UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE

DEPARTAMENTO DE COMPUTAÇÃO

RENATO NOLASCO DA ROCHA

AVALIAÇÃO E IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES EM UM SIMULADOR PARA REDES MÓVEIS TOLERANTES A ATRASOS E DESCONEXÕES

ALEGRE - ES

2016

RENATO NOLASCO DA ROCHA

AVALIAÇÃO E IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES EM UM SIMULADOR PARA REDES MÓVEIS TOLERANTES A ATRASOS E DESCONEXÕES

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, sob orientação do Professor M. Sc. Rodrigo Freitas Silva.

ALEGRE- ES

2016

RENATO NOLASCO DA ROCHA

AVALIAÇÃO E IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES EM UM SIMULADOR PARA REDES MÓVEIS TOLERANTES A ATRASOS E DESCONEXÕES

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 01 de dezembro de 2016.

COMISSÃO EXAMINADORA

Prof. M.Sc. Rodrigo Freitas Silva
Universidade Federal do Espírito Santo
Orientador

Prof. M.Sc. Helder de Amorim Mendes
Universidade Federal do Espírito Santo

Prof. M.Sc. Valéria Alves da Silva
Universidade Federal do Espírito Santo

Dedico todo esse trabalho aos meus pais, Ângelo Renato da Rocha e Creuza Maria Nolasco, por todo incentivo, apoio e força que me deram.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por tudo o que ele tem feito em minha vida ao longo desses últimos anos.

Agradeço aos meus pais e ao meu irmão pelo apoio e por me incentivarem nos momentos que mais precisei e que nunca pouparam esforços para que eu pudesse me dedicar inteiramente às atividades acadêmicas; pelo carinho, dedicação, paciência; por terem me dado uma excelente educação, discernimento, força e fé para realizar meus sonhos. Sem eles não seria possível a realização deste trabalho.

Aos amigos e demais familiares pela motivação.

Aos amigos do alojamento a minha eterna casa em Alegre.

À Roselice, pelo companheirismo, motivação e por me ajudar a superar as dificuldades.

Ao orientador Rodrigo, por todo comprometimento, paciência e confiança em meu trabalho.

Aos professores em geral, por toda a atenção, comprometimento e profissionalismo durante toda a graduação.

A todos, muito obrigado.

AVALIAÇÃO E IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES EM UM SIMULADOR PARA REDES MÓVEIS TOLERANTES A ATRASOS E DESCONEXÕES

RESUMO

Redes *Ad Hoc* são redes formadas por dispositivos computacionais sem fio. Essas redes não possuem infraestrutura física nenhuma e todo processo de comunicação é desempenhado unicamente pelos próprios nós da rede. O objetivo deste trabalho é avaliar um simulador de redes e desenvolver um módulo de mobilidade para uma nova plataforma de simulação para redes sem fio capaz de avaliar protocolos de rede utilizados em dispositivos móveis. Além disso, o simulador chamado *SimDTN*, implementa modelos de mobilidade sintéticos e aceita traces reais como parâmetro de entrada no simulador e uma ampla gama de protocolos de roteamento utilizados em DTN. Em Adicional, as simulações resultantes das métricas de conectividade e roteamento podem ser explicadas graficamente.

Palavras-chave: *Redes Móveis. Primeiro Contato. Conectividade*.

EVALUATION AND IMPLEMENTATION OF NEW FUNCTIONALITIES IN A SIMULATOR FOR MOBILE NETWORKS TOLERANT TO DELAYS AND DISCONNECTIONS

ABSTRACT

Ad Hoc networks are networks made up of wireless computing devices. These networks have no physical infrastructure whatsoever and every communication process is performed solely by the network nodes themselves. The objective of this work is to evaluate a network simulator and to develop a mobility module for a new simulation platform for wireless networks capable of evaluating network protocols used in mobile devices. In addition, the simulator called SimDTN implements synthetic mobility models and accepts real traces as input parameter in the simulator and a wide range of routing protocols used in DTN. In addition, the simulations resulting from the connectivity and routing metrics can be explained graphically.

Keywords: Mobile Networks. First contact. Connectivity.

LISTA DE FIGURAS

FIGURA 1 - TIPOS DE REDES SEM FIO 15
FIGURA 2 - REDES SEM FIO 17
FIGURA 3 - REDES SEM FIO NÃO ESTRUTURADA (*AD HOC*) 19
FIGURA 4 - REDES DTN 21
FIGURA 5 - REGRA DE BORDA *BOUNCE* 25
FIGURA 6 - REGRA DE BORDA DELETE AND REPLACE 26
FIGURA 7 - REGRA DE BORDA *WRAP AROUND* 26
FIGURA 8 - MOVIMENTAÇÃO NO *RANDOM WALK* 28
FIGURA 9 - MOVIMENTAÇÃO NO *RANDOM WAYPOINT* 29
FIGURA 10 - GRAU DE UM NÓ 31
FIGURA 11 - NÓS ISOLADOS 32
FIGURA 12 - NÚMERO DE COMPONENTES CONEXAS 33
FIGURA 13 - DURAÇÃO DO ENLACE 34
FIGURA 14 - SIMULADOR *OMNET++* 37
FIGURA 15 - SIMULADOR ONE 38
FIGURA 16 - SIMULADOR *SUMO* 40
FIGURA 17 - SIMULADOR *NS 2* 41
FIGURA 18 - SIMDNT UTILIZANDO O PROTOCOLO EPIDÊMICO 45
FIGURA 19 - SIMULAÇÃO UTILIZANDO O PROTOCOLO *SPRAY AND WAIT* 49
FIGURA 29 - TRACES DE MOBILIDADE REAL 56

LISTA DE TABELAS

TABELA 1 - PARÂMETROS DOS PROTOCOLOS DE ROTEAMENTO 42
TABELA 2 - PARÂMETROS DOS MODELOS DE MOBILIDADE 43
TABELA 3 - PARÂMETROS UTILIZADO NOS RESULTADOS..... 50
TABELA 4 - CÓDIGO FONTE DO PROTOCOLO *PRIMEIRO CONTATO*..... 54
TABELA 5 - CÓDIGO FONTE DO MÓDULO DE LEITURA. 58
TABELA 6 - PARÂMETROS UTILIZADO NOS RESULTADOS..... 60
TABELA 7 - RESULTADO DA ANÁLISE DOS SIMULADORES 64

Lista de Gráfico

GRÁFICO 1 - GRÁFICO DE NÚMERO DE SALTOS..... 61
GRÁFICO 2 - TAXA DE ENTREGA..... 62
GRÁFICO 3 - *TEMPO DE ENTREGA* 63

LISTA DE ABREVIATURAS E SIGLAS

DTN	Disruption Tolerant Networks
FC	First Contact
MANET	Mobile Ad Hoc Network
NM	Nó Móvel
NÓS	Elementos receptores, transmissores e armazenadores de informação
P2P	Point-To-Point
PROPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
RWP	*Random* Waypoint
SUMO	Simulation of Urban Mobility
UFES	Universidade Federal do Espírito Santo
WLAN	Wireless Local Area Network
AODV	Ad Hoc on Demand Distance Vector Routing
DSR	Dynamic Source Roting
TORA	Temporally Ordered Routing Algorithm
DSDV	Destination Sequenced Distance Vector Routing
BGP	Border Gateway Protocol
OLSR	Optimized Link State Routing
OMNeT++	Objective Modular Network Testbed
TCL	Tool Command Language

SUMÁRIO

1 INTRODUÇÃO 14

2 OBJETIVOS 16

2.1 Objetivo geral 16

2.2 Objetivos específicos 16

3 FUNDAMENTAÇÃO TEÓRICA 17

3.1 Redes sem fio 17

3.2 Rede móvel sem fio *Ad Hoc* 19

3.3 Redes DTN 20

3.4 Protocolo de roteamento 21

3.4.1 Epidêmico 22

3.4.2 Prophet 22

3.4.3 Spray and Wait 23

3.4.4 Primeiro Contato 24

3.5 Regra de borda 24

3.5.1 Bounce 24

3.5.2 Delete and Replace 25

3.5.3 WRAP AROUND 26

3.6 Mobilidade dos nós 26

3.6.1 RANDOM WALK 27

3.6.2 RANDOM WAY POINT 28

3.6.3 SMOOTH RANDOM 29

3.7 Métricas de conectividade 30

3.7.1 GRAU DE UM NÓ 31

3.7.2 NÚMERO DE NÓS ISOLADOS 31

3.7.3 NÚMERO DE COMPONENTES CONEXAS 32

3.7.4 DURAÇÃO DOS ENLACES 33

4 REVISÃO BIBLIOGRÁFICA 35

5 METODOLOGIA 46

5.1 Desenvolvimento de um módulo para o *SimDTN* 46

5.1.1 Protocolo Primeiro Contato 46

5.1.2 Módulo de leitura 47

5.2 Avaliação do simulador *SimDTN* 48

5.3 Avaliação comparativa das funcionalidades dos simuladores ... 50

6 RESULTADOS 53

7 CONCLUSÃO 66

8 REFERÊNCIAS 67

1 INTRODUÇÃO

Nos últimos anos, presencia-se um aumento significativo do número de usuários móveis nas redes sem fio. Dentro desse contexto, pesquisadores tendem a desenvolver e avaliar novos protocolos de redes que objetivam, em geral, maximizar a taxa de entrega dos dados, diminuir o atraso de transmissão fim a fim, evitar a sobrecarga de dados e o descarte de pacotes na rede. Dada à dificuldade e o alto custo de se testar tais protocolos em um ambiente real, simuladores de redes são propostos a fim de oferecer uma plataforma de mobilidade e de comunicação capaz de simular a transmissão de dados entre os nós da rede. Portanto, existem muitos trabalhos relacionados as redes sem fio *Ad Hoc* sendo desenvolvidos.

Este tipo de rede sem fio possui diversos problemas a serem resolvidos, devido ao fato da maioria dos protocolos existentes serem implementados para redes estruturadas, não levando em consideração as limitações existentes nos dispositivos móveis, tornando o perfil desses protocolos inadequados a mobilidade em vista que esses equipamentos não utilizam redes estruturadas (PINHEIRO, 2005).

Uma rede não estruturada é caracterizada pela mobilidade de seus nós, podendo existir tanto nós fixos quanto nós móveis. No entanto, uma forma de distinguir e caracterizar as redes móveis está relacionado com a mobilidade de seus nós. Ocasionalmente quando alguns nós são fixos, estes, de certa forma implementam uma infraestrutura básica como a apresentada na Figura 1, a qual é responsável pelo gerenciamento da rede. Enquanto na rede móvel *Ad Hoc*, não há necessidade de nenhum tipo de infraestrutura, todos os nós móveis são capazes de se comunicaram diretamente com os seus vizinhos ou por meio da propagação.

Figura 1 - Tipos de redes sem fio

Fonte: Bastos (2009, p. 5)

Por outro lado a rede móvel *Ad Hoc*, apresenta várias particularidades que devem ser levadas em consideração no desenvolvimento de seus protocolos de roteamento, que são: a existência de topologia dinâmica, a largura de banda limitada de seus enlaces e a necessidade de conservação de energia (REZENDE, 2004). Para que os protocolos de roteamento funcionem da melhor forma, descobrindo rotas e encaminhando mensagens.

Dessa forma, os principais simuladores de redes móveis sem fio tolerantes a atrasos e desconexões existentes na literatura são: o *ONE* (KERANEN et al.,2009), o *SUMO* (BEHRISCH et al., 2011), o *NS* (FALL; VARADHAN ,2009) e o *OMNeT*++ (VARGA, 2013).

2 OBJETIVOS

A seguir serão descritos os objetivos gerais e específicos relacionados ao desenvolvimento deste trabalho.

2.1 Objetivo geral

O objetivo geral deste trabalho, consiste no desenvolvimento de um módulo com um algoritmo de roteamento e o mecanismo de leitura de mobilidade real para o simulador alternativo de redes móveis, o *SimDTN,* e avaliação das características gerais do simulador em relação a outros simuladores de redes DTN existentes.

2.2 Objetivos específicos

Para alcançar o objetivo geral, pretende-se:

- Implementar o protocolo de roteamento *Primeiro Contato*.
- Implementar a opção de entrada de arquivo externo referente a mobilidade dos nós.
- Avaliar a utilização do simulador *SimDTN*.
- Testar e avaliar o simulador verificando e comparando as funcionalidades deste simulador com as de outros encontrados na literatura.

3 FUNDAMENTAÇÃO TEÓRICA

A seguir será introduzida a fundamentação teórica que servirá como base para a elaboração desse trabalho.

3.1 Redes sem fio

As Redes sem fio ou *Wireless* (*WLANs*) surgiram da mesma forma que muitas outras tecnologias através da invenção para uso militar. Havia a necessidade de implementação de um método simples e seguro para troca de informações em ambiente de combate. O tempo passou e a tecnologia evoluiu, deixando de ser restrita ao meio militar e se tornou acessível a empresas, faculdades e ao usuário doméstico (FERREIRA, 2013). Atualmente, as redes sem fio estão ganhando bastante espaço e vem substituindo as redes convencionais cabeadas, e no cenário atual estão presentes em quase todos os locais de acesso coletivo e residenciais, proporcionando um importante meio de acesso à internet.

Figura 2 - Redes sem fio

Fonte: Cabral (2014, p. 5)

Esse tipo rede permite a mobilidade dos equipamentos como os apresentados na Figura 2, flexibilidade, diminuição de custos de infraestrutura, instalações em áreas de difícil cabeamento, e proporciona maior confiabilidade e robustez, dentre outros.

Todas estas características permitem que a tecnologia móvel sem fio ofereça aos usuários a possibilidade de acesso aos mesmos serviços a qualquer instante e em diferentes lugares (BEZERRA, 2009). Deste modo, as redes sem fio vêm ganhando bastante atenção por parte de alguns estudiosos, devido a sua simplicidade e utilidade por meio da adaptabilidade a qualquer ambiente.

Existem dois tipos de redes sem fio, a com infraestrutura (redes *WLANs*) a qual a transferência de dados ocorre sempre entre uma estação e um ponto de acesso, onde mesmo que os dispositivos estejam próximos não conseguem realizar nenhum tipo comunicação entre eles e a sem infraestrutura (redes *Ad Hoc*, redes de celulares, redes veiculares entre outras) onde não há necessidade de qualquer infraestrutura para funcionar, havendo apenas a necessidade de um ou mais dispositivos para estabelecer a comunicação espontânea por meio destes dispositivos.

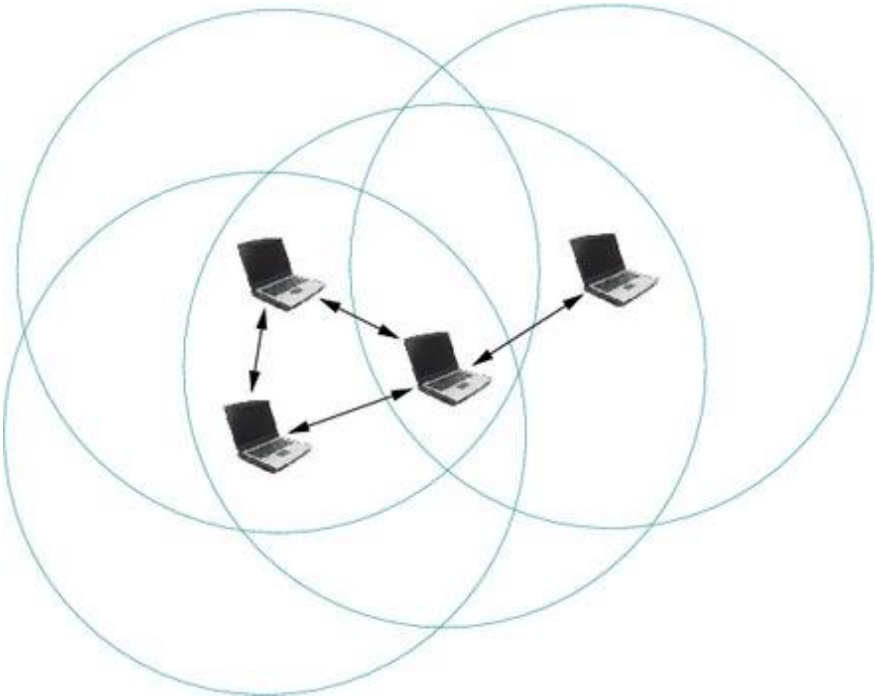
Todo o tipo de comunicação entre os dispositivos de rede ocorre através de um conjunto de regras e procedimentos que tornam possível a comunicação entre os dispositivos de rede, tais regras e procedimentos são chamados de protocolo de rede. De fato, em redes e comunicação entre computadores, um protocolo de rede é uma especificação formal que define como os nós devem comporta-se ou se comunicar uns com os outros. Dentre outras coisas, os protocolos de rede definem formatação, integridade e transmissão de dados entre os dispositivos da mesma rede (GALLO; HANCOCK, 2003).

Por sua vez o protocolo de rede é que define como a rede irá funcionar e também é a linguagem utilizada pelos dispositivos de rede, de uma forma que eles possam se entender entre si e trocar informações, para que isto ocorra todos na rede devem utilizar o mesmo protocolo (COZER, 2006).

3.2 Rede móvel sem fio *Ad Hoc*

Uma Rede Móvel Sem Fio *Ad Hoc* é formada por um conjunto de nós móveis, capazes de se comunicar com seus vizinhos por difusão (*broadcast*) ou conexões Ponto A Ponto (*Point-To-Point ou P2P*), sem a necessidade de qualquer infraestrutura fixa de suporte (LIMA, 2013). A rede *Ad Hoc* se difere dos outros tipos de redes sem fio, por não possuir um nó ou terminal principal, sendo assim, todo o nó possui a capacidade tanto de rotear dados quanto o de ser um dispositivo que recebe tais dados ou podendo exercer as duas funções ao mesmo tempo como apresentado na Figura 3. Para cada nó existente provavelmente o mesmo estará conectado com mais de um nó.

Figura 3 - Redes sem fio não estruturada (*Ad Hoc*)



Fonte: Askoura (2015, p. 3)

Existem diversos benefícios na utilização das redes *Ad Hoc*, sendo que algumas dessas vantagens consiste em não haver necessidade de instalação de uma infraestrutura, a mobilidade e a auto-organização dos *Mobile Ad Hoc Networking* (*MANET*). Além disso, uma *MANET* é um sistema autônomo

composto por plataformas móveis, ou nós, que não depende de nenhuma infraestrutura para operar.

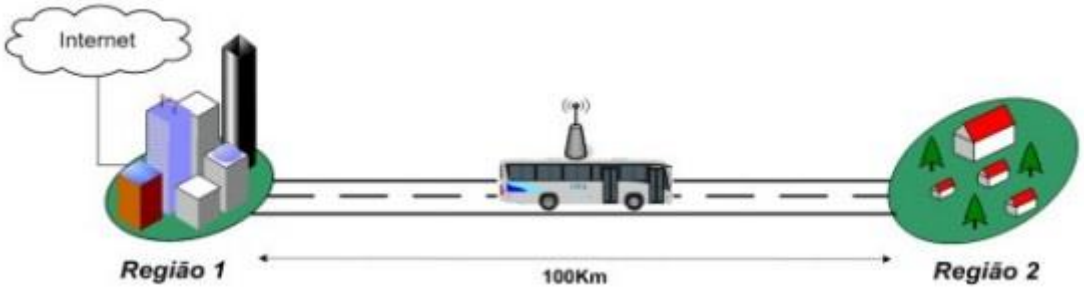
Devido a tanta facilidade de utilização e auto adaptação deste tipo de rede, existem dezenas de aplicações eficientes, nas quais, esta rede pode ser aplicada, iremos tomar como exemplo, um estudante utilizando algum dispositivo eletrônico com capacidade de se conectar a uma rede *Ad Hoc* para poder participar de uma aula interativa ou até mesmo equipes de resgates e salvamento em regiões com impossibilidade de estabelecer conexões através de outros meios (REZENDE, 2004). Portanto, caso dois nós que não são vizinhos queiram se comunicar, os outros nós móveis agem como roteadores encaminhando pacotes de dados até o destino. Dessa forma, uma rota entre esses dois nós é estabelecida para que a comunicação *P2P* se torne possível (BEZERRA, 2009).

No entanto, atualmente um dos maiores desafios nas redes móveis é manter a escalabilidade, confiabilidade e robustez do algoritmo, pois em algumas situações especificas são grandes os problemas causados por congestionamento de informações nesse processo. Além disso, as redes *Ad Hoc* não seguem um padrão, podendo variar rapidamente conforme a variação dos nós. Atualmente, existem vários protocolos de roteamento implementados para as redes sem fio infraestruturada, porém, são limitados quanto a algum a algum aspecto.

3.3 Redes DTN

Uma rede *Delay–tolerant networking (*DTN*),* ou Rede Tolerante a Atrasos e Desconexões, é uma rede *Ad-Hoc*, ou *MANET (Mobile Ad-Hoc Network)*, composta por conexões sem fio entre os nós, que se auto conFigura de acordo com a inserção e remoção de novos nós em sua estrutura. Além disso, consegue comportar se de forma adequada mesmo com desconexões repentinas ou atrasos na transmissão de dados entre dois nós (TABARELLI et al, 2009).

Figura 4 - Redes DTN



Fonte: Cruz e Assakura (2013, p. 11)

Além disso, a arquitetura DTN também provê um processo de gerenciamento que é responsável, por exemplo, pela definição da rota que cada mensagem irá utilizar para chegar a seu destinatário, como exemplo em uma linha de ônibus apresentada na Figura 4, qual rota a mensagem deve seguir da região 1 para chegar até a região 2. Com isso, os agentes remetentes e receptores de informação são aplicações locais. Cada nó possui um armazenamento para manter a informação enviada a ele, ou que está sob sua custódia, assim sendo, as redes DTN podem utilizar uma grande variedade de protocolos de entrega diferentes durante a transmissão dos dados entre os seus nós intermediários ou finais.

3.4 Protocolo de roteamento

A seguir será introduzido alguns conceitos dos principais protocolos de redes DTN para o roteamento de dispositivos móveis que definem as regras de envio de mensagens para a comunicação dos nós que servem como roteadores para redes não estruturadas, sendo assim, cada protocolo deve fornecer aos roteadores basicamente as mesmas informações, tais como: destino, o endereço de origem de roteamento, uma lista de caminhos, o melhor caminho e uma maneira de atualização para redes não estruturadas que servirão de base para o desenvolvimento deste trabalho.

3.4.1 Epidêmico

Vahdat e Becker (2000) propuseram um protocolo de roteamento estocástico *Epidêmico* para DTN, no qual o nó de origem difunde a mensagem para todos os seus vizinhos e cada um destes por sua vez repassa a mensagem para seus vizinhos. Deste modo, a mensagem é repetida até encontrar o destinatário ou pelo menos até que o tempo de vida termine. Neste algoritmo, as mensagens são repassadas para o maior número de nós possível, como em uma epidemia. Sendo assim, possui a menor latência e a maior taxa de entrega dentre os protocolos DTN, considerando um cenário ideal, onde não existem restrições de banda, energia e *buffer*. Este protocolo possui uma grande desvantagem, um grande número de informações é trocado e por isto uma grande quantidade de recursos da rede é utilizada. Em cenários onde existem restrições de energia, banda ou *buffer* este algoritmo pode apresentar queda no seu desempenho (ALMEIDA, 2011).

Este modelo de protocolo de roteamento inunda rapidamente a rede com mensagens, porém com custos bastante elevados. Devido à alta troca de mensagens e a grande quantidade de mensagens duplicadas na rede, o algoritmo tende a congestionar a rede, com isso pode ser que venha a ocasionar o estouro de *buffer* dos dispositivos (LIMA, 2013).

3.4.2 Prophet

O protocolo de roteamento probabilístico utilizando históricos de encontros e transitividade *PROPHET* (*Probabilistic Routing Protocol Using History of Encounters And Transitivity)* é um algoritmo probabilístico de roteamento para redes DTN que explora a não aleatoriedade nos movimentos dos nós em uma rede, esperando assim melhorar a entrega de agregados mantendo baixo o consumo de recursos. Seu funcionamento é baseado no *Epidêmico*, porém é utilizada uma métrica de previsibilidade de entrega para selecionar os nós que irão repassar os agregados (ALMEIDA, 2011).

A diferença entre o algoritmo *PROPHET* e algoritmo *Epidêmico* é que agora existe uma informação extra para cada mensagem indicada na lista. Essa informação corresponde à probabilidade de cada nó a entregar

mensagens para um destino conhecido *b (P (a, b)* ∈ [0, 1]). O valor de *P (a, b)* aumenta sempre que a e b se encontram. Se a e b deixam de se encontrar frequentemente, *P (a, b)* diminui à proporção que o tempo transcorre. Esse tempo é controlado por uma constante *k* denominada constante de envelhecimento, que corresponde ao número de unidades de tempo transcorridas desde a última vez que a métrica foi atualizada. A probabilidade de entrega também possui uma propriedade transitiva, que se baseia na seguinte observação: se um nó a encontra um nó b frequentemente, e o nó b encontra frequentemente um nó c, logo o nó c provavelmente é um bom nó para encaminhar mensagens destinadas para a (OLIVEIRA, 2008).

3.4.3 Spray and Wait

O protocolo de roteamento *Spray and Wait* consiste em uma melhoria do protocolo *Epidêmico*, por meio de duas etapas:

- A primeira etapa o *Spray*, realiza a distribuição N vezes de uma mensagem para vários nós vizinhos;
- E a segunda, a *Wait* (estado de espera), efetua a parada da infestação e paralização dos nós que receberam cópias das mensagens durante a fase inicial, e por fim espera se que os nós com as cópias das mensagens façam a entrega diretamente aos nós de destino.

A etapa inicial conhecida como *Spray* possui dois tipos de distribuição de mensagens por meio de espalhamento binária e a normal. Na distribuição binária o nó que deu origem a mensagem inicialmente começa com N cópias da mesma mensagem e passa a distribui-las conforme encontra os nós vizinhos, e os entrega N/2 e mantém N/2 mensagens até que N seja igual que um, onde só haverá entrega caso o nó destinatário seja encontrado. Já o espalhamento normal realiza a distribuição das mensagens, até que N seja igual a um (FERNANDES et al, 2009). Durante o desenvolvimento deste algoritmo, pode se observar um dos melhores resultados quando a rede

apresenta alta mobilidade, porém a infestação é eficaz apenas no início (BATISTA, 2011).

3.4.4 Primeiro Contato

O protocolo de roteamento *Primeiro Contato* (*FC - First Contact*) é considerado um dos protocolos com maior facilidade de implementação. Esse protocolo transfere as mensagens uma única vez. Tal transferência funciona da seguinte forma, o nó remetente envia a mensagem para o primeiro nó que aparecer estabelecendo um tipo de contato inicial agora no próximo passo seria repetir o passo anterior até que seja encontrado o destinatário.

O protocolo de roteamento *Primeiro Contato* alcança uma taxa de entrega maior do que o roteamento contato direto, já que o algoritmo permite que a mensagem seja enviada através de caminhos com múltiplos saltos (OLIVEIRA, 2008). No entanto, levando em consideração o desempenho de outros protocolos como o *Epidêmico*, mesmo utilizando tais métricas, o protocolo *Primeiro Contato* sofre as baixas taxas de entrega.

3.5 Regra de borda

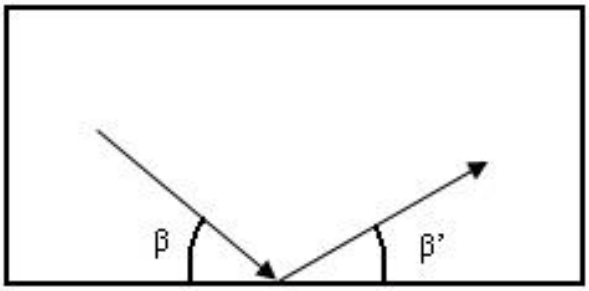
As simulações realizadas pelo simulador *SimDTN* são realizadas em uma região limitada, portanto, os movimentos são realizados somente dentro de uma área de simulação. No entanto, alguns modelos de mobilidade permitem que o nó alcance as bordas da área de simulação, sendo assim, as regras de borda são definidas para decidirem a ação a ser tomada quando algum dos nós atingirem o limite da borda. A seguir serão descritas as principais regras de borda: *Bounce*, *Delete And Replace* e *Wrap Around*.

3.5.1 Bounce

A regra de borda *Bounce*, ou rebatida, reflete o movimento do nó móvel de volta para dentro da área de simulação. Desta forma, a nova trajetória continuará dentro do retângulo limitador estipulado como fronteira da região de simulação. O novo sentido da velocidade, após a rebatida, é caracterizado por dois componentes: o ângulo de incidência β e o módulo da velocidade V do nó quando o mesmo atinge a borda (ANTAS, 2006).

O ângulo de reflexão β será – β’ e o módulo da velocidade se manterá V, como ilustrado na Figura 5.

Figura 5 - Regra de Borda *Bounce*



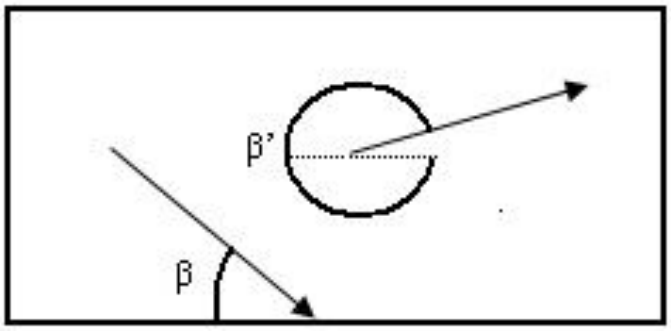
Fonte: Antas (2006, p. 3)

3.5.2 Delete and Replace

Na regra *Delete And Replace*, quando um NM (Nó Móvel) atinge a borda, este é retirado da área de simulação e inserido novamente em uma posição de coordenadas (x; y) aleatoriamente escolhidas, com um novo ângulo de direção do movimento. Essa regra tem a característica de representar a saída do NM da área de simulação que, às vezes, é uma característica realista. Entretanto, uma característica não realista, é colocar o mesmo NM aleatoriamente numa posição qualquer da área, simplesmente para garantir que o número de nós da rede simulada permaneça constante (MENDES et al, 2004).

O ângulo de reflexão β será – β’ e o modulo da velocidade se manterá V, como ilustrado na Figura 6.

Figura 6 - Regra de Borda Delete And Replace

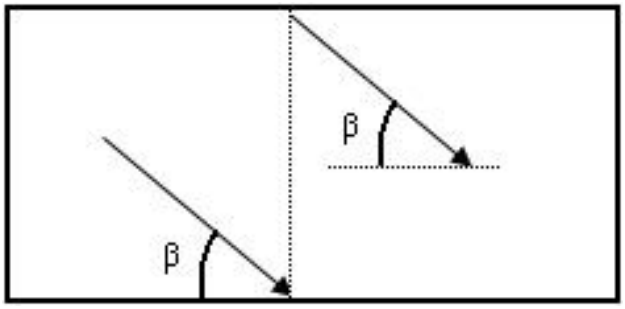


Fonte: Antas (2006, p. 3)

3.5.3 WRAP AROUND

A regra de borda *Wrap Around*, ou dar a volta, o nó móvel que alcança uma borda tem seu movimento espelhado na borda oposta. Este espelhamento preserva tanto a direção e o sentido, quanto o módulo de sua velocidade. A Figura 7 ilustra o método, que tem como objetivo melhorar a técnica de posicionamento da regra *Delete And Replace* (ANTAS, 2006), na qual o nó era inserido aleatoriamente numa posição qualquer da área de simulação, não levando em consideração algumas informações anteriores como ângulo e velocidade.

Figura 7 - Regra de Borda *Wrap Around*



Fonte: Antas (2006, p. 3)

3.6 Mobilidade dos nós

Os modelos de mobilidade são fatores comportamentais de movimentações dos componentes da rede, sendo assim, são empregados na avaliação de desempenho de redes móveis. Um bom modelo deve tentar imitar os padrões de movimentação de uma estação móvel real, de modo que suas

variações de velocidade e direção ocorram semelhantemente à maneira como elas ocorrem no mundo real (HARGREAVES, 2006). Deste modo, a arte da modelagem de movimentação consiste na criação de modelos que possuam algumas características esperadas de movimentação de usuários reais.

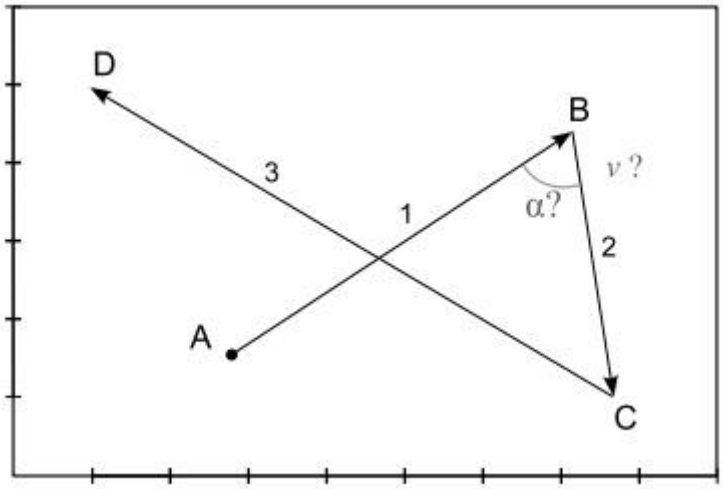
Para realizar a análise de desempenho de redes não estruturada é necessário utilizar algum padrão de movimentação. Uma alternativa é a utilização da mobilidade real (*trace*), que é o padrão obtido através da observação de um sistema realista. No entanto, conseguir os *traces* pode ser muito difícil, devido a quantidade de componentes elevada e tempo de observação necessários para obter um padrão consistente. A outra possibilidade é a utilização de padrões sintéticos, os quais definem comportamentos da movimentação para os dispositivos (DELAMARE, 2008).

A seguir os principais modelos de mobilidade existentes e que serão utilizados neste trabalho.

3.6.1 RANDOM WALK

O modelo *Random Walk* foi originalmente proposto para simular as partículas na área da física, cujo os movimentos são imprevisíveis. Tal modelo possui as seguintes características: normalmente o primeiro elemento é escolhido uniformemente dentro do espaço de modo aleatório. Em seguida, a próxima velocidade é escolhida dentro do intervalo [Velocidade mínima, Velocidade máxima] e a posição é escolhida por meio da posição atual mais uma variável aleatória entre o intervalo [0, 2π] cujo valor é escolhido em função de uma determinada distribuição de probabilidade e independente do estado atual (HARGREAVES, 2006).

Figura 8 - Movimentação no *Random Walk*



Fonte: Delamare (2008, p. 29)

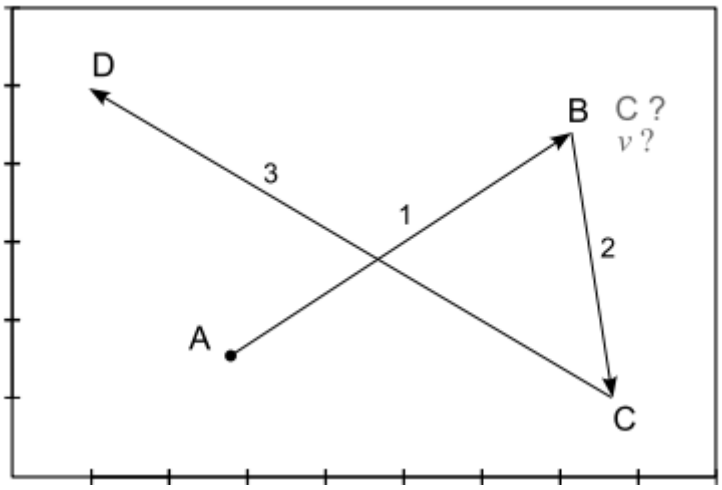
Sendo assim, a direção e velocidade do movimento num novo instante de tempo não tem relação nenhuma com os valores de instantes anteriores, ou seja, é um modelo de mobilidade sem memória. Por isso, esse modelo pode gerar um comportamento não realístico, por exemplo com mudanças bruscas de direção, paradas abruptas e acelerações bruscas no movimento do NM como apresentado na Figura 8 (CAMPOS et al, 2007).

Portanto, o modelo de mobilidade *Random Walk* não possui informação sobre o estado anterior, logo as decisões sobre o estado futuro são independentes do estado anterior. Ou melhor dizendo, a velocidade atual não interfere na velocidade futura. Desta forma, qualquer tipo de mudança é recebido de forma brusca, o que em um cenário real seria percebida de forma incremental e suave.

3.6.2 RANDOM WAY POINT

Atualmente é o modelo de mobilidade mais utilizado para avaliar protocolos de roteamento em *MANETs* devido a sua simplicidade. Este modelo foi proposto por Boudec e Vojnovic (2006).

Figura 9 - Movimentação no *Random Waypoint*



Fonte: Fonte: Delamare (2008, p. 31)

A movimentação segundo esse modelo é feita da seguinte maneira: inicialmente o nó escolhe um ponto aleatório qualquer na área de simulação para o qual o nó irá se mover com velocidade constante escolhida de forma aleatória. O destino e velocidade de um nó são escolhidos independentemente dos outros nós. Ao chegar ao destino, o nó permanece parado por um determinado intervalo definido como “tempo de pausa”. Se esse intervalo for zero, temos uma mobilidade contínua (CAMPOS et al, 2007). Após esse tempo, o nó novamente escolhe um destino e velocidade, e o processo se repete até o final do tempo de simulação como apresentado na Figura 9.

Entretanto, este modelo não é bastante representativo, pois apresenta características aleatórias, desconsiderando a existência de algumas características restritivas e outras de dependências espacial e temporal presentes em cenários reais.

3.6.3 SMOOTH RANDOM

O modelo de mobilidade *Smooth Random* é um modelo de mobilidade que leva em consideração a dependência temporal da velocidade do nó que é calculado segundo uma distribuição de probabilidade em que se considera que os nós possuem velocidades preferenciais e que a frequência de mudança do

módulo da velocidade segue de modo continuo (CAVALCANTI, 2009). Sendo assim, propõe que as mudanças de direção e velocidade sejam realizadas de forma incremental e suave. Enquanto, o modelo *Random Way Point* não leva em consideração tais comportamentos.

Sendo assim, no modelo *Smooth Random* a distribuição da velocidade ocorre da seguinte forma: alguns valores preferenciais para velocidade têm alta probabilidade, ao mesmo tempo que uma distribuição uniforme é assumida para o restante do intervalo.

3.7 Métricas de conectividade

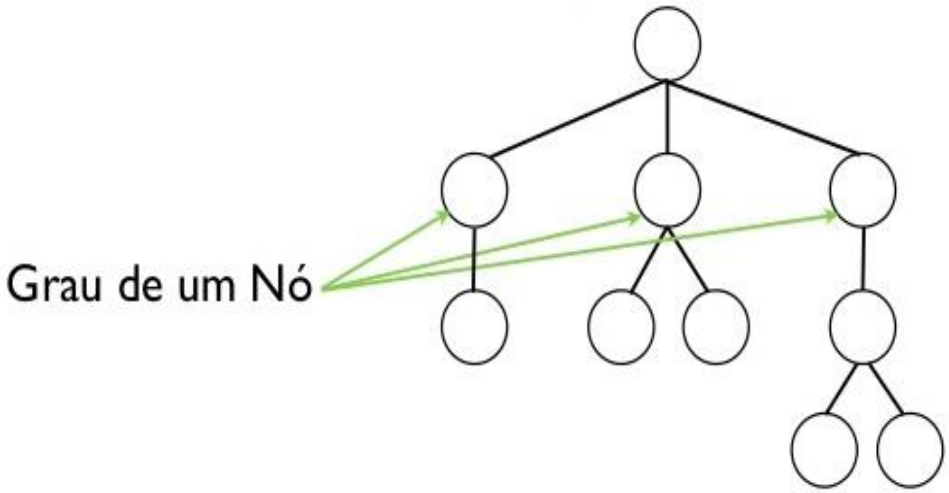
As métricas de conectividade são características capturadas da topologia de uma rede em um cenário de mobilidade especifico. Além disso, as métricas permitem realizar uma análise detalhada da conectividade e topologia da rede, e verificar o nível de confiabilidade do uso de mobilidade sintética na representação da movimentação real. Sendo assim, as métricas de conectividade utilizadas na metodologia foram proposta para caracterizar a conectividade e compara lá neste trabalho são grau de um nó, número de nós isolados, número de componentes conexas e duração do enlace (BETTSTETTER, 2001).

Dessa forma, a conectividade entre os nós é estabelecida através do roteamento feito pelos próprios dispositivos móveis, pois nas redes sem fio *Ad Hoc* não existe um ponto de acesso fixo para que seja realizada a conexão com os mesmos. Deste modo, os dispositivos móveis vizinhos são responsáveis por retransmitir entre si mensagens até que seja entregue ao destinatário (LIU et al, 2005). Logo, o método para selecionar os nós intermediários, assim como a resposta às quebras de enlaces nas rotas são características que diferem os protocolos de roteamento *Ad Hoc* existentes. Para que ocorra um melhor entendimento do comportamento da mobilidade do ponto de vista da conectividade é necessário que ocorra o controle de topologia, no qual pode ser estudada a escolha de um raio de alcance de forma a maximizar o uso da bateria dos dispositivos a rede conectada.

3.7.1 GRAU DE UM NÓ

É o número de vizinhos que são alcançáveis diretamente a partir do nó de origem, ou melhor dizendo o Grau de um Nó representa a quantidade de vizinhos de um nó, como apresentado na Figura 10.

Figura 10 - Grau de um nó



Fonte: Martins (2013, p. 13)

Uma das principais características de um nó em uma rede qualquer é quantidade de nós vizinhos desse nó. Caso o nó não possua nenhum vizinho é chamado de nó isolado, logo o grau do nó é zero, pois ele não pode trocar mensagens, sendo assim, não possui utilidade para a rede.

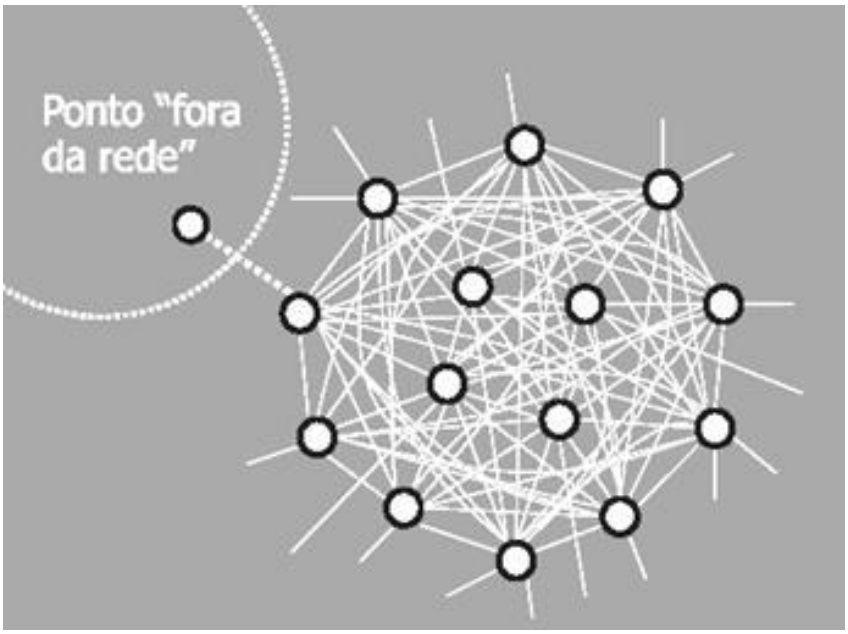
Uma das formas encontrada para se contornar esse problema de nós isolados é feito através do aumento do alcance do raio de certa forma a se controlar a quantidade de vizinhos (HERNANDEZ, 2015).

3.7.2 NÚMERO DE NÓS ISOLADOS

Os Números de Nós Isolados são os nós que não possuem nenhuma conexão com outros nós da rede como o ponto fora de rede apresentado na Figura 11. A existência de Nós Isolados é certamente uma característica indesejável em redes sem fio, pois significa que aquele nó não pode trocar

informações com nenhum outro nó, e muito menos encaminhar mensagens de outros nós.

Figura 11 - Nós Isolados



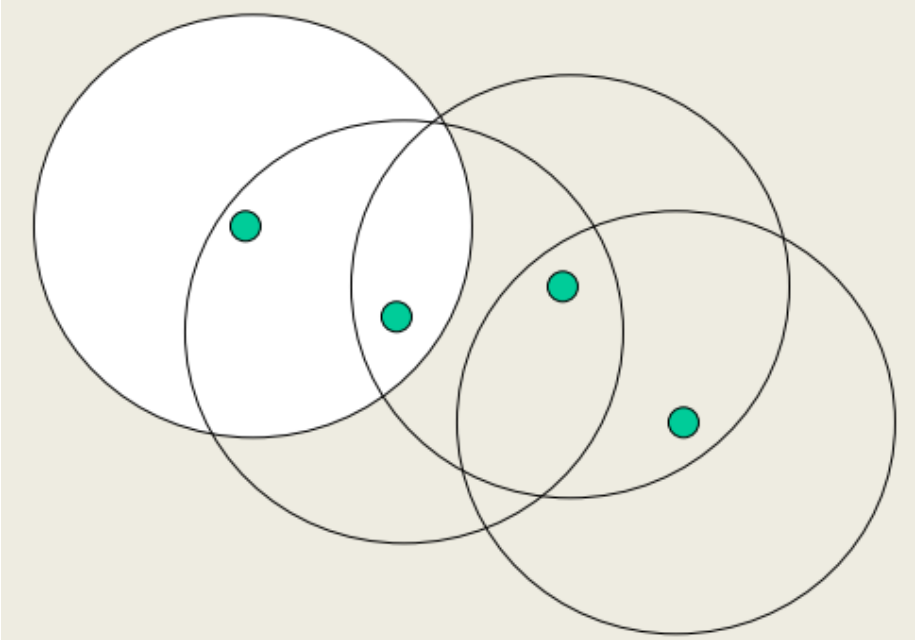
Fonte: Percília (2007)

3.7.3 NÚMERO DE COMPONENTES CONEXAS

Para que um grafo seja considerado conexo é necessário que para cada par de nós exista um caminho entre eles e caso isso não seja verdade chamamos de grafo desconexo.

Em uma rede conexa todos os nós podem se comunicar por meio de um ou mais saltos. Sendo assim, um componente conexo em um grafo é um conjunto de nós tal que todos os nós conseguem alcançar todos os demais nós, como apresentado na Figura 12. Frequentemente é desejável ter uma rede conectada, por exemplo, algumas avaliações de desempenho de protocolos de roteamento assumem uma topologia conectada (BEZERRA, 2009).

Figura 12 - Número de Componentes Conexas

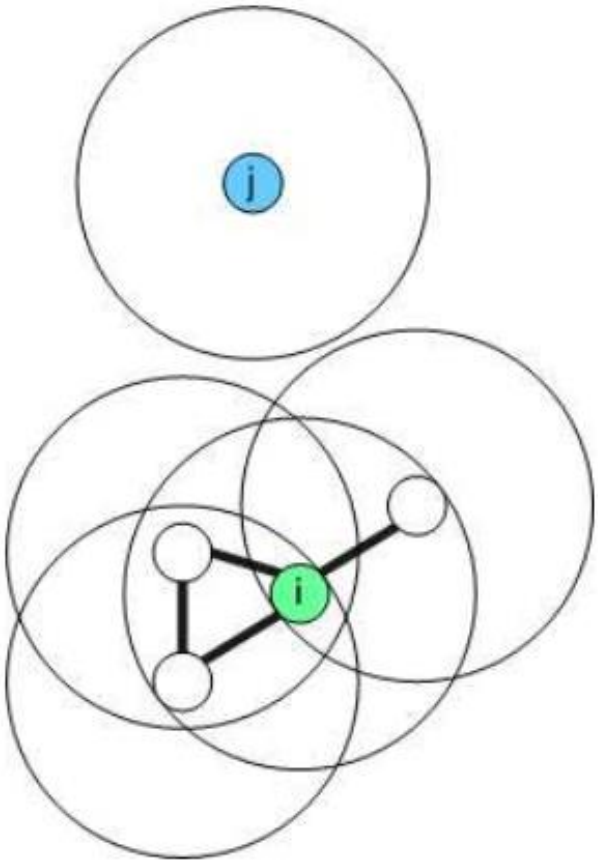


Fonte: LOPES (2014, p. 5)

3.7.4 DURAÇÃO DOS ENLACES

Para que ocorra a comunicação entre os nós é necessário que esses nós sejam conexos como os nós ilustrados na Figura 13. No entanto, devido a mobilidade exercida pelas redes móveis sem fio, os caminhos entre os nós mudam constantemente, deste modo, tornam os antigos caminhos inválidos e forçam os protocolos de roteamento a diminuir a eficiência da rede. Para a ilustração da Duração de um Enlace, tome como exemplo dois nós i e j, no instante t1, a duração do enlace é definida como o tamanho do maior intervalo de tempo [t1, t2] durante o qual os dois nós estão dentro do raio de alcance um do outro. Além disso, os nós não estão dentro do raio de alcance um dos outros no instante anterior a t1 e posterior a t2 (BEZERRA, 2009).

Figura 13 - Duração do Enlace



Fonte: BEZERRA (2009, p. 48)

4 REVISÃO BIBLIOGRÁFICA

Constam na literatura diversos trabalhos relacionados as redes sem fio. A seguir, é apresentada uma revisão bibliográfica dos principais trabalhos sobre esse assunto.

Pontes (2009) demonstrou que durante as simulações o modelo *Random Waypoint* (RWP) apresenta uma falha caracterizada por não atingir um estado estável, uma vez que a velocidade instantânea média das simulações tende a zero com o passar do tempo, para certos valores de parâmetros. Portanto, a fim de sanar este problema propôs uma alteração na escolha da função uniforme de densidade de probabilidade da velocidade dos nós da simulação do RWP por uma distribuição baseada em *BETA* (α, β). No entanto, os resultados obtidos das simulações utilizando o simulador de redes *JiST/SWANS* não corresponderam ao demonstrado na análise matemática e como em trabalhos anteriores que utilizaram o simulador *NS-2*. Concluiu-se, assim, que o *JiST/SWANS* não apresenta resultados suficientemente cofiáveis e deve ser refinado.

Spyropoulos, Psounis e Raghavendra (2005) apresentaram uma melhoria do algoritmo *Epidêmico* que consiste em administrar de forma significativa o envio de mensagens para diminuir o número dos envios dos esquemas baseados em inundação de pacotes e deve possuir o melhor desempenho relacionado aos cenários reais.

O método *PROPHET* proposto por Lindgren, Doria e Schelén (2003) realiza cálculos de probabilidade para determinar a entrega de uma mensagem ao nó de destino. A probabilidade é calculada por meio de observações de um histórico de contatos, a entrega só é realizada depois de se obter uma alta probabilidade de entrega ao destinatário.

O trabalho de Oliveira (2008) apresenta e valida uma proposta de roteamento probabilístico para redes tolerantes a atrasos e desconexões capaz de lidar com a incerteza dos contatos previsíveis, e garantir uma alta taxa de

entrega de mensagens com um baixo custo em transmissões de réplicas e espaço ocupado nos *buffers*.

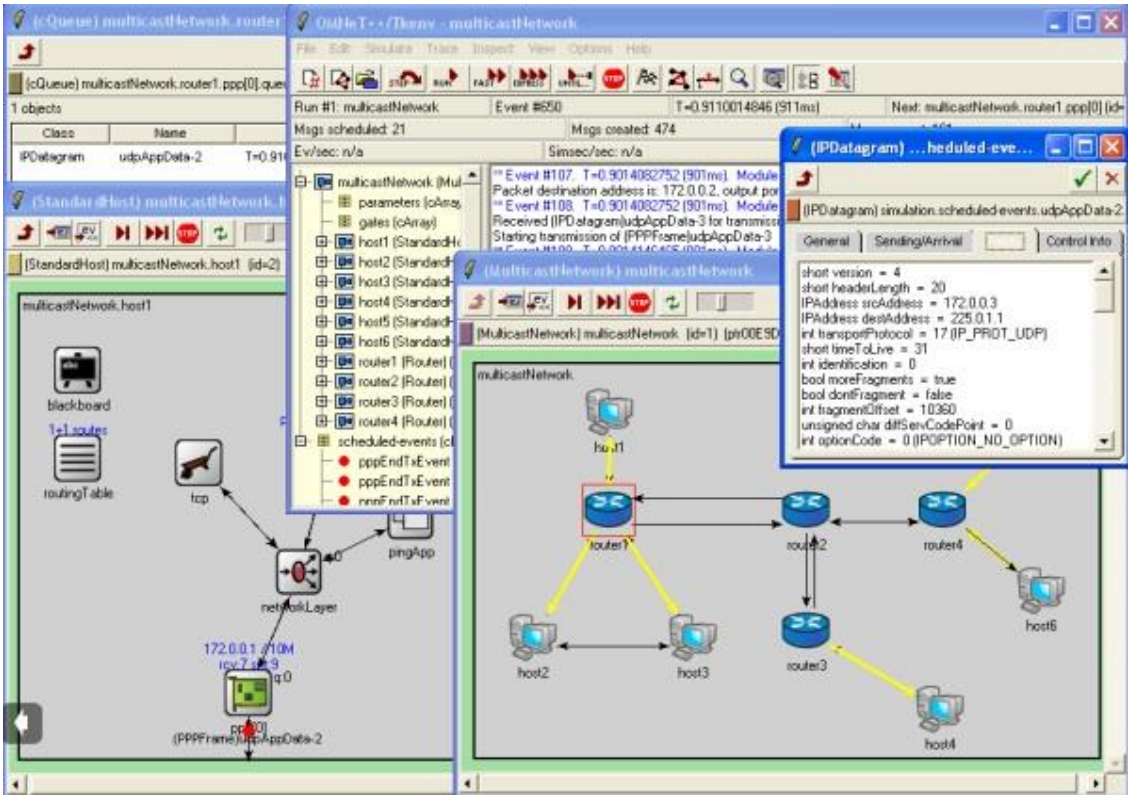
Batista (2011) avalia os principais protocolos de roteamento utilizados nas redes DTNs, levando em consideração o tempo de vida da rede como a principal medida de desempenho. Além de analisar a quantidade de mensagens descartadas, retransmitidas e geradas na rede.

Segundo Varga (2013), o *OMNeT*++ é um framework simulador de eventos modular baseado em eventos discretos e orientado a objeto e, apesar de oferecer recursos para simulação de redes, não está limitado a somente esta área. Todo modelo que puder ser representado por módulos trocando mensagens, pode ser modelado no OMNeT++. Desenvolvido em C++ e distribuído por meio do licenciamento *open source*. Além disso, o simulador apresenta uma série de recursos referentes à modelagem de tráfego e tratamento de filas em redes de telecomunicação, possibilita a verificação da modelagem de multiprocessamento (SOARES, 2009), sendo assim, tais recursos são visíveis na Figura 14. Deste modo, é muito difundido e utilizado por toda a comunidade científica, bem como em ambientes industriais e educacionais.

Sendo assim, em uma segunda análise o simulador *OMNeT*++ possui um modelo básico de um switch composto por uma camada básica de enlace e uma unidade de encaminhamento chamada *relayUnit*, que recebe e envia as mensagens segundo sua Tabela de encaminhamento, através de um *Rapid Spanning Tree Protocol* implementado. Por ser tratar de um framework dividido em módulos o OMNeT++, utiliza o módulo INET que por sua vez implementa os modelo de mobilidade *Random Walk, Random Waypoint, Random Direction, Gauss-Markov, City Section* e os protocolos de roteamento de redes IPv4, IPv6, PPP, MPLS, 802.11, SCTP, TCP, SCTP, UDP, dentre outros.

No entanto, funcionalidades da norma IEEE802.1Q-2005 define que os elementos da rede devem possuir o mesmo cabeçalho (identificador), sendo assim, redes virtuais locais e etiquetas de prioridade, não estão presentes nesse modelo (TESSINARI, 2009).

Figura 14 - Simulador *OMNET++*



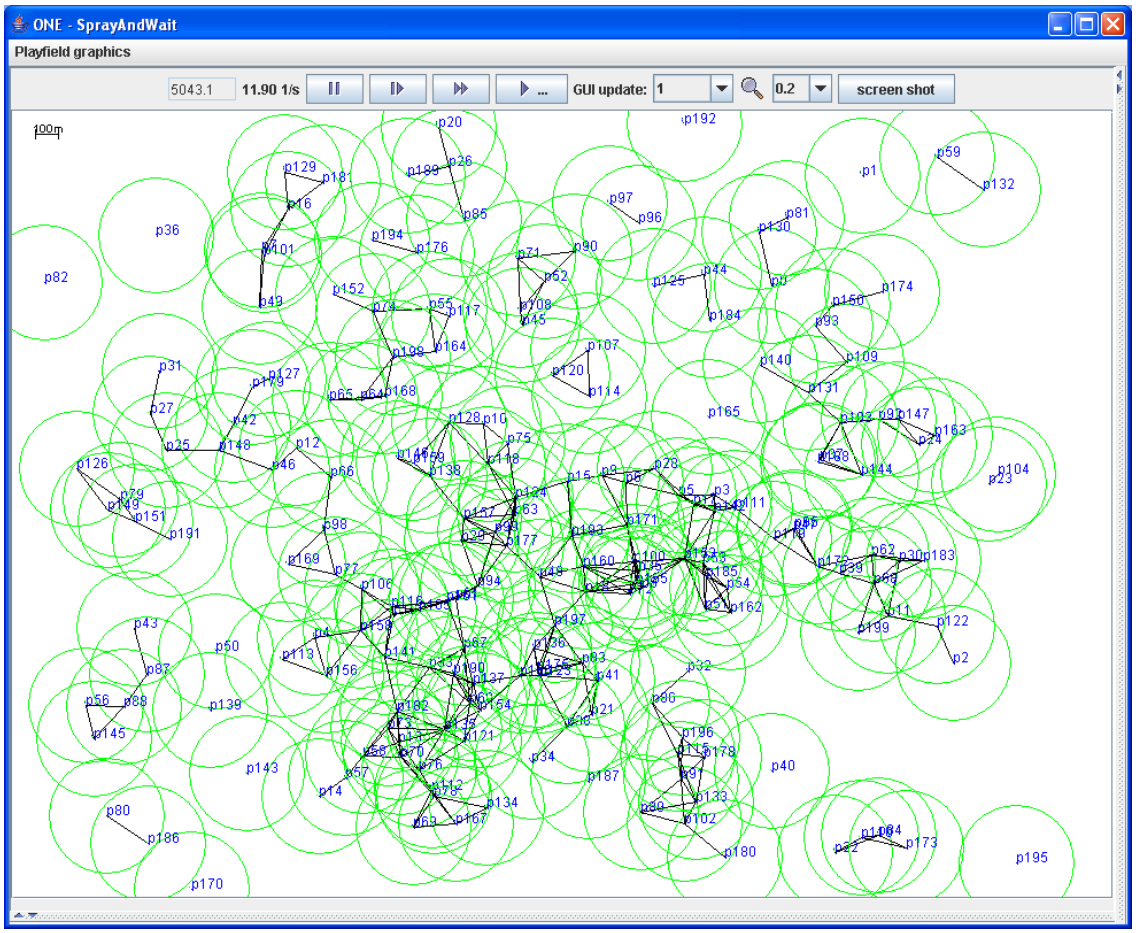
Fonte: VARGA (2013)

O *Opportunistic Network Environment* (*ONE*) é uma ferramenta de simulação de código aberto desenvolvida por Keranen e Ott (2009) na linguagem de programação *Java, que* simula de forma realista redes DTN. Deste modo, *e*ste simulador possui as ferramentas necessárias para o funcionamento de uma rede tolerante a atrasos e as fornece de forma poderosa para simular situações especificas de movimentações de nós em caminhos predeterminados ou aleatórios, sendo possível personalizar a execução de acordo com a necessidade. Para o início da simulação é necessário a construção de um cenário de simulação. A propósito um cenário pode ser criado por meio da inserção de diversos elementos como, grupos de nós, eventos, protocolos de roteamento, modelos de movimentação, relatórios e mapas.

Além disso, o simulador *ONE,* por padrão implementa os modelos de mobilidade *Random Waypoint, Random Walk, Random Movement, Map Constrained Random Movement, Huam Behavior Based Movement* e inclui uma estrutura para a criação de modelos de movimento, bem como interfaces para carregamento de dados de movimento exposto na Figura 15. Existem também os protocolos de roteamento *Direct Delivery, Epidêmico, Primeiro Contato, Spray-And-Wait, MaxProp* e *PROPHET*.

Contudo, dentre os inúmeros motivos que levaram o simulador *ONE* ser um dos mais lembrado e mencionado em publicações acadêmicas a respeito de mobilidade de redes móveis, ainda assim, existem alguns obstáculos em seu projeto, tais como: a falta de um gerador de gráficos com as métricas Grau de um nó, duração do enlace e número de nós isolados.

Figura 15 - Simulador ONE



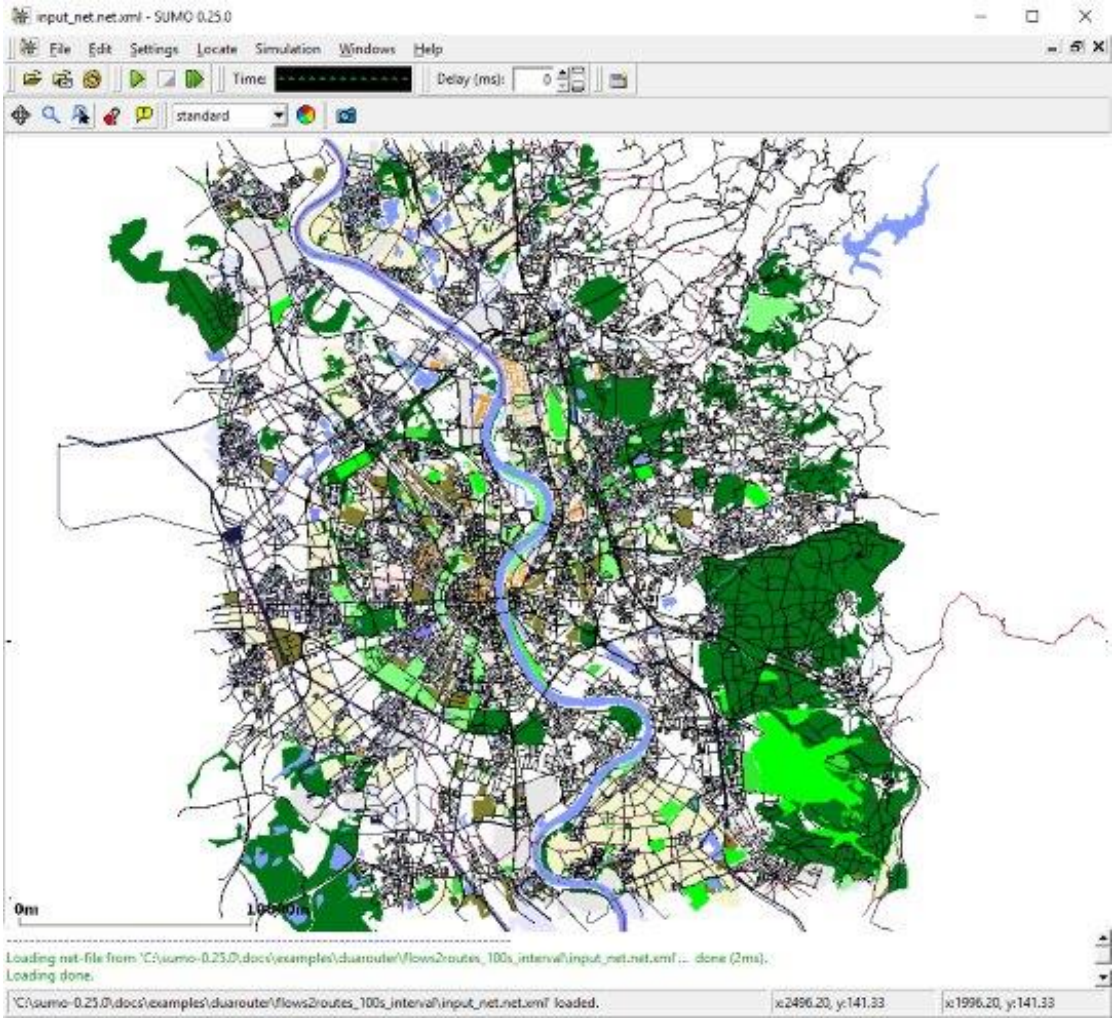
Fonte: KERÄNEN (2009)

De acordo com Behrisch et. al. (2011) o “*Simulation of Urban MObility*", ou "*SUMO*" teve início do seu desenvolvimento no ano de 2000 pelo Centro Aeroespacial Alemão com os objetivos básicos: pouco consumo de memória, facilidade em estender a aplicação e velocidade nas simulações. A qual sua principal razão de desenvolvimento se deu pela necessidade de um simulador de código fonte aberto, de modo, a servir de suporte aos pesquisadores como uma ferramenta com a capacidade de implementar e avaliar, sem a necessidade de obter um tráfego completo de simulação. A simulação é feita de forma microscópica, onde é possível reconhecer veículos, pedestres, transporte público, e também é possível simular a temporização dos semáforos. Sua aplicabilidade é ampla e envolve a avaliação de algoritmos modernos e planos de sinalização semafórica, facilita a análise de emissão de poluentes e permite analisar escolhas autônomas de rotas.

As simulações de tráfego no *SUMO* exigem que a representação de redes rodoviárias e de tráfego necessitam está em um formato próprio para serem inseridas no simulador, ambos têm de ser importados ou gerados usando fontes externas. Em seguida, a utilização se dá através de linhas de comando. Instalar o programa, simular um veículo percorrer uma via, configurar uma malha viária, entre outros, dependem destas linhas de comando. As entradas e saídas do programa são realizadas diretamente em arquivos XML.

Sendo assim, as redes rodoviárias utilizadas no simulador *SUMO* podem ser geradas utilizando aplicativo chamado *Netgen* ou gerado por meio da importação de um mapa rodoviário digital (KRAJZEWICZ et al., 2012), bem como o proposto na Figura 16.

Figura 16 - Simulador *SUMO*

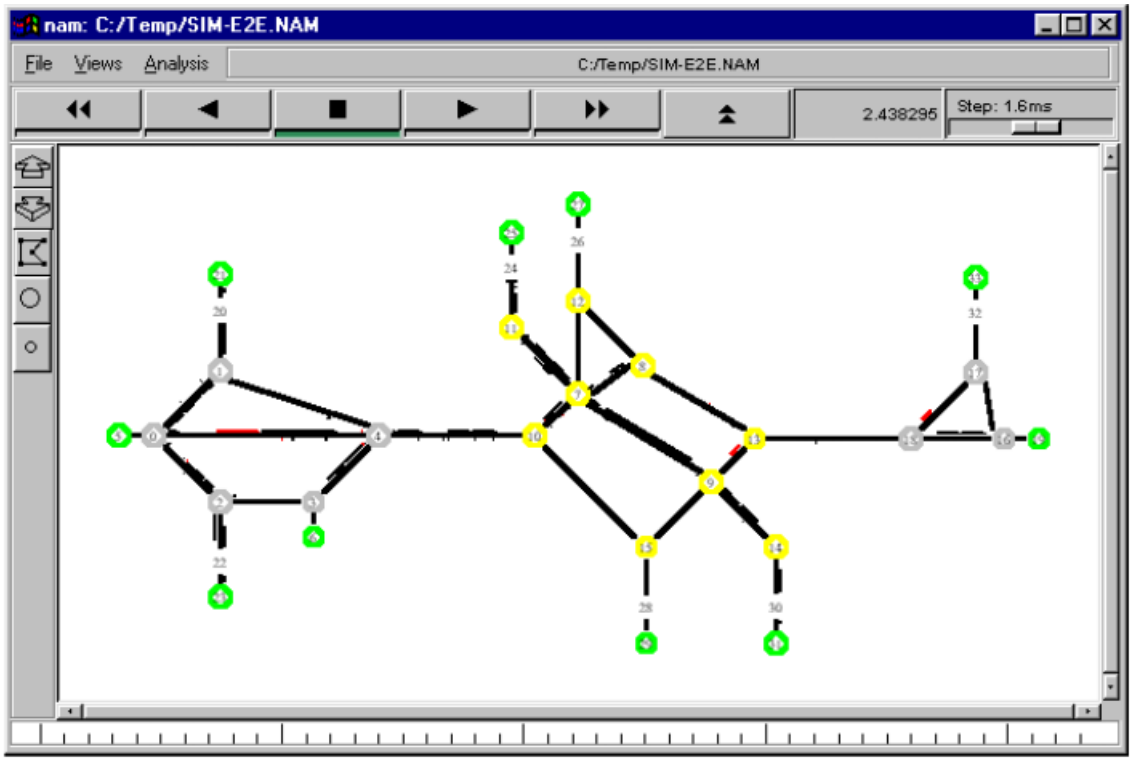


Fonte: Behrisch (2006) adaptado pelo autor

Segundo Fall e Varadhan (2009) o *Network Simulator (NS)* é um *software* utilizado na simulação de redes de computadores, tem sua distribuição gratuita, seu código fonte aberto e escrito em C++ mantido pelo projeto *VINT*, que é uma colaboração entre pesquisadores da Universidade de *Berkeley*, *LBNL, USC/ISI e Xerox PARC* para desenvolvimento desse simulador. Ele dá suporte para vários tipos de roteamento, incluindo bibliotecas para roteamento *unicast, multicast* e *MANETs*. Da mesma forma, podem ser construídos diferentes cenários de rede e variações de tráfegos de pacotes para execução do algoritmo de roteamento, como o cenário apresentado na Figura 17. O NS também fornece implementações para nós de rede, link entre nós, algoritmos de roteamento, protocolos de nível de transporte e alguns geradores de tráfego.

Levando se em consideração esses aspectos o *NS* é um simulador de redes de computadores muito popular nos meios acadêmico por possuir código aberto e por ser utilizado em pesquisas relacionadas a redes móveis. Além disso, suporta os protocolos que podem ser utilizados para redes LAN (*Local Area Network*), MAN (*Metropolitan Area Network*), WAN (*Wide Area Network*). Os protocolos para redes Ad Hoc são AODV (Ad Hoc On Demand), DSR (Dynamic Source Routing) e o DSDV (Destination Sequenced Distance Vector).

Figura 17 - Simulador *NS 2*



Fonte: Kamienski (2014)

O simulador *SimDTN* simula o comportamento de dispositivos móveis dentro de uma área pré-definida de simulação, gerando movimentos dos nós utilizando os modelos de mobilidade aleatórios e roteando mensagens entre os nós por meio de protocolos de roteamento, além do que, o simulador conta uma interface gráfica que facilita e possibilita a visualização da mobilidade e a troca de mensagens em tempo real em sua interface gráfica.

No simulador, ainda é possível que o usuário escolha como parâmetros de simulação: quantidade de nós, tamanho da área, tempo total, raio dos nós, tamanho dos pacotes gerados e a regra de borda. Além do que, as regras de borda implementadas foram: *Bounce, Delete and Replace* e *Wrap Around*. Dessa maneira, os modelos de mobilidade inicialmente implementados são: *Random Waypoint*, *Random Walk* e o Smooth Random.

Com isso, o simulador conta com a implementação dos seguintes protocolos de roteamento: *Epidêmico*, *SPRAY AND WAIT e o PROPHET*. Portanto, as métricas de conectividade utilizadas nas simulações são: grau de conectividade do nó, número de nós isolados, duração do enlace e coeficiente de clusterização.

Em vista dos argumentos apresentados, o simulador possui sua implementação dividida em módulos, facilitando assim inserção de novos módulos ou a alteração de algum parâmetro *default* do simulador. A propósito a Tabela 1, apresenta os parâmetros que inicialmente são carregados no simulador para a execução dos protocolos de roteamento, tais informações foram definidas de forma empírica e podem ser alteradas durante as simulações para melhor precisão de alguma característica da rede *DTN*.

Tabela 1 - Parâmetros dos protocolos de roteamento

Parâmetro	Valor
Buffer(Kb)	4096.0
tempoGeracaoMin(s)	10.0
tempoGeracaoMax(s)	10.0
tamanhoMensagemMin(Kb)	256.0
tamanhoMensagemMax(Kb)	256.0
taxaTransferenciaMin(Kb/s)	1024.0

taxaTransferenciaMax(Kb/s)	1024.0
quantidadeMensagemMin	5.0
quantidadeMensagemMax	5.0

Fonte: Elaborada pelo autor.

Os parâmetros mencionados acima fornecem informações de como os protocolos de roteamento devem agir: tais como o *buffer* que define o quantidade máxima de dados que cada nó suporta encaminhar mensagens para que não ocorra o estouro de buffer durante a simulações, sendo assim, o tempoGeracaoMin(s) e o tempoGeracaoMax(s) determina o tempo utilizado pelo simulador para gerar mensagens, enquanto a taxaTransferenciaMin(Kb/s) e a taxaTransferenciaMax(Kb/s) indicam a qual velocidade as mensagens são transferidas durante o enlace entre os nós, e por último a quantidadeMensagemMin e quantidadeMensagemMax são os parâmetros que restringem o número de mensagens enviadas na rede durante o processo de simulação.

A seguir na Tabela 2, são apresentados os parâmetros para os modelos de mobilidade selecionados pelo usuário.

Tabela 2 - Parâmetros dos modelos de mobilidade

Parâmetro	Valor
tempoTotalSimulacao	50.0
velocidadeMinima	1.0
velocidadeMaxima	100.0
tempoDeslocamento	10.0

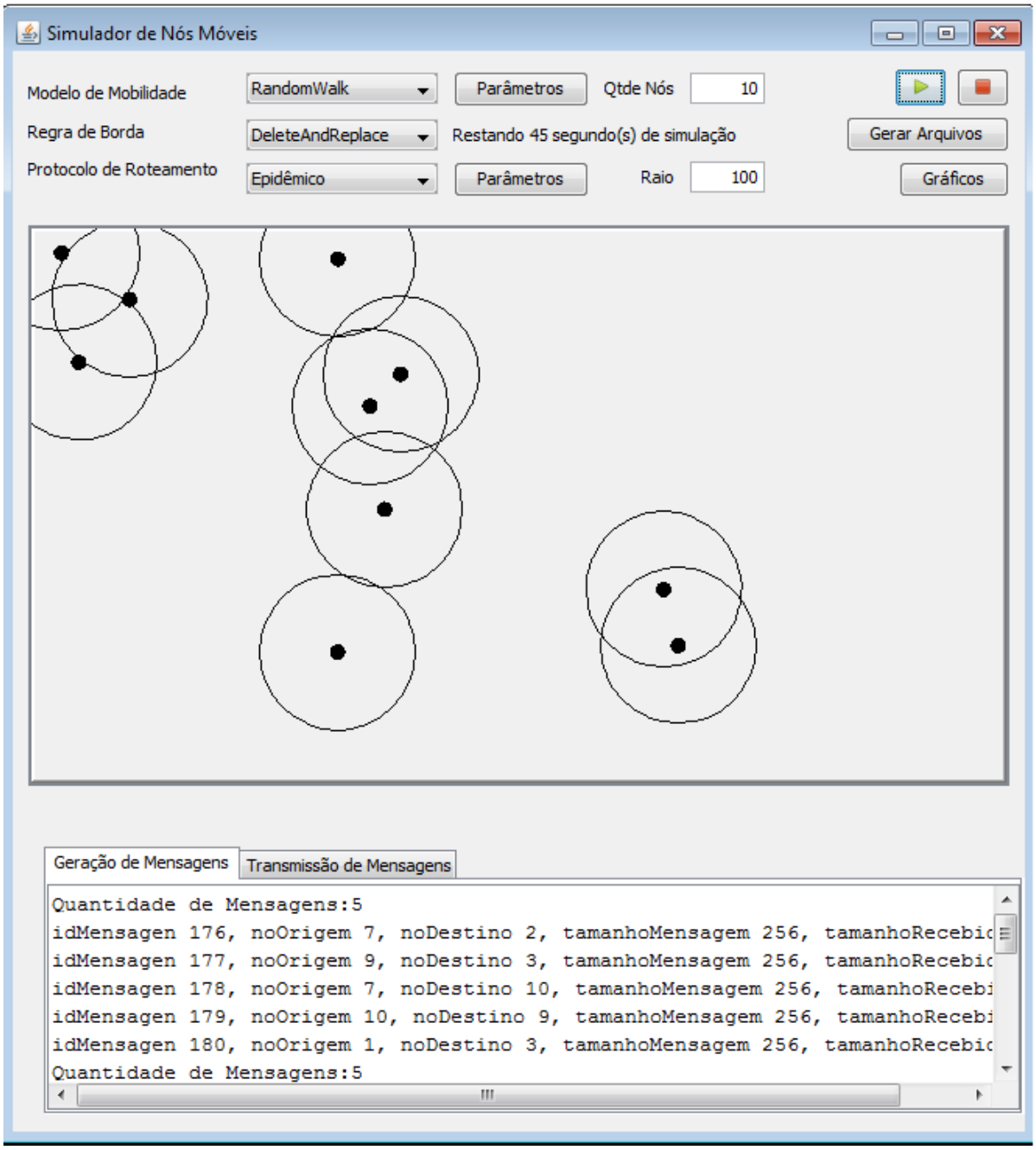
yMaxOrigen	420.0
xMaxOrigen	730.0

Fonte: Elaborada pelo autor.

Os parâmetros mencionados acima fornecem informações de como os modelos de mobilidade devem se mover: tais como o tempoTotalSimulacao que estabelece o tempo total ao qual as simulações serão feitas, a velocidadeMinima e a velocidadeMaxima que indicam a aceleração do nó durante o percurso dentro do tempo de simulação, já o tempoDeslocamento determina o tempo em que o nó deve ser mover para chegar do ponto atual ao seu ponto de destino, por sua vez xMaxOrigem e o yMaxOrigem definem o tamanho máximo da área de simulação.

Logo na Figura 18, é possível visualizar o *SimDTN* em execução. Neste caso, a simulação ocorreu com 10 nós de raio 100, e o protocolo de roteamento utilizado foi Epidêmico e a mobilidade regida pelo *Random Walk* com a regra de borda *Delete and Replace* durante o tempo de execução de 50 segundos.

Figura 18 - SimDNT utilizando o Protocolo Epidêmico



Fonte: *Print Screen* do simulador *SimDTN* em execução

5 METODOLOGIA

A metodologia utilizado neste trabalho pode ser dividida em três partes. A primeira parte consiste na implementação de um módulo do *SimDTN* composto pelo protocolo de roteamento *Primeiro Contato* e a implementação da leitura de opção de entrada de arquivos externo referente a mobilidade dos nós na linguagem de programação *JAVA*. Enquanto a segunda parte aborda a análise das funcionalidades do *SimDTN.* E por fim, a terceira parte ocorre a realização dos testes e a avaliação das funcionalidades do *SimDTN* com as de outros simuladores.

5.1 Desenvolvimento de um módulo para o *SimDTN*

5.1.1 Protocolo Primeiro Contato

Visando à implementação, é realizada uma modelagem do protocolo de roteamento *Primeiro Contato*, de modo que essa etapa transcorra o mais simples possível. Para isso é necessário criar uma classe chamada *FirstContact.java* que estende a classe abstrata Protocolos no interior do pacote designado protocolos. Seguindo esse raciocínio sua implementação é composta pelos métodos *roteamentoDeMensagens*, *geracaoDeMensagens*, *adicionaMensagem, calculaTamanhoTotal e adicionaMensagemRecebida.*

Sendo assim, o principal método mencionado é o de *roteamentoDeMensagens*, o qual inicialmente receberá o tempo total de execução para dar início na simulação do protocolo, e em seguida durante o tempo de simulação o protocolo de roteamento irá realizar a transmissão de mensagens de acordo com a lista dos modelos de mobilidade preenchida na classe Protocolos. Deste modo, o princípio básico do protocolo *First Contact* deve ser respeitado, o de transmissão de uma única mensagem para o primeiro vizinho aleatório.

Por sua vez, o método *geracaoDeMensagens* cria mensagens aleatórias para serem adicionados pelo método *adicionaMensagem* na lista de mensagens do nó. Além disso, o método *calculaTamanhoTotal* realiza a contagem da quantidade de mensagens existentes na lista de um nó. Por

último quando o nó finalmente encontra o destinatário da mensagem, ocorre a adição à uma lista de mensagens entregues por meio do método *adicionaMensagemRecebida*.

5.1.2 Módulo de leitura

De forma geral, a metodologia proposta neste trabalho como alternativa para melhorar a precisão das simulações dos modelos de mobilidade, consiste na implementação de um módulo de leitura para mobilidade real. Para isso é necessário criar uma classe chamada *ModeloRealArquivo.java* que estenda a classe abstrata *modeloMobilidade* dentro do pacote *modeloMobilidade* do código fonte do simulador *SimDTN*.

Também torna se necessário a alteração da interface com o intuito de melhorar a disponibilidade das informações e a fim da inserção do botão Abrir Arquivo, o qual é responsável pela chamada do método *btnArquivoPerformed*, que se localiza dentro da classe FPrincipal.java no pacote *View*. Em outras palavras, esse método é responsável por abrir e ler o arquivo de *traces* no formato de texto e posteriormente inseri-lo em uma lista HashMap, sendo que essa lista é formada por um valor, que neste caso é o ponto e uma chave de identificação do tipo *String* composta pelo tempo de simulação, mais o índice do nó atual.

Além disso, os modelos de mobilidade gerados aleatoriamente ou podem ser extraídos através da utilização de dispositivo GPS, ao qual é possível capturar e identificar as posições de todas as localizações percorridas por um nó, sendo assim, posteriormente é possível configurar os movimentos do modelo de mobilidade, fazendo com que se comportem na rede de forma realística. Para isso, é necessário definir a ordem e o padrão de entrada, deste modo, é adotado o seguinte formato de leitura: tempo, índice do nó, posição x e posição y que corresponde a localização do nó na área de simulação.

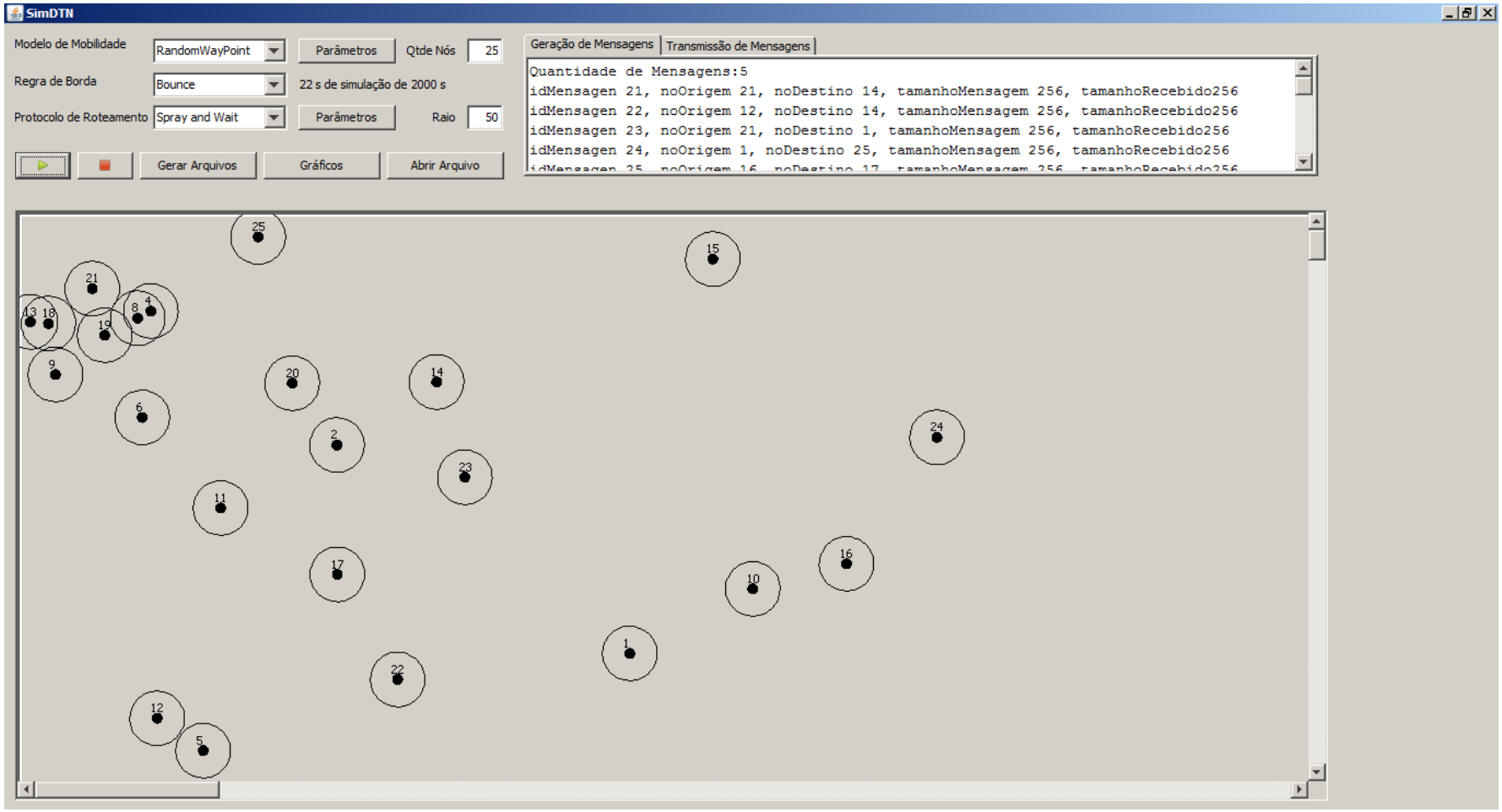
5.2 Avaliação do simulador *SimDTN*

Tendo em vista que para avaliar o *SimDTN* é necessário realizar o estudo do código fonte, o qual são analisados as implementações dos protocolos de roteamento, modelos de mobilidade e regras de borda. Além disso, será necessário realizar à análise dos parâmetros de entrada, utilização da interface, relatórios e os gráficos de saída. Em seguida, executar simulações do simulador utilizando as características mencionadas anteriormente e os parâmetros de entrada.

Para efetuar as simulações devem ser utilizados os protocolos de roteamento *Epidêmico, PROPHET, SPRAY AND WAIT* e o *Primeiro Contato*. Inicialmente todos os testes serão realizados utilizando os parâmetros da Tabela 3, em seguida é fixado para análise o modelo de mobilidade *Random WayPoint*, sendo assim, todas as simulações serão realizadas utilizando a regra de borda *Bounce*. Em sequência, serão gerados gráficos extraídos dos resultados dos relatórios após o tempo de 2000 segundos de simulações para análise, sendo que as métricas analisadas são taxa de entrega, *Tempo de Entrega* e o número de saltos realizados para se entregar uma mensagem, deste modo, as simulações utilizaram a medida do raio entre 50 à 400 metros.

A Figura 19, apresenta a simulação com raio 50, utilizando o protocolo *Spray and Wait* durante o tempo de 2000 segundo.

Figura 19 - Simulação utilizando o protocolo *Spray and Wait*



Fonte: Elaborada pelo autor.

A seguir na Tabela 3, são apresentados os parâmetros iniciais fixados para à avaliação do simulador, como resultado da revisão bibliográfica adota-se alguns valores como parâmetros. Sendo assim, servirão para calibração de novas simulações fazendo com que cada métrica utilizada possua maior precisão dos resultados.

Tabela 3 - Parâmetros utilizado nos resultados

Parâmetro	**Valor**
tempoDeslocamento	10.0
tempoTotalSimulacao	2000
velocidadeMinima	2.0
velocidadeMaxima	83.0
Raio	50 a 400
yMaxOrigen	9000.0
xMaxOrigen	8000.0
Quantidade de nós	25

Fonte: Elaborada pelo autor.

5.3 Avaliação comparativa das funcionalidades dos simuladores

Para realizar uma avaliação comparativa entre os simuladores, um estudo deles foi feito através da análise de sua documentação, disponível em artigos, sites, vídeos e fóruns de discussão. Em sequência, os parâmetros de comparação foram definidos baseando-se em métricas propostas nos trabalhos relacionados e nas características pertinentes entre os simuladores. Os parâmetros de comparação definidos e discutidos a seguir são: tipos de

licença, utilização da interface, emissão de relatórios, geração de gráficos, alteração de parâmetro e a documentação.

O tipo de licença do simulador é importante para sua escolha como ferramenta de análise na área educacional, reconhecimento de novos protocolos e avaliação dos protocolos de rede existentes. Pois, podem ter o código fonte modificado para que, por exemplo, uma nova funcionalidade seja adicionada ao simulador.

Por meio da análise da interface será possível verificar a usabilidade do simulador, definindo assim, a facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar uma tarefa especifica como a simulação de uma rede móvel. Através da interface também é possível realizar testes de funcionalidade que são necessários para se verificar a robustez e a qualidade de uso da implementação.

A emissão de relatórios e a geração de gráficos são importantes, pois permitem a verificação dos resultados emitidos após as simulações. Deste modo, ajuda na tomada de decisões, pois obtém dados e informações para medir a performance de troca de mensagens. Portanto, são de grande ajuda para a tomada decisões, como exemplo: tamanho do raio, escolha do protocolo de roteamento entre outras métricas de conectividade.

Uma das características que estimulam o emprego de simuladores no desenvolvimento de novos protocolos de roteamento está ligado com a facilidade da escolha e a rapidez na alteração dos parâmetros definidos durante as simulações. Mediante dessa característica é possível a um baixo custo criar e testar diversos protocolos de roteamento. Sendo assim, por meio dessa análise, percebe-se como é importante que a troca de parâmetros seja acessível.

Por fim, espera-se que um simulador de redes *DTN* possua uma boa documentação de forma a facilitar o uso da ferramenta por novos

pesquisadores e possibilite uma maximização de aprendizado. O parâmetro *documentação* avalia esse quesito dos simuladores.

6 RESULTADOS

Neste capítulo, são apresentados os resultados desse trabalho, deste modo, inicialmente foi proposto o desenvolvimento em um módulo do simulador, o qual realizou-se em duas etapas, sendo a primeira a implementação do protocolo de roteamento *Primeiro Contato* e a segunda a implementação do método de leitura de arquivos para a mobilidade dos nós. Em sequência, após a discussão dos resultados dessas duas etapas são apresentados os resultados da análise e das simulações realizadas com o SimDTN.

A seguir na Tabela 4, a implementação do protocolo de roteamento *Primeiro Contato* que funciona da seguinte maneira, enquanto o tempo de simulação é maior do que zero, são realizadas várias verificações a respeito do estado do nó. Deste modo, a princípio é efetuado a verificação da lista de mensagens do nó atual para comprovar se mesma está vazia. Caso a lista possua alguma mensagem é iniciado outro loop percorrendo a lista de modelos de mobilidade, verificando os nós que são diferentes do nó atual e que estejam dentro do raio de alcance. Se todos os requisitos anteriores forem preenchidos é realizado a adição da mensagem a lista mensagens entregues.

No entanto, se algum dos requisitos não forem realizados ocorre o reenvio da única cópia da mensagem para o próximo vizinho mais próximo de forma aleatória. Até que ocorra a entrega das mensagens ao destinatário.

Tabela 4 - Código fonte do protocolo *Primeiro Contato*

```
1.  long tempoInicio = System.currentTimeMillis();
2.      while (Tempo.calculaSegundosCorridos(tempoInicio) < tempoTotalSimulacao) {
3.          for (ModeloMobilidade modelo : nos) {
4.              if (!modelo.getPontoAtual().getMensagens().isEmpty()) {
5.                  for (Mensagem mensagem : modelo.getPontoAtual().getMensagens()) {
6.                      if (modelo.getPontoAtual().getIndex() != mensagem.getNoDestino()) {
7.                          for (ModeloMobilidade modelo1 : nos) {
8.                              if (modelo1.getPontoAtual().getIndex() != modelo.getPontoAtual().getIndex()) {
9.                                  if (Matematica.dist(modelo.getPontoAtual(), modelo1.getPontoAtual()) <= raio
10.                                     && !modelo1.getPontoAtual().getBufferMensagens().contains(mensagem)) {
11.                                 FPrincipal.console1().append("Tranferindo mensagem "
12.                                         + mensagem.getIdMensagem()
13.                                         + " do nó " + modelo.getPontoAtual().getIndex()
14.                                         + " para o nó " + modelo1.getPontoAtual().getIndex() + "\n");
15.                                 adicionaMensagem(mensagem, modelo1.getPontoAtual());
16.                                 if (mensagem.getNoDestino() == modelo1.getPontoAtual().getIndex()) {
17.                                     Mensagem msg = obtemMensagem(modelo1.getPontoAtual().getMensagens(), mensagem.getIdMensagem());
18.                                     if (msg.getTamanhoAtual() >= msg.getTamanhoMensagem()) {
19.                                         msg.setTempoFim(System.currentTimeMillis());
20.                                         adicionaMensagemRecebida(msg);
21.                                     }
22.                                 }
23.                             }
```

```
24.                 }
25.             }
26.         }
27.     }
28.   }
29.   }
30.   try {
31.     Thread.sleep(1000);
32.   } catch (InterruptedException ex) {
33.     Logger.getLogger(Epidemico.class.getName()).log(Level.SEVERE, null, ex);
34.   }
35. }
```

Fonte: Elaborado pelo autor.

São extraídos do arquivo de mobilidade informações a respeito dos parâmetros de tempo, quantidade de nós e o tamanho da área de simulação. Assim, a seguir na Figura 29, é apresentado um exemplo de traces de mobilidade real utilizados como padrão no *SimDTN,* contendo 4 colunas, no qual, a primeira coluna representa o tempo atual de movimentação, a segunda coluna o índice do nó, e por último as duas últimas colunas representam respectivamente X e Y na área de simulação.

Figura 20 - Traces de mobilidade real

```
5868    25    2222    4125
5870    1    4730    3773
5870    2    5488    4397
5870    3    5710    4633
5870    4    1670    4783
5870    5    3652    4499
5870    6    2853    4001
5870    7    1448    4871
5870    8    3916    4002
5870    9    4047    3714
5870    10    2725    4368
5870    11    6418    3165
5870    12    2962    4079
5870    13    1071    4231
5870    14    6014    3989
5870    15    4141    4072
5870    16    1386    3878
5870    17    2731    5934
5870    18    2810    4181
5870    19    1209    3829
5870    20    2491    4130
5870    21    5489    4395
5870    22    6064    4858
5870    23    2305    4408
5870    24    3655    4222
5870    25    2222    4125
5872    1    4740    3773
5872    2    5476    4388
5872    3    5690    4633
5872    4    1660    4783
5872    5    3647    4491
5872    6    2853    4001
5872    7    1445    4879
5872    8    3914    4003
```

Fonte: Elaborada pelo autor.

A seguir a Tabela 5, apresenta o código fonte do módulo de leitura de mobilidade, deste modo, sua implementação necessita de dois métodos o *btnArquivoActionPerformed(java.awt.event.ActionEvent evt)* e o *run*().

Inicialmente, com o objetivo de que os métodos anteriores funcionem da melhor maneira é necessário acionar e selecionar o arquivo através do botão Abrir Arquivo presente na *interface* do *SimDTN*. Desta maneira, após selecionar o arquivo de mobilidade o *btnArquivoActionPerformed* inicializa os objetos MaxTempo, MaxNos, MaxX e MaxY com valores nulos e executa o método *run*().

O método *run*() é necessário, pois sincroniza a leitura do arquivo externo com as outras execuções simultâneas do *SimDTN*. Basicamente o módulo de leitura, abre o arquivo selecionado, e após a leitura de cada linha é criado um objeto ponto, no qual, serão inseridas as informações pertinentes, tais como: tempo, índice e posição. E em sequência, o ponto criado é inserido na lista de modelos de mobilidade *point* com a chave concatenada por tempo e índice do objeto *pontoArquivo*.

Durante a leitura do arquivo, são realizadas algumas comparações afim de verificar e preencher os valores máximos de cada objeto inicializado no método *btnArquivoActionPerformed*. Com isso, todos os objetos serão utilizados durante a execução do simulador para inicializar e/ou substituir os valores *default*.

Tabela 5 - Código fonte do módulo de leitura.

```
1.  private void btnArquivoActionPerformed(java.awt.event.ActionEvent evt) {
2.          MaxNos = 0;
3.          MaxTempo = 0.;
4.          MaxX = 0.;
5.          MaxY = 0.;
6.          this.run();
7.      }
8.  public void run() {
9.          int returnVal = FileChooser.showOpenDialog(this);
10.         int linha, coluna;
11.         int TamReal = 1;
12.         Ponto pontoArquivo = null;
13.         point = new HashMap<String, Ponto>();
14.         point = Collections.synchronizedMap(point);
15.         if (returnVal == FileChooser.APPROVE_OPTION) {
16.             File file = FileChooser.getSelectedFile();
17.             BufferedReader br = null;
18.             try {
19.                 //Aqui vai o objeto File criado lá em cima.
20.                 br = new BufferedReader(new FileReader(file));
21.                 String Valores[] = null;
22.                 while (br.ready()) {
23.                     String linhaArquivo = br.readLine();
24.                     Valores = linhaArquivo.split("            ");
25.                     pontoArquivo = new Ponto(Double.parseDouble(Valores[0]), Integer.parseInt(Valores[1]), Double.parseDouble(Valores[2]), Double.parseDouble(Valores[3]));
26.                     String chave = pontoArquivo.getTempo() + "-" + pontoArquivo.getIndex();
27.                     point.put(chave, pontoArquivo);
28.                     if (pontoArquivo.getIndex() > MaxNos) {
29.                         MaxNos = Integer.parseInt(Valores[1]);
```

```
30.                 }
31.                 if (pontoArquivo.getX() > MaxX) {
32.                     MaxX = Double.parseDouble(Valores[2]);
33.                 }
34.                 if (pontoArquivo.getY() > MaxY) {
35.                     MaxY = Double.parseDouble(Valores[3]);
36.                 }
37.             }//end while
38.             MaxTempo = Double.parseDouble(Valores[0]);
39.             br.close();
40.             try {
41.                 Thread.sleep(10);
42.             } catch (InterruptedException ex) {
43.                 Logger.getLogger(FPrincipal.class.getName()).log(Level.SEVERE, null, ex);
44.             }
45.         } catch (IOException ex) {
46.             Logger.getLogger(FPrincipal.class
47.                     .getName()).log(Level.SEVERE, null, ex);
48.         }
49.     } else {
50.         System.out.println("File access cancelled by user.");
51.     }
52. }
```

Fonte: Elaborado pelo autor.

A seguir na Tabela 6, são apresentados os parâmetros iniciais fixados no simulador para a realização dos testes de avaliação.

Tabela 6 - Parâmetros utilizado nos resultados

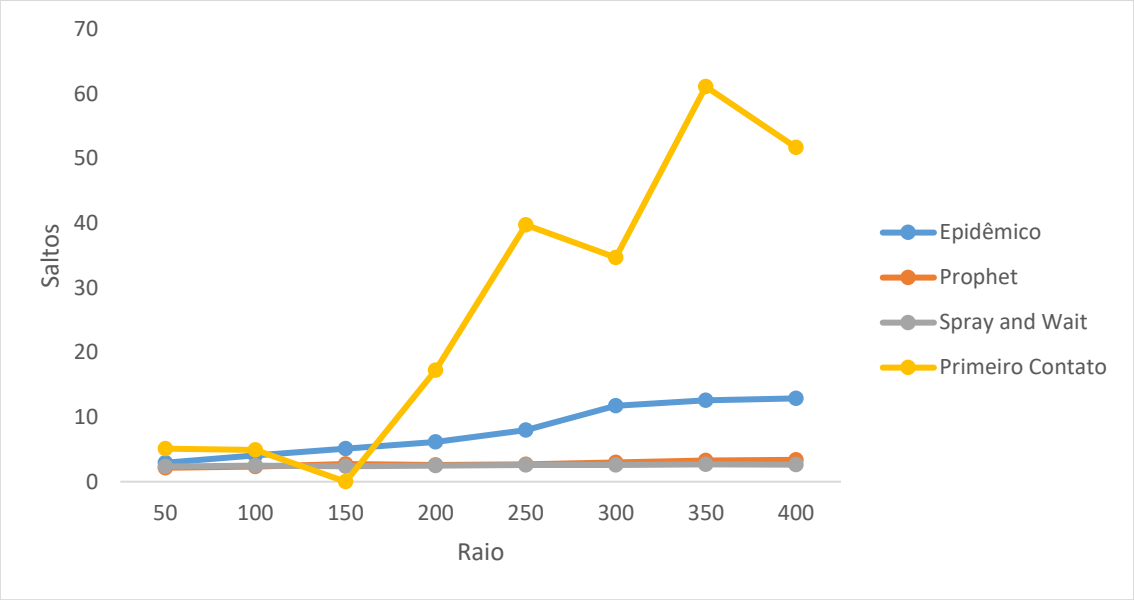
Parâmetro	**Valor**
tempoDeslocamento	10.0
tempoTotalSimulacao	2000
velocidadeMinima	2.0
velocidadeMaxima	83.0
Raio	50 a 400
yMaxOrigen	9000.0
xMaxOrigen	8000.0
Quantidade de nós	25

Fonte: Elaborada pelo autor.

Abaixo os resultados das simulações realizadas para a avaliação do *SimDTN*, os protocolos de roteamento utilizados são: *Epidêmico, PROPHET, SPRAY AND WAIT* e o *Primeiro Contato*.

O Gráfico 1, exibe o número de saltos realizados por cada protocolo de roteamento para realizar a entrega das mensagens ao destinatário. Nota se, que o protocolo *Epidêmico* realizou um número de saltos gradual em relação ao aumento do raio de alcance, enquanto o *Primeiro Contato* oscilou de maneira bem acentuada e realizou muitos saltos. Sendo os *Spray and Wait* e *PROPHET* mantiveram o número de saltos retilíneo de saltos durante o tempo e mesmo com o aumento do raio de alcance.

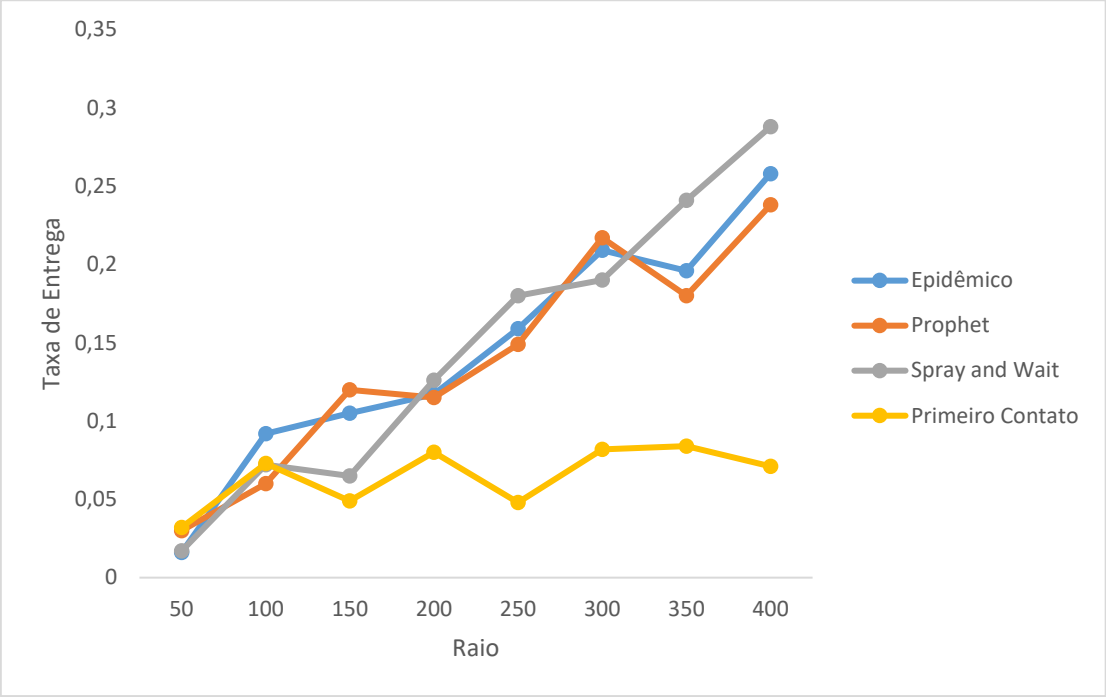
Gráfico 1 - Gráfico de número de Saltos



Fonte: Elaborada pelo autor.

O Gráfico 2, aponta a relação do raio de alcance com taxa de entrega de mensagens realizadas por meio dos quatro protocolos de roteamento. Sendo assim, já era esperado que o protocolo *FC* obtivesse as piores taxas de entrega, por se tratar de um protocolo muito simples. Nota se também, que o protocolo *SPRAY AND WAIT* não obteve bons resultados com o raio de alcance 150, no entanto, conseguiu resultados significativos com o raio de maior do que 150 metros. O *PROPHET* e o *Epidêmico* obtiveram desempenho de entrega bem similares em todos os raios apresentados.

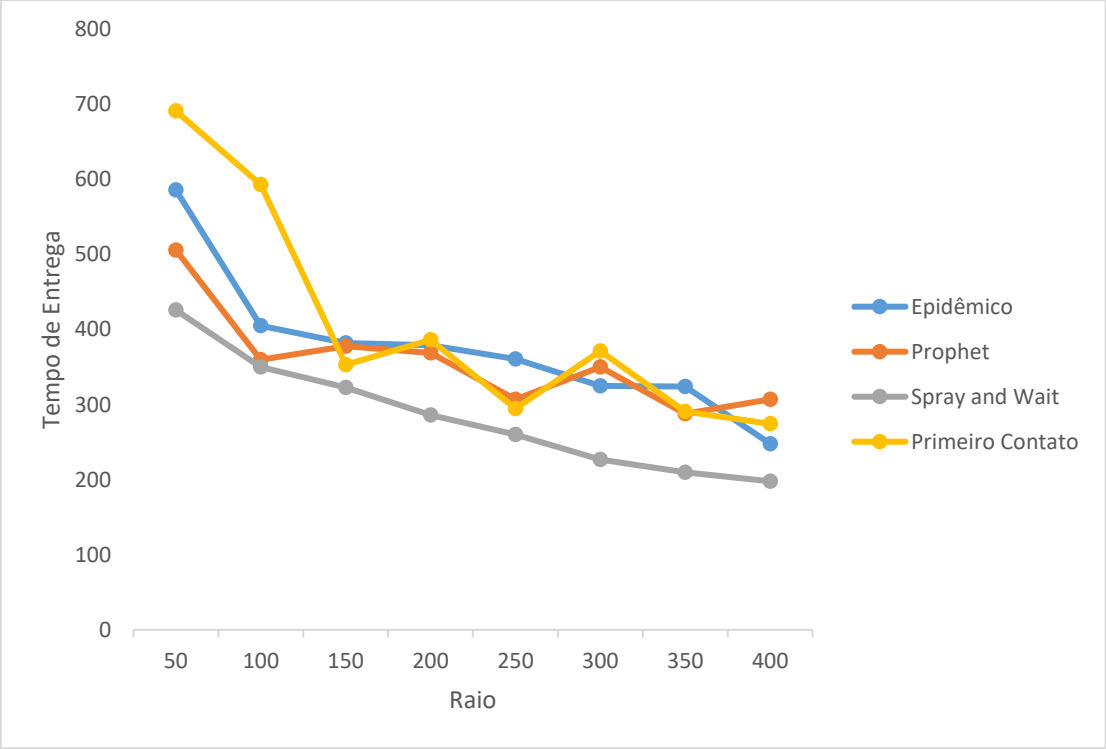
Gráfico 2 - Taxa de Entrega



Fonte: Elaborada pelo autor.

A seguir no Gráfico 3 é apresentado o *Tempo de Entrega*, em resumo é o tempo gasto após a geração da mensagem para ela ser entregue ao seu destinatário. No gráfico, ainda é possível visualizar e relacionar a alta taxa de entrega com o aumento do raio de alcance dos protocolos. Também é possível salientar o bom desempenho relacionado ao *Spray and Wait*, durante as simulações realizadas no simulador *SimDTN*.

Gráfico 3 - *Tempo de Entrega*



Fonte: Elaborada pelo autor.

A seguir na Tabela 7, é apresentado o resultado da análise entre os simuladores *SimDTN, NS, OMNeT++, SUMO* e o *ONE.* O resultado da análise é exposto da forma de prós e contras de cada simulador.

Tabela 7 - Resultado da análise dos simuladores

	Prós	Contras
SimDTN	Interface amigável e simples;	Pouca documentação;
	Implementa os principais protocolos, modelos de mobilidade e regras de borda da literatura;	Em fase de testes.
	Aceita adição de módulos;	
	Gera relatórios e gráficos com os resultados das simulações.	
ONE	Construção do cenário de simulação por parte do usuário (mapas, GPS, etc.);	Falta de informações em sua interface;
	Aceita adição de módulos.	Não apresenta os resultados gráficos.
SUMO	Ampla documentação;	Requisito conhecer a linguagem XML;
	Aceita adição de módulos;	Não apresenta os resultados em gráficos.
	Permite gerar várias saídas para cada simulação;	
	Vasta definições de característica para simulações veiculares.	
NS	Oferece suporte à simulação de um grande número de tecnologias de rede;	Obtenção de modelo de mobilidade definida por Script Tool Command Language (TCL);
	Código aberto.	Necessário a cada alteração de parâmetro a recompilação.
OMNeT++	Arquitetura modular;	Criação de um arquivo de configuração chamado de “omnetpp.ini” ;

	Gera dados estatísticos durante a execução;	Os nós ativos do modelo são programados em C++.
	Fornece ferramentas para análise do resultados coletados.	

Fonte: Elaborada pelo autor.

7 CONCLUSÃO

Nesta seção, são apresentadas as principais contribuições e conclusões alcançadas durante a realização deste trabalho. Sendo assim, uma das maiores contribuições será para a área educacional, a qual se espera que através do desenvolvimento e avaliação das novas funcionalidades em um módulo do *SimDTN* consiga-se auxiliar pesquisadores e profissionais da área. Também almeja-se que o *SimDTN* sirva como base para projetos de pesquisas futuro.

Logo o protocolo *Primeiro Contato* apresentou baixas taxas e tempo elevado de entrega mesmo em simulações com recursos abundantes. Apesar disso, pode-se levar em consideração algumas situações onde este protocolo é capaz de ser uma boa escolha em algum determinado momento. Como exemplo em uma rede de celulares, na qual a maior necessidade é a economia de energia, deste modo, pelo fato do protocolo enviar uma única cópia pela rede não ocasionando assim a inundação e evitando o congestionamento e o excesso de informações na rede. Portanto, o *Primeiro Contato* possui alguma vantagem em relação à economia de energia diante dos outros protocolos.

Em seguida, com base nos estudos que deram suporte a este trabalho por meio de análise e comparativo de funcionalidades entre os outros simuladores da literatura, é possível afirmar que, o simulador *SimDTN* pode ser utilizado como ferramenta por pesquisadores, levando em consideração os resultados dos Gráficos 1, 2 e 3 é possível perceber que os resultados simulador são compatíveis com as métricas de saídas de outros simuladores.

Deste modo, é possível concluir que o simulador *SimDTN* atende os objetivos de um simulador de redes DTN podendo ser utilizado também na disciplina de redes como ferramenta de ensino. A comparação entre os temas os simuladores abordados, demonstram a viabilidade na utilização do simulador, sendo assim, contribuindo para uma melhor compreensão e aprendizado dos temas relacionados a redes móveis.

8 REFERÊNCIAS

ALMEIDA, V. D. D. **Análise de desempenho de protocolos de roteamento *Ad Hoc* e DTN em redes de emergência**. 2011. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2011.

ANTAS, R. B. Z. Redes de Computadores II. Modelos de Mobilidade, Rio de janeiro, 29 out 2006. Disponível em: <http://www.gta.ufrj.br/grad/06_2/zuquim/index.html>. Acesso em: 17 de jun. 2015.

ASKOURA, Osama. MAC Protocol for Multi-Hop Multicast in *Ad Hoc* Networks. ***Slideshare***, 20 dez. 2015. Disponível em: <http://pt.slideshare.net/hectorfelipecabral/redes-de-computadores-redes-wireless-32580528>. Acesso em: 18. nov. 2016.

BASTOS, C. Redes Sem Fios: tecnologia Wi Fi. ***Slideshare***, 9 jul. 2009. Disponível em: <http://pt.slideshare.net/scb.carlos/redes-sem-fios>. Acesso em: 17 nov. 2016.

BATISTA, T. S. **Avaliação do Consumo de Energia em Protocolos de Roteamento para Redes Tolerantes a Atrasos e Interrupções**. 2011. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Campina Grande, Campina Grande, Paraíba, 2011.

BEHRISCH, M. et al. *SUMO – Simulation of Urban MObility: An Overview.* In: Proceedings of SIMUL 2011. *2011,* Espanha, Barcelona. **Publications of the DLR**. 2011. Disponível em: <http://elib.dlr.de/71460/>. Acesso em: 19 nov. 2016.

BETTSTETTER, C. ***Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects***, New York, NY, USA, v. 5, n. 3, p. 55 – 66, Jul. 2001.

BEZERRA, R. L. **Análise da Conectividade em Redes Móveis Utilizando Dados Obtidos da Mobilidade Humana.** 2009. Dissertação (Mestrado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

BOUDEC, J. Y. L; VOJNOVIC, M. The random trip model: stability, stationary regime, and perfect Simulation. **IEEE/ACM Transactions on Networking (TON)**, v. 14, Ed. 6, p. 1153-1166, 01 jan, 2006.

CABRAL, H. F. Redes de Computadores - Redes Wireless. ***Slideshare***, 21 mar. 2014. Disponível em: <http://pt.slideshare.net/hectorfelipecabral/redes-de-computadores-redes-wireless-32580528>. Acesso em: 18. nov. 2016.

CAMPOS, C. A. V.; MORAES, L. F. M. **Modelos Markovianos de Mobilidade Individual para Redes Móveis *Ad Hoc***. 2007. Disponível em: <http://www.garf.coppe.ufrj.br/PDFs/ST2003_Modelos_Markoviano_Mobilidade_AdHoc.pdf>. Acesso em: 17 de junho 2015. Publicado durante o Simpósio Brasileiro de Redes de Computadores – SBR2003, Natal, RN, Brasil.

CAMPOS, C. A. V.; MORAES, L. F. M.; SILVA, R. F. Caracterização da Mobilidade Veicular e o seu Impacto nas Redes Veiculares Tolerantes atrasos e Desconexões, **XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Gramado, RS, 28, p. 585-598, 2010.

CAVALCANTI, E.R. **Avaliação de Modelos de Mobilidade em Redes *Ad Hoc* Sem Fio**, 2009. 131 f. Dissertação (Mestrado em ciência da computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, Campina Grande, PB, 2009.

COZER, F. L. **Segurança Redes Sem fio**. 2006. Monografia (Bacharelado em Ciência da Computação) - Faculdade de Jaguariúna, Jaguariúna, SP, 2006.

CRUZ, I. F.; ASSAKURA, R. **Avaliação de Desempenho de Protocolos de Roteamento de Redes DTN Oportunistas**. 2013. 71 f. Monografia (Trabalho de Conclusão de Curso Bacharelado em Sistemas de Informação) - Universidade Federal Do Estado Do Rio de Janeiro - Centro de Ciências Exatas e Tecnologia, Rio de Janeiro, RJ, 2013.

DELAMARE, F.L. **Análise de Padrões de Mobilidade Utilizando Redes de Autômatos Estocásticos**, 2008. 81 f. Dissertação (Mestrado em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, 2008.

FALL, K.; VARADHAN, K. **The NS Manual**, 2009. Disponível em <http://www.isi.edu/nsnam/ns/ns-documentation.html>. Acesso em: 14 de Nov. 2016.

FERNANDES, R. M. S.; CAMPOS, C. A. V.; MORAES, L. F. M. Uma Avaliação das Redes Tolerantes a Atrasos e Desconexões Através de Traces Reais de Mobilidade Humana, **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Recife, PE, 27, p. 785-797, 2009.

FERREIRA, J, L, M. **Segurança em Redes Sem Fio**. 2013. 49 f. Monografia (Trabalho de Conclusão do Curso de Especialização em ConFiguração e Gerenciamento de Servidores e Equipamentos de Rede) - Universidade Tecnológica Federal do Paraná - Departamento Acadêmico de Eletrônica, Curitiba, PR, 2013.

GALLO, M. A.; HANCOCK, W. M. **Comunicação entre Computadores e Tecnologia de Rede**. São Paulo: Pioneira Thomsom Learning, 2003. 5 p. ISBN: 85 -221 -0293 - 7.

HARGREAVES, E.M. **Um Novo Modelo de Mobilidade Para Redes Sem Fio**: Distribuições Exatas Para Velocidade e Direção com Aplicações em Simulações Perfeitas, 2006. 94 p. Dissertação (Mestrado em ciências em engenharia de sistemas e computação) - Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, RJ, 2006.

HERNANDEZ, D. A. B. **Uma Avaliação da Influência da Velocidade dos Nós do Estabelecimento de Caminhos em Redes *Ad Hoc* Veiculares**. 2015. 66 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro - Centro de Ciências Exatas e Tecnologia, Rio de Janeiro, RJ, 2015.

KAMIENSKI, C. A. Avaliação de Desempenho: simulando a internet. ***Livro Zilla***. 6 jun. 2014. Disponível em <http://livrozilla.com/doc/51443/simulando-a-internet>. Acesso em: 18. nov. 2016.

KERÄNEN, A.; OTT, J. ***The ONE Simulator for* DTN *Protocol Evaluation***. In: *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, Roma, Italia. *Akeranen Github The ONE*, 2009. Disponível em: <https://akeranen.github.io/the-one/>. Acesso em: 19 nov. 2016.

KRAJZEWICZ, D. et al. Recent Development and Applications of *SUMO* – Simulation of Urban MObility, **International Journal on Advances in Systems and Measurements**, Berlim, Alemanha, 2012. Disponível em:<http://www.iariajournals.org/systems_and_measurements/>. Acesso em: 20 nov. 2016.

LI, E. **Routing and Topology Control in Multihop Wireless *Ad Hoc* Networks**. 2001. 135 f. Dissertação (Doutorado) - Cornell University, Ithaca, NY, USA, 2001.

LIMA, P. A. **Gerenciamento de *Buffers* em Redes DTN baseado em Entropia de Mobilidade**. 2013. 52 f. Dissertação (Mestrado em Informática – Redes de Computadores) - Universidade Federal do Amazonas - Centro de Ciências Exatas e Tecnologia, Manaus, AM, 2013.

LINDGREN, A., DORIA, A., and SCHELÉN, O. Probabilistic Routing in Intermittently Connected Networks. ***ACM SIGMOBILE Mobile Computing and Communications Review***, 2003, 7.3: 19-20.

LIU, J. et al. Empirical Validation of Wireless Models in Simulations of *Ad Hoc* Routing Protocols. **Society for Computer Simulation International**, San Diego, CA, USA, 307-323, Abr. 2005

LOPEZ, R., F. **Introdução à Computação Móvel**. São Luís, MA: Universidade Federal do Maranhão, 16 abr. 2014 (Aula).

MARTINS, C. P. Aula 08 - árvores. ***Slideshare***, 20 jun. 2013. Disponível em: <http://pt.slideshare.net/cpmart/aula-08-rvores>. Acesso em: 18. nov. 2016.

MENDES, A.; NUNES, B.A.A.; MORAES, L. F. M. Uma avaliação dos efeitos das regras de borda e dos modelos de mobilidade no comportamento dos nós em redes *Ad Hoc*. In: **VI Workshop de Comunicação Sem Fio e Computação Móvel-WCSF 2004**, Fortaleza, CE, 2004. p. 117- 126.

MOTA, V. F. S.; SILVA, T. H.; NOGUEIRA, J. M. S. Introduzindo Tolerância a Interrupção em Redes *Ad Hoc* Móveis para Cenários de Emergência, **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Recife, PE, 27, p. 671-684, 2009.

OLIVEIRA, C. T. **Uma Proposta de Roteamento Probabilístico para Redes *Tolerant*es a Atrasos e Desconexões**. 2008. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2008.

OLIVEIRA, C. et al. (2007). **Redes tolerantes a atrasos e desconexões**. In: Simpósio Brasileiro de Computadores BRC 2007, Minicursos, p. 203–256, 2007.

OLIVEIRA, R. **Introdução Ao Simulador De Redes - The Network Simulator Ns-2**. Almada, Portugal: Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2010.

PERCÍLIA, E. Redes sem Escalas. ***Brasil Escola***. Disponível em <http://brasilescola.uol.com.br/informatica/redes-sem-escalas.htm>. Acesso em: 18. nov. 2016.

PINHEIRO, J. M. S. Redes Móveis *Ad Hoc*, **Projeto de Redes**, Volta Redonda, RJ, 5 abr. 2005. Disponível em <http://www.projetoderedes.com.br/artigos/artigo_redes_moveis_ad_hoc.php>. Acesso em: 15 de Nov. 2016.

PONTES, A. S. L. **Análise de Estabilidade do Modelo de Mobilidade *Random Waypoint* em Redes *Ad Hoc* Utilizando o Simulador *Jist/Swans***. 2009. 56 f. Monografia (Trabalho de Conclusão de Curso de Engenharia da

Computação) - Escola Politécnica de Pernambuco - Departamento de Sistema de Computação, 2009.

PRADHAN, N. Creating Real Time Vehicular Traffic Simulation Using *SUMO* and *OpenStreetMap*, **Techerina:** be the first to know, India, 2015. Disponível em:<http://www.techerina.com/2015/03/creating-real-time-vehicular-traffic-simulation-using-*SUMO*.html>. Acesso em: 20 nov. 2016.

REZENDE, N. S. **Redes Móveis Sem Fio *Ad Hoc***. 2004. Monografia (Especialização em Gerência de Redes) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

RUBINSTEIN, M. G. A formação universitária do profissional de educação física. In: PASSOS, S. de C. E. (Org.). **Educação Física e Esporte na Universidade**. Brasília: Secretaria de Educação Física e Desportos, 1988. p. 47-58.

SOARES, S. A. **Simulação de Algoritmos de Roteamento Unicast Utilizando o Simulador Ns-2**, 2009. 78 f. Monografia (Trabalho de Conclusão de Curso) - FACULDADE FARIAS BRITO, Fortaleza, CE, 2009.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. ***SPRAY AND WAIT****: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*. Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, New York, NY, USA, p. 252–259, 26 ago. 2005.

TABARELLI, A.; SILVA, C. C. **Redes Tolerantes a Atrasos, Protocolos de Disseminação e Aplicações**, 2009. 40 f. Monografia (Trabalho de formatura supervisionado) – Universidade de São Paulo, São Paulo, 2009.

TESSINARI, R. S. **Mapeamento de Equipamentos Ópticos e Modelagem de Redes OTN no Simulador *OMNeT*++ de Acordo com a Recomendação ITU-T G.798**. 2009. 75 f. Monografia (Trabalho de Conclusão de Curso Engenharia de Computação) - Universidade Federal do Espírito Santo - Centro Tecnológico - Departamento de Informática, Vitória, ES, 2009.

VAHDAT, A.; BECKER, D. Epidemic Routing For Partially Connected *Ad Hoc* Networks. **Technical report, Duke University**, 2000.

VARGA, A. *The OMNeT++ Discrete Event Simulation System.* ***Omnetpp org****,* Praga, República Tcheca, jun. 2001. Disponível em: <https://omnetpp.org/intro>. Acesso em: 03 de Nov. 2016.

VOSS, G. B. et al. **Proposta de utilização de Laboratórios Virtuais para o ensino de Redes De Computadores**: articulando ferramentas, conteúdos e possibilidades. (Fase I)". RENOTE - Revista Novas Tecnologias na Educação, UFRGS, V. 10 Nº 2. (2012) 2012.