

COPPE



Instituto Alberto Luiz Coimbra  
de Pós-Graduação  
e Pesquisa de Engenharia  
Universidade Federal  
do Rio de Janeiro

Programa de Engenharia de Sistema e Computação

## Mobilidade em Redes Sem Fio *Ad hoc*

Carlos Alberto Vieira Campos

Alexandre Mendes da Silva

Daniel Cavas Otero

Luciano Renovato de Albuquerque

Luís Felipe Magalhães de Moraes

{beto, alexandre, daniel, aluciano, moraes}@ravel.ufrj.br

Laboratório de Redes de Alta Velocidade - RAVEL

<http://www.ravel.ufrj.br>

Outubro de 2003

# Conteúdo

<b>1</b>	<b>Redes Sem Fio</b>	<b>2</b>
1.1	Introdução . . . . .	2
1.2	Redes Móveis <i>Ad hoc</i> . . . . .	2
1.2.1	Roteamento . . . . .	4
1.2.2	Segurança . . . . .	5
1.3	Noções de Probabilidade e Estatística . . . . .	6
<b>2</b>	<b>Modelos de Mobilidade</b>	<b>11</b>
2.1	Modelos de Mobilidade para Redes <i>Ad hoc</i> . . . . .	12
2.1.1	Modelos de Mobilidade Individual . . . . .	12
2.1.2	Modelos de Mobilidade em Grupo . . . . .	17
2.2	Regras de Borda para Simulação de Áreas Limitadas . . . . .	19
2.2.1	<i>Bounce</i> . . . . .	19
2.2.2	<i>Delete and Replace</i> . . . . .	19
2.2.3	<i>Wrap around</i> . . . . .	20
2.3	Métricas de Avaliação de Modelos de Mobilidade . . . . .	20
<b>3</b>	<b>Ferramentas de Simulação</b>	<b>22</b>
3.1	Simuladores de Redes . . . . .	22
3.1.1	<i>Network Simulator</i> – NS-2 . . . . .	22
3.1.2	O Simulador GlomoSim . . . . .	25
3.1.3	Diferenças entre o GloMoSim e o NS-2 . . . . .	28
3.2	Geradores de Cenários de Mobilidade . . . . .	29
3.2.1	<i>Scenario Generator</i> – ScenGen . . . . .	29
3.2.2	<i>BonnMotion</i> . . . . .	33
<b>4</b>	<b>Pesquisas Atuais</b>	<b>35</b>
<b>5</b>	<b>Considerações Finais</b>	<b>37</b>

# Capítulo 1

## Redes Sem Fio

### 1.1 Introdução

As redes de comunicação sem fio vêm adquirindo importância cada vez maior nos últimos anos, fato este que pode ser observado pelo aumento da demanda por este tipo de tecnologia e sua utilização cada vez mais freqüente.

Diversos grupos de trabalho vem sendo formados com o objetivo de estudar áreas e problemas específicos, e propor soluções que se tornem padrões no mercado. Como exemplo de esforços nesse sentido, o IEEE (*Institute of Electrical and Electronics Engineers*) [1] formou diversos grupos de trabalho para a elaboração de novos padrões como extensões ao atual padrão 802.11, tais como o grupo 802.11E, preocupado com aspectos de QoS, o 802.11N, que visa aumentar a vazão destas redes, o 802.11I, que propõe mecanismos de segurança, e o 802.11F, que estuda mecanismos de *handoff*.

Dentre as redes sem fio, muitos estudos têm sido feitos em torno de redes móveis *ad hoc*. Estas redes não necessitam da disponibilidade de uma infra-estrutura, gerando maior flexibilidade e acrescentando novas aplicações à tecnologia sem fio. No entanto, nestas redes, a mobilidade dos usuários acrescenta problemas que devem ser tratados, devido principalmente ao dinamismo da topologia de rede existente. Além disso, o desempenho de soluções propostas para redes *ad hoc* é afetado diretamente pela mobilidade dos usuários, fazendo com que esta deva ser levada em consideração em qualquer avaliação de protocolos e soluções propostas. Acrescenta-se ainda que uma mesma aplicação ou protocolo pode apresentar desempenho bem diferente quando o comportamento de movimentação dos usuários muda.

Este curso estará focando aspectos de mobilidade em redes *ad hoc*, apresentando os modelos de mobilidade mais utilizados, características de configuração dos cenários de simulação, ferramentas de simulação e pesquisas recentes na área. Alguns conceitos de probabilidade e estatística necessários para o entendimento dos modelos e das simulações também serão abordados.

### 1.2 Redes Móveis *Ad hoc*

As redes móveis *ad hoc* são tecnologias de comunicação sem fio, onde os dispositivos computacionais móveis são capazes de trocar informações diretamente entre si ou através de múltiplos

saltos sem a necessidade de uma infra-estrutura de comunicação [2, 3, 4]. Assim sendo, os dispositivos computacionais possuem a capacidade de ser tanto um roteador quanto um dispositivo de comunicação terminal, ou ainda ambos simultaneamente, dando à rede a característica de auto-configuração dinâmica.

Com essas características, as redes *ad hoc* são indicadas para situações onde não se pode, ou não é adequado instalar uma rede fixa ou uma rede sem fio infra-estruturada como, por exemplo, uma rede de telefonia celular. Por este motivo estão surgindo tecnologias como o padrão *bluetooth* e redes de sensores, que são extensões das redes *ad hoc* [3]. Nesse trabalho, os dispositivos computacionais móveis serão denominados de nós móveis - NMs.

Como cenários de utilização de redes *ad hoc* pode-se citar:

- Campos de batalhas, onde, aviões, helicópteros, tanques de guerra e militares trocam informações táticas;
- Empresários compartilhando informações numa reunião;
- Vendedores comunicando-se no centro comercial de uma cidade;
- Equipes de resgate e salvamento numa situação de desastre, como furacão, terremoto ou inundação, onde estas precisam se coordenar;
- Estudantes usando computadores portáteis para participar de uma aula interativa.

Com o objetivo de estudo e padronização desse tipo de rede, o IETF (*Internet Engineering Task Force*) criou um grupo de trabalho dentro da sub-área de roteamento, denominado *Mobile Ad hoc Networks* - MANET [5]. Esse grupo de trabalho já publicou três *Request-for-Comments* - RFCs: uma sobre a visão geral das redes *ad hoc*, RFC 2501 [2] e, recentemente, outras duas sobre os algoritmos de roteamento: *Ad Hoc On-Demand Distance Vector Routing* - AODV, RFC 3561 [6] e *Optimized Link State Routing Protocol* - OLSR, RFC 3626. Além das RFCs, diversos *drafts* vêm sendo propostos pelo grupo MANET.

Dentro desse contexto, são descritas as principais características de uma rede *ad hoc* [2, 7]:

- Topologia dinâmica: os NMs podem se locomover de forma arbitrária, fazendo com que a topologia da rede seja dinâmica, ou seja, a topologia muda sua configuração frequentemente. Além disso, os enlaces podem ser unidirecionais ou bidirecionais;

- Largura limitada de banda: os enlaces sem fio continuam tendo menor capacidade do que enlaces das redes fixas;

- Segurança física limitada: as redes sem fio são mais vulneráveis às ameaças do que as redes fixas, tendo uma maior possibilidade de escuta e invasão.

Como vantagens podem ser citadas:

- Rápida instalação: podem ser instaladas rapidamente em locais sem nenhuma infra-estrutura prévia;

- Tolerância à falhas: o mal funcionamento ou o desligamento de um dispositivo pode ser facilmente sanado com a reconfiguração dinâmica da rede. Em uma rede fixa, ao contrário, quando ocorre uma falha em um roteador, o redirecionamento de tráfego é uma operação complexa, quando possível. Falhas em redes estruturadas são mais graves ainda se ocorrerem no ponto de acesso dos dispositivos sem fio, pois todos os nós dependentes deste ficam sem comunicação;

- Conectividade: se duas estações estão dentro da área de alcance das ondas de rádio, elas têm um canal de comunicação direto. Em uma rede fixa, mesmo que duas estações estejam uma ao lado da outra é necessário que as estações estejam ligadas por um meio guiado para que troquem informações. Em uma rede estruturada é necessária a comunicação do dispositivo móvel com o ponto de acesso e deste para o outro dispositivo móvel;

- Mobilidade: os dispositivos móveis podem movimentar-se durante o período de conexão na rede.

Como desvantagens podem ser citadas [7]:

- Pequena banda passante: canais de comunicação sem fio geralmente possuem menor banda passante que enlaces de redes com fios;

- Erros no enlace sem fio: a taxa de erros do enlace sem fio é tipicamente de um bit errado a cada  $10^5$  ou  $10^6$  bits transmitidos, enquanto que em uma fibra ótica esta taxa é tipicamente de um a cada  $10^{12}$  ou  $10^{15}$  bits transmitidos;

- Localização: existe a dificuldade de identificar a posição física do NM e enviar, para esse ponto, suas mensagens. Em redes fixas não existe essa dificuldade, pois o endereço IP indica implicitamente a localização do nó. Para redes infra-estruturadas esse problema foi resolvido através do protocolo IP-Móvel, que se encarrega de localizar o nó. Já em redes *ad hoc*, localizar o NM é um problema, pois não se tem nenhuma informação geográfica e o endereço da máquina não tem necessariamente nenhuma relação com sua posição;

- Roteamento: em uma rede fixa a topologia dificilmente se altera, os nós ficam geralmente nas mesmas posições da rede. Em uma MANET os nós podem se mover de uma região para outra de forma não determinística, fazendo com que uma rota válida entre dois nós torna-se uma rota inválida. Neste caso é necessário encontrar, de alguma forma, uma outra rota válida entre os nós em questão. Com isso, a mudança na topologia afeta o desempenho da rede.

Dessa forma, as redes móveis *ad hoc* possuem características específicas, onde a mobilidade influencia o funcionamento de todos os seus componentes, protocolos e aplicações. Além disso, o roteamento e a segurança são tarefas complexas.

### 1.2.1 Roteamento

Uma rota entre dois NMs, numa rede *ad hoc*, pode ser formada por um ou vários saltos através dos NMs na rede. Um dos principais problemas em redes *ad hoc* é determinar e manter as rotas, já que a mobilidade de um NM pode causar mudanças na topologia. Por isso vários protocolos e algoritmos de roteamento foram e estão sendo propostos e desenvolvidos para realizar esta tarefa de maneira mais eficiente.

É importante que esses protocolos, ao realizarem a manutenção das tabelas de roteamento em cada nó, gerem o mínimo de sobrecarga (*overhead*) possível e que a quantidade de banda consumida por eles também seja pequena. As atualizações das tabelas de roteamento podem ser realizadas de maneira isolada, distribuída, centralizada ou hierárquica.

Os protocolos de roteamento para redes *ad hoc* são classificados em *unicast* e *multicast* [3], [8], [9]. No grupo de protocolos *unicast* temos os pró-ativos (DSDV [10], WRP [3]), os reativos (AODV [11], DSR [12], TORA [13]), os híbridos que são pró-ativos dentro de uma zona de roteamento e reativo entre essas zonas (ZRP [14]) e temos também os com informações de localização (DREAM

[15], LAR [16]). Já o grupo de protocolos *multicast* são divididos em dois sub-grupos [4], [17]: baseado em árvores e baseado em uma malha (*mesh*). Os protocolos baseados em árvores, que são: MAODV, AMRoute, AMRIS, usam uma árvore compartilhada para entregar os pacotes à um grupo de NMs. Já os protocolos baseados em malhas, que são: ODMRP e CAMP, usam um conjunto de NMs ou uma malha para encaminhar os pacotes. Existem vários protocolos de roteamento para redes *ad hoc*, além desses aqui citados, tanto para comunicação *unicast*, quanto para *multicast*.

O protocolos pró-ativos são assim classificados pelo fato de manterem informações sobre rotas para todos os nós da rede, mesmo que o nó onde o protocolo está sendo executado nunca tenha utilizado muitas dessas rotas, tanto para enviar seus próprios pacotes como para enviar pacotes de outros nós, fazendo papel de roteador. São usadas mensagens periódicas, trocadas entre todos os nós da rede, para manter a tabela de rotas de cada um constantemente atualizada. Normalmente esse tipo de protocolo consegue ter um melhor desempenho do que os protocolos reativos, sendo mais veloz no tempo de resposta para o nó origem que solicitou uma determinada rota, dado que todas as rotas possíveis devem existir na tabela de roteamento de cada nó. O exemplo mais citado desse tipo de protocolo é o *Destination-Sequenced Distance Vector - DSDV* [10].

Já os protocolos reativos realizam o estabelecimento de uma rota apenas quando ela é solicitada pelo nó origem. Um processo de descoberta de rota é iniciado quando um determinado destino deve ser alcançado, e não existe rota estabelecida para o mesmo. As rotas são mantidas na tabela de roteamento até que elas deixem de existir, ou após um determinado tempo sem que elas tenham sido utilizadas. Como o processo de descoberta de rota é realizado apenas quando uma origem qualquer faz uma requisição, esse tipo de protocolo normalmente gera menos sobrecarga do que os pró-ativos, em detrimento do tempo de espera para ter a rota solicitada estabelecida. Os protocolos *Ad hoc On Demand Distance Vector- AODV* [11] e o *Dynamic Source Routing - DSR* [12] são exemplos de protocolos reativos.

Para avaliar o desempenho de um protocolo de roteamento [4, 18] são necessárias várias análises em relação ao alcance de transmissão dos NMs, características do tráfego, tamanho dos *buffers* para o armazenamento das mensagens e padrões de movimentação dos usuários da rede. Nesse curso, será abordado com um nível maior de detalhamento o último tópico de avaliação que são os padrões de movimentação dos usuários.

Após terem sido apresentadas brevemente as possíveis técnicas para desenvolvimento de protocolos de roteamento, serão mencionadas algumas das possíveis técnicas utilizadas para realizar ataques em redes móveis *ad hoc*. Todas elas têm como principal alvo as falhas e fraquezas dos protocolos de roteamento, lembrando que inicialmente esses protocolos foram desenvolvidos sem qualquer preocupação quanto a segurança.

## 1.2.2 Segurança

A própria natureza de uma rede *ad hoc* a faz insegura. O grau de comprometimento entre seus membros é alto, já que todos dependem uns dos outros para o pleno funcionamento da rede, e a qualidade conseguida depende do trabalho de cada nó. Com isto, é possível perceber que o mau funcionamento de um único nó pode trazer grandes prejuízos para toda a rede.

Os protocolos de roteamento desenvolvidos inicialmente não se preocupavam com os aspectos de segurança. Dessa maneira, as vulnerabilidades intrínsecas de uma rede *ad hoc*, devidas ao alto

grau de dependência entre seus membros, tornaram-se falhas de segurança para os protocolos de roteamento.

Sendo assim, percebe-se que a forma como uma rede desse tipo deve ser protegida não será a mesma adotada em redes cabeadas. Cada um de seus membros deverá estar preparado para enfrentar um adversário, garantindo indiretamente maior grau de segurança para toda a rede. É sabido que em redes de outros tipos, onde o meio físico compartilhado não é o ar, a segurança total da rede depende, também, das ações preventivas tomadas por cada membro, porém em redes *ad hoc* essas ações têm um significado ainda mais forte.

Nós inimigos ou comprometidos podem participar do processo de descoberta de rotas e aproveitarem-se disso. Os pacotes de pedido de rota (*route request* - RREQ) e resposta de rota (*route reply* - RREP) podem ser alterados enquanto trafegam, ou podem ser forjados causando diversas anomalias no funcionamento da rede [19, 20, 21, 22].

Os pacotes usados pelos protocolos de roteamento quando sujeitos a ações como as citadas no parágrafo anterior podem causar:

- Rotas com loops
- *Timeouts* demorados
- Métricas falsas ou exageradas
- Repetição de *updates* antigos/desatualizados

O que leva à negação de serviços (DoS).

Baseado no funcionamento de alguns protocolos de roteamento, os seguintes ataques podem ser realizados, explorando suas falhas:

- Alteração do campo *destination sequence numbers*;
- Alteração da quantidade de saltos;
- Mensagens de erro forjadas;
- Envenenando tabelas de rotas.

## 1.3 Noções de Probabilidade e Estatística

Nesta seção serão apresentados os principais conceitos estatísticos utilizados em simulações.

### Variáveis Aleatórias

Uma variável aleatória é definida como uma função, ou regra, que associa um número real a cada ponto do espaço amostral de um experimento aleatório, ou seja, são variáveis cujo valor depende do resultado de um experimento aleatório [23].

Como exemplo é possível citar a velocidade de um nó móvel, escolhida ao acaso em determinados pontos da simulação (ou ainda sua própria velocidade real, caso esta esteja sendo medida), o valor obtido a partir de um lançamento de um dado, ou então uma regra que associe à variável o valor 0

se o resultado obtido no dado for menor ou igual a 3, ou 1 se o resultado obtido for maior ou igual a 4.

As variáveis aleatórias podem ser discretas, quando o conjunto de valores possíveis é enumerável, ou contínua, quando isto não ocorre.

## Distribuição de Probabilidade

A função distribuição, ou CDF (*cumulative distribution function*), representa a probabilidade da variável aleatória assumir um valor menor ou igual a determinado valor  $x$ , ou seja:

$$F_X(x) = P(X \leq x) = P(X \in (-\infty, x])$$

A função distribuição é definida de duas formas, uma para variáveis aleatórias contínuas e outra para variáveis aleatórias discretas, no entanto, ambas representam a probabilidade descrita acima. Para variáveis contínuas é definida como:

$$F_X(x) = \int_{-\infty}^x f_X(u) du$$

Onde  $f_X(u)$  é a função densidade, definida mais abaixo. Para variáveis discretas a função distribuição é definida como:

$$F_X(x) = \sum_{k \leq x} P(X = k) = \sum_{k=0}^{\infty} p_k u(x - x_k)$$

Onde  $p_k = P(X = x_k)$ , e  $u(x) = 1$  se  $x \geq 0$  ou  $0$  se  $x < 0$ .

A derivada da função distribuição de variáveis aleatórias contínuas é chamada de função densidade, ou PDF (*probability density function*), e através dela é possível determinar a probabilidade da variável  $X$  assumir um valor em determinado intervalo, ou seja, dado um intervalo  $\Delta x$ :

$$P(X \in [x, x + \Delta x]) = \int_x^{x+\Delta x} f_X(u) du$$

É interessante notar que para variáveis contínuas a probabilidade da variável assumir um valor específico ( $P(X = x)$ ) é igual a zero. Para variáveis discretas não é possível definir a função densidade como a derivada da função distribuição devido a esta apresentar pontos de descontinuidade, então define-se a PMF (*probability mass function*), que representa a probabilidade da variável assumir um determinado valor, ou seja,  $P(X = x)$ .

## Média e Variância

A média, ou valor esperado de uma variável aleatória é definido para variáveis discretas como:

$$\sum_{j=0}^{\infty} x_j p(x_j)$$

E para variáveis contínuas como:



$$\int_{-\infty}^{+\infty} xf(x)dx$$

Onde  $f(x)$  é a função densidade. A média é um parâmetro muito utilizado para obter estimativas do comportamento de um sistema, assim como para avaliações de performance e comparação de alternativas.

A variância e o desvio padrão complementam a média, informando a dispersão dos dados em relação a esta. Como um exemplo simples, os conjuntos  $A=\{10, 50, 90\}$  e  $B=\{49, 50, 51\}$  possuem a mesma média, porém os valores estão bem menos dispersos em B. A variância é definida como:

$$\sigma^2 = E[(X - \mu)^2] = E(X^2) - \mu^2$$

E o desvio padrão, denominado  $\sigma$ , é a raiz quadrada da variância, ou seja,  $\sigma = \sqrt{\sigma^2}$ .

### Transformadas

As transformadas são muito utilizadas como funções geradoras de momento, ou seja, funções a partir das quais é possível gerar qualquer momento da variável aleatória ( $\bar{X}, \bar{X}^2, \bar{X}^3$ , etc.). Para variáveis contínuas usualmente é utilizada a transformada Laplace, definida como:

$$F^*(s) = E[e^{-sX}] = \int_{-\infty}^{+\infty} e^{-sx} f(x)dx$$

E para variáveis discretas, utiliza-se a transformada Z:

$$G(z) = E[z^X] = \sum_k z^k p_k$$

Onde  $p_k = p(x_k) = P(X = x_k)$ . Através da transformada Laplace de uma variável contínua o n-ésimo momento desta variável é obtido da seguinte forma:

$$F^{*(n)}(0) = (-1)^n \bar{X}^n$$

E para variáveis discretas,  $G^1(1) = \bar{X}$ , e  $G^2(1) = \bar{X}^2 - \bar{X}$ .

### Amostragem e Intervalo de Confiança

Em situações práticas os parâmetros utilizados para a avaliação ou comparação do comportamento do sistema sendo estudado são desconhecidos, o que cria a necessidade da utilização de estimativas para os mesmos [24]. Esta estimativa é realizada através da amostragem, onde são coletadas  $n$  observações do parâmetro em questão,  $X_1, X_2, \dots, X_n$ . Com base nestes dados a média amostral (ou aritmética) é calculada da seguinte forma:

$$\bar{X}(n) = \frac{1}{n} \sum_{j=1}^n X_j$$

Observa-se que esta média amostral é uma estimativa de  $\mu$ , que é o valor esperado da variável

aleatória. Da mesma forma, a variância amostral, que é uma estimativa de  $\sigma^2$ , é calculada da seguinte forma:

$$S^2(n) = \frac{\sum_{i=1}^n [X_i - \bar{X}(n)]^2}{n - 1}$$

Pela lei forte dos grandes números, sabe-se que para um número infinito de observações, a média amostral (aritmética) iguala-se ao valor esperado. Porém, como na prática o número de observações é finito, é necessária a obtenção de limites probabilísticos que determinam o quanto este valor estimado está próximo do valor esperado da variável aleatória.

Este limite probabilístico é obtido através do teorema do limite central, segundo o qual para um grande número de observações, a média amostral tem distribuição normal com média  $\mu$  e variância  $\sigma^2/n$ . Com base neste teorema, determinam-se os limites  $c_1$  e  $c_2$ , onde  $P[c_1 \leq \mu \leq c_2] = 1 - \alpha$ . Neste limite, o nível de confiança é de  $100(1 - \alpha)\%$ , e o intervalo de confiança é  $[c_1, c_2]$ .

Para um número de observações suficientemente grande, o intervalo de confiança para algum nível desejado é obtido por:

$$\bar{X}(n) \pm z_{1-\alpha/2} \sqrt{\frac{S^2(n)}{n}}$$

No entanto, existe o problema de determinar um valor de  $n$  suficientemente grande para que este intervalo possa ser obtido. Alguns autores recomendam a utilização desta aproximação com  $n \geq 30$ , no entanto, constata-se que quanto mais assimétrica for a distribuição da variável, maior será o valor de  $n$  necessário. Uma definição alternativa que calcula o intervalo de confiança em função de  $n$  e válido para um número menor de observações é dada por:

$$\bar{X}(n) \pm t_{n-1, 1-\alpha/2} \sqrt{\frac{S^2(n)}{n}}$$

Onde os valores de  $t_{n-1, 1-\alpha/2}$  são tabelados, em função de  $n$  e  $\alpha$ . No entanto, quanto maior o número de observações maior será a aproximação de  $\mu$  obtida.

## Cadeias de Markov

Cadeias de Markov são formadas através da representação de um conjunto de estados que um sistema pode assumir e da especificação de alguma transição entre eles. Podem ser de parâmetro discreto ou contínuo.

No caso de cadeias com parâmetro discreto, como visto na figura 1.1, são representados os estados e as probabilidades de transição entre os mesmos. Nestas cadeias os instantes de observação do sistema são pré-definidos, e através da matriz de transição  $P$  são definidas todas as probabilidades de transição entre estados em dois instantes de observação do sistema consecutivos.

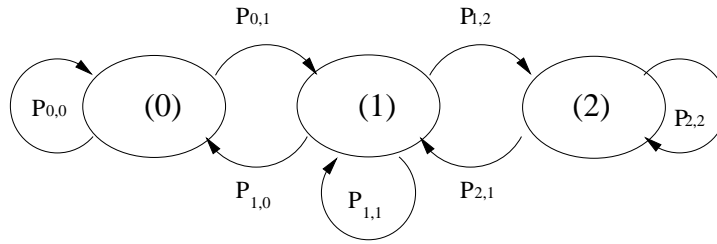


Figura 1.1: Cadeia de Markov com parâmetro discreto

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} \\ P_{1,0} & P_{1,1} & P_{1,2} \\ P_{2,0} & P_{2,1} & P_{2,2} \end{bmatrix}$$

Já para cadeias com parâmetro contínuo, como visto na figura 1.2, são utilizadas taxas de transição entre os estados, representadas na matriz  $\Lambda$ . Análises elaboradas de cadeias de Markov podem ser encontradas na literatura. Em suma, estas cadeias representam processos de Markov, onde apenas o presente influencia o futuro, sem que o passado seja levado em consideração.

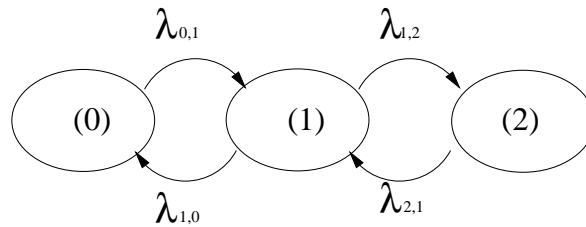


Figura 1.2: Cadeia de Markov com parâmetro contínuo

$$\Lambda = \begin{bmatrix} 0 & \lambda_{0,1} & \lambda_{0,2} \\ \lambda_{1,0} & 0 & \lambda_{1,2} \\ \lambda_{2,0} & \lambda_{2,1} & 0 \end{bmatrix}$$

## Capítulo 2

# Modelos de Mobilidade

Atualmente existem duas maneiras de se representar padrões de movimento de usuários de uma rede móvel [4]. Uma forma é através da captura de informações do comportamento real de movimentação do NM, ou seja, com o uso de registros de movimentação (*traces*). A outra maneira é através da utilização de modelos de mobilidade, onde tenta-se representar o comportamento de movimentação dos nós sem o uso desses registros de movimentação.

A captura dos registros de movimentação possibilita uma observação do comportamento real de movimentação dos NMs, principalmente quando se tem um grande número de nós durante um longo período de observação. Entretanto, em ambientes muito dinâmicos, como em redes *ad hoc*, capturar esses registros não é uma tarefa fácil e quando se trata de dados confidenciais há uma dificuldade maior ainda de acesso aos mesmos.

Além disso, as redes *ad hoc* ainda estão em fase de pesquisa e desenvolvimento, o que dificulta a obtenção de *traces* de cenários reais. Com isso é necessário a utilização de modelos de mobilidade para representar a movimentação dos NMs. Esses modelos possuem duas categorias [4, 18]: modelos para redes de telefonia celular e modelos para redes móveis *ad hoc*.

Dentro do contexto da modelagem da mobilidade, Bettstetter apresenta em [25] um *framework* da representação da mobilidade usada na simulação e análise de sistemas de comunicações sem fio. Como pode ser visto na figura 2.1, a representação da mobilidade é realizada através dos componentes: nível de aleatoriedade, nível de detalhamento, representação por simulação ou por modelagem, quantidade de dimensões e regras de borda. Essa representação pode ser aplicada tanto em redes sem fio infra-estruturadas (rede celular), quanto em redes sem infra-estrutura (rede *ad hoc*).

A modelagem da mobilidade dos usuários de uma rede celular já foi bastante estudada e desenvolvida na literatura [25, 26], apesar de ainda ser estudada e novas propostas estarem surgindo. Nessas redes, a mobilidade dos usuários influencia diversos aspectos relacionados ao tráfego de dados e à carga de sinalização na rede. Além disso, surge a necessidade de suporte à mobilidade dos usuários, onde é necessário gerenciar o deslocamento através de mecanismos, como: registro e localização do usuário, atualização de localização, estratégias de *paging* e *handoff*. Para o funcionamento eficiente desses procedimentos é necessário uma adequação entre esses e o comportamento de movimentação dos usuários.

Diversos modelos de mobilidade para redes de telefonia celular foram propostos na literatura

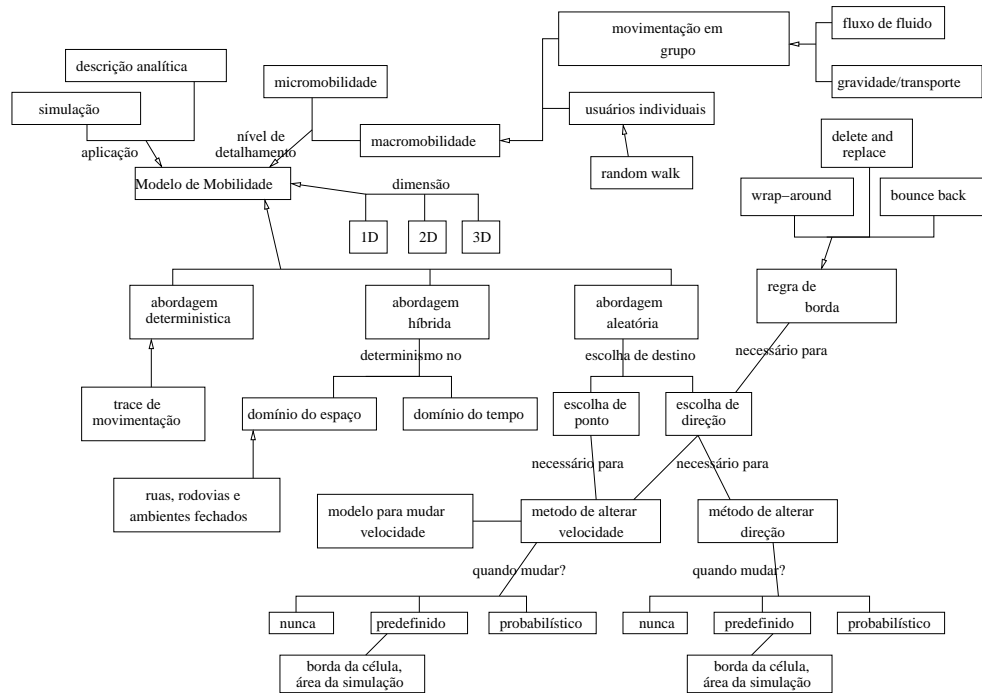


Figura 2.1: *Framework* da representação da mobilidade em comunicações sem fio[25].

[25, 26]. Como áreas de aplicação desses modelos em uma rede celular, têm-se: gerenciamento de localização, estratégias de *paging*, gerenciamento dos recursos de rádio (esquemas de alocação de canal, técnicas de múltiplos acessos ao meio), propagação e rádio-transmissão (interferência co-canal e gerenciamento de *handoff*).

## 2.1 Modelos de Mobilidade para Redes *Ad hoc*

Como foi dito anteriormente, os modelos de mobilidade buscam representar o comportamento de movimentação dos NMs numa rede móvel *ad hoc*. Esses modelos são usados na simulação dessas redes para a avaliação do desempenho de aplicações e sistemas de comunicação, permitindo analisar o impacto causado pela mobilidade no funcionamento dos mesmos. Têm-se como exemplos de ambientes onde se aplicam os modelos de mobilidade: aplicações de gerenciamento de distribuição de chave criptográfica, suporte à descoberta de serviços, suporte ao gerenciamento do tráfego suportado pela rede, avaliação da perda de pacotes em redes *ad hoc* [27], avaliação da capacidade, avaliação dos protocolos de roteamento [4, 8], predição do particionamento das redes *ad hoc* [28], cobertura de serviços em redes *ad hoc* particionáveis [29] e em protocolos de acesso ao meio para redes *ad hoc*.

Esses modelos são classificados em dois tipos: modelos de mobilidade individual e modelos de mobilidade em grupo.

### 2.1.1 Modelos de Mobilidade Individual

São modelos que representam o comportamento de movimentação de um NM de forma in-

dependente do restante dos NMs da rede. Consiste numa modelagem mais simples e de fácil implementação. Devido a tais características, esses modelos são os mais usados na literatura para avaliação de sistemas para redes *ad hoc*[4].

Os modelos de mobilidade individual tem importante aplicação na predição da disponibilidade dos enlaces sem fio, com o objetivo de melhorar, por exemplo, a eficiência dos algoritmos de roteamento e construção de rotas mais estáveis. A causa de mudanças na disponibilidade dos enlaces sem fio são as mudanças locais na topologia da rede.

### **Modelo de Mobilidade Aleatória**

O modelo de mobilidade aleatória é o modelo de mobilidade de percurso aleatório (*Random Walk Mobility Model*) [9, 26] para as redes de telefonia celular e que também é aplicado em redes *ad hoc*. Ainda hoje, é um dos modelos mais utilizados pelos pesquisadores para avaliação de protocolos em redes *ad hoc* [18]. Ele tem a seguinte característica: 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.

Algumas modificações para esse modelo têm sido propostas. Por exemplo, em [15], os NMs possuem uma direção aleatória, mas uma velocidade constante durante todo o tempo de simulação. Em [16], os NMs percorrem um caminho que é formado de vários segmentos. O tamanho desses segmentos segue uma distribuição exponencial, a direção de cada segmento é escolhida de forma aleatória e a velocidade é uniformemente distribuída entre  $[1, n]$ .

### **Modelo de Mobilidade *Waypoint***

O modelo de mobilidade *waypoint*, descrito em [12], divide o percurso de um NM em períodos de movimentação e pausa. O NM fica num local por um determinado intervalo de tempo, denominado de tempo de pausa e depois move-se para um novo local escolhido aleatoriamente com uma velocidade que segue uma distribuição uniforme entre  $[\text{minspeed}, \text{maxspeed}]$ , conforme pode ser visto na figura 2.2. Através da configuração do intervalo do tempo de pausa em zero, esse modelo se comporta de forma semelhante ao modelo de mobilidade aleatória.

Esse modelo também é sem memória, com isso ele possui as desvantagens descritas na seção 2.1.1. Trabalhos como, [30, 31, 32], mostram que o *waypoint* apresenta características indesejáveis, evidenciando o seu comportamento não realístico. Entretanto, ele é um dos modelos de mobilidade mais usados pelos pesquisadores devido à sua simplicidade de implementação e configuração.

### **Modelo Markoviano de Percurso Aleatório**

O Modelo Markoviano de Percurso Aleatório - MPA foi proposto por Chiang em [33], onde o movimento é modelado através de uma Cadeia de Markov, e portanto, ele é um modelo de mobilidade com memória. Esse modelo possui três estados para representar as coordenadas  $x$  e  $y$ . O estado zero (0) representa a posição atual do NM em  $x$  e  $y$ , o estado um (1) representa a posição anterior do NM em  $x$  e  $y$ , e o estado dois (2) representa a próxima posição do NM também em

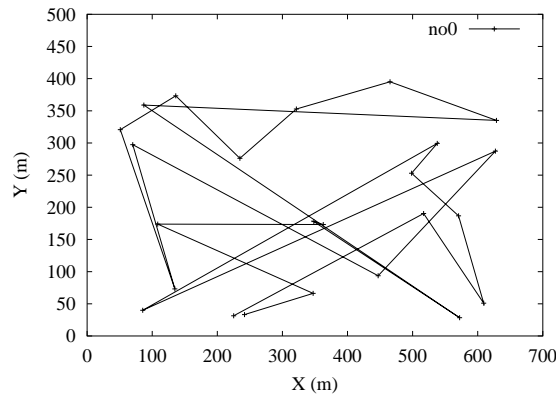


Figura 2.2: Percurso do no0 usando o *Waypoint*.

$x$  e  $y$ , conforme mostrado na figura 2.3. O modelo utiliza a seguinte matriz de probabilidade de transição para determinar a posição de um NM específico no próximo instante de tempo.

$$P = \begin{bmatrix} P(0,0) & P(0,1) & P(0,2) \\ P(1,0) & P(1,1) & P(1,2) \\ P(2,0) & P(2,1) & P(2,2) \end{bmatrix}$$

Nessa matriz, cada elemento  $P(a, b)$  representa a probabilidade de mudança do estado  $a$  para o estado  $b$ . Em [33] são usados os seguintes valores (matriz  $P_1$ ) para a transição de estados e que são ilustrados na figura 2.3.

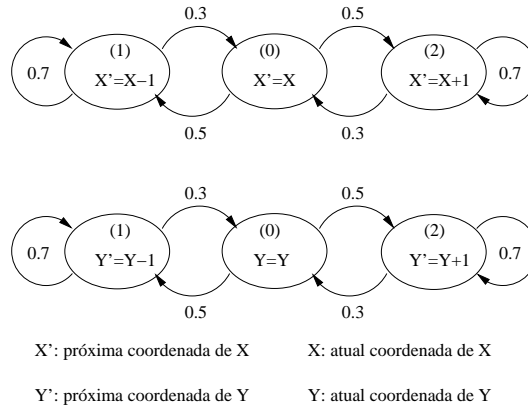


Figura 2.3: Diagramas de transições de estado do Modelo MPA.

$$P_1 = \begin{bmatrix} 0 & 0,5 & 0,5 \\ 0,3 & 0,7 & 0 \\ 0,3 & 0 & 0,7 \end{bmatrix}$$

Nesse modelo os NMs têm a possibilidade de movimentação somente nas diagonais, pois a probabilidade de transição de  $P(0, 0) = 0$ . Com isso, não é possível movimentar-se para as direções vertical e horizontal ou ficar parado numa posição qualquer durante um intervalo de tempo superior

a uma unidade. Além disso, a probabilidade do movimento continuar na mesma direção é maior que a de mudar de direção, fazendo com que o NM, uma vez em movimento, tenda a permanecer na mesma direção. Outra característica desse modelo é que ele não permite mudanças bruscas de sentido no movimento, pois para o NM mudar o sentido do movimento é necessário parar o movimento para depois mudar de sentido. Essas características mostram que esse modelo é mais realístico que o modelo de mobilidade aleatória e o modelo *waypoint*.

### Modelo de Mobilidade *Smooth Random*

O modelo de mobilidade *smooth random* proposto por Christian Bettstetter em [25], também é uma extensão do Modelo Aleatório. Neste, são definidos dois processos estocásticos para permitir a mudança de direção e de velocidade dos NMs. A principal contribuição desse modelo é permitir uma mudança suave na velocidade evitando paradas abruptas e acelerações bruscas no movimento, como descrito na figura 2.4. Entretanto, o modelo *smooth random* permite mudanças bruscas de direção. Além disso, nesse trabalho também são discutidas as principais regras de borda, e por fim é feita uma avaliação da distribuição dos nós em função das regras de borda.

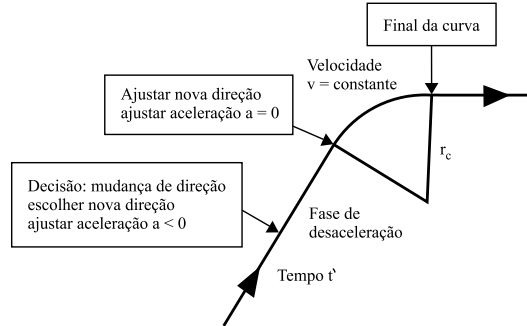


Figura 2.4: Comportamento de desaceleração antes de uma mudança de direção no modelo *smooth random*[25]

### Modelo A

O Modelo A, proposto em [30] usa uma cadeia de Markov, com parâmetro de tempo discreto, para a coordenada  $x$  e uma outra cadeia para  $y$ . O modelo permite movimentos nas direções vertical e horizontal, além de paradas no movimento dos NMs durante um ou mais intervalos de tempo. O Modelo A usa a matriz de probabilidade  $P_A$ , definida abaixo.

$$P_A = \begin{bmatrix} 0,2 & 0,4 & 0,4 \\ 0,3 & 0,3 & 0 \\ 0,3 & 0 & 0,7 \end{bmatrix}$$

Na figura 2.5 é mostrado o diagrama de transições de estado do Modelo A, onde a probabilidade de transição do estado (0) para o próprio estado (0) é 0,2. A transição do estado (0) para os outros estados é dada por 0,4. Já a permanência no estado (1) ou (2) é dada pela probabilidade 0,7, logo, a probabilidade de transição desses estados para o estado (0) é dada por 0,3.



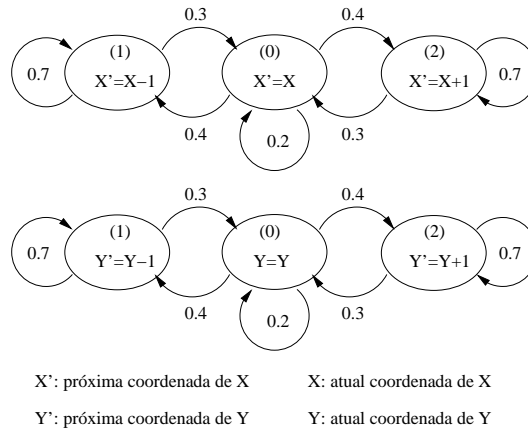


Figura 2.5: Diagrama de transições de estado do modelo A.

Em [30], é feita uma avaliação desse modelo, comparando suas características de movimentação com as dos modelos *waypoint* e o modelo MPA. Nessa avaliação é mostrado que esse modelo é mais realístico que os outros dois modelos.

### Modelo B

Como apresentado na seção 2.1.1, o Modelo A não permite grandes variações na velocidade. Sendo assim, o Modelo B também proposto em [30], visa resolver essa limitação.

Para ser possível uma velocidade variável é atribuído um conjunto de valores no incremento da posição inicial, que variará entre  $[1, n]$ , ou seja, o incremento corresponde à velocidade que o NM usa para mudar da posição atual que é  $X$  para a próxima posição que é  $X'$ . A variação do valor da velocidade segue o comportamento de uma série geométrica, onde o valor inicial é 1 e o valor máximo é  $n$ .

Como valores das probabilidades de transição de cada estado do Modelo B, mostrado na figura 2.6, é definido em [30].

As equações que definem todas essas probabilidades são apresentadas em [30]. Como mostrado analiticamente também em [30], o Modelo B possui as seguintes características:

- A distribuição das probabilidades de um estado ( $i$ ) segue uma série geométrica de razão  $\frac{1}{2}$ , ou seja, o valor de  $P_{\frac{1}{4}(|i|+2),i}$  é a metade do valor de  $P_{\frac{1}{4}(|i|+1),i}$ ,  $|i| > 0$ . Essa característica define uma alta possibilidade de estando num estado ( $i$ ) no instante  $t$ , ir para os estados vizinhos mais próximos à direita ou à esquerda no instante  $t + 1$ , ou seja, o modelo evita acelerações e desacelerações bruscas no movimento;
- A velocidade aumenta exponencialmente até o valor máximo  $n$ , dessa maneira, o modelo apresenta uma aceleração variada;
- Estando num estado ( $i \rightarrow positivo$ ), não é possível passar para um estado ( $i \rightarrow negativo$ ) sem passar pelo estado (0) e vice-versa. Através dessa característica, o Modelo B evita mudanças bruscas de direção.

Além dessas características, o Modelo B tem a capacidade de representar tanto padrões de

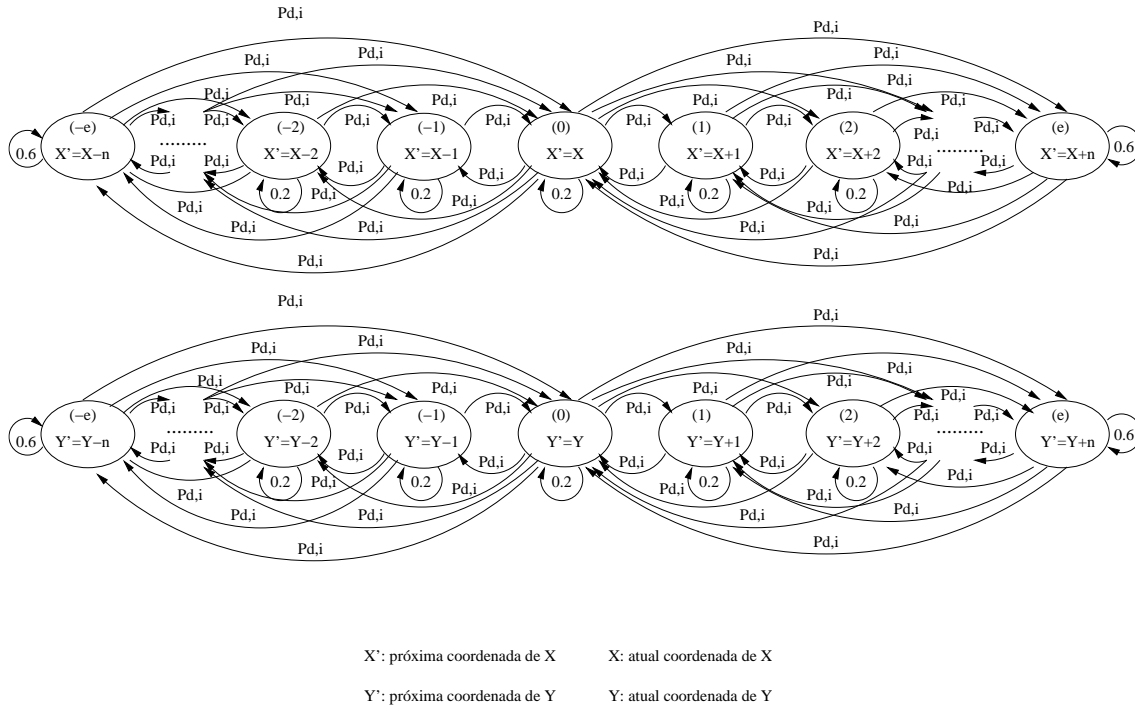


Figura 2.6: Diagrama de transição de estado do Modelo B

movimentação de apenas um incremento nas coordenadas  $x$  e  $y$  (como por exemplo o Modelo A), como diferentes valores de incrementos nessas coordenadas. Sendo assim, o Modelo B é genérico, possibilitando representar vários padrões de movimentação.

Existem outros modelos de mobilidade individual propostos na literatura. Em [4], é apresentado um modelo com uma área de simulação sem fronteira. No trabalho [18], é descrito o modelo de mobilidade *Gauss-Markov* utilizado em redes *ad hoc*. O modelo de mobilidade *City Section* é proposto em [4], onde busca-se representar a movimentação de NMs numa área urbana.

### 2.1.2 Modelos de Mobilidade em Grupo

Os modelos de mobilidade em grupo são modelos que representam o movimento de um grupo de NMs, onde o comportamento de movimentação dos NMs é dependente, tanto em relação a intervalos de tempo quanto ao relacionamento entre eles. Além desses modelos serem utilizados para predizerem a disponibilidade do enlace sem fio, recentemente estão sendo usados também na predição do particionamento das redes *ad hoc*, que são mudanças de escala global na topologia da rede [28]. A principal causa desse particionamento é o próprio comportamento de movimentação em grupo dos NMs.

#### Modelo de Mobilidade em Grupo com Ponto de Referência - MMGPR

Esse modelo foi desenvolvido por Hong *et al.* em [18]. Nele cada grupo tem um centro de referência lógico. O movimento do centro define o comportamento de todo o grupo incluindo localização, velocidade, direção, aceleração etc. Desse modo, a trajetória do grupo é composta por

um caminho até o centro. Os NMs são distribuídos uniformemente dentro da área geográfica do grupo. Para cada NM, é atribuído um ponto de referência, aos quais, segue o movimento do grupo. Um NM é colocado aleatoriamente na vizinhança do ponto de referência. O esquema de referência num ponto admite o comportamento de movimentação aleatória independente para cada NM, além do movimento do grupo.

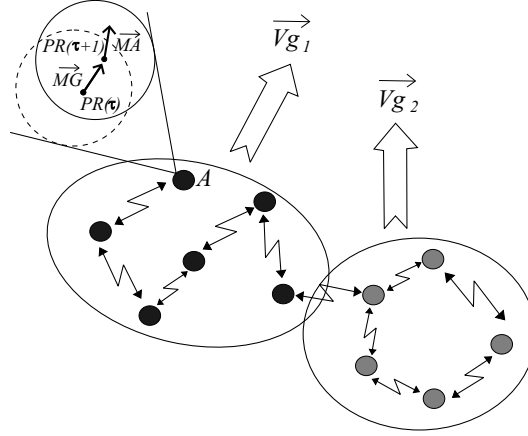


Figura 2.7: Modelo MGPR [18].

Na figura 2.7 é ilustrado o MMGPR com dois grupos. Cada grupo tem um vetor de movimentação de grupo  $\vec{V}_{gi}$ . Na movimentação do NM  $A$ , o ponto de referência move de  $PR(\tau)$  para  $PR(\tau + 1)$  com o vetor de movimentação do grupo  $\vec{MG} = \vec{V}_{g1}$ . Então, a nova posição do NM é gerada adicionando o vetor de movimentação aleatória  $\vec{MA}$  para o novo ponto de referência  $PR(\tau + 1)$ . Esse vetor tem o comprimento seguindo uma distribuição uniforme dentro de um certo raio centrado no ponto de referência e a direção distribuída uniformemente entre 0 e 360°. Além disso, o vetor é independente da localização anterior do nó.

O MMGPR define o movimento de grupo explicitamente, dando um caminho do movimento para cada grupo. Esse caminho será dado definindo uma seqüência de *checkpoints* (pontos de verificação) ao longo da trajetória correspondendo a um intervalo de tempo. Em cada tempo, o centro do grupo procura um novo *checkpoint* e calcula o novo vetor de movimentação de grupo  $\vec{V}_{gi}$ , o próximo local do *checkpoint* e seu intervalo de tempo.

Esse modelo é usado nas seguintes aplicações:

- Grupos em diferentes áreas executando a mesma tarefa, como em campos de batalha ou em recuperação de desastres. Nesses cenários, os grupos têm o mesmo vetor de movimentação  $\vec{V}_{gi}$ ;
- Diferentes grupos executando tarefas diferentes sobre a mesma área, ou seja, grupos com padrões de mobilidade diferentes na mesma região, onde cada grupo possui um vetor de movimentação  $\vec{V}_{gi}$  distinto.

### Modelo de Mobilidade em Grupo baseado numa Velocidade de Referência - MMGVR

Uma proposta de extensão do modelo de mobilidade em grupo - MMGPR [18], apresentado na seção 2.1.2, é o Modelo de Mobilidade em Grupo baseado numa Velocidade de Referência -

MMGVR. Neste modelo, propõe-se um comportamento de movimentação em grupo baseado numa velocidade média.

## 2.2 Regras de Borda para Simulação de Áreas Limitadas

A observação do comportamento de movimentação dos NMs é feita necessariamente dentro de uma determinada área, que pode ser, tanto ambientes abertos (rodovias, deslocamento terrestre de tropas militares, cidades ou centros comerciais), quanto ambientes fechados (*shoppings*, prédios ou salas). Para representar a mobilidade dos NMs em cenários simulados, também é utilizada uma área limitada. O limite dessa área é denominado de borda ou fronteira. Dessa maneira, quando um NM atinge a borda de simulação, é necessário decidir o que fazer com o movimento desse NM. Esse processo é chamado de Regra de Borda.

Na literatura, existem várias regras de borda propostas. A seguir serão descritas as principais regras de borda: *bounce*, *delete and replace*, e *wrap around*, apresentadas em [25].

### 2.2.1 *Bounce*

A regra de borda *bounce*, apresentada em [14, 25], é definida como sendo uma reflexão do movimento do NM na borda da área de simulação (tendo como forma geométrica um retângulo qualquer), fazendo com que a nova trajetória do NM permaneça dentro da área de simulação. Esse novo movimento é caracterizado por dois componentes, o ângulo da direção -  $\theta$  e a velocidade -  $v$ , como pode ser visto na figura 2.8. O novo valor do ângulo será  $-\theta$  nas bordas superior, inferior e  $\Phi - \theta$  nas bordas laterais da área de simulação. O valor da velocidade  $v$  permanecerá o mesmo de antes da reflexão.

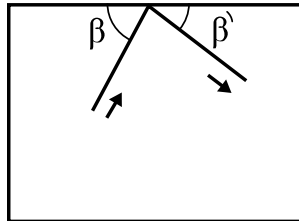


Figura 2.8: Regra de Borda *Bounce*

Existem algumas extensões dessa regra, como a apresentada em [25], onde o novo valor de  $\theta$  é uniformemente distribuído entre  $[0, 180^\circ]$  nas bordas: superior, inferior e laterais, e  $[0, 90^\circ]$  nos cantos da área de simulação. O valor da velocidade  $v$  também segue uma distribuição uniformemente distribuída entre  $[Vmin, Vmax]$ .

### 2.2.2 *Delete and Replace*

A regra de borda *delete and replace*, descrita em [25], busca representar um cenário onde os NMs podem atravessar a fronteira da área de simulação, como acontece em muitas situações reais (movimentação de automóveis em estradas e rodovias, entrada e saída de pessoas de uma sala,

prédio ou *shopping*). Nessa regra é definido que quando um NM atinge a borda, este é retirado da simulação e substituído por um novo NM que será inserido aleatoriamente dentro da área simulada, como pode ser mostrado na figura 2.9.

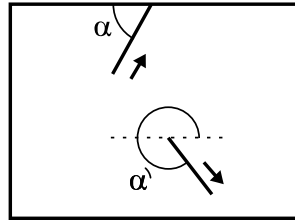


Figura 2.9: Regra de Borda *Delete and Replace*

### 2.2.3 *Wrap around*

A regra de borda *wrap around*, apresentada em [25], utiliza o mecanismo de espelhamento do movimento do NM na borda oposta à fronteira alcançada. Esse espelhamento do movimento preserva a mesma direção e velocidade do NM no momento de alcance à borda, conforme ilustrado na figura 2.10.

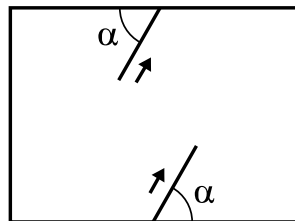


Figura 2.10: Regra de Borda *Wrap around*

Dentro desse contexto, devem haver critérios para a escolha da regra de borda e cuidados devem ser tomados na sua utilização, pois essas regras influenciam a avaliação do desempenho de sistemas e aplicações simuladas.

## 2.3 Métricas de Avaliação de Modelos de Mobilidade

Para avaliar o nível de realismo ou de desempenho de um determinado comportamento dos modelos de mobilidade é necessário a utilização de métricas de avaliação. Têm-se como métricas de avaliação: mudanças bruscas de direção, paradas abruptas, acelerações bruscas e variações bruscas de velocidade. Cada métrica será definida a seguir.

**Mudança brusca de direção:** é quando o ângulo de mudança de direção do movimento estiver no intervalo  $[90^\circ, 180^\circ]$  ou  $[135^\circ, 180^\circ]$ . Essa métrica avalia se as mudanças de direção são suaves ou não.

**Paradas abruptas no movimento:** é caracterizada por uma desaceleração muito rápida do NM fazendo com que a sua velocidade chegue a zero muito rapidamente. Para que uma parada

seja considerada brusca, duas condições devem ser satisfeitas. Primeira, um NM que no instante de tempo  $(t)$  possuía uma velocidade  $v$ , ao passar para o instante de tempo  $(t + 1)$  sua velocidade deve se igualar a zero. Segunda, sua  $\Delta V$  (variação de velocidade) deve ser maior ou igual a 50% da velocidade máxima simulada.

As **acelerações bruscas** possuem uma definição semelhante a de paradas abruptas, porém com uma aceleração positiva. Novamente devem ser consideradas duas condições para que uma aceleração no movimento seja considerada brusca. Primeira, no instante de tempo  $(t)$ , o NM possuía uma velocidade igual a zero e ao passar para o instante de tempo  $(t + 1)$  a sua velocidade alcança o valor  $v$ . Segunda, sua  $\Delta V$  deve ser maior ou igual a 50% da velocidade máxima simulada.

**Variações bruscas de velocidade** é a métrica definida pela  $\Delta V$  de um NM.  $\Delta V$  é calculada a partir da diferença entre a velocidade  $v_{t+1}$  e  $v_t$ , as quais ocorrem nos instantes de tempo  $(t + 1)$  e  $(t)$ , respectivamente.  $\Delta V$  é considerada brusca se seu valor for maior ou igual a 50% da velocidade máxima simulada, onde  $v_t$  e  $v_{t+1}$  são definidas no intervalo  $[0, \text{velocidade máxima simulada } (vmax)]$ .

## Capítulo 3

# Ferramentas de Simulação

Nesse capítulo serão apresentadas as ferramentas de simulação mais utilizadas em redes *ad hoc*. Serão descritos em detalhes os simuladores de redes e geradores de cenários de mobilidade.

### 3.1 Simuladores de Redes

#### 3.1.1 *Network Simulator* – NS-2

##### Introdução

O simulador de redes NS-2 [34] simula eventos discretos, permitindo a execução de simulações de diversos tipos de redes e componentes: cabeadas, sem fio, locais, via satélite e também redes *ad hoc*. Sua implementação foi feita com a linguagem C++ e apresenta-se disponível para instalação em pacotes de dois tipos, o *allinone* onde todos os pacotes necessários para que o ambiente suporte o simulador são incluídos, e o pacote contendo apenas os fontes do NS, onde a instalação de todos os pré-requisitos necessários para a instalação do NS-2 ficam a cargo do usuário.

##### Como o Simulador Está Organizado

O NS-2 apresenta-se organizado de maneira que sua utilização torna-se bastante flexível. Seus scripts de configuração podem ser criados de diferentes maneiras, porém determinados parâmetros de entrada, como era de se esperar, devem ser informados obrigatoriamente.

É possível obter parâmetros de entrada através de linha de comando, dispostos no próprio *script* principal de entrada para o NS-2, ou mesmo através da inclusão de outros arquivos de configuração complementares nesse *script* principal.

O *script* de entrada para o NS-2 utiliza a linguagem TCL, nele a quantidade de nós, o padrão de mobilidade, protocolo de roteamento, dentre outras características importantes para o cenário da simulação são informados. Nesse *script* também deve-se informar quais camadas de rede devem ser simuladas, aquelas selecionadas terão seus dados de simulação incluídos no arquivo de trace gerado pelo NS.

Estão listadas a seguir as informações incluídas no *script* TCL que são esperadas como parâmetros de entrada pelo simulador:

- Características da Topologia;
- Indicação de quais camadas devem ser simuladas;
- Padrão de movimento;
- Tipo de tráfego gerado pela comunicação entre os nós;
- Estabelecer quais dados devem ser gerados pela simulação;

Um ponto importante em relação a configuração dos cenários de simulação é justamente qual padrão de mobilidade utilizar. Para gerar a movimentação dos nós segundo um determinado padrão de mobilidade podem ser utilizadas as ferramentas: SETDEST, que já vem no pacote do simulador, o SCENGEN e o *BonnMotion*, que são ferramentas adicionais que geram cenários de mobilidade compatíveis com o NS.

Todas essas ferramentas devem ser configuradas de acordo com o modelo de mobilidade desejado, e quando executadas geram um arquivo de *trace* contendo cada movimento feito pelos nós. É justamente esse arquivo que o NS-2 espera receber como entrada para simular a movimentação dos nós. A forma como configurar e executar essas ferramentas fogem ao escopo desse trabalho, mas essas tarefas são a princípio simples, além da documentação das mesmas mostrar-se adequada e suficiente.

Visualizar graficamente as simulações feitas no NS-2 é realmente muito útil, e com este objetivo pode ser usada a ferramenta *Network AniMator* - NAM. O uso dessa ferramenta é particularmente importante em simulações relativas a modelos de mobilidade, onde deseja-se observar a movimentação dos nós.

Quando o NS-2 termina a execução de uma simulação ele gera um arquivo .nam, como mencionado anteriormente. Esse arquivo é usado pelo NAM para mostrar na tela do computador a simulação, com a movimentação dos nós, a comunicação entre eles e tudo mais que tiver sido gerado nesse arquivo de entrada.

Antes mesmo de executar a simulação no NS, pode-se ter em mãos o arquivo contendo o padrão de movimentação dos nós. O conteúdo desse arquivo pode ser visualizado graficamente sem a necessidade de realizar a simulação, pois para isso existe a ferramenta Ad Hockey, que espera como parâmetro de entrada arquivos gerados pelo SCENGEN ou SETDEST. O auxílio dessa ferramenta (Ad Hockey) é de grande valor, tendo em vista que a movimentação desejada pode ser ostensivamente estudada antes mesmo de realizar-se uma simulação.

## Como Configurar

Por ser bastante flexível, em um primeiro momento pode parecer complicado configurar uma simulação no NS. Porém estabelecendo um padrão que seja adequado para suas próprias necessidades é possível tornar a configuração simples e eficiente. O padrão de entrada de parâmetros a seguir mostrou-se eficiente durante a realização de diversas simulações, podendo ser adotado, mas não devendo ser tratado como uma regra.

```
# Verifica se os parâmetros para simulação foram informados corretamente
if [expr $argc != 8] {
    puts " "
```



```

    puts "Syntax: ns config.tcl N X Y Scenario_File Connection_Pattern Stop_time Algo
rit_Rote Trace_file"
    puts " "
exit 0
}
puts " "

# Definição das características da simulação

set val(chan)      Channel/WirelessChannel
set val(prop)      Propagation/TwoRayGround
set val(netif)     Phy/WirelessPhy
set val(mac)       Mac/802_11
set val(ifq)       Queue/DropTail/PriQueue
set val(ll)        LL
set val(ant)       Antenna/OmniAntenna
set val(x)         [lindex $argv 1]      ;# X dimensão
set val(y)         [lindex $argv 2]      ;# Y dimensão
set val(ifqlen)    50                    ;# quantidade máxima de pacotes na fila
set val(seed)      1.0
set val(adhocRouting) [lindex $argv 6];# AODV, DSDV, DSR
set val(nn)        [lindex $argv 0]      ;# quantidade de nós simulados
set val(cp)        [lindex $argv 4]      ;# padrão de conexão
set val(sc)        [lindex $argv 3]      ;# padrão de cenário
set val(stop)      [lindex $argv 5]      ;# tempo de simulação
set val(trace_file) [lindex $argv 7];# arquivo de trace

puts "1"
puts $val(adhocRouting);
puts "2"

# Inicialização da variáveis globais
# Criação da instância do simulador

set ns_ [new Simulator]
$ns_ use-newtrace

set topo [new Topography]

# Criação do objeto para o ns e o nam
# Estabelecimento de quais dados devem ser gerados pela simulação (traces e *.nam)

set tracefd [open $val(trace_file) w]
#set namtrace [open wireless1-out.nam w]

```

```

$ns_ trace-all $tracefd
#$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# Definição das características da Topologia
$topo load_flatgrid $val(x) $val(y)

```

Como mostrado nesta seção, o NS-2 é um simulador de redes bastante abrangente e genérico, o que o torna muito utilizado pela comunidade científica.

### 3.1.2 O Simulador GlomoSim

#### Introdução

O *Global Mobile Information System Simulator - GloMoSim* [35] é usado para realizar simulações de redes de diferentes tamanhos, permitindo a comunicação de nós em diversos cenários. Seus criadores preocupados com o desempenho do simulador, desenvolveram-no utilizando a linguagem *Parallel Simulation Environment for Complex Systems - PARSEC*, uma linguagem para execução paralela de modelos de simulação, através da qual mesmo simulações com um número muito grande de nós podem ser realizadas com duração de tempo aceitáveis.

#### Como o simulador está organizado

O simulador após configurado corretamente, seguindo as dicas encontradas na sua documentação, pode ser executado com o comando `./glomosim config.in`. Mas não ficando limitado a essa forma de execução, também é possível utilizar a interface gráfica, implementada em java, para executar o simulador e visualizar o *trace* gerado.

Através dos menus da interface gráfica é possível selecionar 3 formas diferentes de execução do simulador:

- Tempo Real (*Real Time*)
- Geração de Trace (*Write Trace*)
- Reprodução do Trace (*Play Back*)

A configuração utilizada pelo simulador como entrada para sua execução deve ser configurada utilizando-se o arquivo `config.in`, que possui um formato muito simples e de fácil compreensão para manipulação das opções presentes nele. Informações como tipo de protocolo utilizado na camada MAC ou utilizado para o roteamento, todas elas devem ser configuradas através desse arquivo.

Para cada característica do cenário que deseja-se simular, como os protocolos que acabaram de ser citados, existem diversas opções pré-definidas, por exemplo, quando configurando o protocolo de roteamento utilizado no arquivo `config.in` encontram-se disponíveis as opções: BELLMANFORD, AODV, DSR, LAR1, WRP, FISHEYE, ZRP, STATIC. A opção desejada deve ser descomentada, deixando as demais com o sinal de comentário(`#`) antes de cada uma, como pode-se observar no próprio arquivo `config.in`.

Parte importante da configuração do cenário são as características da comunicação que deve ocorrer entre os nós da rede. O tipo de tráfego de acordo com diferentes tipo de aplicação, como

FTP e HTTP, e os instantes que devem ser gerados, todas essas informações devem ser configuradas através do arquivo **app.conf**. Nesse arquivo estão presentes diversos comentários e exemplos a respeito de como utilizá-lo.

### Como configurar

Como comentado na seção anterior, configurar os parâmetros de entrada para o GloMoSim não é uma tarefa complicada, isso se deve a forma como o simulador está organizado. Nessa seção será mostrado, de forma sucinta, o objetivo de cada arquivo utilizado e gerado pelo GloMoSim.

#### Arquivo: **config.in**

Nesse arquivo encontram-se todos os parâmetros e suas opções, para que o simulador saiba qual cenário deve ser simulado. Tempo de simulação, perímetro da área de simulação, quantidade de nós, disposição inicial dos nós, padrão de mobilidade dos nós, protocolo usado na camada MAC, protocolo de roteamento e alguns outros parâmetros devem ser configurados nesse arquivo. Também deve-se informar quais dados devem estar presentes nas estatísticas geradas no final da simulação.

#### Arquivo: **app.conf**

Através desse arquivo deve-se informar o tipo de tráfego gerado, estando disponíveis os seguintes tipos: FTP, FTP/GENERIC, TELNET, CBR, e HTTP. Exemplos e comentários estão presentes e são suficientes para auxiliar sua configuração.

#### Arquivo: **nodes.input**

Quando a opção NODE-PLACEMENT FILE é utilizada no parâmetro de disposição inicial dos nós um arquivo deve ser informado, através do qual o simulador pode obter as coordenadas iniciais de cada nó. O formato para configuração desse arquivo é simples, bastando seguir os exemplos encontrados nele mesmo.

#### Arquivo: **mobility.in**

Se a opção MOBILITY TRACE for selecionada no parâmetro mobilidade, então esse arquivo é utilizado pelo GloMoSim como fonte para o padrão de movimentação de cada nó. Como é possível ver através dos exemplos no próprio arquivo, para cada movimento de um nó deve-se informar o instante, a identificação do nó, as coordenadas(x, y, z) e a velocidade do nó.

#### Arquivo: **routes.in**

Utilizando a opção ROUTING-PROTOCOL STATIC no parâmetro protocolo de roteamento esse arquivo é usado pelo simulador para obter a tabela de rotas.

#### Arquivo: **Trace**

Os arquivos de *trace* gerados podem ser renomeados, ou seja, não chamam-se *trace* necessariamente. Através desse arquivo é possível acompanhar todo o histórico de uma execução, sendo esse mesmo arquivo utilizado pela interface gráfica na opção *Play Back*.

#### Arquivo: **glomo.stat**

Todas as informações geradas nesse arquivo baseiam-se nas opções de estatísticas selecionadas no arquivo de configuração **config.in**. O resultado da simulação apresenta-se organizado de forma bastante clara para cada nó presente no cenário simulado.

## Opções de configuração pré-definidas

Abaixo estão os parâmetros e cada uma de suas opções presentes no arquivo de configuração **config.in**.

Modelos de Mobilidade: *Randon Waypoint*, *Randon Drunken*, *Trace Based*

Modelos de Propagação: *Free Space*, *Two Ray*

Radio Noise: *Accumalated noise*, *No noise*

Modelos de recebimento de pacotes: SNR, BER

Protocolos da camada MAC: CSMA, MACA, 802.11, TSMA, *Wiredlink*

Protocolos de Roteamento: Bellman-Ford, *Fisheye*, WRP, AODV, DSR, LAR, ODMPR

Modelos da camada de Transporte: *FreeBSD* TCP, UDP

Protocolos da camada de Aplicações: CBR, HTTP, TELNET, FTP

O resultado da simulação pode ser mostrado através de um visualizador gráfico, como pode ser visto na figura 3.1.

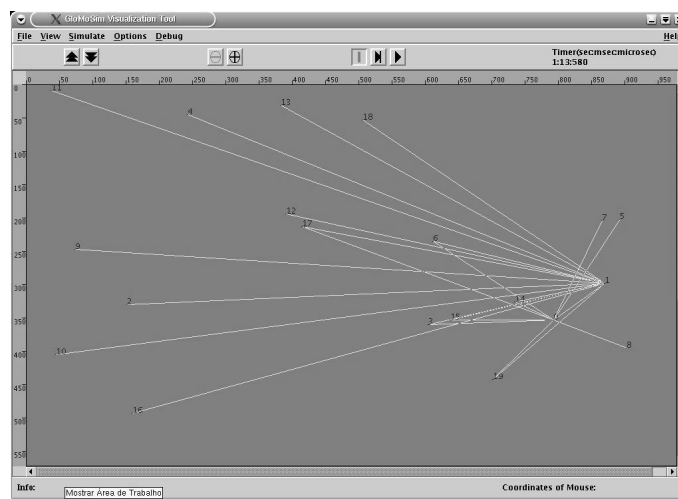


Figura 3.1: Visualizador de simulações do GloMoSim

## Inclusão de novos modelos

Como mostrado nas seções anteriores, o GloMoSim possui diversos protocolos, modelos de mobilidade e outras opções quanto as características desejadas para a rede que se deseja simular. Porém, como pode-se imaginar, é possível que um determinado modelo ou protocolo não esteja presente na versão atual do simulador. Assim, caso necessário, o simulador admite que novas opções para as características dos cenários de simulação sejam adicionadas ao simulador.

Entretanto, o pesquisador que desejar implementar um novo de modelo de mobilidade, por exemplo, deve ter conhecimentos de programação, mais especificamente da linguagem C, além de saber utilizar a biblioteca PARSEC, utilizada pelo GloMoSim em sua própria implementação.

É necessário programar o próprio modelo através da criação dos arquivos: **modelo-teste.pc**, **modelo-teste.h** e **modelo-teste.pi**. Informações sobre a melhor forma assim como modelos sugeridos pelos criadores do simulador para implementação de novos modelos podem ser encontradas junto a seu pacote de instalação. Porém, a adição de um novo modelo implica em algumas alterações

em alguns dos arquivos do simulador. No exemplo mencionado acima, para inclusão de um novo modelo de mobilidade os arquivos **mobility.h** e **mobility.pc** também sofreriam alterações.

Para a inclusão de novos mecanismos no GloMoSim, conforme mencionado acima, é necessário conhecimento de programação em C e também conhecimentos básicos em relação à utilização da biblioteca PARSEC, no entanto, não são necessárias mudanças no “Kernel” do simulador, para as quais o usuário necessitaria de profundos conhecimentos da biblioteca PARSEC

### 3.1.3 Diferenças entre o GloMoSim e o NS-2

Nas seções anteriores foram apresentadas duas ferramentas de simulação para redes sem fio. A maneira de configurar, alterando e implementando de formas diferentes os dados de entrada para as simulações foi mostrada para cada simulador. A partir dessas informações podem ser avaliadas de forma genérica as melhores características de cada um quanto a utilização. Certamente, cada usuário que desejar utilizar um simulador para desenvolver um trabalho específico terá de avaliar segundo seus próprios critérios quais são os pontos fortes e fracos de cada um.

#### Prós e contras - GloMoSim

Como dito anteriormente o ponto forte do simulador GloMoSim é a forma como seu arquivo de configuração para entrada de parâmetros mostra-se organizado. Sendo de simples compreensão, o que permite um aprendizado mais rápido quanto a forma de utilização do mesmo. Tornando a implementação de novos cenários de simulação mais rápida em uma primeira análise.

Além do arquivo de configuração principal, o **config.in**, os demais arquivos onde devem ser informados o modelo de tráfego, modelo de mobilidade, etc. também possuem cabeçalhos que facilitam o entendimento de como o arquivo funciona e conseqüentemente como deve ser configurado para simulações particulares.

Apesar da maioria de seus arquivos mostrarem-se bem organizados, tornando o manuseio por parte dos usuário mais eficiente, alguns arquivos não apresentam uma legenda ou cabeçalho explicativo quanto a sua configuração, ou seja, não é possível saber de imediato o que cada campo ou coluna apresentados significam exatamente.

A documentação incluída no pacote do simulador (contida no arquivo **glomosim-2.0.tar.gz**) não cobre todos os pontos referentes às funcionalidades e configurações do simulador. No entanto, haja vista a boa apresentação dos arquivos de configuração, esse fato não se apresenta como um grande empecilho.

Entretanto, dentre os pontos fortes dessa ferramenta de simulação as opções de estatísticas que podem ser selecionadas dentro do arquivo principal de configuração mostram-se imediatamente como um diferencial em relação ao NS.

#### Prós e contras - NS

Esse simulador exige mais experiência e conhecimento para a sua utilização do que o GloMoSim. Para configurá-lo, o usuário necessita de conhecimentos sobre a linguagem TCL e por isso alguma noção de programação é indispensável.

O NS é muito flexível quanto a configuração de seus parâmetros de entrada, permitindo a

utilização de métodos diferentes com tal objetivo, como mostrado anteriormente nesse capítulo. Toda essa flexibilidade exige maior grau de conhecimento do usuário, pois a implementação de um cenário simples para simulação pode apresentar erros na elaboração dos *scripts* TCL, e para depurar e resolver estes erros um usuário leigo pode encontrar dificuldades nas questões relativas à programação e ao uso do ambiente UNIX.

A geração de resultados a partir dos dados gerados pelo NS exige conhecimentos que vão além da simples utilização do próprio simulador. Não havendo a opção, como acontece no GloMoSim, de solicitar que o próprio simulador gere alguns resultados estatísticos para avaliação do cenário simulado. Para a obtenção destes dados estatísticos é necessária a utilização de uma linguagem adicional como AWK ou C para o processamento dos *logs* de simulação do NS.

O NS é muito utilizado em trabalhos de pesquisa, artigos etc. Por esse motivo, quando buscase utilizar uma ferramenta mais amplamente testada e avaliada, tendo como referência diversos trabalhos que também utilizaram esse simulador, sua utilização apresenta-se como a melhor opção.

## 3.2 Geradores de Cenários de Mobilidade

### 3.2.1 Scenario Generator – ScenGen

#### A Ferramenta

Com o objetivo de desenvolver simulações em redes móveis *ad hoc*, Li Qiming, um pesquisador da Universidade de Singapura, desenvolveu o gerador de cenários ScenGen [36]. O ScenGen é uma ferramenta destinada a gerar cenários de mobilidade, através de modelos de mobilidade ou a partir de *traces* de movimentação dos nós da rede, exibindo uma saída configurada para o uso no simulador de rede NS-2.

A ferramenta Adhockey [37], que é integrada ao NS-2, pode ser utilizada para a visualização da movimentação dos nós dentro do cenário gerado, como pode ser visto nas figuras 3.2(a) e 3.2(b).



(a) Visualizando o cenário

(b) Visualizando o *trace* de movimentação

Figura 3.2: O visualizador Adhockey

O ScenGen foi desenvolvido na linguagem C++ e implementa os seguintes modelos de mobilidade: *Random Waypoint*, *Pursue*, *Gauss-Markov*, *Brownian*, *Column*.

O ScenGen dá suporte à alteração dos modelos existentes e também à inclusão de novos modelos. Todos os modelos são implementados em linguagem C++ através de uma classe com herança da

classe **model** (já implementada pelo ScenGen). Dentro da classe que representa o modelo, existem dois métodos indispensáveis para a simulação: o **init** e o **makeMove**.

**init**: método chamado para carregar os parâmetros iniciais necessários para o modelo. Neste método devem ser carregados parâmetros iniciais tais como velocidade máxima, semente inicial da simulação, regra de borda a ser utilizada, etc. Estes parâmetros são lidos do arquivo **scen-spec**.

cabeçalho (*header*) da função **init**:

```
init(model_time_t startTime, model_time_t stopTime, node_id_t
startID,int num_nodes, Area *area, bool cp)
```

Os parâmetros que estão no cabeçalho da função devem ser repassados para a função **init** da classe pai **Model**. No método **Model::init** são definidas as características do cenário, tais como número de nós, tempo inicial, tempo final, área de simulação, etc.

**makeMove**: método chamado para calcular a próxima posição de um nó presente na simulação. Para estabelecer a próxima movimentação do nó, o ScenGen chama esta função passando como parâmetro o nó a sofrer movimentação. Deste modo as variáveis que representam a velocidade e posição deste nó devem ter seus valores atribuídos neste método respeitando as regras do modelo de mobilidade implementado.

cabeçalho da função **makeMove**:

```
makeMove(Node *node)
```

## Configurações da Simulação

Todos os parâmetros admitidos por cada modelo estão presentes no arquivo **model-spec**. Dentro deste arquivo também encontra-se o valor padrão para cada parâmetro. O arquivo **scen-spec** por sua vez representa um cenário a ser simulado, ou seja, os parâmetros presentes neste arquivo se referem somente a este cenário.

Dentro do arquivo **scen-spec** existe uma seção chamada global que deve conter configurações referentes ao:

- Tamanho da área retangular de simulação;
- Tempo de início da simulação;
- Tempo de término da simulação.

Além da seção global, para cada grupo presente na simulação existe uma seção nomeada com o nome do grupo e possui como parâmetros: o número de nós presentes neste grupo, o modelo que rege o movimento destes nós e outros parâmetros referentes ao modelo simulado. Os parâmetros do modelo que não são referenciados no **scen-spec** serão inicializados com o valor padrão contido no arquivo **model-spec**.

Exemplo de um arquivo **model-spec**:

```
#####  
## The Waypoint model  
#####
```

```
[Waypoint]
```

```
# The pause time  
T_min = 0  
T_max = 0  
T_dist = uniform
```

```
# The min and max speed  
V_min = 0  
V_max = 30  
V_dist = uniform
```

```
#####  
## The Smooth model  
#####
```

```
[Smooth]
```

```
Accel_max=32  
Accel_min=0  
Prob_Veloc_Change=1.0  
Time_Moves=1.0  
num_veloc_pref=1
```

```
Veloc_Pref1=2  
Prob_Veloc_Pref1=0.3  
Prob_Direction_Change=0.6
```

```
Border_Rule=0
```

Exemplo de um arquivo **scen-spec**:

```
[Global]
```

```
area = rect(700,500)  
start_time = 0  
stop_time = 150
```

```
[helicopter]
```



```

num_nodes = 50

member_model = Smooth
member_model.Border_Rule=3
member_model.Accel_max = 1
member_model.Accel_min = 0

member_model.Prob_Veloc_Change = 0.75
member_model.Time_Moves = 1

member_model.num_veloc_pref = 3

member_model.Veloc_Pref0 = 10
member_model.Prob_Veloc_Pref0 = 0.25
member_model.Veloc_Pref1 = 20
member_model.Prob_Veloc_Pref1 = 0.50
member_model.Veloc_Pref2 = 30
member_model.Prob_Veloc_Pref2 = 0.25

```

### Alterando um Modelo Existente

Para alterar um modelo existente deve-se alterar a classe que representa o modelo em questão. As alterações com relação aos parâmetros devem ser refletidas no arquivo **model-spec** e no método **init** da classe. Já as alterações referentes a mobilidade dos nós devem ocorrer no método **makeMove**.

### Adicionando um Modelo

Para adicionar um modelo à ferramenta deve-se criar um arquivo com o código do novo modelo na linguagem C++. Este código deve estar estruturado em uma classe com herança da classe **model**, que é implementada pelo ScenGen. Dentro desta classe todo o código que representa o movimento dos nós deve estar no método **makeMove**. Este método é chamado sempre que a ferramenta necessite calcular a próxima posição de um nó. Deste modo, as variáveis que representam a velocidade e a posição, devem ter seus valores atribuídos neste método.

Todo nó da simulação é um objeto da classe **node**. Esta classe possui algumas propriedades relativas à movimentação do nó. Estas propriedades são:

`node->dest_`: representa a posição de destino que o nó possuirá após a execução do movimento atual. É representada por uma estrutura que contém os valores *x* e *y* de uma posição dentro da área de simulação.

`node->pos_`: representa a posição atual do nó. É representada por uma estrutura que contém os valores *x* e *y* de uma posição dentro da área de simulação.

`node->startTime_`: representa o tempo em que o nó irá iniciar o movimento atual.

`node->arrivalTime_`: representa o tempo em que o nó irá atingir a posição `node->dest_`.

`node->nextStartTime_`: representa o tempo em que o nó irá iniciar o próximo movimento.

node->speed.: representa a velocidade do nó no movimento atual.

Assim, as variáveis descritas acima devem ter os respectivos valores atribuídos (no método **makeMove**) para que a movimentação ocorra corretamente.

Após criar o código para o modelo deve-se acrescentar no arquivo **model.h** a linha “enum NomeModelo” para que o ScenGen possa fazer referência ao novo modelo. Também deve-se acrescentar no arquivo **modelspec.cc** uma chamada para o modelo, como no exemplo abaixo, no método **ModelSpec::createModelByName()**:

```
else if (! strcmp(modelName, "NomeModelo")) {  
    model = new NomeModelo ();          }  
}
```

Vale lembrar que todos os parâmetros que serão fornecidos ao modelo devem estar presentes no arquivo **model-spec**.

Como o ScenGen deve ser recompilado após a adição de um novo modelo, o arquivo **Makefile** deve conter os arquivos com o código da implementação que foi criado. Além disso, o arquivo **modelspec.cc**, que foi alterado para incluir uma chamada para o novo modelo, agora deve conter um **#include** para o arquivo com o cabeçalho (arquivo com extensão **.h**) da classe criada.

### 3.2.2 *BonnMotion*

#### A Ferramenta

BonnMotion é uma ferramenta para criar cenários de mobilidade, desenvolvido na plataforma Java, na *University of Bonn* da Alemanha. Esta ferramenta contém um módulo para visualizar os cenários gerados, que é o Visplot. Esse visualizador cria um arquivo contendo todas as posições dos nós em momentos de alterações das suas posições e velocidades. Este arquivo pode ser utilizado pela ferramenta Gnuplot para que seja visualizada a mobilidade dos nós. Os dados gerados também podem ser exportados para os simuladores: NS-2 e GloMoSim. Atualmente existem quatro modelos de mobilidade implementados pelo BonnMotion: *Manhattan Grid*, *Waypoint*, *Gauss-Markov*, *Reference Point*.

O BonnMotion possui um arquivo com extensão **.bat** que facilita sua execução. A execução da ferramenta ocorre pela seguinte linha de comando:

```
bm -f <cenario> <parametros do cenario>
```

Parâmetros importantes comuns a todos os modelos implementados:

- Número de nós: -n
- Duração do cenário (segundos): -d
- Número de segundos no início do cenário que são desprezados: -i
- Tamanho da área de simulação: -x e -y
- Valor inicial da semente para o gerador de números aleatórios: -R

Exemplo de utilização da ferramenta:

```
bm -f cenario RandomWaypoint -n 100 -d 900 -i 3600
```

O comando acima cria um cenário usando o modelo Waypoint com 100 nós e 900 segundos de duração. Uma fase inicial de 3600 segundos é desprezada.

### Saída Gerada

Como resultado são gerados dois arquivos: um com a extensão ".params", que contém o conjunto dos parâmetros utilizados na simulação e outro arquivo com extensão ".movements.gz" que contém os movimentos realizados pelos nós.

A saída gerada é interpretada pelo Visplot, que pode ser utilizado pelo seguinte comando:

```
bm -ha Visplot -f <cenario>
```

Após a execução do comando acima o Visplot irá criar um arquivo (cenario.visplot) que pode ser interpretado pelo Gnuplot para a visualização da movimentação.

A saída gerada também pode ser traduzida para o formato do NS-2 ou do GloMoSim através dos comandos:

Para NS-2:

```
bm -ha NSFile -f <cenario>
```

Para GloMoSim:

```
bm -ha GlomoFile -f <cenario>
```

Após a execução dos comandos acima, o NSFile irá criar dois arquivos (cenario.ns\_params e cenário.ns\_movements) que podem ser interpretados pelo NS-2. O GlomoFile, por sua vez, também irá criar dois arquivos (cenário.glomo\_nodes e cenário.glomo\_mobility) que podem ser interpretados pelo GloMoSim.

### Visualizando Ajuda para Modelos Implementados

Para saber quais são os possíveis parâmetros que podem ser fornecidos para um modelo de mobilidade implementado pelo BonnMotion deve-se utilizar o seguinte comando:

Nesse capítulo foram descritas as principais ferramentas de simulação de redes sem fio e de geração de cenários de mobilidade para as redes móveis *ad hoc*.

## Capítulo 4

# Pesquisas Atuais

Como foi descrito no capítulo 2, a mobilidade influencia o comportamento dos componentes e aplicações das redes móveis *ad hoc*. Como esta área de estudo é recente, têm-se ainda, muita investigação a ser feita sobre o impacto da mobilidade no desempenho desses sistemas. Dentro desse contexto, a seguir, serão descritos alguns tópicos de intensa investigação científica:

### **Protocolos de Acesso ao Meio**

Atualmente as redes *ad hoc*, geralmente, utilizam o protocolo de acesso ao meio do padrão IEEE 802.11, porém novas propostas têm surgido visando aumentar a capacidade do canal. Estudos de protocolos de acesso ao meio específicos para redes *ad hoc* vem sendo feitos, levando em consideração a mobilidade dos nós [38], além do problema de terminais escondidos e expostos.

### **Protocolos de Roteamento**

Em geral as propostas existentes na literatura quanto a protocolos de roteamento para redes *ad hoc* supõem algum perfil de mobilidade, porém não são avaliadas em diferentes cenários onde diferentes padrões de movimentação existem. Uma área de oportunidade para novos estudos é justamente o impacto dos perfis de mobilidade nestes protocolos, onde pode ser avaliado o desempenho de cada um deles sob ambientes de movimentação dinâmicos. Além disso, avaliar esses protocolos integrando a mobilidade com a qualidade de serviço (QoS) requisitada ou oferecida e ainda, com soluções de segurança seria uma interessante avaliação.

### **TCP *Westwood***

Uma característica do ambiente sem fio é a alta taxa de erros em comparação com redes cabeadas. Esta alta taxa de erros impacta o desempenho do TCP, que interpreta uma perda de pacote como sinal de congestionamento, fato este que motivou o desenvolvimento do TCP *Westwood*.

Com a movimentação dos nós, as rotas tornam-se extremamente dinâmicas, fato este que influencia o desempenho do TCP. Identifica-se como importante o estudo do impacto da mobilidade neste contexto.

## Particionamento da Rede

Em redes *ad hoc* a topologia é extremamente dinâmica. Em casos de mobilidade em grupo existe o efeito do particionamento da topologia da rede, quando formam-se grupos isolados entre si, fato este que prejudica uma série de serviços oferecidos pela rede.

Através do estabelecimento de mecanismos de predição do particionamento cria-se a possibilidade de realizar ações que visam manter os serviços operacionais, como por exemplo, a réplica de algum servidor ou base de dados em nós que permanecerão nos grupos de interesse. Um dos problemas desta abordagem é identificar os grupos e prever qual nó estará em qual grupo, o que continua sendo objeto de estudos na área.

## Utilização de Mecanismos de Segurança

A área de segurança independente da tecnologia ou tipo de rede sendo utilizados, sempre teve, e provavelmente sempre terá, uma importância muito grande devido a necessidade de garantir a integridade, autenticidade e disponibilidade de informações e serviços. A pesquisa e o desenvolvimento de técnicas e soluções nessa área é intensa e vasta, podendo seus conceitos, normalmente, serem aplicados diretamente nas mais diversas tecnologias de redes.

Porém, quando se estudam as questões de segurança em uma rede *ad hoc*, logo se percebe grandes diferenças e empecilhos que impedem a aplicação das técnicas de segurança desenvolvidos para outros tipos de rede.

É nesse contexto que surgem grandes desafios, como por exemplo o de implementar os conceitos de um Sistema de Detecção de Intrusão (IDS - *Intrusion Detection System*) e conceitos de autenticação e tudo que envolve a garantia de integridade, autenticidade e disponibilidade dos dados que trafegam através de uma rede *ad hoc*. Com isso, prover segurança para as aplicações e subsistemas das redes móveis *ad hoc* também é um esforço da comunidade científica atualmente.

Além dos tópicos descritos também podem ser citadas como pesquisas atuais: o desenvolvimento de novos modelos de mobilidade, o suporte à descoberta de serviços, a integração das redes *ad hoc* com a Internet e a utilização de mecanismos de QoS, incluindo o uso da predição de mobilidade e particionamento da rede para a reserva de recursos.

## Capítulo 5

# Considerações Finais

A diversidade de cenários onde se usa uma rede *ad hoc* é muito grande, com isso, desenvolver um modelo de mobilidade que represente todos esses cenários não é uma tarefa fácil, até mesmo porque não se tem o padrão de movimentação dos usuários desses cenários. Como foi descrito nesse trabalho, existem vários modelos de mobilidade que buscam essa representação do movimento, os quais são usados para avaliação de desempenho de aplicações e sistemas de comunicação em redes *ad hoc*, principalmente em algoritmos e protocolos de roteamento. Entretanto, características desses modelos os restringem a comportamentos de movimentação específicos, ou simplesmente não representam a realidade. Com isso, deve-se ter critérios na escolha do modelo de mobilidade, caso contrário pode-se ter uma avaliação dos sistemas não condizente com a realidade, como é mostrado em [4] e [18]. Assim sendo, se faz necessário o desenvolvimento de novos modelos de mobilidade, tanto individual, quanto em grupo.

Além disso, a mobilidade influencia o funcionamento das aplicações e sub-sistemas das redes móveis *ad hoc*. Dessa maneira, os estudos atuais neste campo focam os tópicos apresentados no capítulo 4, de forma que esta é uma área de pesquisas aberta e com muitas oportunidades de estudo e geração de resultados.

# Bibliografia

- [1] IEEE P802.1: The Working Group for Wireless LANs. <http://grouper.ieee.org/groups/802/11/index.html>.
- [2] S. Corson and J. Marker. Mobile Ad hoc Networking (MANET). *RFC 2501*, jan 1999.
- [3] C. M. Cordeiro and D. P. Agrawal. Mobile Ad hoc Networking. *OBR Research Center for Distributed and Mobile Computing, ECECS*, 2002.
- [4] V. Davies. Evaluating Mobility Models within an Ad hoc Network. Master's thesis, Colorado School of Mines Los Angeles, USA, 2000.
- [5] Mobile Ad-hoc Networks (manet) Charter. <http://www.ietf.org/html.charters/manet-charter.html>.
- [6] C. Perkins, E. Belding-Royer, and S. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. *RFC 3561*, july 2003.
- [7] D. Câmara. Estudo de Algoritmos de Roteamento para Redes Móveis Ad Hoc. Master's thesis, Departamento de Ciência da Computação / UFMG, Brasil, 2000.
- [8] J. Yan S. R. Das, R. Castaneda and R. Sengupta. Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks. In *7th International Conference on Computer Communications and Networks (IC3N)*, pages 153–161, oct 1998.
- [9] G. Pei. *Scalable Routing Strategies for Large Ad hoc Wireless Networks*. PhD thesis, Department of Computer Science, University of California, Los Angeles, USA, 2000.
- [10] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM*, volume 24, pages 234–244, oct 1994.
- [11] C. E. Perkins and E. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, feb 1999.
- [12] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad hoc Wireless Networks. *Mobile Computing*, 1996.
- [13] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of IEEE INFOCOM'97*, pages 1405–1413, apr 1997.
- [14] Z. J. Haas and M. R Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In *ACM SIGCOMM*, pages 167–177, 1998.

- [15] V. R. Syrotiuk S. Basagni, I. Chlamtac and B. A. Woodward. A distance Routing Effect Algorithm for Mobility (DREAM). In *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 98)*, pages 76–84, 1998.
- [16] Y. B. Ko and N H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 98)*, pages 66–78, 1998.
- [17] W. W. Su. *Motion Prediction in Mobile/Wireless Networks*. PhD thesis, Department of Computer Science, University of California, Los Angeles, USA, 2000.
- [18] G. Pei X. Hong, M. Gerla and C. Chiang. A Group Mobility Model for Ad hoc Wireless Networks. In *Proc. of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, aug 1999.
- [19] S. Yi, P. Naldurg, and R. Kravets. Security-Aware Ad-Hoc Routing for Wireless Networks. August 2001.
- [20] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad Hoc Networks. *MobiCom'2000*, August 2000.
- [21] B. Dahill, B. N. Levine, E. Royer, and C. Shields. A Secure Routing Protocol for Ad Hoc Networks. August 2001.
- [22] Y. Hu, A. Perrig, and D. B Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. *Mobicom*, 2002.
- [23] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1991.
- [24] Averill M. Law and W. David Kelton. *Simulation and Modeling Analysis*. McGraw Hill, 2001.
- [25] C. Bettstetter. Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects. *ACM Mobile Computing and Communications Review*, 5, 2001.
- [26] M. M. Zonoozi and P. Dassanayake. User Mobility Modeling and Characterization of Mobility Patterns. *IEEE Journal on Selected Areas in Communications*, 15(7), sep 1997.
- [27] R. Sinha K. Dantu, S. Kapadia and A.Helmy. Modeling of Mobility-Induced Losses in MANETs (MILMAN). *ACM SIGCOMM Computer Communications Review*, 32(3):30–30, jul 2002.
- [28] K. H. Wang and B. Li. Group Mobility and Partition Prediction in Wireless Ad-hoc Networks. In *Proceedings of the IEEE International Conference on Communications 2002*, New York City, NY, USA, apr 2002.
- [29] K. H. Wang and B. Li. Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks. In *Proceedings of IEEE INFOCOM'02*, 2002.
- [30] C. A. V. Campos and L. F. M. de Moraes. Modelos Markovianos de Mobilidade Individual para Redes Móveis Ad hoc. In *Anais do XXI Simpósio Brasileiro de Redes de Computadores (SBRC2003)*, pages 135–150, Natal, RN, Brasil, mai 2003.



- [31] H. Hartenstein C. Bettstetter and X. Péres-Costa. Stochastic Properties of the Random Waypoint Mobility Model: Epoch Length, Direction Distribution, and Cell Change Rate. In *Proc. of the 5th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, Atlanta, GA, sep 2002.
- [32] M. Liu J. Yoon and B. Noble. Random Waypoint Considered Harmful. In *Proc. of the IEEE INFOCOM'03*, 2003.
- [33] C. Chiang. *Wireless Networks Multicasting*. PhD thesis, Department of Computer Science, University of California, Los Angeles, USA, 1998.
- [34] K. Fall and K. Varadhan. *The NSManual*. The VINT Project, jan 2002.
- [35] R. Ahuja K. Tang R. Bagrodia L. Bajaj, M. Takai and M. Gerla. *GloMoSim: A Scalable Network Simulation Environment*. Computer Science Department, University of California, USA.
- [36] L. Qiming. The Scenario Generator: a tool to generate MANET mobility scenarios for NS-2. <http://www.comp.nus.edu.sg/liqm/scengen/>.
- [37] NS-2 and Adhockey. <http://www.monarch.cs.rice.edu/ns-faq/faq.html>.
- [38] Chris Barrett, Martin Drozda, Achla Marathe, and Madhav V. Marathe. Characterizing the Interaction Between Routing and MAC Protocols in Ad-hoc Networks. In *The ACM Symposium on Mobile Adhoc Networking and Computing (MOBIHOC 2002)*, Lausanne, Switzerland, June 9–11 2002.