



INSTITUTO TECNOLÓGICO DE AERONÁUTICA  
CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq



PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC

# **ANÁLISE DE MÉTODOS PARA IDENTIFICAÇÃO E VISUALIZAÇÃO DE COMUNIDADES EM REDES MÓVEIS**

**Felipe Vieira Frujeri**

**RELATÓRIO FINAL DE ATIVIDADES**

Orientador: Carlos Henrique Costa Ribeiro  
Coorientadora: Luciana Babberg Abiuzy

**AGOSTO / 2013**



INSTITUTO TECNOLÓGICO DE AERONÁUTICA  
CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq



PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC

## Relatório Final de Atividades

# ANÁLISE DE MÉTODOS PARA IDENTIFICAÇÃO E VISUALIZAÇÃO DE COMUNIDADES EM REDES MÓVEIS

São José dos Campos, \_\_\_\_ / \_\_\_\_ / \_\_\_\_

Nome do aluno	
Assinatura do aluno	

Nome do orientador	
Assinatura do orientador	

**INSTITUTO TECNOLÓGICO DE AERONÁUTICA**  
**PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA -**  
**PIBIC**

Formulário de Aprovação de Relatório pelo Orientador

**Relatório:**      ☐ Rel. Parcial      ☐ Rel. Final

---

**1- CONSIDERO O RELATÓRIO APROVADO COM BASE NOS SEGUINTE**  
**ASPECTOS**

---

---

**2- APRECIÇÕES DO ORIENTADOR SOBRE O DESEMPENHO DO BOLSISTA**  
**NA EXECUÇÃO DO TRABALHO DE INICIAÇÃO CIENTÍFICA**

---

---

**Local e data:**

**Assinatura do Orientador:**

# INDICE

<b>RESUMO DO PLANO INICIAL .....</b>	<b>5</b>
<b>RESUMO DAS ATIVIDADES REALIZADAS .....</b>	<b>6</b>
<b>DESCRIÇÃO DO PROBLEMA.....</b>	<b>8</b>
<b>RESULTADOS OBTIDOS .....</b>	<b>10</b>
REDES COMPLEXAS.....	10
▪ Métricas relacionadas à conectividade ( <i>grau</i> ).....	11
▪ Métricas relacionadas a ciclos.....	11
▪ Métricas relacionadas à distância .....	12
▪ Modelo de redes aleatórias ( <i>random graph</i> ) de <i>Ērdos-Rényi</i> .....	13
▪ Modelo <i>small-world</i> de <i>Wattz-Strogatz</i> :.....	17
▪ Modelo livre de escala ( <i>scale-free</i> ) de <i>Barabási-Albert</i> e generalizações:.....	20
REDES MÓVEIS .....	23
▪ Visão Geral ( <i>MANETs</i> ).....	23
▪ Protocolos de roteamento.....	25
MODELOS DE MOBILIDADE.....	27
▪ <i>Random Walk</i> .....	27
▪ <i>Random Waypoint</i> .....	28
▪ <i>Street Random Waypoint (STRAW)</i> .....	29
▪ <i>Direção aleatória (Random Direction)</i> .....	29
▪ Modelo <i>Markoviano</i> de <i>Percurso Aleatório - MPA</i> .....	30
▪ Modelo <i>Smooth</i> .....	30
▪ <i>Gauss-Markov</i> .....	30
▪ Área de simulação sem fronteira ( <i>Boundless Simulation Area</i> ).....	31
▪ Modelo de grupo baseado num ponto de referência – <i>MGPR</i> .....	33
▪ Modelo de Grupo Uniforme .....	33
SIMULADOR <i>JIST/SWANS</i> .....	34
▪ Visão geral .....	34
▪ Modelos implementados.....	34
▪ Arquivos de <i>Trace</i> – parâmetros.....	35
▪ Ambiente de simulação utilizando o protocolo <i>AODV</i> .....	36
COMUNIDADES.....	38
▪ Detecção de Comunidades.....	38
▪ <i>K-means</i> .....	41
▪ <i>Betweenness (Intermediação)</i> .....	41
▪ Modularidade .....	42
▪ Otimização espectral de modularidade.....	43
▪ <i>FastGreedy (Clauset, Newman and Moore)</i> .....	45
▪ <i>Walktrap - Random Walks (caminhadas aleatória)</i> .....	45
▪ Modelo de <i>Potts</i> .....	46
SIMULAÇÃO E ANÁLISE DOS MÉTODOS.....	47
▪ Visualização das redes móveis.....	47
▪ <i>Network Animator</i> .....	49
▪ Configuração dos experimentos.....	51
▪ Análise qualitativa .....	52
▪ Análise quantitativa .....	57
<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>63</b>
CONTRIBUIÇÕES E CONCLUSÕES .....	63
TRABALHOS FUTUROS .....	63
<b>AGRADECIMENTOS.....</b>	<b>64</b>
<b>BIBLIOGRAFIA .....</b>	<b>65</b>

## RESUMO DO PLANO INICIAL

A comunicação móvel (sem-fio) tem crescido cada vez mais na sociedade nos últimos anos, viabilizando atividades importantes e trazendo benefícios ao meio onde atua. Dentro desse contexto de redes sem-fio, as chamadas Redes *Ad hoc* Móveis (MANETs) *Mobile Ad hoc Networks* vêm tendo um crescimento evidente nos últimos anos e sendo motivação para vários estudos.

Um desses estudos é a identificação de comunidades (conjuntos de nós densamente ligados), que é importante para a definição ou adaptação de protocolos de comunicação, visto que, no caso das MANETs, a ocorrência de variações espaciais na densidade do tráfego de informações é frequente.

Diversos métodos têm sido propostos para identificar comunidades em redes estáticas ou definir modelos de mobilidade baseados no conceito de comunidade. Entretanto, poucos estudos vêm buscando avaliar a dinâmica de formação dessas comunidades em redes móveis.

Nesse sentido, o presente estudo teve como objetivo para contribuir com o tema, avaliar quantitativamente (métricas) e qualitativamente (visualização) a dinâmica de formação de comunidades de nós móveis, a partir de modelos de mobilidade clássicos e baseados no conceito de comunidade (*community-based mobility models*).

Para isso, planejou-se uma revisão bibliográfica básica sobre redes complexas e redes móveis com enfoque em MANETs para assimilação de definições fundamentais e métricas básicas. Seguida de uma revisão de algoritmos para determinação de comunidades e modelos de mobilidade. Na fase seguinte, planejou-se a familiarização com algum simulador de mobilidade, possivelmente JiST/SWANS ou CONNY (já utilizados pelo grupo de pesquisa) e a implementação e avaliação de alguns modelos de mobilidade. Por fim, indicou-se a implementação de algoritmos de detecção de comunidades e obtenção de dados quantitativos de detecção cotejados com análise qualitativa (baseada em visualização).

Para execução do projeto, acertaram-se reuniões periódicas com o orientador e co-orientadora, sempre com documentação escrita (log de atividades) para servir de base para preparação dos relatórios.

## RESUMO DAS ATIVIDADES REALIZADAS

Como previsto no projeto, iniciou-se por uma revisão bibliográfica sobre redes complexas, com o objetivo de se familiarizar com os conceitos fundamentais e as métricas básicas. As preocupações centraram-se em absorver as características topológicas básicas dos principais modelos de redes existentes desde sua origem, além de algumas aplicações em redes reais, como redes biológicas (neurais, reações metabólicas e genéticas), Internet e outras.

Relembrou-se os conceitos básicos da teoria dos grafos e seus termos correspondentes nas redes complexas, como nós (vértices), ligações (arestas), listas e matriz de adjacências. Além de métricas interessantes para quantificar propriedades das redes complexas, como a conectividade (grau) e sua distribuição, grau médio, caminho mais curto, caminho médio, diâmetro da rede e coeficiente de agregação.

Estudou-se os principais modelos existentes, desde as redes aleatórias de Erdos e Rényi, com sua distribuição tendendo a distribuição de Poisson para grandes quantidades de nós; passando pelos modelos de Pequeno Mundo (*Small-World*) de Watts-Strogatz até as Redes Livre de Escala (*Scale Free*) de Barabási e Albert com algumas generalizações.

Para cada modelo, a topologia da rede foi estudada a partir da dinâmica do seu processo construtivo, através de suas representações e caracterização pelas principais propriedades, como conexidade e variação das métricas já citadas. Para observação mais prática desses conceitos e medidas utilizaram-se dois *softwares*: o Network Workbench 1.0®, onde foram geradas as diversas redes e analisadas pelas diversas ferramentas da plataforma. E o Pajek 3.0®, principalmente para visualização das redes geradas.

Essa fase da revisão, inicialmente programada para consumir dois meses (agosto/setembro) acabou ocorrendo por cerca de três meses, atrasando um pouco a etapa seguinte sobre redes móveis.

A etapa seguinte, como citado, iniciou-se no meio do mês de novembro estendendo-se até dezembro. Nela, foi feita uma revisão bibliográfica sobre redes móveis (sem-fio), passando pelas características das redes infraestruturas e das sem infraestrutura *Ad hoc*. Absorveu-se o principal conceito nessas redes, o de mobilidade,

suas vantagens e desvantagens, bem como suas principais características e problemas (desafios) enfrentados pela computação moderna relacionados a esse tema.

Dentro desses desafios, é importante conhecer a localização e a movimentação de cada nó, dado que a topologia da rede é dinâmica. Essa abstração da topologia é importante para manter rotas com roteamento mais eficientes e controlar o tráfego de informações, economizando bateria, mantendo a qualidade do canal, entre outros. Para manter essas rotas, têm-se a necessidade da criação de vários protocolos e algoritmos de roteamento eficientes. E para prever o desempenho desses protocolos e algoritmos de roteamento em redes reais, bem como avaliar sua performance, tem-se os modelos de mobilidade, já que a captura de informações do comportamento real de movimentação dos nós (*trace*) é mais custosa e inviável.

Assim, também fez um levantamento dos principais modelos de mobilidade (tanto individuais quanto em grupo), como *Random Walk*, *Random Waypoint*, *Random Direction*, *Gauss-Markov*, *Boundless Simulation Area* e modelos em grupo baseados num ponto de referência – MGPR –, caracterizando e conhecendo as propriedades definidoras de cada um deles.

No mês de fevereiro, fez-se um apanhado dos principais protocolos de roteamento existentes e classificações (*unicast*, *muticast*, *geocast*). E iniciou-se a utilização do JiST/SWANS, realizando observações acerca dos modelos de mobilidade já disponíveis no intuito de ganhar familiaridade com o simulador.

Por fim, estudou-se o conceito de comunidades e de maneira geral alguns algoritmos para detecção das mesmas em redes complexas.

## DESCRIÇÃO DO PROBLEMA

A importância da comunicação sem-fio tem crescido cada vez mais na sociedade. Nos últimos anos assistiu-se a uma vasta proliferação de tecnologia de redes *wireless* (WLAN) tais como *Bluetooth*, *WiMAX*, entre outras. É grande o número de usuários que carregam consigo dispositivos portáteis como aparelhos celulares, notebooks e *tablets* que oferecem uma solução atraente de rápida e simples comunicação sem fio, sem problemas associados ao cabeamento. Entretanto, a maioria desses dispositivos não se comunica entre si, e essa característica vem abrindo um vasto campo de pesquisas denominado redes móveis *Ad hoc* (MANETs – *Mobile Ad hoc Network*), em que tecnologias buscam permitir essa comunicação e interação entre os dispositivos móveis de maneira independente de uma unidade central [1].

Dentro desse contexto de redes móveis, existem diversas dificuldades atreladas à principal característica dessas redes – a mobilidade – que traz consigo uma série de desafios vinculados à limitação de recursos nos dispositivos da rede, tais como limitações de comunicação, armazenamento, processamento e energia. E para contornar essas restrições buscam-se maneiras mais eficientes de gerenciamento de rotas, protocolos e algoritmos de roteamento, com diminuição do tráfego da rede, entre outras soluções melhor exploradas numa seção seguinte.

Para que se possa desenvolver maior eficiência em algoritmos, protocolos e gerenciamento de pacotes são necessários modelos que permitam simular a mobilidade dos dispositivos. Esses modelos encontram uma dificuldade na abstração da topologia da rede, pelo seu caráter dinâmico, não conseguindo prever ou descrever a localização e movimentação dos dispositivos.

Para tentar auxiliar na definição ou adaptação desses protocolos de comunicação, especialmente em MANETs nas quais a ocorrência de variações espaciais na densidade do tráfego de informações é frequente, o presente estudo propõe um caminho de investigação ainda pouco explorado que é o da dinâmica de formação de comunidades, com identificação e visualização das mesmas, buscando uma análise quantitativa (métricas) e qualitativa (visualização) dessa dinâmica a partir de modelos de mobilidade clássicos e baseados no conceito de comunidade (*community-based mobility models*).



O estudo, entretanto, deve se iniciar pelas redes estáticas, nas quais vários métodos já foram propostos para identificar comunidades. Além dos principais conceitos e métricas que surgiram nas denominadas redes complexas.

Em seguida, deve-se buscar estender os conceitos às redes móveis e modelos de mobilidade e, através dos simuladores de mobilidade, implementar esses modelos e algoritmos de detecção de comunidades, para que se possa obter dados quantitativos para análise proposta.

## RESULTADOS OBTIDOS

### REDES COMPLEXAS

Atualmente, a teoria de redes complexas tem sido usada para modelagem e caracterização de sistemas complexos naturais ou artificiais. Essa teoria nasceu da aplicação de medidas desenvolvidas pela teoria dos grafos, mecânica estatística, sistemas lineares e sistemas complexos e tem aplicações nas mais diversas áreas de pesquisa, como física, química, matemática, biologia, medicina, computação, sociologia, linguística, engenharia telecomunicações e astronomia [2], [3].

Suas origens remontam a trabalhos de percolação (filtração) e grafos aleatórios por Flory, Rapoport e Erdős-Rényi, sendo que a teoria dos grafos teve início com a representação de Leonhard Euler para resolver o problema das “Sete Pontes de Königsberg” [3].

Os componentes fundamentais da estrutura das redes complexas são: um conjunto de **nós** (vértices), que são ligados por **ligações** (arestas, conexões), devido a algum tipo de interação [4].

Matematicamente:  $R = (N, A)$ , em que  $N$  é o conjunto de  $N$  vértices:

$N = \{n_1, n_2, \dots, n_n\}$  e  $A$  é o conjunto de  $m$  arestas:  $A = \{a_1, a_2, \dots, a_m\}$ .

Essas redes também podem ser classificadas em orientadas (ou dirigidas), quando existe um sentido na ligação e não-dirigidas (como nos principais modelos utilizados nesse trabalho), ou ainda em estáticas (quando não há variação do número de vértices, arestas ou configuração das ligações) e dinâmicas (quando é possível modelar seu crescimento pela análise da variação de sua estrutura no tempo [3]. A cada aresta também se pode associar um peso (em caso onde existe diferenças de intensidade nas ligações, nesse caso surge mais um conjunto  $W = \{w_1, w_2, \dots, w_m\}$ , e a rede torna-se mais geral:  $R = (N, A, W)$ .

Em termos computacionais, como na teoria dos grafos, as redes podem ser armazenadas através de listas ou matrizes de adjacências.

Em se tratando de redes complexas, é interessante conhecer basicamente os principais conceitos, principais métricas, formas de representação, caracterização, classificação e modelagem (processos construtivos). No presente estudo, a compreensão da estrutura das ligações entre os dispositivos móveis são interessantes

para detecção das comunidades e determinação de aspectos topológicos que possam influenciar os processos dinâmicos das mesmas.

Os conceitos serão apresentados na medida em que for surgindo a necessidade de explorá-los. Já algumas métricas básicas para caracterização da estrutura de redes são apresentadas abaixo:

▪ **Métricas relacionadas à conectividade (grau)**

- a **conectividade** (grau)  $k_i$  é igual ao número de ligações estabelecidas por um nó (número de “vizinhos” de um vértice). A **conectividade média (grau médio)** -  $z = \langle k \rangle$  - é a média do número de conexões estabelecidas entre os vértices.

$$z = \langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i$$

- **distribuição de conectividades**: representa a probabilidade de um vértice escolhido aleatoriamente ter conectividade  $k$ . Através dela é possível caracterizar diversos tipos de redes, assim como determinar se a configuração de conexões é definida de forma aleatória ou por algum padrão (lei de formação).

$$P(k) = \frac{n(k)}{N}$$

em que  $n(k)$  é o número de vértices com grau  $k$

- **componentes fracamente conectados**: são subgrafos que apresentam apenas um vértice se conectando a outros componente da rede. Esses tipo de subgrafo é importante, pois se o rompimento dessa ligação pode representar problemas significativos à transmissão de informação na rede. Se a ligação for rompida, diz-se que se têm um **componente isolado** da rede.

▪ **Métricas relacionadas a ciclos**

- **coeficiente de agrupamento ou aglomeração (*clustering coefficient*)**: mede a razão entre o número de arestas entre os vizinhos de um dado vértice, e o número máximo possível de arestas entre esses vizinhos. Essa métrica tenta reproduzir uma propriedade muito comum em redes reais que é a quantidade de *loops* de ordem três (subgrafos formados por três vértices conectados), ela representa a probabilidade de dois nós quaisquer  $a$  e  $b$ , terem um vizinho  $c$  em comum.

$$CC_i = \frac{2e_i}{k_i(k_i - 1)}$$

em que  $e_i$  é o número de vizinhos do vértice  $i$ . Para uma análise global é comum o cálculo do coeficiente de agrupamento médio. Existem outras métricas interessantes relacionadas a ciclos de ordem superior, além de outras medidas relacionadas à tendência dos denominados **hubs** (vértices com maior número de conexões) se conectarem a outros vértices, formando comunidades (coeficiente *rich-club*) [3].

▪ **Métricas relacionadas à distância**

- **comprimento do caminho** entre dois vértices  $i$  e  $j$  é dado pelo número de arestas ao longo deste caminho. E o **menor caminho**  $d_{ij}$  entre dois vértices é dado pela extensão de todos os caminhos que conectam estes vértices cujos comprimentos são mínimos [5]. A importância dessa métrica está na caracterização da estrutura interna das redes e na investigação de efeitos dinâmicos relacionados ao transporte e comunicação, como o das outras métricas relacionadas à distância [6]. A média entre os valores de menor caminho fornece o **menor caminho médio** (caminho característico da rede):

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}$$

e se  $i$  e  $j$  não pertencerem a um mesmo componente conectado,  $d_{ij} = \infty$ . Assim, é interessante para que a soma não divirja que a rede não possua componentes desconectados. Caso haja algum componente desconectado a métrica mais apropriada é a média harmônica do menor caminho médio e medida de eficiência global [3].

O **diâmetro** da rede é o maior dos caminhos mínimos entre dois vértices da rede:

$$\phi = \max_{i,j} d_{ij}$$

Com o passar do tempo, as investigações sobre redes complexas e análises das conexões das mesmas levaram ao desenvolvimento de modelos para reproduzir a estrutura de redes reais.

Esses modelos se iniciaram em 1959, com dois matemáticos húngaros Paul Erdos e Alfred Rényi que consideraram os grafos como objetos estocásticos, sugerindo um processo construtivo baseado em ligações aleatórias. Em seguida, ao se confrontar a modelagem teórica aleatória com redes reais, identificou-se a não correspondência e passaram a surgir novos modelos no sentido de tentar tornar mais verossímeis as topologias das redes geradas pelos processos construtivos [3]. No

presente estudo, são descritos os principais modelos importantes na abstração da topologia das redes, com seu processo construtivo e principais propriedades:

- **Modelo de redes aleatórias (*random graph*) de Ęrdos-Rényi**

- **Processo construtivo**

Essas redes são construídas, iniciando-se com um conjunto de  $N$  vértices totalmente desconectados e a cada passo dois vértices são escolhidos aleatoriamente e conectados com uma probabilidade fixa  $p$ , sendo cada par de vértices considerado apenas uma vez. Assim, todas as ligações possuem a mesma probabilidade de ocorrência, ou seja, a rede gerada tem uma estrutura altamente homogênea [3].

- **Propriedades**

A **distribuição de conectividades** para essas redes, quando  $N$  é grande e a conectividade média é mantida constante, tende à distribuição de Poisson (**Figura 1**). Além disso, o **caminho mínimo médio** é pequeno nessas redes, caindo com o logaritmo do tamanho da rede [3]:

$$l \sim \frac{\ln N}{\ln \langle k \rangle}$$

sendo  $z = \langle k \rangle = \frac{2M}{N} = p(N - 1) \approx pN$ ;  $q$  a probabilidade de conexão,  $N$  o número de nós e  $M$  o número total de arestas.

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-k-1}$$

para  $n \rightarrow \infty \Rightarrow P(k) = \frac{z^k}{k!} e^{-z}$ .

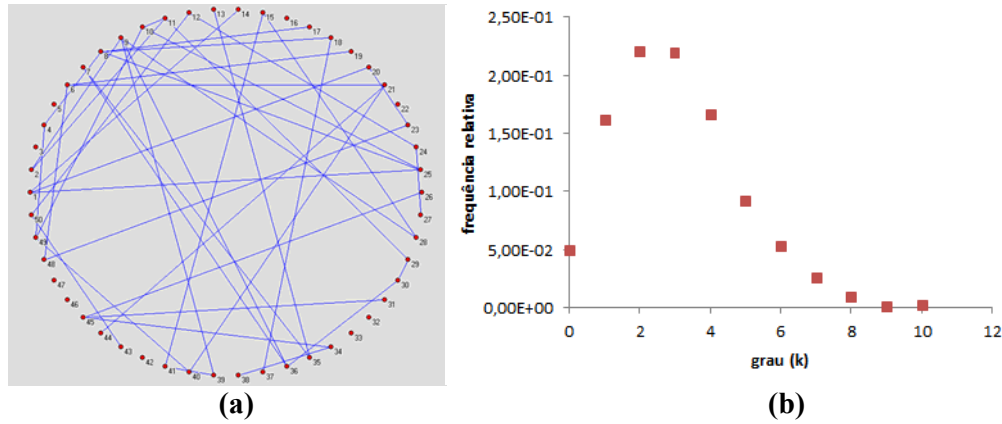
Para ser mantida a propriedade da **conexidade** nesses grafos aleatórios, a probabilidade de conexão  $p(n)$  deve crescer mais rapidamente que a função  $q^{conexidade} = \frac{\log n}{n}$ , ou seja, o grau médio deve crescer mais rapidamente que a função logarítmica do número de nós:

$$\lim_{n \rightarrow \infty} \frac{p(n)n}{\log n} = \lim_{n \rightarrow \infty} \frac{z(n)}{\log n} = +\infty$$

- **Resultados experimentais**

Para visualização de um exemplo de rede aleatória, tomou-se, utilizando o Network Workbench 1.0<sup>®</sup>, uma rede com  $N = 50$  e probabilidade de conexão igual a

$p = 0,033979$  para manter a conectividade. Já para detectar a distribuição de conectividades se aproximar de Poisson, tomou-se  $N = 1000$ .



**Figura 1: (a) exemplo de rede aleatória e respectiva distribuição de conectividades (b) que se aproxima da de Poisson**

Para verificar as propriedades do grau médio  $z_{exp}$  e do caminho mínimo médio  $l$  entre dois vértices na rede, experimentalmente, produziram-se, redes variando-se o número de nós entre 500 e 5000 (com passo de 500) com probabilidade de conexão de  $p = \frac{\log n}{n}$ . Além do baixo coeficiente de agrupamento ( $c$ .  $c$ .) propriedade da conectividade, investigando o número de nós em componentes fracamente conectados (*weakly connected – w.c.*) e número de nós em componentes isolados. Os resultados foram organizados da **Tabela 1**:

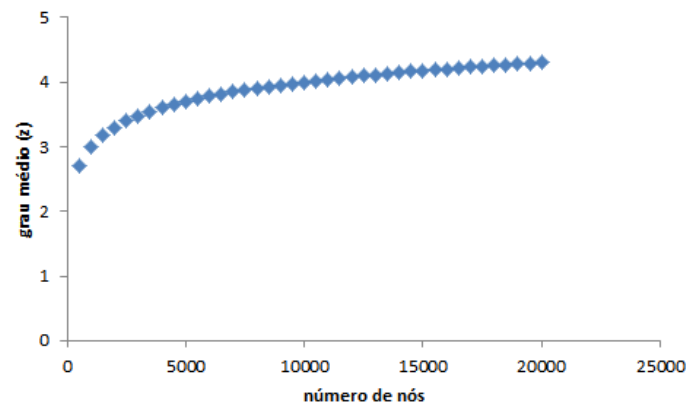
**Tabela 1: verificação experimental das principais propriedades em redes aleatórias**

$n$	$p(n)$	$z_{teo}$	$l_{teo}$	$z_{exp}$	$l_{exp}$	$w.c$	$n^{\circ} isol.$	$c.c$
500	0,005398	2,69897	6,3	2,644	6,3	35	28	0,0045
1000	0,003000	3,000000	6,3	2,986	6,2	54	49	0,00089
1500	0,002117	3,176091	6,3	3,183	6,3	68	63	0,0020
2000	0,001651	3,301030	6,4	3,334	6,3	91	87	0,0025
2500	0,001359	3,397940	6,4	3,379	6,4	91	84	0,0023
3000	0,001159	3,477121	6,4	3,521	6,4	95	88	0,0021
3500	0,001013	3,544068	6,4	3,589	6,4	110	102	0,0013
4000	0,000901	3,602060	6,5	3,636	6,5	109	102	0,00056
4500	0,000812	3,653213	6,5	3,688	6,5	117	109	0,00078
5000	0,000740	3,698970	6,5	3,730	6,5	124	118	0,00087

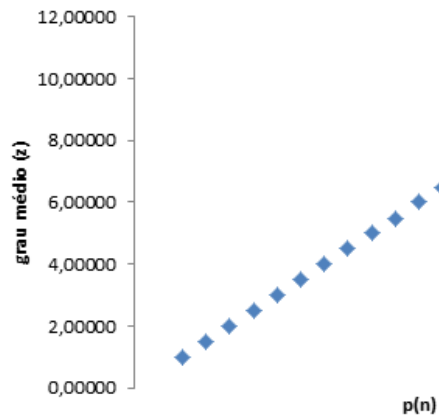
Percebe-se de fato que o grau médio e o menor caminho médio (baixo) acompanha a previsão teórica. Quanto ao limiar de conectividade, pode-se perceber que

o percentual de nós em componentes isolados é baixo (cerca de 2%), o que se justifica pela probabilidade de conexão exatamente no limiar. É interessante observar também o baixo coeficiente de agrupamento, justificado pela aleatoriedade nas conexões.

Ainda evolução logarítmica do grau médio com o número de nós e sua relação linear com a probabilidade de conexão linear com a probabilidade de conexão (para  $N = 10000$  fixo) e o é mostrada na **Figura 2**:



(a)



(b)

**Figura 2: relação logarítmica do grau médio em relação ao número de nós e linear em relação a probabilidade de conexão**

Para melhor verificar o limiar de conexidade, fixou-se o número de nós em  $N = 10000$ , correspondente a uma probabilidade no limiar de  $p(n) = 0,0004$  e a um grau médio de  $z = 4$  e geraram-se redes com diversas probabilidades de conexão de 0,0001 a 0,01. Os resultados foram organizados na **Tabela 2**:

**Tabela 2: Verificação do limiar de conexidade**

$n = 10000$			
$p(n)$	$z_{exp}$	$w. c.$	$n^{\circ} isol.$
0,0001000	0,98560	5073	3710
0,0001500	1,46700	2955	2275
0,0002000	1,97040	1668	1395
0,0002500	2,47300	931	813
0,0003000	2,97860	565	511
0,0003500	3,48420	329	310
0,0004000	3,993	187	179
0,0004500	4,50040	116	114
0,0005000	4,99320	68	66
0,0005500	5,49240	42	41
0,0006000	6,00320	30	29
0,0006500	6,49280	20	19
0,0007000	6,99040	12	11
0,0007500	7,49240	7	6
0,0008000	7,99360	2	1
0,0010000	9,97540	0	0

Percebe-se que o número de componentes fracamente conectadas ou mesmo de nós em componentes isolados vai diminuindo com o aumento da probabilidade de conexão e para um valor de 2,5 vezes o limiar de conexidade já não existem mais nós em componentes isoladas, ou seja, a rede está totalmente conectada.

A variação do diâmetro da rede também, como o menor caminho entre dois, também tem uma variação teórica esperada como logarítmica em relação ao número de nós. Para observar essa propriedade, variou-se o número  $N$  de vértices novamente entre 500 e 5000 com as respectivas probabilidades de modo a manter a conexidade da rede. Os Resultados obtidos foram organizados na **Tabela 3**:

**Tabela 3: Variação do diâmetro da rede**

$n$	$z$	$\phi_{teo}$	$\phi_{exp}$
500	1,6989	6	14
1000	3,0000	6	15
1500	3,1761	6	15
2000	3,3010	6	15
2500	3,3979	6	12
3000	3,4771	6	13
3500	3,5441	6	13
4000	3,6021	6	13
4500	3,6532	6	13
5000	3,6990	7	14



Encontrou-se um diâmetro experimental praticamente o dobro do teórico esperado de maneira constante para todas as redes, possivelmente devido a alguma definição diferente para o algoritmo no Network Workbench 1.0<sup>®</sup>. De qualquer maneira a variação logarítmica foi detectada.

Os trabalhos de Erdős e Rényi em 1959 foram mais voltados para a riqueza matemática das redes aleatórias e não tanto para aplicações práticas. Sua importância residiu no fato de que acabou suscitando questões relacionadas à aplicabilidade de seu modelo a redes reais e caso não fosse possível, qual seria a estrutura de conexões em redes reais. O primeiro passo para essa descoberta só foi dado em 1998 por Duncan Watts e Steven Strogatz, que na observação de redes de neurônios do *Caenorhabditis elegans* e da rede de distribuição de energia nos EUA, tiveram a percepção de que havia grande quantidade de *loops* de ordem 3 em relação às redes aleatórias com mesmo número de vértices e arestas. Baseados nessa descoberta propuseram o [3]:

- **Modelo *small-world* de Wattz-Strogatz:**

- **Processo construtivo:**

Essas redes são construídas a partir de uma rede regular, religando as arestas com probabilidade  $p$ .

O modelo visa a obtenção do efeito denominado *small world*, no qual se obtém uma diminuição do menor caminho médio, com alto coeficiente de agrupamento. Isso ocorre devido a distância de separação crescer mais lentamente do que o tamanho da rede [3].

Para isso, inicia-se com uma rede regular formada por  $N$  vértices ligados a  $\kappa$  vizinhos mais próximos em cada direção, totalizando  $2\kappa$  conexões iniciais, sendo  $N \gg \kappa \gg \log(N)$ .

A seguir, cada aresta é aleatoriamente reconectada com uma probabilidade fixa, que introduz o caráter aleatório à rede [7].

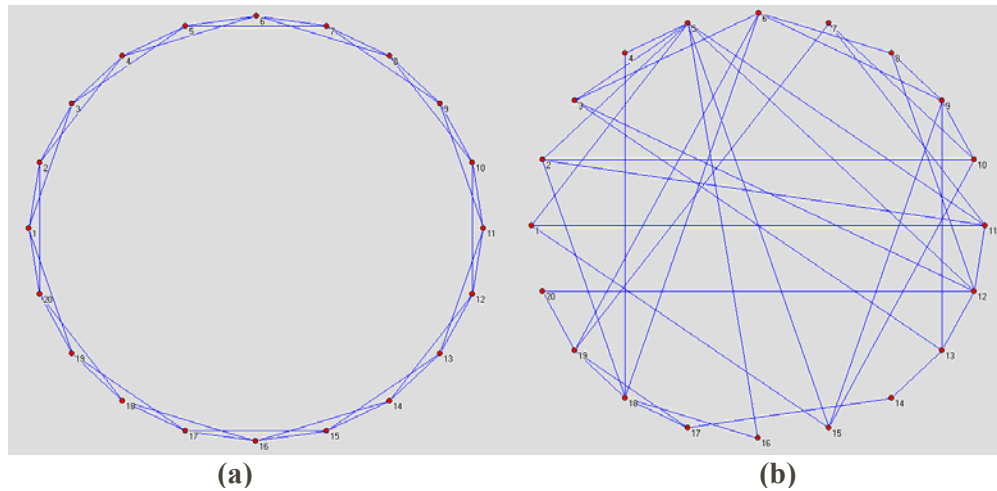
- **Propriedades**

Quando  $p = 0$  a rede é completamente regular, apresentando alta quantidade de *loops* e caminho médio alto, e quando  $p = 1$ , a rede é aleatória, apresentando baixa quantidade de *loops* de ordem três, mas pequeno caminho médio. Portanto, tal modelo se situa entre a completa regularidade e a aleatoriedade. A emergência do regime

*small world* ocorre para  $p > 0,01$ , quando o menor caminho médio converge para o valor encontrado nos grafos aleatórios e a ocorrência de ciclos de ordem três permanece da ordem das redes regulares [3],[7]. À medida que há realocações, o menor caminho médio cai sem que se altere o coeficiente de agrupamento.

### - Resultados experimentais

Para visualização de exemplos de redes completamente aleatórias e completamente regulares no modelo, tomou-se, utilizando o Network Workbench 1.0<sup>®</sup>, uma rede com  $N = 20; \kappa = 2; p = 0$  e em seguida  $N = 20; \kappa = 2; p = 1$ .

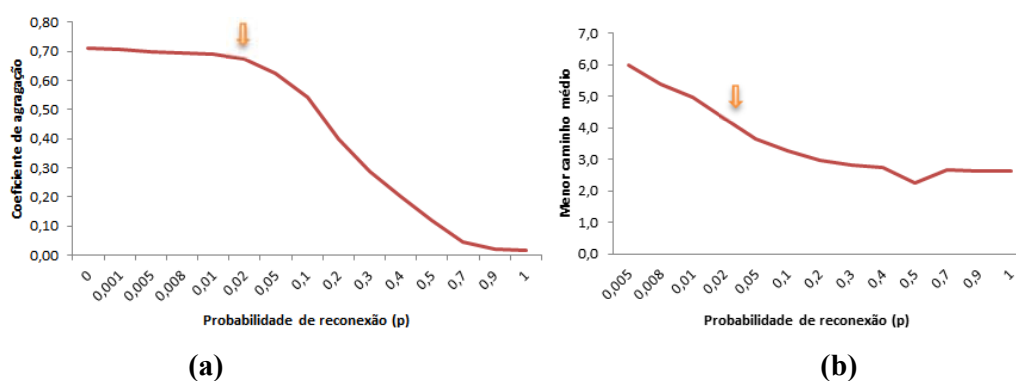


**Figura 3: caracterização da estrutura regular em (a), com não reorganização de arestas. E o segundo caso (b) onde tem-se a recombinação para todas as ligações, com característica aleatória.**

Para verificação do limiar  $p > 0,01$  em que o modelo ganha a característica de *small-world*, fixou-se o número de vértices em  $N = 1000$  e o número de vizinhos iniciais  $\kappa$  e variou-se a probabilidade de reconexão de 0 a 1. Os resultados foram organizados na **Tabela 4** e na **Figura 4**:

**Tabela 4: Verificação do limiar para geração do efeito de *small world***

$n$	$\kappa$	$p$	Menor caminho médio	Coeficiente de agregação médio
1000	10	0	25,5	0,71
1000	10	0,001	8,6	0,71
1000	10	0,005	6,0	0,70
1000	10	0,008	5,4	0,69
1000	10	0,01	5,0	0,69
1000	10	0,02	4,3	0,67
1000	10	0,05	3,7	0,63
1000	10	0,1	3,3	0,54
1000	10	0,2	3,0	0,40
1000	10	0,3	2,8	0,29
1000	10	0,4	2,8	0,20
1000	10	0,5	2,3	0,12
1000	10	0,7	2,7	0,05
1000	10	0,9	2,6	0,02
1000	10	1	2,6	0,02



**Figura 4: (a) coeficiente de agregação e (b) menor caminho médio em função da probabilidade de reconexão  $p$**

De fato, comprovou-se a existência de um limiar para o efeito de *small-world*; entretanto, para uma probabilidade  $p > 0,025$ , ligeiramente maior que o previsto teoricamente, mas em concordância na ordem de grandeza.

Em 1999, dois pesquisadores da Universidade de Notre Dame, Estados Unidos, Albert-László Barabási e Reka Albert, decidiram verificar se o efeito de *small world* estava presente na *World Wide Web* e através do uso de um *web crawler*, conseguiram mapear a topologia de conexões entre as páginas e descobriram não só

um caminho mínimo médio por volta de  $l \approx 11$ , mas também a emergência de um novo tipo de distribuição de conexões do tipo **livre de escala (scale-free)** [8]:

▪ **Modelo livre de escala (*scale-free*) de Barabási-Albert e generalizações:**

**- Processo construtivo:**

A principal ideia para sua construção reside no modo de adição de novos vértices, no qual vértices mais conectados tendem a receber mais conexões – “o rico fica mais rico” [3].

Dois princípios regem seu processo construtivo:

1. *Crescimento*: iniciando-se com um pequeno número de vértice  $N_0$ , a cada passo é adicionado um vértice com  $m$  ( $m \leq N_0$ ) arestas que se conectam com vértices já presentes na rede.

2. *Ligação preferencial*: o novo vértice, que vai ser adicionado à rede, tende a se conectar com vértices mais conectados, ou seja, a probabilidade de um vértice  $j$ , presente na rede, ser escolhido é proporcional à sua conectividade,

$$P_{i \rightarrow j}(n+1) = \frac{k_j(n)}{\sum_{u=-N_0+1}^n k_u(n)}$$

Onde  $n$  é o tempo e o número de vértices adicionados à rede.

**- Propriedades:**

A motivação para essa modelagem se deu pela falta de uniformidade nas redes complexas de maneira geral, não respeitando a aleatoriedade de Ęrdos-Rényi. Essa distribuição tem a forma

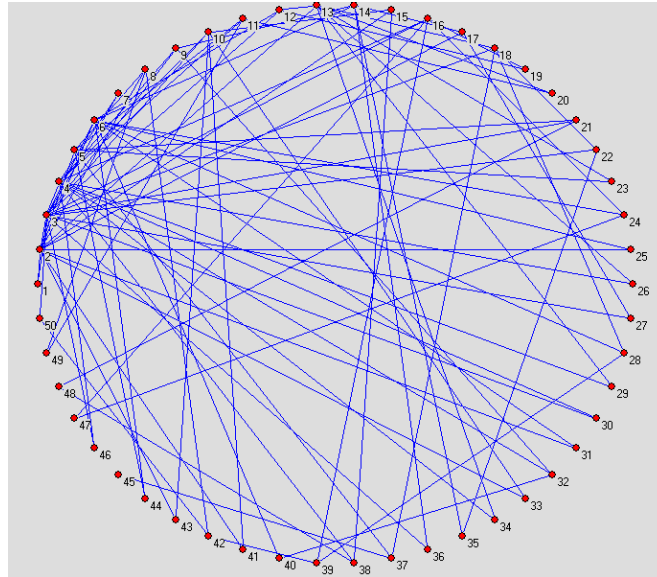
$$P(k) = k^{-\gamma}$$

e é denominada lei de potência de Zipf ou distribuição de Pareto, nela, ao contrário da uniformidade, um número reduzido de vértices é altamente conectado, sendo denominados *hubs*, e uma grande quantidade de vértices são pouco conectados, o que define uma estrutura livre de escala.

Nesse modelo para  $n \rightarrow \infty$ , tem-se  $\gamma \rightarrow 3$ .

### - Resultados experimentais:

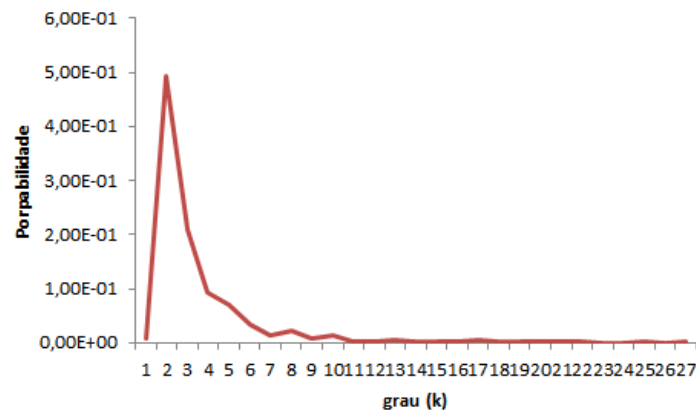
Para visualização de exemplos de redes livre de escala, tomou-se, utilizando o Network Workbench 1.0<sup>®</sup>, uma rede com  $N = 50$  vértices e o número de novas arestas  $m = 2$ :



**Figura 5:** redes livre de escala com 50 vértices e 2 ligações a cada novo vértice

Percebe-se, de maneira indubitável, pela análise da **Figura 5**, o aparecimento de *hubs* nos nós mais antigos (2, 3, 4, 5 e 6).

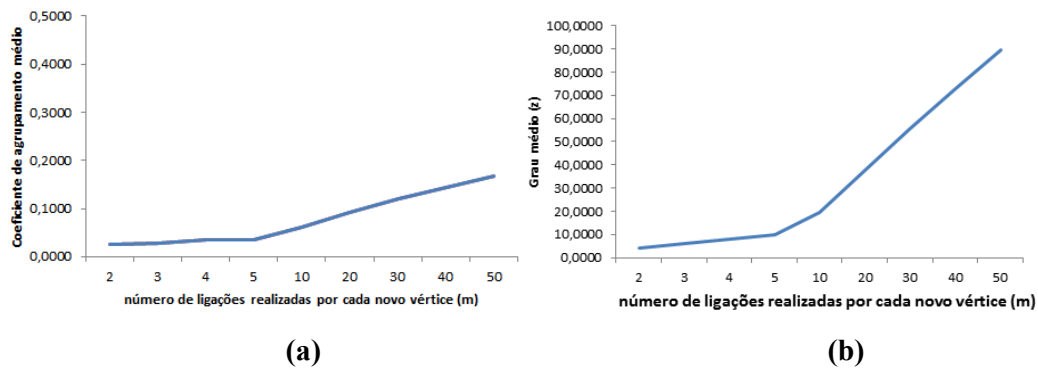
A distribuição de Pareto para as conectividades também pôde ser comprovada. Para isso, tomou-se uma rede com  $N = 1000$ ;  $m = 2$  e os resultados colocados na **Figura 6**:



**Figura 6:** Distribuição de Pareto num modelo livre de escala

Percebe-se a existência de muito nós com grau mais baixo e poucos nós com graus mais elevados (*hubs*). Nesse caso alguns *hubs* chegaram a  $k_{max} = 27$ , sendo um valor limítrofe para essa propriedade um grau  $k > 5$ .

É interessante perceber ainda o baixo coeficiente de agrupamento obtido por essa distribuição, o que pode ser explicado pelo predomínio de ligações com os *hubs*. Para perceber esse baixo coeficiente de agregação, buscamos a curva do coeficiente de agregação em função do número de ligações estabelecido por um novo vértice, cujo aumento proporcionaria uma maior possibilidade de ciclos de ordem 3. Por fim, é interessante perceber o aumento do grau médio, com o aumento linear do  $m$ , o que é representado na **Figura 7**:



**Figura 7: relação entre o coeficiente de agrupamento médio (a) e o grau médio (b) com  $m$**

Alguns autores, como *Amaral et al*, propuseram a generalização desse modelo, buscando outros fatores que poderiam limitar a ligação preferencial, permitindo certa flexibilidade ao valor de  $\gamma$ .

Tais fatores foram:

- *idade dos vértices*: alguns vértices tendem a não receber mais conexões a partir de um certo tempo, mas continuam contribuindo para formação da estrutura da rede
- *custo da adição de novas arestas e capacidade limitada*: em algumas redes, os vértices podem não receber ligações devido ao alto custo que tais conexões podem gerar ou devido à sua capacidade limitada de gerenciar novos *links*
- *filtragem de informações*: as conexões podem se limitar aos vértices mais semelhantes

Além dessa generalização, outros autores propuseram outras formas de adaptabilidade para  $\gamma$ , com diversos outros pesos na probabilidade de ligação de um novo vértice.

Ainda, Bianconi e Barabási propuseram um modelo de ligação preferencial baseado na “popularidade” dos vértices (através de um coeficiente de adaptação), permitindo um modelo onde vértices mais novos podem atrair um grande número de conexões e se tornar mais conectados, como na Internet ocorreu com o *Google*, em apenas dois anos de existência.

Embora a maioria dos modelos de redes reproduza de certa maneira a lei de potência para as conexões e outras propriedades estruturais, como o caminho médio pequeno e alto coeficiente de agregação, eles falham na previsão de estruturas importantes como as comunidades e determinados tipos de subgrafos, assim as leis de formação da maioria das redes reais ainda não foram identificadas.

Assim, faz-se necessário analisar diversas medidas topológicas e tentar inferir os mecanismos dinâmicos determinantes das redes. O que traz de volta ao estudo das redes móveis.

## REDES MÓVEIS

### ▪ Visão Geral (MANETs)

O surgimento das redes sem fio data da década de 1970, quando universidades das ilhas Havaí montaram a rede chamada ALOHAnet, utilizando micro-ondas para se interligaram [9]. Entretanto, apenas nos últimos anos é que as redes móveis vêm se difundindo e despertando interesses. Atualmente, se desenvolve um novo paradigma na computação móvel para lidar com os novos requisitos exigidos por essas em relação às redes tradicionais [10].

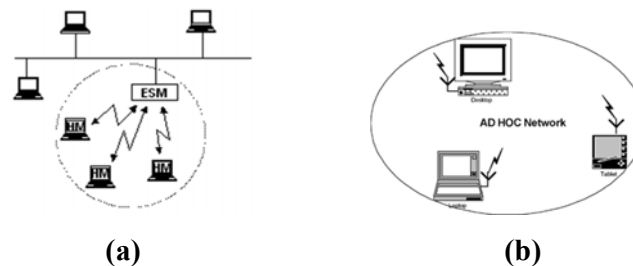
Entre as suas principais vantagens para o usuário, tem-se a **Mobilidade**, característica fundamental dessas redes, que pode ser definida como a capacidade de um terminal móvel continuar em contato contínuo com os recursos da rede; ou seja, nem o sistema, nem as aplicações precisam ser encerrados e reinicializados. Assim, o modo de acesso à rede é pela interface é sem-fio [11].

É interessante diferenciar esse termo de **Portabilidade**, que é a capacidade de um terminal móvel operar a partir de diferentes pontos de conexão, mas perde o contato durante o tempo de mudança do ponto de acesso. Nesse paradigma, ao se mover, as conexões são encerradas e reinicializadas no novo ponto de conexão.

As redes móveis (sem-fio) podem ser divididas em dois grupos com infraestrutura e sem infraestrutura:

- **Redes infraestruturadas:** são as redes tradicionais, as quais são compostas por partes fixas como estações de rádio base (ERB), estações de suporte à mobilidade (ESM) e outras estações fixas (servidores e roteadores). Tem-se como exemplo a rede de telefonia celular e as redes wireless LANs (IEEE 802.11, HIPERLAN), wireless ATM, redes via satélite.

- **Redes sem infraestrutura:** mais comumente conhecidas como redes móveis *Ad-hoc* (*Mobile Ad hoc Networks* - MANETs). Elas são formadas de modo dinâmico como um sistema autônomo pelos nós móveis com enlaces sem fio, que são capazes de trocar informação diretamente entre si sem uma estação-base (ou qualquer outra unidade centralizadora). Como cada nó deve se comunicar apenas com os nós no seu raio de alcance, que é limitado, geralmente as transmissões são ditas multi-saltos e os próprios nós devem agir tanto como roteadores das mensagens dos outros nós como dispositivos terminais, ou ainda ambos simultaneamente, o que dá à rede a característica de **autoconfiguração**. Exemplos que vem surgindo são: a WPAN – IEEE 802.15 (Bluetooth) e a rede de sensores [12], [13], [14], [15].



**Figura 8: (a) rede infraestruturada e (b) rede *Ad hoc* [1]**

Essas características tornam essas redes propícias para aplicações em diversas situações onde não se pode, ou não é adequado instalar uma rede fixa. Como em cenários de campos de batalha, onde aviões, helicópteros, tanques de guerra e militares trocam informações táticas; ou de empresários compartilhando informações numa reunião; de vendedores comunicando-se no centro comercial de uma cidade; de estudantes usando computadores portáteis para participar de uma aula interativa; ou ainda de equipes de resgate e salvamento numa situação de desastre, como furacão, terremoto ou inundação, onde estas precisam se coordenar.

Ou seja, as vantagens fundamentais são rápida instalação, tolerância a falhas (pois vários caminhos podem ser criados) e conectividade (os nós dentro da área de alcance podem trocar informações diretamente).



Entretanto, ao se introduzir essa característica de mobilidade surgem algumas dificuldades e requisitos como localização (encontrar o nó móvel); movimentação dos nós (não necessariamente seguem algum padrão de movimentação, desligamento sem aviso dos nós (o nó pode passar por períodos sem contato com a rede, ou mesmo desligados e reaparecer em algum lugar imprevisível); qualidade do canal (canal sujeito a variações na qualidade, taxas de erro de  $10^{-5}$  bits errados); baixa banda passante; consumo de energia (tráfego de mensagens, que não dizem respeito diretamente ao nó); suporte a QoS (*Quality of Service*) e nós com capacidades e características diferentes.

- **Protocolos de roteamento**

Alguns conceitos importantes que são aproveitados das redes complexas são: o conceito de **nó móvel** de maneira a cunhar os dispositivos móveis na abstração da rede para estudo. E o caminho entre dois nós denominado **rota**.

Uma rota entre dois nós móveis numa rede *Ad hoc* pode ser formada por um ou vários saltos através dos nós da rede. Um dos principais problemas nessas redes é determinar e manter as rotas, já que a mobilidade dos nós pode causar rápidas mudanças topológicas (o que gera uma dificuldade em manter o caminho ótimo para todos os nós da rede). Além da inexistência de uma entidade central (o que gera uma dificuldade em conhecer o estado real da rede), e de que todas as comunicações devem ocorrer através do meio sem-fio (o que acaba por gerar perdas de pacotes com informações de roteamento).

Dai a necessidade da criação de vários **protocolos e algoritmos de roteamento** eficientes.

Para o desenvolvimento de algoritmos e protocolos alguns requisitos devem ser atendidos como o funcionamento correto, simplicidade, robustez, imparcialidade, estabilidade, rápida convergência para a rota ótima, flexibilidade, aceitação de parâmetros de QoS (*Quality of Service*) e independência da tecnologia de rede.

Quanto aos protocolos de roteamento para as Redes *Ad hoc*, tem-se alguns pontos de análise quantitativa: vazão e atraso fim-a-fim dos pacotes de dados, taxa de perdas de pacotes, tempo para aquisição de uma rota, porcentagem de pacotes entregues fora de ordem, eficiência do protocolo (medida pelo número de pacotes de controle necessários pelo protocolo).

Os principais protocolos de roteamento podem ser divididos em:

- *Unicast*:

- Pró-ativos: DSDV, WRP.
- Reativos: AODV, DSR, TORA.
- Híbridos: ZRP.

- *Multicast*:

- B. árvores: MAODV, AMRoute, AMRIS
- B. malhas: ODMRP, CAMP

- *Geocast*: DREAM, LAR

Quanto aos algoritmos, algumas características são desejáveis, como: operar de forma distribuída, não apresentar *loops* de roteamento, operar de acordo com a demanda, modo de operação pró-ativa, segurança, observar períodos de desligamento, suporte a *links* unidirecionais.

Para avaliar o desempenho de um protocolo de roteamento [16] e [17], são necessárias várias análises relacionadas ao alcance de transmissão dos nós, tamanho da rede, conectividade, características de tráfego, tamanho dos *buffers* para o armazenamento das mensagens (capacidade do canal), porcentagem da rede desligada ou em *doze mode*, roteamento baseado em QoS e padrões de movimento dos usuários da rede.

Existem duas maneiras de se representar padrões de movimento de usuários de uma rede móvel [16]:

- através da **captura de informações do comportamento real de movimentação do nó** com uso de registros (*traces*):

Essa forma possibilita uma observação do comportamento real de movimentação dos nós móveis, principalmente quando se tem um grande número de nós durante um longo período. Entretanto, em ambientes dinâmicos como as redes *ad hoc*, essa não é uma tarefa fácil e quando se trata de dados confidenciais há uma dificuldade de acesso aos mesmos.

- através de **modelos de mobilidade**, onde se tenta representar o comportamento de movimentação dos nós sem o uso desses registros de movimentação.

## MODELOS DE MOBILIDADE

Os modelos de mobilidade visam representar o comportamento de movimentação dos dispositivos móveis numa rede *Ad hoc* dado a dificuldade na obtenção de registros de movimentação (*traces*) dos nós móveis [18],[19].

Esses modelos podem auxiliar na avaliação do desempenho de aplicações e sistemas de comunicação; permitindo mensurar, por exemplo, o impacto causado pela mobilidade no funcionamento dos mesmos.

Alguns exemplos de ambientes onde se aplicam os modelos de mobilidade são: nas redes celulares, para estimar o *handoff*, tráfego, atualização de localização dos usuários, *paging*, duração das chamadas, registro; nas *MANETs*, para simular o tráfego oferecido, suporte à descoberta de serviços, implementação de infraestrutura de chave pública, avaliação da perda de pacotes, protocolos de roteamento, predição do particionamento da rede, cobertura de serviços em redes *ad hoc* particionáveis, protocolos de acesso ao meio, gerenciamento de energia das baterias [18], [19].

É importante lembrar ainda que como o comportamento de movimentação é bastante complexo, os modelos não conseguem representar perfeitamente a realidade.

Os modelos de mobilidade possuem duas categorias: modelos para redes de telefonia celular e modelos para redes *Ad hoc*.

Sendo essa última dividida em modelos de mobilidade **individual** e em **grupo**.

Nos modelos de **mobilidade individual**, representa-se o comportamento de um nó móvel de forma independente dos restantes. Esses modelos têm importante aplicação na predição da disponibilidade dos enlaces sem fio, a fim de melhorar, como 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 [18], [19].

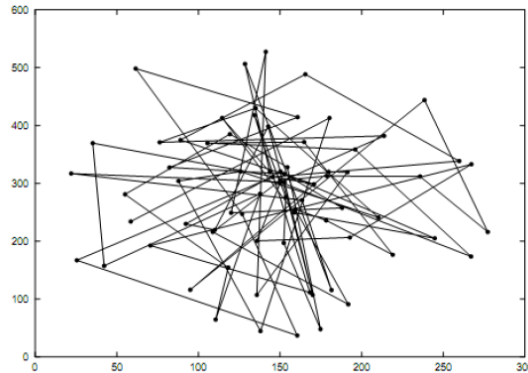
### ▪ *Random Walk*

Nesse modelo, a velocidade e a direção de cada partícula são aleatórias com distribuição uniforme. Possui independência temporal, ou seja, a não existência de memória (as sucessivas posições são independentes das posições anteriores dos nós).

Conforme [20], funciona da seguinte maneira:

- em um intervalo de tempo constante  $\Delta t$ , um nó se move de sua posição atual para outra posição escolhendo aleatoriamente uma direção  $D$  (no intervalo  $(0, 2\pi]$ ) e uma velocidade constante  $V$  (escolhida entre um  $v_{min}$  e  $v_{max}$ ). Ao se alcançar o

destino, escolhe-se novamente uma direção e uma velocidade e o nó se move até esta posição. A cada novo instante de tempo, o valor das variáveis de direção e velocidade não possui nenhuma relação com os valores anteriores, podendo ocorrer movimentos bruscos.

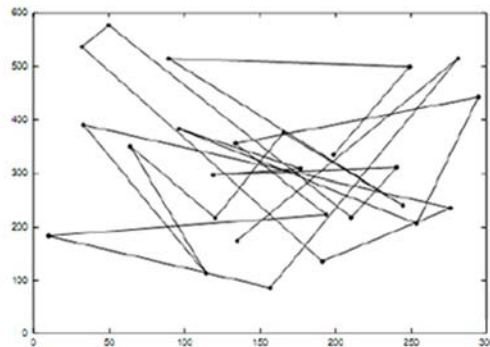


**Figura 9: Exemplo de movimentação gerada no *Random Walk* [20]**

- ***Random Waypoint***

Nesse modelo, inicia-se com um posicionamento aleatório dos nós dentro da área de simulação.

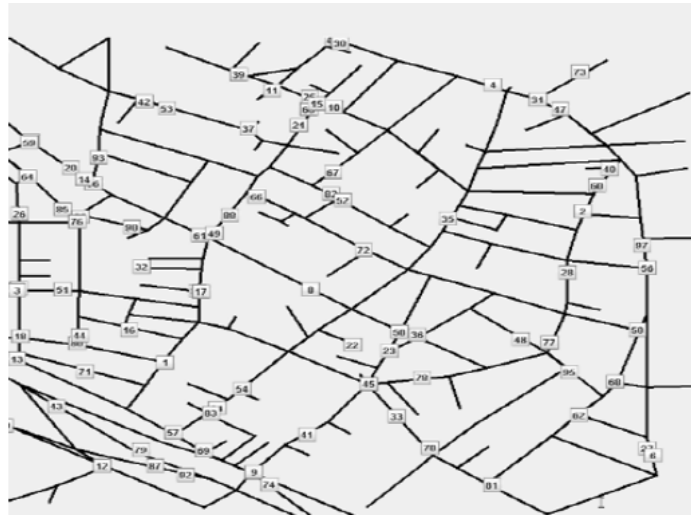
Os nós permanecem nesta posição por um certo tempo aleatório (denominado pausa), e, após esse período, o nó escolhe uma nova posição (aleatoriamente) dentro da área da simulação. A velocidade e direção de cada nó também são aleatórias uniformemente distribuídas entre  $v_{min}$  e  $v_{max}$  e entre  $[0, 2\pi)$ , respectivamente. O nó móvel percorre o caminho até o seu destino com a velocidade constante. Uma vez alcançado o destino, fica estático durante o tempo de pausa e após o término deste período o processo é reiniciado. [20]



**Figura 10: Exemplo de movimentação gerada no *Random Waypoint* [20]**

- **Street Random Waypoint (STRAW)**

É um modelo de mobilidade veicular, baseado nas operações de tráfego real de veículos, emulando os percursos de carros pelas ruas de uma cidade (demonstrando ser mais realista). Tem um padrão de mobilidade baseado no Waypoint, entretanto com resultados mais precisos. Um exemplo padrão de mobilidade pode ser visto na **Figura 11**:

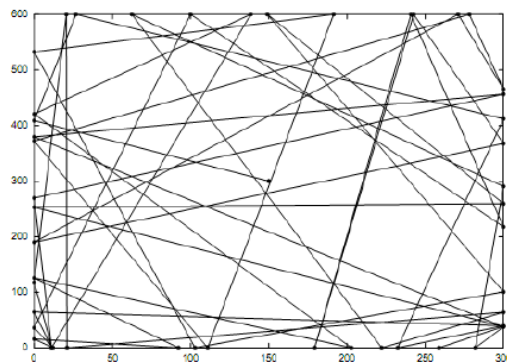


**Figura 11: Exemplo de movimentação do STRAW [21]**

- **Direção aleatória (*Random Direction*)**

Esse modelo de mobilidade força os nós a se deslocar até a borda de simulação e depois trocar de direção e velocidade.

Inicialmente, seleciona-se aleatoriamente uma direção e deslocam-se até alcançar o limite da área de simulação. Quando esse limite é alcançado, o nó para por um determinado tempo, escolhe aleatoriamente outra direção angular (de  $[0, \pi]$ ) e continua o processo.



**Figura 12: Exemplo de movimentação gerada pelo padrão *Random Direction* [20]**

- **Modelo Markoviano de Percurso Aleatório - MPA**

Nesse modelo não existe o movimento na vertical ou na horizontal por duas ou mais vezes consecutivas. Nem paradas no movimento como no Waypoint. Com isso evitam-se mudanças bruscas de sentido e de direção tem-se uma movimentação mais intensa nas diagonais.

- **Modelo Smooth**

Esse modelo utiliza processos estocásticos para a mudança de direção e de velocidade. Permite mudanças suaves na velocidade (e aceleração), entretanto não evita mudanças bruscas na direção.

- **Gauss-Markov**

Esse modelo usa intervalos de tempo discretos para dividir seu movimento. A próxima localização de um nó é gerada em função da localização e velocidades anteriores. Nele existe um parâmetro de ajuste para se adaptar a diferentes níveis de aleatoriedade. A relação entre um novo movimento e um movimento anterior é definida pelas equações abaixo [22]:

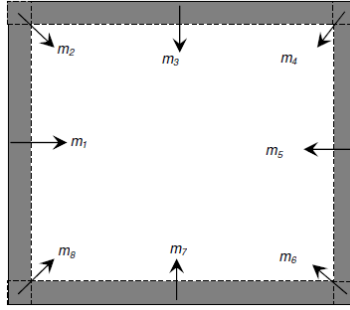
$$\begin{aligned} s_n &= \alpha s_{n-1} + (1 - \alpha) \bar{s} + \sqrt{1 - \alpha^2} s_{x_{n-1}} \\ d_n &= \alpha d_{n-1} + (1 - \alpha) \bar{d} + \sqrt{1 - \alpha^2} d_{x_{n-1}} \\ x_n &= x_{n-1} + s_{n-1} \cos d_{n-1} \\ y_n &= y_{n-1} + s_{n-1} \sin d_{n-1} \end{aligned}$$

Onde  $s_n$  e  $d_n$  são as novas velocidade e direção do nó móvel no tempo  $n$ .

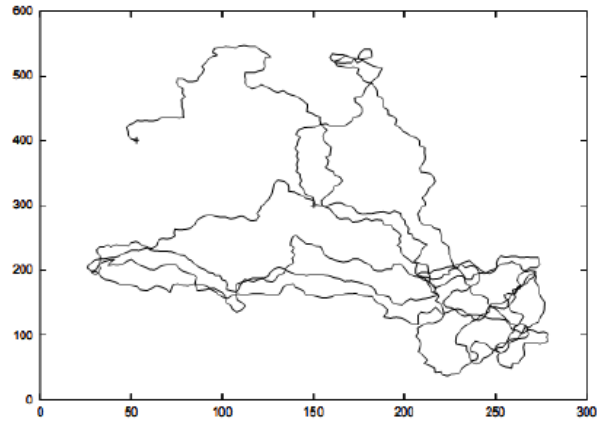
$0 \leq \alpha \leq 1$ , é o parâmetro de ajuste usado para variar a aleatoriedade

$\bar{s}$  e  $\bar{d}$  são os valores médios da velocidade e direção quando  $n \rightarrow \infty$  e  $s_{x_{n-1}}$  e  $d_{x_{n-1}}$  são variáveis aleatórias que seguem uma distribuição Gaussiana.

Para evitar que o nó saia da área de movimentação, quando este atinge as regiões próximas às bordas, a variável  $d$  (direção) do novo movimento do nó receberá os valores das constantes  $m_i$  ( $m_1 = 0, m_2 = 45, \dots, m_8 = 315$ ) dependente da região em que o nó estiver localizado, de forma que o nó retorne em direção ao centro da área de movimentação.



**Figura 13: Constantes de direção [22]**



**Figura 14: Exemplo de movimentação gerada pelo padrão *Gauss-Markov* [22]**

▪ **Área de simulação sem fronteira (*Boundless Simulation Area*)**

Nesse modelo, tem-se memória, ou seja, existe uma relação entre um novo movimento e os anteriores [20], [23]. Cada nova combinação de velocidade e direção é gerada a partir da combinação anterior. O movimento de cada nó na área de movimentação é determinado por dois elementos  $M = M(v, \theta)$ , onde  $v$  é a velocidade e  $\theta$  é a direção, os quais são atualizados a cada período de tempo  $\Delta t$ . A definição das novas velocidade e direção é obtida pelas equações:

$$v(t + \Delta t) = \min[\max[v(t) + \Delta v], V_{m\acute{a}x}]$$

$$\theta(t + \Delta t) = \theta(t) + \Delta\theta$$

$$x(t + \Delta t) = x(t) + v(t) \cdot \cos \theta(t)$$

$$y(t + \Delta t) = y(t) + v(t) \cdot \sin \theta(t)$$

Onde:  $V_{m\acute{a}x}$  é a velocidade máxima definida na simulação,

$\Delta v$  é a variação da velocidade uniformemente distribuída entre

$[-A_{m\acute{a}x} \cdot \Delta t, A_{m\acute{a}x} \cdot \Delta t]$ , sendo  $A_{m\acute{a}x}$  a máxima aceleração do dado móvel

$\Delta\theta$  é a variação na direção que é uniformemente distribuída entre

$[-\alpha \cdot \Delta t, \alpha \cdot \Delta t]$ , sendo  $\alpha$  a máxima variação angular de direção

Outra característica deste padrão é quanto à área de movimentação. Nos padrões anteriores, se o nó móvel atinge uma borda de limite da área de movimentação, ele inicia um novo movimento, com nova direção, de modo a manter-se dentro da área, como um efeito de reflexão. Neste padrão, o nó ao atingir a borda é transferido para outro lado da área, como um toro.

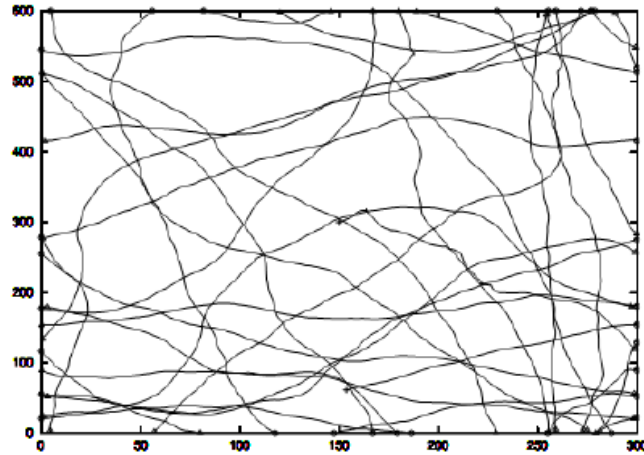


Figura 15: Exemplo de movimentação gerada pelo padrão *Boundless Simulation*  
*Area* [20],[23]

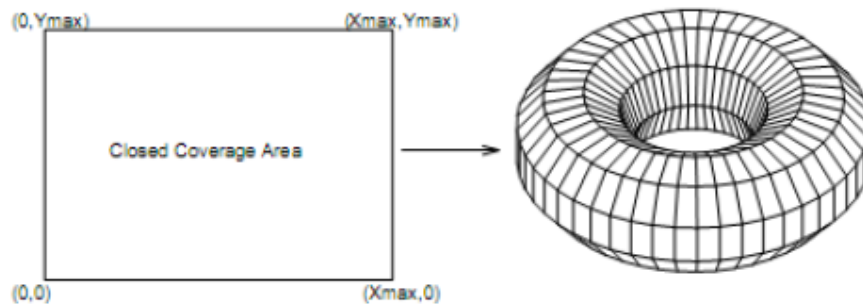


Figura 16: Área de movimentação redesenhada para uma área sem fronteiras  
[20]

Os modelos de **mobilidade em grupo** são modelos que representam o movimento de um grupo de nós móveis, com comportamento de movimentação dependente tanto de intervalos de tempo quanto do relacionamento entre eles. Esses modelos servem para prever a disponibilidade do enlace sem fio e, recentemente,



vem sendo usados para predição do particionamento das redes *Ad hoc*, que são **mudanças de escala global** na topologia da rede.

- **Modelo de grupo baseado num ponto de referência – MGPR**

Nesse modelo tem-se um vetor de movimentação –  $V_{gi}$  e um ponto de referência PR(r). Para lidar com os efeitos nas bordas do cenário de simulação, pode-se inserir alguns tipos de regra de borda, como *bounce* (pulo) em que o nó móvel incide na borda com um certo ângulo e volta com um ângulo diferente. *Delete and replace* (em que o nó incide em uma das bordas e emerge da borda oposta com o mesmo ângulo). E *wrap around* em que um novo ângulo de retorno é escolhendo após se executar uma meia volta.

- **Modelo de Grupo Uniforme**

Baseado no MGPR e no modelo uniforme, foi implementado em [24] um modelo de grupo uniforme. Tal modelo foi de fundamental importância para o presente estudo, pois gera comunidades artificiais a priori. Nele existem 3 (três) tipos de nós e, conseqüentemente, 3 (três) tipos de movimentos, descritos a seguir:

- **nó-referência:** são os nós que são escolhidos para servir como referência para os nós internos. É parâmetro do modelo a quantidade de nós-referências que devem existir na simulação, sendo importante salientar que as posições desses nós, inicialmente, são geradas aleatoriamente na área de simulação. Também é um parâmetro do modelo a distância D que é a diagonal de um quadrado, com origem partindo da posição de um nó-referência. Esta área determinada por esse quadrado é onde possíveis nós internos podem se mover em relação ao nó referência. O nó referência é livre para se mover dentro da área de simulação, mas quando ele se move todos os nós internos à área limite desse nó-referência se movem junto com ele.

- **nó-interno:** é um nó que está dentro da área limite de um nó-referência e pode se mover dentro dessa área livremente. Quando o seu nó-referência move-se, ele acompanha o movimento seguindo o mesmo trajeto do seu nó-referência.

- **nó-externo:** o nó externo não está relacionado a nenhum outro nó. Esse nó é livre para se movimentar pela área de simulação.

## SIMULADOR JiST/SWANS

### ▪ Visão geral

JiST é o acrônimo para *Java in Simulation Time*, um simulador de eventos discretos que usa a máquina virtual Java para escalonar eventos, modificando a maneira de chamadas dos métodos entre as entidades da simulação. Esse simulador é bastante eficiente em termos de utilização de memória e tempo de processamento, comparado com outros como o GloMoSim® e o NS-2® [21].

No JiST, as entidades de simulação são executadas e compiladas como classes regulares de Java e para introduzir o “tempo de simulação”, as simulações são divididas em “Entidades” (classes Java marcadas com a interface *JistAPI.Entity*), cujos métodos públicos são chamados através do escalonador. Antes da execução, as classes de uma simulação são transformadas e reescritas pelo editor de *bytecodes*. Por fim, a simulação é executada pelo núcleo da simulação (uma máquina virtual Java padrão).

Na base do JiST, encontra-se o SWANS ou JiST/SWANS, acrônimo de *Scalable Wireless Ad hoc Network Simulator*, simulador de MANETs escaláveis. Esse é projetado com diferentes e independentes módulos que podem ser combinados para formar uma rede, como protocolos de camada de rede, roteamento, acesso ao meio (MAC), rádios de transmissão, recepção, ruído, além de modelos de mobilidades. Segundo sua documentação, sua interface de mobilidade apresenta os seguintes modelos implementados: Random Walk, Random Waypoint, Random Direction, Boundless Simulation Area, Random Street Waypoint [21], [24].

### ▪ Modelos implementados

No simulador, tem-se acesso a diversos modelos de mobilidade, os quais podem ser selecionados pelo usuário, que deve entrar como argumentos os parâmetros relevantes dos mesmos. Para cada um dos modelos colocam-se abaixo os parâmetros a ordem de entrada correta para simulação:

#### **Gauss Markov**

```
-m gauss_markov:vMin:vMax:v0Med:θ0Med:α:borda
```

#### **Circular Uniforme**

```
-m uniform_circle:k1:k2
```

#### **Uniforme**

```
-m uniform_rect:v_Min:v_Max:μ:passos
```

### Nômade Circular Uniforme

-m nomande\_circle\_uniform:k1:k2

### Grupo Uniforme

-m nomande\_rect\_uniform:vMin:vMax:μ:qtdNósRefer:diagDeGrupo:pausa:passos

### Random Way Point

-m waypoint:pausa:precisao:vMin:vMax

### Teleport

-m teleport:pausa

### RandomWalk

-m walk:raioFixo,raioAleatório,pausa

### Random Direction

-m direction:v:pausa

### Boundless Simulation Area

-m boundless: v\_Max:a\_Max:Δt:Δθ\_Max

#### ▪ Arquivos de *Trace* – parâmetros

Um formato básico de *trace* para simulação é disponibilizado pelo JiST/SWANS. Nele, são registrados eventos para cada simulação executada [24]:

Por exemplo, para o rótulo ‘**n**’, que denota eventos associados ao estado do nó, têm-se seis atributos:

‘-t’: indica tempo em que o evento ocorreu.

‘-a’: denota o endereço.

‘-s’: denota a identificação do nó.

‘-S’: indica o *status* do nó que pode ser *UP* ou *DOWN*, ou seja, tem seu estado representado como ativo ou inativo.

‘-v’: representa a forma do nó, podendo ser quadrado (*square*) ou círculo (*circle*)

‘-c’: representa a cor do nó.

Associados a pacotes, tem-se cinco eventos:

‘h’: *hop* (salto) – pacote transmitido de um nó-fonte a um nó-destino e redirecionado para o próximo salto em direção ao seu destino.

‘r’: *receive* (recepção) – pacote que foi recebido no destino com sucesso.

‘d’: *drop* (perda) – pacote que foi perdido da pilha ou link entre fonte e destino. Perda aqui não distingue perda de pilha ou de link.

‘+’: entrada na pilha – pacote que entrou na pilha do nó-fonte para ser transmitido para nó-destino.

‘-’: saída da pilha – pacote que deixou a pilha do nó-fonte para ser transmitido para o nó-destino.

Outros atributos relacionados a pacotes são:

‘-s’ <src>: nó fonte.

‘-d’ <dst>: nó destino.

‘-p’ <pkt-type>: nome descritivo do tipo de pacote.

‘-e’ <extent>: tamanho em bytes do pacote.

‘-i’ <id>: identificação do pacote na transmissão.

‘-a’ <attr>: uma opção ou atributo para o pacote, que é usado como identificação da cor do pacote.

Os *traces* são gerados em formato *.txt* pelo próprio simulador. Como o interesse do presente estudo foi apenas relacionado aos modelos de mobilidade, fez-se necessário uma filtragem dos arquivos de *traces*, obtendo-se apenas eventos relacionados à mudança de posição dos nós, sendo os outros eventos retirados.

Um exemplo de arquivo de *trace* gerado é mostrado abaixo.

```
n -t 9.157472 -s 38 -S UP -v circle -c blue -x 499.83408 -y 398.1323 -z 4
n -t 9.604041 -s 33 -S UP -v circle -c blue -x 358.19427 -y 104.19745 -z 4
n -t 9.833025 -s 23 -S UP -v circle -c blue -x 369.86334 -y 201.5441 -z 4
+ -t 10.00086 -s 11 -d -1 -p aodv -e 44 -i 4166 -a 0
- -t 10.00086 -s 11 -d -1 -p aodv -e 44 -i 4166 -a 0
r -t 10.0013275 -s 11 -d 38 -p aodv -e 44 -i 4166 -a 0
r -t 10.0013275 -s 11 -d 19 -p aodv -e 44 -i 4166 -a 0
n -t 10.117062 -s 30 -S UP -v circle -c blue -x 458.5308 -y 63.233578 -z 4
n -t 10.127101 -s 22 -S UP -v circle -c blue -x 371.365 -y 398.1226 -z 4
+ -t 10.240146 -s 45 -d -1 -p aodv -e 44 -i 4167 -a 0
- -t 10.240146 -s 45 -d -1 -p aodv -e 44 -i 4167 -a 0
n -t 10.4750185 -s 41 -S UP -v circle -c blue -x 438.62402 -y 249.00677 -z 4
```

#### ▪ Ambiente de simulação utilizando o protocolo AODV

Para gerar os *traces* de mobilidade desejados utilizou-se da classe *aodvsim.java* que iniciava a simulação de uma rede sem-fio utilizando o protocolo AODV, que possui uma boa implementação no JiST/SWANS e é estável e confiável, por já ter sido testado em diversos trabalhos na literatura [25], [26], [27].

Para configurar o ambiente de simulação para cada tipo de modelo desejado utilizou-se a seguinte linha de argumentos:

```
jlist.swans.Main driver.aodvsim -n 300 -f 500x500 -l none -a
random -m nomande_rect_uniform:0:20:100:10:70:0:20 -x true -w
false -s 0 -t 10,1000,10 -r 0
```

onde

- n: *nodes* - é o número de nós desejado para a simulação [*n*].
- f: *field* – dimensões do campo [*largura*×*comprimento*]
- l: *loss* - é o modelo de perdas de pacotes (não utilizado nesse trabalho) [*x*].
- a: *arrange* - arranjo inicial dos nós, podendo ser [*random*] ou [*grid i × j*].
- m: *mobility* - modelo de mobilidade desejado entre os listados acima.
- x: *GUI (Graphical User Interface) support*, instanciada sempre como [*true, false*].
- w: *wraparound* – envolve as bordas do campo [*true, false*].
- s: *send rate* – taxa de envio de mensagens por minuto [*x*].
- t: *timing* – define a atividade do nó [*início, duração, resolução*].
- r: *randomseed* – semente aleatória [*x*].

Sendo que a taxa de mensagens por minuto foi instanciada como zero, pois não havia interesse nos eventos de transferência, mas sim na mobilidade dos nós.

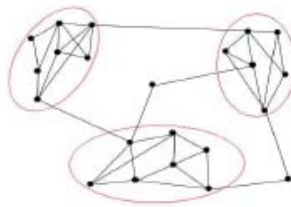
## COMUNIDADES

Ao se analisar redes complexas, uma propriedade que parece ser bastante comum relacionada à interação entre os nós é a formação de estrutura de comunidades (ou *clusters* ou módulos), ou seja, a divisão da rede em grupos que mostram uma alta densidade de conexões internas e uma baixa densidade de conexões externas [28], [29]. As comunidades são de grande interesse, pois geralmente correspondem a unidades comportamentais e funcionais (em redes reais). Assim, a partir da detecção das mesmas, pode-se extrair características específicas, bem como estudar a organização e evolução da dinâmica da rede.

A questão da detecção de comunidades em ambientes sociais é atualmente um tópico de grande importância em áreas tão distintas como a Biologia, detectando unidades funcionais, ou na indústria, correlacionando preços de ações, ajudando prestadores de serviços a identificar grupos de interesse para o seus clientes, ou na World Wide Web, classificando e agrupando websites. Preocupa-se em reconhecer se os dados podem ser agrupados através de alguma propriedade que se manifeste semelhante dentro de cada um dos grupos identificados [30], [31].

Trata-se de um problema de aprendizado de máquina não supervisionado e a definição de comunidade não é tão clara, dependendo do contexto do sistema.

Assim, de maneira geral, comunidades revelam similaridade, segundo algum critério, por meio de conexões entre os vértices pertencentes a um mesmo grupo. Por exemplo, em redes sociais, indivíduos com mesmos interesses, relações de amizade, relações profissionais, entre outras; ou redes biológicas, agrupamentos de células com a mesma função [32].



**Figura 17: Exemplo de rede com três comunidades [32]**

### ▪ Detecção de Comunidades

Uma grande quantidade de métodos para detecção de comunidades pode ser encontrada na literatura. Sendo a detecção de comunidades em redes de alta escala um

desafio e foco de muitos estudos, pois o problema é NP-completo, então busca-se uma solução boa que tenha sido encontrada num tempo viável [28],[33],[34].

As técnicas de *clustering* podem ser divididas em dois grupos principais de acordo com a abordagem que fazem do problema de particionamento, podendo ser:

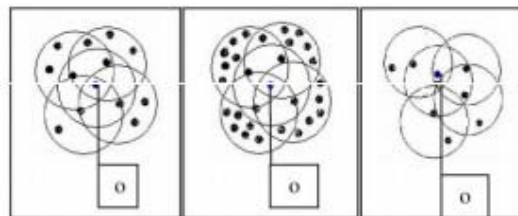
- **globais**, olhando para as redes como um todo e procurando a partir do conhecimento global da rede dividi-la em comunidades ou módulos. Como os módulos hierárquicos.

- **locais**, quando atendem aos padrões existentes em partes particulares da rede, como quando se estuda a existência de cliques completos, como no método de percolação de cliques, onde as comunidades são obtidas identificando os cliques completos adjacentes que constituem as comunidades.

Dentro dessa primeira classificação, surgem as abordagens desde de conceito de *clustering* da mineração de dados até o conceito mais atual de detecção de comunidades. Jain and Dubes (1988) dividiram essas abordagens nos principais tipos: [35]

- **algoritmos particionais**, que são baseados na minimização de uma função custo, com padrões agrupados em um número  $k$  de agrupamentos escolhidos previamente. São exemplos *K-means* e suas otimizações. Nesse caso, obtem-se subgrafos (comunidades) a partir de um número pré-definido de comunidades existentes. Trata-se de um dos mais clássicos algoritmos da área de mineração de dados. Existem maneiras de se estimar esse número de agrupamentos *a priori*.

- **algoritmos baseados em densidades**, que buscam diferenciar regiões de alta densidade de objetos das de regiões de baixa densidade. Tem-se como exemplo o *Density-Based Spatial Clustering of Application with Noise (DBSCAN)*, baseado no conceito de objetos *densamente alcançáveis* e o *Chameleon*.



**Figura 18: exemplo de diferentes conjuntos reconhecidos pelo DBSCAN.**

- **algoritmos hierárquicos**, em que os dados são particionados sucessivamente, produzindo uma representação hierárquica dos agrupamentos, a qual facilita a visualização sobre a formação dos agrupamentos em cada estágio e com que grau de semelhança entre eles. Não requerem que seja definido um número *a priori* de agrupamentos. Através da análise do dendograma pode-se inferir o número de agrupamentos adequados. Se dividem em:

- **métodos divisivos**, os quais utilizam métricas como *Betweenness* ou o *cluster coefficient* (coeficiente de agrupamento entre as arestas). Nas quais as arestas com alto valor da métrica são removidas iterativamente. O próprio *K-means*, poderia ser considerado um método divisivo. Muitas vezes as classificações se confundem.

- **métodos aglomerativos**, inicia-se com cada padrão formando seu próprio agrupamento e gradualmente os grupos são unidos até que um único agrupamento contendo todos os dados seja gerado. No início, os agrupamentos são pequenos e os elementos de cada grupo possuem um alto grau de similaridade. Ao final do processo, tem-se poucos agrupamentos, cada um podendo conter muitos elementos e menos similares entre si. baseados em métricas de *Similaridade* ou também com a *Modularidade*.

As técnicas mais conhecidas, desenvolvidas para a detecção de comunidades são: os métodos espectrais para otimização de modularidade (utilizando autovalores e autovetores), técnicas baseadas em *betweenness* (intermediação), outras otimizações de modularidade (Extrema ou Monte Carlo), detecção baseada no modelo de *Potts*, sincronização, *random walks*, aprendizado competitivo etc [29], [31].

Encontram-se também na literatura estudos comparativos de métodos para detecção de comunidades, geralmente baseados na qualidade das partições geradas e na eficiência computacional dos mesmos [28].

A ideia de comunidade torna-se mais evidente, como definido acima, quando surge uma diferença de densidade de ligações internas e externas num grupo de nós. Assim, é necessário definir critérios (métricas) para avaliar a qualidade do particionamento em comunidades, através dos quais pode definir métodos que, sem análise prévia da estrutura da rede, possam diferenciar entre redes com uma estrutura de comunidades bem definida e redes basicamente formadas por conexões aleatórias [28].

Matematicamente, considerando o grafo  $G(V, E)$  em que  $V$  é o conjunto de  $n$  nós e  $E$  o conjunto de  $m$  arestas e  $k_i$  o grau de cada nó. Considera-se uma estrutura de



comunidade, ou seja, uma partição de  $V$  em  $c$  comunidades, tais que  $\mathcal{C} = \{C_1, \dots, C_c\}$ , em que  $V = \bigcup_{i=1}^c C_i$  e  $\bigcap_i C_i = \emptyset$ .

Denomina-se ainda  $n_{C_i}$  o número de nós em  $C_i$ ,  $m_{C_i}$  o número de intraconexões em  $C_i$  tais que:  $m_{C_i} = |\{(u, v): u \in C_i, v \in C_i\}|$  e  $b_{C_i}$  o número de interconexões, tais que  $b_{C_i} = |\{(u, v): u \in C_i, v \notin C_i\}|$ .

Essa ideia de comunidades que não se sobrepõem não está tão presente em redes reais, nas quais se observa que os nós geralmente pertencem a mais de uma comunidade. Trabalhos mais recentes vêm propondo novas técnicas para detecção de comunidades que possuem interseções, mas o presente trabalho não se preocupará com isso, implementando algoritmos apenas para comunidades disjuntas.

Como último comentário, deve-se ter em mente ainda que as comunidades podem ser descobertas em vários graus de “resolução” em uma determinada rede (e.g. micro e macro escalas). Algumas métricas de qualidade oferecem a possibilidade de explorar as diversas escalas, outros não.

#### ▪ ***K-means***

O algoritmo inicia-se escolhendo-se aleatoriamente um número  $k$  de protótipos (centros) para os *clusters*. Atribui-se cada objeto para o cluster de centro mais próximo (segundo alguma distância, e.g. Euclidiana). Move-se, então, cada centro para a média (centróide) dos objetos do cluster correspondente. Repete-se os dois passos anteriores até algum critério de convergência, por exemplo, um número máximo de iterações ou um limiar mínimo de mudanças nos centróides.

A seleção dos protótipos iniciais tem um papel importante para a eficácia do algoritmo, existindo alternativas para a inicialização.

A complexidade computacional é linear em todas as variáveis críticas.

Existem variantes que utilizam *k-medians*, *k-medóides*.

#### ▪ **Betweenness (Intermediação)**

A métrica de centralidade *Betweenness*, foi definida por Girvan e Newman (2002). Ao invés de tentar construir uma métrica que mostra que arestas são mais centrais para as comunidades, este método foca nas arestas que são menos centrais, aquelas que estão mais “entre” as comunidades. E, ao invés de construir comunidades

adicionando arestas mais fortes a um conjunto vazios de vértices, constrói-se os conjuntos removendo arestas do grafo original. [36]

A métrica é definida como:

$$B_u = \sum_{ij} \frac{\sigma(i, u, j)}{\sigma(i, j)}$$

onde  $\sigma(i, u, j)$  é o número de caminhos mínimos entre os vértices  $i$  e  $j$  e que passam pela aresta  $u$ .

$\sigma(i, j)$  é o número total de caminhos mínimos entre os vértices  $i$  e  $j$ .

O algoritmo segue os seguintes passos: inicialmente, calcula-se o *Betweenness* para todas as arestas na rede. Em seguida, remove-se a aresta com mais alto valor da métrica. Recalcula-se então, a intermediação para todos as arestas afetadas pela remoção. E repete-se o processo até que não sobrem mais arestas.

A complexidade computacional é para o pior caso de ordem  $O(m^2n)$ , para uma rede de  $m$  arestas e  $n$  vértices.

#### ▪ Modularidade

A métrica mais frequente para avaliação de estruturas de comunidade é a modularidade  $Q$ , que é baseada na ideia que um subconjunto de vértices pode ser considerado uma comunidade se o número de conexões internas é maior que o esperado numa configuração aleatória. Quanto maior o valor de  $Q$ , melhor a estrutura de comunidades encontrada. [28], [29].

Considerando que a chance de a conexão de  $i$  ser uma das  $k_j$  conexões de  $j$  é de  $\frac{k_j}{2m}$ , a modularidade  $Q$  de um particionamento pode ser calculada por:

$$Q = \frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, c_j) - \frac{1}{2m} \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j)$$

em que o primeiro termo é o número de conexões entre vértices da mesma comunidade e o segundo é o esperado numa versão aleatória.

Sendo  $\delta(a, b) = \begin{cases} 1, & \text{se } a = b \\ 0, & a \neq b \end{cases}$ , o delta de Kronecker.

A matriz de modularidade  $B$ , pode então ser definida como:

$$B = A - \frac{kk^T}{2m}$$

onde:  $A$  é a matriz de adjacências,

$k$  é o vetor de graus para cada nó,

$m$  é a quantidade de arestas.

Assim, é esperado que uma estrutura de comunidade correspondente a uma máxima modularidade seja o melhor esquema de particionamento, ou no mínimo um muito bom. O que leva ao problema de maximizar a modularidade, descrito em diversos estudos na literatura [28], [33].

Pode-se ainda reescrever a modularidade como:

$$Q = \sum_i (e_{ii} - a_i^2)$$

onde:  $e_{ii}$  é a fração de arestas inseridas no grupo  $i$ .

$a_i^2$  é a fração de arestas inseridas aleatoriamente no grupo  $i$ .

#### ▪ Otimização espectral de modularidade

Um método eficiente para otimizar a modularidade em redes complexas é através do particionamento espectral dos grafos e da relação entre suas propriedades estruturais e os autovetores de sua matriz de adjacências.

Dada uma matriz  $A_{n \times n}$ , o problema de autovalores é definido pela busca de um escalar  $\lambda$  e um vetor  $v$  tais que  $Av = \lambda v$ , em que o escalar  $\lambda$  é o autovalor e o  $v$  o autovetor correspondente.

Considerando a divisão da rede em somente duas comunidades, tal divisão pode ser representada por um vetor  $s \in \{-1, 1\}^m$ , em que, para cada vértice  $v_i$ , associa-se  $+1$ , se  $v_i \in C$  e  $-1$ , se  $v_i \notin C$ . Então

$$Q = \frac{1}{4m} s^T B s$$

Assim, o objetivo é encontrar o valor de  $s$  que maximiza  $Q$  para uma dada matriz  $B$ . Os elementos de  $s$  são restritos a  $\pm 1$ , mas se relaxarmos esta restrição, ele pode apontar em qualquer direção e uma solução aproximada do problema de otimização é:

$$Bs = \beta s$$

em que  $s$  é um dos autovetores da matriz de modularidade  $B$ . Considerando que  $s^T s = \sum_i s_i^2 = n$ , obtem-se:

$$Q = \frac{n}{4m} \beta$$

Para maximização de modularidade, escolhe-se o valor de  $s$  como autovetor  $u_1$  correspondente ao maior autovalor da matriz de modularidade. Entretanto, não se

pode escolher  $s = u_1$ , pois os elementos de  $s$  são sujeitos à restrição  $s_i = \pm 1$ . Mas  $s$  pode ser aproximado adotando os valores  $+1$ , se  $[u_1]_i \geq 0$  e  $-1$ , se  $[u_1]_i < 0$ .

Em suma, o autovetor da matriz de modularidade correspondente ao maior autovalor é calculado e então os vértices são atribuídos a cada comunidade de acordo com os sinais dos elementos (positivos para uma comunidade, negativos para outra).

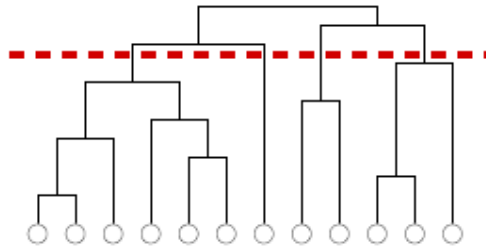
Entretanto, é possível que os nós possam se dividir em  $c$  grupos sem nenhuma hipótese acerca de  $c$ , assim a formulação original do algoritmo proposta por Newman deve ser adaptada, permitindo detectar mais de duas comunidades [34], e isso pode ser feito através de um processo de sucessivas bipartições. Neste caso, a divisão de uma comunidade é realizada somente se isso resultar em algum ganho em modularidade. E a possibilidade da divisão em grupos menores é analisada para cada uma das partes até que se aceite que uma maior divisão não resultará em nenhum ganho de modularidade.

Para medir a variação da modularidade da nova rede gerada pela bipartição da comunidade  $C$ , Newman definiu [37]:

$$\Delta Q = \frac{1}{2m} \left[ \frac{1}{2} \sum_{i,j \in C} B_{ij} (s_i s_j + 1) - \sum_{i,j \in C} B_{ij} \right] = \frac{1}{4m} s^T B^{(c)} s$$

Para detecção de mais de duas comunidades, o algoritmo é implementado de maneira recursiva, atuando para cada nova comunidade detectada enquanto existir uma variação de modularidade  $\Delta Q > 0$ .

Assim, escolhe-se um grau de hierarquia de um dendograma para definir o particionamento em comunidades, como ilustrado na **Figura 18**:



**Figura 19: “corte” do dendograma, ilustrando o particionamento em comunidades pelo método espectral.**

Como a otimização espectral envolve a resolução de um problema de autovalores e autovetores, o custo computacional é alto. Para superar essa inconveniência, outros métodos são propostos para redes de alta escala.

- ***FastGreedy* (Clauset, Newman and Moore)**

O algoritmo usa uma estratégia “gananciosa” (“gulosa”) para a detecção de comunidades. Inicialmente, ele associa cada vértice da rede a uma comunidade. E, repetidamente, fusiona as comunidades cuja união produz a maior elevação da modularidade  $Q$ .

Considerando uma rede de  $n$  vértices, após  $n - 1$  combinações, o resultado é uma grande comunidade contendo todos os vértices e o algoritmo para.

Para um vértice  $v$ , pertencente à comunidade  $c_v$ ,  $e_{ij}$  (fração de arestas que conectam vértices na comunidade  $i$  com vértices na comunidade  $j$ ) pode ser definida como:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, j)$$

E  $a_i$  (fração de extremidades de arestas associadas a vértices na comunidade  $i$ ). Assim, a modularidade é calculada como:

$$Q = \frac{1}{2m} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{2m} \sum_i \delta(c_v, i) \delta(c_w, j) \right] = \sum_i (e_{ii} - a_i^2)$$

O algoritmo objetiva encontrar a combinação de comunidades  $i$  e  $j$  que resulta no maior aumento de  $Q$ , e então, procede à operação, juntando comunidades com maior afinidade.

A complexidade do algoritmo é de  $O(n \log^2 n)$ , podendo ser usada para redes de grande escala. Entretanto, os valores resultantes de modularidade são piores comparados a outros métodos como o espectral.

Tem-se ainda algumas variações desse método com pequenas alterações para melhorar os valores de modularidade e até mesmo melhorar o tempo de execução do *FastGreedy*. Pode-se citar Danon, Diaz e Arenas [39], Wakita e Tsurumi [40], Leon-Suematsu e Yuta [33] e Blondel, Guillaume, Lambiotte e Lefebvre [41]

- ***Walktrap - Random Walks* (caminhadas aleatória)**

A abordagem baseia-se na ideia intuitiva de que caminhadas aleatórias num grafo tendem a ser aprisionadas em partes altamente conectadas, que correspondem a comunidades. Tem a grande vantagem de baixo custo computacional  $O(n^2 \log n)$ . [42]

▪ **Modelo de Potts**

Esse algoritmo move os nós identificando para qual comunidade eles devem ser movidos de tal forma que a energia do sistema seja diminuída. Procedendo desta maneira até que movimentos de nós não sejam mais possíveis. Apresenta a citada resolução local para detecção de comunidades como pode ser percebido a seguir, com certa tolerância a ruído. [44]

Os passos do algoritmo são os seguintes:

- 1) *Inicia-se o sistema*, através das matrizes  $A_{ij}$  e  $J_{ij} = I - A_{ij}$  e dos pesos das arestas  $a_{ij}$  e  $b_{ij}$ . Caso o grafo seja não ponderado  $a_{ij} = b_{ij} = 1$ . O sistema se inicia em um estado “simétrico” em que cada nó forma sua própria comunidade individual ( $q_0 = N$ ). Se o número de comunidades for pré-estabelecido, inicia-se o sistema randomicamente com  $q_0 = q$  comunidades.
- 2) *Otimiza-se as associações dos nós*. Para cada nó, percorre-se a lista de vizinhos, calcula-se a mudança de energia se o nó for movido para cada *cluster*. Imediatamente coloca-o na comunidade com menor energia.
- 3) *Itera-se até a convergência*. Repete-se o passo (2) até que a o mínimo de energia seja atingido, em que nenhum movimento de nó diminui a energia do sistema.
- 4) *Testa-se para cada mínimo local de energia*. Manualmente. Funde-se manualmente quaisquer comunidades que irá baixar mais a energia do sistema. Se alguma fusão for encontrada, retornar ao passo (2) para um refinamento adicional a nível de nó.
- 5) *Repetir para várias tentativas*. Repetir os passos (1)-(4) para  $t$  tentativas independentes e selecione o resultado de menor energia como melhor solução. Como tentativa entende-se uma cópia da rede na qual o sistema inicial é randomizado.

O custo computacional fica da ordem de  $O(tL^{1,3} \log k)$ .

## SIMULAÇÃO E ANÁLISE DOS MÉTODOS

### ▪ Visualização das redes móveis

Para visualização e análise qualitativa dos resultados dos algoritmos de detecção de comunidades, fez-se necessário a utilização de um *Network Animator*, que transformasse os traces obtidos no JisT/Swans numa matriz de adjacências, na qual seriam executados os algoritmos de detecção de comunidades e, em seguida, realizações visuais dos nós, com suas arestas e, respectivas comunidades.

Assim, inicialmente, buscou-se os principais visualizadores existentes em literatura. Encontrou-se o NAM (Network Animator), que utilizava os próprios *traces* em formato *.nam* do JisT/Swans para gerar a animação da movimentação dos nós. Entretanto, tal *framework* não possibilitava a formação da matriz de adjacências nem mesmo uma integração com os algoritmos de comunidade.

Em seguida, tentou-se o VAT4Net e o Javis 0.3, também sem muitos resultados. Além do TopVis e da própria biblioteca Prefuse (em Java) para exibições de grafos, entretanto, não se encontrou possibilidades para integração com os algoritmos.

Por fim, optou-se por se desenvolver um *Network Animator* próprio para o estudo, também utilizando a linguagem Java, através das classes da GUI (Graphic User Interface).

Assim, o arquivo de *trace* obtido do JisT/Swans era lido e para um determinado raio de alcance (de antena) construía-se, para cada nó, as matrizes de adjacências correspondentes a cada instante de tempo. Sendo considerados vizinhos de um dado nó todos aqueles outros que estivessem dentro do seu raio de alcance.

Para cada instante de tempo, era necessário rodar o algoritmo de detecção de comunidades, mostrando-se necessário uma biblioteca para lidar com álgebra linear, visto as complexidades computacionais dos algoritmos.

Em primeira ideia, encontrou-se o pacote JAMA, um pacote básico de álgebra linear desenvolvido para Java, cujas principais características são colocadas, em resumo, na **Tabela 5** [45].

Entretanto, o tempo computacional para as redes com maior quantidade de nós era inviável e muitas funções básicas não existentes acabavam tendo de ser implementadas em Java, sem muita eficiência.

Assim, buscou-se uma segunda alternativa - uma API de Java para interagir com o MATLAB – denominada *MATLABcontrol* [46], que permite rodar os comandos diretamente no *prompt* do programa.

**Tabela 5: Summary of JAMA Capabilities [45].**

<b>Object Manipulation</b>	constructors set elements get elements copy clone
<b>Elementary Operations</b>	addition subtraction multiplication scalar multiplication element-wise multiplication element-wise division unary minus transpose norm
<b>Decompositions</b>	Cholesky LU QR SVD symmetric eigenvalue nonsymmetric eigenvalue
<b>Equation Solution</b>	nonsingular systems least squares
<b>Derived Quantities</b>	condition number determinant rank inverse pseudoinverse

Deste modo, o problema passou a ser a implementação dos algoritmos em linguagem de mais alto nível (MATLAB), conseguindo-se melhor tratamento com matrizes e uma maior quantidade de funções já existentes. Para esta opção, as redes de maior escala ainda tomavam um tempo razoável, mas significativamente menor (avaliando-se qualitativamente) que com o pacote JAMA.



- ***Network Animator***

O programa basicamente lê o arquivo de *traces* gerado pelo JisT/Swans já devidamente tratado para que contivesse apenas eventos relacionados à movimentação dos nós e apresenta para cada instante de tempo o grafo com as comunidades encontradas pelo algoritmo em cores distintas, considerando todas as comunidades com apenas um nó (nós isolados) com a cor verde.

A estrutura do programa contém as seguintes classes:

- ***Main***, que é a classe principal em que se inicia o programa, na qual optou-se por colocar a interface gráfica (JButton, JSlider, JtextField, Jpanel, JCheckBox) e a referência para as classes *DataReader*, *GraphInformation* e *GraphPainter*;

- ***DataReader***, que lê o arquivo de texto com os eventos e gera um vetor de eventos, em que cada elemento contém o *id* do nó, a posição e o tempo;

- ***GraphInformation***, que organiza as informações dos eventos para que sejam desenhados na tela. Esta classe utiliza um vetor com os instantes de tempo de cada evento e, para cada instante, obtém um vetor com as posições dos nós a serem desenhados, a matriz de adjacências e a matriz de comunidades, através da classe *FastModularityAlgorithm*;

- ***FastModularityAlgorithm***, que, a partir das matrizes de adjacências já obtidas em *GraphInformation*, calcula as matrizes de comunidade. Para isso, utilizou-se a classe *MatlabControl*, que permite abrir o MATLAB e enviar comandos e obter os valores calculados, possibilitando a implementação dos algoritmos diretamente em MATLAB.

- ***GraphPainter***, que herda de *JPanel* e utiliza as informações obtidas em *GraphInformation* e, para um determinado tempo, desenha o grafo, com os nós e *ids*, as arestas e as comunidades coloridas.

Um diagrama de classes é mostrado a seguir na **Figura 20**:

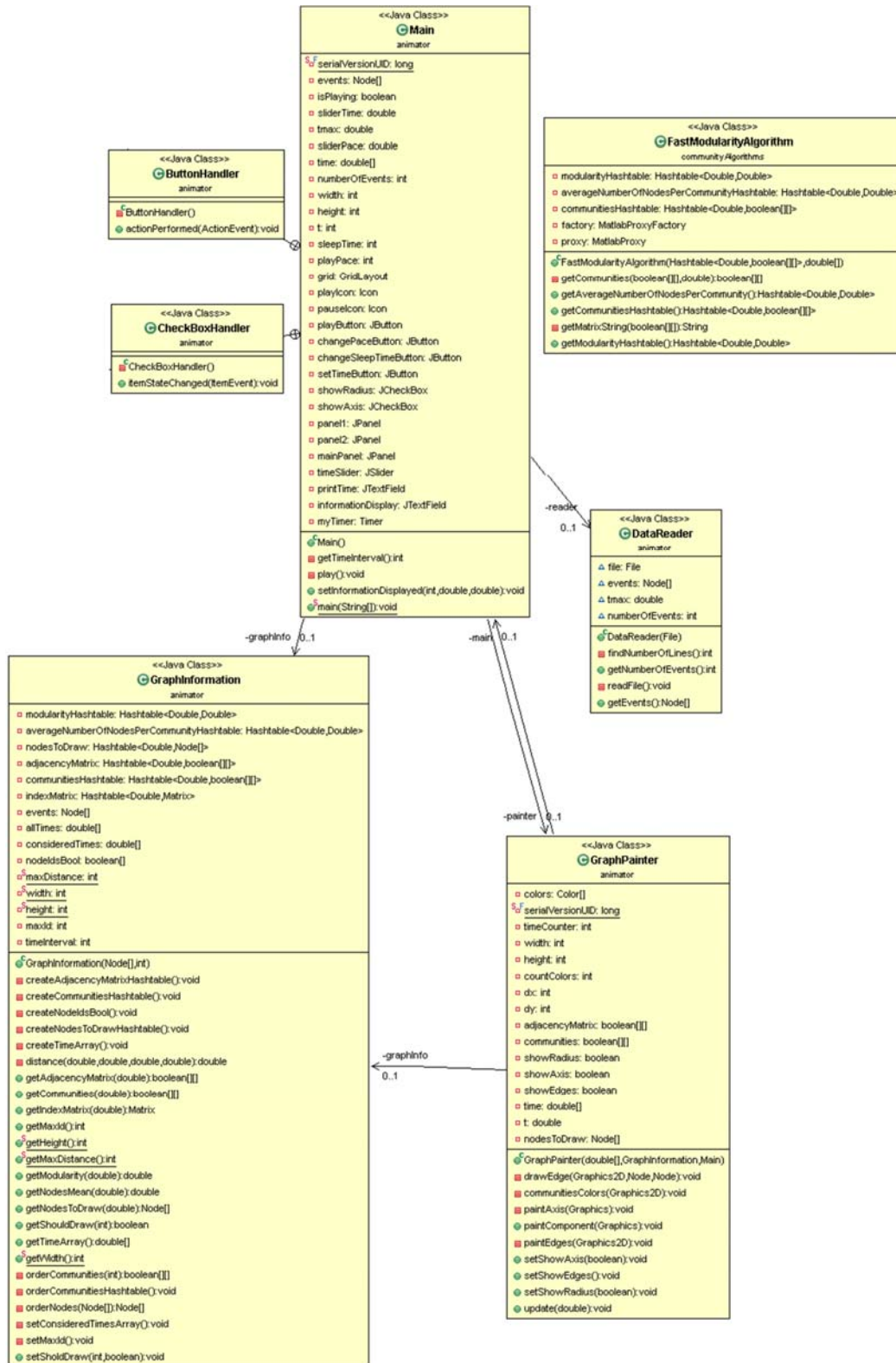


Figura 20: diagrama de classes para o *Network Animator*

#### ▪ Configuração dos experimentos

Para demonstração e análise do método de detecção de comunidades nas redes móveis (a cada instante de tempo), escolheram-se dois modelos de mobilidade o **Random Way Point (RWP)**, devido à sua característica aleatória e não formadora de comunidades e o **Grupo Uniforme (GU)**, pela sua característica de grupo, podendo ser utilizado para gerar comunidades artificiais *a priori* (as quais os algoritmos deveriam detectar). Assim, pôde-se avaliar a evolução dos nós e a evolução das métricas em cada um dos casos.

Inicialmente, para validação qualitativa (visual) dos modelos e algoritmos de detecção de comunidades, realizaram-se simulações no JisT/Swans durante um tempo de 1000s e uma pausa igual a 0s, numa área de simulação igual a  $500 \times 500 m^2$  com uma quantidade de 50 nós com velocidades escolhidas uniformemente entre 0 e 20m/s. Gerando-se ainda para os dois modelos uma configuração inicial dos nós aleatória ou em um grupo concentrado inicial (*grid*).

Em seguida, foram realizadas simulações para comparação e observação dos modelos (através da métrica de modularidade e outros índices) num tempo de 1000s e uma pausa igual a 0s, numa área de simulação igual a  $500 \times 500 m^2$  com uma quantidade de 50, 100, 150, 200, 250 e 300 nós com velocidades escolhidas uniformemente entre 0 e 20m/s. Gerando-se ainda para o caso do modelo **Random Way Point**, a configuração inicial dos nós aleatória ou em um grupo concentrado inicial (*grid*).

Para computar a matriz de adjacências (vizinhos de cada nó), utilizaram-se circunferências de raio de 70m, ou seja, o nó que estivesse dentro da área da circunferência seria considerado vizinho.

Para o modelo Grupo Uniforme foi utilizada uma média  $\mu = 100$ , empregada na distribuição exponencial para gerar as distâncias as serem percorridas pelos nós. Tal distância é justificada por ser o mais próximo das distâncias percorridas pelo RWP. E uma diagonal também de 70m correspondente ao raio de alcance.

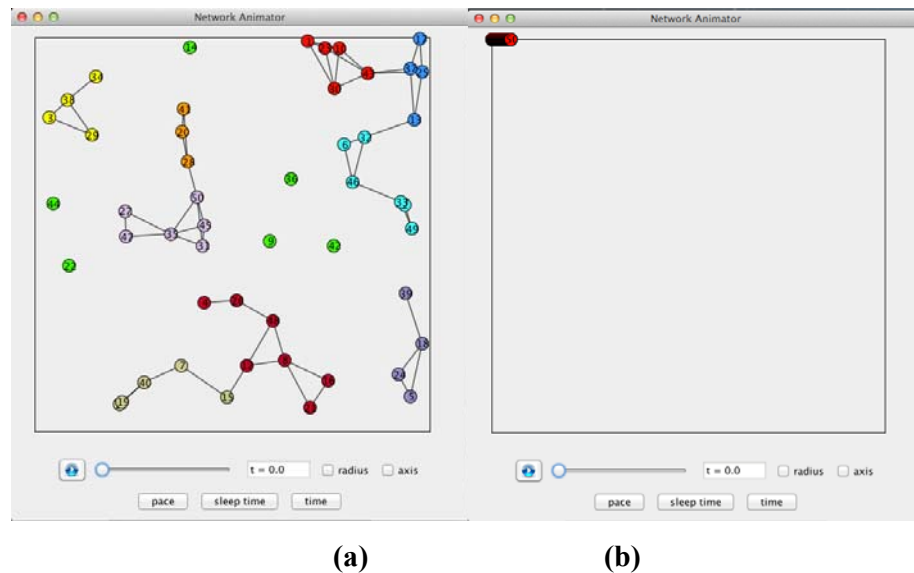
Para construção das curvas, tomou-se a média aritmética de 30 amostras para cada cenário testado, obtendo-se a métrica para 50 valores de tempo (com um intervalo de 20s entre elas).

Escolheu-se, entre os algoritmos pesquisados para detecção de comunidades, o *FastGreedy*, devido à eficiência que o algoritmo apresenta em relação aos outros métodos, como já densamente mostrado em literatura. [28]

- **Análise qualitativa**

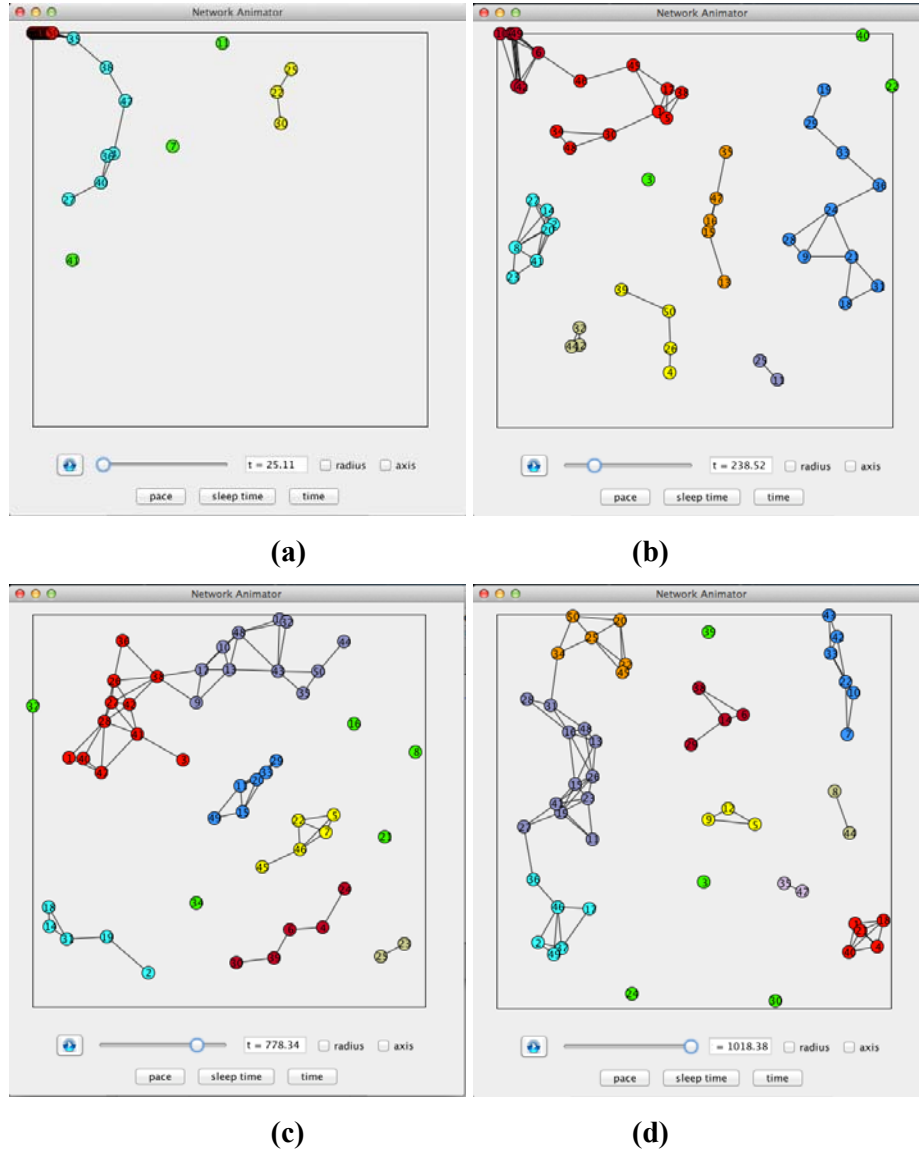
- **Validação dos modelos**

Para a validação visual e qualitativa dos modelos, observou-se a evolução dos nós a partir de uma configuração inicial aleatória e a partir de um *grid* concentrado, para os dois modelos. Condições iniciais essas ilustradas na **Figura 21**:



**Figura 21: configurações iniciais para os modelos (a) aleatória e (b) *grid* concentrado**

Para o *Random Way Point*, observou-se, após cerca de 400s, que a caminhada aleatória dos nós gerava comportamentos praticamente sem diferença entre as duas condições iniciais. Percebe-se ainda uma ligeira predominância das comunidades próxima ao centro do campo (corroborando com [28]) como ilustrado na **Figura 22**:

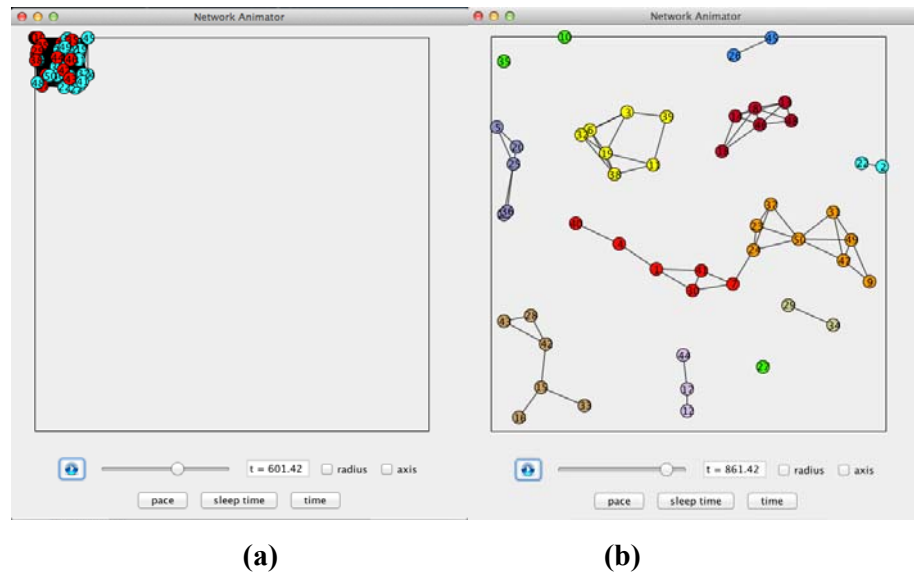


**Figura 22: posição intermediária para o Random Way Point para (a) (b) e (c) condição inicial num *grid* concentrado e (d) para condição inicial aleatória**

Já para o modelo Grupo Uniforme, a configuração inicial em *grid* gera uma concentração dos nós no local inicial, como ilustrado na **Figura 23(a)**, o que serve para validação do modelo, segundo o qual os nós interiores que estivessem dentro do quadrado com vértice nos nós de referência, só poderiam se mover dentro desse quadrado (no presente estudo para diagonal  $D = 70$ ).

Para o caso inicial de configuração aleatória, tem-se uma configuração intermediária como na **Figura 23(b)**, na qual pode-se perceber a formação de 10 comunidades (quantidade de comunidades essa que predomina por grande parte da

simulação), como era esperado pelo modelo de mobilidade que fora instanciado para 10 nós de referência, o que mais uma vez serve para validação do modelo de mobilidade e do próprio *Network Animator*.



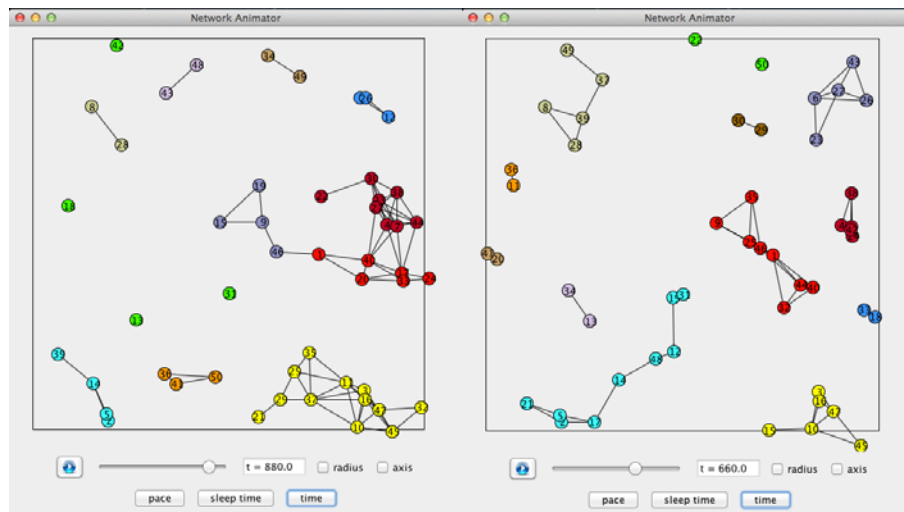
**Figura 23: posição intermediária para o Grupo Uniforme para (a) condição inicial num *grid* concentrado e (b) para condição inicial aleatória**

#### ○ Validação dos algoritmos de detecção de comunidades

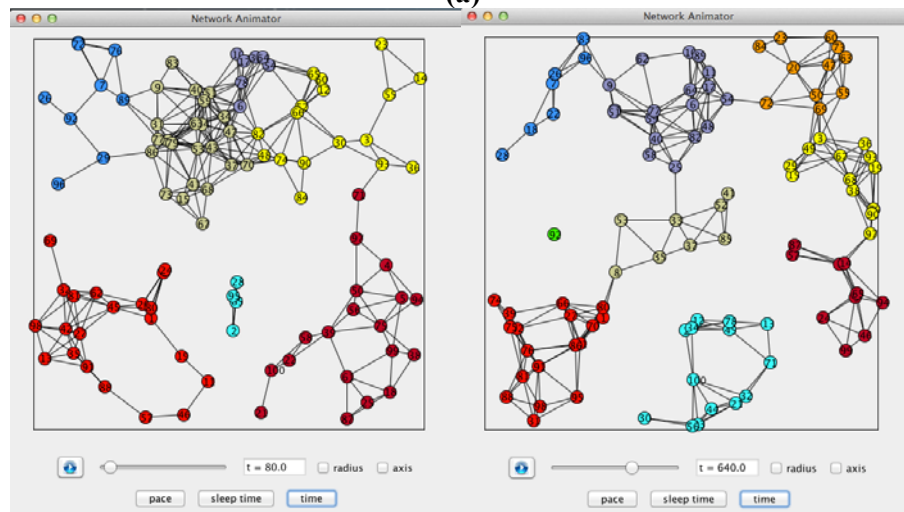
A validação qualitativa da detecção de comunidades é facilitada pela utilização de cores diferentes para cada comunidade encontrada pelo algoritmo, sendo que os nós isolados foram pintados de verde.

Percebe-se, para cada um dos modelos, particionamentos aceitáveis do grafo. Lembrando-se que o conceito de comunidades em redes reais é razoavelmente nebuloso e deve-se considerar a escala (macro ou microscópica) para detecção das comunidades.

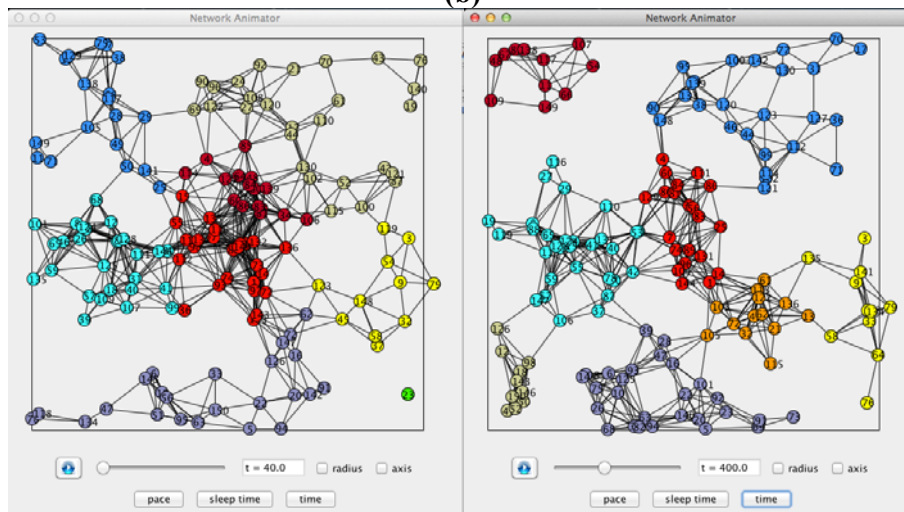
Avaliando-se a densidade de ligações internas a um certo grupo de nós em relação à densidade de ligações externas, pode-se considerar visualmente justificável o particionamento encontrado pelo algoritmo, o que pode ser visualizado nas **Figura 24**, nas quais selecionou-se instantes de maior e menor modularidade para o modelo de *Random Way Point*, durante os experimentos em que se variou o número de nós entre 50 e 300.



(a)

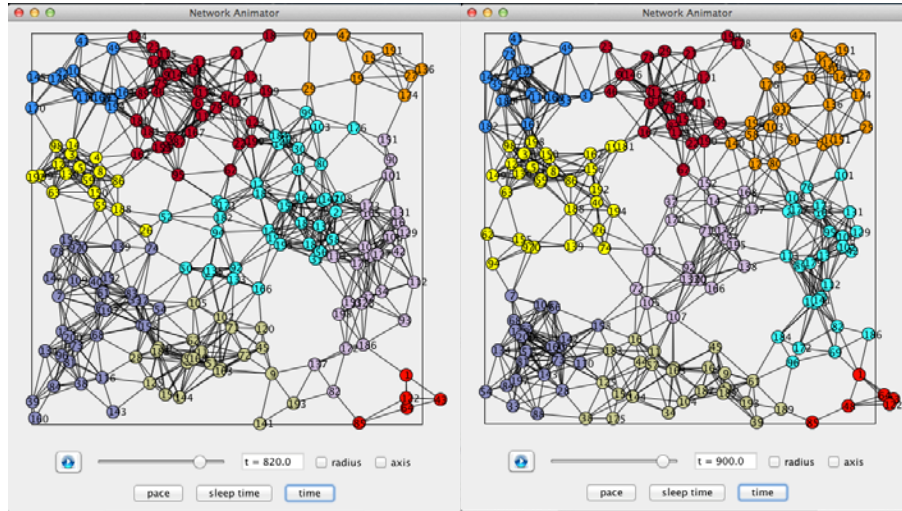


(b)

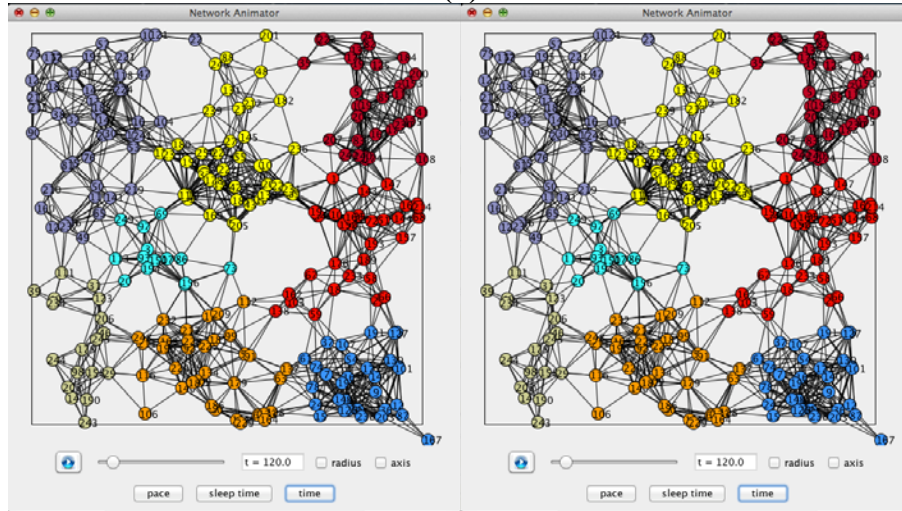


(c)

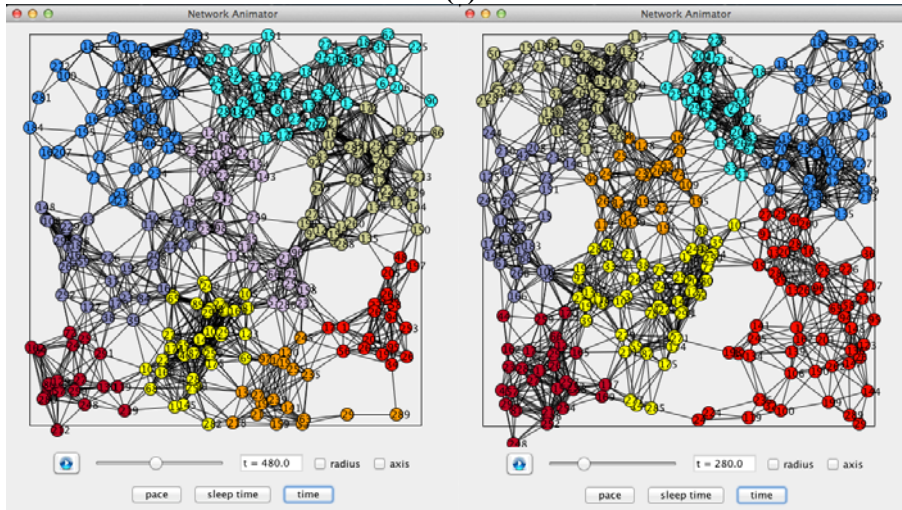




(d)



(e)



(f)

Figura 24: detecção das comunidades no modelo de *Random Way Point*, para os valores máximo e mínimo de modularidade para (a) 50 (b) 100 (c) 150 (d) 200 (e) 250 (f) 300.



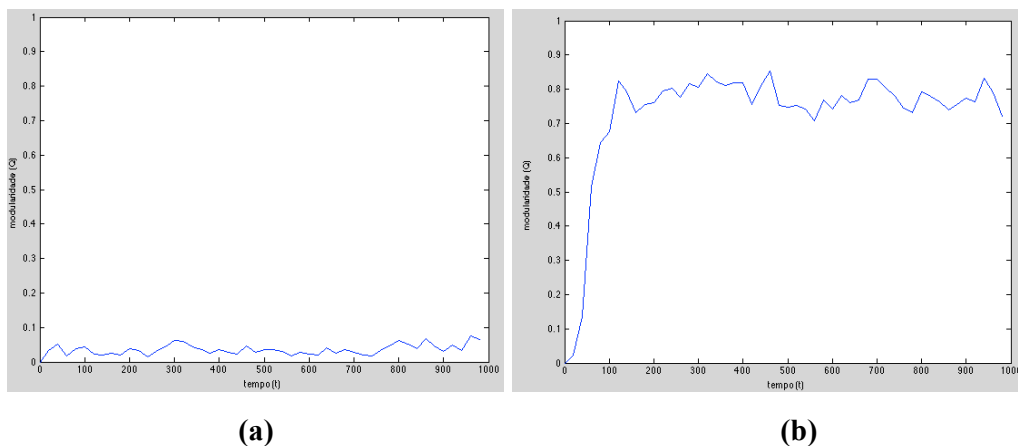
### ▪ Análise quantitativa

Para realização da análise quantitativa da detecção e acompanhamento de comunidades pelo método proposto, observou-se o comportamento da própria métrica de **modularidade**, utilizada como fator de qualidade para otimização do particionamento do grafo pelo algoritmo. Além de propor-se o índice de média dos nós por comunidade para cada modelo, como uma métrica para acompanhamento do comportamento das comunidades para cada modelo de mobilidade.

#### ○ Avaliação da métrica de modularidade

Inicialmente, acompanhou-se a modularidade obtida para a rede de 50 nós, utilizada na seção de **avaliação qualitativa**.

Faz-se necessário iniciar com dois resultados, que demonstram como a métrica de modularidade evolui com o tempo. Selecionou-se, para a configuração inicial de *grid* concentrado, o modelo de Grupo Uniforme - representado na **Figura 24(a)** - na qual os nós não saíram para mais que um raio de 70m no canto superior esquerdo do campo. E para o *Random Way Point*, representado na **Figura 24(b)**, na qual os nós se espalharam pelo campo com o passar do tempo. Para o caso da configuração inicial aleatória foi realizada investigação em tópico posterior.



**Figura 25: evolução da modularidade com condição inicial de *grid* concentrado (a) para o modelo de Grupo Uniforme e (b) para o *Random Way Point***

É interessante notar que no caso do Grupo Uniforme, em que os nós ficaram concentrados, as maiores modularidades encontradas pelo algoritmo são menores que  $Q < 0,1$ , o que indica a dificuldade em se encontrar um bom particionamento em

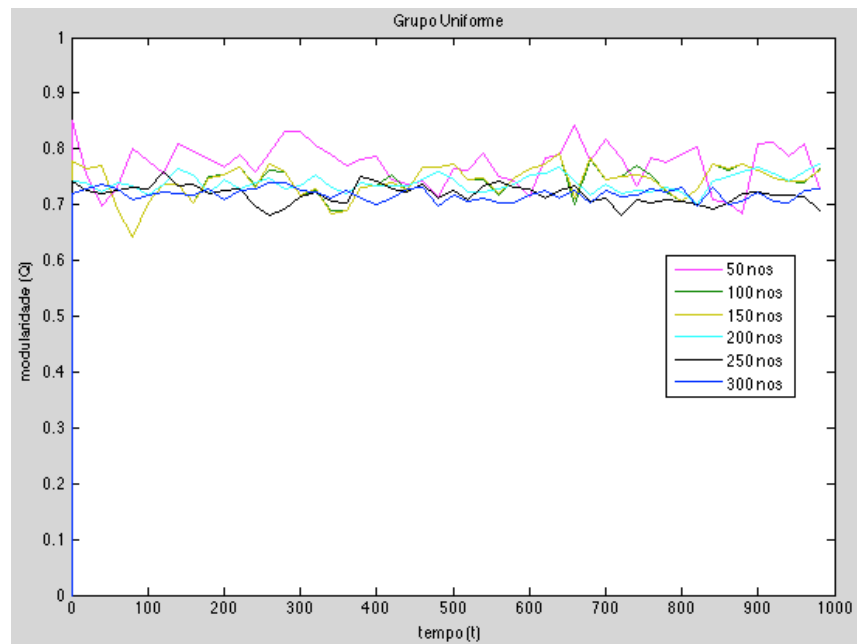
comunidades, dado que todos os nós pertencem a uma única comunidade (pois todos são vizinhos de todos).

Já no caso do *Random Way Point*, percebe-se a evolução da modularidade de valores nulos, quando dos nós concentrados, até valores de  $0,7 < Q < 0,8$ , quando os nós se espalham pelo campo. Tal situação evidencia que pelo fato dos nós se espalharem pelo campo, o particionamento em comunidades fica mais bem definido, o que permite ao algoritmo encontrar maiores valores de modularidade, durante a otimização.

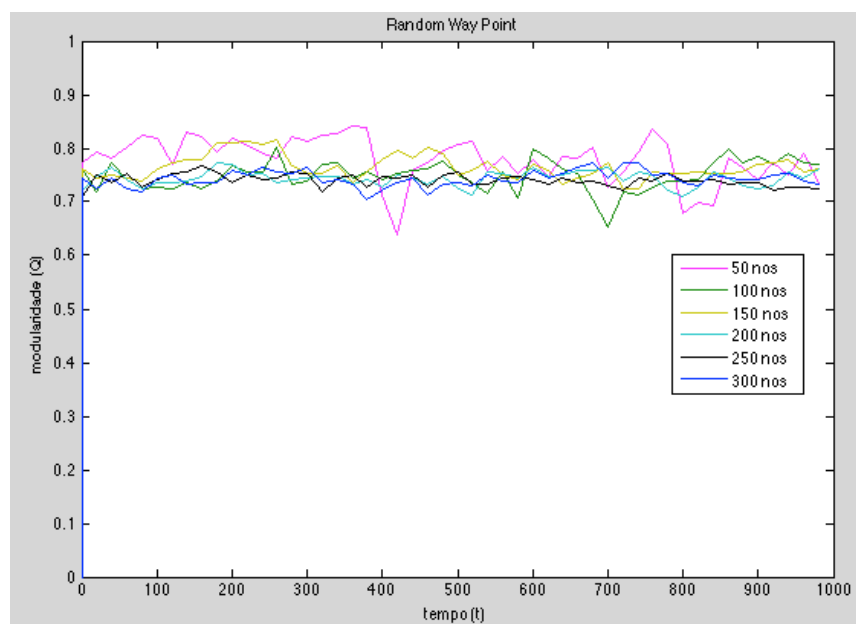
Assim, pode-se perceber que a métrica de modularidade está relacionada à facilidade ou dificuldade de se encontrar uma estruturada de comunidade na rede.

#### ○ **Variação da modularidade com o número de nós.**

Para investigar como a modularidade poderia variar com a variação do número de nós para cada modelo e ao mesmo tempo comparar os dois modelos quando iniciados com uma configuração aleatória, realizou-se as experiências descritas anteriormente. Obtendo-se os seguintes resultados:



**(a)**



(b)

**Figura 26: evolução da modularidade com o tempo, variando-se o número de nós entre 50 e 300 para (a) Grupo Uniforme (b) *Random Way Point***

Para ambos os modelos, observou-se uma diminuição da amplitude de variação da modularidade com o aumento do número de nós, e também da modularidade máxima encontrada. Isso é justificável devido ao aumento da dificuldade em se discriminar um melhor estrutura de comunidades. Com uma maior densidade de nós, maior a quantidade de vizinhos possíveis.

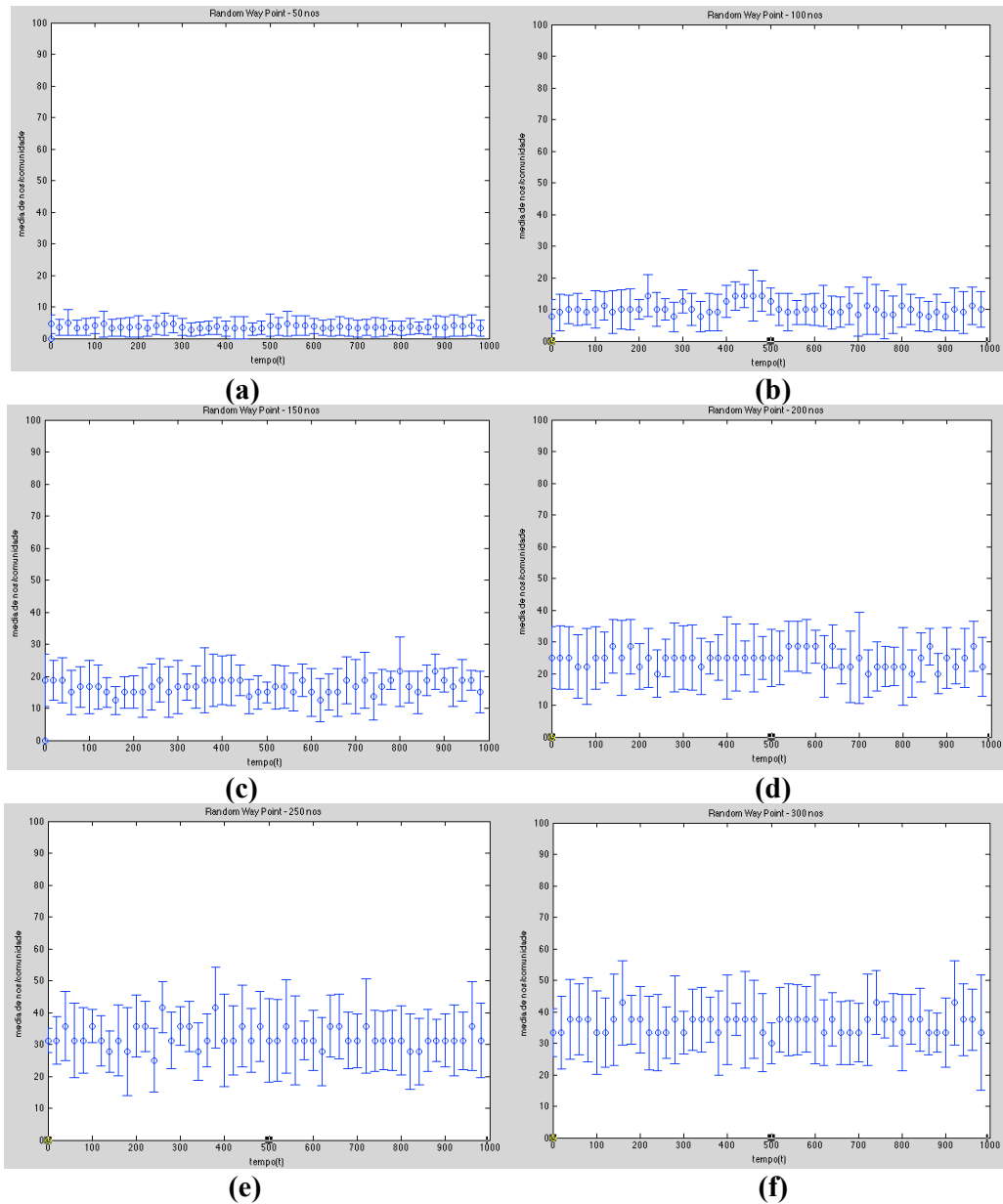
Ainda para os dois modelos, obteve-se modularidade  $0,7 < Q < 0,8$ , sendo bastante semelhantes as curvas para ambos. Assim, não se pode usar a modularidade para se diferenciar se entre um modelo aleatório ou em grupos pré-definidos, pois a métrica só indica quão melhor ou pior foi possível encontrar uma estrutura de comunidade.

Percebe-se, portanto, que através da modularidade não se pode tecer comparações entre os dois tipos de modelos, mesmo que diferentes pela características *a priori* de aleatoriedade x grupos.

#### ○ Número médio de comunidades

A fim de se tentar encontrar algumas diferenças entre a evolução da detecção de comunidades nos dois modelos utilizados, avaliou-se ainda o número médio de nós por comunidade em função do tempo de simulação.

Para o modelo de *Random Way Point*, variando-se o número de nós como nos experimentos anteriores, encontraram-se os seguintes resultados:



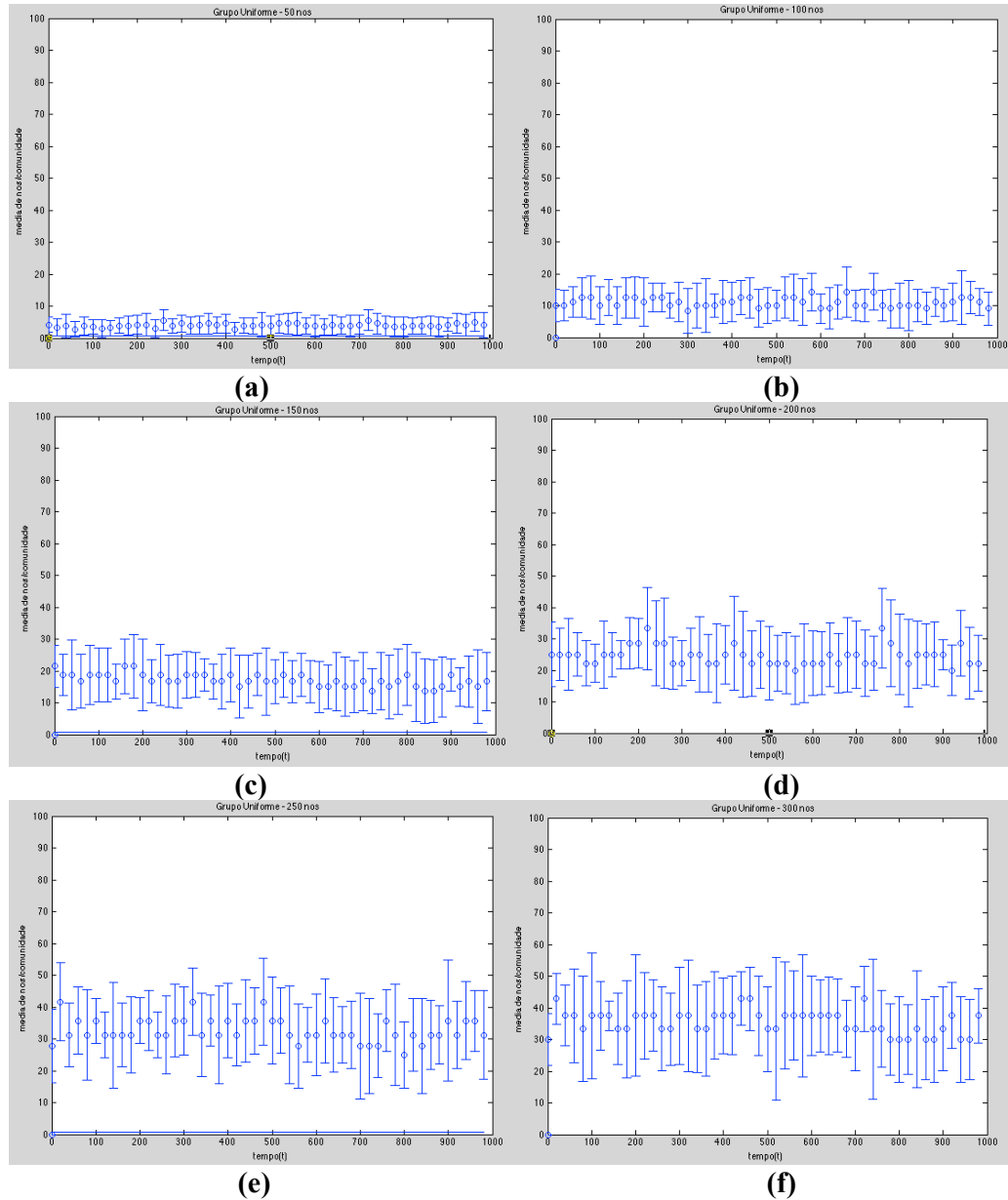
**Figura 27: número médio de nós por comunidade em função do tempo para o modelo *Random Way Point* para (a) 50 nós (b) 100 nós (c) 150 nós (d) 200 nós (e) 250 nós (f) 300 nós.**

Percebe-se que o número médio de nós por comunidade é praticamente linear com o aumento do número de nós, variando de 5 para 50 nós à 40 para 300 nós.

Para cada quantidade de nós, percebe-se que a densidade de nós em cada comunidade praticamente se mantém, o que indica que o número de comunidades

encontrado, desconsiderando nós isolados, é praticamente o mesmo durante o tempo de simulação. Entretanto, o que pode ser visualizado qualitativamente, é que a composição das comunidades varia.

Para o modelo de Grupo Uniforme, encontraram-se os seguintes resultados.



**Figura 28: número médio de nós por comunidade em função do tempo para o modelo Grupo Uniforme para (a) 50 nós (b) 100 nós (c) 150 nós (d) 200 nós (e) 250 nós (f) 300 nós.**

Os resultados são semelhantes aos encontrados pelo modelo aleatório, sendo o número de comunidades praticamente fixo para cada quantidade de nós. E,

analisando-se como a distribuição varia com o aumento do número de nós, percebe-se novamente a tendência linear.

Embora, para o modelo de Grupo Uniforme, a dispersão em relação a tendência central seja maior em relação ao *Random Way Point*, comparando grupos de mesma quantidade de nós, o que mostra uma maior heterogeneidade das comunidades encontradas em contraponto à homogeneidade esperada no modelo aleatório. Essa observação sugere uma maneira de buscar a diferença entre os dois modelos – aleatório e de grupo.

## CONCLUSÕES E TRABALHOS FUTUROS

### CONTRIBUIÇÕES E CONCLUSÕES

A importância das redes móveis (*ad hoc*) é indiscutível e as aplicações dessas redes nos mais diversos ambientes vêm crescendo a cada dia, assim, surge a necessidade de maior quantidade de pesquisas e testes, inicialmente por meio de simulações, como as realizadas nesse trabalho, principalmente pelo fator custo e complexidade de gerenciamento serem bem menores que a realização de testes em equipamentos reais.

O presente estudo possibilitou, através do desenvolvimento de um *Network Animator* que integra o JisT/Swans e o MATLAB, a fusão de algoritmos de detecção de comunidades em redes complexas estáticas com modelos de mobilidade de redes móveis, permitindo o desenvolvimento de diversos estudos quantitativos a partir das métricas utilizadas para o particionamento e ainda o cotejamento com análise visual qualitativa.

Como exemplo, estudaram-se a evolução da métrica de modularidade da detecção de comunidade em função do tempo para dois modelos de mobilidade diferentes quanto à presença de comunidades. Foi possível mostrar que essa métrica não pode ser utilizada para diferenciar os dois modelos.

Analizou-se ainda o número médio de nós por comunidade com o tempo, buscando diferenciar os dois modelos.

### TRABALHOS FUTUROS

A análise de métricas tanto para avaliar a qualidade de cada o particionamento, quanto para diferenciar diversos tipos de modelos pode ser bastante aprofundada, com maior quantidade de nós, outros modelos e padrões de mobilidade, além de estudo de diversos outros parâmetros que possam alterar o particionamento da rede.

Cabe ainda investigar a eficácia e eficiência de outros algoritmos de detecção de comunidade, avaliando o desempenho dos mesmo quando integrados com as redes complexas. Principalmente as características relacionadas às escalas micro e macro de detecção. Além da busca de algoritmos que consideram comunidades que se sobrepõem, simulando com maior verossimilhança as redes reais.

## **AGRADECIMENTOS**

Cabe agradecer até aqui o apoio à pesquisa concedido pela equipe responsável pela organização do PIBIC no ITA (Instituto Tecnológico de Aeronáutica), bem como a instituição que concede o apoio financeiro: o CNPQ.

Agradeço ainda o imenso apoio e orientação do estimado professor Carlos Henrique Costa Ribeiro e da coorientadora Luciana Babberg Abiuzy pela ajuda e compreensão do tão escasso tempo durante o ano e pelas produtivas discussões acerca dos diversos temas relacionados à pesquisa.



## BIBLIOGRAFIA

- [1] **Projeto de redes**, disponível em <[http://www.projetoederedes.com.br/artigos/artigo\\_redes\\_moveis\\_ad\\_hoc.php](http://www.projetoederedes.com.br/artigos/artigo_redes_moveis_ad_hoc.php)>, acessado em 8/12/2012.
- [2] Luciano da F. Costa, Francisco A. Rodrigues, Gonzalo Travieso and P. R. Villas Boas **Characterization of complex networks: A survey of Measurements**. *Advances in Physics*, Volume 56 (1), 167 - 242, 2007.
- [3] Francisco Aparecido Rodrigues. **Caracterização, classificação e análise de redes complexas**. Tese (Doutorado em Física) - Instituto de Física de São Carlos, Fundação de Amparo à Pesquisa do Estado de São Paulo. 2007.
- [4] M. E. J. Newman. **Detecting community structure in networks**. *The European Physical Journal B*, 38:321–330, 2004.
- [5] D. J. Watts and S. H. Strogatz. **Collective dynamics of smallworld networks**. *Nature*, 393(6684):440–442, 1998.
- [6] S. Boccaletti, V. Latora, Y. Moreno, M. Chaves, and D.-U. Hwang. **Complex networks: structure and dynamics**. *Physics Reports*, 424:175–308, 2006.
- [7] D. J. Watts and S. H. Strogatz. **Collective dynamics of smallworld networks**. *Nature*, 393(6684):440–442, 1998.
- [8] A.-L. Barabási and R. Albert. **Emergence of scaling in random networks**. *Science*, 286(5439):509–12, 1999.
- [9] Abramson, N. **Development of the alohanet**. *IEEE Transactions on Information Theory*, 31. 1985.
- [10] Carlos Alberto V. Campos e Luís Felipe M. de Moraes. **Modelos Markovianos de Mobilidade Individual para Redes Móveis Ad hoc**. Laboratório de Redes de Alta Velocidade – RAVEL Programa de Engenharia de Sistemas e Computação, COPPE / UFRJ.
- [11] Henrique S. dos Santos, Raqueline R. M. Penteado, Luiz Carlos P. Albini, Carmem Hara. **Um Modelo de Grupos para MANET**. Escola Regional de Banco de Dados (ERBD'2011), Novo Hamburgo-RS, 2011.
- [12] Sesay, S.; Yang, Z.; He, J. (2004) “**A Survey on Mobile Ad Hoc Wireless Network**”, *Information Technology Journal* 3(2), pp. 168-175.
- [13] S. Corson and J. Marker, “**Mobile Ad hoc Networking (MANET)**,” *RFC 2501*, jan 1999.
- [14] C. M. Cordeiro and D. P. Agrawal, “**Mobile Ad hoc Networking**”, *OBR Research Center for Distributed and Mobile Computing, ECECS*, 2002.

- [15] V. Davies, **“Evaluating Mobility Models within an Ad hoc Network”**, Master’s thesis, Colorado School of Mines Los Angeles, USA, 2000.
- [16] V. Davies, **“Evaluating Mobility Models within an Ad hoc Network”**, Master’s thesis, Colorado School of Mines Los Angeles, USA, 2000.
- [17] G. P. 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.
- [18] Guolong Lin, Guevara Noubir, and Rajmohan Rajaraman. **Mobility Models for Ad hoc Network Simulation**. In: Proceedings of IEEE INFOCOM 2004.
- [19] V. Davies. **Evaluating mobility models within an ad hoc network**. Master’s thesis, Colorado School of Mines, 2000.
- [20] Tracy Camp et.al. **A Survey of mobility models for mobile ad hoc network research, Wireless Communication & Mobile Computing: Special issue on mobile Ad hoc Networking: Research ,Trends and Applications**, v. 2, n. 5, p. 483-502, 2002.
- [21] **JiST/SWANS Homepage**, disponível em <http://jist.ece.cornell.edu/>, acessado em 3/3/2013.
- [22] B. Liang and Z. J. Haas. **Predictive distance-based mobility management for pcs networks**. In Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, NY, March 1999.
- [23] Z. J. Haas. **A new protocol for the reconfigurable wireless networks**. In: IEEE International Conference on Universal Personal Communication (ICUPC), pages 562–565, October 1997.
- [24] Araújo, F. P. **Implementação e Avaliação de Novos Modelos de Mobilidade para o Simulador JiST/SWANS..** Trabalho de Conclusão de Curso. (Graduação em Engenharia da Computação) - Universidade de Pernambuco. Orientador: Renato Mariz de Moraes, 2008.
- [24] Palla, G. I. Derényi, I. Farkas & T. Vicsek. **Uncovering the overlapping community structure of complex networks in nature and society**. Nature 435, 814-818, 2005.
- [25] Frank Kargl and Elmar Schoch. **Simulation of MANETs: a qualitative comparison between JiST/SWANS and ns-2**. In: MobiEval '07: Proceedings of the 1st international workshop on System evaluation for mobile platforms, San Juan, Puerto Rico, p. 41-46, 2007.
- [26] Elmar Schoch et.al. **Simulation of Ad Hoc Networks: ns-2 compared to JiST/SWANS**. In: First International Conference on Simulation Tools and

Techniques for Communications, Networks and Systems (SimuTools), Marseille, France, 2008.

[27] Zheng Kai; Wang Neng; Liu Ai-fang, **A new AODV based clustering routing protocol**. In: Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on , vol.2, no., pp. 728-731, 23-26 Sept. 2005.

[28] Vieira, V. F. ; **A. G. Evsukoff**. **A Comparison of Methods for Community Detection in Large Scale Networks**. In: 3rd Workshop on Complex Networks. Studies in Computational Intelligence (Print), v. 424, p. 75-86, 2012.

[29] Newman, M. E. J. & M. Girvan. **Finding and evaluating community structures in networks**. Physical Review E 69(2), 026113(1-15), 2004.

[30] Jeong, H., B. Tombor, R. Albert, Z N. Oltvai, & A. -L. Barabási (2000). **The large scale organization of metabolic networks**. Nature 607, 651-654.

[31] Alonso, R. L. Tese de doutorado: **Dinâmica de Partículas e Aprendizado Competitivo para Detecção de Comunidades em Redes Complexas**. Universidade de São Paulo, 2008.

[32] Pothen, A., H. Simon, & K. -P. Liou (1990). **Partitioning sparse matrices with eigenvectors of graphs**. SIAM Journal on Matrix Analysis and Applications 11(3), 430-452.

[33] Leon-Suematsu, Y.I., Yuta, K.: **Framework for fast identification of community structures in large-scale social networks**. In: Data Mining for Social Network Data, Annals of Information Systems, vol. 12, pp. 149–175. Springer US (2010)

[34] Newman, M.E.J.: **Modularity and community structure in networks**. Proceedings of the National Academy of Sciences of the United States of America 103(23), 8577–8582 (2006)

[35] Jain, A. K. and Dubes, R. C. (1988). **Algorithms for Clustering Data**. Prentice Hall.

[36] Girvan, M. and Newman, M. E. J. (2002). **Community Community Structure Structure in Social and Biological Biological Networks Networks**. Proceedings of the National Academy of Science USA, 99(12):7821–7826.

[37] Newman, M.E.J.: **Networks: An Introduction**, 1 edn. Oxford University Press, USA (2010).

[38] Clauset, A., Newman, M. E. J., and Moore, C. (2004). **Finding Community Structure in Very Large Networks**. Physical Review E, 79(066111).

- [39] Danon, L. Diaz-Guilera, A. Arenas, A.: **Effect of size heterogeneity on community identification in complex networks**. Journal of Stat. Mech.: Theory and Experiment 2006(11), 6.
- [40] Wakita, K., Tsurumi, T.: **Finding community structure in mega-scale social networks**. Analysis 105(2), 9(2007).
- [41] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: **Fast unfolding of communities in large networks**. Journal of Statistical Mechanics: Theory and Experiment 2008(10).
- [42] Pons, P. and Latapy, M. **Computing Communities in Large Networks using Random Walks**. Proceedings of the 20th International Symposium on Computer and Information Sciences, ISCIS'05, LNCS 3733, 284-293 (2005)..
- [43] Reichardt, J. and Bornholdt, S. **Statistical Mechanics of Community Detection**. Physical Review E, 74, 016110. (2006).
- [44] P. Ronhovde, Z. Nussinov. **Local resolution-limit-free Potts model for community detection**. Phys. Rev. E. 81, 046114 (2010).
- [45] **Internet:** <<http://math.nist.gov/javanumerics/jama/>>, acessado em 29/7/2013.
- [46] **Internet:** <<https://code.google.com/p/matlabcontrol/wiki/Walkthrough>>, acessado em 29/7/2013.