



ESCOLA POLITÉCNICA



# ANÁLISE DE ESTABILIDADE DO MODELO DE MOBILIDADE *RANDOM WAYPOINT* EM REDES *AD HOC* UTILIZANDO O SIMULADOR JIST/SWANS

Trabalho de Conclusão de Curso

Engenharia da Computação

Alisson Soares Limeira Pontes  
Orientador: Prof. Renato Mariz de Moraes



UNIVERSIDADE  
DE PERNAMBUCO

**ALISSON SOARES LIMEIRA PONTES**

**ANÁLISE DE ESTABILIDADE DO  
MODELO DE MOBILIDADE *RANDOM  
WAYPOINT* EM REDES *AD HOC*  
UTILIZANDO O SIMULADOR  
JIST/SWANS**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, junho de 2009.

*Aos meus pais.*

# Agradecimentos

Agradeço, antes de tudo, aos meus pais; 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.

À minha companheira, Layane Milena, que sempre me apoiou, me deu forças e esteve sempre ao meu lado, incondicionalmente.

Ao professor Renato Moraes, que esteve sempre disposto a ajudar e com quem aprendi muito.

À todos os meus professores e amigos, pelo apoio e pelo companheirismo durante esses cinco anos de lutas e conquistas.

# Resumo

Redes *ad hoc* são redes formadas por dispositivos computacionais sem fio. Essas redes não possuem infra-estrutura física nenhuma e todo processo de comunicação é desempenhado unicamente pelos próprios nós da rede. Este trabalho trata da importância na escolha do modelo de mobilidade utilizado em simulações para avaliação de desempenho de protocolos de comunicação em redes *ad hoc* e da importância de se ter ambientes de simulação confiáveis. O modelo de mobilidade mais usado na literatura é o *Random Waypoint* (RWP), no entanto trabalhos anteriores mostraram que em algumas condições o RWP falha ao não atingir um estado estacionário e que por isso não deve ser usados em simulações. A proposta deste trabalho é que uma alteração na escolha da função densidade de probabilidade da velocidade dos nós permita ao RWP sempre atingir o estado estacionário, estabilizando este modelo para uso nos simuladores de rede *ad hoc*. Por meio de simulações utilizando o simulador de redes JiST/SWANS, buscou-se avaliar medidas que têm impacto na vazão e qualidade de serviço das aplicações que utilizam redes *ad hoc*. Portanto, um modelo mais realista será proposto, o qual será validado com simuladores de redes e através de estudo analítico.

# Abstract

Ad hoc networks are networks composed by wireless computing devices. These networks do not have any physical infrastructure and all the communication process is performed only by the network nodes. This work approaches the importance of choice of the mobility model used in simulations to evaluate the performance of communication protocols in ad hoc networks, and the importance of having reliable simulation environments. The most used mobility model in literature is the Random Waypoint (RWP), however previous works showed that in some conditions the RWP fails to reach a steady-state and therefore should not be used in simulations. The purpose of this study is that a change in the probability density function of nodes speed allows RWP always to reach the steady-state, stabilizing this model for use in simulations of ad hoc network. The network simulator JiST/SWANS is used to perform simulations to evaluate metrics that have impact on throughput and quality of service for applications that use ad hoc networks. Therefore, a more realistic model is proposed, which will be validated with simulations and through analytical study.

# Sumário

<b>Índice de Figuras</b>	<b>iv</b>
<b>Tabela de Símbolos e Siglas</b>	<b>v</b>
<b>Capítulo 1 Introdução</b>	<b>6</b>
1.1 Motivação	8
1.2 Objetivos	9
1.3 Estrutura da Monografia	10
<b>Capítulo 2 Modelos de Mobilidade: RWP e problemas relacionados</b>	<b>12</b>
2.1 A importância na escolha do modelo de mobilidade	12
2.2 O RWP	15
2.2.1 Problemas relacionados ao RWP	16
<b>Capítulo 3 O Modelo de Mobilidade <i>Random Waypoint</i> Estabilizado</b>	<b>20</b>
3.1 Distribuição de Probabilidade BETA( $\alpha, \beta$ )	22
3.1.1 Distribuição de Probabilidade BETA(1,1)	23
3.1.2 Distribuição de Probabilidade BETA(2,2)	23
3.1.3 Distribuição de Probabilidade BETA(2,1)	25
<b>Capítulo 4 Visão Geral do Simulador JiST/SWANS</b>	<b>29</b>
4.1 JiST	30
4.2 SWANS	32
4.3 Instalação e Configuração	33
<b>Capítulo 5 Simulação e Análise dos Resultados</b>	<b>36</b>
<b>Capítulo 6 Conclusão e Trabalhos Futuros</b>	<b>41</b>
6.1 Contribuições	41
6.2 Trabalhos Futuros	42
<b>Bibliografia</b>	<b>44</b>
<b>Apêndice A Problemas Encontrados no JiST/SWANS Durante Este Trabalho</b>	<b>47</b>
<b>Anexo A Problemas Encontrados no JiST/SWANS Pela Comunidade</b>	<b>49</b>

# Índice de Figuras

<b>Figura 1.</b> Ilustração de uma rede <i>ad hoc</i> com saída para uma rede fixa. ....	8
<b>Figura 2.</b> Número de pacotes de controle de roteamento AODV em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade <i>teleport</i> e <i>Random walk</i> . ....	15
<b>Figura 3.</b> Número de pacotes de dados descartados em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade <i>teleport</i> e <i>Random walk</i> . ....	15
<b>Figura 4.</b> Atraso de entrega de pacotes de dados em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade <i>teleport</i> e <i>Random walk</i> . ....	16
<b>Figura 5.</b> Percentagem média de vizinhos vs. tempo. ....	18
<b>Figura 6.</b> Velocidade média instantânea em função do tempo no RWP original para intervalo da velocidade dado por $(0,20]$ e tempo de pausa nulo. ....	19
<b>Figura 7.</b> Gráfico da função densidade de probabilidade, descrita na Eq. (2), para escolha das velocidades dos nós no modelo RWP modificado. ....	24
<b>Figura 8.</b> Gráfico da função densidade de probabilidade, descrita na Eq. (11), para escolha das velocidades dos nós no modelo RWP modificado. ....	25
<b>Figura 9.</b> Gráfico da função densidade de probabilidade, descrita na Eq. (16), para escolha das velocidades dos nós no modelo RWP modificado. ....	26
<b>Figura 10.</b> Velocidade instantânea média para simulações utilizando o RWP original, modificado (BETA(2,2)) e estabilizado (BETA(2,1)). ....	28
<b>Figura 11.</b> Arquitetura do JiST. ....	31
<b>Figura 12.</b> Arquitetura do SWANS. ....	33
<b>Figura 13.</b> Carregando o re-editor de bytcodes. ....	34
<b>Figura 14.</b> Inserindo os parâmetros da simulação. ....	35
<b>Figura 15.</b> Número de pacotes de controle de roteamento AODV em função do tempo de simulação. ....	37
<b>Figura 16.</b> Número de pacotes de dados descartados em função do tempo de simulação. ....	38
<b>Figura 17.</b> Atraso de entrega de pacote de dados em função do tempo de simulação. ....	38
<b>Figura 18.</b> Número de pacotes de controle AODV em função do tempo de simulação obtidos do simulador NS-2 para o RWP original. ....	39
<b>Figura 19.</b> Número de pacotes de dados descartados em função do tempo de simulação obtidos do simulador NS-2 para o RWP original. ....	40

# Tabela de Símbolos e Siglas

PDA – Personal Digital Assistant (Assistente Pessoal Digital)

MANET – Mobile Ad Hoc Network (Rede Ad Hoc Móvel)

PAN – Personal Areal Network (Rede de área pessoal)

JiST – Java in Simulation Time (Java em Tempo de Simulação)

SWANS – Scalable Wireless Ad hoc Network Simulator (Simulador Escalável para Redes Ad Hoc Sem Fio)

RWP – Random Waypoint

AODV – Ad Hoc On-demand Distance Vector

DSR - Dynamic Source Routing

NS-2 – Network Simulator versão 2 (Simulador de Redes versão 2)

GloMoSim – Global Mobile Information Systems Simulation Library (Biblioteca )

MAC – Media Access Control (Controle de Acesso ao Meio)

SBRC – Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

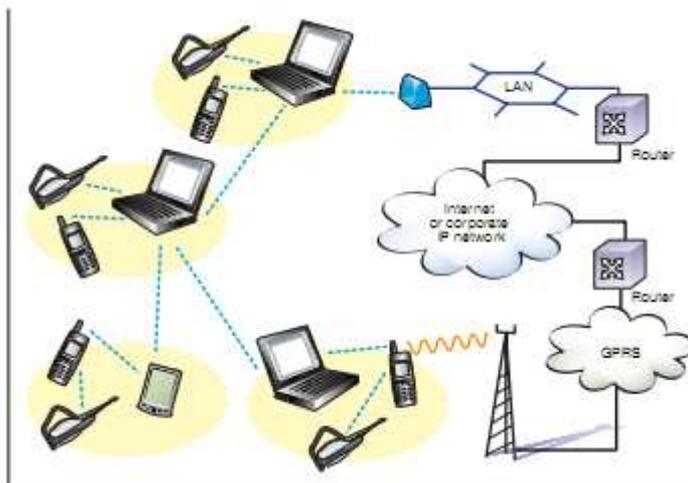
# Capítulo 1

## Introdução

A disponibilidade dos dispositivos portáteis e avanços na computação sem fio têm contribuído para a crescente aceitação de aplicações de computação móvel e abriram as portas para a possibilidade de serviços ubíquos em ambientes móveis. Esses dispositivos (ou nós) móveis, incluindo telefones, PDA's, *laptops* e sensores cooperam entre si para formar e criar uma rede *ad hoc* móvel (MANET) através da comunicação sem fios com outros dispositivos, sem o apoio de uma infra-estrutura fixa ou sistema de controle centralizado.

MANETs são formadas por dispositivos computacionais sem fio que podem se mover livremente e podem atuar tanto como agentes diretos de uma comunicação de rede quanto como roteadores. No papel de agentes, os nós podem ser classificados como geradores ou receptores de tráfego de dados. Já no papel de roteadores, os nós encaminham pacotes entre agentes que não podem se comunicar diretamente devido às limitações de alcance de sinal entre ambos [12]. Para troca de informações uma rota determinada dinamicamente é necessária. Ela é criada e mantida em função de diversos parâmetros, tais como a distância entre os nós e a topologia da rede, com nós entrando, saindo e movendo-se pela rede a qualquer momento.

Essas redes são formadas dinamicamente e são gerenciadas pelos próprios nós que compõem a rede. Portanto, uma rede *ad hoc* pode ter qualquer topologia, e ter ou não conexões para uma rede fixa (Figura 1). Os usuários devem poder trazer seus dispositivos e se conectarem à rede sem necessidade de configuração anterior. Quando o dispositivo é ligado, ele é conectado espontaneamente e pode se comunicar com os demais nós da rede ou usar os serviços que estão disponíveis nessa rede. Por essa espontaneidade e por sua natureza dinâmica, projetar esse tipo de rede pode ser bastante desafiador.



**Figura 1.** Ilustração de uma rede *ad hoc* com saída para uma rede fixa. [12]

A falta de um controle central em redes *ad hoc* impõe enormes desafios para os projetistas de tais sistemas e formam, em verdade, um campo de pesquisa aberto e repleto de perguntas sem respostas. Mesmo assim, existem possibilidades cada vez mais realistas de viabilização em larga escala desse tipo de rede em aplicações comerciais como na combinação entre redes de área pessoal, mais conhecida pela sigla PAN (*Personal Area Network*), e redes metropolitanas, como as redes celulares [12].

Dentre os grandes desafios para realizar pesquisas principalmente em redes sem fio está a construção de sistemas de simulação eficientes e confiáveis. Um importante aspecto que deve ser bem investigado nos sistemas de simulação é a mobilidade dos nós sob diferentes cenários de movimento. O padrão de mobilidade representa a sucessiva movimentação das estações comunicantes numa rede sem fio móvel. Diversos estudos [6], [11] e [14] têm demonstrado que o modelo de mobilidade escolhido em termos de simulação tem efeito significativo no desempenho de protocolos de comunicação sob investigação nessas redes. O padrão de mobilidade de nós é, portanto, um fator chave que impacta diretamente nos resultados da simulação e consequentemente no mundo real.

Esses dois últimos aspectos (sistemas de simulação e modelos de mobilidade) serão os abordados em maior profundidade neste trabalho. Será utilizado o simulador para redes *ad hoc* sem fio JiST/SWANS (*Java in Simulation*

*Time/Scalable Wireless Ad hoc Networks*) [7] e será feita uma avaliação do modelo de mobilidade *Random Waypoint*.

## 1.1 Motivação

Desde o início desta década tem-se intensificado as pesquisas em tecnologias relativas à proposta de Mark Weiser [25], chamada de Computação Ubíqua (*Ubiquitous Computing*), a qual propõe uma mudança do paradigma centrado em processo para o centrado em atividades do usuário final, resgatando um dos objetivos primordiais da computação em sua criação. Por prever uma mudança de rumo e uma inserção onipresente da computação no cotidiano, a computação ubíqua traz muitos desafios.

Dado o contínuo progresso técnico em comunicação e computação, parece que se está caminhando a uma total integração da computação nas atividades humanas. Previsões indicam que, em poucos anos, microprocessadores se tornarão pequenos e baratos o suficiente para serem embutidos em quase tudo - não somente em dispositivos digitais, carros, eletroeletrônicos, brinquedos, ferramentas, mas também em objetos (lápiz, por exemplo) e roupas. Todos esses artefatos devem estar entrelaçados e conectados em uma rede sem fio. De fato, a tecnologia espera uma revolução na qual bilhões de pequenos e móveis processadores estejam incorporados ao mundo físico, compondo objetos 'espertos' (*smart*), os quais sabem onde estão, se adaptam ao ambiente, fornecem serviços úteis em adição ao seu propósito original, formam redes espontâneas e altamente distribuída numa ordem de magnitude muito maior que a de hoje.

Esse cenário da computação ubíqua está sendo considerado como o novo paradigma do século 21, o qual permitirá o acoplamento do mundo físico ao mundo da informação e fornecerá uma abundância de serviços e aplicações onipresentes visando que usuários, máquinas, dados, aplicações e objetos do espaço físico interajam uns com os outros de forma transparente.

Redes móveis *ad hoc* sem fio caracterizam-se por não possuírem infraestrutura de redes nem pontos de acesso para coordenação e controle e toda a tarefa de acesso ao canal de comunicação é distribuída entre os próprios nós.

Dessa forma, elas são extremamente versáteis e têm um enorme potencial de serem amplamente utilizadas num futuro próximo no ambiente da computação ubíqua.

As redes *ad hoc* devem possuir protocolos de comunicação descentralizados e que façam pouco uso de recursos computacionais e energia dada a falta de infraestrutura. Estes protocolos devem ser testados sob condições que procurem retratar os mais diversos cenários práticos que um usuário deste sistema pode encontrar.

Um aspecto muito importante em redes *ad hoc* refere-se ao movimento realizado pelos nós. Neste sentido, modelos de mobilidade que procuram retratar padrões de comportamentos dos nós ou usuários têm sido propostos [1]. Entretanto, os estudos apresentados até hoje têm simplificado de sobremaneira a realidade de deslocamentos. Vale ressaltar que vários parâmetros relacionados ao desempenho da rede, tais como capacidade de canal, atraso de entrega de pacotes, relação sinal-ruído e interferência, como também taxa de colisão de pacotes, estão estritamente relacionados com o padrão de movimento executado pelos nós da rede [13].

Este trabalho trata da importância na escolha do Modelo de Mobilidade utilizado em simulações para avaliação de desempenho de protocolos de comunicação em redes *ad hoc*. O modelo de mobilidade mais usado na literatura é o *Random Waypoint* (RWP) [1], no entanto trabalhos anteriores mostraram que em algumas condições o RWP falha ao não atingir um estado estacionário e que por isso tal modelo não deve ser usado diretamente em simulações [6] e [9].

## 1.2 Objetivos

Apesar da popularidade do uso do modelo de mobilidade RWP, foi observado que, durante as simulações, este modelo apresenta uma falha caracterizada por não atingir um estado estável, uma vez que a velocidade média dos nós tende a zero com o passar do tempo para certos valores de parâmetros. Esta situação de instabilidade dos nós foi atestada intuitiva e formalmente [6], demonstrando que o modelo é falho e que não deve ser usado diretamente em simulações. Mais preocupante é que vários autores têm testado diversos protocolos de comunicação sob estas condições [19],[27],[28], o que pode levar a resultados imprecisos ou até mesmo inválidos.

A proposta deste trabalho é que uma alteração na escolha da função densidade de probabilidade da velocidade dos nós permita ao RWP sempre atingir o estado estacionário, estabilizando este modelo para uso nos simuladores de rede *ad hoc*. Através do resultado de simulações será demonstrado que estas modificações solucionam o problema do decaimento da velocidade média dos nós no modelo RWP original, permitindo que a mobilidade na rede sempre atinja o regime estacionário. Uma grande vantagem da modificação feita neste trabalho se dá pelo fato de que a alteração pode ser facilmente implementada nos simuladores de redes existentes, bastando para isso, realizar mudanças na escolha da função de distribuição de probabilidades das velocidades dos nós.

A proposta aqui apresentada dará continuidade ao estudo iniciado em [9], onde foi considerada as conseqüências no desempenho de redes *ad hoc* móveis para o caso de uma distribuição BETA(2,2) [22]. Porém os resultados deste trabalho monográfico são mais abrangentes do que em [9], visto que, além de avaliar vários valores para a função BETA( $\alpha, \beta$ ), consideram simulações de 2000 segundos, gerando resultados mais confiáveis para confirmar a tendência de instabilidade do RWP original e estabilidade do RWP modificado quando o tempo tende a infinito.

Este trabalho provê resultados fundamentados, de modo que fique claro que o RWP modificado retrata mais precisamente o ambiente real, conforme sua proposição inicial, tornando-o um modelo de mobilidade mais adequado para simulações.

### 1.3 Estrutura da Monografia

No **Capítulo 2**, disserta-se sobre a importância dos modelos de mobilidade e faz-se uma revisão do modelo de mobilidade RWP, ressaltando sua deficiência em atingir um regime estacionário caso o parâmetro da velocidade não seja apropriadamente escolhido.

No **Capítulo 3**, são realizados estudos das características estocásticas do RWP. É abordada principalmente a área de distribuições de probabilidade para aplicação direta na descrição formal do problema e elaboração de possíveis soluções. Neste sentido, é dada ênfase à distribuição BETA( $\alpha, \beta$ ) para determinação da velocidade dos nós na rede. Esta ação é tomada para obter-se maior respaldo no

que diz respeito à noção intuitiva do problema e respectiva solução. Para isso, faz-se uso de funções de densidade de probabilidade para escolha da velocidade de tal forma que condições estáveis da simulação sejam atingidas com precisão e dentro de parâmetros mais aceitáveis e confiáveis em relação às soluções existentes atualmente.

Com o intuito de mostrar a importância de se empregar um modelo de mobilidade estável ao analisar desempenho de redes *ad hoc*, são realizadas algumas simulações dos modelos RWP original e estabilizado utilizando o simulador de redes JiST/SWANS [7]. O **Capítulo 4** traz uma visão geral desse simulador, as extensões aplicadas sobre sua versão original e bibliotecas utilizadas.

O **Capítulo 5** traz resultados para simulações do modelo RWP original e do estabilizado onde são comparados seus desempenhos e, finalmente, no **Capítulo 6** apresentamos nossas conclusões e trabalhos futuros.

# Capítulo 2

## Modelos de Mobilidade: RWP e problemas relacionados

Pesquisas em redes sem fio são fundamentalmente dependentes de simulação. Derivar analiticamente o comportamento de um protocolo é muitas vezes impreciso, por outro lado, realizar experiências reais é oneroso, conseqüentemente, a maioria das publicações nesta área é baseada exclusivamente em simulação.

Numa avaliação de desempenho de um protocolo para redes *ad hoc*, o protocolo deve ser testado sobre condições que simulem o mais próximo possível um ambiente real: modelo de interferência e transmissão de sinal; limitação de *buffers*; modelo de transmissão de dados e movimento dos usuários da rede. Este capítulo aborda a importância na escolha do modelo de mobilidade, apresentará o *Random Waypoint (RWP)*, atualmente o modelo de mobilidade mais empregado na avaliação de desempenho de redes *ad hoc*, e faz referência aos problemas relacionados a esse modelo.

### 2.1 A importância na escolha do modelo de mobilidade

Nesta seção é demonstrado que o modelo de mobilidade pode ter um efeito significativo na avaliação de protocolos para redes *ad hoc*. Os resultados apresentados ilustram a importância em se escolher um modelo apropriado na execução das simulações.

O desempenho das MANETs é altamente influenciado pelo padrão de mobilidade dos nós na rede [1], [17], [18]. De fato, o comportamento da mobilidade é um dos mais importantes fatores ambientais que determinam o desempenho e influenciam no projeto da rede. Por exemplo, podemos esperar que num dado momento, numa área com alta densidade de nós, o número de colisões de pacotes de dados será bem maior que numa outra área menos densa. Essas densidades

podem ser consequência direta do padrão de movimento adotado pelos nós. Dessa forma, é importante que sejam desenvolvidos modelos que tenham precisão na captura dos dados em movimento e, ao mesmo tempo, eficiência em simulação e comunicação.

Há um grande número de pesquisadores que utilizam mobilidade em suas avaliações, entretanto vários trabalhos [6], [11], [17] e [14] indicam que a mobilidade é um aspecto que não está completamente investigado em redes dinâmicas. O efeito da mobilidade nos dispositivos nesse tipo de rede é um componente essencial na avaliação de desempenho das MANETs.

A fim de demonstrar isso, neste capítulo, foi utilizado o simulador para redes *ad hoc* JiST/SWANS para comparar o desempenho dos modelos de mobilidade implementados nesse simulador: *Random Walk*, *Teleport* [7] e um cenário de simulação sem mobilidade. No caso, usou-se 50 nós que se movem em uma rede de área retangular de dimensões  $1500 \times 500 \text{ m}^2$  com os nós executando cada um dos modelos de mobilidades acima citados. Para cada conjunto de parâmetros, realizaram-se médias sobre 10 diferentes cenários. Dos 50 nós, 10 geram tráfego a 2 pacotes UDP por segundo, tendo cada pacote tamanho de 64 bytes. Para roteamento foi utilizado o protocolo *ad hoc On-Demand Distance Vector* (AODV) [8]. As medidas foram extraídas a cada 100 segundos de simulação. A intenção não é produzir uma análise detalhada de desempenho, porém deseja-se ilustrar como a alteração no modelo de mobilidade influencia as medidas investigadas que têm impacto na sobrecarga, na vazão, e na qualidade de serviço das aplicações que utilizam redes *ad hoc* sem fio.

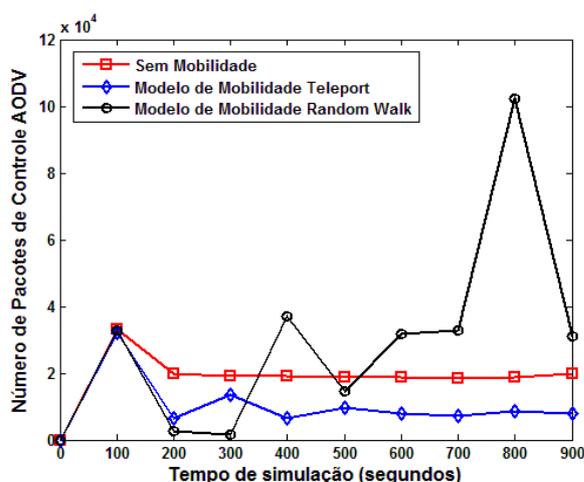
As medidas utilizadas foram as seguintes:

- **Número de pacotes de controle de roteamento:** Esta medida considera todos os pacotes de controle utilizados pelo AODV na descoberta e manutenção de rotas. Cada salto realizado pelo pacote é levado em consideração, assim um pacote que percorre cinco nós contabiliza cinco pacotes de controle. A Figura 2 ilustra o resultado obtido para essa medida na simulação.

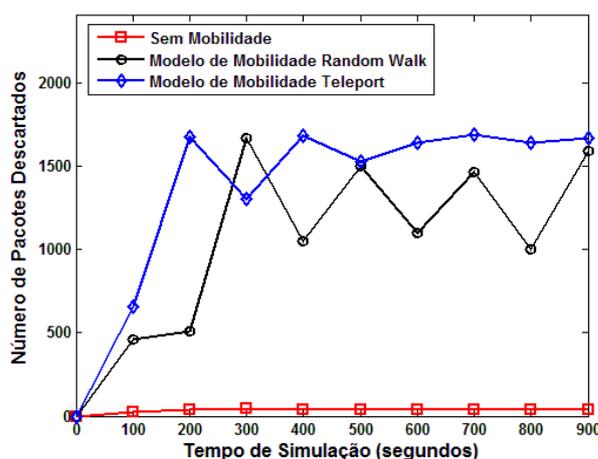
- **Pacotes de dados descartados:** Esta medida descreve o número de pacotes perdidos que ocorrem na rede ao se transmitir pacotes de dados (UDP). Estas perdas podem acontecer em qualquer das camadas (física, enlace, etc.)

devido a erros de transmissão, colisões, etc., o que impede que pacotes de dados sejam entregues aos seus destinos. A Figura 3 apresenta o resultado obtido para essa medida na simulação.

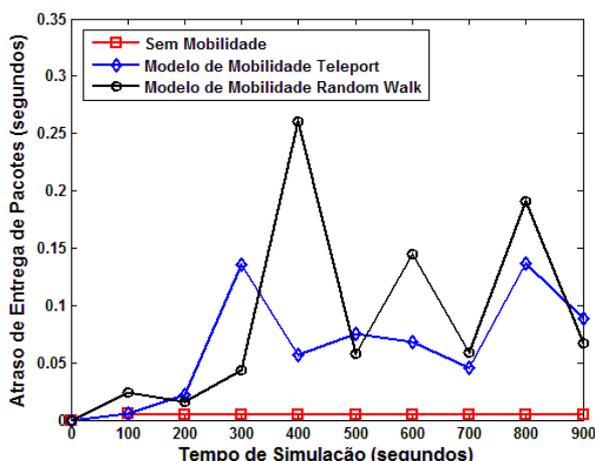
● **Atraso de entrega de pacote de dados:** Quantifica o tempo médio despendido pelos pacotes de dados (UDP) desde o momento de suas transmissões nas fontes até suas entregas com sucesso nos destinos. Para análise de rede, o atraso é medido em segundo. Os resultados para essa medida podem ser vistos na Figura 4.



**Figura 2.** Número de pacotes de controle de roteamento AODV em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade *teleport* e *Random walk*.



**Figura 3.** Número de pacotes de dados descartados em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade *teleport* e *Random walk*.



**Figura 4.** Atraso de entrega de pacotes de dados em função do tempo de simulação para cenários sem mobilidade e utilizando os modelos de mobilidade *teleport* e *Random walk*.

Em todos os casos anteriores foram considerados cenários idênticos, com exceção apenas do modelo de mobilidade adotado. Apesar disso nota-se grande diferença entre as curvas obtidas para uma mesma medida. A partir desses resultados, pode-se afirmar que o modelo de mobilidade adotado possui, realmente, grande influência nos resultados das simulações. Sendo assim, considera-se, neste trabalho, que o desenvolvimento de modelos de mobilidade mais adequados deve ser um campo levado bastante em consideração e alvo de estudos aprofundados.

## 2.2 O RWP

Esse modelo foi primeiramente usado por Johnson e Maltz na avaliação do protocolo de roteamento para redes *ad hoc Dynamic Source Routing (DSR)* [23], e posteriormente foi refinado pelo mesmo grupo. A versão refinada tornou-se o padrão para pesquisas em computação móvel. Por exemplo, no ACM MobiHoc 2002, dez artigos consideraram mobilidade e nove deles usaram o RWP.

O procedimento de funcionamento do RWP é determinado pela seguinte proposição, onde se considera uma rede de área retangular com dimensões  $X_{máx}$  x  $Y_{máx}$ .

**Proposição 1** - O modelo de mobilidade RWP original [1], [19]: **(i)** Escolhe-se aleatoriamente uma posição inicial para cada nó dada pela coordenada  $(x_0, y_0)$  onde  $x$  e  $y$  são uniformemente distribuídos no intervalo  $[0, X_{máx}]$  e  $[0, Y_{máx}]$ , respectivamente. **(ii)** Em seguida, cada nó seleciona um ponto de destino  $(x, y)$  uniformemente distribuído na área da rede, e uma velocidade  $v$  uniformemente distribuída no intervalo  $[V_{mín}, V_{máx}]$ , onde  $V_{mín}$  e  $V_{máx}$  são as possíveis velocidades mínima e máxima, respectivamente, que um nó pode escolher, sendo  $0 < V_{mín} < V_{máx}$ . **(iii)** Antes de iniciar seu movimento, o nó permanece parado por um tempo de pausa que pode ser fixo ou aleatório. **(iv)** Ao se expirar este tempo, o nó move-se em linha reta para o ponto de destino  $(x, y)$  com a velocidade escolhida  $v$ . **(v)** Ao atingir o destino, o nó repete o procedimento a partir de (ii), e assim sucessivamente até o fim da simulação.

O RWP é, sem dúvidas, o modelo de mobilidade mais utilizado pela comunidade científica até o momento [6],[9],[27],[28]. Embora esse modelo continue sendo usado em análise e simulação, recentemente foi descoberto que estava sendo utilizado de forma errônea. Em [6], Yoon *et al* demonstraram que o RWP não descreve bem a realidade, influenciando diretamente na precisão dos resultados de simulações. Nos últimos anos, grande esforço tem sido feito no desenvolvimento de modelos mais fiéis à realidade.

### 2.2.1 Problemas relacionados ao RWP

Alguns trabalhos já demonstraram alguns dos problemas do RWP [1],[6]. Em [1] foi demonstrado que os nós são inicialmente distribuídos aleatoriamente sobre a área de simulação. Essa distribuição inicial não representa bem a maneira como os nós se distribuem no decorrer da simulação. Isso acarreta a grande variação na porcentagem média de vizinhos à medida que a simulação decorre, o que pode resultar em grande variação no desempenho de protocolos testados sobre essas condições.

Perceba na Figura 5 que a grande variação no número de vizinhos nos primeiros 500 segundos de simulação pode gerar uma instabilidade na simulação que não traduz o cenário de mobilidade de uma rede real. Como grande parte dos trabalhos publicados considera tempos de simulação de algumas centenas de

segundos, a utilização do RWP em tais experimentos pode gerar resultados não confiáveis.

A porcentagem média de vizinhos é o somatório da porcentagem de nós que se encontram dentro do raio de transmissão de cada nó dividido pela quantidade de nós na rede.

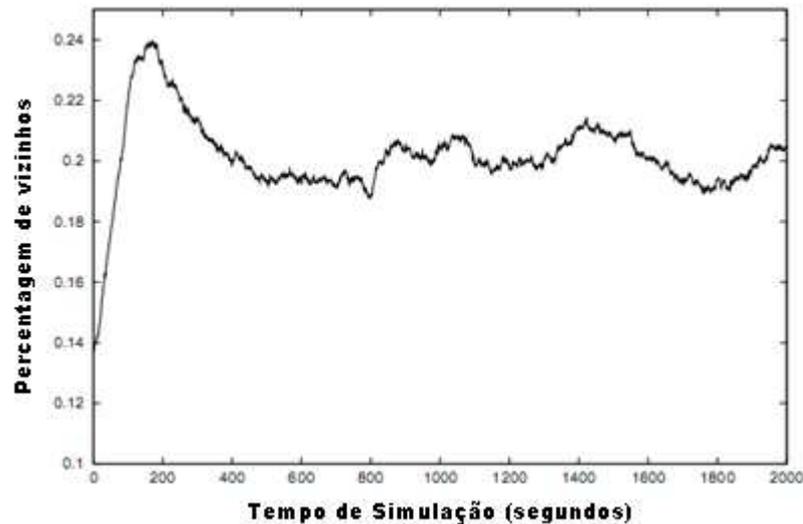


Figura 5. Porcentagem média de vizinhos vs. tempo [1].

Um outro problema do RWP está relacionado com a dinâmica de mobilidade deste modelo. Para demonstrar isso, Yoon *et al* [6] definiram a velocidade média instantânea dos nós na rede para um dado cenário de mobilidade como

$$\bar{v}(t) = \frac{\sum_{i=1}^N v_i(t)}{N}, \quad (1)$$

onde  $N$  é o número total de nós, e  $v_i(t)$  é a velocidade do  $i$ -ésimo nó no tempo  $t$ . Espera-se comumente que  $\bar{v}(t) = \frac{V_{máx} + V_{mín}}{2}$ , para todo  $t$ , entretanto isto não é verdade. Em [6], foi mostrado que sequer um regime estacionário pode ser atingido nos casos em que  $V_{mín} \rightarrow 0$ , ou seja, demonstrou-se que  $\bar{v}(t)$  decai ao longo do tempo tal que  $\lim_{t \rightarrow \infty} \bar{v}(t) = 0$ . Em termos práticos, isso significa dizer que ao se utilizar um intervalo  $(0, V_{máx}]$  para escolha das velocidades, os nós jamais atingirão um regime estacionário de mobilidade e isso pode levar a resultados errôneos

quando esse modelo é empregado para validar protocolos de comunicação em redes *ad hoc* móveis.

Uma explicação intuitiva para este comportamento é observar que o RWP seleciona destino e velocidade para cada nó na rede de forma aleatória e independente, onde cada nó mantém a velocidade escolhida até atingir o destino selecionado; alcançado o destino esse processo repete-se até o fim do tempo de simulação. Neste procedimento, os nós que escolheram velocidades baixas e grandes distâncias permanecerão neste movimento por um longo tempo e dependendo do período total de simulação eles sequer alcançarão seus destinos. Os nós que escolherem velocidades maiores e distâncias curtas alcançarão seus destinos e logo voltarão a selecionar novos rumos e velocidades. Ao repetirem o procedimento, esses novos nós podem escolher baixas velocidades e ficar presos a viagens longas que tenderão a dominar a velocidade média da rede levando a sua gradual estagnação.

A Figura 6 mostra a evolução temporal da velocidade média instantânea dos nós para intervalo de escolha (0,20] metros por segundo (m/s), com tempo de pausa nulo, numa rede de área retangular com dimensões 1500 x 500  $m^2$  contendo 50 nós. Os resultados foram obtidos realizando-se a média de 30 diferentes cenários de simulação utilizando-se o simulador de redes JiST/SWANS, sendo a média da velocidade calculada a cada segundo. Observa-se que no início, ou seja, quando  $t = 0$ ,  $\bar{v}(t) = \frac{V_{máx} + V_{mín}}{2} = V_{inic}$ , porém  $\bar{v}(t)$  decai ao longo do tempo consistentemente.

É importante frisar que o tempo de pausa não altera as conclusões, já que o mesmo não é a causa do decaimento da velocidade, embora induza a uma maior variação no início da simulação, conforme observado em [6]. Entretanto, essa flutuação é gradualmente reduzida com o passar do tempo não mudando a consistência do decaimento da média instantânea da velocidade. Dito isto, a fim de facilitar a análise, assume-se tempo de pausa sempre nulo neste trabalho.



**Figura 6.** Velocidade média instantânea em função do tempo no RWP original para intervalo da velocidade dado por  $(0,20]$  m/s e tempo de pausa nulo.

Esta situação de instabilidade dos nós é descrita no próximo capítulo, demonstrando analiticamente que o modelo é falho e que não deve ser usado em simulações.

## Capítulo 3

# O Modelo de Mobilidade *Random*

## *Waypoint* Estabilizado

A condição do RWP de selecionar a velocidade uniformemente no intervalo  $[V_{mín}, V_{máx}]$  no início de cada movimento traduz-se na seguinte função uniforme de densidade de probabilidade (ou distribuição uniforme de probabilidade) para a variável aleatória  $v$  que descreve a escolha da velocidade do nó

$$f_V(v) = \begin{cases} \frac{1}{V_{máx} - V_{mín}} & V_{mín} \leq v \leq V_{máx} \\ 0 & \text{caso contrário,} \end{cases} \quad (2)$$

onde  $0 < V_{mín} < V_{máx}$  e média  $E[V] = \int_0^\infty v f_V(v) dv = \frac{V_{máx} + V_{mín}}{2} = \bar{V}_{inic}$ . É importante ressaltar que a distribuição acima pode ser reproduzida nos simuladores a partir da função pseudo-aleatória *RAND* disponível em várias linguagens de programação, sendo necessário alterá-la de acordo com os valores de  $V_{mín}$  e  $V_{máx}$  gerando uma nova função *NEW\_RAND* pseudo-aleatória no intervalo  $[V_{mín}, V_{máx}]$ . Por exemplo, se *RAND* estiver definida no intervalo  $[0,1]$ , então tem-se que

$$NEW_{RAND} = (V_{máx} - V_{mín}) * RAND + V_{mín}. \quad (3)$$

Por outro lado, a distância percorrida por um nó ao executar o RWP é função da sua posição atual dado que a área da rede é limitada. Porém, de forma a facilitar a análise e não alterar as principais conclusões pode-se assumir que o nó escolhe o próximo destino até uma máxima distância  $R_{máx}$  da sua posição atual. Então, a função densidade de probabilidade da distância  $R$  percorrida por um nó é dada por [6]

$$f_R(r) = \begin{cases} \frac{2r}{R_{máx}^2} & 0 \leq r \leq R_{máx} \\ 0 & \text{caso contrário,} \end{cases} \quad (4)$$

tendo o seguinte valor médio

$$E[R] = \int_0^{\infty} r f_R(r) dr = \frac{2}{3} R_{m\acute{a}x}. \quad (5)$$

Partindo das Eqs. (2) e (4), pode-se mostrar que a função densidade de probabilidade para a variável aleatória  $S$  do tempo de viagem do nó é dada por [6]

$$f_S(s) = \begin{cases} \frac{2s}{3R_{m\acute{a}x}^2} (V_{m\acute{a}x}^2 + V_{m\acute{i}n}^2 + V_{m\acute{a}x}V_{m\acute{i}n}) & 0 \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \\ \frac{2R_{m\acute{a}x}}{3(V_{m\acute{a}x}-V_{m\acute{i}n})s^2} - \frac{2sV_{m\acute{i}n}^3}{3R_{m\acute{a}x}^2(V_{m\acute{a}x}-V_{m\acute{i}n})} & \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{i}n}} \\ 0 & s \geq \frac{R_{m\acute{a}x}}{V_{m\acute{i}n}}, \end{cases} \quad (6)$$

com o seguinte valor esperado

$$E[S] = \int_0^{\infty} s f_S(s) ds = \frac{2R_{m\acute{a}x}}{3(V_{m\acute{a}x}-V_{m\acute{i}n})} \ln\left(\frac{V_{m\acute{a}x}}{V_{m\acute{i}n}}\right). \quad (7)$$

Em [6], foi mostrado que a média temporal  $\bar{V}$  da velocidade para um dado nó pode ser obtida a partir da sua velocidade instantânea e das equações acima, tal que

$$\bar{V} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T v(t) dt = \frac{E[R]}{E[S]}, \quad (8)$$

que pode ser tomada como a velocidade média estacionária de um nó na rede. Substituindo as Eqs. (5) e (7) na Eq. (8), resulta a seguinte relação para a média temporal da velocidade ( $\bar{V}_{orig}$ ) de um nó no modelo RWP original [6]

$$\bar{V}_{orig} = \frac{V_{m\acute{a}x}-V_{m\acute{i}n}}{\ln\left(\frac{V_{m\acute{a}x}}{V_{m\acute{i}n}}\right)}. \quad (9)$$

Das Eqs. (7), (8) e (9), vê-se que  $\bar{V}_{orig} \rightarrow 0$  à medida que  $V_{m\acute{i}n} \rightarrow 0$ , visto que,  $E[R] \xrightarrow{V_{m\acute{i}n} \rightarrow 0} \infty$  confirmando o resultado da simulação apresentado na Figura 6. Dessa forma, para se atingir uma velocidade média não nula, os autores de [6] propuseram que fosse escolhida velocidade mínima sempre maior que zero ( $V_{m\acute{i}n} > 0$ ).

Essa proposta apresenta o viés de se restringir os possíveis parâmetros da simulação, o que, por conseguinte, restringe também o total de cenários possíveis

de representação de uma rede. Além disso, em termos práticos, quanto mais próximo de zero for  $V_{\min}$ , mais tempo a rede levará até atingir um regime estacionário com velocidade média constante.

No intuito de resolver o problema da estabilização da velocidade média dos nós na rede, tal que impeça seu decaimento quando se usa o intervalo  $(0, V_{\max}]$ , este trabalho propõe uma alteração no modelo RWP original. Demonstramos que, ao se tomar uma função densidade de probabilidade para a escolha das velocidades baseada na distribuição BETA(2,1) ou BETA(2,2) [22], ao invés da distribuição uniforme, resulta que a velocidade média dos nós na rede não mais decai a zero quando  $V_{\min} \rightarrow 0$ .

Esta alteração, entretanto, não modifica drasticamente o modelo de mobilidade que continua com suas demais características e a única variação na sua implementação nos simuladores de rede é uma mudança na chamada de função para escolha da velocidade dos nós.

### 3.1 Distribuição de Probabilidade BETA( $\alpha, \beta$ )

Em estatística uma distribuição beta é uma distribuição de probabilidade contínua, com dois parâmetros  $\alpha$  e  $\beta$  cuja função de densidade para valores  $0 < x < 1$  é [22]

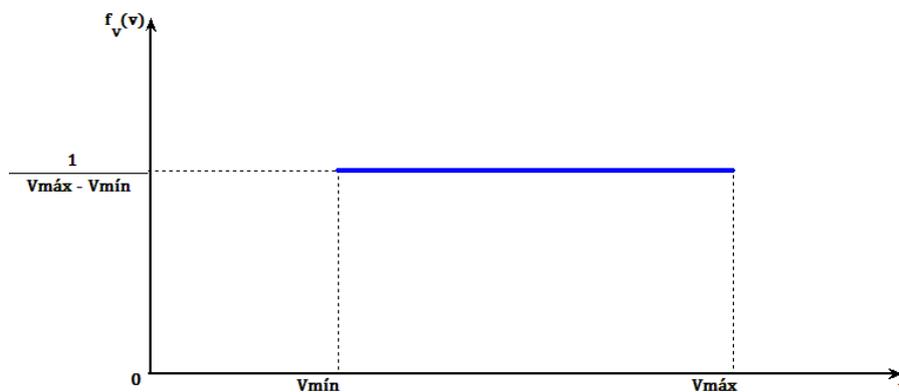
$$f(x, \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} du} \cdot \quad (10)$$

A escolha da distribuição BETA( $\alpha, \beta$ ) para escolha das velocidades dos nós na simulação justifica-se pelo fato de que, ao variarmos os parâmetros  $\alpha$  e  $\beta$ , alguns casos particularmente interessantes são reproduzidos. Ou seja, quando  $\alpha=1$  e  $\beta=1$  obtém-se o RWP original com a distribuição uniforme para a velocidade; quando  $\alpha=2$  e  $\beta=2$  recaímos nos resultados obtidos em [9]; e quando  $\alpha=2$  e  $\beta=1$ , o caso da distribuição linear é alcançado conforme observado em [5]. Desta forma, verifica-se que a distribuição BETA( $\alpha, \beta$ ) é uma generalização dos casos anteriores.

### 3.1.1 Distribuição de Probabilidade BETA(1,1)

A derivação da função BETA(1,1) recai no caso da função de densidade de probabilidade uniforme derivada no capítulo anterior (ver Eq.(2)) e pode ser representada pelo seguinte gráfico ilustrado na Figura 7.

Como demonstrado anteriormente, essa é a função utilizada para escolha das velocidades no RWP original e que ocasiona decaimento da velocidade instantânea média dos nós da simulação.



**Figura 7.** Gráfico da função densidade de probabilidade, descrita na Eq. (2), para escolha das velocidades dos nós no modelo RWP modificado.

### 3.1.2 Distribuição de Probabilidade BETA(2,2)

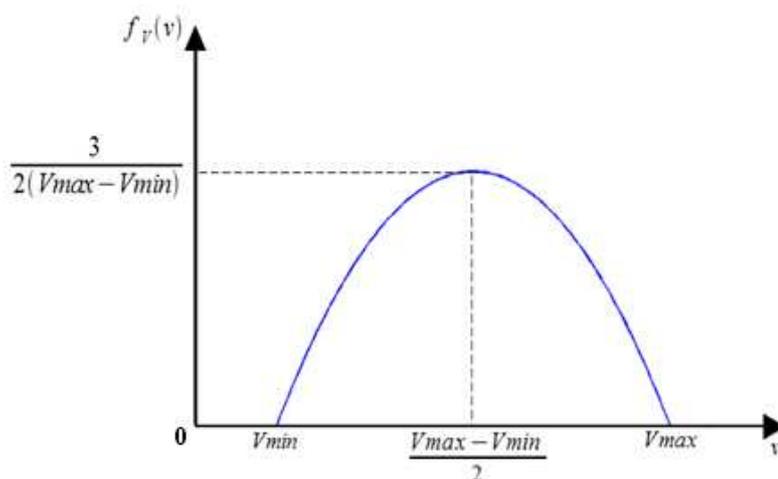
Como demonstrado em [9] a função densidade de probabilidade para a escolha das velocidades dos nós quando utilizando a função BETA(2,2) é

$$f_V(v) = \begin{cases} -\frac{6}{(V_{máx}-V_{mín})^3} \left(v - \frac{V_{máx}+V_{mín}}{2}\right)^2 + \frac{3}{2(V_{máx}-V_{mín})} & V_{mín} \leq v \leq V_{máx} \\ 0 & \text{caso contrário.} \end{cases} \quad (11)$$

A Figura 8 ilustra graficamente essa função.

Ao se utilizar a função de densidade de probabilidade dada pela Eq. (11) para escolha das velocidade dos nós no modelo RWP, resulta que a função densidade de probabilidade para a variável aleatória S do tempo de viagem de um nó é dada por

$$f_S(s) = \begin{cases} \frac{2Ks}{R_{m\acute{a}x}^2(V_{m\acute{a}x}-V_{m\acute{i}n})^3} & 0 \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \\ -\frac{12R_{m\acute{a}x}^3}{5(V_{m\acute{a}x}-V_{m\acute{i}n})^3s^4} + \frac{3(V_{m\acute{a}x}+V_{m\acute{i}n})R_{m\acute{a}x}^2}{(V_{m\acute{a}x}-V_{m\acute{i}n})^3s^3} - \frac{4V_{m\acute{a}x}V_{m\acute{i}n}R_{m\acute{a}x}}{(V_{m\acute{a}x}-V_{m\acute{i}n})^3s^2} + \frac{V_{m\acute{i}n}^4(V_{m\acute{a}x}-\frac{3V_{m\acute{i}n}}{5})s}{R_{m\acute{a}x}^2(V_{m\acute{a}x}-V_{m\acute{i}n})^3}, & \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{i}n}} \\ 0 & s \geq \frac{R_{m\acute{a}x}}{V_{m\acute{i}n}} \end{cases} \quad (12)$$



**Figura 8.** Gráfico da função densidade de probabilidade , descrita na Eq. (11), para escolha das velocidades dos nós no modelo RWP modificado.

Uma vez determinada a densidade de probabilidade da variável aleatória  $S$  do tempo de viagem, é possível obter sua média tal que

$$E[S] = \int_0^{\infty} s f_S(s) = \frac{R_{m\acute{a}x}}{(V_{m\acute{a}x}-V_{m\acute{i}n})^3} \left[ \frac{2K}{3V_{m\acute{a}x}^3} + \frac{9(V_{m\acute{a}x}^2 - V_{m\acute{i}n}^2)}{5} - 4V_{m\acute{a}x}V_{m\acute{i}n} \ln\left(\frac{V_{m\acute{a}x}}{V_{m\acute{i}n}}\right) + \frac{(V_{m\acute{a}x} - \frac{3V_{m\acute{i}n}}{5}) \left( V_{m\acute{i}n} - \frac{V_{m\acute{i}n}^4}{V_{m\acute{a}x}^3} \right)}{3} \right]. \quad (13)$$

Usando as Eqs. (5), (8) e (13) obtém-se a seguinte relação para a média temporal da velocidade, no estado estacionário, de um nó no RWP modificado

$$\bar{V}_{modif} = \frac{E[R]}{E[S]} = \frac{2(V_{m\acute{a}x}-V_{m\acute{i}n})^3}{\frac{2K}{V_{m\acute{a}x}^3} + \frac{27(V_{m\acute{a}x}^2 - V_{m\acute{i}n}^2)}{5} - 12V_{m\acute{a}x}V_{m\acute{i}n} \ln\left(\frac{V_{m\acute{a}x}}{V_{m\acute{i}n}}\right) + V_{m\acute{i}n} \left( V_{m\acute{a}x} - \frac{3V_{m\acute{i}n}}{5} \right) \left( 1 - \frac{V_{m\acute{i}n}^3}{V_{m\acute{a}x}^3} \right)}. \quad (14)$$

A partir desta, pode-se verificar que

$$\lim_{V_{m\acute{i}n} \rightarrow 0} \bar{V}_{modif} = \frac{V_{m\acute{a}x}}{3}, \quad (15)$$

ou seja, a média temporal da velocidade dos nós é igual a um terço da velocidade máxima de simulação, no estado estacionário, quando  $V_{min} \rightarrow 0$ , ao contrário do que acontece no RWP original, em que a velocidade decai ao longo do tempo tal que a média temporal tende a zero.

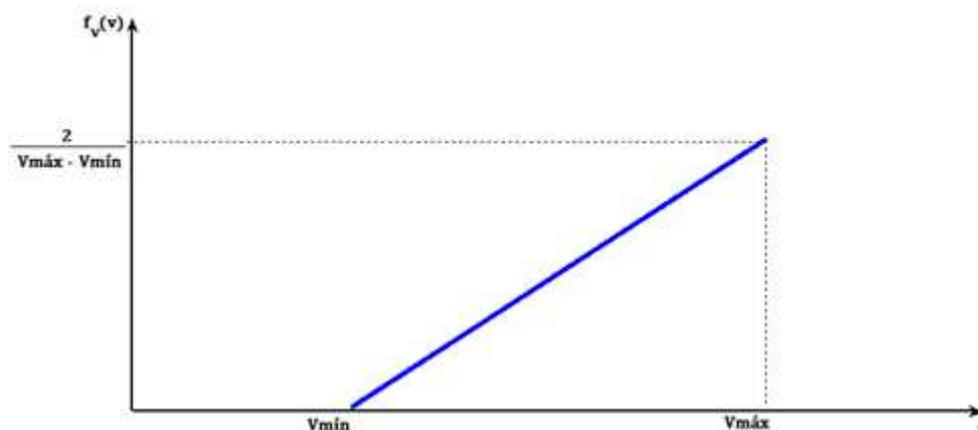
A utilização da função de distribuição de probabilidades BETA(2,2) soluciona o problema do decaimento da velocidade média dos nós no modelo RWP original, e permite que a mobilidade na rede atinja um regime estacionário quando  $V_{min} \rightarrow 0$ . No entanto, essa função ainda não resulta no valor de velocidade média esperado pelos criadores do RWP,  $\frac{V_{máx}+V_{mín}}{2}$ . Para se chegar a esse valor desejado, propusemos o desenvolvimento analítico da distribuição BETA(2,1) análogo ao feito para a função BETA(2,2) em [9].

### 3.1.3 Distribuição de Probabilidade BETA(2,1)

A função densidade de probabilidade para a escolha das velocidades dos nós quando utilizando a função BETA(2,1) é [22]

$$f_V(v) = \begin{cases} \frac{2(v-V_{Min})}{(V_{Max}-V_{Min})^2} & V_{Min} \leq v \leq V_{Max} \\ 0 & \text{Caso contrário.} \end{cases} \quad (16)$$

A Figura 9 ilustra graficamente essa função.



**Figura 9.** Gráfico da função densidade de probabilidade, descrita na Eq. (16), para escolha das velocidades dos nós no modelo RWP modificado.

Ao se utilizar a função de densidade de probabilidade dada pela Eq. (16) para escolha das velocidades dos nós no modelo RWP, resulta que a função densidade de probabilidade para a variável aleatória S do tempo de viagem de um nó é dada por

$$f_S(s) = \begin{cases} \frac{1}{(V_{m\acute{a}x}-V_{m\acute{i}n})^2} \left( \frac{V_{m\acute{a}x}^4}{R_{m\acute{a}x}^2} - \frac{4V_{m\acute{a}x}^3 V_{m\acute{i}n}}{3R_{m\acute{a}x}^2} + \frac{V_{m\acute{i}n}^4}{3R_{m\acute{a}x}^2} \right) s & 0 \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \\ \frac{1}{(V_{m\acute{a}x}-V_{m\acute{i}n})^2} \left[ (R_{m\acute{a}x}^2) \frac{1}{s^3} - \left( \frac{4V_{m\acute{i}n} R_{m\acute{a}x}}{3} \right) \frac{1}{s^2} + \left( \frac{V_{m\acute{i}n}^4}{3R_{m\acute{a}x}^2} \right) s \right] & \frac{R_{m\acute{a}x}}{V_{m\acute{a}x}} \leq s \leq \frac{R_{m\acute{a}x}}{V_{m\acute{i}n}} \\ 0 & \text{Caso contr\acute{a}rio.} \end{cases} \quad (17)$$

Calculando-se o valor esperado de S, ou seja, E[S], obtém-se o valor da velocidade média, no estado estacionário, a partir da função densidade de probabilidade dada pela Eq.(16), igual a

$$\bar{V}_{estabilizado} = \frac{2(V_{m\acute{a}x}-V_{m\acute{i}n})^2}{V_{m\acute{a}x} - \frac{4V_{m\acute{i}n}}{3} + \frac{V_{m\acute{i}n}^4}{3V_{m\acute{a}x}} + 3(V_{m\acute{a}x}-V_{m\acute{i}n}) - 4V_{m\acute{i}n} \ln\left(\frac{V_{m\acute{a}x}}{V_{m\acute{i}n}}\right) + \frac{V_{m\acute{i}n}}{3} - \frac{V_{m\acute{a}x}^4}{3V_{m\acute{a}x}}} \cdot (19)$$

A partir desta, pode-se verificar que

$$\lim_{V_{m\acute{i}n} \rightarrow 0} \bar{V}_{estabilizado} = \frac{V_{m\acute{a}x}}{2}, \quad (20)$$

ou seja, a média temporal da velocidade dos nós é igual à metade da velocidade máxima de simulação, no estado estacionário, quando  $V_{m\acute{i}n} \rightarrow 0$ , ao contrário do que acontece no RWP original em que a velocidade decai ao longo do tempo tal que a média temporal tende a zero

Note que para a função BETA(1,1) os resultados são idênticos aos observados para a distribuição uniforme de probabilidade do RWP original, acarretando uma queda constante na velocidade instantânea média da simulação. Para a função BETA(2,2) chega-se a um estado estacionário após certo tempo de simulação. Essa distribuição poderia ser usada para estabilizar o *Random Waypoint*, no entanto, essa função ainda não resulta no valor de velocidade média esperado pelos criadores do RWP,  $\frac{V_{m\acute{a}x}+V_{m\acute{i}n}}{2}$ , sendo razoável, portanto, que se busque uma distribuição de probabilidade que alcance esse valor. Como observado para a função BETA(2,1) a velocidade média no estado estacionário é  $V_{m\acute{a}x}/2$ , valor

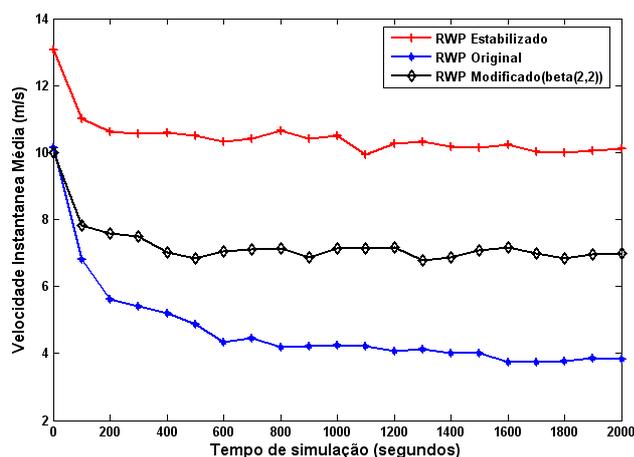
considerado ideal quando  $V_{mín} \rightarrow 0$ . Portanto, as comparações entre o RWP original e o RWP proposto neste trabalho são feitas considerando a função de densidade de probabilidade BETA(2,1) para escolha da velocidade dos nós.

Observe que esta função pode ser reproduzida nos simuladores a partir da distribuição BETA(2,1), a qual está definida no intervalo [0,1], através da seguinte modificação designada NEW\_BETA(2,1):

$$NEW\_BETA(2,1) = (V_{máx} - V_{mín}) * BETA(2,1) + V_{mín} \quad (21)$$

**Proposição 2** - O modelo RWP estabilizado: o modelo *Random Waypoint* estabilizado constitui-se da Proposição 1, alterando-se a escolha das velocidades dos nós baseada na função densidade de probabilidade descrita na pela Eq. (16), ao invés da distribuição uniforme mostrada na Eq. (2).

A fim de corroborar a análise realizada, foi simulada em C++ uma rede com dimensões  $1500 \times 500 \text{ m}^2$  utilizando 50 nós empregando os modelos de mobilidade RWP original, modificado e estabilizado. O resultado comparativo está ilustrado na Figura 10, onde foram realizadas a média de 30 diferentes cenários de simulação, sendo a média da velocidade dos nós exibida a cada 100 segundos.



**Figura 10.** Velocidade instantânea média para simulações utilizando o RWP original, modificado (BETA(2,2)) e estabilizado (BETA(2,1)).

Como pode ser facilmente verificado visualmente, o gráfico confirma os resultados obtidos da análise feita neste capítulo. Confirma-se a tendência de queda

na velocidade instantânea média das simulações quando utilizando-se o RWP original, em concordância com a Eq.(9); com o RWP modificado alcança-se um estado estacionário tendo  $\bar{V}_{modif}$  bastante próximo de  $V_{máx}/3$  em concordância com a Eq.(15) e com o RWP estabilizado tem-se  $\bar{V}_{estabilizado}$  bastante próximo de  $V_{máx}/2$  em concordância com a Eq.(20).

# Capítulo 4

## Visão Geral do Simulador

### JiST/SWANS

Da física à biologia, das previsões meteorológicas à previsão de desempenho de um novo processador, pesquisadores em muitos campos da ciência dependem cada vez mais de simulações em software para modelar vários fenômenos ou cenários que muitas vezes não podem ser satisfatoriamente expressos analiticamente nem facilmente reproduzidos e observados empiricamente. Modelos de simulação são codificados como programas de eventos discretos e os eventos são mensagens que carregam consigo um componente temporal. A progressão de eventos do simulador não possui tempo igual aos eventos do mundo real, mas simula um tempo real através dessa componente à medida que a simulação progride.

Simuladores são ferramentas científicas importantes e o foco de um grande número de cientistas da computação que presam pela eficiência na execução e construção das simulações. O JiST (**J**ava in **S**imulation **T**ime) é um novo simulador de eventos discretos escrito em Java com uma série de novas funcionalidades e características exclusivas.

Pesquisas em redes sem fio são fundamentalmente dependentes de simulação. Derivar analiticamente o desempenho e o comportamento de um protocolo em larga escala é muitas vezes impreciso. Por outro lado, realizar experiências reais é oneroso: aquisição de centenas de dispositivos, a gestão de seu software de configuração, controlar um experimento distribuído, fazer os dispositivos adquirirem movimento real e isolar o experimento de interferências são apenas algumas das dificuldades que tornam esforços empíricos complexos. Consequentemente, a maioria das publicações nesta área é baseada exclusivamente em simulação.

Sobre o JiST roda o SWANS ou JiST/SWANS [7], acrônimo em língua inglesa de *Java in Simulation Time/Scalable Wireless Ad hoc Network*

*Simulator*, um simulador resultante das necessidades de investigações atuais nas redes de grande porte e que provê todos os mecanismos necessários para simular MANETs escaláveis.

É objetivo deste trabalho, familiarizar o leitor com a ferramenta, para que este torne-se apto a manuseá-la e reproduzir os resultados aqui obtidos. Este capítulo traz uma visão geral do JiST/SWANS, as extensões aplicadas sobre sua versão original e bibliotecas utilizadas. Também justificamos o uso da ferramenta neste trabalho demonstrando suas vantagens em relação aos simuladores correntemente mais utilizados.

## 4.1 JiST

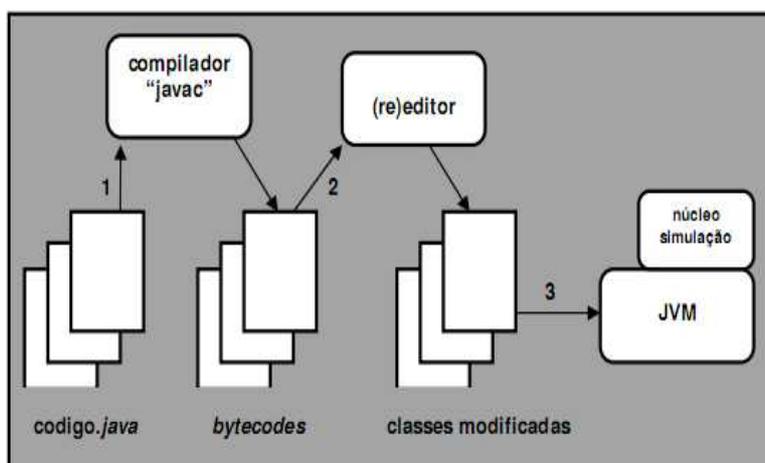
JiST [7], é uma abreviação de *Java in Simulation Time*, trata-se de uma plataforma e um simulador de eventos discretos de alto desempenho [16]. Esse simulador de código aberto é um protótipo desenvolvido por Rimon Barr [24] na Universidade de Cornell. A idéia principal dele é transformar a máquina virtual Java em um escalonador de eventos, modificando a maneira como as chamadas de métodos entre as entidades de simulação são conduzidos.

Ele é um protótipo de uma nova abordagem para construir simuladores de propósito geral de eventos discretos baseada em máquina virtual. A plataforma resultante é bastante eficiente. Seu desempenho é superior a simuladores tanto em relação a consumo de memória quanto a tempo de processamento [24].

Essa abordagem também é bastante flexível, possibilitando realizar-se facilmente mudanças de código e, por conseguinte, otimizações. Essa transparência é uma característica chave: códigos de simulação do JiST não precisam ser escritos em linguagens criadas especificamente para escrever simulações, ao invés disso, o JiST converte uma máquina virtual java numa plataforma de simulação. Portanto simulações em JiST são escritas em java, compiladas utilizando-se um compilador java padrão e executadas sobre uma maquina virtual java.

As entidades de simulação do JiST são executadas e compiladas como classes regulares de Java. Portanto, todas as vantagens da plataforma Java como a independência, o grande apoio de bibliotecas, familiaridade com a linguagem, se

aplicam também para o JiST. Para introduzir a semântica de “tempo simulação”, o JiST divide as simulações em “Entidades”, que são na realidade classes Java normais marcadas com a interface *JistAPI.Entity* (ver Figura 11), cujos métodos públicos são sempre chamados através de um escalonador. Antes da execução da simulação, as classes de uma simulação JiST são transformadas ou re-escritas usando o editor de *bytecodes*. Em seguida, a simulação é executada pelo núcleo da simulação que é composto por uma máquina virtual Java padrão.



**Figura 11.** Arquitetura do JiST.

Primeiro os programas de simulação são escritos em linguagem Java normal e compilados em *bytecodes* usando um compilador Java padrão. Em seguida, essas classes compiladas são modificadas dinamicamente através do re-editor (*rewriter*) de *bytecode* para acrescentar a semântica de “tempo de simulação”, pois cada entidade no JiST possui seu próprio tempo de simulação que determina qual o tempo de chamadas de outras entidades que estão escalonadas.

Nessa etapa, as chamadas para métodos públicos de entidades são desacoplados, isto é, as chamadas do método irão retornar imediatamente e serão colocadas numa pilha de eventos. O código correspondente ao método do objeto invocado é executado somente quando o tempo de entidade alcança o mesmo tempo de simulação. Para uma entidade avançar o seu tempo é utilizado o método *JistAPI.Sleep()*. Após os *bytecodes* serem re-escritos, a simulação segue para o núcleo do simulador que juntamente com a máquina virtual Java executa a simulação [24].

É comum que, à primeira vista, argumentar contra o uso de um simulador escrito em Java, por motivos de desempenho. De fato a grande maioria dos atuais sistemas de simulação foram escritos em C e C ++, ou seus derivados. No entanto, JiST apresenta desempenho superior que o ns-2 [21] e o GloMoSim [15].

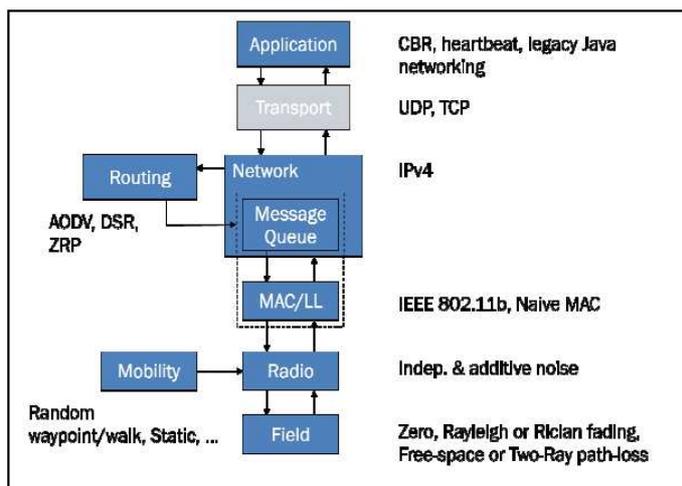
## 4.2 SWANS

SWANS foi inicialmente criado porque os simuladores de redes existentes não supriam todas as necessidades dos pesquisadores. Seu desempenho serviu como prova do potencial da abordagem baseada em máquina virtual. Ele é organizado em componentes independentes de software que podem ser compostos de forma a completar as configurações de uma rede sem fio ou rede de sensores mostrando-se uma arquitetura bastante modular.

Este simulador apresenta capacidades similares às do NS-2 [21] e do GloMoSim [15], além de dar suporte à simulação de redes maiores, ou seja, com uma maior quantidade estações. A comunidade acadêmica vem investindo muito nesse simulador, desenvolvendo novas funcionalidades como protocolos de camada de acesso ao meio (MAC) e de roteamento, novos tipos de modelos de mobilidade e modelos de energia [20].

Como o JiST/SWANS é dividido em componentes, existem instâncias de entidades de cada tipo de componente disponível que estão esquematizadas na Figura 12. Há componentes para protocolos da camada de rede, roteamento, acesso ao meio (MAC), rádios de transmissão, recepção, ruído, assim como modelos de propagação do sinal e de desvanecimento (*fading*), além de modelos de mobilidades.

Como mencionado anteriormente, o software SWANS é projetado com diferentes e independentes módulos que podem ser combinados para formar uma rede. Cada entidade JiST é responsável por armazenar seu estado interno e interagir com outros componentes via interfaces. Cada nó SWANS é uma entidade. Os nós também são formados por outras entidades da pilha da camada do modelo OSI.



**Figura 12.** Arquitetura do SWANS.

Para maiores explicações e informações sobre o JiST/SWANS, procure pelo guia do usuário SWANS [7].

### 4.3 Instalação e Configuração

O JiST/SWANS é compatível com diversas plataformas como Linux, Windows e Solaris. O presente trabalho utilizou a versão do JiST/SWANS 1.0.6 com o auxílio do ambiente de desenvolvimento Eclipse para facilitar a codificação e depuração na plataforma Windows.

A instalação do JiST/SWANS pode ser feita de maneira bem simples. Primeiramente, fazendo o *download* do JiST/SWANS em formato .zip que pode ser obtido em [24]. Para descompactar o arquivo, é preciso ter algum programa descompactador zip instalado no sistema. Uma versão do JiST/SWANS com todas as modificações do Apêndice A e Anexo A pode ser obtida em [26].

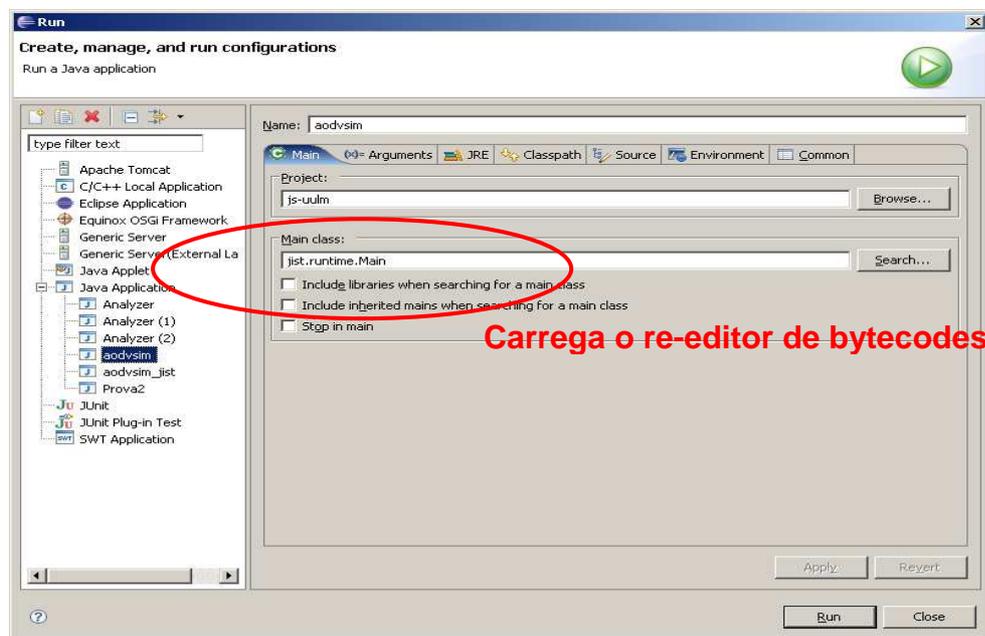
Qualquer ambiente de programação de desenvolvimento Java pode ser usado. Neste trabalho, foi utilizado o Eclipse Ganymede. Depois de instalados o descompactador zip e o Eclipse, basta seguir os passos descritos abaixo para assegurar a correta instalação do JiST/SWANS:

1. Descompacte o código fonte do JiST/SWANS em algum diretório;
2. Inicie o Eclipse;

3. Do menu “File” selecione “New => Project”;
4. Selecione “Java Project”, e clique em Next;
5. Defina um nome para o seu projeto JiST/SWANS. Selecione “Create project from existing source” da opção “Contents” na mesma janela. Navegue até o código fonte do JiST/SWANS descompactado no diretório escolhido no passo 1;
6. Clique *Finish* na última janela. Seu projeto deve aparecer na guia Package Explorer.

Após esses passos é possível desenvolver e executar programas de simulação assim como alterar o código do simulador pelo Eclipse.

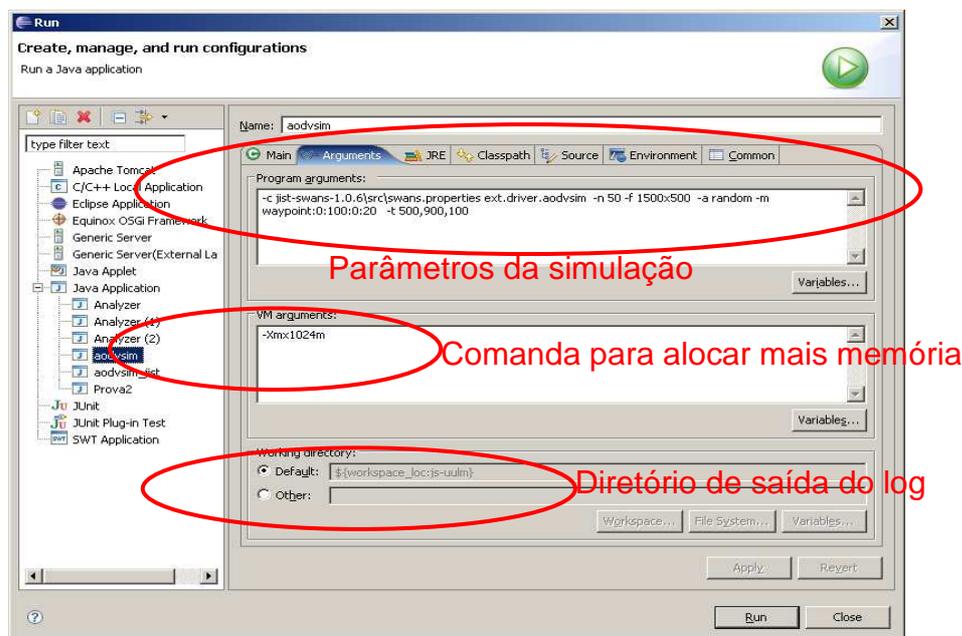
Para rodar a simulação via eclipse clique com o botão direito no arquivo aodvsim.java em “js-ext/src/ext/driver” e escolha a opção “run as->run configuration”, a caixa de diálogo ilustrada na Figura 13 irá abrir.



**Figura 13.** Carregando o re-editor de bytecodes.

No campo Main class digite “jst.runtime.Main” para carregar o re-editor de bytecodes

Na mesma caixa, na aba *Arguments* (ver Figura 14) insira os argumentos da simulação (exemplo: “-c jst-swans-1.0.6\src\swans.properties ext.driver.aodvsim -n 50 -f 1500x500 -m Waypoint:0:100:0:20 -t 200,2200,0”). Para maiores esclarecimento sobre cada um desses parâmetros procure pelo guia do usuário SWANS [7].



**Figura 14.** Inserindo os parâmetros da simulação.

Todos os parâmetros da simulação podem ser vistos no próprio código fonte do simulador ou em [20]. Depois de finalizada a simulação será terminado o preenchimento/criação de um arquivo de log no diretório de saída do eclipse.

Para extrair as medidas é necessário executar no eclipse o aplicativo Analyzer no pacote analyzer/src/. Esse aplicativo recebe três parâmetros de entrada: tempo de simulação, tempo de início da simulação e intervalo em que a medida será feita. Neste executável, o arquivo de log chamado “NET.log” é lido.

O conteúdo desse arquivo de log é controlado pelo arquivo de propriedades do projeto diretório do chamado “jst-swans-1.0.6\src\swans.properties” que é passado quando você roda sua simulação. As saídas dessa última execução serão as medidas de desempenho consideradas neste trabalho: sobrecarga, perda de pacotes e atraso.

## Capítulo 5

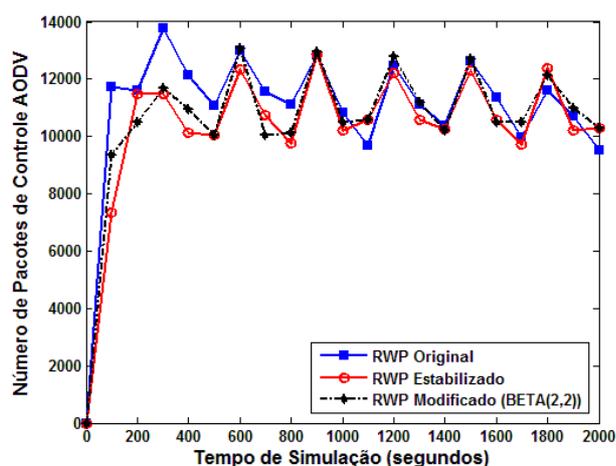
# Simulação e Análise dos Resultados

Com o objetivo de produzir resultados relevantes, foram consideradas medidas de desempenho análogas aos trabalhos anteriores relacionados [6], [19]: número de pacotes de controle roteamento (sobrecarga), pacotes de dados descartados e atraso de entrega de pacotes de dados. Essas medidas são as mesmas descritas anteriormente no Capítulo 2, com a diferença de que agora considera-se tempo de simulação igual a 2000 segundos.

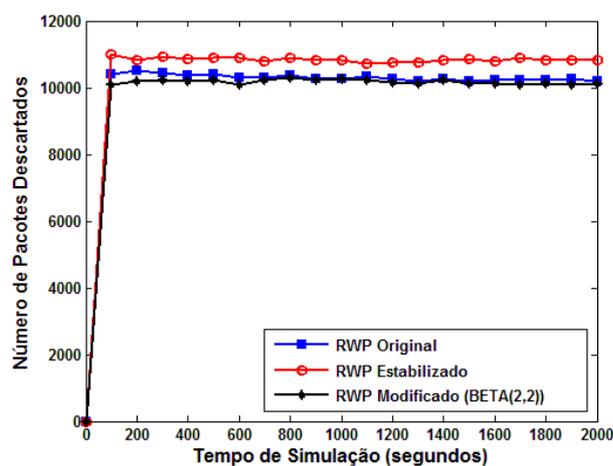
Foi utilizado o simulador para redes *ad hoc* JiST/SWANS para comparar o desempenho do RWP original e do RWP estabilizado. No caso, considerou-se 50 nós que movem-se com velocidade no intervalo (0,20] m/s, em uma rede de área retangular de dimensões 1500 x 500  $m^2$ . Para cada conjunto de parâmetros, foram calculadas médias sobre 30 diferentes cenários. Dos 50 nós, 30 geram tráfego a 4 pacotes UDP por segundo, tendo cada pacote tamanho de 64 bytes. Para roteamento foi utilizado o protocolo *ad hoc On-Demand Distance Vector* (AODV) [8].

Os resultados de cada medida avaliada foram calculados a cada 100 segundos, gerando desta forma 20 pontos diferentes de zero para um tempo total de 2000 segundos. Nossa intenção não é produzir uma análise detalhada de desempenho, porém desejamos ilustrar como a proposta de alteração no modelo RWP estabiliza as medidas investigadas.

A Figura 15 ilustra os resultados obtidos para o número de pacotes de controle de roteamento enviados; a Figura 16 ilustra os resultados obtidos para o número de pacotes de dados enviados; a Figura 17 ilustra os resultados obtidos para o atraso de entrega de pacotes de dados. Essas medidas são as mesmas descritas no Capítulo 2.

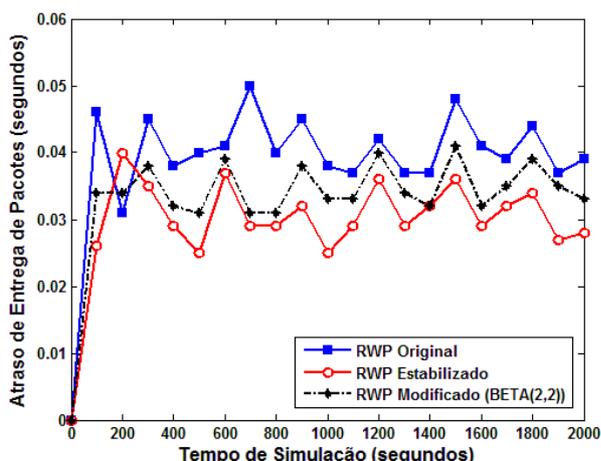


**Figura 15.** Número de pacotes de controle de roteamento AODV em função do tempo de simulação.



**Figura 16.** Número de pacotes de dados descartados em função do tempo de simulação.

Os resultados obtidos para essas três medidas não assumiram os valores esperados. Foi tomado como hipótese desse projeto monográfico que o RWP original possui uma tendência de diminuição contínua da velocidade instantânea média da simulação quando tomado um intervalo de escolha de velocidades em (0,20] m/s e que considerando-se uma função densidade de probabilidades baseada numa distribuição beta chegar-se-ia a um estado estacionário dessa velocidade.



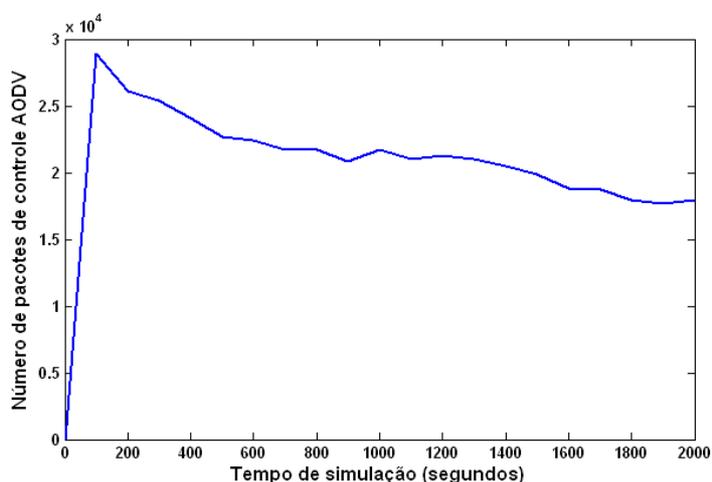
**Figura 17.** Atraso de entrega de pacote de dados em função do tempo de simulação.

Essa característica deveria ocasionar uma menor perda de pacotes à medida que a rede fica mais estática e, por conseguinte, haveria menor necessidade de troca de pacotes de controle para estabelecer e manter as rotas no AODV quando do uso do modelo de mobilidade RWP original. Isso não se confirma nos resultados visualizados nas Figuras 15 e 16.

De posse dos resultados destes experimentos, duas hipóteses podem ser levantadas para justificar esses resultados. Primeiro, que nossa hipótese inicial é precipitada: não há tendência de queda da velocidade instantânea média da simulação para o RWP original e a mudança da função densidade de probabilidade da velocidade da simulação não afeta significativamente os resultados de simulações, como apresentado nas Figuras 15, 16 e 17. Segundo, que o simulador de redes utilizado não produza resultados confiáveis, já que o JiST/SWANS é um simulador de código aberto, relativamente recente e com um grupo de usuários ainda restrito.

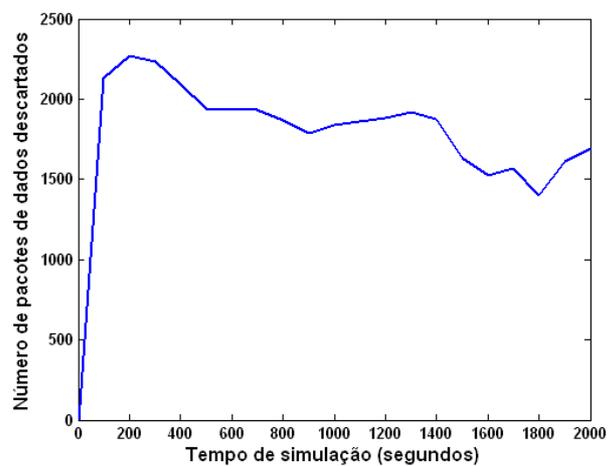
Com relação à primeira hipótese, todo desenvolvimento analítico do Capítulo 3 comprova que realmente há uma tendência de queda da velocidade da simulação quando do uso do RWP original, e essa tendência é anulada mudando-se a função uniforme de densidade de probabilidade para uma função baseada numa distribuição BETA(2,1) ou BETA(2,2) para escolha das velocidades dos nós.

Já quanto à segunda hipótese, o grande número de aspectos de uma rede real que precisam ser traduzidos para linguagem de programação torna bastante complexa a construção de um ambiente de simulação. Além disso, a grande quantidade de correções feitas no simulador durante o decorrer desse trabalho (ver Apêndice A e Anexo A) leva-nos a acreditar que existam outros problemas que tornem a ferramenta não confiável. Ademais, foram reproduzidos cenários de simulação no simulador NS-2 idênticos aos gerados no JiST/SWANS. Os resultados obtidos apresentam a tendência de queda para o RWP original, de acordo com o que era esperado, e podem ser vistos nas Figuras 18 e 19 a seguir. Em [6], também foram apresentados resultados de simulações utilizando-se o simulador de redes NS-2, compatíveis com o resultado do modelo analítico apresentado neste trabalho.



**Figura 18.** Número de pacotes de controle de roteamento AODV em função do tempo de simulação obtidos do simulador NS-2 para o RWP original.

Portanto, considera-se que estudos mais aprofundados a respeito do JiST/SWANS devam ser realizados a fim de identificar os motivos pelos quais não se chegou a resultados precisos para as medidas apresentadas neste capítulo.



**Figura 19.** Número de pacotes de dados descartados em função do tempo de simulação obtidos do simulador NS-2 para o RWP original.

# Capítulo 6

## Conclusão e Trabalhos Futuros

Este trabalho trata da importância na escolha do Modelo de Mobilidade utilizado em simulações para avaliação de desempenho de protocolos de comunicação em redes *ad hoc*.

Apesar da popularidade do uso do modelo de mobilidade *Random Waypoint*, foi demonstrado neste trabalho que durante as simulações este modelo 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.

A fim de sanar este problema, foi proposto trocar-se a função uniforme de densidade de probabilidades da escolha da velocidade dos nós da simulação por uma distribuição baseada na função BETA( $\alpha, \beta$ ). Foi demonstrado analiticamente que a mudança proposta estabiliza o modelo de mobilidade. Quando utilizando-se a função BETA(2,2), tem-se velocidade média no estado estacionário igual a  $\frac{V_{máx}}{3}$ ; quando utilizando-se a função BETA(2,1), tem-se velocidade média no estado estacionário igual a  $\frac{V_{máx}}{2}$ .

A fim de corroborar nossas análises realizamos simulações no simulador de redes JiST/SWANS. Os resultados obtidos das simulações 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 confiáveis e deve ser refinado.

### 6.1 Contribuições

Parte deste trabalho resultou na publicação de um artigo no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) [9], o mais importante evento científico nacional em redes de computadores e sistemas distribuídos, e um

dos mais concorridos em informática, tendo sido classificado pela CAPES com o conceito A.

Além dos resultados publicados no SBRC, as contribuições deste trabalho dizem respeito, primeiro, ao modelo de mobilidade *Random Waypoint* utilizando-se a distribuição de probabilidades BETA(2,1), e depois, mas não menos importante, aos resultados obtidos das simulações no JiST/SWANS.

Quanto ao modelo de mobilidade, este trabalho demonstra, analiticamente e por meio de simulações, que o RWP estabilizado (utilizando a função BETA(2,1)) chega a um estado estacionário com velocidade média igual a  $\frac{V_{m\acute{a}x}+V_{m\acute{i}n}}{2}$  quando  $V_{m\acute{i}n}$  tende a zero.

Quanto ao simulador, foi demonstrado que é necessário avaliar-se melhor seu desempenho a fim de encontrar por qual motivo não se chegou a resultados semelhantes aos resultados obtidos quando do uso do simulador NS-2.

## 6.2 Trabalhos Futuros

A monografia aqui apresentada possui diversos desdobramentos possíveis. Tais desdobramentos têm como alvo generalizar e escalar os resultados atingidos. A análise das medidas de desempenho de redes *ad hoc* ainda pode ser bastante aprofundada.

Buscando-se generalização pode-se modelar analiticamente a velocidade instantânea média das simulações considerando distribuições BETA(k,k) e BETA(k+1,k), sendo k um inteiro qualquer, ao invés dos valores específicos de k=1 e k=2 como feito aqui.

Buscando-se escalabilidade pode-se ampliar o número de nós e a área de simulação para se ter cenários compatíveis com redes locais (centenas de usuários) e com redes metropolitanas (milhares de usuários) e verificar a validade do RWP estabilizado para essas redes.

Os experimentos realizado neste trabalho com o JiST/SWANS devem ser reproduzidos utilizando-se outros simuladores de redes, para alcançar resultados

compatíveis entre simulações e modelo analítico. Isso seria fundamental para validar os resultados deste trabalho.

# Bibliografia

- [1] CAMP, T.; BOLENG J.; DAVIES V.; A Survey of Mobility Models for *Ad Hoc* Network Research; In: Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile *Ad hoc* Networking: Research, Trends and Applications, v. 2; no. 5, pp. 483-502, 2002.
- [2] JOHANSSON, P.; LARSSON T.; HEDMAN N.; MIELCZAREK, B.; DEGERMARK, M.; Routing protocols for mobile ad-hoc networks – a comparative performance analysis; In: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 195-206, 1999.
- [3] BOLENG, J.; Normalizing mobility characteristics and enabling adaptive protocols for *ad hoc* networks; In: Proceedings of the Local and Metropolitan Area Networks Workshop (LANMAN), pp. 9-12, March 2001.
- [4] KARP, B.; Geographic Routing for Wireless Networks, 2000. PhD thesis. Harvard University.
- [5] GARCIA-LUNA-ACEVES, J.J.; SPOHN, M.; Source-tree routing in wireless networks; Proceedings of the 7<sup>th</sup> International Conference on Networks Protocols (ICNP); 1999.
- [6] YOON, J.; LIU, M.; NOBLE, B.; *Random Waypoint* Considered Harmful; Proceedings of IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 2003.
- [7] JiST/SWANS: Java in Simulation Time/Scalable Wireless *Ad hoc* Network Simulator. Disponível em: <<http://jist.ece.cornell.edu>> Acesso em 08 de maio de 2009.
- [8] PERKINS, C. E.; BELDING-ROYER, E. M.; *Ad hoc* On-Demand Distance Vector Routing; Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.
- [9] MORAES, R. M.; ARAÚJO, F. P.; PONTES, A. S. L.; Uma proposta para estabilizar o modelo de mobilidade *Random Waypoint* em redes *Ad hoc* sem fio. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2009, Recife-PE.

- [10] SSJ: Stochastic Simulation in Java. Disponível em <<http://www.iro.umontreal.ca/simandr/ssj/indexe.html>> Acesso em: 16/05/09.
- [11] GUOLONG, Lin; GUEVARA, Noubir and RAJMOHAN, Rajaraman. Mobility Models for *Ad hoc* Network Simulation. In: Proceedings of IEEE INFOCOM 2004.
- [12] FRODIGH, M., JOHANSSON, P., LARSSON, P., Wireless *Ad hoc* Networking – The Art of Networking Without a Network, Ericsson Review no. 04, 2000, Disponível em: <[http://www.ericsson.com/ericsson/corpinfo/publications/review/2000\\_04/files/2000046.pdf](http://www.ericsson.com/ericsson/corpinfo/publications/review/2000_04/files/2000046.pdf)> Acesso em: 11 de maio de 2009.
- [13] MORAES, R. M. de; SADJADPOUR, H. R. and J.J. Garcia-Luna-Aceves, "Mobility-Capacity-Delay Trade-off in Wireless *Ad hoc* Networks," *Elsevier Ad hoc Networks Journal*, vol. 4, no. 5, pp. 607-620, Netherlands (Holanda), September 2006.
- [14] DAVIES, V. Evaluating mobility models within an *ad hoc* network. Master's thesis, Colorado School of Mines, 2000.
- [15] ZENG X.; BAGRODIA, R. and GERLA, M. GLOMOSIM: A Library for Parallel Simulation of Large-scale Wireless Networks. In Workshop on Parallel and Distributed Simulation, 1998.
- [16] BARR, R.; HAAS, Z. and VAN RENESSE, R. JiST: An Efficient Approach to Simulation Using Virtual Machines. In: Software Practice and Experience, 2005, V.35(6), P.539-576.
- [17] FAN, Bai, et al. Important: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for *Ad hoc* Networks. In: IEEE INFOCOM 2003. Twenty Second Annual Joint Conference of the IEEE Computer and Communications Societies, V.3, P.825-835, 2003.
- [18] HONG, Xiaoyan, et al. A Mobility Framework for *Ad hoc* Wireless Networks. In: MDM'01: Proceedings of the Second International Conference on Mobile Data Management, P 185-196, 2001.
- [19] BROCH, J.; MALTZ, D. A.; JOHNSON, D. B.; HU, Y. C. and JETCHEVE, J. A Performance Comparison of Multi-hop Wireless *Ad hoc* Networking Routing Protocols. Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking, 1998, Dallas, TX, USA.

- [20] SWANS++ Homepage, Disponível em: <<http://www.aqualab.cs.northwestern.edu/projects/swans++>> Acesso em 16 de maio de 2009.
- [21] The Networking Simulator – NS-2, Disponível em: <<http://www.isi.edu/nsnam/ns>> Acesso em: 11/05/08
- [22] PAPOULIS, A. and PILLAI, S. U. Probability, Random Variables and Stochastic Processes. McGraw Hill, 2002.
- [23] D. B. JOHNSON and D. A. MALTZ, Dynamic Source Routing in *Ad hoc* Wireless Networks, In: Mobile Computing, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, Vol.353.
- [24] BARR, Rimon. An efficient, unifying approach to simulation using virtual machines. PhD dissertation, Cornell University, May 2004.
- [25] WEISER, M.; The Computer for the 21<sup>st</sup> Century. ACM SIGMOBILE Mobile Computing and Communications Review Special issue dedicated to Mark Weiser Pages: 3 – 11, 1999.
- [26] <http://www.dsc.upe.br/~aslp/arquivos/js-uulm.rar>.
- [27] PERKINS, D. D.; Hughes, H. D. and Owen, C.B. (2002). Factors affecting the performance of ad hoc networks. Proc. Of IEEE International Conference on Communications (ICC), New York, NY, USA.
- [28] JAYAKUMAR, G. and Gopinath, G. Performance comparison of two on-demand routing protocols for ad-hoc networks based on random waypoint mobility model. American Journal of Applied Sciences, 5(6):649-664, 2008.

# Apêndice A

## Problemas Encontrados no JiST/SWANS Durante Este Trabalho

### 1. Mac802\_11 renomeado para Mac802

Por algum motivo não identificado, os métodos da classe org.apache.log4j.Logger não estavam sendo corretamente executados por incompatibilidade com o nome da classe Mac802\_11, que possui o símbolo '\_' e causa confusão com o formato do script "swans.properties". Por isso a classe Mac802\_11 foi renomeada para Mac802.

### 2. Bug in Route.Aadv.generateRouteReplyMessage()

O método routeTable.lookup(this.netAddr) estava retornando uma referência nula, o que gerava erro ao se chamar o método setDestSeqNum(this.seqNum);

Código com problema:

```
private void generateRouteReplyMessage(RouteRequestMessage rreqMsg,
boolean isDest, RouteTableEntry destRouteEntry)
{
    ...

    if (!rreqMsg.getUnknownDestSeqNum() && rreqMsg.getDestSeqNum() >
this.seqNum)
    {
        this.seqNum = rreqMsg.getDestSeqNum(); // update SN

        //update route-to-self with updated SN
        RouteTableEntry selfRoute = routeTable.lookup(this.netAddr);
        selfRoute.setDestSeqNum(this.seqNum);
        routeTable.printTable();
    }

    ...
}
```

O código corrigido é:

```
private void generateRouteReplyMessage(RouteRequestMessage rreqMsg,
boolean isDest, RouteTableEntry destRouteEntry)
{
    ...

    if (!rreqMsg.getUnknownDestSeqNum() && rreqMsg.getDestSeqNum() >
this.seqNum)
    {
        this.seqNum = rreqMsg.getDestSeqNum(); // update SN

        //update route-to-self with updated SN
RouteTableEntry selfRoute = routeTable.lookup(this.netAddr);
//selfRoute.setDestSeqNum(this.seqNum);
if(selfRoute==null)
    {
        this.routeTable.add(this.netAddr,
            new RouteTableEntry(MacAddress.NULL, this.seqNum, 0));
    }
    else
    {
        selfRoute.setDestSeqNum(this.seqNum);
    }
    routeTable.printTable();
}

    ...
}
```

# Anexo A

## Problemas Encontrados no JiST/SWANS Pela Comunidade

### 1. Bug em `jist.swans.Mac802.java`

#### 1.1 Eliminação de Mensagens Duplicadas

O problema foi gerado devido a um tratamento errado na recepção de uma mensagem da camada MAC que não deveria ser repassada para a camada superior. Como resultado, a mesma mensagem poderia ser entregues duas vezes , apesar do mecanismo de eliminação de mensagens duplicadas do 802.11.

Esta alteração garante que se uma mensagem é retransmitida (porque o Ack da primeira transmissão não tenha sido recebida), a mensagem mantém seu número seqüencial original. Isto significa que se o nó destino ouviu a mensagem original e a mensagem 'ack' se perdeu devido a alguns problemas de colisão, quando o destino re-enviar o que ele sabe que já tem, o número sequencial será o mesmo.

O código defeituoso e o corrigido é originado no método `sendDataUnicast()`, e são, respectivamente, encontrados a seguir.

```

public void sendDataUnicast(boolean afterCts)
{
    MacMessage.Data data = new MacMessage.Data(
        packetNextHop, localAddr, (int)(
            MacMessage.Ack.SIZE*Constants.SECOND/bandwidth
            + (SYNCHRONIZATION + PROPAGATION + SIFS)
            + PROPAGATION),
        incSeq(), (byte)0, false,
        (shouldRTS() ? longRetry : shortRetry) > 0,
        packet);
    setMode(MAC_MODE_XUNICAST);
    long delay = afterCts ? SIFS : RX_TX_TURNAROUND, duration =
    transmitTime(data);
    radioEntity.transmit(data, delay, duration);

    JistAPI.sleep(delay+duration);
    self.startTimer(MacMessage.Ack.SIZE*Constants.SECOND/bandwidth
        + (SYNCHRONIZATION + PROPAGATION + SIFS)
        + SLOT_TIME
        + PROPAGATION,
        MAC_MODE_SWFACK);
}

```

```

public void sendDataUnicast(boolean afterCts)
{
    //Changed by Noam Mori
    short augmentedSeq = (shortRetry+longRetry)==0?incSeq():seq;
    // create data packet
    MacMessage.Data data = new MacMessage.Data(
        packetNextHop, localAddr, (int)(
            MacMessage.Ack.SIZE*Constants.SECOND/bandwidth
            + (SYNCHRONIZATION + PROPAGATION + SIFS)
            + PROPAGATION),
        augmentedSeq, (byte)0, false,
        (shouldRTS() ? longRetry : shortRetry) > 0,
        packet);
    setMode(MAC_MODE_XUNICAST);
    long delay = afterCts ? SIFS : RX_TX_TURNAROUND, duration =
    transmitTime(data);
    radioEntity.transmit(data, delay, duration);

    JistAPI.sleep(delay+duration);
    self.startTimer(MacMessage.Ack.SIZE*Constants.SECOND/bandwidth
        + (SYNCHRONIZATION + PROPAGATION + SIFS)
        + SLOT_TIME
        + PROPAGATION,
        MAC_MODE_SWFACK);
}

```

## 2. Bug em Location.Location2D

Os sintomas desse problema é que depois de executar a simulação por um tempo, alguns nós não recebem algumas das mensagens enviadas a eles, mesmo por um vizinho próximo fisicamente e, mesmo quando não haja interferência.

Esse *bug* é originado no seguinte método:

```
public boolean inside(Location min, Location max)
{
    Location2D min2d = (Location2D)min, max2d = (Location2D)max;
    if(Main.ASSERT) Util.assertion(min2d.x<=max2d.x &&
min2d.y<=max2d.y);
    return x<=max2d.x && y<=max2d.y && x>=min2d.x && y>=min2d.y;
}
```

O código corrigido está destacado em vermelho:

```
public boolean inside(Location min, Location max)
{
    Location2D min2d = (Location2D)min, max2d = (Location2D)max;
    if(Main.ASSERT) Util.assertion(min2d.x<=max2d.x &&
min2d.y<=max2d.y);
    return x<max2d.x && y<max2d.y && x>=min2d.x && y>=min2d.y;
    //Changed by Noam Mori (= removed)
}
```

Da mesma forma deve ser mudado no método

```
public boolean inside(Location bounds)
{
    Location2D l2d = (Location2D)bounds;
    return x<l2d.x && y<l2d.y && x>=0 && y>=0; //Changed by Noam
Mori (= removed)
}
```

### **3. Modelo de Interferência RadioNoiseIndep**

RadioNoiseIndep é o modelo de interferência padrão utilizado no SWANS. Ele foi desenvolvido para imitar o pobre modelo de interferência que foi utilizado no NS-2 em versões passadas deste simulador. O modelo RadioNoiseIndep foi muito otimista e resulta em um número muito pequeno de colisões. Ao invés de tentar corrigir esse modelo foi utilizado o modelo de interferência RadioNoiseAdditive.

O RadioNoiseAdditive é um modelo de interferência mais refinado e correto. É similar ao modelo de interferência atualmente usado no GloMoSim e no NS-2 versão 2.33. Esse modelo inclui acumulação da relação sinal ruído, ao contrário do ruído independente do RadioNoiseIndep.