

UNIVERSIDADE DE LISBOA  
Faculdade de Ciências  
Departamento de Informática



**Sistema probabilista de edição-subscrição para  
redes móveis ad hoc**

**José Pedro Santos Manso Côrte-Real**

Mestrado em Engenharia Informática

2007



UNIVERSIDADE DE LISBOA  
Faculdade de Ciências  
Departamento de Informática



**Sistema probabilista de edição-subscrição para  
redes móveis ad hoc**

**José Pedro Santos Manso Côrte-Real**

Projecto orientado pela Prof. Dra. Dulce Domingos  
Orientador externo Prof. Dr. Luís Eduardo Teixeira Rodrigues

Mestrado em Engenharia Informática

2007



Este trabalho foi parcialmente suportado pela FCT com o co-financiamento do FEDER através do projecto MICAS Middleware para Sistemas Adaptáveis ao Contexto (POSC/EIA/60692/2004).



## Declaração

*José Pedro Santos Manso Côrte-Real*, aluno nº 29037 da Faculdade de Ciências da Universidade de Lisboa, declara ceder os seus direitos de cópia sobre o seu Relatório de Projecto em Engenharia Informática, intitulado "Sistema probabilista de edição-subscrição para redes móveis ad hoc", realizado no ano lectivo de 2006/2007 à Faculdade de Ciências da Universidade de Lisboa para o efeito de arquivo e consulta nas suas bibliotecas e publicação do mesmo em formato electrónico na Internet.

FCUL, 11 de Novembro de 2007

*Luís Eduardo Teixeira Rodrigues*, supervisor do projecto de *José Pedro Santos Manso Côrte-Real*, aluno da Faculdade de Ciências da Universidade de Lisboa, declara concordar com a divulgação do Relatório do Projecto em Engenharia Informática, intitulado "Sistema probabilista de edição-subscrição para redes móveis ad hoc".

Lisboa, 11 de Novembro de 2007



## Resumo

A evolução da Internet veio introduzir novos desafios no âmbito das aplicações distribuídas de grande escala, onde o uso exclusivo de comunicação ponto-a-ponto rapidamente se torna impraticável devido à sua incapacidade de escala com o número de utilizadores, e também devido à rigidez de operação a que este tipo de protocolo obriga. É neste contexto que o paradigma editor/subscritor emerge como uma solução viável para suportar aplicações distribuídas, garantindo uma desagregação entre os vários participantes no sistema, e permitindo que um editor gere informação que será encaminhada pelo sistema para os subscritores que expressarem interesse nesse tipo de dados.

Este trabalho foca o problema de suportar o paradigma editor/subscritor no âmbito das redes *MANet*. Estas redes caracterizam-se por operarem com nós móveis, estando sujeitas a reconfigurações topológicas imprevisíveis, introduzindo assim uma complexidade acrescida ao problema.

### PALAVRAS-CHAVE:

Redes *ad-hoc* móveis, Edição/Subscrição, *Gossip*, DSR



# Abstract

The evolution of the Internet introduced new challenges in the field of large-scale distributed applications, where typical point-to-point solutions became unfeasible due to their inability to scale with the number of users, and also due to the rigid operation that this sort of protocol enforces. In this context, the publish/subscribe paradigm presents itself as a viable solution to the current problems of distributed applications, ensuring a decoupling between the various participants in the system and allowing an editor to create information that will be routed by the system towards the subscribers which expressed interest in that particular information. This work addresses the problem of implementing the publish-subscribe paradigm in the scope of MANet networks. These networks are characterized by operating with mobile nodes, being subject to unexpected topological reconfigurations, thus introducing an added complexity to the problem.

## KEYWORDS:

Mobile Ad-hoc Networks, Publish/Subscribe, Gossip, DSR



# Conteúdo

<b>Lista de Figuras</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Redes MANet . . . . .	1
1.2 Paradigma Editor/Subscritor . . . . .	1
1.3 Motivação . . . . .	2
1.4 Contribuição . . . . .	3
1.5 Estrutura da Tese . . . . .	3
<b>2 Enquadramento</b>	<b>5</b>
2.1 Redes MANet . . . . .	5
2.1.1 Modelos de Movimento . . . . .	7
2.1.2 Encaminhamento em Redes <i>MANet</i> . . . . .	9
2.1.3 Aplicações Práticas de Redes MANet . . . . .	13
2.2 Sistemas Editor/Subscritor . . . . .	14
2.2.1 Modelo de Interação Básico . . . . .	14
2.2.2 Modelos de Subscrição . . . . .	15
2.3 Arquitectura de Sistemas Editor/Subscritor . . . . .	17
2.3.1 Camadas Funcionais . . . . .	17
2.3.2 Protocolos de Rede . . . . .	18
2.3.3 Infra-estrutura de <i>Overlay</i> . . . . .	19
2.3.4 Encaminhamento de Eventos . . . . .	23
2.4 Alguns Exemplos . . . . .	25
2.4.1 <i>Gryphon</i> . . . . .	25
2.4.2 <i>Scribe</i> . . . . .	25
2.4.3 SIENA . . . . .	26
2.4.4 <i>Semi-Probabilistic Publish/Subscribe</i> . . . . .	26
2.4.5 Protocolo Q . . . . .	28
<b>3 DSR-WALK</b>	<b>31</b>
3.1 Resumo . . . . .	31
3.2 Propriedades do Sistema . . . . .	33

3.3	Arquitectura . . . . .	34
3.3.1	Nível Rede . . . . .	34
3.3.2	Nível de Rede Sobreposta . . . . .	34
3.3.3	Protocolo de Edição/Subscrição . . . . .	43
<b>4</b>	<b>Avaliação</b>	<b>49</b>
4.1	<i>NS-2</i> . . . . .	49
4.2	Parâmetros de Configuração . . . . .	49
4.3	Métricas . . . . .	50
4.4	Cenário 1: Rede Estática . . . . .	51
4.4.1	Desempenho Previsto por Modelação . . . . .	52
4.4.2	Resultados da Simulação . . . . .	52
4.4.3	Diferenças entre o Modelo e a Simulação . . . . .	53
4.4.4	Rede Sobreposta Não Regular: Comportamento Emergente . . . . .	56
4.4.5	Escolha de Número de Percursos Aleatórios . . . . .	59
4.4.6	Rede Sobreposta Simétrica . . . . .	60
4.5	Cenário 2: Efeitos da Mobilidade e Carga . . . . .	61
4.5.1	Influência da Mobilidade . . . . .	63
4.5.2	Influência da Carga na Rede . . . . .	64
4.5.3	Comportamento numa Rede com Filiação Parcial . . . . .	64
4.6	Discussão . . . . .	65
<b>5</b>	<b>Conclusão</b>	<b>74</b>
5.1	Conclusões . . . . .	74
5.2	Trabalho futuro . . . . .	75
	<b>Bibliografia</b>	<b>82</b>



# Lista de Figuras

2.1	Rede <i>Ad Hoc</i> : raio de alcance. . . . .	5
2.2	Rede <i>Ad Hoc</i> : topologia. . . . .	6
2.3	Rede <i>Ad Hoc</i> : falha de elo de comunicação devido à mobilidade. . . . .	6
2.4	Rede <i>Ad Hoc</i> : reparação de elo de comunicação. . . . .	7
2.5	Rede <i>Ad Hoc</i> : Colisão. . . . .	7
2.6	Processo de junção a uma árvore multicast em MAODV. . . . .	12
2.7	<i>Broker</i> . . . . .	15
2.8	<i>Desagregação espacial</i> . . . . .	15
2.9	<i>Desagregação temporal</i> . . . . .	16
2.10	Decomposição funcional do paradigma editor/subscritor . . . . .	18
2.11	Rede <i>Super-Peer</i> . . . . .	19
2.12	Topologia em anel . . . . .	20
2.13	Topologia em polígono . . . . .	21
2.14	Topologia em árvore . . . . .	22
2.15	Semi-Probabilistic routing . . . . .	27
2.16	Funcionamento do protocolo Q . . . . .	29
3.1	Arquitectura da solução proposta . . . . .	32
3.2	Processo de junção à rede sobreposta. . . . .	38
3.3	Random-walks . . . . .	44
3.4	Funcionamento do protocolo de edição subscrição . . . . .	46
3.5	Funcionamento do protocolo de edição subscrição . . . . .	47
4.1	Distribuição geométrica . . . . .	52
4.2	Taxas de entrega com 3 editores 5 subscritores . . . . .	53
4.3	Efeitos das optimizações . . . . .	57
4.4	Rede com elementos na "periferia", e com um elemento popular . . . . .	58
4.5	Distribuição de mensagens numa rede de filiação total e de filiação parcial . . . . .	59
4.6	Distribuição de subscrições numa rede de filiação total e de filiação parcial . . . . .	60
4.7	Rede sobreposta assimétrica com <i>fanout</i> de publicação variável . . . . .	67

4.8	Rede sobreposta simétrica . . . . .	68
4.9	Rede sobreposta assimétrica (3 m/s) , 5 editores e 10 subscritores, 1 mensagem por segundo . . . . .	69
4.10	Entrega de pacotes com 3 editores e 5 subscritores . . . . .	70
4.11	Rede com 50 nós movendo-se a 20 m/s constantes . . . . .	71
4.12	Rede com 5 editores a publicar 10 mensagens segundo . . . . .	72
4.13	Rede com 250 nós . . . . .	73



# Capítulo 1

## Introdução

### 1.1 Redes MANet

A proliferação de dispositivos portáteis com capacidade de comunicação sem fios (por exemplo, 802.11a/b/g, etc...) tem incentivado a investigação na área de protocolos e aplicações para redes móveis *ad-hoc* (*Mobile Ad-hoc Networks* - MANets) [10]. MANets são redes sem fios *multi-hop* auto-configuráveis, onde os membros da rede (nós da rede) cooperam para garantir interconectividade. Neste tipo de rede as ligações são definidas pelos raios de comunicação de cada nó da rede. Quando dois nós se afastam, ficando fora do raio de comunicação um do outro, ocorre uma falha de ligação.

Dada a liberdade de movimento dos nós na rede e o seu comportamento tipicamente imprevisível, as falhas de comunicação podem tornar-se comuns neste ambiente de operação. É portanto necessário ter em conta alguns requisitos quando se desenha um protocolo para operar neste tipo de cenário. Um sistema destinado a operar numa MANet deve ser capaz de tolerar falhas de comunicação e ter em conta as restrições de processamento e memória típicas dos dispositivos móveis. Finalmente, a largura de banda disponível tende a ser também limitada, pelo que uma distribuição uniforme da carga de comunicação pelos nós da rede reduz o esforço individual de comunicação, processamento e consumo energético dos intervenientes.

### 1.2 Paradigma Editor/Subscritor

O advento da Internet obrigou a repensar os mecanismos de comunicação utilizados em sistemas distribuídos. Actualmente uma aplicação distribuída pode ter milhões de utilizadores dispersos geograficamente que partilham informação na qualidade de editores ou subscritores. Um bom exemplo deste tipo de aplicação é o sistema de distribuição de conteúdo *BitTorrent*<sup>1</sup>, onde milhares de utilizadores se interligam

---

<sup>1</sup><http://www.bittorrent.com>

através de um sistema de rastreamento que coordena o encaminhamento de informação.

Este aumento de escala de operação vem introduzir novos desafios para aplicações distribuídas, nomeadamente:

- A capacidade de um sistema suportar um número crescente de participantes sem uma perda de desempenho proporcional ao crescimento da rede (capacidade de escala);
- A tolerância a reconfigurações abruptas do conjunto de participantes (entrada e saída de clientes);
- A desagregação do emissor e do receptor, visto que num sistema com muitos utilizadores é vantajoso não ter que manter informação de todos os receptores ou emissores de dados;
- Finalmente, a capacidade de disseminar informação para um conjunto vasto de receptores de uma forma rápida e eficaz.

É neste contexto que surge o paradigma de comunicação assíncrona editor/subscritor [13], onde os emissores (editores) não necessitam de enumerar explicitamente os receptores (subscritores) das suas mensagens. Quando um editor publica um evento, o sistema de edição-subscrição é responsável por garantir o correcto encaminhamento da mensagem até todos os subscritores que a desejem receber. De modo a expressar interesse em eventos, um subscritor pode explicitar que categoria de eventos pretende receber, ou definir o conjunto de características que um evento deve possuir para que esteja interessado nele.

Existem essencialmente três tipos de desagregação que advêm da utilização de um sistema editor/subscritor: desagregação espacial, temporal e de sincronização. Como as decisões de encaminhamento são feitas pelo sistema conforme as características dos eventos produzidos, o editor e o subscritor não têm de se conhecer mutuamente (desagregação espacial). Adicionalmente, em certos sistemas é possível que eventos publicados permaneçam no sistema mesmo após a saída dos seus editores, estando acessíveis a subscritores que se liguem ao sistema posteriormente (desagregação temporal). Finalmente, a comunicação entre o editor e o subscritor é assíncrona, visto que o editor publica a informação para o sistema mas o subscritor irá apenas receber essa informação mais tarde, dependendo do funcionamento do protocolo utilizado (desagregação de sincronia).

### 1.3 Motivação

Como vimos, o paradigma editor/subscritor fornece uma importante desagregação entre emissores e receptores de informação. No contexto das redes MANet, onde os

intervenientes no sistema podem mudar aleatória e rapidamente, esta desagregação é benéfica, dado que um emissor pode não saber a todo o momento quais os possíveis receptores dos eventos por si gerados. No entanto, os sistemas editor/subscritor desenvolvidos para redes fixas não são facilmente adaptados a ambientes móveis, devido aos sistemas de encaminhamento de eventos que utilizam. A motivação do trabalho proposto reside na necessidade de se criar um sistema editor/subscritor adequado para funcionamento numa rede MANet que consiga operar em cenários de mobilidade imprevisível, e onde potencialmente apenas parte dos elementos da rede participam no sistema editor/subscritor.

## 1.4 Contribuição

O trabalho apresentado neste documento introduz uma nova aproximação à materialização do paradigma editor/subscritor em redes móveis, tendo como principais características os seguintes pontos:

- Funcionamento semi-probabilista - permitindo ao protocolo operar sem que seja obrigatória a existência de informação de encaminhamento;
- Capacidade de integração numa rede MANet em que apenas um subconjunto dos membros da rede utiliza o sistema editor/subscritor;
- Funcionamento contínuo face a falhas arbitrárias de nós da rede, desde que exista ainda conectividade entre os restantes membros;
- Distribuição uniforme de carga pelos nós;
- Capacidade de reconhecer alterações nas condições de operação da rede;

## 1.5 Estrutura da Tese

A tese está estruturada da seguinte forma:

**Capítulo 2:** Apresenta conceitos relevantes à temática em que o trabalho se insere, relativos a redes MANet e a sistemas Editor/Subscritor;

**Capítulo 3:** Contém a descrição do sistema DSR-WALK desenvolvido no âmbito deste trabalho;

**Capítulo 4:** Mostra uma avaliação do sistema DSR-WALK através de simulação;

**Capítulo 5:** Termina o documento com a apresentação do trabalho futuro e conclusão deste projecto.

# Capítulo 2

## Enquadramento

### 2.1 Redes MANet

As redes *MANet* (*Mobile Ad hoc NETWORK*), são redes *wireless* móveis capazes de operar autonomamente e sem qualquer tipo de infra-estrutura. Os nós da rede são móveis e capazes de interagir entre si através de comunicação sem fios. Cada nó tem um raio de alcance máximo (Figura 2.1), condicionado pelo tipo de equipamento que dispõe e por efeitos de interferência externa ao sinal. A intersecção dos raios de comunicação com outros nós gera elos de comunicação entre os mesmos (Figura 2.2).

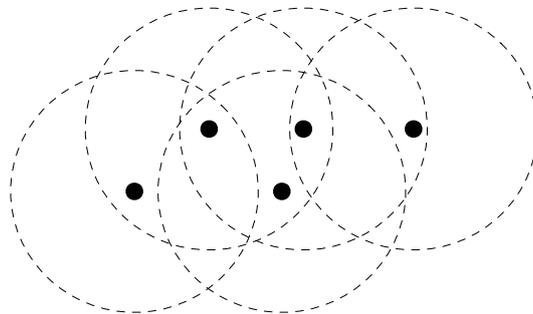
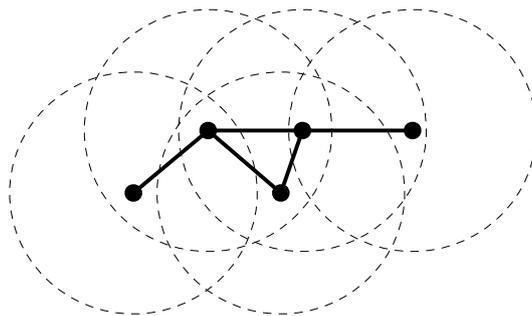
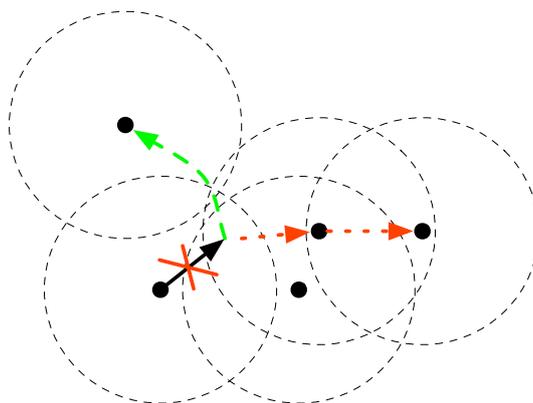


Figura 2.1: Rede *Ad Hoc*: raio de alcance.

Visto ser possível a existência de nós com raios de comunicação diferentes, é possível a criação de elos assimétricos (onde a comunicação é possível apenas num sentido).

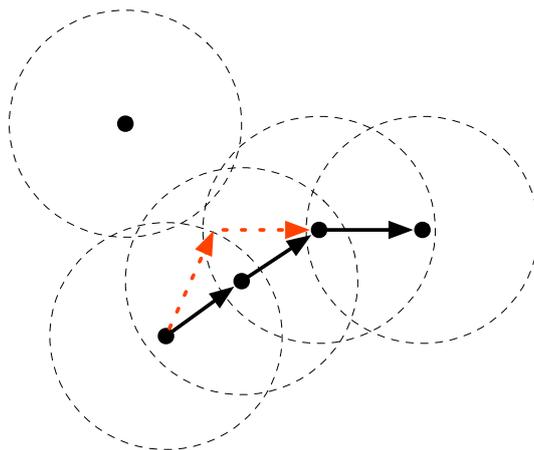
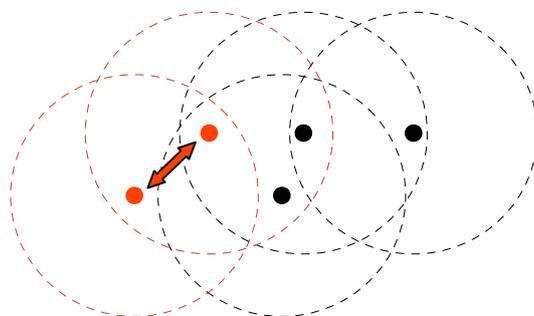
Numa rede *MANet*, a mobilidade dos nós pode gerar situações em que a validade dos elos entre clientes é bastante curta, dado que a movimentação de um nó para fora do raio de comunicação de outro origina uma quebra de ligação entre os dois nós (Figura 2.3). Neste tipo de ambiente, é importante a capacidade de se restabelecer comunicação através de ligações alternativas, como é exemplificado na Figura 2.4. Finalmente, a Figura 2.5 apresenta uma situação em que dois nós que se encontram

Figura 2.2: Rede *Ad Hoc*: topologia.Figura 2.3: Rede *Ad Hoc*: falha de elo de comunicação devido à mobilidade.

dentro do raio de comunicação um do outro, originam uma colisão de comunicação ao tentarem comunicar ao mesmo tempo. Numa rede sem fios, apenas um nó pode comunicar num determinado momento dentro do seu raio de comunicação, visto que duas mensagens simultâneas dentro da mesma área resultam num sinal indecifrável. Outras considerações a ter em conta quando se opera numa rede *MANet* são:

- Nós com limites na energia disponível;
- Capacidade de comunicação reduzida, caracterizada por pequena largura de banda e elevada latência;
- Qualidade de sinal variável, devido a interferências, pouca bateria, entre outras causas.

Tendo em conta as características apresentadas, pode-se concluir que as redes *MANet* apresentam novas possibilidades de aplicação interessantes, como por exemplo a rápida instalação de sistemas distribuídos sem infra-estrutura, mas apresentam também novos desafios e requisitos para os protocolos de comunicação utilizados neste ambiente.

Figura 2.4: Rede *Ad Hoc*: reparação de elo de comunicação.Figura 2.5: Rede *Ad Hoc*: Colisão.

### 2.1.1 Modelos de Movimento

Como já foi referido, numa rede *MANet* os nós móveis podem deslocar-se livre e imprevisivelmente, podendo a sua velocidade e direcção alterar-se inesperadamente. Tendo em conta a incipiência deste tipo de rede, e o limitado espectro de aplicações práticas actualmente disponíveis, a investigação nesta área é quase exclusivamente feita através de simulação, e um dos parâmetros fundamentais é o modelo de movimento dos nós. Um modelo de movimento representa um padrão gerado pela deslocação de nós móveis numa área pré-determinada, e considerando a falta de aplicações práticas já referida, é difícil prever o tipo de movimento que os membros da rede vão apresentar.

Com vista a solucionar este problema, vários modelos foram propostos para representar diversos tipos de movimento possível. Alguns destes modelos são puramente aleatórios, enquanto outros têm em consideração aplicações específicas, zonas de implementação particulares ou mesmo comportamentos sociais dos membros da rede. A utilização de um modelo de movimento apropriado é de extrema importância, visto que a deslocação de nós determina a estabilidade da rede, a sua topologia e

conectividade [23] [3] . No entanto, não existe ainda consenso sobre que modelos melhor capturam os padrões de utilização de redes móveis em cenários reais.

### Freeway

O modelo *Freeway* [23] representa o comportamento de nós móveis que se deslocam ao longo de uma via rápida. Este cenário tem como alvo, por exemplo, aplicações de disseminação de informação de tráfego ou de localização de um veículo numa estrada. As principais características deste modelo de movimento são:

- Restrição dos nós a uma via ou caminho;
- Velocidade dos nós é temporalmente dependente da sua velocidade anterior - ou seja, a velocidade a que um nó se desloca pode ser extrapolada de acordo com a sua velocidade anterior e os limites máximos de aceleração/desaceleração do nó;
- Se dois nós na mesma 'faixa' estiverem dentro da distância de segurança, a velocidade de um nó é limitada pela velocidade do nó que o precede.

Este modelo de movimento é portanto caracterizado por limitações geográficas estritas e por uma modelação de velocidade temporalmente dependente.

### Manhattan

O modelo *Manhattan* [23] emula o movimento de nós em ruas definidas por um mapa (ex: mapa da cidade de *Manhattan*). Este cenário é útil para modelar movimento de nós em áreas urbanas e tem como principais características:

- Mapa composto por ruas verticais e horizontais com dois sentidos;
- Os nós podem movimentar-se horizontal ou verticalmente;
- Quando um nó chega a uma esquina do mapa, decide aleatoriamente para que lado vira;
- A velocidade de um nó é limitada pela velocidade de um nó que o preceda na mesma rua.

Este modelo partilha várias semelhanças com o modelo *Freeway*, tendo como principal diferença a capacidade de um nó poder mudar de direcção graças à natureza menos restritiva do mapa que define os possíveis caminhos.

## RandomWaypoint

O modelo *RandomWaypoint* é uma aproximação simplista e aleatória que pretende representar o movimento de nós independentes através dos seguintes termos:

- Cada nó move-se ao longo de uma linha definida pela sua posição actual e um ponto de destino calculado aleatoriamente;
- Os pontos de destino são distribuídos uniformemente pela área geográfica de operação;
- Quando um nó inicia o movimento num caminho, escolhe uma velocidade aleatoriamente dentro do intervalo previamente definido;
- Em cada ponto de destino, um nó pode parar por um tempo previamente estipulado (tempo de pausa).

Este modelo, embora simples, não parte de pressupostos determinísticos como os outros modelos já apresentados, e como tal, permite modelar o movimento de nós de uma maneira completamente inesperada e aleatória. No entanto, existe o debate de que este movimento totalmente não determinístico não representará a realidade do movimento de nós numa rede *MANet*. Tendo em conta que ainda não existem sistemas largamente implementados com redes *MANet*, é ainda difícil prever se esta objecção ao modelo *RandomWaypoint* tem fundamento ou não.

### 2.1.2 Encaminhamento em Redes *MANet*

A disseminação de dados numa rede *MANet* é um problema complexo. A natureza volátil e imprevisível destas redes aumenta a complexidade inerente de um protocolo de encaminhamento. Nestas redes a comunicação é feita através de difusão sem fios. Em consequência, se um nó pretender enviar uma mensagem para outro nó no seu raio de comunicação, todos os nós dentro do seu raio de comunicação conseguem receber essa mensagem. Com o intuito de fornecer uma abstracção de comunicação mais rica que *one-hop broadcasts*, vários protocolos de comunicação foram desenvolvidos. Nesta secção apresentam-se três protocolos de comunicação para redes *MANet*, com especial destaque para o protocolo *Dynamic Source Routing* que será utilizado como primitiva de comunicação no trabalho.

De seguida apresentam-se dois protocolos concorrentes para efectuar *unicast* em redes *MANet*:

## AODV

AODV (*Ad hoc On Demand Distance Vector*) [19] é um protocolo de encaminhamento destinado a operar em redes *MANet* capaz de efectuar encaminhamento *Unicast* e *Multicast*. O protocolo é reactivo, ou seja, apenas constrói rotas entre nós quando desejado pelos nós de origem. Através da utilização de números de sequência é possível assegurar a validade de uma rota e garantir rotas sem ciclos. Quando um nó de origem deseja criar uma rota com um nó de destino, um pedido de rota é inundado pela rede. Nós que recebam este pedido actualizam a sua informação de encaminhamento para o nó de origem. O pedido de rota contém o número de sequência da rota mais recente que o nó de origem conhece para o nó de destino. Se um nó intermediário conhecer uma rota mais recente para o nó de destino, responde ao nó de origem com essa informação, caso contrário, retransmite este pedido. Se um nó receber um pedido de rota do mesmo IP que já tenha processado, este pedido é descartado. Quando um nó recebe uma resposta de rota para um destino, pode começar a utilizar essa rota. No entanto, se mais tarde receber uma resposta de rota com um número de sequência maior (ou com um número de sequência igual mas com um número de saltos até ao destino menor que a rota actual) essa rota é actualizada. Uma rota é apenas mantida enquanto estiver a ser utilizada, pelo que quando um nó de origem deixa de enviar dados para uma rota, esta irá eliminar-se por *time-out*. Se no entanto uma rota falhar enquanto estiver a ser utilizada, uma mensagem de falha de rota é devolvida ao nó de origem, e este terá de reiniciar o processo de descoberta da rota.

## DSR

DSR (*Dynamic Source Routing*) [14] é um protocolo de encaminhamento reactivo para redes sem fios *multi-hop*, capaz de operar em redes completamente auto-organizáveis. Este protocolo é composto por dois mecanismos essenciais: descoberta de rota (*Route Discovery*) e manutenção de rota (*Route Maintenance*). Descoberta de rota permite a um nó descobrir outro elemento da rede através do seu endereço, e criar um caminho entre ambos. O mecanismo de manutenção de rota verifica se a informação de encaminhamento de uma dada rota ainda é válida. Como já foi referido, o protocolo DSR é reactivo, pelo que se ajusta automaticamente ao número de rotas usadas pelos nós.

Segue uma explicação mais detalhada do protocolo:

- *Route Discovery* - Quando um nó S tenta enviar uma mensagem a um nó D e não conhece ainda um caminho entre ambos, o nó S iniciará um pedido de rota (ROUTE REQUEST). Este pedido vai ser disseminado pela rede através de inundação (memorizando os endereços dos nós que encaminham o pedido) até

encontrar o nó de destino. O nó D irá então responder ao nó S confirmando o seu pedido de rota e devolvendo uma lista de nós intermediários, que formam o caminho pelo qual o nó S o pode contactar.

- *Route Maintenance* - Quando um nó tenta enviar uma mensagem para outro utilizando um caminho conhecido, cada nó intermediário tem de confirmar que o próximo salto da mensagem recebeu o pacote com sucesso. Se este processo falhar, esse pacote será retransmitido (até um número máximo de vezes), e se mesmo assim falhar, um erro de rota é gerado e devolvido ao nó de origem. Quando um nó recebe um erro de rota, remove a informação pertinente da sua *cache* de encaminhamento.

Existem dois factores importantes a reter destes mecanismos. Primeiro, o processo de descoberta de rotas é muito dispendioso em termos de carga na rede, visto que inunda a rede com um pedido de procura de um nó. Segundo, a manutenção de rotas é ela também reactiva, pelo que um nó só sabe que uma rota sua falhou se a tentar usar (excepto se a validade da informação na *cache* expirar). Cada nó DSR contém uma *cache* de encaminhamento onde mantém informação em como contactar outros nós. Esta informação pode ser obtida através do mecanismo de descoberta de rotas, ou por *overhearing* de informação de encaminhamento a ser transmitida para outros nós. Adicionalmente, um nó pode receber múltiplas rotas para comunicar com um determinado nó quando efectua um pedido de rota, e como tal, as tabelas de encaminhamento contêm informação de encaminhamento redundante, o que aumenta a capacidade de tolerar reconfigurações topológicas da rede.

### Encaminhamento *Multicast* em MANets

Tendo em conta que numa rede *MANet* a comunicação ao mais baixo nível é feita por difusão, a implementação de protocolos de *multicast* (difusão em grupo) é facilitada pela própria natureza da rede. Nesta secção apresentam-se duas aproximações distintas para alcançar este fim. Em primeiro lugar aborda-se o protocolo MAODV [22] cuja operação é análoga à do protocolo AODV. De seguida, apresenta-se um protocolo probabilístico para efectuar *multicast* utilizando *unicast* DSR como primitiva de comunicação básica.

#### MAODV

Como já foi referido, o protocolo AODV permite operação em modo *multicast*. Rotas *multicast* são construídas de uma maneira semelhante às rotas *unicast* e na Figura 2.6 apresentam-se os passos envolvidos neste processo. Quando um nó deseja participar num grupo de comunicação, este dissemina na rede um pedido de rota com o endereço IP de destino desse grupo *multicast* (RREQ). Qualquer nó membro do grupo que

receba este pedido pode responder ao nó de origem (RREP), que ao receber esta resposta irá proceder à activação desta rota (MACT). Se um nó não activar a rota de *multicast* após efectuar um pedido de entrada num grupo, a sua entrada irá ser removida por *time-out* do nó de contacto que utilizou.

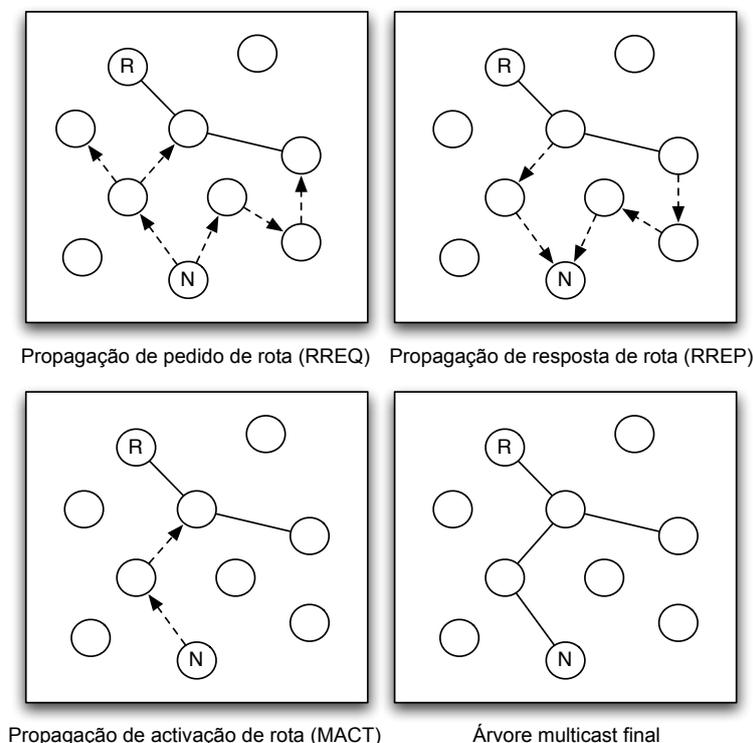


Figura 2.6: Processo de junção a uma árvore multicast em MAODV.

## RDG

RDG (*Route Driven Gossip*) [17] é um protocolo de disseminação de mensagens desenvolvido para operar em redes *MANet* tomando partido da natureza não determinística destas redes.

O protocolo RDG utiliza DSR como a sua primitiva básica de comunicação, e leva em consideração a existência de informação de encaminhamento na *cache* dos nós DSR de modo a minimizar o impacto de novos pedidos de rota na rede. Cada nó RDG mantém três vistas da rede:

- Vista activa (AView) - contém membros do grupo conhecidos para os quais existe informação de encaminhamento na *cache* DSR;
- Vista passiva (PView) - contém membros do grupo conhecidos para os quais não existe informação de encaminhamento na *cache* DSR; e

- Vista de remoção (RView) - contém membros que expressaram interesse em abandonar o grupo.

De modo a entrar num grupo, um nó tem de inundar a rede com um pedido de entrada. Cada nó do grupo que recebe este pedido adiciona o novo nó à sua vista activa, e devolve uma resposta com uma probabilidade proporcional ao tamanho estimado do grupo (calculado pelo número de pedidos de adesão recebidos). Quando o novo nó receber estas respostas, vai adicionar os identificadores dos nós contactados à sua vista activa. Desta maneira, o protocolo RDG cria efectivamente uma rede sobreposta [1] utilizando DSR como primitiva de comunicação ponto-a-ponto.

Quando um membro de um grupo deseja enviar uma mensagem, adiciona-a a um *buffer* temporário. De seguida, o comportamento do protocolo irá depender de três parâmetros:

- *fanout* - determina o número de destinos escolhidos da vista activa para cada mensagem enviada por este nó;
- *quiescence threshold* - determina o número de vezes que esta mensagem é transmitida; e
- *age threshold* - determina o limite de propagação de cada mensagem na rede.

Sempre que existam mensagens para enviar, um nó irá disseminar periodicamente mensagens para membros escolhidos aleatoriamente da sua vista activa, enviando também parte da sua vista em cada mensagem. De modo a minimizar o impacto da comunicação na rede, o protocolo RDG atribui pesos para os elementos das vistas de acordo com a distância em *hops* de cada membro.

O protocolo RDG leva em consideração o alto custo dos pedidos de descoberta de rota do protocolo DSR, e toma partida de informação da *cache* DSR de modo a minimizar este fenómeno. Favorecendo a utilização de membros da vista para os quais existe informação de encaminhamento, o protocolo consegue reduzir o *overhead* e latência de comunicação. O protocolo resultante efectua encaminhamento probabilístico sobre comunicação DSR, permitindo utilizar um protocolo de comunicação ponto-a-ponto e ao mesmo tempo reduzir o seu impacto na performance da rede.

### 2.1.3 Aplicações Práticas de Redes MANet

A crescente ubiquidade de dispositivos com capacidade de comunicação sem fios permite prever o aparecimento de novas aplicações no âmbito dos protocolos distribuídos. Possíveis aplicações para este tipo de rede são:

- Rápido estabelecimento de redes de comunicação em situações de catástrofe ou em cenários militares, sem recorrer ao uso de infra-estrutura;

- Criação de redes cooperativas para portáteis *low-cost* como o projecto OLPC [25];
- Coordenação e recolha de informação de tráfego rodoviário em redes veiculares (VANET - *Vehicular Ad hoc Network*).

Tendo em conta o tipo de aplicações previstas para este tipo de rede, o paradigma editor/subscritor apresenta-se como uma abstracção importante para facilitar a disseminação de informação por um subconjunto arbitrário de elementos da rede, de uma maneira transparente e escalável.

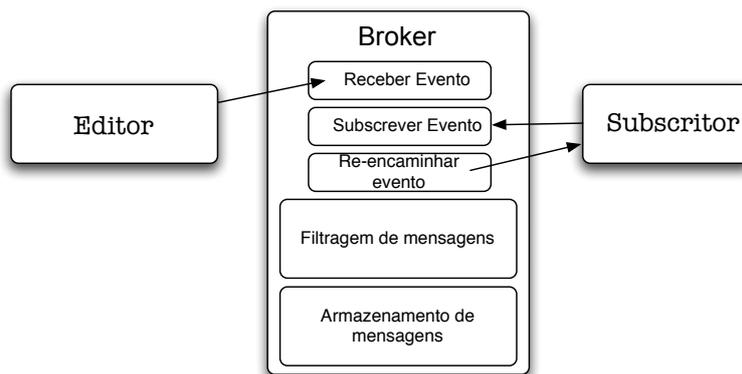
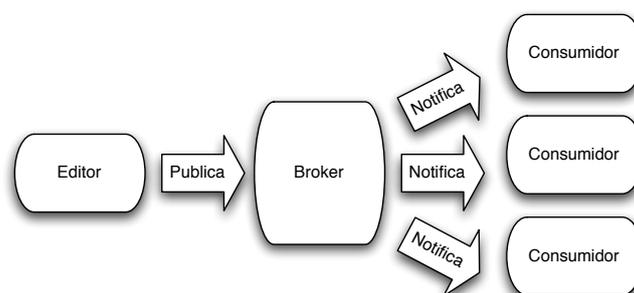
## 2.2 Sistemas Editor/Subscritor

Um sistema editor/subscritor fornece aos subscritores a capacidade de expressarem interesse numa classe ou padrão de eventos que uma vez publicados por um editor, serão posteriormente encaminhados para os subscritores correspondentes. Dado que existem várias arquitecturas que permitem alcançar esta funcionalidade, apresenta-se em primeiro lugar um esquema de interacção base para ilustrar o funcionamento do paradigma. Para finalizar esta secção, descrevem-se modelos mais complexos e com parâmetros de funcionamento distintos.

### 2.2.1 Modelo de Interacção Básico

O modelo básico para um sistema editor/subscritor (Figura 2.7) é constituído por um elemento central ao sistema (normalmente denominado *broker*) ao qual os editores de informação se ligam para publicar informação. Este *broker* pode depois armazenar essa informação e relacionar o tipo de evento com os pedidos de subscrição que possa ter recebido de subscritores. Esta aproximação embora tenha a desvantagem de o *broker* ser um ponto único de falha e um ponto de estrangulamento do sistema, permite ilustrar as principais características de um sistema editor/subscritor, nomeadamente a assincronia entre a publicação de um evento e a sua recepção. Esta assincronia verifica-se tanto em termos espaciais, como se ilustra na Figura 2.8 (o editor e o subscritor não têm de se conhecer), como temporalmente, como se ilustra na Figura 2.9 (o editor pode falhar após ter publicado a informação, mas como o *broker* armazena a informação recebida, pode mais tarde notificar os subscritores).

É também importante notar que este modelo permite que seja o *broker* a notificar os subscritores (*push-consuming*) ou os subscritores a interrogar o *broker* (*pull-consuming*).

Figura 2.7: *Broker*Figura 2.8: *Desagregação espacial*

Como se pode ver na Figura 2.7, um editor invoca o método *notify()* no *broker* para propagar informação para o sistema, os subscritores invocam *subscribe()* para expressar interesse num tipo de eventos, ou *unsubscribe()* para revogarem interesse num tipo de evento previamente subscrito.

### 2.2.2 Modelos de Subscrição

Existem diversas formas de associar a informação publicada num sistema editor/subscritor com as subscrições realizadas pelos subscritores. A principal diferença entre os modelos de subscrição existentes na literatura é a sua flexibilidade em descrever quais os eventos pretendidos. Quanto maior for a liberdade de definição dos padrões de emparelhamento entre publicações e subscrições, maior será a carga de processamento no serviço responsável por esta tarefa.

Apresentam-se de seguida os três tipos de subscrição predominantes nos sistemas editor/subscritor utilizados actualmente.

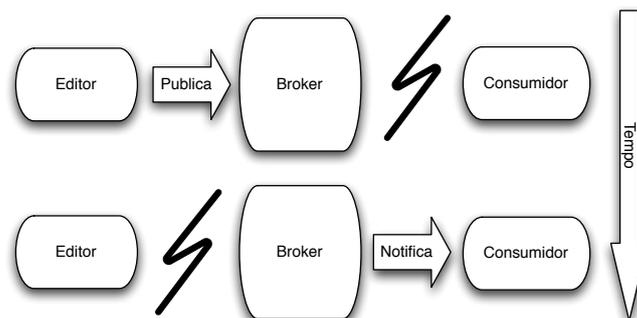


Figura 2.9: *Desagregação temporal*

### Subscrição baseada em tópicos

O modelo de subscrição baseado em tópicos tem sido utilizado em vários sistemas editor/subscritor, nomeadamente em soluções comerciais (ex: TIBCO[1999]). Neste modelo cada evento é etiquetado com o identificador de um tópico; os subscritores podem subscrever determinados tópicos, e o sistema encarregar-se-á de encaminhar mensagens publicadas para os destinatários interessados. Este modelo é similar ao modelo de comunicação em grupo: a subscrição a um tópico X pode ser vista como um pedido de junção ao grupo X, onde todas as mensagens com tópico X são entregues a subscritores interessados (tal como todas as mensagens propagadas no grupo X serão encaminhadas para os membros do grupo X).

Este modelo de subscrição é normalmente concretizado recorrendo a uma estrutura hierárquica, permitindo criar sub-tópicos.

Essencialmente a subscrição por tópicos projecta diferentes tópicos em 'canais' de comunicação. As principais vantagens desta abordagem são 1) a sua relativa simplicidade de concretização, 2) fácil interoperabilidade entre sistemas (um tópico pode ser uma *string*), e 3) pouca sobrecarga de processamento no mecanismo de emparelhamento de subscrições. Em contrapartida, este modelo oferece pouca flexibilidade para expressar interesse em padrões de eventos ou eventos caracterizados pelo seu conteúdo.

### Subscrição baseada em conteúdo

Em contraste com o modelo de subscrição por tópicos, o modelo de subscrição baseado em conteúdo oferece uma maior flexibilidade de subscrição. Neste modelo, os eventos não são classificados *a priori* em tópicos bem definidos, mas é o seu conteúdo que determina as condições de subscrição. Ou seja, um subscritor pode expressar interesse, por exemplo, em todos os eventos que contenham uma determinada chave, ou que satisfaçam alguma expressão regular.

### Subscrição baseada em tipos

A subscrição baseada em tipos é motivada pela observação que o modelo por tópicos normalmente agrupa eventos não apenas de conteúdo semelhante mas também de estrutura semelhante. Resumidamente, este tipo de subscrição utiliza o tipo de evento para fazer o emparelhamento da subscrição. Numa linguagem orientada aos objectos podemos aproveitar também a natureza hierárquica dos objectos para aumentar a flexibilidade deste modelo. Esta técnica permite uma maior integração com a linguagem e o *middleware*.

## 2.3 Arquitectura de Sistemas Editor/Subscritor

Neste capítulo pretende-se aprofundar 1) as principais características dos sistemas editor/subscritor, 2) os tipos principais de topologias de rede utilizadas e 3) os algoritmos de encaminhamento de eventos para diferentes cenários. Esta secção usou por base *surveys* por [16], [4], [13].

### 2.3.1 Camadas Funcionais

A concretização do paradigma editor/subscritor envolve vários aspectos, tornando a arquitectura num modelo complexo que, se decomposto em camadas funcionais, é de mais fácil compreensão. As quatro camadas apresentadas são as seguintes: infra-estrutura de rede, infra-estrutura de *overlay*, encaminhamento de eventos, e emparelhamento (Figura 2.10).

Apresenta-se de seguida a funcionalidade das camadas de protocolos de rede, infra-estrutura de *overlay* e encaminhamento de eventos. Dada a natureza do trabalho a desenvolver (que será discutida no Capítulo 3), não será necessário apresentar a problemática do emparelhamento.

### Modelo distribuído

Previamente apresentou-se um modelo centralizado para a concretização do paradigma editor/subscritor caracterizado pela existência de um ou mais *brokers* que servem de elo de ligação entre editores e subscritores. No entanto, como foi já referido, esta aproximação sofre de algumas limitações, nomeadamente a dificuldade em escalar com um número crescente de utilizadores. Adicionalmente, a utilização deste modelo gera pontos de falha críticos no sistema, sendo que se um *broker* falhar, uma parte significativa do sistema pode ficar sem serviço.

Nesta secção apresentam-se diferentes aproximações para a implementação deste

paradigma, umas utilizando redes de *brokers*, e outras utilizando uma aproximação distribuída onde a funcionalidade de um *broker* está distribuída por todos os intervenientes do sistema.

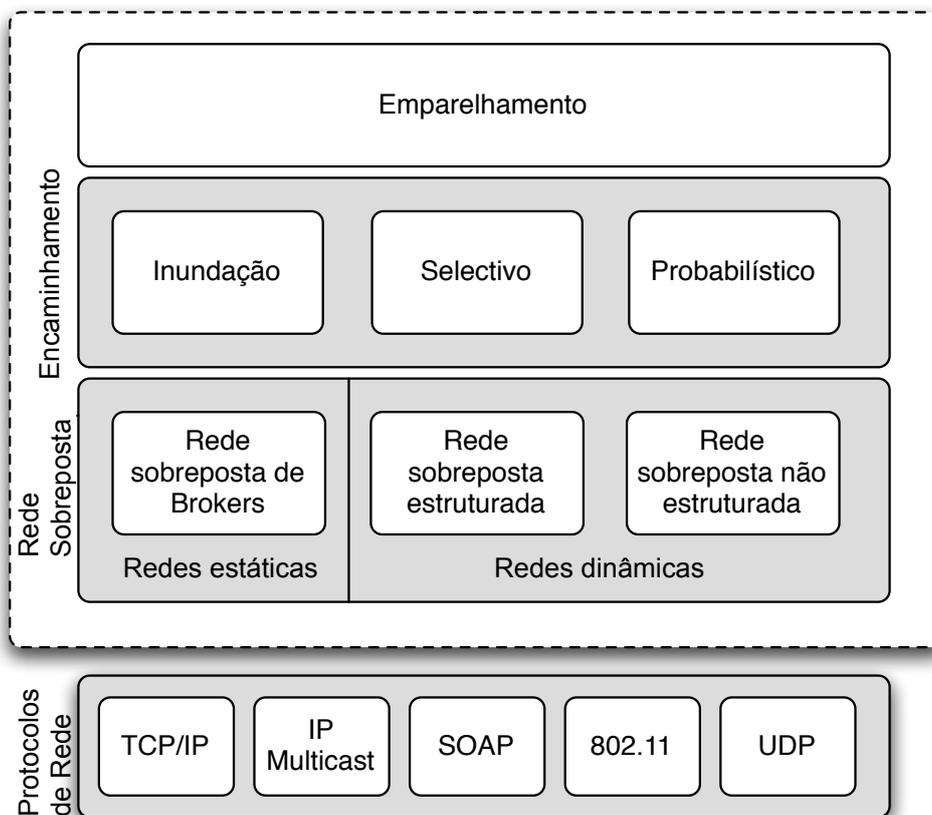


Figura 2.10: Decomposição funcional do paradigma editor/subscritor

### 2.3.2 Protocolos de Rede

Um sistema editor/subscritor necessita de um protocolo de comunicação na base da sua estrutura. Existem actualmente diversos protocolos adaptados a diferentes requisitos de funcionamento e cenários de operação, como por exemplo operação em redes fixas, móveis, com ou sem suporte para difusão. A escolha de um algoritmo base de comunicação, tendo em conta o contexto onde o trabalho se insere, restringe-se às opções actualmente existentes para comunicação *wireless* entre nós móveis (nomeadamente 802.11g).

### 2.3.3 Infra-estrutura de *Overlay*

Uma *overlay* (rede sobreposta) é caracterizada por uma rede lógica que tem por base uma rede física, sendo que a topologia da *overlay* é independente da organização da rede física. Existem essencialmente três classes de *overlay* utilizadas para suportar o funcionamento do paradigma editor/subscritor: 1) redes de *brokers*, 2) redes sobrepostas estruturadas e 3) redes sobrepostas não estruturadas. De seguida apresenta-se cada uma dessas classes, explicitando qual a topologia em que os nós se organizam, bem como as funções de cada nó, os requisitos de funcionamento e o tipo de qualidade de serviço que a *overlay* oferece.

#### *Broker Overlay*

Uma *overlay* de *brokers* é constituída por uma rede estruturada de *brokers* (ou servidores), que têm as funcionalidades típicas de *store-and-forward* apresentadas na secção 2.2.1. Estas redes são especialmente apropriadas para garantir a interligação de clientes através de uma WAN (*Wide Area Network*). Neste tipo de contexto, a infra-estrutura oferecida pela rede de servidores melhora as características de encaminhamento dentro de uma rede de grande escala. Tipicamente, os participantes do sistema ligam-se a diferentes servidores, resultando numa rede do tipo *Super-Peer* (Figura 2.11).

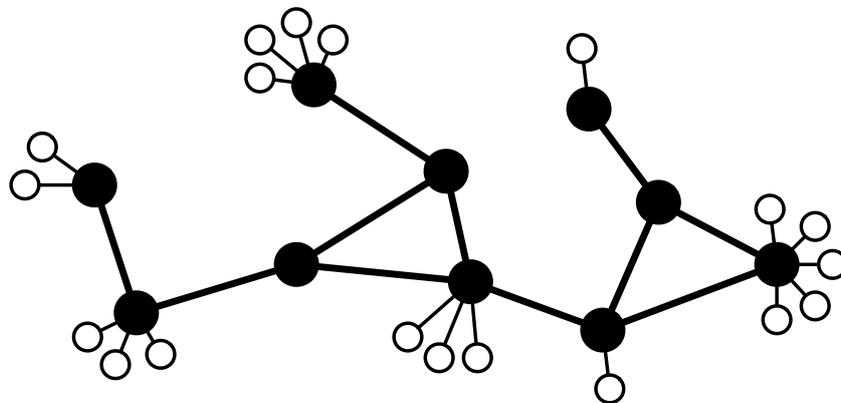


Figura 2.11: Rede *Super-Peer*

Dentro deste modelo é possível organizar os servidores em diferentes topologias, oferecendo diferentes tipos de serviços, como se apresenta de seguida. Na Figura 2.12 pode-se ver uma topologia em anel, onde os servidores têm apenas de conhecer os seus vizinhos imediatos.

Nesta topologia uma parcela dos clientes liga-se a cada um dos servidores. Esta técnica pode ser aperfeiçoada com estratégias como a organização de nós em diferen-

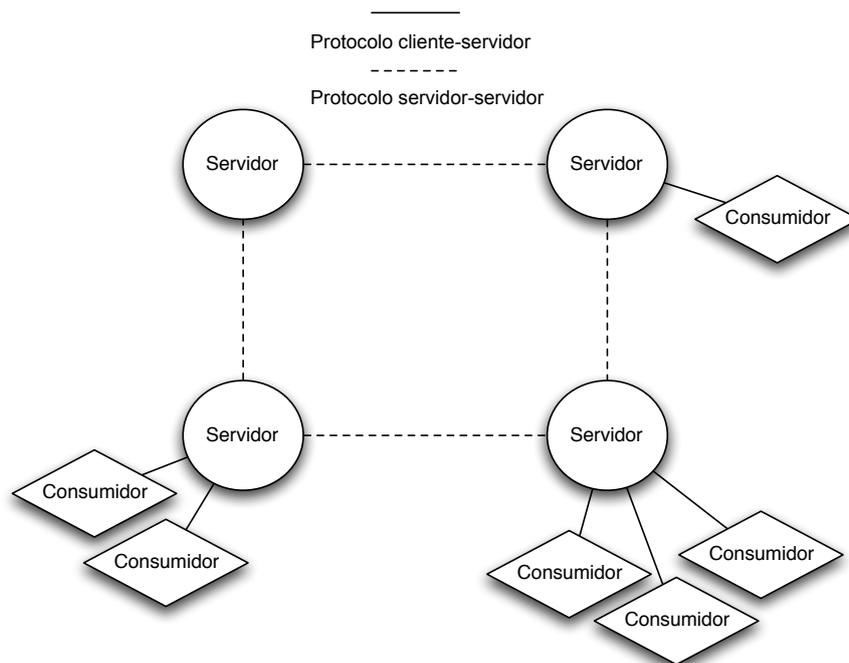


Figura 2.12: Topologia em anel

tes servidores conforme os seus interesses, de modo a facilitar depois o encaminhamento de eventos. Na Figura 2.13 observa-se uma extensão ao modelo de anel, onde existe uma maior flexibilidade de ligação entre servidores, o que pode permitir optimizações de encaminhamento (visto não ser necessário percorrer todo o perímetro do anel para atravessar a rede).

Na Figura 2.14 apresenta-se uma topologia em árvore. Nesta topologia é possível tomar partido das características intrínsecas das árvores, organizando os subscritores nas folhas de acordo com os seus interesses, e fazendo os editores publicarem informação na raiz da árvore, que será depois correctamente encaminhada pela sub-árvore apropriada em cada *hop*.

A principal vantagem deste tipo de *overlay* é a capacidade de criar uma infraestrutura que o protocolo de encaminhamento de eventos poderá tomar partido para maximizar o seu funcionamento. No entanto, este tipo de solução normalmente implica um esforço de configuração considerável, ou mesmo que a *overlay* seja disposta manualmente por técnicos. Em cenários estáticos com pouco *churn* (por *churn* entende-se a entrada e saída de participantes numa *overlay*), este tipo de solução é altamente vantajosa. Este modelo foi aplicado em vários sistemas, nomeadamente *TIB/RV*, *Gryphon*, *SIENA*, *JEDI*, *REDS* ...

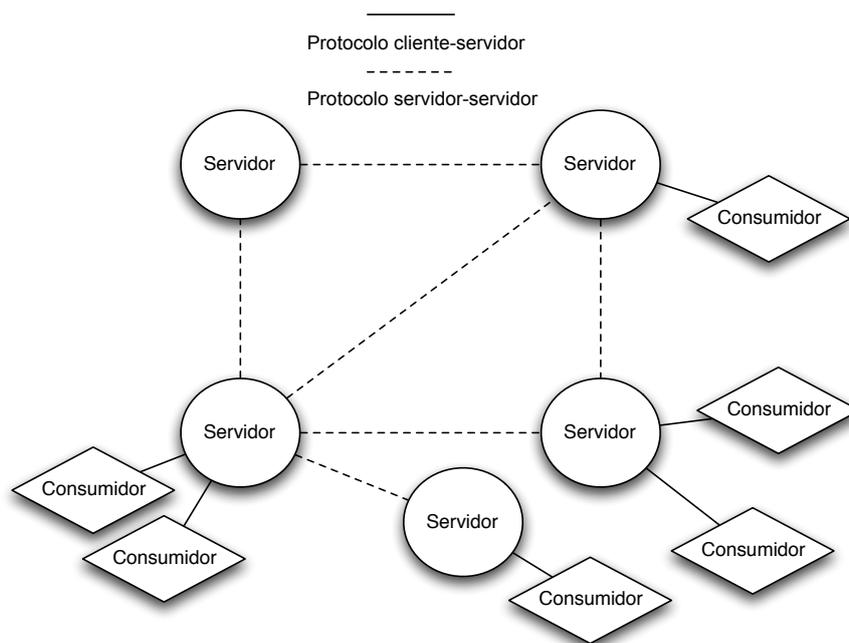


Figura 2.13: Topologia em polígono

### ***Rede sobreposta entre-pares estruturada***

Uma rede sobreposta entre-pares estruturada é constituída por uma interligação dos nós da rede através de elos virtuais, de modo a que a sua disposição permita uma maior eficiência do encaminhamento de conteúdos. Este tipo de *overlays* têm a vantagem de serem auto-organizáveis, sendo portanto mais facilmente aplicadas em ambientes mais dinâmicos, onde a entrada e saída de nós na rede não implica uma reconfiguração manual como nas *Broker Overlay* (visto que a própria rede sobreposta garante que a sua topologia se mantém coerente).

Dadas as propriedades acima apresentadas, foram desenvolvidos vários sistemas que concretizam este conceito (ex: *Pastry, Chord, Tapestry...*). A ideia chave na utilização deste tipo de redes para suportar um sistema editor/subscritor é aproveitar as características oferecidas por este tipo de sistemas, conseguindo assim retirar complexidade do problema da comunicação num ambiente dinâmico da implementação do sistema. Exemplos de sistemas existentes que tomam partido deste tipo de *overlay* são por exemplo *Bayeux* [28] e *Scribe* [9].

Já foi referido que as vantagens deste modelo são a capacidade de se auto-organizar, e como tal de se adaptar mais facilmente a ambientes dinâmicos. No entanto, em situações em que o *churn* seja significativo, o sistema pode sofrer de uma sobrecarga de mensagens na rede e de uma latência de comunicação impraticáveis, visto que a *overlay* pode passar mais tempo a tentar reconfigurar a topologia do que

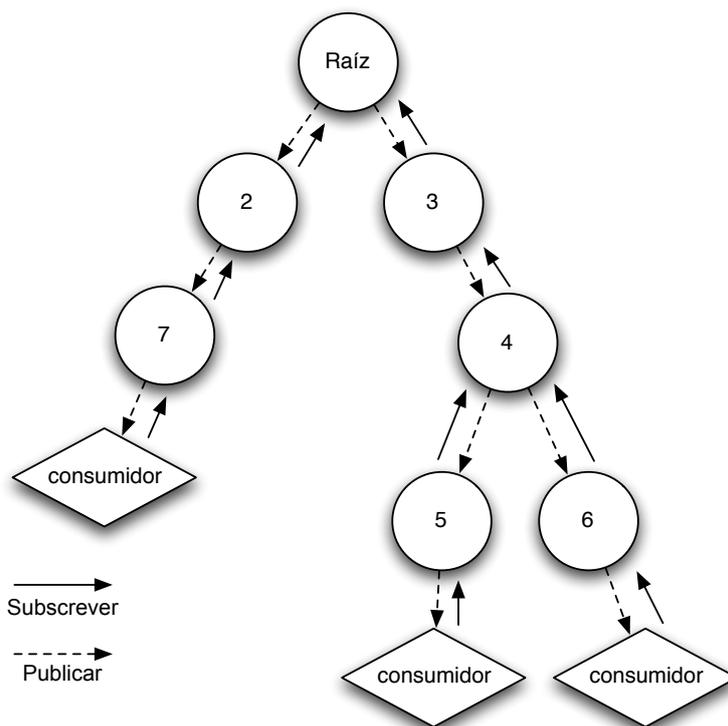


Figura 2.14: Topologia em árvore

a disseminar informação.

### Rede sobreposta entre-pares não estruturada

Quando a dinâmica de uma rede é suficientemente grande para comprometer as capacidades de regeneração de uma *overlay* estruturada, a solução passa pela utilização de uma *overlay* sem estrutura. Ou seja, na base do sistema editor/subscritor, não consideramos informação da topologia da rede para tomar decisões de encaminhamento.

Neste modelo é necessário recorrer a técnicas de inundação ou disseminação probabilística para suportar comunicação entre os nós, devido à falta de estrutura e imprevisibilidade do comportamento da rede (mudanças de topologia constantes, falhas inesperadas, entre outras causas).

Tendo em conta os pontos apresentados, é fácil perceber que este tipo de modelo é probabilístico e *best-effort* por natureza (dada a imprevisibilidade da rede), pois sem se fazerem algumas predefinições sobre a estrutura e o comportamento da topologia da rede, não é possível garantir sempre a correcta entrega de mensagens. No entanto, este modelo é mais adequado a situações de alta mobilidade/falha dos nós. Não existem actualmente muitos trabalhos que se enquadrem neste modelo, com a excepção de *Semi-Probabilistic Publish-Subscribe* [11].

### 2.3.4 Encaminhamento de Eventos

Um dos sub-componentes de um sistema editor/subscritor é o componente de encaminhamento de mensagens. Este pode ser realizado pela rede, pela rede sobreposta, ou directamente pelos participantes.

Existem várias categorias de algoritmos de encaminhamento como os algoritmos de inundação, encaminhamento selectivo e encaminhamento probabilístico, que se apresentam de seguida.

#### Inundação de eventos

Esta é uma solução trivial para a propagação de mensagens numa rede, onde cada nó envia mensagens para todos os seus vizinhos, e estes repetem o processo. Isto resulta numa inundação da rede com essa informação, sem a necessidade de recorrer a informação de encaminhamento e com taxas de entrega bastante altas. No entanto, esta solução gera muitas mensagens redundantes no sistema, o que a torna pouco eficaz.

Apesar das limitações desta aproximação, existem aplicações viáveis de algoritmos de inundação. Por exemplo, os subscritores podem inundar informação de subscrição na rede, tornando depois a ligação entre os editor e os subscritores determinística (os editores analisam as mensagens de inundação recebidas e utilizam os caminhos por elas atravessados).

As principais vantagens desta técnica são a simplicidade do algoritmo e a pouca utilização de memória e processamento nos nós. Em contrapartida, gera uma sobrecarga de mensagens muito grande, que resulta numa incapacidade de escalar com o número de mensagens.

#### Encaminhamento selectivo de eventos

Os algoritmos de encaminhamento selectivo têm como objectivo minimizar o número de mensagens trocadas para disseminar informação no sistema. Para alcançar este objectivo, o encaminhamento de mensagens tem de ser feito de modo a propagar informação apenas para os nós nelas interessados. Existem duas classes principais de algoritmos de encaminhamento selectivo:

- Encaminhamento baseado em filtros:

Esta classe de algoritmos cria uma *overlay* de difusão (normalmente uma árvore) que permite saber em cada *hop* por onde encaminhar mensagens de modo a chegar apenas a nós interessados. Desta maneira, consegue-se reduzir o número de mensagens na rede mas é necessário manter informação de encaminhamento nos nós, e em cenários dinâmicos pode-se tornar difícil manter as características da *overlay*.

- Encaminhamento baseado em ponto-de-contacto:

Encaminhamento baseado em ponto-de-contacto consiste na divisão da tarefa do encaminhamento de eventos em duas partes. Nós que desejem enviar dados encaminham essa informação até um ponto de contacto, de seguida, esse ponto de contacto reenvia a mensagem recebida para nós que tenham expressado interesse em receber esse tipo de dados. Um exemplo deste tipo de encaminhamento em sistemas editor/subscritor é *DHT Rendezvous Routing*, onde um editor envia dados para o nó com o identificador mais próximo do identificador da mensagem que deseja enviar. Da mesma maneira, subscritores enviam pedidos de subscrição para um nó com o identificador mais próximo do identificador que desejam subscrever.

### **Encaminhamento baseado em disseminação probabilística**

Encaminhamento baseado em disseminação probabilística consiste na disseminação de informação pela rede de uma forma aleatória. Por exemplo, um algoritmo probabilístico típico poderia ser tão simples como cada nó da rede reenviar mensagens recebidas para um subconjunto dos seus vizinhos, escolhidos aleatoriamente. No entanto, variantes deste tipo de protocolos podem aproveitar informação local do nó (potencialmente adquirida durante a sua execução) para encaminhar mensagens preferencialmente por um dado caminho. Este tipo de técnica denomina-se *Informed Gossip*. Exemplos desta técnica podem ser vistos por exemplo no protocolo *pmcast* [12].

Este tipo de protocolos têm a característica de não garantirem a entrega de mensagens enviadas dada a sua natureza probabilística. No entanto, compensam a sua falta de determinismo por serem protocolos completamente distribuídos (podem funcionar apenas com informação local de cada nó), escalarem bem com o número de clientes, e conseguirem níveis de entrega elevados.

## 2.4 Alguns Exemplos

Nesta secção apresenta-se uma pequena panorâmica sobre sistemas representativos registados na literatura, correlacionando as características de cada sistema com as classes de protocolos anteriormente apresentadas.

### 2.4.1 *Gryphon*

O *Gryphon* [24] é um sistema editor/subscritor com subscrição baseada em conteúdos desenhado para distribuição de grandes volumes de informação numa rede de larga escala. Para auxiliar o encaminhamento de mensagens, o *Gryphon* utiliza uma rede sobreposta de *brokers* organizados numa árvore de decisão, onde em cada nível da árvore é feito um refinamento da condição de emparelhamento, acabando por ter todas as subscrições individuais nas folhas da árvore. Adicionalmente, os vértices da árvore de decisão são constituídos por um *cluster* de servidores totalmente ligados para fornecer tolerância a faltas nas ligações.

Este sistema utiliza uma rede sobreposta de *brokers* e encaminhamento baseado em filtros, logo, pelas características previamente descritas, este sistema apresenta-se como uma proposta indicada para ambientes pouco dinâmicos e de grande escala. No contexto das redes *MANET*, este tipo de solução tornar-se-ia demasiado inflexível para suportar a natureza dinâmica deste ambiente.

### 2.4.2 *Scribe*

*Scribe* [9] é um sistema editor/subscritor baseado em tópicos completamente descentralizado e entre-pares, que toma partido das funcionalidades da rede sobreposta *Pastry* para garantir o encaminhamento. O *Pastry* é uma *Distributed Hash Table* [21] capaz de efectuar encaminhamento em redes *peer-to-peer* com  $O(\log N)$  passos de comunicação. O encaminhamento é feito através de um identificador e o *Pastry* encarrega-se de entregar a mensagem no nó com o identificador mais próximo do pretendido.

Para disseminar informação relativa a um tópico, o *Scribe* cria uma árvore de difusão com a raiz num nó de *rendez-vous*, cujo identificador é o numericamente mais próximo do tópico a disseminar (os tópicos são identificadores numéricos). A árvore é depois criada utilizando *reverse path forwarding* (uma utilização desta técnica é detalhada em [18]) dos pedidos de subscrição recebidos dos subscritores. Para expressarem interesse num tópico, os subscritores têm apenas de enviar uma mensagem de subscrição com destino para o tópico pretendido, visto que o nó ponto-de-contacto tem o identificador numericamente mais próximo e o *Pastry* encaminhará a mensagem para esse nó. Os editores podem posteriormente enviar mensagens para esse tópico, e ao chegar ao ponto-de-contacto, este disseminará esses dados pela árvore

criada.

As características principais do *Scribe* são: emparelhamento baseado em tópicos, arquitectura descentralizada e simplicidade de implementação (visto que grande parte da complexidade está contida na solução de um sistema de encaminhamento entre pares, o *Pastry* neste caso). Este sistema tem algumas vantagens relativamente ao *Gryphon*, como a capacidade de operar em ambientes entre-pares apenas com informação local, e também a capacidade de regeneração da rede sobreposta do *Pastry* perante *churn*. No entanto, esta solução continua a depender da utilização de uma rede sobreposta para fazer o encaminhamento, e embora o *Pastry* possa tomar partido de protocolos de transporte para relacionar vizinhos e endereços IP, esta solução não é aplicável num ambiente móvel.

### 2.4.3 SIENA

*SIENA* [7] é um sistema editor/subscritor com emparelhamento baseado em conteúdo que utiliza uma rede de servidores *SIENA* (*brokers*) onde os clientes se ligam. Utiliza encaminhamento baseado em filtros e técnicas de limitação de propagação de informação de subscrição para diminuir a sobrecarga de mensagens (como por exemplo, se um nó tentar subscrever a um tipo de eventos contido numa subscrição já feita nesse nó, o pedido é ignorado). O encaminhamento é auxiliado por anúncios de editores e subscritores que ajudam a formar caminhos para diferentes tipos de eventos (editores avisam os tipos de eventos que vão publicar e subscritores avisam em que eventos estão interessados).

Esta solução é claramente pouco adequada a um ambiente móvel. Tanto a construção da rede sobreposta de servidores, como a troca de mensagens para construção de caminhos, tornar-se-ia pouco viável se a topologia da rede se alterasse frequentemente.

### 2.4.4 *Semi-Probabilistic Publish/Subscribe*

*Semi-Probabilistic Publish/Subscribe* [11] é uma aproximação híbrida à problemática da disseminação de dados numa rede MANet, visto que depende de encaminhamento probabilístico e determinístico para alcançar uma alta taxa de entrega de mensagens, escalabilidade e tolerância a reconfigurações topológicas.

O protocolo pode ser dividido em dois blocos, primeiro o encaminhamento probabilístico que utiliza uma aproximação típica de *gossip*, e segundo, uma inundação de subscrição semelhante à apresentada em [26]. Uma das características principais deste protocolo é que ele não depende de uma rede sobreposta organizada para efectuar encaminhamento. Como tal, os nós da rede comunicam com os seus vizinhos que se encontrem dentro do raio de comunicação sem fios, através de disseminação. Esta aproximação tem vantagens e desvantagens que vamos detalhar nesta secção.

## Funcionamento do protocolo

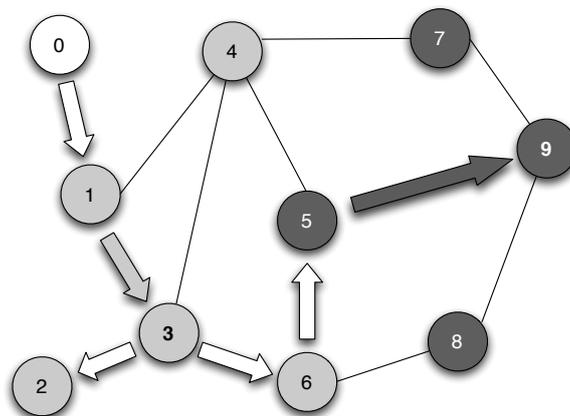


Figura 2.15: Semi-Probabilistic routing

A vista de um nó contém todos os membros da rede dentro do seu raio de comunicação sem fios, e como tal, os elementos da vista podem mudar rápida e imprevisivelmente. Os dois mecanismos básicos envolvidos no funcionamento do protocolo são:

- *Gossiping* de mensagens - utilizado quando um nó pretende publicar um evento e não possui informação de encaminhamento. A mensagem é enviada para um subconjunto de elementos da vista escolhidos aleatoriamente;
- Inundação de subscrição - utilizado por um subscritor para avisar nós vizinhos do seu interesse num determinado tipo de eventos. Esta inundação é limitada a um número de *hops* configurável.

Quando um editor deseja enviar um evento, verifica primeiro se possui informação de encaminhamento para o tipo de evento a enviar. Se não tiver essa informação, então o encaminhamento é puramente probabilístico, caso contrário, informação de subscrição é utilizada para dar prioridade à ordem dos elementos de *fanout*, enviando a mensagem primeiro para os nós com informação de encaminhamento determinística. Existem dois parâmetros principais envolvidos no funcionamento deste protocolo. Primeiro temos o horizonte de subscrição, que determina a extensão da inundação de informação de interesse em eventos dos subscritores. Por exemplo, se o horizonte de subscrição for 1, cada subscritor comunica com os seus vizinhos a 1 salto informando-os dos seus interesses. O segundo parâmetro principal é o factor de *fanout*, que determina o número de nós para os quais um nó publica um evento. A Figura 2.15 apresenta os mecanismos básicos envolvidos no protocolo. O nó 0 é

um editor, e os nós 3 e 9 são subscritores interessados nos eventos do nó 0. Neste exemplo, o horizonte de subscrição é 1, pelo que os vizinhos directos dos nós 3 e 9 possuem informação de encaminhamento determinística. Na figura, setas brancas representam encaminhamento probabilístico, e setas coloridas representam encaminhamento determinístico.

O nó 0 envia o seu evento probabilisticamente (visto que este não possui nenhuma informação de encaminhamento para este tópico) para o nó 1, visto este ser o seu único vizinho. O nó 1 por outro lado possui informação de encaminhamento que recebeu do nó 3. Logo, este nó prefere encaminhar a mensagem para o nó 3, do que para o seu outro vizinho, o nó 4. É também importante referir que, de modo a evitar ciclos, um nó não envia um evento para o vizinho que acabou de lhe encaminhar esse pacote.

Este protocolo possui algumas características que o distinguem de outras soluções editor/subscritor para MANets:

- O protocolo não depende da utilização de uma rede sobreposta estruturada para efectuar encaminhamento, visto que cada nó comunica com os seus vizinhos directos utilizando disseminação sem fios;
- O funcionamento do protocolo pode ser facilmente ajustado apenas com a alteração de dois parâmetros principais (horizonte de subscrição e *fanout*), permitindo reconfigurar facilmente o algoritmo de modo a adaptar-se a diferentes condições da rede. Por exemplo, se a rede ficar muito instável, o horizonte de subscrição pode ser diminuído e o factor de *fanout* aumentado, de modo a tornar o protocolo mais probabilístico e resistente a reconfigurações topológicas. Por outro lado, se a rede se encontrar mais estável, o protocolo pode utilizar um factor de *fanout* menor e aumentar o horizonte de subscrição.

No entanto, considerando que este protocolo utiliza a topologia física da rede como a sua *overlay*, está sujeito a certas limitações, como à necessidade de frequentes actualizações de informação de subscrição, e a uma distribuição de graus dos nós variável dependente do seu padrão de movimento.

### 2.4.5 Protocolo Q

Finalmente apresenta-se o trabalho detalhado em [2].

#### Criação de caminhos

A infra-estrutura implementada neste trabalho toma partido de uma rede sobreposta que opera sobre a primitiva de comunicação DSR, e recolhe informação de encami-

nhamento do protocolo DSR de modo a otimizar o seu funcionamento. Nesta rede sobreposta a informação de encaminhamento a nível aplicacional reflecte a informação de encaminhamento na *cache* DSR. Como tal, este protocolo considera qualquer nó DSR como participante na rede sobreposta, e não tem de implementar mecanismos de construção e manutenção da rede sobreposta.

De modo a disponibilizar um modo de editores e subscritores comunicarem, este protocolo executa os seguintes passos:

- De  $t$  em  $t$  segundos, editores de informação enviam mensagens de *advertisement* indicando o tipo de eventos que vão publicar. Cada nó que recebe esta pedido adiciona a informação pertinente à sua tabela de encaminhamento e reencaminha esta mensagem para os seus vizinhos. A informação mantida na tabela de encaminhamento contém: 1) o editor; 2) o tipo de eventos; e 3) o vizinho a um salto do qual recebeu esta informação. Esta informação é descartada após  $2*t$  segundos;
- Um subscritor interessado em receber um tipo de eventos regista-se como consumidor de eventos num vizinho a um salto do qual recebeu informação de *advertisement*. Este nó passa a ser um reencaminhador de eventos para o subscritor e repete o processo com o nó a um salto do qual recebeu a informação de *advertisement* até alcançar o editor. Desta maneira, através de um processo de *reverse-path-forwarding*, um caminho é criado entre o subscritor e o consumidor;
- Se existirem vários subscritores para um determinado tipo de eventos, estes actuam independentemente. No entanto, à medida que o sistema fica activo, nós responsáveis por reencaminhar eventos tentam agregar pedidos sempre que possível.

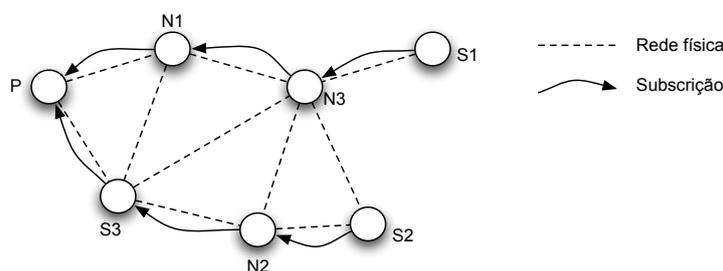


Figura 2.16: Funcionamento do protocolo Q

Na Figura 2.16 pode-se observar o processo de subscrição previamente apresentado. De notar que neste exemplo pressupõe-se que a informação de *advertisement*

do editores já foi propagada pela rede. Neste exemplo podemos observar uma rede com três subscritores (S1, S2 e S3) e um editor P.

### Reconfiguração da rede sobreposta

Visto que o sistema apresentado tem de operar no ambiente de uma rede MANet, a rede sobreposta diverge da rede real à medida que os nós se movem. O sistema Q reconfigura-se através da utilização de mensagens de *ping* enviadas periodicamente entre editores e subscritores. Este pacote é reencaminhada por nós intermédios da mesma maneira que um evento normal, e em cada salto o receptor da mensagem adiciona o seu identificador à mensagem e envia o pacote para todos os seus subscritores. Desta maneira, utilizando a informação contida nas mensagens de *ping*, os nós conseguem saber em que nível da cadeia de encaminhamento se encontram (o editor pode ser visto como o nível 0 que comunica com nós nível 1 que por sua vez comunicam com nós nível 2, etc...). Finalmente, um nó que receba uma mensagem de *ping* fica a saber o caminho actual para o editor, e consultando a tabela de encaminhamento consegue decidir se a cadeia de encaminhadores tem de ser reconfigurada. Comparando o número de saltos que a mensagem teve de efectuar na rede sobreposta com o número de saltos na rota para o endereço do editor na *cache* DSR, um nó consegue determinar se existe uma alternativa melhor para receber eventos do editor, e como tal, efectuar uma reconfiguração parcial da rede sobreposta subcrevendo com outro nó seu vizinho.

O sistema proposto nesta secção é semelhante ao sistema apresentado nesta dissertação relativamente ao modo como efectua encaminhamento. No entanto, embora o sistema Q também utilize uma rede sobreposta acedendo a informação da *cache* DSR, todo o protocolo tem um carácter determinístico, desde a construção de rotas entre editores e consumidores até à reconfiguração parcial da rede sobreposta acedendo a informação de encaminhamento na *cache* DSR.

# Capítulo 3

## DSR-WALK: Um sistema editor/subscritor para *MANets*

Neste capítulo descreve-se a nossa proposta para suportar edição/subscrição (E/S) em *MANets*. Primeiro apresenta-se um resumo da arquitectura adoptada incluindo o funcionamento básico das três camadas principais que compõem o protocolo: 1) DSR; 2) rede sobreposta e 3) edição/subscrição. De seguida expõe-se uma panorâmica referente às propriedades do sistema de E/S e finalmente, faz-se uma descrição pormenorizada da arquitectura.

### 3.1 Resumo

Materializar o paradigma editor/subscritor numa rede *MANet* é um problema complexo para o qual existem ainda poucas soluções concretizadas. A adaptação de protocolos de E/S desenvolvidos para redes fixas é não trivial dadas as características das redes *MANet*. É portanto necessário desenvolver novas soluções concebidas tendo em conta as particularidades deste tipo de redes.

O objectivo do trabalho proposto visa a criação de um sistema editor/subscritor capaz de operar em redes *MANet* dinâmicas com padrões de movimento imprevisíveis, onde possivelmente apenas um subconjunto dos elementos da rede participa no protocolo E/S. De modo a alcançar este objectivo, a solução proposta assenta numa arquitectura com três camadas apresentada na Figura 3.1: 1) um protocolo de comunicação *unicast* com tabela de encaminhamento; 2) uma rede sobreposta; e 3) um protocolo editor/subscritor semi-probabilístico.

O sistema apresentado utiliza o protocolo de encaminhamento ponto-a-ponto para redes *MANet*, DSR. Este protocolo permite que os participantes no sistema de E/S comuniquem entre si mesmo não estando em alcance rádio directo, desde que a rede *MANet* permaneça conexas. Adicionalmente, a *cache* de rotas do protocolo DSR

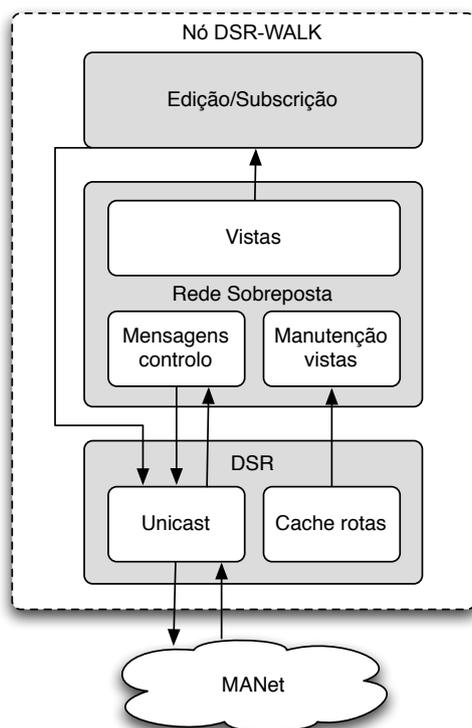


Figura 3.1: Arquitectura da solução proposta

permite obter informação relativa à disponibilidade de rotas para um determinado destino e permite também a um nó determinar a quantos saltos se encontra de outro nó na sua *cache*.

A rede sobreposta tem como função fornecer vistas parciais de membros da rede à camada de E/S, o que permite abstrair esta camada da topologia real da rede. Esta rede sobreposta toma partido da *cache* de encaminhamento do protocolo DSR de modo a otimizar o seu funcionamento.

Finalmente, no topo da arquitectura encontra-se a camada responsável pela execução do protocolo de E/S. Esta camada usufrui das abstrações fornecidas pela camada de rede sobreposta, e pela camada do protocolo DSR, logo, possui uma vista parcial de elementos da rede, bem como a capacidade de comunicar ponto-a-ponto com esses mesmos elementos. A publicação de eventos é feita através de percursos aleatórios na rede sobreposta (*random-walks*) [15], e a subscrição de eventos pode ser efectuada por meio de inundação limitada, ou recorrendo também à utilização de mensagens de percurso aleatório.

## 3.2 Propriedades do Sistema

O sistema desenvolvido possui um conjunto de propriedades resultantes da articulação das três camadas da arquitectura.

1) A escolha do protocolo DSR como primitiva de comunicação permite separar a complexidade inerente ao encaminhamento de mensagens em redes MANet do sistema de edição/subscrição. Ao mesmo tempo, o protocolo DSR fornece várias propriedades importantes para o sistema proposto, nomeadamente:

- **Integração numa rede MANet:** Utilizando o protocolo DSR, pode-se abstrair a camada aplicacional da forma de como é feito o encaminhamento na rede *MANet*. Adicionalmente, se todos os nós da rede utilizarem o protocolo DSR, mesmo não participando no sistema de edição/subscrição, contribuem para a conectividade do sistema de E/S, através da sua capacidade de encaminhamento de mensagens DSR.
- **Tolerar reconfigurações da topologia física:** Devido à mobilidade arbitrária dos nós típica de uma rede MANet, a topologia física da rede pode-se alterar imprevisivelmente. Quando isto acontece, a comunicação entre dois nós pode ser interrompida dado que a rota que os une deixa de ser válida. No entanto, o protocolo DSR consegue operar correctamente em cenários com mobilidade, pois reconfigura automaticamente as rotas existentes perante alterações na topologia da rede. Finalmente, considerando que todos os níveis do sistema utilizam comunicação DSR, estes beneficiam desta característica do protocolo.
- **Operar de modo auto-configurável:** A informação presente na *cache* DSR de cada nó permite à camada da rede sobreposta reconfigurar automaticamente as vistas de acordo com a informação de encaminhamento, nomeadamente acerca de quais os nós para os quais existem rotas. Deste modo, é possível manter uma topologia virtual auto-configurável que minimiza a geração de tráfego de controlo no protocolo DSR, evitando a utilização desnecessária de novos pedidos de rota.

2) Em segundo lugar, a utilização da rede sobreposta apresentada fornece uma importante abstracção da topologia dinâmica da rede subjacente, e permite controlar o grau externo <sup>1</sup> dos nós intervenientes. Adicionalmente, como já foi discutido, esta rede sobreposta minimiza o impacto de comunicação introduzido na rede pelo protocolo DSR.

---

<sup>1</sup>Número de pontos de contacto à distância de um salto que um dado nó conhece.

3) Finalmente, a execução de um protocolo de edição/subscrição probabilístico sobrejacente a uma rede sobreposta permite uma operação robusta e escalável, dado que o protocolo depende de pouca informação determinística para efectuar encaminhamento.

### 3.3 Arquitectura

Nesta secção apresentam-se pormenorizadamente os aspectos de concretização relativos às três camadas do sistema desenvolvido.

#### 3.3.1 Nível Rede

O sistema proposto neste capítulo depende da existência de um protocolo de comunicação ponto-a-ponto que possua uma *cache* de encaminhamento. Adicionalmente, este protocolo tem de ser capaz de operar em cenários com mobilidade, dado que todo o funcionamento do sistema depende da robustez da primitiva de comunicação do nível rede. A concretização apresentada neste trabalho utiliza o protocolo DSR como primitiva de comunicação, dado que este satisfaz os requisitos apresentados.

Como foi referido na secção 2.1.2, uma das principais limitações da utilização do protocolo DSR advém do custo do mecanismo de descoberta de rotas utilizado por este protocolo. Com o intuito de minimizar o impacto deste mecanismo, utiliza-se uma estratégia semelhante à do protocolo RDG. Uma descrição detalhada da implementação desta estratégia encontra-se na secção seguinte.

#### 3.3.2 Nível de Rede Sobreposta

De modo a concretizar a abstracção da rede sobreposta, assume-se que o processo pelo qual um novo nó se junta à rede depende da existência de um nó de contacto predefinido. Embora esta abordagem seja simplista, considera-se que esta problemática sai fora do contexto do problema proposto. Uma alternativa à solução apresentada seria a utilização de mensagens de inundação por parte de novos membros que se juntassem à rede (como é descrito em [17]).

Cada nó da rede possui duas vistas parciais dos elementos pertencentes ao sistema editor/subscritor, uma vista activa e uma vista passiva. A vista activa de um nó contém elementos para os quais existe informação de encaminhamento na *cache* DSR desse nó. A vista passiva contém elementos conhecidos, mas para os quais pode não existir informação de encaminhamento DSR. Estas vistas têm tamanhos máximos predeterminados, sendo que o tamanho máximo da vista activa é tipicamente inferior ao da vista passiva. A vista activa de um nó é actualizada conforme

a disponibilidade de informação de encaminhamento, ou seja, se um elemento da vista activa deixar de ter informação de encaminhamento na cache de rotas do nó, esse membro é removido da vista.

Finalmente, a rede sobreposta criada gera um único grupo de comunicação global (em oposição a soluções como a apresentada em [17]). No entanto, os membros do grupo têm apenas acesso a vistas parciais dos elementos que compõem a totalidade do sistema.

Um factor importante na construção de uma rede sobreposta é o grau interno de um nó (*in-degree*). Este factor representa a quantidade de elementos da rede que conhecem um determinado nó. Se um nó for conhecido por poucos elementos, a probabilidade de receber mensagens provenientes da rede sobreposta é baixa. No entanto, se um nó for conhecido por um número excessivo de elementos da rede, a comunicação na rede sobreposta torna-se imparcial, dado que nós com um maior grau interno têm uma maior probabilidade de receber mensagens.

Para a camada de rede sobreposta foram concretizadas duas soluções distintas, uma com vistas activas simétricas e outra com vistas activas assimétricas. Na rede sobreposta com vistas activas simétricas, se um nó A tem o nó B na sua vista activa, o nó B também tem o nó A na sua vista activa. Na rede sobreposta com vistas activas assimétricas não existe esta reciprocidade.

### **Rede sobreposta com vistas activas simétricas**

A utilização de vistas simétricas é uma abstracção importante na concretização da camada da rede sobreposta, dado que permite determinar o grau interno e externo de cada nó, pois estes são conhecidos por todos os nós que conhecem. No entanto, para assegurar a simetria das vistas activas, é necessária a troca de mensagens de controlo sempre que se adicione ou remova um nó da vista. Esta troca de mensagens resulta da necessidade de garantir que quando um nó adiciona ou remove um elemento da sua vista activa, o elemento visado também adiciona ou remove este nó.

O funcionamento desta rede sobreposta depende de quatro mecanismos base: 1) mecanismo de junção, 2) mecanismo de manutenção das vistas; 3) mecanismo de reparação; e 4) mecanismo de saída.

#### **Mecanismo de junção**

Como referido, assumimos que um novo nó N que pretenda juntar-se à rede sobreposta obteve previamente o identificador de um nó de contacto C. De seguida, o nó N envia uma mensagem de JOIN REQUEST para o nó C. Se o nó C não tiver a sua vista activa completamente preenchida, ele irá automaticamente adicionar o nó N à sua vista activa e devolver uma mensagem de recibo para o nó N. De seguida, o nó C irá gerar  $n$  percursos aleatórios de JOIN REQUEST em nome do nó N e encaminhá-los

para elementos da sua vista activa. O número de percursos aleatórios gerados é igual ao tamanho máximo da vista activa (parâmetro de sistema, Max Aview, equivalente para todos os nós). Se a vista activa do nó C não possuir elementos suficientes para enviar os pedidos (ou seja, se não estiver preenchida), os percursos aleatórios restantes serão enviados para elementos da vista passiva. Se ambas as vistas não contiverem elementos suficientes para satisfazer o número de percursos aleatórios a enviar, as mensagens em falta serão enviadas para nós já utilizados. Dada a natureza probabilística dos percursos aleatórios, duas mensagens enviadas para o mesmo nó poderão eventualmente percorrer caminhos diferentes. Cada mensagem de percurso aleatório tem um parâmetro de tempo de vida (*TTL - time to live*), que determina o número de saltos que a mensagem vai efectuar. Deste modo é possível controlar a propagação deste tipo de mensagens na rede.

Quando um nó recebe um percurso aleatório de JOIN REQUEST, este envia uma mensagem de PING ao nó N, que irá responder com uma mensagem similar. A utilização desta troca de mensagens permite antes de mais obrigar o protocolo DSR a criar informação de encaminhamento entre os dois nós caso ela não exista já. Por outro lado, permite também obter métricas actualizadas sobre a distância em *hops* entre ambos os nós. Esta solução pode parecer dispendiosa, na medida em que o nó a ser contactado poderia simplesmente adicionar o novo nó caso a sua vista o permitisse, e reencaminhar o pedido caso esta estivesse cheia. No entanto, este processo permite também otimizar a construção da rede sobreposta, visto que ao receber esta informação o nó contactado pode dar preferência a nós que se encontrem a menos *hops* dele. Adicionalmente, pode ocorrer a eventualidade de o nó contactado e o nó N não conseguirem comunicar num determinado momento devido a uma partição na rede. A não recepção de uma resposta a um pedido de PING pode significar esta ocorrência, e o nó contactado irá simplesmente reencaminhar o pedido para um elemento escolhido aleatoriamente da sua vista activa. Caso a resposta ao pedido de PING seja recebida, o nó contactado possui agora a informação necessária para tomar uma decisão sobre a aceitação do nó N para a sua vista activa. Em primeiro lugar, o nó contactado verifica se a sua vista permite a adição de um novo membro. Se a sua vista activa não estiver cheia, o processo é semelhante ao que ocorreu no nó C: o nó N é adicionado à vista activa do nó contactado, e uma mensagem de recibo é enviada para o nó N. Se no entanto a sua vista estiver cheia, é feito um teste para determinar se um elemento da vista activa é removido para dar lugar ao nó N. O teste efectuado compreende um cálculo probabilístico com um enviesamento (*bias*) relativo à distância do nó N ao nó contactado, ao *TTL* restante no percurso aleatório e ao número de elementos na vista activa do nó N:

$$distanceBias = (dsrheader - hopCount()) * K \quad (3.1)$$

$$ttlBias = (overlayhdr - ttl) * K \quad (3.2)$$

$$connectionBias = (overlayhdr - confactor) * K \quad (3.3)$$

$$probability = ((100) - (ttlBias + distanceBias + connectionBias)) \quad (3.4)$$

A equação 3.1 apresenta como é calculado o *bias* de distância entre os nós. A variável *srhdr* representa o cabeçalho DSR da mensagem recebida, e a função *hopCount()* devolve o número de *hops* que a mensagem efectuou para chegar ao seu destino. A equação 3.2 apresenta o cálculo referente ao *TTL* do percurso aleatório, retirando este valor do cabeçalho da mensagem referente ao protocolo da rede sobreposta (*overlayhdr*). Finalmente, é também retirado do cabeçalho do protocolo da rede sobreposta o número de elementos que o nó N tinha na vista activa aquando o envio da mensagem recebida. Este valor é utilizado para calcular o *bias* de ligação na equação 3.3. De seguida é calculado um valor entre 0 - 100 (equação 3.4), que permitirá efectuar um teste probabilístico para determinar a aceitação do JOIN REQUEST. Este teste tem como objectivo promover a aceitação de nós que se encontrem mais próximos, que tenham menos elementos na vista activa e cujo JOIN REQUEST esteja quase a expirar (*TTL* baixo).

A Figura 3.2 apresenta um exemplo de funcionamento do mecanismo de junção. Neste cenário utiliza-se uma vista activa máxima de três elementos para simplificar a explicação. No exemplo da Figura 3.2, os nós 1, 2 e 3 recebem o JOIN REQUEST do novo nó, mas apenas o nó 2 tem espaço na sua vista activa. Após a troca de mensagens de obtenção de métricas, o nó 3 resolve o teste probabilístico de aceitação do novo nó, e remove o nó 8 da sua vista. O nó 1 tendo falhado o teste de aceitação, reencaminha o pedido para um elemento da sua vista activa (mas nunca para o nó que lhe enviou este pedido). Finalmente, o último pedido resolve com sucesso no nó 5, que tem espaço na sua vista activa para mais um membro.

Observando o funcionamento deste mecanismo, é possível perceber que quando um nó decide remover um elemento da sua vista activa para adicionar um novo nó, este processo pode estar a particionar a rede. Por exemplo, observando a Figura 3.2, se o nó 2 remover o nó de contacto da sua vista, os nós 2 e 4 ficam isolados do resto da rede sobreposta. No entanto, dado que diferentes nós podem ter elementos semelhantes nas suas vistas activas, uma manutenção cuidada das vistas pode reduzir a probabilidade desta ocorrência. (Este problema voltará a ser abordado na secção 4).

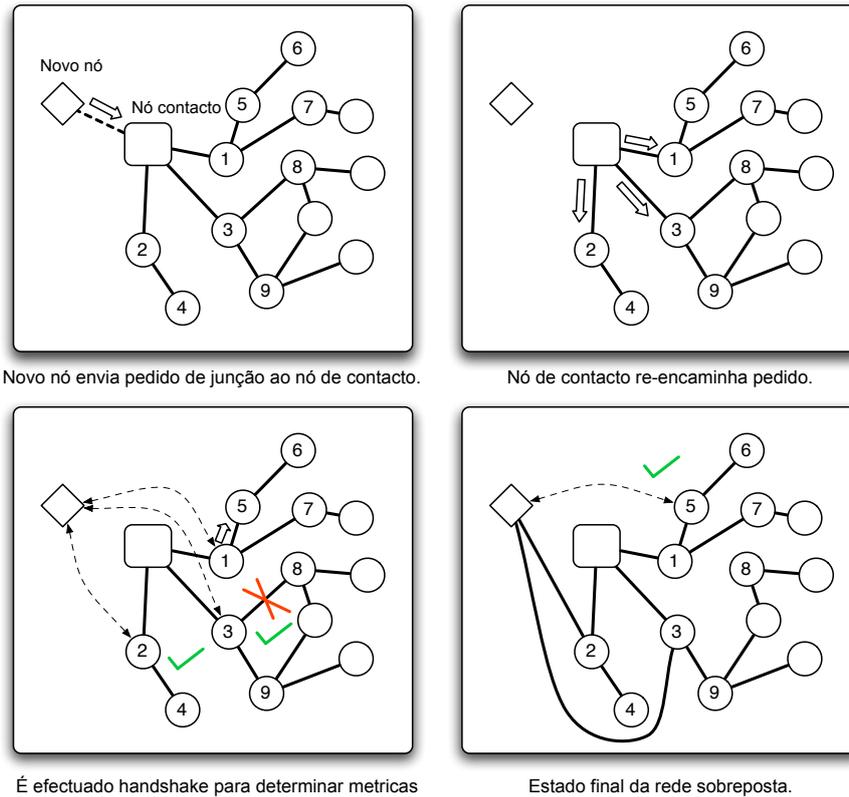


Figura 3.2: Processo de junção à rede sobreposta.

Considerando que o protocolo DSR não garante a entrega de mensagens, utiliza-se um passo final no processo de junção destinado a colmatar esta limitação. Quando o nó contactado envia uma mensagem de recibo para o nó N (3º passo da Figura 3.2), adiciona também esta mensagem à lista de pedidos de junção pendentes. Periodicamente, se o nó contactado não tiver ainda recebido uma mensagem de JOIN CLOSURE do nó N, este irá retransmitir a mensagem de recibo. Após  $n$  tentativas ou após receber a confirmação do nó N, a mensagem recibo é removida da lista de pedidos de junção pendentes. É importante no entanto notar que mesmo com esta aproximação a comunicação pode sempre falhar, resultando em vistas activas assimétricas entre o nó N e o nó contactado.

#### Mecanismo de manutenção das vistas

A aproximação trivial para a manutenção da vista activa passa pela remoção de elementos cuja informação de encaminhamento não se encontra disponível na *cache* DSR, e consequente adição destes elementos à vista passiva. Existem três situações nas quais uma entrada expira na *cache* DSR: 1) Quando não é utilizada há  $n$  segundos; 2) quando o nó tenta utilizar a rota dessa entrada e é devolvida uma mensagem de ROUTE ERROR; ou 3) quando um nó ouve informação relativa à falha dessa rota

por meio de outro nó. Como tal, podem acontecer situações em que rotas ainda sejam válidas, mas tenham sido removidas da *cache* DSR (por *timeout*), bem como situações opostas onde a rota já não é válida mas como ainda não foi utilizada, a entrada na *cache* ainda existe. Tendo em conta este comportamento, bem como o custo de reconfiguração da rede sobreposta (a remoção de nós da vista activa implica uma futura adição de outros nós - ambos os processos geram mensagens de controlo), a solução concretizada utiliza uma estratégia diferente da aproximação trivial apresentada. A nossa aproximação utiliza um parâmetro de *fitness* que reflecte a estabilidade da informação de encaminhamento para cada nó na vista activa. A aproximação utilizada opera do seguinte modo: periodicamente (de  $t$  em  $t$  segundos, sendo  $t$  um parâmetro configurável), cada nó verifica se os elementos presentes na sua vista activa ainda têm informação de encaminhamento na *cache* DSR. Cada elemento da vista activa para o qual exista informação de encaminhamento tem o seu campo de *fitness* incrementado. Caso um elemento não tenha informação de encaminhamento disponível, o seu *fitness* é decrementado. De seguida, todos os elementos da vista activa cujo parâmetro de *fitness* esteja abaixo de um valor predefinido são removidos da vista activa e adicionados à vista passiva. Utilizando esta estratégia consegue-se diminuir a instabilidade das vistas dos elementos da rede, dando preferência a elementos cujas rotas se têm mantido válidas por mais tempo. Esta solução, no entanto, poderá causar um encargo de mensagens acrescido na rede devido à utilização de rotas estagnadas/quebradas, já que podem existir elementos na vista activa sem informação de encaminhamento.

### Mecanismo de reparação

Quando o número de elementos da vista activa baixa até um valor predefinido (*minimum view threshold*), o mecanismo de reparação tentará periodicamente adicionar novos elementos à vista. Este mecanismo tem dois níveis de funcionamento:

- Primeiro nível de funcionamento - O mecanismo de reparação da vista activa verifica se algum elemento da vista passiva tem informação de encaminhamento na *cache* DSR. Embora o nó em questão não tenha solicitado a criação de rotas para os elementos da sua vista passiva, o protocolo DSR é capaz de recolher esta informação ao reencaminhar mensagens de controlo provenientes de outros nós (*overhearing*). Se existir mais do que um nó com informação de encaminhamento na vista passiva, o nó que tiver o menor valor de *hopCount* (número de saltos entre os nós) será escolhido. De seguida, este nó é removido da vista passiva e é-lhe enviado um percurso aleatório equivalente ao apresentado no mecanismo de *join*. O processo é em tudo semelhante ao mecanismo de *join* a partir deste ponto.
- Segundo nível de funcionamento - Se um nó ficar sem nenhum elemento na

vista activa, este reinicia o processo de *join* através do nó de contacto, como se se tratasse de um novo elemento na rede sobreposta.

### Mecanismo de saída

Considere-se um nó A com um nó B na sua vista activa. Se a rota DSR entre A e B for quebrada, o nó B é removido da vista activa de A apenas quando o seu valor de *fitness* alcançar um valor inferior a um mínimo predeterminado. Este mecanismo depende da capacidade do protocolo DSR de reflectir o estado real da rede na *cache* de encaminhamento do nó A. Alternativamente, o nó B pode ser removido da vista activa de A quando A deseja acrescentar um novo nó à sua vista activa, e esta se encontre preenchida.

Quando um nó remove um elemento da sua vista activa, ambos os nós envolvidos efectuam uma troca de mensagens para se excluírem mutuamente de modo a manter as vistas simétricas. Este é outro ponto do protocolo onde as vistas podem perder simetria, dado que o protocolo DSR não garante a entrega de mensagens. De modo a minimizar este efeito é possível utilizar uma solução semelhante à utilizada para a finalização do processo de junção (manter uma lista de pedidos de *leave* pendentes e reenviar um recibo  $n$  vezes até receber confirmação). No entanto devido a restrições de tempo, esta optimização não foi concretizada.

Pode parecer contra-intuitivo enviar mensagens de controlo para um nó para o qual já não exista informação de encaminhamento, dado que ao enviar esta mensagem o protocolo DSR cria um rota válida para o elemento a remover. No entanto, visto que o processo de remoção de nós visa apenas remover elementos cujas rotas não tenham permanecido estáveis durante um período de tempo, é expectável que a rota do elemento a remover (mesmo tendo actualmente informação de encaminhamento válida) venha a falhar em breve. Logo, a remoção deste elemento pode vir a beneficiar o nó em questão, pois este irá remover um elemento da sua vista activa cuja ligação se tem demonstrado pouco fiável, criando espaço para a adição de outro nó com uma ligação potencialmente mais estável.

### Vantagens/Desvantagens

Como foi já referido, a utilização de vistas simétricas permite melhorar a conectividade da rede sobreposta, dado que o grau interno e externo de cada nó é determinado pelo número de elementos que cada nó tem na sua vista activa. No entanto, de modo a assegurar a simetria entre as vistas activas, este protocolo introduz um custo acrescido devido à troca de mensagens de controlo necessária para os mecanismos de junção e saída de nós das vistas activas. Finalmente, dado que um nó pode remover elementos aleatórios da sua vista activa para adicionar novos contactos, a rede sobreposta pode ficar particionada.

### Rede sobreposta com vistas assimétricas

Nesta aproximação todos os nós continuam a ter uma vista activa e uma vista

passiva, e não existe simetria entre os elementos das vistas activas. A dimensão destas vistas é limitada apenas pelo tamanho máximo da tabela de encaminhamento do protocolo DSR, o que resulta em tamanhos de vistas superiores aos utilizados na aproximação anterior. Não existindo a necessidade de manter simetria entre as vistas activas, todas as mensagens de controlo relativas a pedidos de junção e de remoção de nós da vista activa deixam de ser necessárias. Finalmente, sempre que um nó recebe uma mensagem de outro nó que ainda não conhece, este adiciona o novo nó à sua vista activa caso tenha actualmente informação de encaminhamento para ele, ou à sua vista passiva caso contrário.

Embora esta aproximação não garanta uma conectividade equilibrada (dado que o grau interno e externo dos nós não é controlado), ao se prescindir desta abstracção consegue-se reduzir a complexidade e o custo da camada de rede sobreposta.

### Mecanismo de junção

Quando um nó  $N$  deseja juntar-se à rede sobreposta, este envia um JOIN REQUEST a um nó  $C$  que serve de ponto de contacto. Ao receber este pedido, o nó  $C$  adiciona o novo nó à sua vista activa, e devolve uma mensagem com uma parcela dos elementos das suas vistas. O número de elementos devolvido é determinado por um parâmetro configurável, e os elementos são escolhidos aleatoriamente da vista activa e passiva. Finalmente, o nó  $C$  envia  $n$  *random-walks* para elementos da sua vista activa que vão adicionar o novo nó às suas vistas, e vão também devolver ao nó  $N$  um subconjunto dos elementos das suas vistas.

O número de percursos aleatórios gerados pelo nó  $C$  e o seu *TTL* vão determinar a percentagem de nós da rede que o nó  $N$  vai conhecer (grau externo) bem como a quantidade de nós da rede que vão conhecer o nó  $N$  (grau interno).

### Mecanismo de manutenção das vistas

Periodicamente (de  $t$  em  $t$  segundos) cada nó verifica se os elementos de ambas as vistas possuem informação de encaminhamento na *cache* do protocolo DSR. Os elementos que não tenham informação de encaminhamento na *cache* são colocados na vista passiva, e os elementos com informação de encaminhamento na vista activa. Dado que quando um nó adiciona um novo elemento à sua vista activa não precisa de o contactar, toda a manutenção de vistas é feita localmente.

### Vantagens/Desvantagens

A principal desvantagem desta aproximação é a falta de simetria entre elementos das vistas activas que pode resultar numa conectividade da rede sobreposta menos equilibrada (dado que diferentes nós podem ter graus internos e externos muito diferentes). Quando isto acontece, certos nós podem raramente receber mensagens, enquanto outros podem ser inundados com mensagens redundantes (devido à sua excessiva popularidade - grau interno alto). Embora a simetria entre vistas activas

permita uma melhor conectividade, o seu custo de operação pode-se tornar incomportável. No caso da rede sobreposta com vistas assimétricas, consegue-se retirar grande parte da complexidade desta camada e ao mesmo tempo diminuir o *overhead* gerado pelas mensagens de controlo. Adicionalmente, a validade da correcção dos elementos da vista activa está intrinsecamente relacionada com a frequência de utilização da primitiva DSR. Dada a natureza reactiva do protocolo DSR, sempre que um nó envia mensagens para elementos da sua vista activa, a validade da informação de encaminhamento é renovada.

### 3.3.3 Protocolo de Edição/Subscrição

Nesta secção apresenta-se a camada do sistema de edição/subscrição, detalhando-se o seu funcionamento e concretização. Antes de mais é importante salientar que esta camada depende das funcionalidades que as camadas previamente apresentadas lhe conferem. A camada de primitiva de comunicação DSR permite efectuar encaminhamento *multi-hop* numa rede MANet, enquanto que a de rede sobreposta permite abstrair o nível de E/S da topologia real da rede.

O funcionamento do protocolo de edição/subscrição é inspirado no trabalho de [11], tomando partido das suas características semi-probabilísticas para garantir um correcto funcionamento no ambiente de operação proposto. No entanto, dado que a solução proposta neste trabalho utiliza uma primitiva de comunicação ponto-a-ponto, vários dos pressupostos utilizados no trabalho de [11] terão de ser repensados para este novo cenário.

No sistema DSR-WALK, a publicação de eventos é feita por meio de mensagens de percurso aleatório em oposição ao uso de mensagens de *gossip*, visto que a natureza da comunicação é claramente distinta da apresentada em [11]. A subscrição de eventos utiliza um mecanismo de inundação limitada por um horizonte de subscrição (caso se usem vistas simétricas limitadas), ou percursos aleatórios (no caso de vistas assimétricas não limitadas). Quando uma publicação chega a um nó com informação de subscrição, uma mensagem especial (ROUTED PACKET) é gerada e enviada para o subscritor. Ao percorrer a rede, uma mensagem de percurso aleatório escolhe preferencialmente nós pelos quais ainda não tenha passado. De modo a permitir o funcionamento deste mecanismo uma lista dos nós já atravessados é mantida no cabeçalho da mensagem de percurso aleatório.

A Figura 3.3 apresenta uma simples ilustração do funcionamento do protocolo de edição/subscrição. Um nó editor envia, neste caso, dois percursos aleatórios para publicar um evento no sistema. Nós cinzentos são subscritores interessados nos eventos do nó editor, e propagam informação de subscrição com um horizonte de apenas um *hop*. É possível observar na figura que quando um percurso aleatório atravessa um nó com informação de encaminhamento, uma mensagem é gerada com destino ao nó subscritor, mas o percurso aleatório continua a ser propagado enquanto o seu *TTL* for válido.

Embora este protocolo seja probabilístico por natureza, é possível prever a probabilidade de entrega de um pacote assumindo um cenário simplificado. Existem essencialmente dois factores que influenciam a propagação de um percurso aleatório : 1) falhas de comunicação; e 2) grau dos nós membros da rede sobreposta.

- **Falhas de comunicação** - A ocorrência de falhas de comunicação limita a

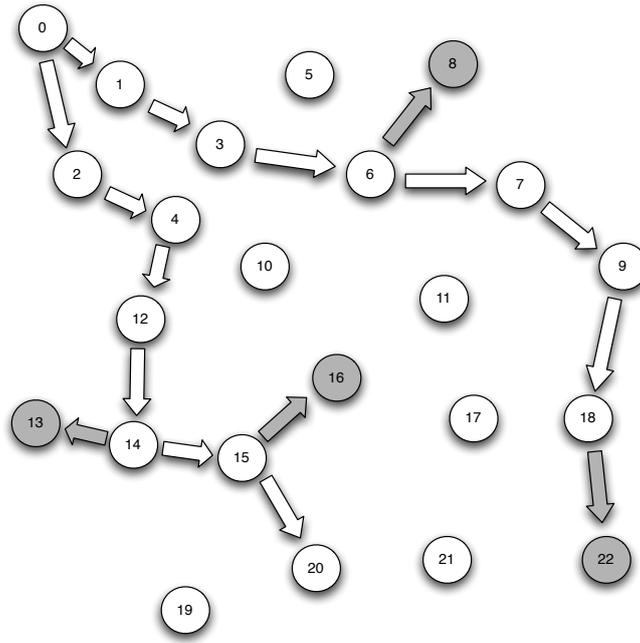


Figura 3.3: Randow-walks

propagação da informação de subscrição na rede, e também a cobertura das mensagens de percurso aleatório. Como tal, este factor arbitrário torna não trivial calcular a probabilidade de entrega para uma dada mensagem.

- **Grau dos nós** - O grau interno de um nó determina a probabilidade deste nó receber mensagens de percurso aleatório provenientes de outros elementos da rede. Quanto maior for o seu grau interno, mais elementos da rede o conhecem, logo maior será a probabilidade deste receber mensagens. Como tal, uma distribuição não uniforme do grau interno dos nós influencia a probabilidade de entrega para uma dada mensagem.

De modo a simplificar este raciocínio, considera-se para este cálculo um cenário onde não ocorrem falhas de comunicação, e onde todos os elementos da rede sobreposta se conhecem (grau interno e externo uniforme). De modo a calcular a probabilidade de entrega para um determinado tópico, temos apenas de saber o número de nós na rede sobreposta, o número de nós com informação de subscrição na rede, e o número de saltos que uma publicação efectua.

$$\frac{SubInfo}{nNodes - 1} * [1 - \frac{SubInfo}{nNodes - 1}]^{nHops-1} + \sum_{k=2}^{nHops} \frac{SubInfo}{nNodes - 2} * [1 - \frac{SubInfo}{nNodes - 2}]^{nHops-k} \quad (3.5)$$

A probabilidade de uma mensagem ser entregue correctamente pode ser expressa da seguinte forma: em cada salto da mensagem de publicação, a probabilidade de o próximo salto ser um subscritor ou um nó com informação de subscrição é igual ao número de nós com informação de subscrição (*SubInfo*), a dividir pela totalidade de nós disponíveis ( $nNodes-2$  : o nó actual e o salto anterior não são contemplados na escolha de um próximo salto). A probabilidade para um número arbitrário de saltos pode ser dada por uma distribuição geométrica <sup>2</sup> como a apresentada na equação 3.5. Desta maneira é possível prever para um cenário sem falhas simplificado, o impacto que os diferentes parâmetros vão ter na entrega de mensagens. Embora este cálculo seja demasiado simples para ser aplicado num cenário prático, permite perceber em que medida a alteração de diferentes parâmetros influenciam o nível de entrega de uma publicação.

### Subscrição de eventos

A estratégia desenvolvida permite subscrição de eventos por conteúdo (semelhante à solução apresentada em [11]), no entanto para simplificar os exemplos apresentados, as subscrições são feitas de acordo com uma lista de tópicos numéricos predeterminada.

Caso se utilize uma rede sobreposta com vistas simétricas limitadas, quando um nó deseja subscrever um tipo de evento, este propaga um pedido de subscrição para esse tópico através de um mecanismo de inundação limitada. Caso se use uma rede sobreposta com vistas assimétricas não limitadas, o processo de subscrição não pode ser feito por inundação, dado que um nó pode conhecer potencialmente todos os elementos da rede sobreposta. Deste modo, subscritores com vistas assimétricas não limitadas enviam  $n$  *random-walks* para elementos da sua vista activa, com um *TTL* predeterminado. Tal como no sistema semi-probabilístico já apresentado, utiliza-se um parâmetro de horizonte de subscrição, que determina por quantos *hops* a informação de subscrição se deve propagar. Tendo em conta que numa rede sobreposta simétrica esta informação é enviada para todos os elementos da vista activa, este parâmetro está intrinsecamente ligado com o tamanho máximo da vista, e com o número de elementos do sistema (i.e. : numa rede com 100 nós e uma vista activa máxima de 6 elementos, um horizonte de subscrição de 3 poderia potencialmente alcançar todos os elementos do sistema). Um exemplo de subscrição num sistema com vista activa de 3 elementos e horizonte de subscrição de 1 pode ser observado na Figura 3.4, e um exemplo de funcionamento com uma rede sobreposta com vistas assimétricas é apresentada na Figura 3.5.

<sup>2</sup>a distribuição de probabilidade do número  $X$  de tentativas de Bernoulli necessárias para alcançar um sucesso, suportadas pelo conjunto  $\{ 1, 2, 3, \dots \}$

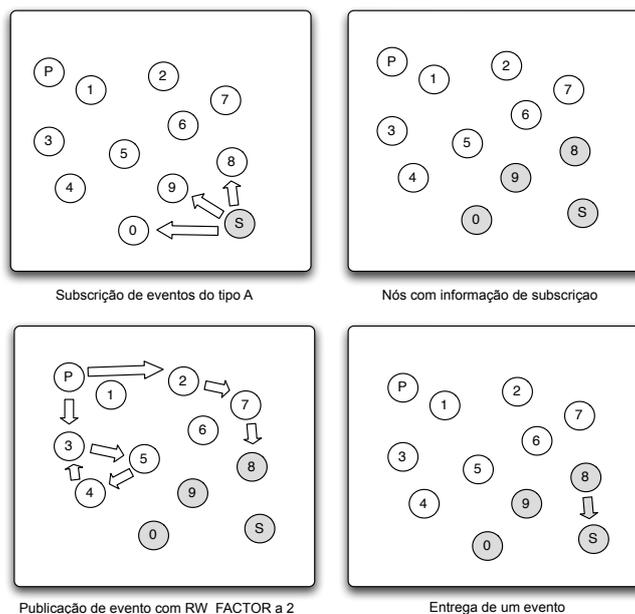


Figura 3.4: Funcionamento do protocolo de edição subscrição

### Publicação de eventos

De modo a publicar eventos para o sistema, um nó editor tem de enviar  $n$  *random-walks* para elementos da sua vista activa, sendo que este número depende de um parâmetro de sistema (RW FACTOR). Embora o processo de escolha de elementos seja aleatório, é dada preferência a elementos que se encontrem a uma distância menor em saltos do nó emissor, e caso não existam elementos suficientes na vista activa, membros da vista passiva são utilizados. Embora esta solução minimize o *overhead* no sistema, introduz também um *bias* na escolha de elementos para os quais se enviam eventos. De modo a minimizar o impacto desta optimização, cada mensagem de percurso aleatório mantém no seu cabeçalho o caminho que já percorreu. Sempre que a mensagem é reencaminhada o caminho percorrido pela mensagem é consultado, e a não ser que não existam alternativas, a mensagem não é enviada para nós pelos quais já tenha passado. Após o envio das mensagens iniciais, os percursos aleatórios vão atravessar a rede até o *TTL* se esgotar, tendo em consideração que em cada salto a mensagem não pode ser reenviada para o salto anterior (excepto quando não exista mais nenhum nó para o qual enviar a mensagem). Quando um percurso aleatório atravessa um nó que possui informação de subscrição para o tipo de evento que o percurso aleatório transporta, uma *ROUTED MESSAGE* é gerada para os nós da tabela de encaminhamento pertinentes. No entanto a *ROUTED MESSAGE* pode ser enviada directamente para o subscritor caso este se encontre na vista activa do nó contactado. Este processo pode ser observado na figura 3.5, onde o nó 5 envia

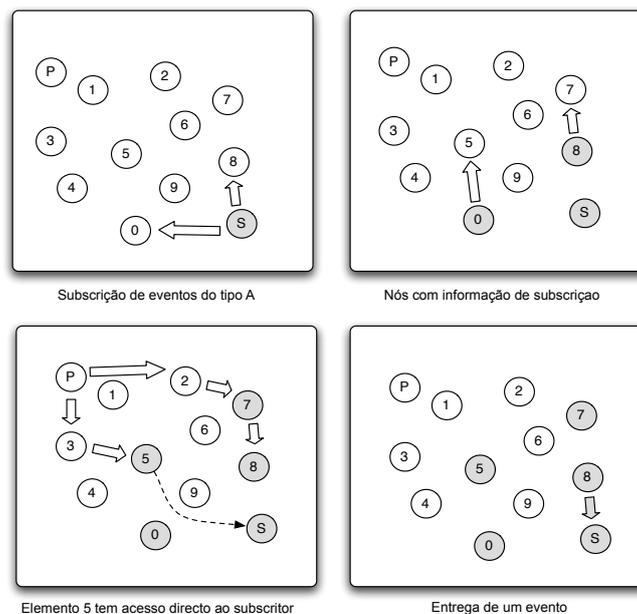


Figura 3.5: Funcionamento do protocolo de edição subscrição

uma publicação directamente para o subscritor. Se uma ROUTED MESSAGE para um evento 'A' passar por um nó que tenha informação de encaminhamento para 'A', então esse nó gera  $n$  cópias da ROUTED MESSAGE com o  $TTL$  actual da mensagem. O número de ROUTED MESSAGES geradas é igual ao número de endereços na tabela de encaminhamento para o evento 'A'.

### Principais parâmetros

O funcionamento deste protocolo é afectado por diversos parâmetros, nomeadamente: o número de saltos que um percurso aleatório de publicação percorre, o factor de *fanout* das mensagens de subscrição; e o número de saltos que as mensagens de subscrição percorrem. A configuração destes parâmetros afecta decisivamente o funcionamento do protocolo, e embora no âmbito deste trabalho os valores utilizados sejam escolhidos previamente, este protocolo tem o potencial para se adaptar a diferentes cenários de utilização.

- **$TTL$  de mensagens de publicação** - A configuração do número de saltos que um percurso aleatório de publicação executa afecta o funcionamento do protocolo por dois motivos principais. Por um lado, um valor elevado de  $TTL$  para mensagens de publicação permite cobrir um maior número de nós na rede, que gera maiores níveis de entrega. Em contrapartida, quanto maior for este valor maior será o *overhead* na rede introduzido por publicações. Como tal,

em cenários de maior mobilidade (onde uma maior carga de mensagens na rede deteriora o funcionamento da primitiva de comunicação) e em cenários com vários editores e/ou taxas de publicação elevadas, um valor de *TTL* menor é preferencial. Em cenários com pouca mobilidade e menos publicações, um *TTL* maior oferece melhores resultados.

- ***fanout* e *TTL* de mensagens de subscrição** - Caso se utilize uma rede sobreposta com vistas simétricas limitadas, o *fanout* de subscrição é sempre 100%, logo, o tamanho máximo da vista activa tem o mesmo efeito que este parâmetro. O *TTL* das mensagens de subscrição em conjunção com o *fanout* de subscrição influencia a percentagem de nós na rede que possuem informação para efectuar encaminhamento determinístico. A escolha destes parâmetros influencia o funcionamento do protocolo de uma maneira semelhante à já apresentada. Valores altos resultam numa maior cobertura dos nós da rede, mas ao mesmo tempo geram mais *overhead* de comunicação. É também importante notar que valores altos de *TTL* podem gerar ciclos nos caminhos formados entre um nó e o subscritor.

Embora se possa escolher valores padrão para estes parâmetros, o protocolo apresenta melhores resultados quando estes são configurados explicitamente para cada tipo de cenário. Na secção de avaliação será quantificado o impacto que estes parâmetros têm na execução do protocolo.

### Vantagens/Desvantagens

Uma das principais características do protocolo apresentado é a facilidade de configuração do seu comportamento através da escolha dos parâmetros previamente apresentados. Embora esta flexibilidade de configuração permita adaptar facilmente o protocolo a cenários distintos, a necessidade de se configurar explicitamente estes parâmetros para obter resultados óptimos torna o protocolo dependente de intervenção humana. Idealmente, os parâmetros de configuração seriam escolhidos dinamicamente pelo protocolo conforme as condições de funcionamento da rede MANet. Na secção de trabalho futuro explora-se esta possibilidade.

# Capítulo 4

## Avaliação

De modo a avaliar o desempenho do sistema desenvolvido foi feita uma concretização do mesmo no simulador NS-2. A apresentação dos resultados encontra-se organizada em dois blocos principais: primeiro apresentam-se resultados numa rede estática com baixa carga para determinar a qualidade do sistema em condições óptimas; de seguida, apresentam-se resultados para uma rede com mobilidade e com níveis de carga superiores.

### 4.1 *NS-2*

O simulador utilizado para efectuar os testes presentes nesta secção foi o *Network Simulator 2* (NS-2). A versão utilizada para realizar as simulações foi a 2.31, e todos os testes foram efectuados em ambiente unix no sistema operativo ubuntu 7.04.

A concretização desenvolvida consiste código C++ (camada de edição subscrição, camada de rede sobreposta simétrica e camada de rede sobreposta assimétrica) e código TCL. O código TCL consiste de um *script* que permite configurar o ambiente de simulação, e iniciar as instâncias dos agentes de rede sobreposta e de edição/subscrição. Adicionalmente, os parâmetros de tráfego e de movimento são carregados também via TCL.

Dado que o sistema desenvolvido necessita de aceder à *cache* de encaminhamento do protocolo DSR, a concretização deste protocolo teve de ser alterada no NS-2. Como tal, as estruturas relativas à *cache* de encaminhamento foram tornadas *public*, e foi introduzido um comando de interface com o *script* TCL para facilitar a localização da instância correcta do agente DSR para cada nó.

### 4.2 Parâmetros de Configuração

Os principais parâmetros de configuração necessários para efectuar a simulação de uma rede MANet no simulador NS-2 são: o número de nós na rede, a dimensão

da área geográfica, o modelo de movimento, o modelo de propagação, o raio de transmissão e o tempo de simulação.

Em todas as simulações apresentadas os seguintes parâmetros mantêm-se inalterados:

- Modelo de propagação: *TwoRayGround*<sup>1</sup>
- Raio de transmissão: 250 metros
- Tempo de simulação: 800 segundos
- Modelo de movimento: *Random Waypoint*

O número de nós, a dimensão geográfica e a velocidade a que os nós se deslocam é apresentada sempre que se introduz uma simulação. O padrão de tráfego utilizado varia conforme o teste efectuado, sendo sempre identificado o número de editores no sistema, o número de subscritores, e a taxa a que eventos de publicação são produzidos.

Finalmente, de modo a minimizar a quantidade de simulações a efectuar, estudou-se apenas o efeito da utilização de *TTL's* de subscrição (SUB TTL) de 2 saltos, mas com *fanout* de subscrição variável. A legenda *sub info* apresentada nos gráficos desta secção refere-se à quantidade máxima de informação de subscrição na rede, calculada por *fanout*\*SUB TTL.

### 4.3 Métricas

Para a avaliação das simulações efectuadas utilizam-se 3 métricas principais:

- **Taxa de entrega (%):** A taxa de entrega de uma simulação é representada pela seguinte formula:

$$\frac{rec}{\sum_{k=1}^{nTopics} P_k \times S_k} \quad (4.1)$$

Onde *rec* representa o número total de eventos recebidos numa simulação, *nTopics* o número de tópicos distintos, *P<sub>k</sub>* o número de publicações de um evento do tipo 'k', e finalmente *S<sub>k</sub>* o número de subscritores de um evento 'k'.

---

<sup>1</sup>Modelo de propagação que tem em conta não apenas a linha directa de comunicação entre dois nós, mas também a reflexão do sinal no solo. Modelo tipicamente usado em simulações de redes MANet na literatura.

- **Número de mensagens enviadas:** Esta métrica contempla todas as mensagens geradas no decorrer da simulação, independente do seu tipo (mensagens de controlo DSR, mensagens do nosso protocolo, etc..).
- **Proporção de perdas:** Calculada pelo rácio entre a quantidade de mensagens perdidas numa simulação, e a quantidade de mensagens enviadas.

Informação referente a legendas de gráficos:

- **TTL de publicação:** Representa o número de saltos que uma mensagem de publicação percorre. Tipicamente, em cada cenário efectuem-se simulações fazendo variar este parâmetro entre 1 e 18 saltos.
- **Pub fanout:** Representa o número de mensagens de percurso aleatório geradas para cada publicação. Por defeito utiliza-se apenas uma mensagem de percurso aleatório para os testes efectuados, com a excepção do gráfico na Figura 4.7 que pretende avaliar o impacto da utilização de *fanout* maior que 1 para publicação.
- **Sub info:** Representa o número máximo de nós na rede que podem possuir informação de subscrição de um dado subscritor. Este valor é calculado multiplicando o *fanout* de subscrição (que fazemos variar entre 5 e 10 elementos, com a excepção de cenários que utilizem a rede sobreposta simétrica, nesse caso o *fanout* é igual ao tamanho da vista activa) pelo número de saltos que a subscrição percorre. Por defeito utilizou-se sempre um *TTL* de subscrição de 2 saltos, pois permite evitar a criação de ciclos de encaminhamento.

De seguida apresenta-se um conjunto de simulações destinadas a avaliar o protocolo desenvolvido.

## 4.4 Cenário 1: Rede Estática

Nesta secção apresentam-se e comparam-se valores obtidos através de modelação, com os resultados obtidos por simulação do sistema numa rede estática com pouca carga. Esta comparação pretende avaliar a capacidade do protocolo operar num cenário onde a perda de pacotes é baixa (uma vez que existem poucas colisões).

Todas as simulações apresentadas nesta secção correspondem a testes efectuados com a rede sobreposta assimétrica, pois o desempenho obtido com a rede sobreposta simétrica foi pouco satisfatória. A justificação para o baixo desempenho da rede sobreposta simétrica é apresentada e discutida posteriormente na Secção 4.4.6.

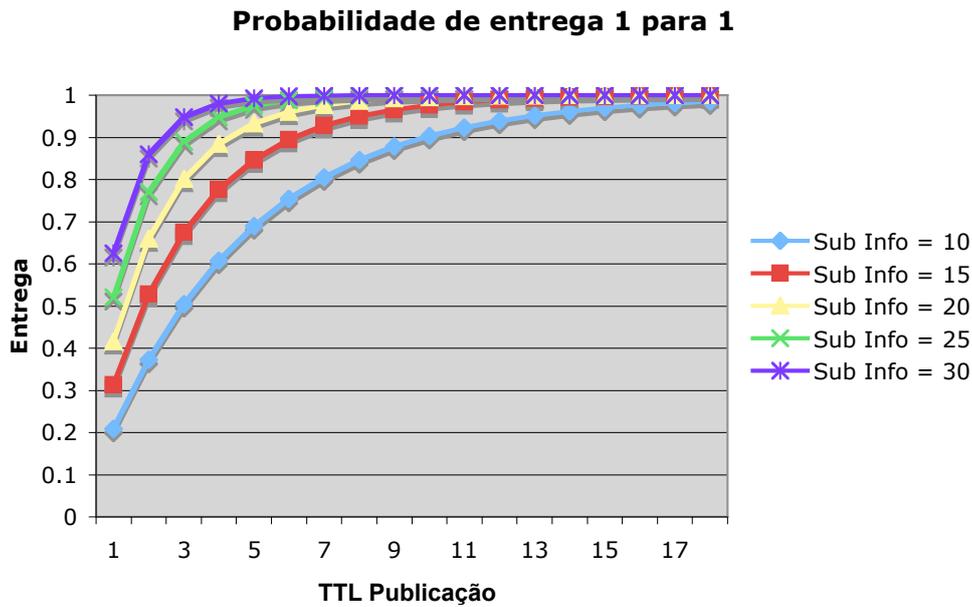


Figura 4.1: Distribuição geométrica

#### 4.4.1 Desempenho Previsto por Modelação

Como apresentado no capítulo anterior, é possível modelar a entrega expectável de pacotes para um cenário sem falhas, utilizando uma distribuição geométrica cumulativa (Equaçã 3.5). É portanto interessante comparar os resultados previstos por este modelo com os resultados obtidos por simulação.

A Figura 4.1 apresenta os resultados obtidos a partir da Equação 3.5 fazendo variar os parâmetros de *TTL* das mensagens de edição e também a quantidade de informação de subscrição na rede. A Figura 4.1 apresenta resultados do modelo para um cenário com apenas um editor e um subscritor, de modo a simplificar a formalização do problema em termos matemáticos. Este modelo assume que não ocorrem falhas de comunicação e que cada salto do percurso aleatório tem igual probabilidade de alcançar qualquer um dos nós da rede.

#### 4.4.2 Resultados da Simulação

Para este cenário simplificado utilizam-se os seguintes parâmetros:

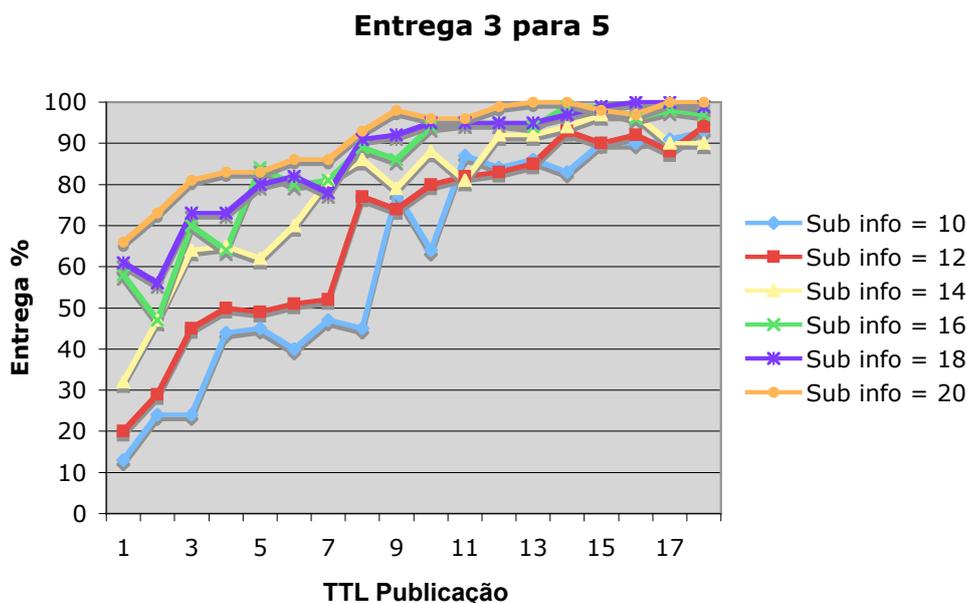


Figura 4.2: Taxas de entrega com 3 editores 5 subscritores

Número de nós	50
Número de editores	3
Taxa de publicação	25 eventos publicados
TTL de publicação	entre 1 e 18 saltos
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	entre 5 e 10
Velocidade dos nós	0 m/s
Duração da simulação	700s
Área	1000m x 1000m

Na Figura 4.2 apresentam-se resultados de simulação, obtidos em condições operacionais que aproximam as hipóteses feitas no modelo matemático. Para tal, foram efectuadas simulações em que se fez variar o *TTL* de publicação entre 1 e 18 saltos, e em que se fez variar o *fanout* de subscrição entre 5 e 10 elementos. O *TTL* utilizado para mensagens de subscrição foi de 2 saltos, de modo a eliminar a possibilidade de se criarem ciclos de encaminhamento [11]. Apenas uma mensagem de percurso aleatório é gerada por publicação.

#### 4.4.3 Diferenças entre o Modelo e a Simulação

Infelizmente, por limitações que enumeramos de seguida, o sistema modelado e o sistema simulado não são o mesmo. Por um lado, para simplificar o modelo ma-

temático, assumimos apenas um editor e um subscritor. No entanto, uma simulação com apenas um editor e um subscritor produz resultados pouco fiáveis pelas seguintes razões, nomeadamente: 1) tipicamente, um sistema editor/subscritor tem como finalidade a disseminação de informação por múltiplos subscritores; e 2) uma simulação com apenas um editor e um subscritor seria pouco reveladora da capacidade de funcionamento real do sistema. Num cenário com pouca ou nenhuma mobilidade, a distribuição geográfica dos nós é determinante para o sucesso da entrega de mensagens. Por exemplo, num cenário em que o editor se encontre próximo do subscritor, a taxa de entrega de eventos tem maior probabilidade de ser elevada, dado que o funcionamento do protocolo beneficia nós geograficamente próximos. Da mesma maneira, se o editor e o subscritor se encontrarem muito distantes, a probabilidade dos eventos de publicação alcançarem o seu destino é menor.

Desta forma, enquanto o sistema modelado possui um editor e um subscritor o sistema modelado possui 3 editores e 5 subscritores. Apesar desta discrepância, julgamos que é possível extrair algumas conclusões da comparação. De modo a facilitar esta tarefa, fez-se variar de modo semelhante os mesmo parâmetros em ambos os casos.

Como se pode observar, as curvas resultantes da simulação (Figura 4.2) têm um comportamento semelhante ao das curvas obtidas por modelação (Figura 4.1). Observa-se que de modo a alcançar a entrega máxima no cenário simulado, necessitam-se de mais saltos nas mensagens de publicação (RW TTL) do que no modelo matemático. Existem essencialmente duas razões que explicam esta discrepância: 1) a taxa de perdas do protocolo DSR; e 2) a irregularidade da rede sobreposta.

**Taxa de perdas do protocolo DSR** - Mesmo operando numa rede estática e com pouca carga, o protocolo de comunicação DSR pode perder mensagens, normalmente devido a colisões. Nas simulações efectuadas foi contabilizada uma taxa de pelo menos 3% de perda de mensagens. Como o protocolo apresentado não possui mecanismos para mascarar as omissões da rede, esta taxa de erro afecta o funcionamento do protocolo.

**Irregularidade da rede sobreposta** - O modelo matemático apresentado parte do princípio que a probabilidade de se escolher um nó aleatoriamente é igual para todos os elementos da rede. Como tal, probabilisticamente, todos os nós são equidistantes na rede sobreposta. A título de exemplo, numa rede com esta propriedade, um percurso aleatório originado num nó 'S' tem igual probabilidade de escolher qualquer nó da rede como próximo salto para esta mensagem. Como tal, quando 'S' envia uma mensagem de percurso aleatório, a probabilidade desta ser entregue a um destino pode ser representado por uma distribuição geométrica cumulativa:

- Quando 'S' envia a mensagem para um dos seus vizinhos, por exemplo um nó 'C', a probabilidade da mensagem ser entregue é igual à probabilidade do nó

'C' ter informação de subscrição para este evento. Ou seja,  $P_1 = NI/(N - 1)$ , em que  $NI$  é o número de nós com informação de subscrição e  $N$  o número total de nós na rede.

- Se 'C' não tiver informação de subscrição (o que corresponde a uma probabilidade de  $1 - P$ ), a mensagem de percurso aleatório continua. O nó 'C' envia a mensagem para um dos seus vizinhos que não o nó 'S', dado que, como já explicado, um percurso aleatório não é enviado de volta para o nó que o originou no salto anterior (neste caso,  $P_2 = NI/(N - 2)$ ).
- Este processo continua até a mensagem ser entregue a um nó com informação de subscrição, ou até esgotar o seu *TTL*. A probabilidade de uma mensagem ser entregue depende portanto apenas da quantidade de informação de subscrição na rede sobreposta, e do número de saltos que a mensagem de percurso aleatório efectua. (Esta probabilidade é representada pela probabilidade de entregar a mensagem em 1, 2, ... ou *TTL* saltos).

Como se pode verificar, um percurso aleatório que percorra uma rede sobreposta com a propriedade apresentada (equidistância dos nós na rede sobreposta) contacta um conjunto uniforme de nós aleatórios da rede. No entanto, as redes sobrepostas geradas pelo nosso algoritmo não apresentam esta propriedade devido essencialmente a dois factores seguintes:

- **Funcionamento do protocolo DSR** - O protocolo DSR gera um enviesamento na probabilidade de um nó contactar outro nó arbitrário da rede. Observa-se que, um nó geograficamente próximo do nosso nó emissor 'S', pode ter uma maior probabilidade de ser contactado com sucesso que outro nó geograficamente distante de 'S'. Esta característica pode ser justificada pela crescente probabilidade de falha de uma mensagem DSR à medida que a distância que a mensagem tem de percorrer aumenta.
- **Optimizações na rede sobreposta** - As optimizações descritas no Capítulo 3 influenciam a probabilidade de um nó contactar com outro nó arbitrário na rede. Estas optimizações são: 1) a utilização preferencial de rotas com informação de encaminhamento; e 2) a utilização preferencial de rotas mais curtas. Este tipo de política de funcionamento pode gerar situações onde apenas alguns nós são muito contactados, enquanto que os restantes têm menor probabilidade de receber uma mensagem. Por exemplo, num cenário onde existe apenas um editor de informação, a probabilidade de um nó receber uma mensagem de edição aumenta com a sua proximidade relativamente ao nó editor.

Estas observações sugerem que apesar da não utilização destas optimizações pudesse vir a melhorar a distribuição de carga na rede e a previsibilidade das taxas de entrega, iria também diminuir drasticamente a carga máxima de mensagens que o sistema suportaria. As optimizações utilizadas visam minimizar a sobrecarga introduzida pelo protocolo DSR, permitindo assim a melhor utilização da rede.

De modo a melhor ilustrar a necessidade destas optimizações, a Figura 4.3 apresenta o impacto da não utilização das mesmas no desempenho do sistema. A simulação que produziu os resultados apresentados na Figura 4.3 utilizou os seguintes parâmetros:

Número de nós	50
Número de editores	5
Taxa de publicação	1 mensagem por segundo
TTL de publicação	18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	10
Velocidade dos nós	10 m/s
Duração da simulação	700s
Área	1000m x 1000m

Concluindo, as optimizações utilizadas na concretização das redes sobrepostas apresentadas não permitem assegurar uma rede com nós probabilisticamente equidistantes, e como tal não nos é possível comparar directamente o comportamento do sistema modelado pela Equação 3.5. Embora o enviesamento da comunicação na rede sobreposta possa ter aspectos prejudiciais para o funcionamento do sistema, este também acarreta alguns efeitos colaterais positivo. Com vista a aprofundar este tema, a próxima subsecção expõe uma relação entre a distribuição de graus de entrada nos elementos da rede, e discute como essa distribuição afecta o funcionamento do sistema. Adicionalmente, apresentam-se dados relativos à distribuição de carga pelos elementos da rede.

#### 4.4.4 Rede Sobreposta Não Regular: Comportamento Emergente

A utilização de optimizações na escolha de rotas nas redes sobrepostas apresentadas revelou um fenómeno interessante relativamente à distribuição de carga na rede. Como foi já referido, um nó irá escolher preferencialmente como próximo salto um elemento que se encontro perto dele, e como tal, num cenário em que a distribuição geográfica dos nós não seja uniforme, a distribuição de carga e o grau de entrada dos nós irá variar. Considere-se, a título de exemplo, a Figura 4.4 onde os nós a preto representam elementos do sistema de edição/subscrição, e os nós a cinzento

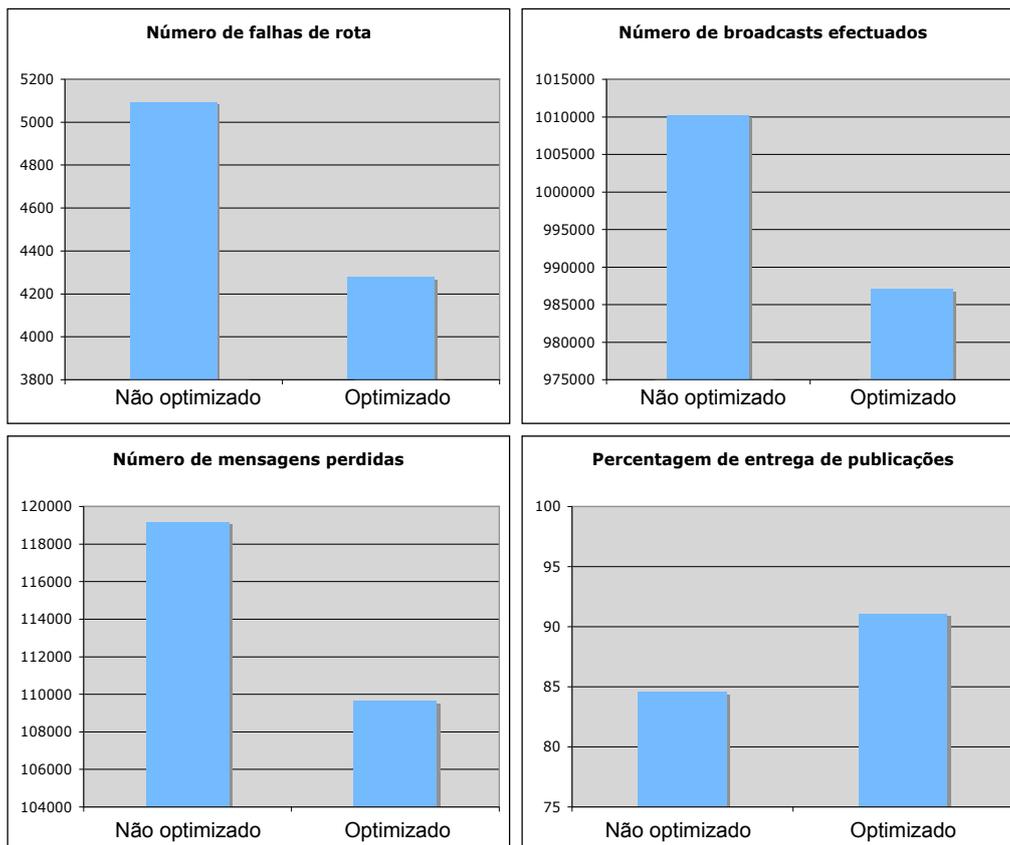


Figura 4.3: Efeitos das optimizações

são nós independentes do sistema mas com capacidade de comunicação DSR. Num cenário como o apresentado, nós que se encontrem longe do aglomerado principal de elementos, irão ser escolhidos com menor probabilidade para receberem mensagens de percurso aleatório. De igual modo, ao escolher elementos para a sua vista activa, um nó irá escolher preferencialmente nós próximos.

Em contrapartida, os elementos no centro do aglomerado poderão ter maior grau de entrada, o que irá resultar em nós com popularidade desigual. No entanto, apesar de um nó na periferia ter menor probabilidade de receber mensagens, a natureza do protocolo apresentado permite minimizar o efeito prejudicial causado pela desigual distribuição de popularidades. Por exemplo, se um dos elementos isolados na rede for um subscritor de um determinado evento, este irá propagar para  $f$  vizinhos seu desejo em receber eventos de um dado tipo. Ao propagar essa informação, as mensagens de pedido de subscrição irão contactar com maior probabilidade elementos da rede com um grau de entrada alto. Da mesma maneira, quando um nó editor publica um evento na rede, esse evento irá com maior probabilidade chegar a elementos com grau de entrada alto do que aos restantes nós. Desta forma, os elementos

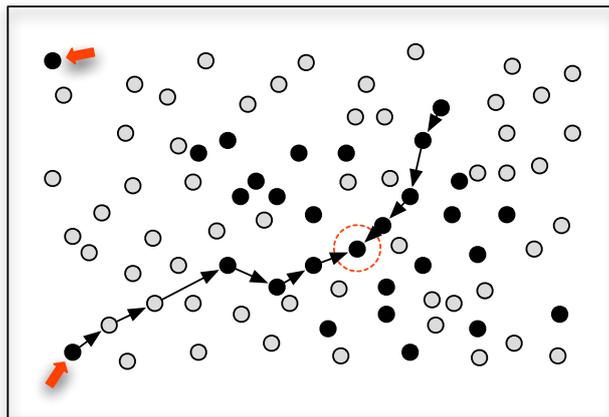


Figura 4.4: Rede com elementos na "periferia", e com um elemento popular

que apresentam um grau de popularidade maior servem de ponto-de-encontro para as mensagens de subscrição e de edição, e a rede sobreposta resultante embora não garanta uma distribuição uniforme da carga na rede, permite ainda assim a correcta operação do sistema.

De modo a quantificar este fenómeno apresentam-se dados relativos à quantidade de mensagens recebidas por cada nó. Estes dados foram obtidos de simulações feitas com dois tipos de rede, uma rede designada por rede de *filiação parcial*, com 250 nós onde apenas 50 fazem parte do sistema de edição subscrição, e uma rede designada por rede de *filiação total* com 50 nós onde todos fazem parte do sistema. Para estas simulações utilizam-se 3 editores e 5 subscritores, e os nós movem-se a 3m/s. Os nós subscritores são os nós com os identificadores 0, 1, 2, 3 e 4, e como tal é expectável que estes recebam mais mensagens que os restantes.

Como se pode observar na Figura 4.5, a distribuição de mensagens recebidas numa rede em que todos os nós fazem parte do sistema é mais uniforme que numa rede de filiação parcial. Tendo em conta que o modelo de movimento utilizado (*RandomWaypoint*) distribui os vectores de movimentos dos nós de uma maneira uniforme na rede, os efeitos de uma distribuição geográfica desigual são minimizados. No entanto, este efeito é mais notório numa rede de filiação parcial, onde embora a distribuição dos nós seja uniforme, a distribuição dos elementos do sistema possa não o ser. Neste cenário é possível verificar que a distribuição de mensagens recebidas é muito diferente, existindo um número reduzido de nós cuja popularidade é consideravelmente maior que a dos restantes elementos da rede. Este tipo de nós de elevada popularidade, embora sirvam de ponto-de-encontro entre edições e subscrições, e facilitem assim o encaminhamento de eventos, são também um factor limitante da performance da rede. Dado o número elevado de mensagens que estes nós recebem

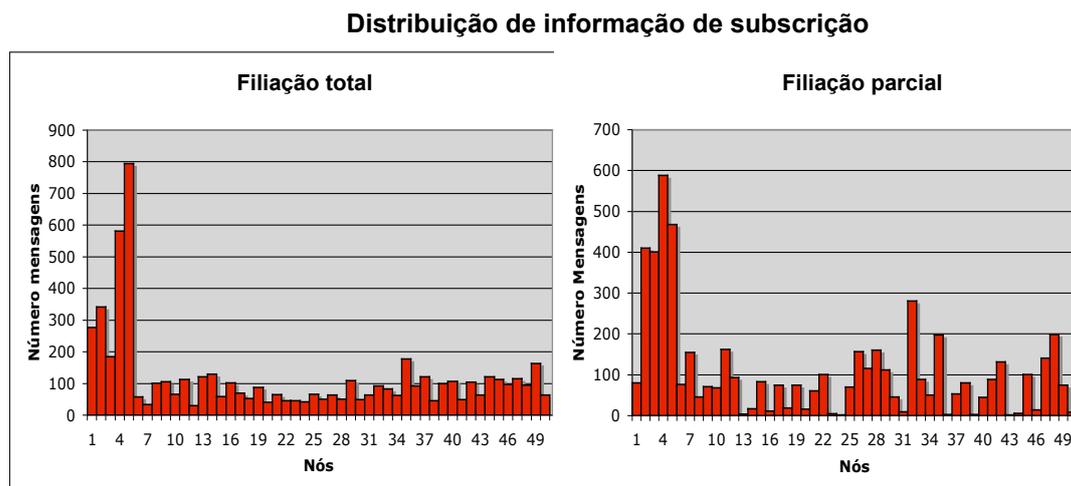


Figura 4.5: Distribuição de mensagens numa rede de filiação total e de filiação parcial

comparativamente com o resto dos elementos da rede, o esforço de comunicação por parte destes nós é também consideravelmente maior. Esta distribuição desigual da carga na rede diminui a carga máxima que esta poderia teoricamente suportar. De igual modo, a Figura 4.6 apresenta a distribuição de informação de subscrição pelos nós da rede, onde se pode verificar uma distribuição semelhante à apresentada na Figura 4.5. Esta distribuição desigual de informação de subscrição é coerente com a desigualdade de popularidade dos nós referida.

#### 4.4.5 Escolha de Número de Percursos Aleatórios

Esta simulação pretende avaliar o impacto da utilização de um *fanout* de 2 para efectuar publicações. Para esta simulação utilizaram-se os seguintes parâmetros:

Número de nós	50
Número de editores	5
Taxa de publicação	1 ou 10 mensagens por segundo
TTL de publicação	entre 1 e 18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	10
Velocidade dos nós	3 ou 20 m/s
Duração da simulação	700s
Área	1000m x 1000m

De seguida apresentam-se dados que pretendem determinar o efeito da utilização de vários percursos aleatórios para cada publicação. Como se pode verificar na Fi-

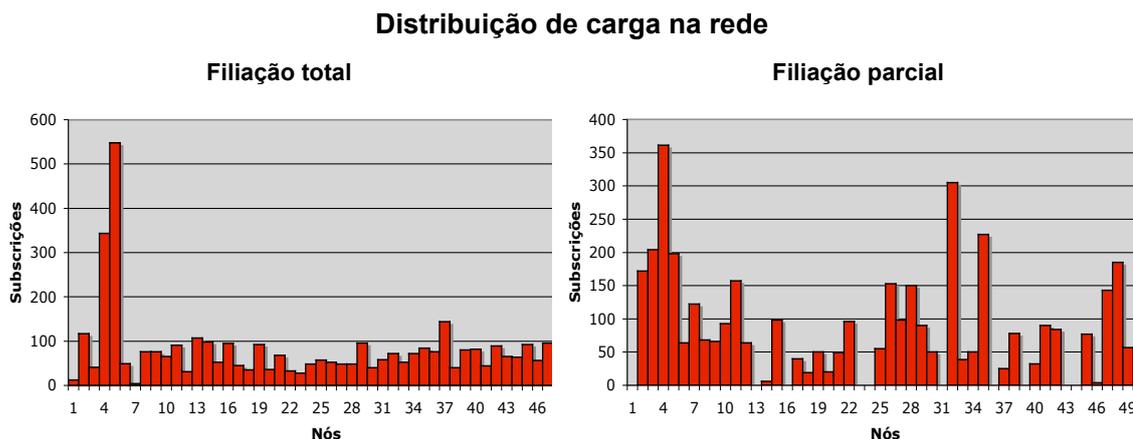


Figura 4.6: Distribuição de subscrições numa rede de filiação total e de filiação parcial

gura 4.7, a utilização de vários percursos aleatórios apresenta resultados menos satisfatórios que a utilização de apenas um. Este facto pode ser explicado devido à maior probabilidade de colisão quando várias mensagens percorrem a rede simultaneamente, e agrava-se quando existe uma maior carga na rede. Adicionalmente, quando várias mensagens de percurso aleatório são utilizadas para uma publicação, estas podem percorrer desnecessariamente nós que já foram contactados por esta mesma publicação. O resto deste capítulo assume para todas as simulações a utilização de apenas uma mensagem de percurso aleatório para cada publicação.

#### 4.4.6 Rede Sobreposta Simétrica

Durante o desenvolvimento do sistema foram contempladas duas aproximações diferentes para a camada de rede sobreposta, uma com vistas activas simétricas, e outra com assimétricas. No entanto, a formulação da camada de rede sobreposta simétrica mostrou-se pouco eficaz devido a uma quantidade excessiva de mensagens de controlo inseridas na rede pelo sistema, e também devido à rigidez das garantias que esta camada pretende assegurar. Devido a constrangimentos de tempo relativos ao desenvolvimento do projecto não foram estudadas alternativas que permitissem melhorar o desempenho desta aproximação, tentando manter o máximo das propriedades benéficas associadas com a utilização de vistas simétricas.

Para as simulações com a rede sobreposta simétrica utilizaram-se os seguintes parâmetros:

Número de nós	50
Número de editores	5
Taxa de publicação	1 mensagem por segundo
TTL de publicação	entre 1 e 18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	1 e 2 saltos
Número de mensagens de subscrição	10
Velocidade dos nós	0 m/s
Duração da simulação	700s
Área	1000m x 1000m

A Figura 4.8 apresenta dados relativos à execução do protocolo com a camada de rede sobreposta simétrica. A proporção de mensagens perdidas com esta solução é muito maior do que a observada nas simulações com a rede sobreposta assimétrica. Em primeiro lugar é possível constatar que a proporção de mensagens perdidas decresce à medida que se aumenta o *TTL* de publicação. Este fenómeno deve-se ao facto de uma grande parte das colisões na simulação ocorrerem aquando a construção da rede sobreposta (antes de se começarem a enviar mensagens de publicação). A grande proporção de colisões deve-se essencialmente aos mecanismos desta rede que pretendem assegurar a simetria entre as vistas. Quando um pedido de junção percorre os elementos da rede, cada elemento contactado envia uma mensagem de PING para o novo nó, quer tenha informação de subscrição para esse nó ou não. Caso o nó contactado não possua informação de encaminhamento para o novo nó, o protocolo DSR inicial o processo de *Route Discovery*. Tendo em conta que este mecanismo utiliza inundação, um número considerável de colisões pode ser gerada aquando a criação da rede sobreposta.

Embora

Esta proporção de perda de mensagens aliada à complexidade do protocolo resultam num sistema pouco flexível e incapaz de operar num ambiente dinâmico típico de uma rede MANet.

## 4.5 Cenário 2: Efeitos da Mobilidade e Carga

De seguida apresenta-se uma análise do sistema utilizando um cenário onde os nós se movimentam, e com uma maior carga de mensagens na rede. A análise apresentada utiliza apenas a rede sobreposta assimétrica.

A Figura 4.9 apresenta os resultados obtidos num cenário com os seguintes parâmetros:

Número de nós	50
Número de editores	5
Taxa de publicação	1 mensagem por segundo
TTL de publicação	entre 1 e 18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	entre 5 e 10
Velocidade dos nós	3 m/s
Duração da simulação	700s
Área	1000m x 1000m

Comparando os resultados apresentados na Figura 4.9 com os previamente expostos na Figura 4.2, é possível observar que para baixos valores de *TTL* de publicação os valores de entrega da simulação com mobilidade são superiores. Este fenómeno acontece devido ao maior número de nós com informação de subscrição no sistema. Considere-se um sistema de 50 nós com 5 nós a subscrever um evento 'A', enviando 3 mensagens de percurso aleatório com um *TTL* de 2 saltos para cada subscrição. Como resultado, cada subscritor envia informação de subscrição para um máximo de 6 nós na rede. Quando um nó publica um evento 'A' para a rede, este envia uma mensagem de percurso aleatório com um *TTL* de 1 salto, ou seja, o nó editor envia uma publicação para um único elemento da sua vista activa. Tendo em conta que existem 5 subscritores na rede para este tipo de evento, e que cada subscritor pode propagar informação de subscrição para um máximo de 6 nós, pode existir um máximo de 30 nós na rede com informação de subscrição. Se a mensagem de publicação contactar um nó com informação de subscrição, este nó envia ROUTED MESSAGES para todos os endereços na tabela de encaminhamento referentes ao evento 'A'. Estas ROUTED MESSAGES têm um *TTL* igual ao *TTL* das mensagens de subscrição (neste caso 2), e ao percorrerem a rede podem passar por outros nós com informação de subscrição para o evento 'A'. Se o fizerem, as novas ROUTED MESSAGES geradas irão ter um *TTL* igual ao *TTL* restante na ROUTED MESSAGE que os gerou. Como tal, a propagação de ROUTED MESSAGES pela rede permite uma maior disseminação das publicações efectuadas, mas ao mesmo tempo introduz um número considerável de mensagens na rede. Logo, para cenários com muitos subscritores de um dado tipo de evento, a quantidade de informação de subscrição que estes propagam para a rede deve ser menor do que em cenários com menos subscritores, de modo a minimizar a quantidade de informação redundante na rede. Observando a Figura 4.2 pode-se constatar que o aumento da informação de subscrição na rede, bem como a utilização de mensagens de publicação mais longas não resulta numa melhoria da taxa de entrega.

De modo a verificar o comportamento do sistema num cenário com menos subscritores, reformula-se a simulação com apenas 3 editores e 5 subscritores. Todos os

restantes parâmetros permanecem inalterados. A Figura 4.10 apresenta resultados de entrega para a simulação proposta, como se pode verificar, neste caso o aumento da informação de subscrição e do *TTL* das mensagens de publicação tem um efeito positivo na taxa de entrega.

#### 4.5.1 Influência da Mobilidade

Nesta secção apresentam-se resultados referentes ao funcionamento do sistema DSR-WALK num cenário com maior mobilidade dos elementos da rede. Este tipo de teste permite observar o impacto que a mobilidade dos nós tem no funcionamento do sistema apresentado, e perceber também em que medida a utilização de diferentes parâmetros de configuração influencia a sua eficácia e eficiência. Considerando que o protocolo DSR, utilizado como primitiva de comunicação, necessita de propagar mais mensagens de controlo em ambientes de maior mobilidade, é expectável observar um aumento de carga na rede.

Para as simulações desta secção utilizaram-se os seguintes parâmetros de configuração:

Número de nós	50
Número de editores	5
Taxa de publicação	1 mensagem por segundo
TTL de publicação	entre 1 e 18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	entre 5 e 10
Velocidade dos nós	20 m/s
Duração da simulação	700s
Área	1000m x 1000m

A Figura 4.11 apresenta os resultados obtidos de simulações efectuadas utilizando um modelo de movimento em que os nós se deslocam a 20 m/s. Neste cenário a mobilidade dos nós obriga o protocolo DSR a recalcular rotas frequentemente, introduzindo assim carga na rede sob a forma de mensagens de controlo DSR. Neste tipo de situações, o aumento da informação de subscrição na rede e/ou o aumento do *TTL* das mensagens de publicação podem ser prejudiciais para o funcionamento do sistema, dado que a quantidade de mensagens na rede pode ser de tal maneira alta que a proporção de mensagens perdidas por colisão afecta a taxa de entrega. Os resultados da Figura 4.11 permitem concluir que neste tipo de cenário a utilização de mais informação de subscrição e/ou *TTLs* de publicação maiores prejudicam a entrega de eventos. apresentam-se resultados relativos a simulações que visam determinar o comportamento do sistema face a uma crescente mobilidade dos nós da rede.

Na Figura 4.11 pode-se observar o efeito que a crescente mobilidade tem na entrega de mensagens. Neste tipo de cenário a utilização de muita informação de subscrição mostra-se menos eficaz dada a volatilidade da validade desta informação para cenários de crescente mobilidade. Adicionalmente, dado que uma maior mobilidade dos elementos da rede interfere com a capacidade do protocolo DSR de manter rotas válidas nas *caches* dos nós, o número de mensagens de controlo na rede também sobe (como se pode observar comparando a carga na rede neste cenário com a carga no cenário anterior).

### 4.5.2 Influência da Carga na Rede

Quando dois nós se encontram em alcance mútuo, se ambos enviarem uma mensagem simultaneamente, ocorre uma colisão e ambas as mensagens são perdidas. O aumento de carga na rede potencia a ocorrência deste tipo de situações, nomeadamente quando os nós se encontram geograficamente próximos uns dos outros. De modo a avaliar como é que o sistema reage perante um aumento de carga, e que efeito a alteração de parâmetros tem neste tipo de cenários, efectuaram-se simulações utilizando um padrão de carga mais elevado que os previamente apresentados.

Os parâmetros das simulações apresentadas são:

Número de nós	50
Número de editores	5
Taxa de publicação	10 mensagem por segundo
TTL de publicação	entre 1 e 18
Número de subscritores	10 com interesse em todos os tópicos
TTL de subscrição	2 saltos
Número de mensagens de subscrição	entre 5 e 10
Velocidade dos nós	3 m/s
Duração da simulação	700s
Área	1000m x 1000m

A Figura 4.12 apresenta resultados obtidos de simulações efectuadas para determinar a influência de carga adicional no sistema. Neste cenário ocorre um fenómeno semelhante ao apresentado na simulação anterior, onde a quantidade de mensagens na rede é impeditiva do correcto funcionamento do protocolo, e o aumento da quantidade de mensagens na rede serve apenas para deteriorar mais a sua capacidade de entrega .

### 4.5.3 Comportamento numa Rede com Filiação Parcial

Uma das principais características do sistema apresentado é a sua capacidade de poder operar numa rede onde apenas um subconjunto dos seus elementos fazem parte

do sistema. De seguida apresentam-se dados relativos a simulações que pretendem quantificar a capacidade do sistema de operar neste tipo de cenário.

Os parâmetros utilizados para a simulação foram os seguintes:

Número de nós	250 (50 fazem parte do sistema E/S)
Número de editores	1
Taxa de publicação	1 mensagem de 3 em 3 segundos
TTL de publicação	entre 1 e 18
Número de subscritores	5
TTL de subscrição	2 saltos
Número de mensagens de subscrição	entre 5 e 10
Velocidade dos nós	3 m/s
Duração da simulação	700s
Área	2000m x 2000m

A Figura 4.13 apresenta os resultados relativos às simulações efectuadas para o cenário com uma rede híbrida de 250 nós. Numa rede com tantos elementos, o mecanismo de descoberta de rotas do protocolo DSR (que utiliza inundação) introduz muitas mensagens na rede. Adicionalmente, a distância média em saltos entre dois nós arbitrários da rede sobreposta é maior do que num cenário com 50 nós. Estes factores contribuem para o aumento do número de mensagens geradas na rede, e também para um aumento da proporção de mensagens perdidas. De notar que para alcançar resultados minimamente satisfatórios para este cenário utilizou-se um padrão de tráfego muito menor do que nos cenários previamente apresentados, nomeadamente, nestas simulações existe apenas um editor que envia eventos de 3 em 3 segundos.

## 4.6 Discussão

Neste capítulo apresentaram-se resultados obtidos por simulação que permitem determinar a eficácia e eficiência do sistema DSR-WALK perante diferentes cenários de operação. As seguintes conclusões podem ser extraídas dos dados expostos:

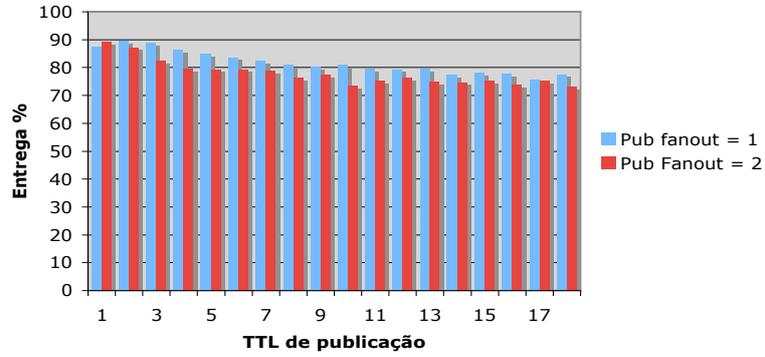
- **Inadequação da rede sobreposta simétrica** - A concretização apresentada para a rede sobreposta simétrica mostrou-se inadequada para operação no tipo de cenário utilizado. A grande quantidade de mensagens de controlo e o elevado número de colisões de pacotes contribuem para limitar a capacidade de operação do sistema quando utiliza esta rede sobreposta.
- **Distribuição desigual de carga/grau na rede** - A rede sobreposta assimétrica utilizada foi construída com base em princípios probabilistas, no entanto a distribuição geográfica dos nós e a disponibilidade de informação

de encaminhamento influenciam a sua formação. Consequentemente, a rede sobreposta não garante uma distribuição uniforme do grau pelos elementos da rede, resultando numa distribuição desigual da carga entre os mesmos.

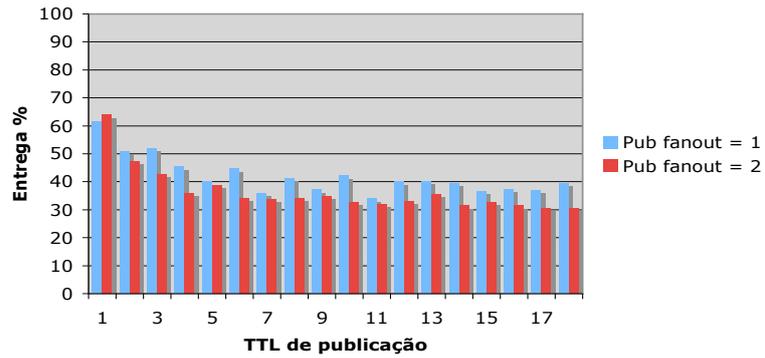
- **Propagação de subscrições dependente do número de subscritores -** A interpretação dos resultados apresentados pela Figura 4.10 permitiu determinar que com o aumento do número de subscritores com interesses em comum verifica-se também um aumento da probabilidade de entrega de publicações. Este fenómeno deve-se à propagação de informação de subscrição pelos nós, e à criação de novas ROUTED MESSAGES quando uma mensagem de percurso aleatório encontra informação de subscrição.
- **O efeito da alteração de parâmetros depende do cenário -** As simulações efectuadas permitiram determinar o efeito que a alteração de parâmetros de configuração tem na operação do sistema. Consegue-se constatar que, para cenários de maior mobilidade ou carga na rede, a utilização de mais informação de subscrição e de mensagens de publicação mais longas não favorece a capacidade de entrega do sistema. Na realidade foi possível observar que a introdução de mais carga na rede pode mesmo deteriorar o comportamento do protocolo apresentado.

**Rede sobreposta assimétrica - 5 editores, 10 subscritores  
1 ou 2 mensagens de publicação**

**Entrega de eventos - 1 mensagem por segundo,  
movimento 3 m/s**



**Entrega de eventos - 1 mensagem por segundo,  
movimento 20 m/s**



**Entrega de eventos - 10 mensagens por segundo,  
movimento 3 m/s**

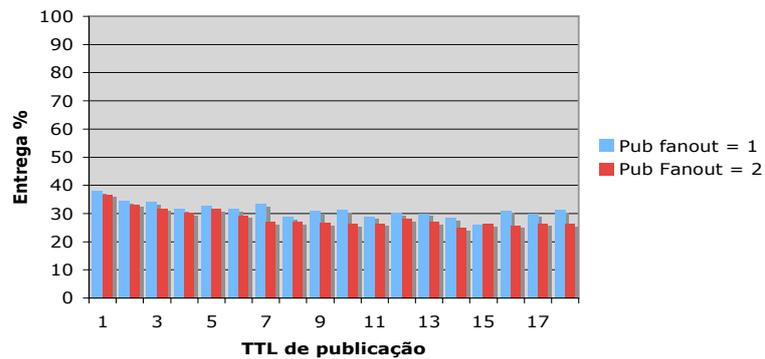


Figura 4.7: Rede sobreposta assimétrica com *fanout* de publicação variável

**Rede sobreposta simétrica - 5 editores, 10 subscritores  
1 mensagem por segundo , movimento 0 m/s**

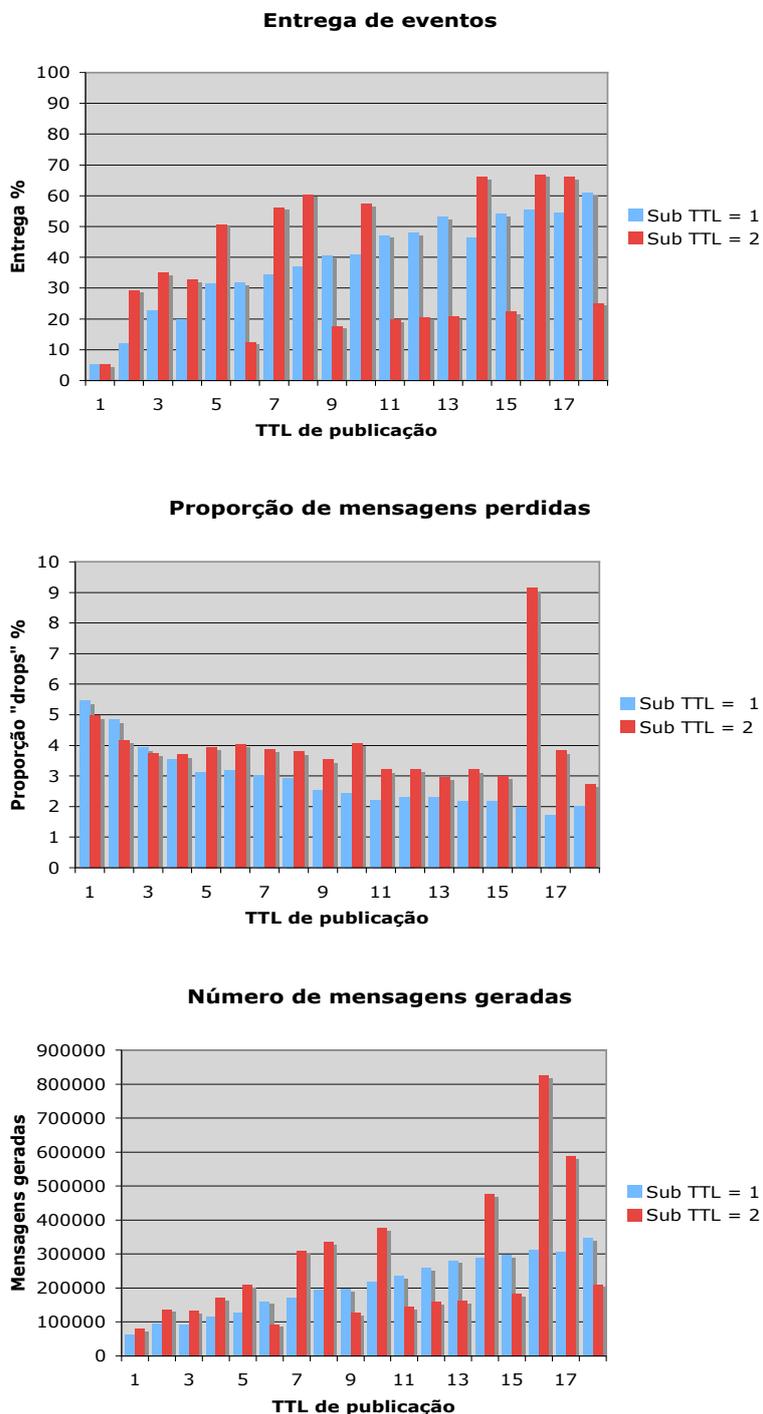


Figura 4.8: Rede sobreposta simétrica

**Rede sobreposta assimétrica, 5 editores 10 subscritores  
1 mensagem por segundo, movimento 3 m/s**

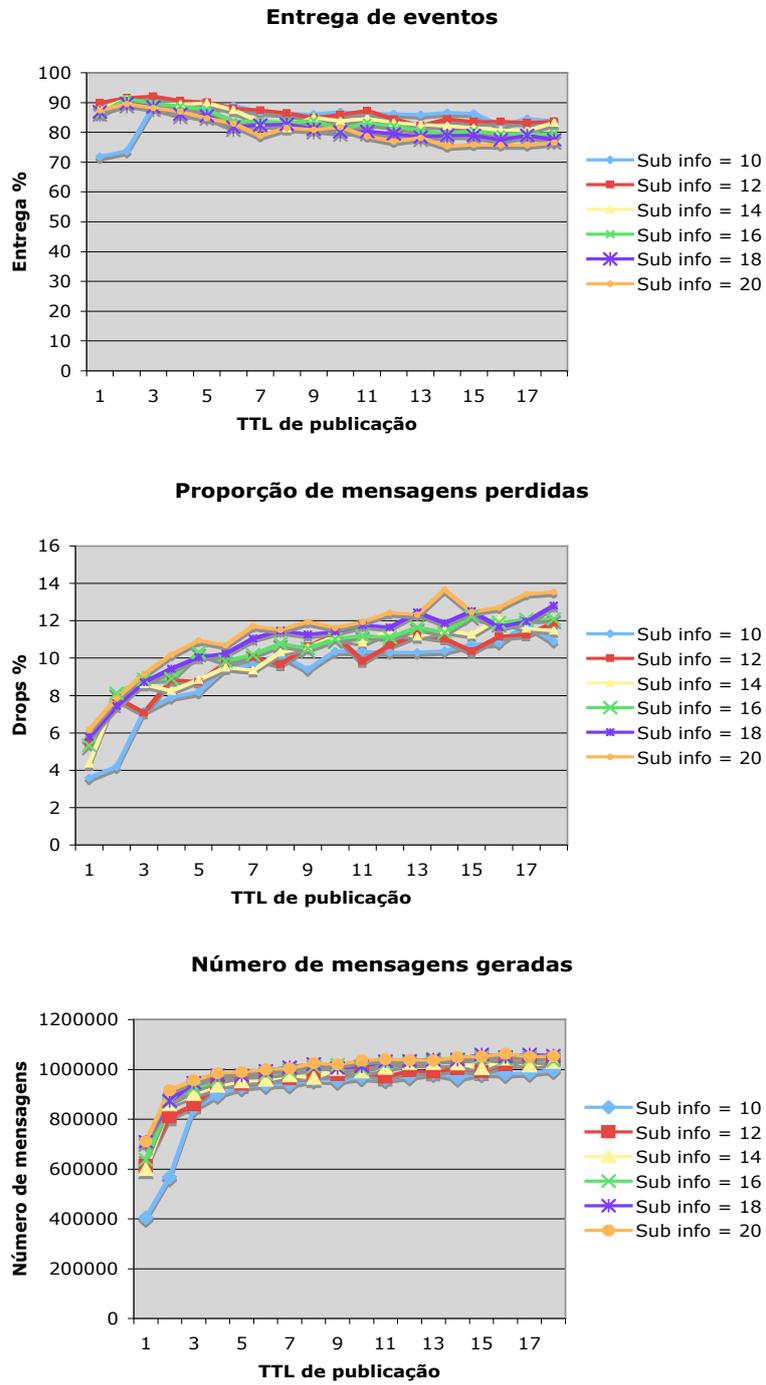


Figura 4.9: Rede sobreposta assimétrica (3 m/s) , 5 editores e 10 subscritores, 1 mensagem por segundo

**Rede sobreposta assimétrica, 3 editores 5 subscritores  
1 mensagem por segundo, movimento 3 m/s**

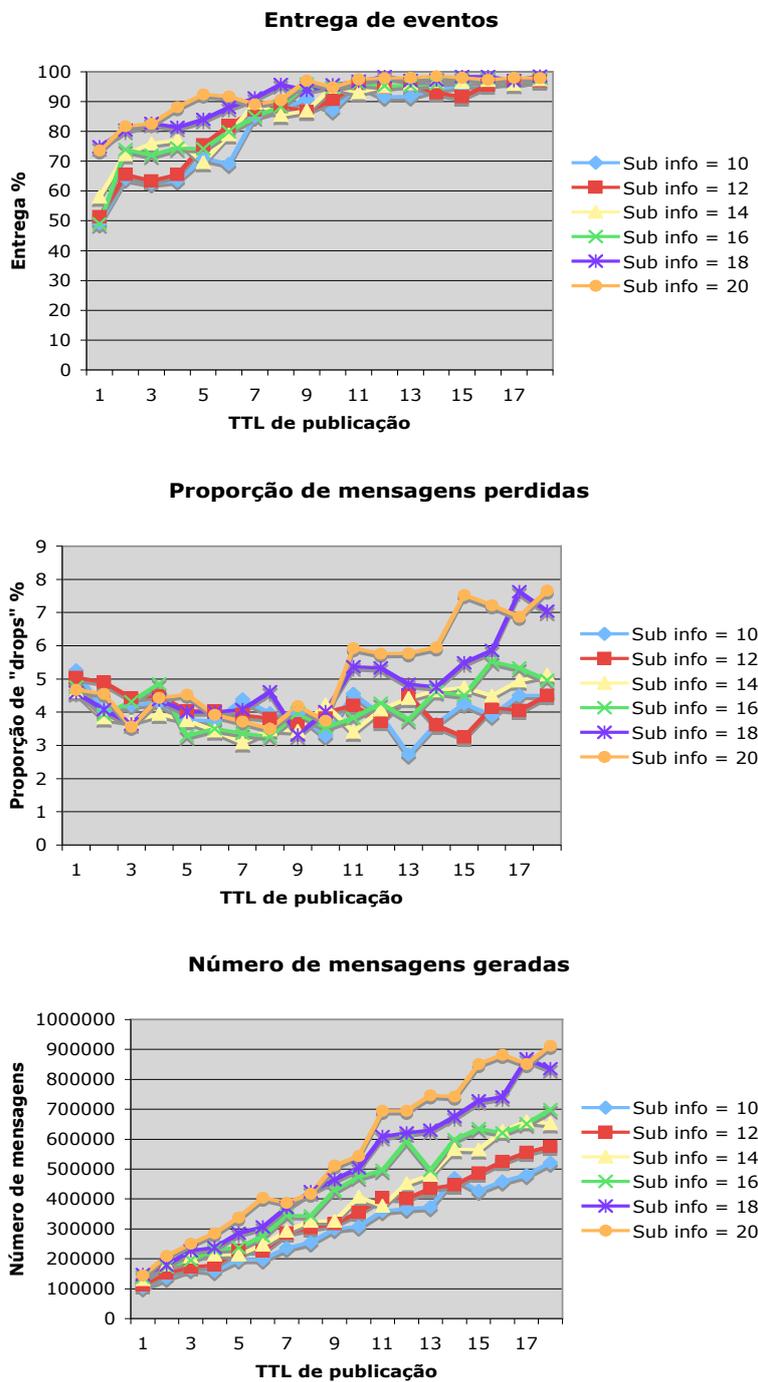


Figura 4.10: Entrega de pacotes com 3 editores e 5 subscritores

**Rede sobreposta assimétrica, 5 editores 10 subscritores  
1 mensagem por segundo, movimento 20 m/s**

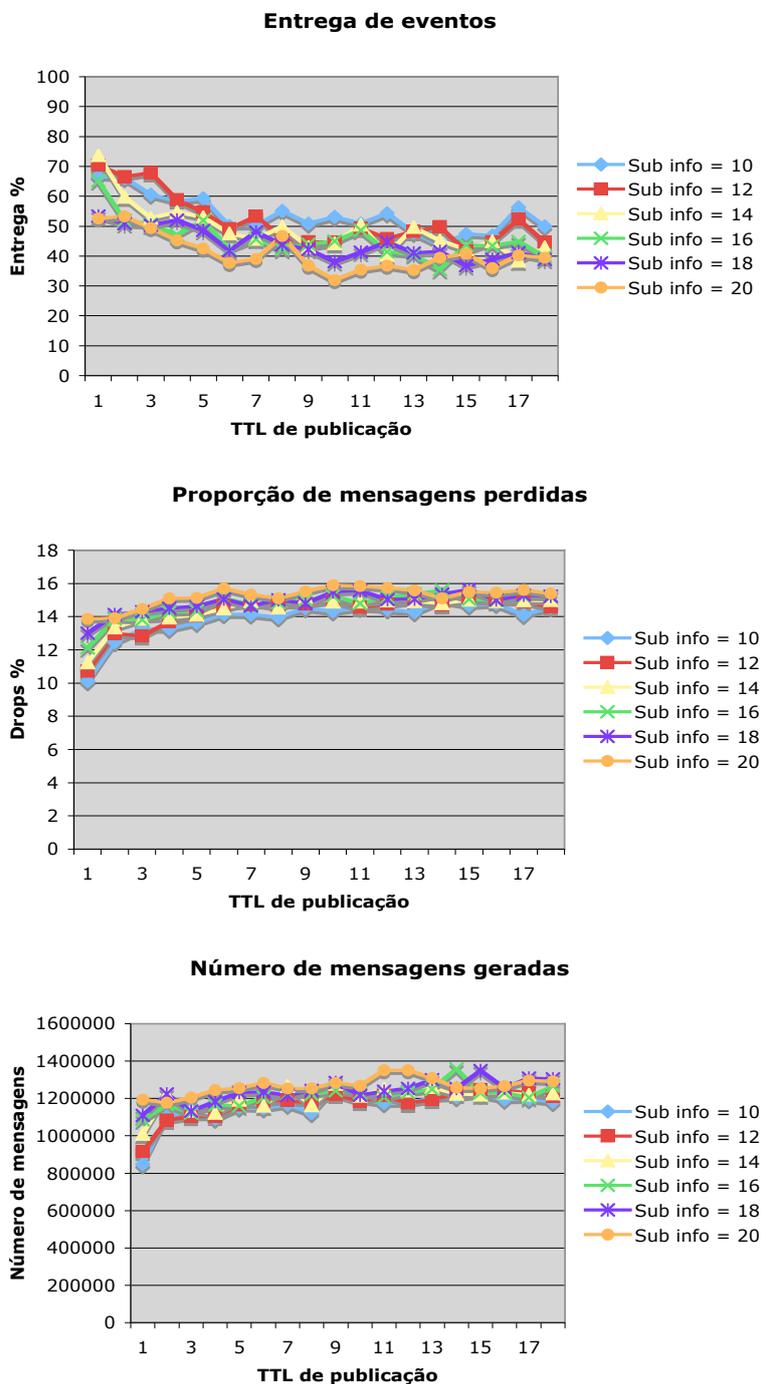


Figura 4.11: Rede com 50 nós movendo-se a 20 m/s constantes

**Rede sobreposta assimétrica, 5 editores 10 subscritores  
10 mensagem por segundo, movimento 3 m/s**

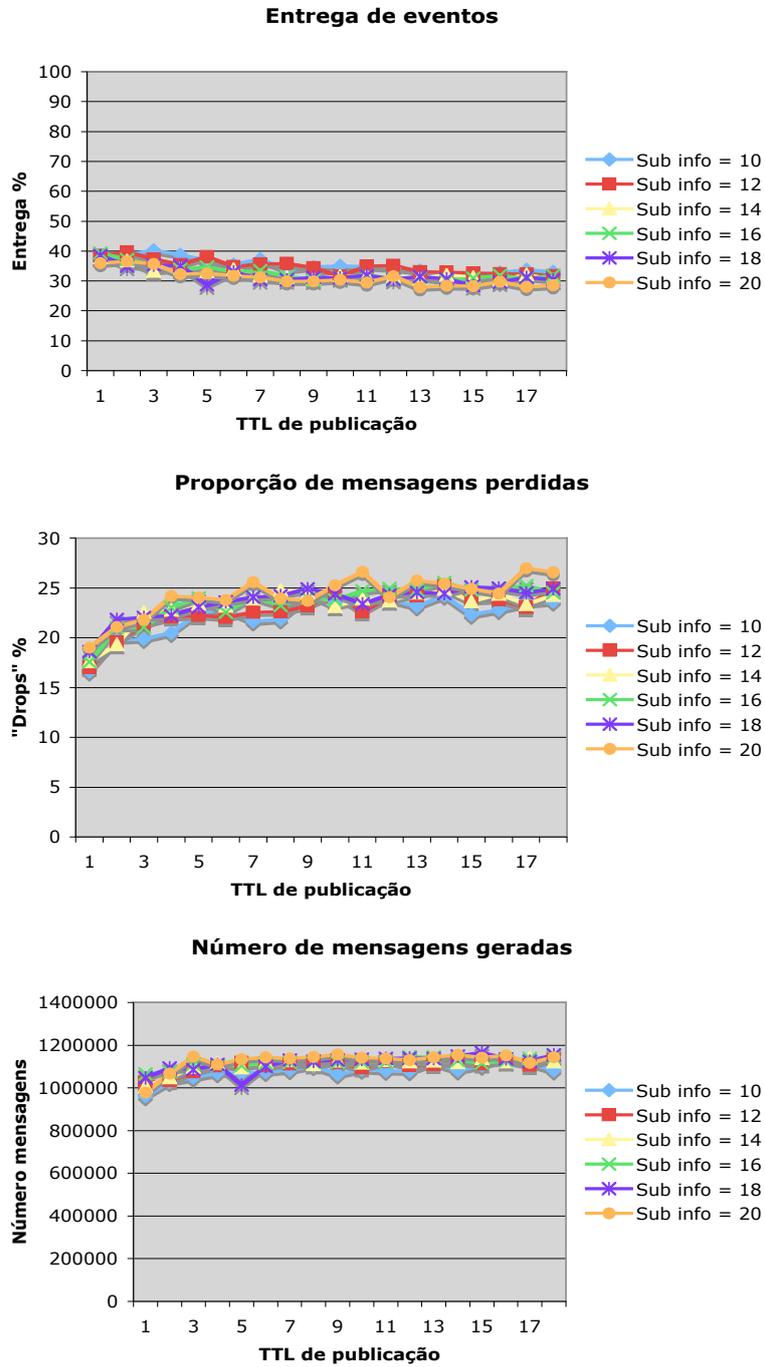


Figura 4.12: Rede com 5 editores a publicar 10 mensagens segundo

**Entrega numa rede sobreposta assimétrica  
com filiação parcial - 1 editor, 5 subscritores  
1 mensagem de 3 em 3 segundos 3 m/s**

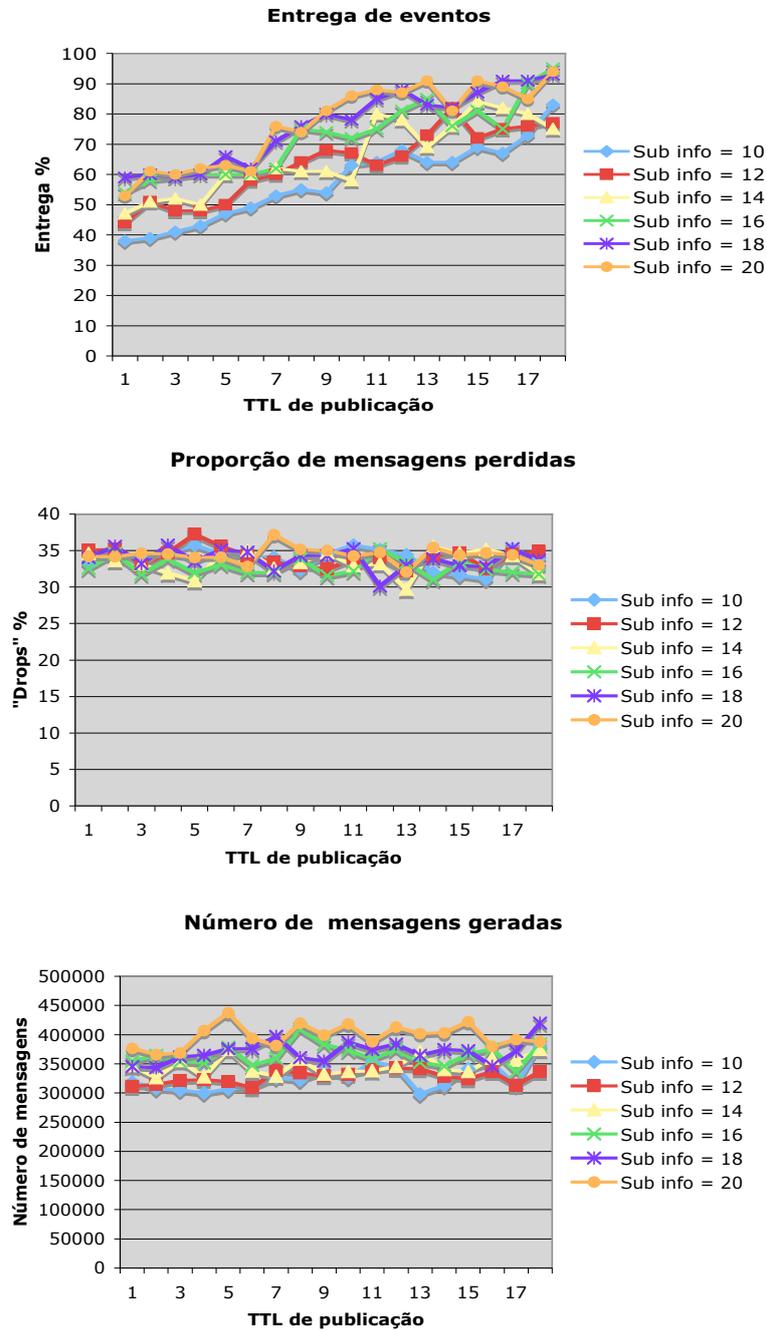


Figura 4.13: Rede com 250 nós

# Capítulo 5

## Conclusão

### 5.1 Conclusões

Nesta dissertação apresenta-se uma solução para concretizar o paradigma editor/subscritor numa rede MANet. Esta solução utiliza uma aproximação semi-probabilística aliada ao protocolo de encaminhamento DSR para permitir uma operação robusta num cenário dinâmico e imprevisível como o das redes MANet.

A utilização do protocolo DSR como base para o paradigma editor/subscritor numa rede MANet apresenta novas possibilidades e desafios. Por um lado, a utilização do protocolo DSR permite abstrair a solução proposta das alterações à topologia da rede, por outro, introduz um custo em mensagens que se pode tornar incomportável. Neste trabalho implementou-se uma rede sobreposta probabilística com baixo custo de manutenção, e que ao mesmo tempo permite minimizar a sobrecarga imposta pelo protocolo DSR tem na operação do sistema. Desta forma, conseguiu-se obter uma base sólida para desenvolver um protocolo de edição/subscrição semi-probabilístico, que é capaz de operar num ambiente dinâmico como o das redes MANet. O resultado é um sistema escalável e configurável, capaz de operar perante falhas arbitrárias de nós na rede, dado que muito pouca (ou mesmo em certos casos, nenhuma) informação determinista é utilizada para assegurar o correcto funcionamento do protocolo.

Experimentaram-se duas aproximações radicalmente diferentes para a construção da camada de rede sobreposta. Em primeiro lugar desenvolveu-se uma rede sobreposta cujo funcionamento pretendia assegurar a existência de vistas simétricas entre os elementos da rede. No entanto, esta solução mostrou-se inadequada para utilização numa rede MANet, devido à quantidade excessiva de mensagens de controlo geradas pela cama de rede sobreposta, e também devido à fragilidade do protocolo perante mensagens perdidas. A segunda aproximação dispensa a utilização de vistas simétricas em benefício de uma menor complexidade algorítmica, e numa menor carga de mensagens de controlo na rede. Esta aproximação revelou-se mais adequada

para o uso em redes MANet, como se pode comprovar pelos resultados obtidos por simulação.

A possibilidade de se utilizar a solução proposta num ambiente em que apenas um subconjunto dos elementos da rede pertencem ao sistema de edição/subscrição foi uma das motivações iniciais para este trabalho. Num possível cenário futuro em que a maioria dos dispositivos com comunicação móvel utilizassem o protocolo DSR (ou outro protocolo com características semelhantes), uma solução como a proposta permite tomar partido de elementos da rede independentes ao sistema, para melhorar a conectividade entre os intervenientes do protocolo de edição/subscrição. Os resultados obtidos neste trabalho demonstram que embora existam algumas dificuldades acrescidas quando se opera neste tipo de cenário (como por exemplo maior proporção de perda de mensagens), o sistema proposto consegue assegurar uma taxa de entrega de eventos perto dos 90% em cenários com pouca carga e mobilidade na rede. De notar que a qualidade do sistema proposto perante este tipo de cenários depende sempre da capacidade do protocolo DSR de operar em redes com muitos nós, e com mobilidade.

Embora o protocolo DSR permita efectuar encaminhamento em redes MANet, este utiliza mecanismos de descoberta de rotas cujo custo em termos de número de mensagens trocadas pode inviabilizar a operação do sistema. As optimizações efectuadas para minimizar o impacto destes mecanismos permitiram reduzir o número de mensagens redundantes na rede, e conseqüentemente, aumentar a carga máxima suportada pelo sistema, melhorar as taxas de entrega, e diminuir a proporção de mensagens perdidas.

Os resultados obtidos por simulação permitiram quantificar a capacidade de operação do sistema proposto perante diferentes condições de mobilidade e carga na rede. Como seria expectável, em situações de grande mobilidade ou carga na rede, a taxa de entrega de eventos diminui, e a proporção de mensagens perdidas aumenta, sendo embora possível diminuir este efeito utilizando parâmetros do sistema que introduzam menos carga na rede (*TTL* de publicações, quantidade de informação de subscrição na rede, etc...). Em cenários de menor mobilidade e carga na rede, o sistema apresentou resultados mais satisfatórios, alcançando em certas simulações taxas de entrega de eventos na ordem dos 90%. Adicionalmente, este tipo de cenários apresentam uma menor proporção de mensagens perdidas e uma menor quantidade de mensagens de controlo na rede.

## 5.2 Trabalho futuro

Existem diversas oportunidades de melhoria ou extensão no trabalho proposto, nomeadamente:

- **Reformulação do protocolo da rede sobreposta simétrica:** A solução proposta para concretizar a rede sobreposta simétrica demonstrou-se inadequada para o tipo de cenário de operação utilizado. Como tal, seria vantajoso estudar uma alternativa mais adequada a este tipo de cenário, onde a criação das vistas simétricas fosse menos rígida e gerasse menos mensagens de controlo do que na solução apresentada.
- **Criação dinâmica de *super-peers*:** Um fenómeno interessante observado no decorrer deste trabalho foi a existência de nós cuja popularidade os tornava em pontos-de-encontro para publicações e subscrições. Seria interessante estudar um mecanismo que permitisse a determinados nós da rede (ou pré-seleccionados ou escolhidos dinamicamente) influenciar o seu grau interno de modo a torná-los *super-peers*. Esta aproximação poderia ser útil em cenários com nós com capacidade de operação variável, onde fosse possível discernir os melhores candidatos para esta tarefa, utilizando assim estes nós como pontos-de-encontro para publicações e subscrições. Adicionalmente, seria interessante também estudar a possibilidade de criar dinamicamente nós com funcionalidades típicas de *broker*, como o armazenamento de mensagens antigas.
- **Reconfiguração automática de parâmetros:** Um problema a resolver no sistema proposto nesta dissertação passa pela reconfiguração automática dos parâmetros de operação. Como já foi explicado, o protocolo apresentado depende de vários parâmetros essenciais cujos valores influenciam drasticamente o funcionamento do sistema perante diferentes cenários de mobilidade e de carga na rede. Embora esta flexibilidade possa ser útil caso se deseje adaptar o protocolo a um cenário previamente conhecido, cujas características se mantenham inalteradas durante a operação do sistema, seria verdadeiramente útil ter um mecanismo que permitisse reconfigurar automaticamente estes parâmetros perante mudanças inesperadas na natureza do cenário de operação.
- **Avaliação estendida:** Devido ao número considerável de variáveis de que o sistema proposto depende, tornou-se impraticável avaliar por simulação todas as combinações possíveis de parâmetros de configuração. No entanto, seria útil estudar o efeito da utilização de um número variável de mensagens de percurso aleatório na rede para publicações. Intuitivamente, a utilização de várias mensagens curtas em vez de uma única mensagem longa pode, em certos cenários, ser preferível devido a diversos factores como: 1) existência de várias cópias da publicação na rede, o que, em caso de perda de uma mensagem não compromete imediatamente a entrega dessa publicação; e 2) entrega de publicações em menos saltos na rede sobreposta.

A proliferação de dispositivos com capacidade de comunicação sem fios permite prever o aparecimento de novos tipos de redes e aplicações, como redes *mesh* urbanas, onde os utilizadores do serviço cooperam entre si para fornecer conectividade e o sistema como um todo pode não depender de uma estrutura fixa. Este tipo de aplicações poderá vir a ser importante no futuro nomeadamente em situações em que a utilização de uma estrutura fixa não seja viável, como por exemplo, quando não seja possível a colocação de infra-estrutura ou quando o número de utilizadores seja excessivamente grande para a infra-estrutura existente. Como tal, a comunidade científica estuda actualmente soluções que possibilitem o desenvolvimento de aplicações distribuídas para este tipo de rede. É no âmbito deste esforço que este trabalho se insere, visando criar uma abstracção que permita a disseminação de informação por um número arbitrário de elementos numa rede MANet.

# Índice

- Avaliação, 49
  - Discussão, 65
- Efeitos da mobilidade e carga, 61
  - Carga, 64
  - Fiação parcial, 64
  - Mobilidade, 63
- Métricas, 50
- NS2, 49
- Parâmetros, 49
- Rede estática, 51
  - Modelação, 52
  - Número percursos aleatórios, 59
  - Rede sobreposta simétrica, 60
  - Simulação, 52
- Conclusão, 74
- DSR-WALK, 31
  - RDG*, 34
  - cache de encaminhamento, 34
  - DSR, 34
  - Edição/Subscrição, 43
    - Distribuição geométrica, 45
    - Parâmetros, 47
    - Publicação de eventos, 46
    - Subscrição de eventos, 45
  - Rede sobreposta, 34
    - vistas assimétricas, 40
    - vistas simétricas, 35
- Editor/Subscritor, 2, 14
  - Broker*, 14
  - Desagregação de sincronização, 2, 14
  - Desagregação espacial, 2, 14
  - Desagregação temporal, 2, 14
  - Subscrição por conteúdo, 16
  - Subscrição por tópicos, 16
  - Subscrição por tipos, 17
- Encaminhamento em Redes *MANet*, 9, 11
  - AODV, 10
  - DSR, 10
  - MAODV, 11
- Encaminhamento em redes *MANet*
  - Encaminhamento probabilístico de eventos, 24
  - Encaminhamento selectivo de eventos, 23
  - Inundação de eventos, 22
- Encaminhamento Multicast em Redes *MANet*
  - RDG*, 12
- MANet*, 1, 5
- Modelos de movimento, 7
  - Freeway, 8
  - Manhattan, 8
  - RandomWaypoint, 9
- Parâmetros, 47
  - Fanout Publicação, 48
  - Fanout Subscrição, 48
  - TTL Publicação, 47
  - TTL Subscrição, 47
- Rede Sobreposta
  - Estruturada, 20
  - Não Estruturada, 21

Rede sobreposta, 19

*Broker overlay*, 19

*Super-peer*, 19

*Topologia em árvore*, 20

*Topologia em anel*, 19

*Topologia em polígono*, 20

Sistemas, 25

*Gryphon*, 25

*Q*, 28

*SIENA*, 26

*Scribe*, 25

*Semi-Probabilistic Publish/Subscribe*,  
26

Trabalho futuro, 75

# Bibliografia

- [1] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Symposium on Operating Systems Principles*, pages 131–145, 2001.
- [2] Marco Avvenuti, Alessio Vecchio, and Giovanni Turi. A cross-layer approach for publish/subscribe in mobile ad hoc networks. In *MATA 05*, 2005.
- [3] F. Bai, N. Sadagopan, and A. Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks, 2003.
- [4] R. Baldoni and A. Virgillito. Distributed Event Routing in Publish/Subscribe Communication Systems: a Survey (revised version). Technical report, MIDLAB 1/2006 - Dipartimento di Informatica e Sistemistica A.Ruberti, Università di Roma la Sapienza, 2006.
- [5] A. Bharambe, S. Rao, and S. Seshan. Mercury: A scalable publish-subscribe system for internet games.
- [6] Kenneth P. Birman, Mark Hayden, Ozgur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. *ACM Trans. Comput. Syst.*, 17(2):41–88, 1999.
- [7] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001.
- [8] Antonio Carzaniga, Matthew J. Rutherford, and Alexander L. Wolf. A routing scheme for content-based networking. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.
- [9] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications (JSAC)*, 20(8):1489–1499, 2002.

- 
- [10] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations, 1999.
  - [11] Paolo Costa and Gian Pietro Picco. Semi-probabilistic content-based publish-subscribe. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 575–585, Washington, DC, USA, 2005. IEEE Computer Society.
  - [12] P. Eugster and R. Guerraoui. Probabilistic multicast, 2002.
  - [13] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Ker-marrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.
  - [14] D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.
  - [15] Bhaskar Krishnamachari, Xi Xie, Bart Selman, and Stephen B. Wicker. Analysis of random noise and random walk algorithms. In *Principles and Practice of Constraint Programming*, pages 278–290, 2000.
  - [16] Y. Liu and B. Plale. Survey of publish subscribe event systems, 2003.
  - [17] J. Luo, P. Eugster, and J. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networking, 2002.
  - [18] R. Ogier. Topology dissemination based on reversepath forwarding (tbrpf, 2004.
  - [19] C. Perkins. Ad hoc on demand distance vector (aodv) routing, 1997.
  - [20] P. Pietzuch. Event-based middleware: A new paradigm for wide-area distributed systems, 2002.
  - [21] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.
  - [22] E. Royer and C. Perkins. Multicast ad hoc on- demand distance vector (maodv) routing, 2000.
  - [23] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols, 2003.

- 
- [24] R. Strom, G. Banavar, T. Chandra, M. Kaplan, K. Miller, B. Mukherjee, D. Sturman, and M. Ward. Gryphon: An information flow based approach to message brokering. In *International Symposium on Software Reliability Engineering (ISSRE '98)*, 1998.
- [25] XO. <http://laptop.org/>, 2006.
- [26] Y. Yi and S. Lee. Demand multicast routing protocol, 2002.
- [27] A. Zeidler and L. Fiege. Mobility support with rebecca, 2003.
- [28] Shelley Q. Zhuang, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz, and John D. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *Proceedings of NOSSDAV*, June 2001.