, o SOCLEER empatou com o Epidêmico. Nas demais cargas, 20000 e 30000 mensagens, o SOCLEER teve a menor taxa média de latência dentre os protocolos *BUBBLE Rap*, Epidêmico e PROPHET. O *BUBBLE Rap* teve a maior latência, seguido do PROPHET e Epidêmico. Assim, o SOCLEER apresentou menor atraso no envio de mensagens, porém, o TTL finito pode ter impactado, pois o SOCLEER apresentou a menor fração média de mensagens entregues.

- **Overhead:**

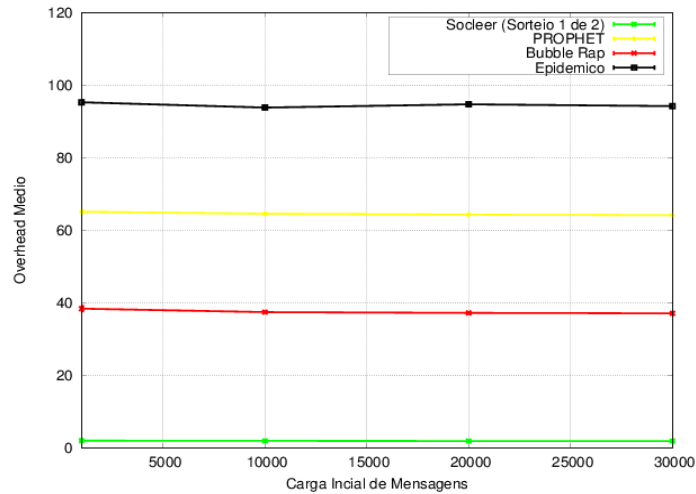


Figura 49 - Overhead médio por mensagens criadas

Utilizando-se o trace INFOCOM2006 nas simulações, pode-se perceber que o SOCLEER obteve o menor overhead médio se comparado com os demais protocolos para todos os cenários de cargas de mensagens (1000, 10000, 20000 e 30000) criadas na rede. Independente da quantidade de mensagens criadas o SOCLEER obteve o melhor desempenho dentre todos os protocolos, apresentando a menor taxa de overhead médio. Em seguida o *BUBBLE Rap* teve a segunda melhor taxa de overhead médio, seguido pelo PROPHET com a terceira posição e o Epidêmico com a pior taxa de overhead médio.

O mecanismo de sorteio, implementado pelo SOCLEER, gera a redistribuição das cópias de mensagens entre os nós com maior centralidade. Assim, nem todos os nós conectados ao nó portador da mensagem, que possuam maior centralidade que este, recebem uma cópia da mensagem. Desta forma, um número menor de cópias é propagado na rede causando um *overhead* menor para o protocolo SOCLEER quando comparado com BUBBLE Rap, Epidêmico e PROPHET.

- **Salto por Rota:**

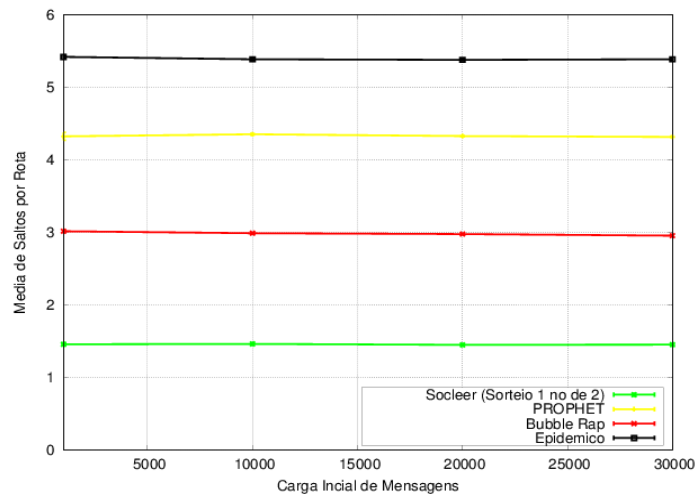


Figura 50 - Média de saltos por rota por mensagens criadas

Utilizando-se o trace INFOCOM2006 nas simulações, pode-se perceber que, independente das cargas de mensagens utilizadas, o SOCLEER obteve a menor média de saltos por rota se comparado com os demais protocolos, apresentando em média aproximada, a metade da média de saltos por rota que o *BUBBLE Rap* apresentou nos cenários de simulação. Em seguida, o PROPHET ficou com a terceira maior média de saltos por rota e o Epidêmico teve a maior média de saltos por rota dentre todos os protocolos analisados.

Considerando que o SOCLEER implementa um mecanismo de sorteio de n nós dentre m de maior centralidade em uma lista ordenada por centralidades, o algoritmo pode ter redistribuído cópias das mensagens para nós com centralidades semelhantes às dos nós preferidos. Isto pode ter contribuído para a diminuição da média de saltos por rota se comparado com o BUBBLE Rap, analisando só esta métrica, tem-se um desempenho positivo do SOCLEER em relação aos demais protocolos.

7 Conclusões

Neste capítulo serão apresentadas as dificuldades encontradas sobre o trabalho realizado, as considerações finais sobre os resultados obtidos e sugestões de possíveis trabalhos futuros.

7.1 Dificuldades

A realização deste trabalho exigiu superar diversas dificuldades. Dentre os principais desafios destacam-se: o tempo para a realização do TCC, a pesquisa e leitura de artigos e trabalhos acadêmicos na área de redes oportunistas, a falta de recursos computacionais com desempenho adequado para realizar as simulações que foram excessivamente demoradas e, o próprio uso do simulador ONE que também exigiu um tempo para o entendimento dos parâmetros de configurações, tendo que realizar os ajustes necessários conforme as simulações eram processadas. Sobre a simulação, a recarga de energia da bateria dos dispositivos móveis poderia ter sido observada para que as trocas de mensagens durassem até o final do *trace* de mobilidade e também o uso do TTL infinito, que também não foi observado neste experimento.

7.2 Considerações Finais

Este Trabalho de Conclusão de Curso propôs-se analisar o desempenho de uma proposta de protocolo de roteamento com base social para redes oportunistas. O protocolo apresentado é o SOCLEER, resultante da alteração do decisor do protocolo *BUBBLE Rap* por meio de um sorteio simples de n nós dentre os melhores elencados em uma lista ordenada de maiores centralidades para efeito de encaminhamento de mensagens. O

As simulações realizadas mostraram que o protocolo proposto supera os consagrados protocolos Epidêmico, PRoPHET e *BUBBLE Rap* em termos de economia de energia dos dispositivos em todos os cenários de carga de mensagens geradas na rede, e, também teve a menor taxa de latência média nos cenários de altas cargas de mensagens. Em relação as outras métricas de avaliação, o protocolo proposto obteve o menor *overhead* médio se comparado com os demais protocolos para todos

os cenários de cargas de mensagens, porém obteve a melhor performance em saltos por rota e na fração média de mensagens entregues por mensagens criadas.

7.3 Trabalhos Futuros

Como trabalhos futuros, sugere-se analisar o desempenho do SOCLEER em relação a outros protocolos com base social e com outros valores de sorteio em um novo cenário denso com outros valores de carga de mensagens.

Referências Bibliográficas

AKESTORIDIS, Dimitrios-Georgios., **CRAWDAD dataset uoi/haggle** (v. 2016-08-28): derived from cambridge/haggle (v. 2009-05-29), traceset: one, downloaded from <https://crawdad.org/uoi/haggle/20160828/one>, <https://doi.org/10.15783/C7Z884>, Aug 2016.

ALBINI, F. L. P. **PTTA: protocolo para distribuição de conteúdo em redes tolerantes ao atraso e desconexões**. 2013. 92 f. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

AL-OMARI, S. A. K.; SUMARI, P. **An Overview of Mobile Ad Hoc Networks for the Existing Protocols and Applications**. International journal on applications of graph theory in wireless ad hoc networks and sensor networks (Graph-Hoc), vol2, No.1, March 2010.

BRANCO, G. R.; MEDEIROS, M. V. B.; SALLES, R. M. **Emprego de redes tolerantes a atrasos e desconexões em cenários de emergência**. Revista Militar de Ciência e Tecnologia, v. 2010, p. 42-51, 2010.

CORREIA, L. H. A.; MACEDO, D. F.; RIBEIRO, M. A. S.; HEIMFARTH, T. **AntRoP - Protocolo de Roteamento Bio-inspirado em Colônia de Formiga Tolerante a Falhas e Desconexões aplicado às Redes Emergenciais**. In: Workshop de Gerência e Operação de Redes e Serviços (WGRS), 2011, Campo Grande. Anais do Workshop de Gerência e Operação de Redes e Serviços (WGRS), 2011.

CRAWDAD. CRAWDAD dataset cambridge/haggle (v. 2009-05-29), downloaded from <https://crawdad.org/cambridge/haggle/20090529>, <https://doi.org/10.15783/C70011>, May 2009.

FREEMAN, L. C. **Centrality in Social Networks: Conceptual Clarification.** Social Networks, Volume 1, Issue 3, 1978, Pages 215-239.

HUI, P.; CROWCROFT, J.; YONEKI, E. (2008) **BUBBLE Rap: Social-Based Forwarding in Delay Tolerant Networks.** MobiHoc '08, Hong Kong, SAR, China.

KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. **The ONE Simulator for DTN Protocol Evaluation.** in Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTOOLS), 2009. DOI: 10.4108/ICST.SIMUTOOLS2009.5674.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet:** Uma abordagem top-down. 3. ed. São Paulo: Addison Wesley, 2007. p. 393-396, 412-413.

LINDGREN, A.; DORIA, A.; SCHELEN, O. (2003). **Probabilistic routing in intermittently connected networks.** ACM Mobile Computing and Communications Review, Vol. 7.

MACHADO, N. J. **SOCLEER: uma proposta de disseminação de dados para redes oportunistas com redistribuição de carga em nós preferidos.** 2013. 106 f. Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2013.

MACHADO, N. J.; CAMPOS, C. A. V.; LUCENA, S. C. (2014). **Uma Proposta de Solução para o Problema de Nós Preferidos em Protocolos Oportunistas com Base Social.** Anais do 32º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC 2014.

PELUSI, L.; PASSARELLA, A.; CONTI, M. **Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks.** IEEE Communications Magazine, v. 44, n. 11, pp. 134-141, Nov. 2006.

QUELHAS, R.; COSTA, A.; MACEDO, J. (2011). **O Fenómeno social no encaminhamento em redes oportunistas**. Universidade do Minho, Braga, Portugal.

RECUERO, R. **Redes Sociais na Internet**. Porto Alegre: Sulina, 2009.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. (2005). **Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks**. In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05). ACM, New York, NY, USA, 252-259.
DOI=<http://dx.doi.org/10.1145/1080139.1080143>

VAHDAT, A.; BECKER, D. (2000). **Epidemic Routing for Partially-Connected Ad Hoc Networks**, Department of Computer Science, Duke University, Durham, UK, pages 1-14.

The ONE. **The Opportunistic Network Environment simulator**. Disponível em: <http://akeranen.github.io/the-one/>. Acessado em: 23 out. 2017.

The ONE. **Readme**. Disponível em: <https://github.com/akeranen/the-one/wiki/README>. Acessado em: 23 out. 2017.

The ONE. **The Opportunistic Network Environment simulator**. Disponível em: <http://www.netlab.tkk.fi/tutkimus/dtn/theone/> >. Acessado em: 23 out. 2017.

APÊNDICE A

```
#####  
#Apêndice A - Inseredb_consumo_medio_completo.sh  
#Autor: Marcelo C. Endson  
#Data: Dezembro/2017  
#####  
  
#!/bin/bash  
BASE='EnergyLevel.db'  
  
echo "Criando banco de dados."  
echo '  
CREATE TABLE consumo (  
    protocolo varchar(20),  
    seed integer,  
    tempo      integer,  
    no         varchar(3),  
    valor      decimal(8,4)  
);  
  
CREATE VIEW tempos as  
SELECT DISTINCT tempo  
FROM consumo;  
  
CREATE VIEW nos as  
SELECT DISTINCT no  
FROM consumo;  
  
CREATE VIEW media_tempo as  
SELECT  tempo,  CAST(REPLACE(REPLACE(tempo, "[", ""),  
    "]", "") as int) as num,  
AVG(valor) as media_tempo  
FROM consumo
```

```
GROUP BY tempo;
```

```
CREATE VIEW media_no as
SELECT no, CAST(REPLACE(no, "p", "") as int) as num,
AVG(valor) as media_nomedia_no
Media_no
FROM consumo
GROUP BY no;
```

```
CREATE VIEW media_tempo_no as
SELECT tempo, no, CAST(REPLACE(REPLACE(tempo, "[",
""), "]", "") as int) as num_tempo, CAST(REPLACE(no,
"p", "") as int) as num_no, AVG(valor) as
media_no_tempo
FROM consumo
GROUP BY tempo, no;
```

```
CREATE VIEW media_seed_tempo as
SELECT seed, tempo, AVG(valor) as media_seed_tempo
FROM consumo
WHERE tempo in
(SELECT tempo FROM media_tempo_SELECT)
GROUP BY seed, tempo;
```

```
CREATE VIEW media_tempo_SELECT_5 as --
SELECT tempo, media_tempo
FROM
(SELECT *, tempo%18000 as gran -- 5 em 5 horas
FROM media_tempo) t
WHERE t.gran = 0
ORDER BY tempo;
```

```
CREATE VIEW media_tempo_SELECT_6 as
```

```

SELECT tempo, media_tempo

FROM

(SELECT *, tempo%21600 as gran -- 6 em 6 horas
FROM media_tempo) t
WHERE t.gran = 0
ORDER BY tempo;

CREATE VIEW tempo_media_erro_DAT as
SELECT tempo, round(media_tempo,4),
(1.812*(variancia/9)/SQRT(10)) as erro
FROM

      (SELECT tempo, media_tempo, sum(quadifmedia) as
variancia

      FROM

      (SELECT seed, mst.tempo, mt.media_tempo,
(mst.media_seed_tempo - mt.media_tempo) as difmedia,
(mst.media_seed_tempo -
mt.media_tempo)*(mst.media_seed_tempo - mt.media_tempo)
as quadifmedia

      FROM media_seed_tempo mst
      JOIN media_tempo mt
      ON mst.tempo = mt.tempo ) t
      GROUP BY tempo) d
GROUP BY tempo;

' > model.sql
sqlite3 $BASE < model.sql
rm model.sql
echo "OK"

echo "Criando arquivo para gerar csv de importacao."
cat *EnergyLevelReport.txt > Seeds.txt

```

```

echo "OK"

echo "Gerando csv para importar no banco de dados."
COUNT=0
TEMPO='[]'
SEED=0

PROTOCOLO=$(echo *EnergyLevelReport.txt | cut -d "_" -f2)
echo $PROTOCOLO

while read line
do
    ((COUNT++))
    if grep -q '\[' <<< $line; then
        #TEMPO=$(tr -dc [:digit:] <<< $line)
        #echo $TEMPO
        #if [ $TEMPO = 900 ]; then
        TEMPO=$(cut -d"[" -f2 <<< $line)
        TEMPO=$(cut -d"]" -f1 <<< $TEMPO)
        if [ $TEMPO -eq 900 ]; then
            ((SEED++))
            #echo "Seed no if = " $SEED
        fi
        continue
    fi
    NO=$(cut -d" " -f1 <<<$line)
    VALOR=$(cut -d" " -f2 <<<$line)

    #if [ $COUNT -eq 37223 ]; then
    #    exit 1
    #fi
    echo $PROTOCOLO,"$SEED","$TEMPO","$NO","$VALOR >>
import.csv

```

```

done < Seeds.txt
rm Seeds.txt

echo "OK"

echo "Importando para o banco de dados"
echo '
.mode csv
.import import.csv consumo
.quit
' > import_cmd_sqlite.txt
sqlite3 $BASE < import_cmd_sqlite.txt
#rm import*
echo "OK"

echo "Criando dat"
echo '
.mode tabs
SELECT num, media_tempo
FROM media_tempo
ORDER BY num;
' > query.sql
sqlite3 EnergyLevel.db < query.sql > EnergyLevel.dat
echo "OK"

```

APÊNDICE B

```
#####  
#Apêndice B - inseredb_participacao.sh  
#Autor: Marcelo C. Endson  
#Data: Dezembro/2017  
#####  
  
#!/bin/bash  
BASE='EnergyLevel.db'  
  
echo "Criando banco de dados."  
echo '  
CREATE TABLE participacao (  
    no          varchar(3),  
    total       integer  
);  
' > model.sql  
sqlite3 $BASE < model.sql  
rm model.sql  
echo "OK"  
  
echo "Importando para o banco de dados"  
echo '  
  
.mode csv  
.import import_participacao.csv participacao  
.quit  
' > import_participacao_sqlite.txt  
sqlite3 $BASE < import_participacao_sqlite.txt
```


APÊNDICE C

```
#####  
#Apêndice C - Script.sql (Criação de views que foram  
criadas após o banco ser implementado)  
#Autor: Marcelo C. Endson  
#Data: Dezembro/2017  
#####
```

```
CREATE VIEW media_tempo_SELECT_6 as  
SELECT tempo, media_tempo  
FROM  
(SELECT *, tempo%21600 as gran -- 6 em 6 horas  
FROM media_tempo  
) t  
WHERE t.gran = 0  
ORDER BY tempo;
```

```
-- SELECT * FROM media_tempo_SELECT_6;
```

```
CREATE VIEW media_seed_tempo_6 as  
SELECT seed, tempo, AVG(valor) as media_seed_tempo  
FROM consumo  
WHERE tempo in (SELECT tempo FROM media_tempo_SELECT_6)  
GROUP BY seed, tempo;
```

```
CREATE VIEW tempo_media_erro_DAT_6 as  
SELECT tempo, round(media_tempo,4),  
(1.812*(variancia/9)/SQRT(10)) as erro  
FROM  
(SELECT tempo, media_tempo, sum(quadifmedia) as  
variancia  
FROM
```

```

        (SELECT      seed,      mst.tempo,      mt.media_tempo,
        (mst.media_seed_tempo      -      mt.media_tempo)      as
        difmedia,
        (mst.media_seed_tempo      -
        mt.media_tempo) * (mst.media_seed_tempo      -
        mt.media_tempo) as quadifmedia
        FROM media_seed_tempo_6 mst
        JOIN media_tempo mt
        ON mst.tempo = mt.tempo
        ) t
    GROUP BY tempo) d
GROUP BY tempo;

CREATE VIEW consumo_medio_participacao_corte_4 as
SELECT      tme.tempo,      p.media,      tme.erro      FROM
tempo_media_erro_DAT_6 as tme
JOIN (SELECT c.tempo, AVG(c.valor) as media
      FROM consumo c
      JOIN participacao p on c."no" = p."no"
      WHERE (p.total/10) > 40 -- corte dos nos com
participação em 3% do envio das mensagens, carga de 1000
mensagens
      GROUP BY tempo) as p
      ON tme.tempo = p.tempo;

```