
Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
Ano 2011

Tomé Frederico Guimarães Gomes

COMUNICAÇÃO ENTRE ELEMENTOS DA REDE NUMA GESTÃO AUTONÓMICA

Universidade de Aveiro
Ano 2011
Departamento de Electrónica, Telecomunicações e Informática

Tomé Frederico Guimarães Gomes

COMUNICAÇÃO ENTRE ELEMENTOS DA REDE NUMA GESTÃO AUTONÓMICA

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Prof. Dra. Susana Sargento, Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Prof. Dr. Paulo Salvador, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

A ti que continuas a existir.

o júri

presidente	Prof. Doutor Nuno Borges de Carvalho professor associado com agregação do Departamento de Engenharia Electrónica Telecomunicações e Informática da Universidade de Aveiro
orientadora	Prof. Doutora Susana Sargento professora auxiliar do Departamento de Engenharia Electrónica Telecomunicações e Informática da Universidade de Aveiro
co-orientador	Prof. Doutor Paulo Salvador professor auxiliar do Departamento de Engenharia Electrónica Telecomunicações e Informática da Universidade de Aveiro
vogal	Prof. Doutor Pedro Sousa professor auxiliar do Departamento de Engenharia Informática da Escola de Engenharia da Universidade do Minho

agradecimentos

Agradeço à minha família, amigos e professores envolvidos no meu percurso académico.

palavras-chave

Redes de próxima geração, Gestão distribuída e autónoma, In-Network Management, Redes sem fios, Protocolo 802.11, Camada MAC, Ad hoc, Métrica social, NS-3, Redes com fios, Hide & Seek, *testbed.*

resumo

Com o aumento da dimensão, complexidade e dinamismo das redes de próxima geração, os protocolos de gestão tradicionais tornar-se-ão ineficientes devido às suas características centralizadas e limitações em termos de escalabilidade.

É neste ponto que surge a necessidade de criar diferentes soluções de gestão de rede que respondam a requisitos como a automatização da resposta às alterações nas condições do meio, a optimização de recursos, a adaptabilidade às mudanças de topologia, a escalabilidade e a eficiência de processos.

Como resposta às necessidades apresentadas surge o paradigma de *In-Network Management* (INM) cuja ideia principal é inserir nas várias entidades que compõem a rede, capacidades e funcionalidades de forma a que estas se tornem autónomas e a gestão da rede deixe de depender de entidades agregadoras externas/servidores. Isto implica um nível elevado de automatização, sincronismo e actualização da informação entre os vários elementos, podendo facilmente conduzir a um forte aumento do *overhead*.

É no contexto do paradigma de INM que a presente dissertação se insere, na qual são propostos vários mecanismos de interacção entre entidades, de modo a incluir processos de comunicação cooperativa. De modo sucinto, ao nível das redes com fios, é testado um protocolo de descoberta, baseado na estratégia *Hide & Seek*. Ao nível das redes sem fios, usando como base o protocolo 802.11 MAC, são propostos mecanismos de comunicação entre entidades e um critério de classificação baseado em relações sociais entre nós (métrica social).

As soluções apresentadas são avaliadas segundo diferentes parâmetros, em cenários com e sem fios. Os cenários de redes com fios foram avaliados numa *testbed* virtual, ao nível do *overhead* de mensagens e do tempo de convergência da informação do protocolo de descoberta desenvolvido, tendo sido ainda elaborada uma análise comparativa com outros protocolos. Nos cenários de redes sem fios implementados no simulador NS-3, analisou-se o impacto ao nível de alguns parâmetros de nível MAC. É objectivo futuro a avaliação do impacto ao nível da camada IP.

keywords

Next generation networks, Distributed and Autonomous Management, In-Network Management, Wireless Networks, 802.11 Protocol, MAC layer, Ad hoc, Social metric, NS-3, Wired Networks, Hide & Seek, testbed.

abstract

With the increasing size, complexity and dynamism of next generation networks, the traditional management protocols will become highly inefficient due to their centralizing characteristics and limitations in terms of scalability.

This is the point where the need to create different network management solutions that answer to requirements such as automation of the response to changes in environmental conditions, resource optimization, adaptability to topology changes, scalability and efficiency.

In response to the needs presented, the In-Network Management (INM) paradigm arises, whose main idea is to place capabilities and features into the various entities that comprise the network, so that they become autonomous and network management no longer rely on external servers. This implies a high level of automation, synchronization and update of the information among all elements, which can easily lead to a high increase in overhead.

This Master's Thesis works in part of the architecture proposed by the INM paradigm, in the proposed mechanisms of interaction between entities in order to enable the overall process of cooperative communication. Briefly, in the wired networks, we propose a discovery protocol, based on the Hide & Seek strategy. In terms of wireless networks, using the 802.11 MAC protocol as base, we propose mechanisms for communication between entities and a social based classification criterion (social metric).

The presented solutions are evaluated according to different parameters, in wired and wireless scenarios. In wired scenarios, the discovery protocol developed was evaluated in terms of messages overhead and convergence time on a virtual testbed, and it was also performed a comparative analysis with other protocols. The wireless scenario was implemented in the NS-3 simulator with an analysis of the impact on some MAC level parameters. It is also a future goal the evaluation of the impact on IP layer parameters.

Índice

Índice i

Lista de Figuras v

Lista de Tabelas vii

Lista de Acrónimos ix

1 Introdução 1

1.1 Motivação . . . 1
1.2 Objectivos . . . 2
1.3 Contribuições . . . 3
1.4 Organização . . . 3

2 Estado de Arte 5

2.1 Introdução . . . 5
2.2 Arquitecturas de gestão de redes . . . 5
2.2.1 Abordagens tradicionais . . . 5
2.2.1.1 Abordagem centralizada . . . 6
2.2.1.2 Abordagem distribuída . . . 7
2.2.1.3 Abordagem cooperativa . . . 8
2.2.2 Abordagens em redes de próxima geração . . . 9
2.3 *In-Network Management* . . . 12
2.3.1 Requisitos funcionais . . . 13
2.4 Comunicação entre nós e domínios . . . 14
2.4.1 Abordagens de *bootstrapping* . . . 14
2.4.2 Abordagens de descoberta . . . 15
2.4.3 Disseminação de informação . . . 16
2.4.4 Abordagens de eleição . . . 17
2.5 Gestão de redes sem fios pelo padrão 802.11 . . . 18
2.5.1 Arquitectura geral . . . 18
2.5.2 Camadas protocolares . . . 19
2.5.3 Modos de operação . . . 19
2.5.4 Modos de *scanning/probing* . . . 20
2.5.5 Espaçamento entre tramas . . . 20

2.5.6 CSMA/CA . . . 21
2.5.7 Pacotes ao nível MAC . . . 22
2.5.7.1 Tipos de pacotes . . . 22
2.5.7.2 Formato genérico . . . 24
2.6 Conclusões . . . 28

3 Mecanismos de Cooperação 31
3.1 Introdução . . . 31
3.2 Objectivos . . . 31
3.3 Arquitectura global . . . 32
3.4 Redes sem fios . . . 33
3.4.1 Interligação das funcionalidades e mecanismos . . . 33
3.4.2 Mecanismos e sua função . . . 33
3.4.3 Funcionamento dos principais mecanismos . . . 35
3.4.4 Processos e conceitos associados . . . 36
3.4.4.1 Associação ao melhor nó . . . 36
3.4.4.2 Perda de ligação ao nó associado . . . 37
3.4.4.3 Identificador da comunidade . . . 38
3.4.5 Métrica Social . . . 39
3.4.5.1 Objectivo . . . 39
3.4.5.2 Fórmula ponderada . . . 40
3.4.5.3 Parâmetros de entrada . . . 40
3.4.5.4 Intervalo adaptativo entre *Beacons* . . . 42
3.4.6 Complementos ao protocolo 802.11 MAC . . . 43
3.4.7 Desafios inerentes . . . 44
3.5 Redes com fios . . . 46
3.5.1 Mecanismos e sua função . . . 46
3.5.1.1 *Bootstrapping* . . . 47
3.5.1.2 Descoberta . . . 47
3.5.1.3 Eleição . . . 48
3.5.2 Desafios inerentes . . . 49
3.6 Conclusões . . . 49

4 Implementação 51
4.1 Introdução . . . 51
4.2 Redes sem fios . . . 51
4.2.1 Protocolo 802.11 MAC e PHY em NS-3 . . . 51
4.2.1.1 Visão geral . . . 51
4.2.1.2 Camada física . . . 52
4.2.1.3 Camada MAC de baixo nível . . . 52
4.2.1.4 Camada MAC de alto nível . . . 52
4.2.1.5 Interligação de camadas ao nível 2 . . . 52
4.2.1.6 Algoritmos de controlo de taxas de transmissão . . . 53
4.2.2 Solução implementada . . . 54

4.2.2.1 Modelos existentes . . . 54
4.2.2.2 Características gerais do modelo *ad hoc* implementado . . . 55
4.2.2.3 Canal de Comunicação . . . 55
4.2.2.4 Máquina de estados finitos . . . 56
4.2.2.5 *Bootstrapping* . . . 58
4.2.2.6 Recepção de pacotes de gestão . . . 58
4.2.2.7 Identificador da comunidade . . . 63
4.2.2.8 Tabelas dinâmicas de informação local . . . 64
4.2.2.9 Eleição . . . 65
4.2.2.10 Estimativa de parâmetros da comunidade . . . 66
4.2.2.11 Cenário de simulação . . . 68
4.2.3 Desafios enfrentados . . . 69
4.3 Redes com fios . . . 70
4.3.1 Módulos funcionais . . . 70
4.3.1.1 *Repository* . . . 70
4.3.1.2 *Proxy* . . . 71
4.3.1.3 *MmMsg* . . . 71
4.3.1.4 *Throughput* . . . 71
4.3.1.5 *TimeSampling* . . . 71
4.3.1.6 *InetAddrv6* . . . 72
4.3.1.7 *HelloReceivedLocal* . . . 72
4.3.1.8 *AuxFunctions* . . . 72
4.3.1.9 LnDiscover . . . 72
4.3.2 *Scripts* de inicialização . . . 72
4.3.3 *Testbed* . . . 73
4.4 Conclusões . . . 74

5 Resultados **75**
5.1 Introdução . . . 75
5.2 Redes sem fios . . . 75
5.2.1 Cenários usados . . . 75
5.2.2 Influência da métrica social e dos seus pârametros individuais . . . 76
5.2.2.1 Variação do número de nós em simulação . . . 76
5.2.2.2 Evolução das comunidades finais no tempo . . . 83
5.3 Redes com fios . . . 88
5.3.1 Apresentação e discussão de resultados . . . 88
5.4 Conclusões . . . 90

6 Conclusão e linhas futuras de investigação **93**

Bibliografia **95**

Lista de Figuras

2.1 Sistema de gestão clássica da rede 6
2.2 Esquema ilustrativo da gestão centralizada 7
2.3 Esquema ilustrativo da gestão fracamente distribuída 7
2.4 Esquema ilustrativo da gestão fortemente distribuída 8
2.5 Esquema ilustrativo da gestão cooperativa 8
2.6 Ciclo de controlo autonómico 9
2.7 Comparação entre as várias abordagens de gestão da rede 12
2.8 Visão global do plano de INM 13
2.9 Arquitectura geral do padrão 802.11 18
2.10 Constituição do BSSID numa IBSS segundo o protocolo 802.11 19
2.11 Camadas protocolares do 802.11 19
2.12 Modo infra-estrutura *vs* modo *ad hoc* 19
2.13 Modos de *Scanning/Probing* 20
2.14 Mecanismo CSMA/CA 21
2.15 Problema do nó escondido 22
2.16 Mecanismo CSMA/CA com RTS/CTS 22
2.17 Formato do cabeçalho do padrão 802.11 ao nível MAC 24
2.18 Constituição do campo *Frame Control* 24
2.19 Constituição do campo *Duration*/ID 27
2.20 Constituição do campo *Sequence Control* 27
2.21 Constituição do *Frame Body* para os diferentes pacotes de gestão 28
2.22 Estrutura de um *Information Element* 28

3.1 Arquitectura do modelo de comunicações segundo o conceito de INM 32
3.2 Interligação entre as várias funcionalidades e mecanismos implementados 33
3.3 Representação genérica do processo de *bootstrapping* 35
3.4 Etapas genéricas do processo de descoberta 36
3.5 Representação genérica do processo de eleição 36
3.6 *Standard Association versus Social-based Association* 37
3.7 Fases desde a perda de ligação até à nova associação 38
3.8 Conceito genérico do identificador de uma comunidade 38
3.9 União de comunidades 39
3.10 Separação de comunidades 39
3.11 Interpretação visual da FIN 41

3.12 Interpretação visual da FQN 41
3.13 Interpretação visual da CQE 42
3.14 Processo de sinalização envolvida nos vários mecanismos do *INM-Discovery* 46
3.15 Etapas genéricas da descoberta pelo *INM-Discovery* 47

4.1 Sub-camadas atravessadas ao nível 2 para transmissão e recepção de pacotes em redes sem fios 53
4.2 Relação entre a probabilidade de erro e a SNR para diferentes tipos de modelação . . 56
4.3 FSM implementada no nó ad hoc 57
4.4 Recepção de *Beacon* 59
4.5 Recepção de *Probe Request* 60
4.6 Recepção de *Probe Response* 60
4.7 Recepção de *Association Request* 61
4.8 Recepção de *Association Response* 61
4.9 Recepção de *Disassociation* 62
4.10 Estrutura genérica de um pacote 802.11 MAC em NS-3 62
4.11 Estrutura do *Community-Based Beacon IE* 62
4.12 Estrutura do *Community-Based Association Response IE* 63
4.13 Processos envolvidos na aquisição e manutenção do identificador da comunidade . . . 64
4.14 Estrutura da tabela *Partial View* de cada nó 65
4.15 Estrutura da tabela de *Known Nodes* de cada nó 65
4.16 Processo de eleição 66
4.17 Exemplo de um vector de nós conhecidos 68
4.18 Esquema geral da *testbed* 73

5.1 Evolução do número de comunidades finais 77
5.2 Evolução do tamanho das comunidades finais 77
5.3 Evolução do número total de associações provocadas pela métrica social 78
5.4 Evolução do número total de associações provocadas pela métrica social e pelo alcance limitado 79
5.5 Evolução do tempo para restabelecimento de associação 79
5.6 Evolução do tempo para estabelecimento da primeira associação em todos os nós . . . 80
5.7 Evolução do *overhead* de manutenção 81
5.8 Evolução do *overhead* de manutenção 82
5.9 Evolução do número de *Beacons* e *Associations* 83
5.10 Evolução do tamanho das comunidades finais em simulações com 100 nós 84
5.11 Evolução da média das FQN nas comunidades finais em simulações com 100 nós . . . 86
5.12 Evolução da melhor métrica social nas comunidades finais em simulações com 100 nós 87
5.13 Evolução dos componentes da melhor métrica social das comunidades finais em simulações com 100 nós 88
5.14 Evolução do tempo de convergência da descoberta 88
5.15 Evolução do *overhead* de mensagens na rede 89
5.16 Evolução do comportamento do protocolo *INM-Discovery* com o número de nós na rede 89

Lista de Tabelas

2.1 Requisitos funcionais do INM 13
2.2 Identificadores *Type* e *Subtype* 25
2.3 Significado dos *bits To DS* e *From DS* 26
2.4 Papel desempenhado pelos vários campos de endereços 27

3.1 Significado dos parâmetros da fórmula 3.3 40
3.2 Significado dos parâmetros da fórmula 3.4 41
3.3 Significado dos parâmetros da fórmula 3.5 42
3.4 Significado das variáveis da equação 3.7 48

4.1 Métodos envolvidos no processo de *bootstrapping* 58
4.2 Esquema de códigos para estimativa do tamanho da comunidade 67
4.3 Decisão local baseada no esquema de códigos da tabela 4.2 67
4.4 Significado das variáveis do comando para execução de uma simulação 69

5.1 Cenários usados para a obtenção de resultados 76

Lista de Acrónimos

AARF Adaptative Auto Rate Fallback

ACE Autonomic Communication Element

ACK Acknowledge

ALBA Autonomic Load Balancing Algorithm

AM Agent Manager

AMS Autonomic Management System

ANA Autonomic Network Architecture

ANEMA Autonomic Network Management Architecture

AON Application-Oriented Networking

AP Access Point

ARB Autonomic Resource Broker

ARF Auto Rate Fallback

ARP Address Resolution Protocol

ASA Autonomic Service Architecture

AUTOI Autonomic Internet

BER Bit Error Rate

BSS Basic Service Set

BSSID Basic Service Set Identifier

CA Collision Avoidance

CARA Collision-Aware Rate Adaptation

CASCADAS Component ware of Autonomic, Situation-aware Communication and Dynamically Adaptable Service

CCA Clear Channel Assessment

CD Collision Detection

CDP Cisco Discovery Protocol

CFP Contetion Free Period

CMIP Common Management Information Protocol

CP Contentio Period

CQE Community Quality of Nodes

CSMA Carrier Sense Multiple Access

CTS Clear to Send

CUE Channel Utilization Estimate

CW Contention Window

DCF Distributed Coordination Function

DDNS Dynamic Domain Name System

DiffServ Differentiated Services

DIFS Distributed Inter Frame Space

DNS Domain Name System

DS Domain System

DSSS Direct Sequence Spread Spectrum

DTN Delay Tolerant Networks

EDCAF Enhanced Distributed Channel Access Function

EIFS Extended Inter Frame Space

ESS Extendend Service Set

FCAPS Fault, Configuration, Accounting, Performance, Security

FHSS Frequency Hopping Spread Spectrum

FIN Friendship Indicator of Nodes

FOCALE Foundation Observation Comparison Action Learn Reason

FQN Friendship Quality of Nodes

FTP File Transfer Protocol

GPRS General Packet Radio Service

HSDPA High-Speed Downlink Packet Access

HTTP Hypertext Transfer Protocol

IBSS Independent Service Set

ID Identifier

IDM Information Dissemination Management

IE Information Element

IEEE Institute of Electrical and Electronics Engineers

INM In-Network Management

LAN Local Area Network

LSA Link State Advertisement

LSR Link State Routing

LTE Long Term Evolution

MAC Media Access Control

MIB Management Information Base

MPDU Media Access Control Protocol Data Unit

MPLS Multiprotocol Label Switching

NGN Next Generation Networks

NLE Network Leader Election

NMS Network Management System

OFDM Orthogonal Frequency Division Multiplexing

OSI Open Systems Interconnection

OSKMV Orchestration, Service Enablers, Knowledge, Management and Virtualization

OSPF Open Shortest Path First

P2P Peer-to-Peer

PCF Distributed Coordination Function

PHY Physical Layer

PIFS Point Inter Frame Space

PS-Poll Power Saver Poll

QoS Quality of Service

RBAR Receiver-Based AutoRate

RIP Routing Information Protocol

RTS Request to Send

RTT Round-Trip-Time

SIFS Short Inter Frame Space

SLA Service Level Agreement

SNMP Simple Network Management Protocol

SNR Signal-to-Noise Ratio

SOA Service-Oriented Architecture

SSID Service Set Identifier

STA Station

STP Spanning Tree Protocol

TMN Telecommunications Management Network

TTL Time-To-Live

UMTS Universal Mobile Telecommunications System

VoIP Voice over Internet Protocol

WEP Wired Equivalent Privacy

WLAN Wireless Local Area Network

Capítulo 1

Introdução

1.1 Motivação

Quando as dimensões das redes eram relativamente reduzidas, os protocolos de gestão tradicionalmente usados eram sobretudo o *Simple Network Management Protocol* (SNMP) [1] ou *Common Management Information Protocol* (CMIP) [2]. As suas características de centralização, aliadas ao crescimento exponencial de equipamentos de rede, introduzem problemas sérios em termos de escalabilidade. Como forma de ultrapassar esta limitação começaram a surgir alguns protocolos de gestão distribuída [3, 4], no entanto estes ainda apresentam algumas lacunas, nomeadamente em termos de complexidade exigida.

A crescente dimensão e dinamismo que se esperam das redes de próxima geração exigem protocolos de gestão que respondam às necessidades das mesmas, quer em termos de automatização, optimização e eficiência de processos, como também ao nível da interacção entre as diversas entidades que compõem a rede.

As *Next Generation Networks* (NGNs) trazem novos desafios que necessitam de diferentes soluções relativamente às redes tradicionais, por exemplo, em termos da mobilidade dos equipamentos, topologia dinâmica de rede, conectividade intermitente, heterogeneidade de terminais e de tecnologias de acesso [5, 6]. Todos estes factores implicam uma constante reconfiguração da rede, o que por si só exige protocolos de encaminhamento com requisitos muito específicos. De forma a ultrapassar estes novos desafios, novas arquitecturas, plataformas, mecanismos e funcionalidades devem ser elaboradas usando conceitos do estado da arte, como a computação e gestão autonómica [7, 8, 9, 10, 11, 12].

A presente Dissertação insere-se no âmbito do paradigma de *In-Network Management* (INM) [13] cuja ideia principal é inserir nas várias entidades que compõem a rede, capacidades e funcionalidades de gestão de forma a que esta adquira capacidade autonómica, tal como referido em [14]. Para que tal seja possível, diversos mecanismos têm de ser implementados, nomeadamente o *bootstrapping* autonómico de cada entidade, a comunicação cooperativa, a tomada de decisões e a disseminação das mesmas pela rede. Pretende-se assim alcançar uma gestão distribuída da rede, isto é, que não dependa de entidades agregadoras externas/servidores. Isto implica não só um elevado nível de automatismo da rede, como de sincronismo e actualização da informação entre as várias entidades, contribuindo para um possível aumento do *overhead*.

O maior foco da presente Dissertação são os mecanismos base da arquitectura global representada

na figura 3.1. Nestes insere-se o *bootstrapping* autonómico (i.e. o nó da rede ter capacidade de se inicializar e configurar de forma autonómica), mecanismos para descoberta de nós na rede e identificação das melhores entidades de cada domínio para a eleição. Face a isto, pretendem-se desenvolver mecanismos de comunicação distribuída entre elementos da rede, através dos quais se fará a troca de informações entre nós, de forma a alcançar a colaboração entre os mesmos.

1.2 Objectivos

A presente Dissertação visa desenvolver mecanismos de comunicação que permitam construir, de forma *bottom-up*[1], uma plataforma de gestão distribuída da rede, de forma a responder aos desafios impostos pelas redes de próxima geração. Estes mecanismos são a chave para uma gestão eficiente, escalável e robusta, ou seja, deve ter baixo *overhead* de mensagens, responder bem em cenários de larga escala e possuir tolerância a falhas em ambientes dinâmicos.

De forma sucinta, os objectivos desta Dissertação são:

- Estudar o estado da arte dos vários conceitos envolvidos, nomeadamente a gestão autonómica e distribuída da rede;
- Implementar os modelos necessários à simulação de redes *ad hoc* sem fios;
- Identificar as alterações necessárias a realizar ao protocolo de controlo existente no simulador para redes sem fios;
- Implementar funcionalidades e mecanismos de comunicação necessários à gestão autonómica e distribuída, nomeadamente:
  - *Bootstrapping* dos vários nós (redes com fios e sem fios);
  - Descoberta de outros nós na rede (redes com fios e sem fios);
  - Troca de informações (redes com fios e sem fios);
  - Associação de nós com base em parâmetros dinâmicos (redes sem fios);
  - Agregação dos nós em comunidades (redes sem fios);
  - Disseminação de informações relevantes (redes com fios e sem fios);
  - Recolha de informações necessárias para a eleição dos líderes (redes com fios e sem fios).
- Desenvolver cenários de teste às funcionalidades e mecanismos implementados;
- Recolher e analisar os resultados obtidos, bem como elaborar análises comparativas com outros modelos existentes.

[1] Neste tipo de abordagens os elementos e mecanismos base são implementados e especificados em detalhe. Apenas numa fase posterior se faz a junção destes formando subsistemas que se integrarão num sistema de maior dimensão e complexidade.

1.3 Contribuições

As soluções implementadas no âmbito desta Dissertação inserem-se no paradigma INM, estando nele incluídas como mecanismos base sobre os quais assenta toda a restante arquitectura. Estas permitem o desenvolvimento de processos de comunicação cooperativa para a gestão distribuída da rede através da implementação de mecanismos como *bootstrapping*, descoberta e eleição. Para tal são usadas soluções inovadoras, nomeadamente o conceito de métrica social como forma de avaliação quantitativa do nó e da sua comunidade nas redes sem fios e a técnica *Hide & Seek* [15] no protocolo de descoberta das redes com fios [16].

Como resultado do cumprimento dos objectivos previamente mencionados, a presente Dissertação fornecerá uma avaliação concreta do desempenho de parâmetros relacionados com os mecanismos implementados. Em termos de simulação de redes sem fios, a avaliação incidirá mais em características ao nível da camada Media Access Control (MAC); na *testbed* virtual de redes com fios, a avaliação irá reflectir o desempenho ao nível da camada IP.

Este trabalho deu já origem à submissão e aceitação de um artigo científico com a avaliação dos mecanismos implementados ao nível das redes com fios. Este foi aceite recentemente na MONAMI, 3rd International ICST Conference on Mobile Networks & Management 2011, intitulado "Nodes Discovery in the In-Network Management Communication Framework". A publicação dos mecanismos de comunicação para redes sem fios encontram-se em preparação.

1.4 Organização

A presente Dissertação está organizada da seguinte forma:

- Capítulo 2 descreve o estado da arte dos tópicos abrangidos por esta Dissertação, nomeadamente ao nível da gestão autonómica e distribuída da rede, do paradigma de INM, bem como dos mecanismos e protocolos que servem de base à comunicação entre entidades e domínios;
- Capítulo 3 introduz a solução desenvolvida de um ponto de vista de alto nível, indicando os objectivos a atingir com os mecanismos implementados e a descrição dos conceitos fundamentais dos mesmos. São também aqui abordados os principais desafios inerentes à Dissertação.
- Capítulo 4 aborda a implementação das soluções propostas. Ao nível das redes sem fios, começa-se por introduzir o simulador utilizado para o efeito, nomeadamente ao nível do protocolo de gestão usado e, em seguida, são apresentados os detalhes dos mecanismos e funcionalidades implementadas. Nas redes com fios é feita uma descrição aprofundada dos módulos funcionais desenvolvidos na *testbed* virtual;
- Capítulo 5 apresenta os resultados e a discussão dos mesmos, para os cenários de simulação sem fios e para a *testbed* com fios. É também apresentada uma conclusão da avaliação geral que os resultados reflectem;
- Capítulo 6 apresenta as conclusões gerais das soluções desenvolvidas. São também propostas as principais linhas de investigação que o trabalho pode seguir, tanto ao nível de possíveis melhoramentos dos mecanismos, como de novas funcionalidades a implementar.

Capítulo 2

Estado de Arte

2.1 Introdução

Neste capítulo é apresentado um conjunto de conceitos de carácter mais teórico, bem como referências para trabalhos relacionados com os vários temas abordados.

Na secção 2.2 é apresentada uma descrição dos vários tipos de arquitecturas de gestão de redes.

Na secção 2.3 é abordado o conceito de In-Network Management, no qual a presente Dissertação se insere.

Em 2.4 são referenciadas e analisadas, de forma breve, várias abordagens já existentes ao nível da comunicação entre nós, ao nível do *bootstrapping*, descoberta, disseminação de informação e eleição.

Em 2.5 é apresentado em detalhe o protocolo 802.11 ao nível MAC e físico, uma vez que, este é a base em que assentam as redes sem fios simuladas.

Em 2.6 é realizada uma breve descrição de três protocolos de descoberta em redes com fios, que posteriormente serão comparados com a nossa proposta.

Por fim, em 2.7 é apresentado um breve sumário relativo ao capítulo.

2.2 Arquitecturas de gestão de redes

2.2.1 Abordagens tradicionais

A gestão de redes visa o controlo de dispositivos de telecomunicações cuja monitorização é feita através de uma estrutura de recursos, sendo que esta pode estar organizada de forma centralizada ou distribuída. Esta gestão é, por isso, essencial ao correcto funcionamento da mesma, através da coordenação entre os diversos dispositivos, bem como para garantir a qualidade dos serviços prestados por esta.

Usualmente, o ciclo de gestão é iniciado pela etapa de recolha de informação. Através das informações recolhidas é feita uma análise sobre o estado actual da rede. Em seguida, pode ser tomado um conjunto de acções, dependendo da análise previamente extraída da rede. As tarefas de gestão funcionais estão agrupadas e descritas em [17].

O *Network Management System* (NMS) é geralmente composto pelo gestor, pelo agente, pelo *Management Information Base* (MIB) e pelo protocolo de gestão (figura 2.1). O gestor tem como

função controlar os agentes através do protocolo de gestão, sendo estes os processos que fazem a recolha de informações da rede, armazenando-as em repositórios de informação (MIBs).

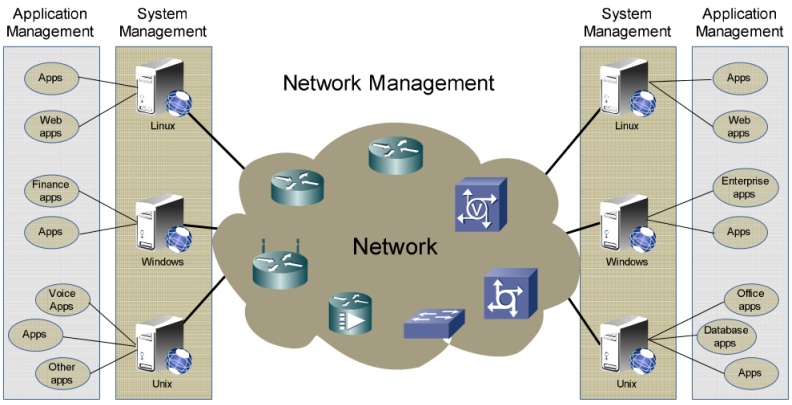


Figura 2.1: Sistema de gestão clássica da rede [18]

Ao nível das principais funcionalidades de gestão de redes usados pelo protocolo *Telecommunications Management Network* (TMN) [19, 20] encontram-se o *Fault, Conguration, Accounting, Performance, Security* (FCAPS). Os modelos de gestão utilizados são o CMIP e SNMP.

Em [21] é proposta uma classificação para as diferentes abordagens à gestão de redes: centralizada, fracamente e fortemente distribuída, e colaborativa. Os aspectos mais relevantes de cada uma são abordados em seguida.

2.2.1.1 Abordagem centralizada

A característica fundamental deste tipo de abordagem é a existência de um gestor central que agrega e controla todas os agentes, como é ilustrado pela figura 2.2. Este tipo de gestão centralizada tem como principais limitações:

- a falta de escalabilidade para cenários de larga escala, onde ocorrerá sobrecarga do gestor central, sendo o desempenho global da rede degradado;

- a inexistência de comunicação entre os diversos agentes, cabendo a estes apenas a recolha de informação, sendo a análise executada em exclusivo pelo gestor;

- o ponto de falha do sistema estar concentrado num único ponto, uma vez que o possível colapso da rede se encontra no gestor central. A inclusão de redundância é uma possível solução mas implicará sempre o aumento dos gastos financeiros.

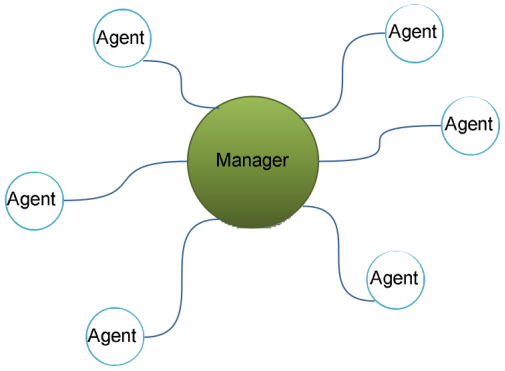


Figura 2.2: Esquema ilustrativo da gestão centralizada [22]

2.2.1.2 Abordagem distribuída

Neste tipo de gestão distinguem-se dois tipos de abordagens: fracamente (figura 2.3) e fortemente (figura 2.4) distribuídas. Na primeira destaca-se a introdução de elementos intermédios de processamento, designados *Agent Managers* (AMs). O objectivo destes é conseguir executar um pré-processamento de forma a libertar alguma da carga do gestor. No entanto, continua sem existir comunicação entre agentes e a comunicação entre AMs é também inexistente. Devido a estes factores percebe-se que todas as comunicações continuam a ter de passar obrigatoriamente pelo gestor, já que não ocorre colaboração entre entidades. É importante também realçar que os pontos de falha continuam a existir: o gestor é ponto de falha global da rede e os vários AMs são os pontos de falha de blocos da rede. Contornar este problema através de redundância pode ser muito dispendioso. Por fim, é importante referir a falta de robustez nas ligações críticas entre AMs e gestor, tornando a abordagem impraticável em ambientes dinâmicos.

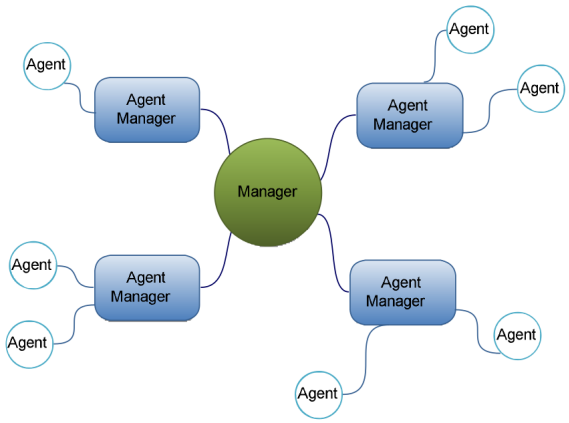


Figura 2.3: Esquema ilustrativo da gestão fracamente distribuída [22]

No outro tipo de abordagem (fortemente distribuídas) passa a existir colaboração entre AMs, ou seja, deixam de fazer apenas recolha de informação dos agentes. Desta forma, é possível implementar mecanismos de disseminação de decisões e de gestão entre AMs. O papel do gestor passa a ser o de coordenador das tarefas da rede, apesar da distribuição destas poder ser feita através da comunicação entre AMs.

Figura 2.4: Esquema ilustrativo da gestão fortemente distribuída [22]

O facto de existir colaboração entre AMs obriga a que haja um aumento do *overhead*. No entanto não existe comunicação entre agentes e estes continuam a estar dependentes de um único ponto (o AM correspondente). A estas, acresce ainda a desvantagem de que os AMs estão subordinados ao gestor, ou seja, apesar da comunicação entre este tipo de entidades, não há tomadas de decisão por parte destas.

2.2.1.3 Abordagem cooperativa

Como é possível observar pela figura 2.5 o gestor central deixa de existir neste tipo de abordagem e passa a existir cooperação entre as diferentes entidades, não estanto esta limitada a um único ponto da rede. Devido a este facto, esta abordagem é passível de ser implementada em cenários muito dinâmicos. Assim, através deste tipo de características, as entidades passam a ter um nível elevado de autonomia e maiores índices de inteligência. Porém, mais inteligência implica maior complexidade interna. Pode também referir-se o facto de que o *overhead* de controlo que é necessário acrescentar para implementar os mecanismos de colaboração pode degradar o desempenho da rede em cenários de larga escala. Por último, questões relacionadas com a segurança, devido à inexistência de pontos de gestão central, têm de ser tidas em consideração.

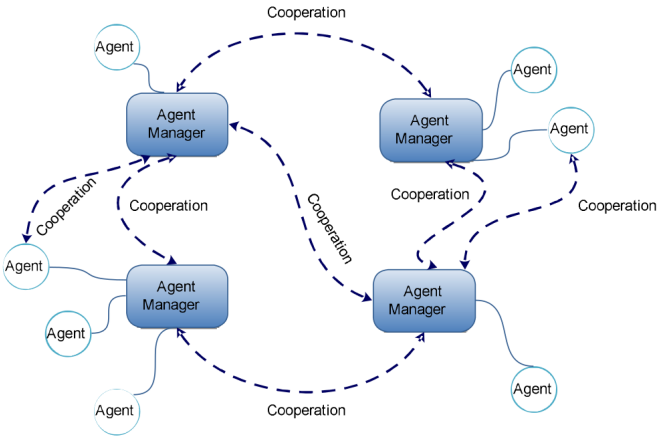


Figura 2.5: Esquema ilustrativo da gestão cooperativa [22]

2.2.2 Abordagens em redes de próxima geração

Os avanços nas telecomunicações, tanto ao nível do número de dispositivos com ligação à rede, como nas diversas tecnologias de acesso e nos diferentes serviços (voz, dados e vídeo), levantam problemas às actuais arquitecturas de rede. Segundo [23], as NGNs são essencialmente redes IP que permitem a qualquer tipo de cliente receber os diferentes tipos de serviços através da mesma rede. Isto implica que a camada de serviços seja independente das camadas *Open Systems Interconnection* (OSI) inferiores e que o acesso seja possível tanto por protocolos de redes sem fios (*General Packet Radio Service* (GPRS), *Universal Mobile Telecommunications System* (UMTS), *High-Speed Downlink Packet Access* (HSDPA), *Long Term Evolution* (LTE), etc) como de redes com fios (Ethernet, fibra, etc).

As NGNs deverão introduzir maior autonomismo na rede, libertando o gestor para outras tarefas de mais alto nível. Espera-se desta forma conseguir resultados mais robustos e uma rede adaptada à dinâmica e exigências de *Quality of Service* (QoS) dos utilizadores, sem que com isso a segurança seja comprometida.

Algumas das técnicas apontadas para contornar este problema são a computação e a gestão autonómica. Através destas esperam-se conseguir solucionar os desafios que as NGNs enfrentam: agregar a crescente diversidade de serviços e tecnologias, respondendo em simultâneo ao aumento do número e da mobilidade dos dispositivos.

Em [24] foram descritas as quatro características principais que estas técnicas devem possuir: Configuração, Recuperação, Optimização e Protecção (CHOP), conhecidas como propriedades *Self-X*. Além destas, foi abordada em [25] outra capacidade dos sistemas de computação autonómica, que é a de cada entidade conseguir ser mais autoconsciente, ou seja, ter conhecimento detalhado acerca do meio envolvente.

A computação autonómica consiste basicamente num ciclo de controlo pelo qual as entidades de rede se regem. Este pode ser visto como um conjunto de etapas que passam desde a recolha de informação, à análise da mesma, passando pela decisão de acordo com os resultados obtidos e regras pré-determinadas e, por fim, a execução dessas mesmas decisões e a sua disseminação pela rede (figura 2.6).

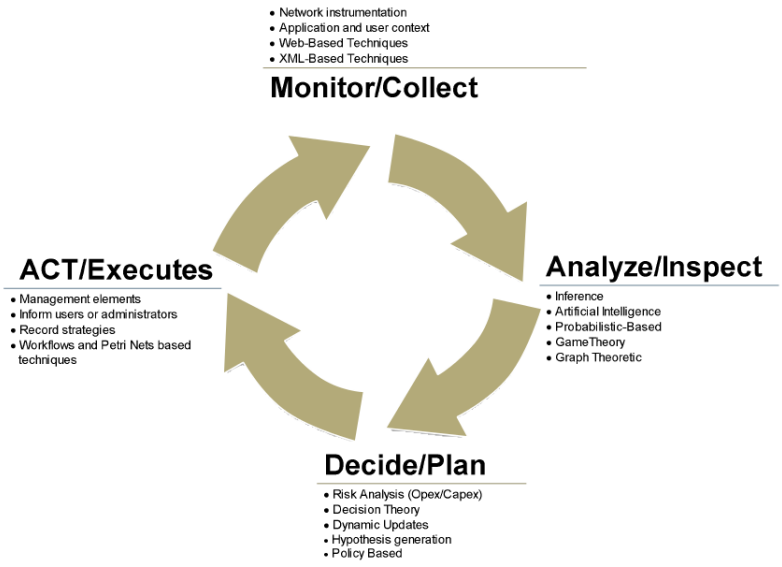


Figura 2.6: Ciclo de controlo autonómico [22]

Em termos de gestão autonómica, esta é vista como uma alternativa à gestão tradicional de redes, que vem acumulando dificuldades nos últimos tempos [26, 8]. Para as ultrapassar, novas abordagens têm sido propostas, envolvendo arquitecturas, estratégias e tecnologias inovadoras [27, 28, 29].

Em [30] é proposto um *framework* para gestão de recursos e serviços de internet, designado *Autonomic Service Architecture* (ASA). A sua arquitectura permite a gestão autonómica de recursos respeitando políticas de *Service Level Agreement* (SLA). As entidades responsáveis pelos serviços de gestão são designadas *Autonomic Resource Brokers* (ARBs) e estão organizadas numa estrutura hierárquica, interagindo entre elas e com recursos virtuais. Os autores demonstram a possibilidade de aplicar este *framework* na gestão de *Multiprotocol Label Switching - Differentiated Services* (MPLS-DiffServ) através de mecanismos de ajuste da largura de banda de acordo com a carga, e utilização eficiente de recursos respeitando SLAs. Embora possua uma arquitectura genérica, a implementação de novas funcionalidades não é simples devido à complexidade da organização hierárquica e da arquitectura interna dos ARBs.

Em [31] é apresentada a arquitectura *Autonomic Network Architecture* (ANA), que possui suporte ao nível da aplicação (*Hypertext Transfer Protoco*l (HTTP), *File Transfer Protocol* (FTP), *Peer-to-Peer* (P2P), *Voice over Internet Protocol* (VoIP), *Domain Name System* (DNS), etc), nós móveis, multidomínios e mecanismos de registo, autenticação e comunicação entre nós. Através desta é possível uma gestão autonómica da rede, adaptada a condições dinâmicas e reorganização do tráfego segundo requisitos do utilizador. No entanto, apresenta problemas semelhantes aos indicados na análise da abordagem referenciada em [30].

Em [32] é descrito o projecto *Component ware of Autonomic, Situation-aware Communication and Dynamically Adaptable Service* (CASCADAS) e as entidades que o compõem. Estas são designadas de *Autonomic Communication Element*s (ACEs) e integram todas as capacidades de auto-organização, sendo responsáveis pela gestão da rede. Em termos de camadas, a arquitectura está desenhada do seguinte modo: ao nível da aplicação existe uma distinção clara entre o desenvolvimento e a integração de componentes para aplicações autónomas; ao nível da rede está prevista a integração de diversos controladores e a partilha de recursos de rede é também abordada. Contudo, a abordagem apresenta limitações em termos de escalabilidade e de intolerância a falhas devido ao facto dos ACEs serem pontos centralizadores de serviços.

Em [33] é proposta a arquitectura Haggle para redes autonómicas com suporte a ligações intermitentes. Em relação ao modelo do nó, existe a preocupação em implementar as diversas camadas do modelo OSI, onde se destaca o suporte ao nível da camada física, da transmissão em redes Ethernet e sem fios segundo o padrão 802.11 e Bluetooth. As limitações encontradas relacionam-se sobretudo com a falta de suporte a transmissões extremo-a-extremo e requisitos de QoS.

Em [34] é abordado o projecto designado de *Foundation Observation Comparison Action Learn Reason* (FOCALE), no qual se insere uma arquitectura adaptável a mudanças do meio envolvente, regras e requisitos impostos pelo utilizador. Na base da sua constituição estão elementos que podem incorporar todas as funcionalidades de gestão autonómica e que colocam a necessidade de intervenção directa apenas ao nível da definição de objectivos. As características diferenciadoras deste projecto são: a capacidade de lidar com terminais com diferentes sistemas operativos; a comunicação através protocolos distintos e em ambientes dinâmicos. O sistema é, por isso, bastante complexo e apresenta dificuldades em recuperar caso ocorram falhas ao nível da gestão, nomeadamente quando surgem situações imprevistas no meio. Isto implica que haja degradação em termos de segurança e QoS.

Em [35] é apresentado um modelo de arquitectura para sistemas de gestão distribuída, designado *Autonomic Internet* (AUTOI). A sua principal característica é a existência de cinco planos de abstracção (planos *Orchestration, Service Enablers, Knowledge, Management and Virtualization* (OSKMV)), que não são mais que funções distribuídas para controlo da infra-estrutura de rede. O uso do AUTOI não é adequado a redes de grande escala devido ao elevado nível de sincronização requerido pelos elementos constituintes da arquitectura (*Autonomic Management System* - AMS)).

Em [36] é realizado um estudo usando métodos computacionais autonómicos para a gestão de NGNs. Demonstrou-se que o uso destes métodos permite optimizar as comunicações, ao efectuar um controlo da rede sem intervenção directa, evitando assim alguma da complexidade dos sistemas actuais. São consideradas duas perspectivas, uma que considera os equipamentos, outra em que é feita uma abstracção dos mesmos. Porém, não são propostos cenários concretos para avaliação dos mecanismos.

Em [9] é apresentado um protótipo de um *framework* para gestão autonómica de serviços em NGNs, que integra mecanismos como *Service-Oriented Architecture* (SOA), *Application-Oriented Networking* (AON) e computação autonómica. Neste protótipo está já prevista uma aplicação de gestão que permite a definição automática de requisitos de QoS extremo-a-extremo para serviços padrão de forma distribuída, e a sua gestão ser realizada autonomamente de acordo com políticas de SLA. No entanto, as restrições impostas por SLAs ao nível de QoS tornam a tarefa de gestão autonómica do *framework* bastante complexa, tornando-se difícil encontrar um ponto de equilíbrio entre a solução adoptada e as regras impostas.

Em [37] é proposta a *Autonomic Network Management Architecture* (ANEMA), cujo objectivo principal é a implementação de mecanismos que permitam a gestão autonómica de redes baseadas em IP. Como forma de o demonstrar foi desenvolvido um simulador do sistema com diferentes cenários e também uma *testbed* de pequena escala para comprovar a solução apresentada. Contudo, não é demonstrado o impacto das decisões locais na rede global, bem como o modo da sua disseminação entre as entidades. Seria também relevante elaborar um conjunto de testes reais em maior escala.

Em [27] é proposto o *Autonomic Load Balancing Algorithm* (ALBA) que pretende aumentar a utilização da capacidade de uma *Wireless Local Area Network* (WLAN), garantindo simultaneamente QoS em aplicações VoIP. A sua arquitectura permite ainda lidar com mudanças da topologia de rede, uma vez que os *Access-Points* (APs) estão dotados de funcionalidades autonómicas. Através de simulação é demonstrado que estas funcionalidades resultam num comportamento de regulação consistente da carga para fora das zonas sobrecarregadas. Esta avaliação é feita através da métrica *Channel Utilization Estimate* (CUE) em vez da largura de banda. Seria importante a integração de outros critérios para fazer o balanceamento de carga e analisar o impacto de cada um, bem como a implementação de mecanismos para a disseminação de requisitos de QoS aos melhores APs vizinhos. No entanto, as limitações em termos de escalabilidade estarão sempre presentes, uma vez que as capacidades de gestão autonómica estão implementadas em APs.

2.3 *In-Network Management*

O paradigma de INM [13] foi apresentado no projecto europeu 4WARD [38] e pretende implementar um novo *design* de arquitecturas de rede que responda às exigências das NGNs. Este novo *design* incorporará arquitecturas distribuídas, através do qual a gestão da rede será feita. O principal objectivo será ter as entidades e os serviços de rede muito próximos, ou seja, que as capacidades de gestão estejam incorporadas nos elementos de rede.

Com este paradigma espera-se atingir um maior nível de automatização na medida em que, cada uma das entidades será capaz de tomar decisões por si só, tendo como base o conhecimento partilhado entre os diversos elementos que compõem a rede. A colaboração será também útil na adaptação da rede a mudanças que ocorram nesta (falhas, mobilidade, entrada e saída de nós, etc) bem como em termos de escalabilidade da mesma.

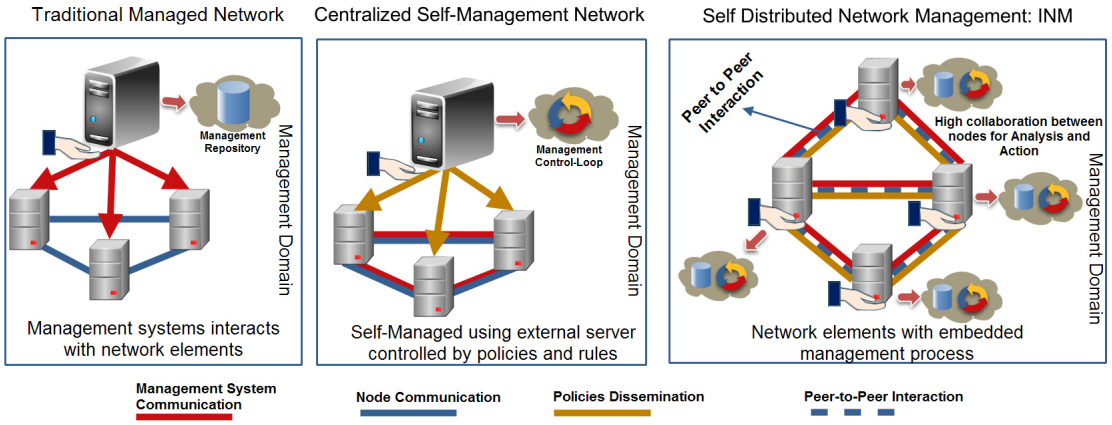


Figura 2.7: Comparação entre as várias abordagens de gestão da rede [38]

De acordo com a figura 2.7, na abordagem tradicional o administrador controla toda a gestão de decisões da rede, estando os servidores fora do plano da rede onde é executado todo o processamento.

Já na abordagem autonómica, todo o controlo e decisões estão inseridas num ciclo de gestão (figura 2.6). Existe comunicação entre entidades para a gestão da rede, no entanto, as regras são determinadas externamente por servidores específicos para o efeito, introduzindo limitações ao nível da escalabilidade da arquitectura.

Por último, segundo o paradigma de INM, existem interacções P2P entre entidades com o objectivo de partilhar conhecimento para atingir uma gestão descentralizada da rede, isto é, sem haver necessidade de recorrer a servidores externos para executar decisões. Desta forma, espera-se conseguir uma arquitectura escalável, robusta, auto-organizada e totalmente distribuída.

Em termos de visão global, a figura 2.8 ilustra a organização do plano de gestão segundo o paradigma de INM. Verifica-se que este se encontra inserido na própria rede e que as entidades deste plano apresentam um elevado nível de interacção, sendo executado pelos processos de gestão incorporados em cada equipamento.

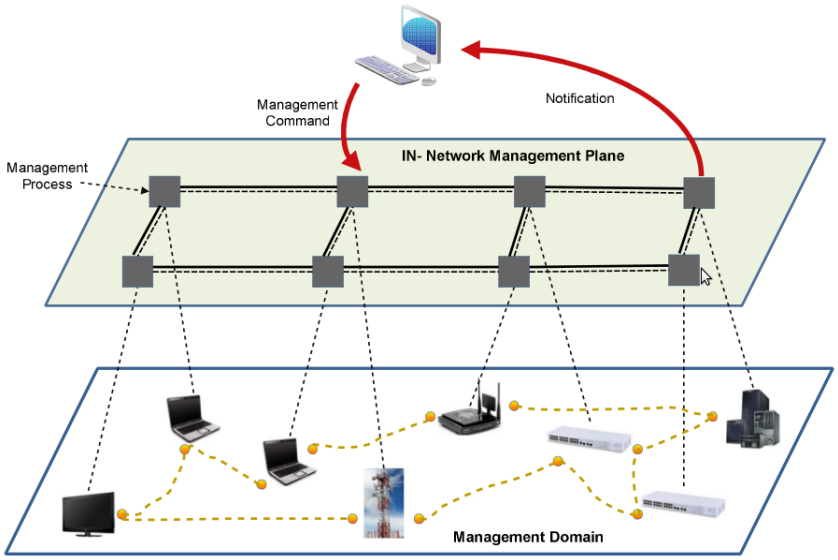


Figura 2.8: Visão global do plano de INM [39]

2.3.1 Requisitos funcionais

Foram estabelecidos alguns requisitos funcionais para os objectivos a que o paradigma de INM se propõe a atingir. Na tabela 2.1 são identificados e descritos alguns desses requisitos.

Requisito	Descrição
Heterogeneidade	Deve haver suporte para heterogeneidade em diferentes níveis (tecnologias de acesso, terminais, etc)
Interoperabilidade	A comunicação, possibilidade de recolha de informação e outras operações entre dispositivos deve ser possível no INM
Suporte em tempo-real	Operações de gestão da rede não devem estar limitadas a janelas temporais definidas, mas sim ser possível fazê-lo em tempo-real
Impacto/Pegada	Devido aos recursos limitados dos equipamentos móveis é necessário que o impacto a este nível seja bastante reduzido.
Previsão	A monitorização em tempo-real deve permitir determinar métricas que permitam antecipar situações de falhas, anomalias, etc.
Integração	De modo a garantir escalabilidade, as funções de gestão devem estar inseridas nas próprias entidades de rede.
Adaptabilidade	Além de detectar, o sistema deve também ser capaz de se adaptar perante as diversas situações com que se pode deparar.
Segurança	Mecanismos como autenticação e identificação, que garantam a segurança da rede devem ser assegurados.
Aprendizagem	Através da colaboração entre as várias entidades, o sistema no seu todo deve apresentar a capacidade de aprendizagem.
Escalabilidade	O aumento do número de equipamentos não deve inserir complexidade nem exigir mais recursos à rede.
Robustez	A rede deve ser tolerante a falhas, ou seja, não devem existir pontos críticos que comprometam todo o sistema.
Interactividade	Deve ser possível ao operador de rede intervir nesta em diferentes níveis de abstracção.
Extensível	Deve ser capaz de integrar novas soluções proprietárias e de código aberto, permitindo a evolução de todo o sistema.

Tabela 2.1: Requisitos funcionais do INM [40]

2.4 Comunicação entre nós e domínios

É essencial que exista comunicação entre os vários elementos que compõem a rede de modo a atingir um elevado nível de conhecimento que assegure uma gestão distribuída. No entanto, esta comunicação tem de ser minimizada na rede para diminuir o congestionamento da mesma devido aos processos de controlo. A partição da rede em domínios auto-organizados, entre os quais as entidades se podem mover livremente, de modo a apenas ser necessário que estas possuam visões parciais da estrutura global da rede, é outro dos requisitos deste tipo de arquitectura.

Nas secções seguintes são apresentadas abordagens a mecanismos de *bootstrapping*, descoberta e eleição que permitirão optimizar o processo de comunicação entre os nós.

2.4.1 Abordagens de *bootstrapping*

Segundo [41, 42, 43], um mecanismo de *bootstrapping* permite detectar e iniciar o contacto com os primeiros nós da rede, ou seja, através deste, a entidade deve ser inicializada e configurada de modo a estar preparada para iniciar a sua actividade na rede. Ao nível das redes P2P, são apresentadas abordagens para arquitecturas centralizadas em [44, 45, 46, 47], para arquitecturas híbridas em [43] e para arquitecturas distribuídas em [48].

O grande desafio deste mecanismo é que a sua implementação não deve comprometer a robustez, o automatismo e a escalabilidade do sistema, portanto, o *overhead* de mensagens necessário ao *warm-up* da rede deve ser baixo.

Em [43] é apresentado um mecanismo de *bootstrapping* para redes *ad hoc* móveis completamente distribuído. Este mecanismo cumpre, portanto, o requisito de ser escalável, uma vez que não depende de qualquer ponto central coordenador. Neste método são usados P2P *multicast join queries* e *responses*, bem como o armazenamento dos resultados de todos os nós. No entanto, o *overhead* de mensagens apresentado é um problema a ter em consideração.

Em [49] é abordada a necessidade de implementar um mecanismo de *bootstrapping* se for integrado um gestor autonómico na rede. Esta solução pode resolver alguns dos problemas de consenso existentes em sistemas distribuídos e possivelmente diminuir o *overhead* necessário à gestão da informação. Contudo, a escalabilidade fica comprometida, uma vez que, em redes de larga escala, o gestor autonómico ficará sobrecarregado. Também a segurança do sistema é colocada em causa, pois uma falha no gestor autonómico afecta a rede no seu todo.

Em [50] é proposto um mecanismo de *bootstrapping* baseado em *Dynamic Domain Name System* (DDNS). Através desta abordagem, são detectadas comunicações P2P e os nós envolvidos são automáticamente inseridos na rede. O objectivo do *bootstrapping* é encontrar um membro já existente no sistema, para que a rede P2P não funcione de forma isolada e seja possível melhorar o seu desempenho global.

Em [51] os autores propõem um modelo de *bootstrapping* auto-organizado para redes sem-fios direccionais. Os seus requisitos são um algoritmo para organização dos nós de forma *bottom-up* formando uma *spanning-tree,* um algoritmo de descoberta de recursos para a disseminação eficiente de informações de conectividade local, e ainda um protocolo de sincronismo que garanta a conectividade a partir de interacções P2P. Os autores identificam como principais desafios a integração destes algoritmos, a escolha de protocolos para determinação do estado das ligações, a troca de informações entre nós de forma coordenada, o sincronismo e coerência da informação local.

2.4.2 Abordagens de descoberta

A descoberta é o processo através do qual um nó passa a conhecer o meio que o envolve, ou seja, indentifica que entidades se encontram na rede e informações relacionadas com estas.

Tanto nas redes de sensores [52, 53, 54, 55] como nas *ad hoc* [56, 57, 58], foram estudados mecanismos de descoberta, muito devido à dinâmica destes tipos de redes. Ao nível MAC [59] e da ligação [60] foram já propostas abordagens para a descoberta de nós e da topologia de rede a 1-salto. Também ao nível IP foi apresentado em [61] um processo de descoberta que permite determinar a topologia lógica da rede através da informação previamente recolhida. Existem ainda propostas de abordagens híbridas [62] e assíncronas [63, 64]. As propostas referenciadas usam, em geral, mensagens *broadcast* com raio de alcance limitado, e através da resposta a estas obtêm informação sobre o meio envolvente, ou seja, dos vizinhos.

Em maior pormenor, os autores de [59] propõem um processo de descoberta para redes sem fios recorrendo a *Beacons*. O mecanismo funciona através da recepção ao nível MAC de *Beacons* modificados vindos dos nós vizinhos. Após a recolha de informação, é escolhida a melhor associação de acordo com critérios estabelecidos. A conclusão é que, numa rede segundo o padrão 802.11b, ocorre uma redução do *overhead* imposto pelo protocolo de encaminhamento ao nível IP. No entanto, a solução apresentada revela limitações ao nível da escalabilidade, uma vez que recorre ao uso de nós *forwarding* da informação.

Em [63] o problema da descoberta em redes sem fios é abordado tendo em conta o nível físico e do meio de acesso. Para tal é proposto o protocolo *gossip-based* T-Man que cria a topologia de rede de forma distribuída, rápida e robusta, exigindo porém um elevado número de mensagens para a sincronização da informação sobre a topologia da rede. O mecanismo permite ainda aos nós ter uma visão global ou parcial da rede, o que é vantajoso devido aos recursos limitados dos dispositivos sem fios. Por outro lado, existe a necessidade de uma sincronização global da informação da rede, o que torna a solução pouco eficaz em cenários reais.

Em [65] é apresentado o protocolo *Spanning Tree Protocol* (STP) [66] aplicado a redes sem fios. Através deste protocolo os nós sem fios e móveis de uma rede *ad hoc* estabelecem e mantêm ligações, sendo as informações armazenadas em tabelas locais. Estas são usadas posteriormente para estabelecer as melhores rotas entre nós; no entanto, o critério para o custo é o número de saltos que é ineficaz quando as ligações se encontram congestionadas. Além disso, não existe critério para a associação ao nó vizinho que, como já foi referido em [59], irá ter impacto ao nível do encaminhamento na camada IP.

Em [67] é abordado o problema da descoberta *ad hoc* em cenários sem fios estáticos. Para tal é realizada uma comparação entre a descoberta directa e usando algoritmos *gossip-based*. Em ambos é necessário ocorrer, pelo menos, a recepção bem sucedida de uma mensagem vinda de cada um dos seus vizinhos, para assim o receptor passar a conhecê-los. Esta limitação implica a degradação do desempenho global da rede quando a estimativa do número de nós se afasta do valor real.

Em [68] é proposta uma abordagem inspirada na comunicação das colónias de formigas (através de feromonas), aplicando-a na troca de informações entre nós da rede. Nas soluções inspiradas na biologia é necessário conhecer bem o modelo real, nomeadamente ao nível das interacções entre indivíduos. Por isso, os métodos usados para modelar estes comportamentos devem ser bastante específicos, impossibilitanto por vezes a sua integração noutras soluções.

Ao nível das redes com fios, os protocolos de encaminhamento (e.g. *Routing Information Protocol*

(RIP), *Open Shortest Path First* (OSPF), etc) integram mecanismos de descoberta de forma a criar as tabelas de adjacência. O OSPF [69] é um protocolo de encaminhamento adaptativo que usa pacotes *Link State Advertisement* (LSA) para descoberta de nós vizinhos, sendo enviada a topologia de rede localmente conhecida através destes, e propagada a todos os nós da mesma área. O valor de intervalo por defeito é 5s, porém, o tempo de convergência é a principal desvantagem em cenários de larga escala, podendo ainda ser melhorados outros aspectos [70]. Também a sua complexidade exige um planeamento adequado, tornando a tarefa do gestor de rede mais complexa em redes de grandes dimensões.

Também o *Cisco Discovery Protocol* (CDP) [71] é usado em redes com fios, apresentando informação sobre as interfaces dos vários equipamentos de rede onde o protocolo esteja em execução. No entanto, o *software* é proprietário, impedindo a implementação de novas funcionalidades ou modificações às existentes, encontrando-se restrito a equipamentos Cisco. O mecanismo de descoberta é realizado através do envio de pacotes CDP *Hello* em intervalos fixos (e.g. o valor por defeito é 60s), sendo o tempo necessário para a convergência uma das suas desvantagens.

Por último, o protocolo Fing [72] desenvolvido pela Overlook apresenta um *overhead* elevado de mensagens para efectuar o processo de descoberta. Este resultado deve-se ao facto do mecanismo se basear no *Address Resolution Protocol* (ARP), ou seja, no envio de ARP-Requests para todos os endereços IP da sua rede, aguardando resposta dos nós existentes (ARP-Reply).

2.4.3 Disseminação de informação

A disseminação de conhecimento é uma característica essencial à alimentação de vários mecanismos (descoberta, eleição, encaminhamento, etc). A disseminação de informação pode ser feita usando diversas técnicas, como *flooding* [73], baseada em probabilidades [74] ou na localização [75], de forma epidémica [76] ou com base em domínios [77].

É importante que as diversos dispositivos de rede colaborem entre si e, para isso, é fundamental que o seu conhecimento individual seja propagado pela rede, de modo a colocar as informações de cada um ao serviço de todos. Apenas desta forma é possível atingir uma gestão distribuída da rede.

Em [78] é apresentado um conjunto de mecanismos integrados para especificação, implementação e disseminação das regras de gestão num domínio hierárquico. Para o efeito é usado um coordenador central que agrega e dissemina todas as regras do domínio. Este facto compromete, contudo, toda a escalabilidade da solução proposta.

Em [79] os autores propõem o mecanismo *Information Dissemination Management* (IDM), através do qual são coordenadas todas as etapas da disseminação de informação. No entanto, quando o número de regras e informação a serem propagadas tende a ser elevado (cenários de larga escala), ter um único mecanismo de controlo que consiga englobar toda a disseminação torna-se bastante complexo.

Em [80] são abordadas algumas técnicas baseadas em ontologias[1], cujo benefício é a organização da informação para disseminação. Porém, é exigido como pré-requisito uma classificação bem definida da semântica aplicada.

É comum a utilização de métricas sociais em *Delay Tolerant Networks* (DTNs) uma vez que neste tipo de redes nós podem mover-se livremente, fazendo com que nem sempre existam ligações entre os eles. Como possível solução, é proposto em [81] um algoritmo para identificação de nós *bridge* para interligação entre entidades disjuntas. A solução apresenta menor *overhead* que o algoritmo

[1] Nas ciências de computação corresponde a uma descrição de conceitos considerados por um conjunto de entidades.

epidémico, dado que a partilha de informação entre os nós se faz apenas com informações parciais da rede. Contudo, os resultados em termos de atraso fim-a-fim e número de saltos ficam abaixo do algoritmo epidémico.

Em [82] é proposta uma organização dos nós em comunidades sociais para o envio de informação em *multicast* em DTNs. Em termos do encaminhamento de dados, o custo em número de *relays* é menor na solução proposta comparativamente ao algoritmo epidémico. No entanto, o desempenho em termos de atraso e a percentagem de entrega de pacotes é pior. Em [83] é proposto um outro algoritmo distribuído com aplicação no mesmo tipo de redes. Neste, o encaminhamento de dados através das várias comunidades de nós formadas resulta num custo menor em termos de mensagens necessárias comparativamente ao algoritmo de *flooding*, porém, a taxa de entregas bem sucedidas é menor.

Em [84] são propostos três algoritmos para detecção distribuída de comunidades estáticas e dinâmicas. No entanto as soluções apresentam várias limitações: apenas é apresentada uma análise do estado final e não da evolução ao longo do tempo; os valores dos limiares são definidos inicialmente de forma estática, em vez de serem dinâmicos ou usar-se lógica *fuzzy*; apenas é realizada a detecção de um tipo de relação social entre nós; a informação recolhida é permanente em vez de ser dinâmica com o tempo e com a evolução das relações entre nós.

2.4.4 Abordagens de eleição

A eleição é o mecanismo através do qual são atribuídas características específicas a determinados nós na rede. Os critérios para a escolha, bem como as características desses nós, podem ser vários.

Em [85] a métrica usada para definir os líderes de cada domínio formado foi o número de vizinhos. Assim, o nó eleito será aquele com a métrica de valor mais elevado. Há que ter em consideração que a mobilidade causará flutuações no valor da métrica, o que leva a que ocorram trocas constantes no nó identificado como líder de cada domínio. Isto implica que haja uma diminuição do *throughput* e do desempenho global da rede.

Já em [86], a eleição é realizada ao nó com a métrica de menor valor (no caso da velocidade) ou com a métrica de valor mais elevado (no caso da potência de transmissão). É apresentada uma comparação com a abordagem *lowest ID first*, ficando demonstrado através dos resultados que a abordagem proposta conduz a um menor número de eleições e de reagrupamentos. Este facto contribui para a diminuição do *overhead* na rede. No entanto, os resultados apresentados na comparação com a métrica da velocidade apenas consideram valores de potência de transmissão que garantem que toda a rede está ligada.

Em [87] são apresentadas versões melhoradas da eleição pelo grau mais baixo, em que há uma combinação ponderada através de pesos e do grau. Já em [88], além de pesos, são também tidos em conta factores como a distância entre nós, velocidade, posição geográfica, etc. A eleição torna-se assim mais robusta ao nível do número de reagrupamentos quando comparada com as heurísticas *highest degree* e *lowest ID*. Porém, a determinação dos factores que entram na eleição não são fáceis de determinar em cenários reais. Como forma de ultrapassar esta limitação é proposta em [89] uma solução baseada apenas em informação local. Contudo, mantém-se a necessidade de ajustar previamente o peso dos parâmetros usados pela heurística, tornando a solução pouco escalável para cenários distintos.

Em [90] é proposto um mecanismo de eleição baseado no desempenho das ligações e dos nós. É apresentado um comparativo do algoritmo de pré-eleição proposto, através do qual se elege um líder provisório, e o mecanismo de eleição tradicional. No entanto, não é apresentado o impacto na rede

ao nível do número de mensagens necessário para esta sinalização. Percebe-se ainda que, em cenários com elevada dinâmica, esta abordagem pode-se tornar pouco eficiente.

Em [91] os autores apresentam o protocolo de eleição *Network Leader Election* (NLE) baseado em *Link State Routing* (LSR), garantindo consenso mesmo em situações de quebra de ligações e partições da rede. O protocolo apresenta um bom desempenho ao nível de *overhead* necessário em situações de falha do líder de um grupo. A dependência do LSR é uma limitação em cenários de larga escala, podendo levar a uma degradação do desempenho global do sistema.

Em [92] é referido um modelo de eleição uniforme, no qual cada estação transmite, com probabilidade igual, a sua informação em janelas de tempo definidas. Apesar de ter sido apresentada uma solução para melhor definir o número de janelas temporais, os autores não explicam como esperam resolver os problemas de coordenação da informação para obter uma eleição consensual entre os vários nós.

2.5 Gestão de redes sem fios pelo padrão 802.11

Nesta secção são apresentadas as características principais dos processos de controlo e gestão realizados na camada MAC do Institute of Electrical and Electronics Engineers (IEEE) 802.11. Esta informação será posteriormente necessária para compreender os mecanismos propostos nas redes sem fios.

2.5.1 Arquitectura geral

O protocolo 802.11 é baseado numa arquitectura onde o sistema se encontra dividido por células. Quando cada uma destas células é controlada por um AP, designam-se por *Basic Service Set* (BSS); quando não existe um controlador central designam-se por *Independent Basic Service Set* (IBSS).

O *Extendend Service Set* (ESS) é a designação dada ao conjunto de uma ou mais BSS interligadas, tal que, do ponto de vista lógico, todas as entidades possam ser consideradas parte da mesma BSS. O *Domain System* (DS) é quem interliga os APs de um ESS, permitindo a comunicação entre BSS. É ainda definido o conceito de Portal, que será um equipamento que permitirá a interligação entre uma rede 802.11 e uma *Local Area Network* (LAN) do padrão IEEE 802.

Ao nível dos identificadores, o mais global é o *Service Set Identifier* (SSID) que corresponde ao nome de uma determinada rede sem fios 802.11. Um ESS terá obrigatoriamente de possuir um único SSID, ou seja, cada uma das BSS que o constitui terá de partilhar o mesmo identificador. Esta *string* pode conter até 32 caracteres e há a possibilidade de ser *broadcast*, permitindo a criação de pontos de acesso virtuais (figura 2.9).

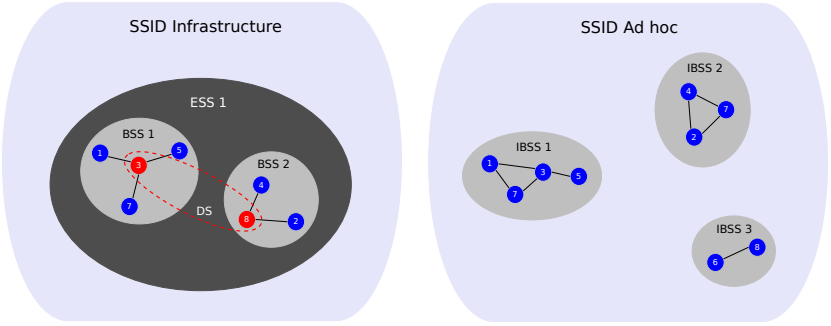


Figura 2.9: Arquitectura geral do padrão 802.11

Numa rede infra-estruturada, o Basic Service Set Identifier (BSSID) identifica cada uma das BSS, sendo por isso o endereço MAC do AP dessa célula.

Numa IBSS, o BSSID é um endereço MAC gerado localmente, contendo 46 *bits* aleatórios, o *bit* universal/local a '1' e o *bit* individual/group a '0' [93]. Um BSSID com todos os *bits* a '1' indica que é *broadcast* e só pode ser usado no modo *active probing*, durante o envio de *Probe Requests*. Os *bits* universal/local e individual/group são os *bits* menos significativos do primeiro octeto. Os restantes 46 *bits* são aleatórios, de modo a que seja provável a criação de endereços únicos (figura 2.10).

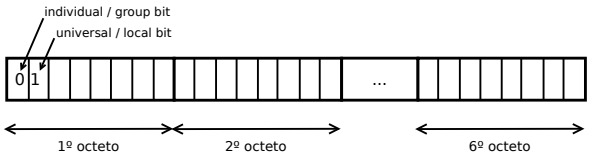


Figura 2.10: Constituição do BSSID numa IBSS segundo o protocolo 802.11

2.5.2 Camadas protocolares

O protocolo 802.11 suporta tanto a camada física como a MAC, estando definido no padrão actual uma única camada MAC que interage com três esquemas de transmissão (figura 2.11): *Frequency Hopping Spread Spectrum* (FHSS), *Direct Sequence Spread Spectrum* (DSSS), *Orthogonal Frequency Division Multiplexing* (OFDM).

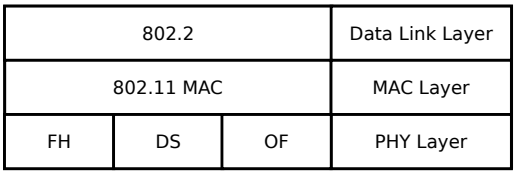


Figura 2.11: Camadas protocolares do 802.11, adaptado de [94]

2.5.3 Modos de operação

O padrão 802.11 disponibiliza dois modos de operação distintos: o infra-estruturado e o *ad hoc* (figura 2.12). No primeiro, existem pontos de gestão centrais (APs) com os quais as restantes entidades, designadas *Stations* (STAs), comunicam. Portanto, não existe qualquer tipo de comunicação entre STAs e os APs são possíveis pontos de falha do sistema, uma vez que a sua avaria isola determinadas porções da rede. Já no modo *ad hoc*, a comunicação é feita sem recurso a qualquer tipo de infra-estrutura, mas sim de forma directa entre as várias entidades.

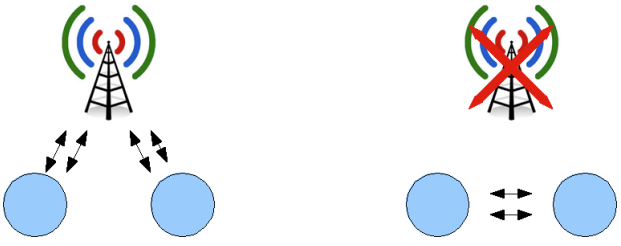


Figura 2.12: Modo infra-estrutura *vs* modo *ad hoc* [95]

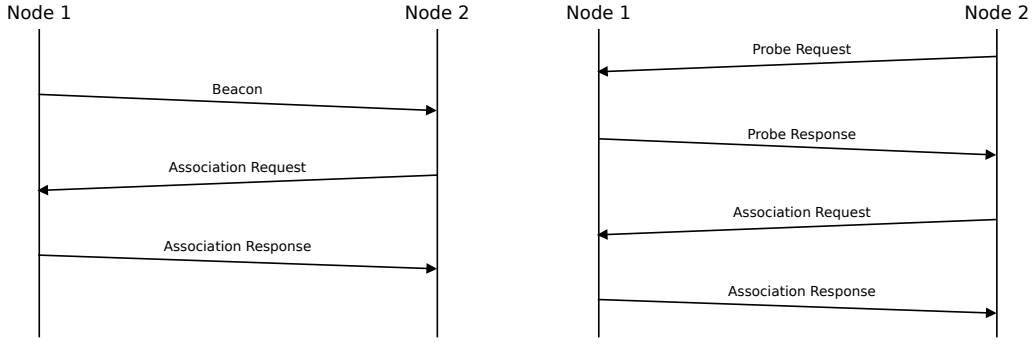
2.5.4 Modos de *scanning/probing*

Após ser inicializado (*bootstrapping*), cada nó irá procurar por outros de forma a associar-se a estes. Esta associação não é definitiva, uma vez que devido à mobilidade, os nós podem sair do raio de alcance daqueles a que estão associados. Quando tal acontece é necessário despoletar um novo processo de procura até encontrar um nó a que se possa associar.

Este processo de procura pode ser feito por dois métodos segundo o padrão 802.11 MAC (figura 2.13), tal como descrito em [96]:

- **Modo passivo:** o nó vai escutar o meio e aguardar a recepção de *Beacons* cujo SSID seja o mesmo que o seu. Após receber este tipo de pacote, o nó iniciará o processo de associação com o envio de um *Association Request*. Uma das possíveis vantagens deste modo será o reduzido número de pacotes trocados, isto é, o baixo *overhead* de mensagens, potenciando o *throughput* da rede.

- **Modo activo:** neste modo o nó, em vez de aguardar a recepção de *Beacons* por parte das outras entidades, envia de imediato um pacote *Probe Request* (em *broadcast*), onde vai o SSID da rede em que está configurado. Haverá respostas (*Probe Responses*) por parte de todos os nós ao alcance que pertencam ao mesmo SSID. O nó pode, alternativamente, enviar *Probe Request* com *broadcast* SSID, fazendo com que todos os nós ao alcance respondam. Uma das possíveis vantagens do modo activo será a rápida descoberta de nós ao alcance, que é importante não só no *start-up* da rede, como na manutenção da mesma (quando ocorrem perdas de ligação).



(a) *Scanning/Probing* Passivo (b) *Scanning/Probing* Activo

Figura 2.13: Modos de *Scanning/Probing*

2.5.5 Espaçamento entre tramas

São definidos quatro tipos de espaçamento entre tramas consecutivas, de modo a permitir diferentes prioridades no acesso ao meio [97]:

- *Short Inter Frame Space* (SIFS): é o menor espaçamento possível, sendo usado para separar pacotes ou fragmentos de um diálogo único, permitindo à estação em transmissão ter prioridade sobre todas as outras. Segundo o 802.11 FH PHY o valor é 28 ms;

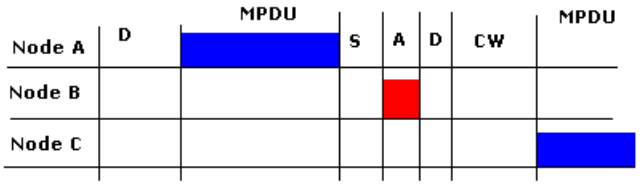
- *Point Inter Frame Space* (PIFS): é usado pelo AP para que este consiga o acesso ao meio antes de qualquer STA, sendo calculado através de SIFS + 78 ms ;

- *Distributed Inter Frame Space* (DIFS): é o tempo que as STAs têm de aguardar até conseguirem acesso ao meio, sendo calculado através de PIFS + 128 ms;

- *Extended Inter Frame Space* (EIFS): é o intervalo de tempo máximo que uma STA pode aguardar caso tenha recebido um pacote que não consiga interpretar, de forma a evitar que ocorram pedidos de retransmissão e colisões derivadas de transmissões/diálogos incompletos.

2.5.6 CSMA/CA

O acesso ao meio em redes sem fios com o protocolo 802.11 é realizado através do mecanismo *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) [98] que se assemelha ao mecanismo *Collision Detection* (CSMA/CD) usado em redes Ethernet. Através deste mecanismo, um elemento da rede que queira enviar um pacote começa por escutar o canal. Se este se encontrar desocupado, ou seja, não existirem transmissões de outros nós, o pacote é enviado. Caso contrário, é necessário aguardar um período de contenção (*Contention Period* - CP), que é um intervalo de tempo aleatório que todos os nós aguardam após qualquer transmissão, permitindo que todos os nós tenham igual probabilidade de acesso ao meio. Se no final deste período o canal se encontrar desocupado, o nó envia o pacote; senão, repete o processo anterior até detectar o meio livre. Este mecanismo é ilustrado na figura 2.14. O algoritmo é designado de *Exponential Backoff* e tem de ser executado sempre que um nó tenta fazer o primeiro envio de um pacote e o meio está ocupado, após qualquer retrasmissão ou transmissão com sucesso. O algoritmo não é usado unicamente quando um nó decide enviar um pacote e o meio está livre, pelo menos, ao tempo correspondente ao DIFS. O mecanismo pode ser distribuído ou centralizado, sendo utilizada respectivamente a *Distributed Coordination Function* (DCF) ou a *Point Coordination Function* (PCF).



Legenda	
D	*Distributed Inter Frame Space* (DIFS)
S	*Short Inter Frame Space* (SIFS)
CW	*Contention Window* (CW)
MPDU	MAC *Protocol Data Unit* (MPDU)
A	*Acknowledge* (ACK)

Figura 2.14: Mecanismo CSMA/CA [99]

Para aumentar a eficiência do protocolo foram implementados alguns melhoramentos. O *Positive Acknowledgement* consiste no seguinte: se um pacote for bem recebido, é retornado um ACK; caso o pacote seja recebido com erros ou nem sequer seja recebido, o receptor não responderá. Isto permite ao emissor saber quando é que a sua informação chega ao destino e, caso não se verifique, pode proceder à retransmissão da mesma. O CP é iniciado após o instante em que o ACK deveria ter sido enviado. Como forma de reduzir a probabilidade de erro dos pacotes de tamanho elevado, existe a possibilidade

de fragmentação dos mesmos em pacotes de menor dimensão que são reagrupados no receptor. Desta forma, tanto a probabilidade de erro como a de retransmissão são reduzidas.

Os pacotes *Request to Send* (RTS) e *Clear to Send* (CTS) fazem parte de uma técnica de *handshake* usada no mecanismo de CSMA/CA para resolver um problema conhecido como "nó escondido" (*hidden node problem*). Na figura 2.15, o nó B consegue ouvir tanto o nó A como o nó C; no entanto estes não se conseguem ouvir. Assim, os nós A e C podem transmitir simultaneamente, ficando o nó B com dados corrompidos.

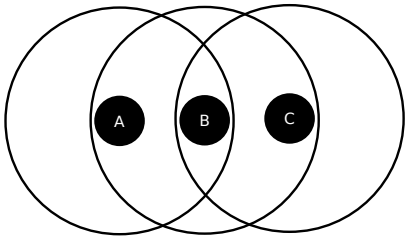
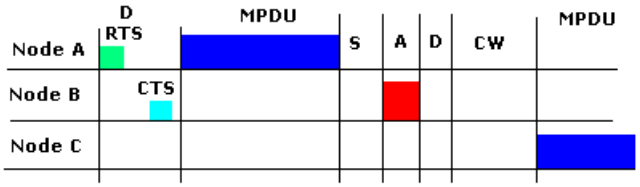


Figura 2.15: Problema do nó escondido, adaptado de [99]

Uma solução proposta foi a introdução de maior controlo nas transmissões, recorrendo ao envio dos pacotes RTS e CTS antes da transmissão do pacote de dados. Na situação da figura 2.16, o nó A envia um pacote RTS que é recebido pelo nó B. Este envia um pacote CTS que é recebido tanto por A como por C, e assim o nó C já não iniciará a transmissão, tendo por isso de aguardar.



Legenda	
D	*Distributed Inter Frame Space* (DIFS)
S	*Short Inter Frame Space* (SIFS)
CW	*Contention Window* (CW)
MPDU	MAC *Protocol Data Unit* (MPDU)
A	*Acknowledge* (ACK)

Figura 2.16: Mecanismo CSMA/CA com RTS/CTS [99]

2.5.7 Pacotes ao nível MAC

2.5.7.1 Tipos de pacotes

O padrão 802.11 define vários tipos de pacotes [100] que os nós usam para fazer a gestão, o controlo ou comunicação através de ligações sem fios.

Pacotes de gestão: Os pacotes de gestão permitem aos nós estabelecer e manter comunicações. Os sub-tipos destes pacotes são:

- *Beacon*: são gerados periodicamente como forma de anunciar a presença e informações da entidade que os envia (e.g. *timestamp*, SSID, etc). As entidades varrem continuamente os canais

como forma de detectar a recepção de novos Beacons.

- *Probe Request*: os pacotes de *Probe* são usados no modo *Active Scanning*, quando uma STA necessita de obter informação acerca da presença de entidades no meio envolvente, em vez de aguardar o envio de Beacons por parte destas.

- *Probe Response*: é a resposta ao pedido anterior, onde estarão encapsuladas informações acerca da entidade que o envia (e.g. *capability information*, *supported data rates*, etc).

- *Authentication*: a autenticação é o processo através do qual uma entidade aceita ou rejeita a identificação de uma STA. Por defeito é usado um sistema de autenticação aberto, mas pode ser usada uma chave partilhada. Neste último, a STA envia o pacote de autenticação e quem o recebe responde com o *challenge text*. Em seguida, a STA deve enviar o *challenge text* encriptado usando a chave *Wired Equivalent Privacy* (WEP). A entidade a quem é feito o pedido deve assegurar-se que a STA tem a chave WEP correcta, e para isso descodifica o texto recebido e compara-o com o original. O resultado desta comparação significará a aceitação ou rejeição da autenticação da STA.

- *Deauthentication*: este pacote é enviado quando se deseja terminar uma comunicação segura entre duas entidades.

- *Association Request*: a associação é iniciada através do envio de um pacote *Association Request*, onde é colocada informação da STA que faz o pedido. A entidade que recebe o pedido deverá verificar alguns parâmetros contidos no pacote (e.g. *supported data rates*), enviando em seguida uma resposta, aceitando ou rejeitando a associação.

- *Association response*: é a resposta ao pedido de associação e pode ser positiva ou negativa. Se a associação for concedida, o pacote incluirá informação relativa à associação (e.g. *supported data rates*).

- *Reassociation Request*: uma STA que se afaste demasiado do AP onde realizou a associação e encontre um outro AP, poderá realizar um pedido de reassociação com o novo AP. Este ficará responsável por encaminhar os dados que possam estar ainda em fila de espera no AP anterior para serem transmitidos à STA.

- *Reassociation Response*: será semelhante ao pacote *Association Response*, mas neste caso relaciona-se com o pedido de reassociação.

- *Disassociation*: uma STA enviará este pacote quando deseja terminar a associação previamente estabelecida (e.g. bateria fraca). Desta forma, a entidade onde estiver associada pode libertar os recursos associados a essa associação e eliminar essa STA da tabela local de associações.

Pacotes de controlo: Os pacotes de controlo garantem a entrega dos pacotes de dados entre dois nós. Os sub-tipos destes pacotes são:

- RTS: o mecanismo de *handshake* RTS/CTS é de uso opcional e visa reduzir as colisões entre pacotes quando ocorre o fenómeno do "nó escondido". A primeira fase consiste no envio do pacote RTS por parte da entidade que pretende enviar o pacote de dados.

- CTS: na segunda fase, a entidade que recebe o RTS responde com um CTS, possibilitando o envio do pacote de dados e impedindo simultaneamente o envio de dados por parte de outras estações durante um determinado período de tempo.

- ACK: se o pacote de dados foi recebido sem erros, é enviado um ACK de volta. Caso a entidade emissora não receba o ACK durante um determinado período de tempo após o envio dos dados, procederá à retransmissão dos mesmos.

Pacotes de dados: O padrão 802.11 define um tipo específico para os pacotes de dados, onde a informação que estes transportam estará encapsulada ao nível do campo *Frame Body*.

2.5.7.2 Formato genérico

Os pacotes trocados terão, ao nível do cabeçalho MAC [101], a seguinte estrutura da figura 2.17:

Figura 2.17: Formato do cabeçalho do padrão 802.11 ao nível MAC [102]

Campo *Frame Control* (figura 2.18)

***Protocol Version*:** Indica qual a versão do protocolo 802.11 MAC que está contida no pacote. Apesar de apenas existir a versão '00', todas as outras combinações estão reservadas para futuras versões que se concluam incompatíveis com as especificações iniciais do padrão;

***Type* e *Sub-Type*:** Identificam o tipo e sub-tipo de pacote em questão (tabela 2.2);

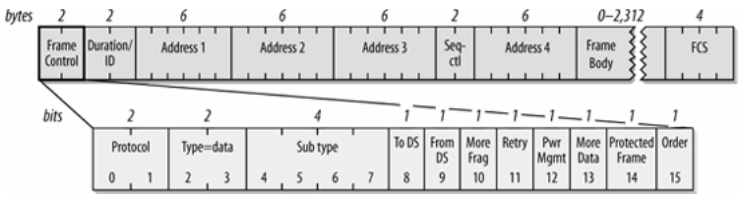


Figura 2.18: Constituição do campo *Frame Control* [102]

Valor do sub-tipo	Nome do sub-tipo
	Management frames (*type*=00)[2]
0000	*Association Request*
0001	*Association Response*
0010	*Reassociation Request*
0011	*Reassociation Response*
0100	*Probe Request*
0101	*Probe Response*
1000	*Beacon*
1001	*Annoucement traffic indication message* (ATIM)
1010	*Disassociation*
1011	*Authentication*
1100	*Deauthentication*
1101	*Action* (para gestão do espectro com 802.11h e para QoS)

(a)

Valor do sub-tipo	Nome do sub-tipo
	Control frames (*type*=01)[3]
1000	*Block Acknowledge Request* (QoS)
1001	*Block Acknowledge* (QoS)
1010	*PS-Poll*
1011	RTS
1100	CTS
1101	ACK
1110	*CF-End*
1111	*CF-End* + *CF-Ack*

(b)

Valor do sub-tipo	Nome do sub-tipo
	Data Frames (*type*=10)[4]
0000	*Data*
0001	*Data* + *CF-Ack*
0010	*Data* + *CF-Poll*
0011	*Data* + *CF-Ack* + *CF-Poll*
0100	*Null data* (sem transmissão de dados)
0101	*CF-Ack* (sem transmissão de dados)
0110	*CF-Poll* (sem transmissão de dados)
0111	*CF-Ack* + *CF-Poll* (sem transmissão de dados)
1000	QoS *Data*[5]
1001	QoS *Data* + *CF-Ack*
1010	QoS *Data* + *CF-Poll*
1011	QoS *Data* + *CF-Ack* + *CF-Poll*
1100	QoS *Null* (sem transmissão de dados)
1101	QoS *CF-Ack* (sem transmissão de dados)
1110	QoS *CF-Poll* (sem transmissão de dados)
1111	QoS *CF-Ack* + *CF-Poll* (sem transmissão de dados)

(c)

Tabela 2.2: Campos *Type* e *Subtype*, adaptado de [102]

[2] Nos management frames, os sub-tipos 0110-0111 e 1110-1111 são reservados e não estão actualmente em uso
[3] Nos control frames, os sub-tipos 0000-0111 são reservados e não estão actualmente em uso
[4] Nos data frames, o tipo 11 é reservado
[5] Proposto pelo protocolo 802.11e mas ainda não standardizado

***Bits ToDS* e *FromDS*:** Estes *bits* indicam se o pacote está a ser trocado num modo centralizado ou distribuído. Todos os pacotes de redes infra-estruturadas têm de ter pelo menos um dos *bits* DS a '1' (tabela 2.3);

	To DS= 0	*To DS*= 1
From DS= 0	Todos os pacotes de gestão e dados trocados numa IBSS	Pacotes de dados transmitidos de uma estação no modo infra-estrutura
From DS= 1	Pacotes de dados recebidos de uma estação no modo infra-estrutura	Pacotes de dados de uma *bridge* sem fios

Tabela 2.3: Significado dos *bits To DS* e *From DS*, adaptado de [102]

Bit More Fragments: Indica se um pacote vindo de uma camada superior foi fragmentado ao nível MAC;

***Bit Retry*:** Indica se o pacote é uma retransmissão;

***Bit Power Management*:** Indica se a estação está em modo de poupança de energia ou em modo activo. No caso dos APs, é obrigatórios estarem no modo activo;

***Bit More Data*:** Utilizado no modo de gestão de energia, é usado pelo ponto de acesso para sinalizar a estação que pacotes suplementares estão armazenadas em espera;

***Bit Protected Frame*:** Também designado de WEP *bit*, quando colocado a '1' indica que o pacote está protegido por protocolos de segurança de camadas superiores;

***Bit Order*:** Quando se exige que os pacotes e fragmentos sejam entregues de forma estritamente ordenada (*strict ordering*).

Campo *Duration/ID* (figura 2.19)

NAV: Representa o número de microssegundos que é expectável que o meio esteja ocupado para a transmissão em progresso. Todas as estações têm de monitorizar os cabeçalhos de todos os pacotes que recebem e actualizar o valor do campo NAV caso seja aumentado o tempo em que o meio está ocupado. O meio permanecerá bloqueado durante esse intervalo de tempo adicional;

Tramas *Contention Free Period* (CFP): Permite que as estações que não tenham recebido o *Beacon* sinalizando um período de contenção livre, possam actualizar o valor do NAV com um valor elevado adequado, de forma a não interferir com transmissões neste período;

Tramas *Power Saver Poll* (*PS-Poll*): Uma estação em modo de poupança de energia ligar-se-à em determinados períodos e, para que não sejam perdidos pacotes, a estação irá requisitá-los ao ponto de acesso. Para tal irá ser enviado um *PS-Poll*, de forma a que a estação tenha acesso aos pacotes armazenados em *buffer*.

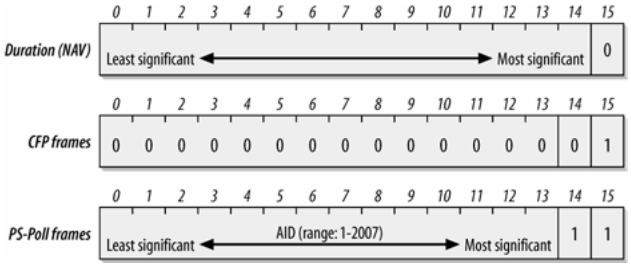


Figura 2.19: Constituição do campo *Duration*/ID [102]

Campos de Endereços Um cabeçalho ao nível MAC do 802.11 pode conter até quatro campos de endereços, cada um constituído por 48 *bits*. Os seus papéis serão diferentes conforme o tipo de pacote em questão (tabela 2.4).

To DS	*From DS*	*Address 1*	*Address 2*	*Address 3*	*Address 4*
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

Tabela 2.4: Papel desempenhado pelos vários campos de endereços [103]
DA = Destination Address; SA = Source Address; TA = Transmitter; RA = Receiver.

Campo *Sequence Control* Permite distinguir os diversos fragmentos de um único pacote. É composto por dois sub-campos que permitem reordená-los: o número de fragmento e o número de sequência (figura 2.20).

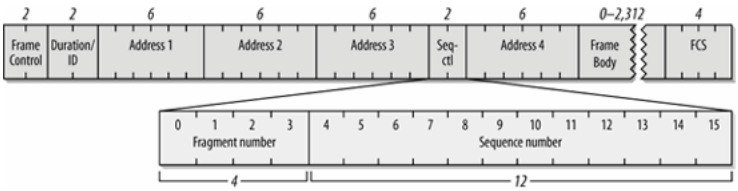


Figura 2.20: Constituição do campo *Sequence Control* [102]

Campo *Frame Body* O seu tamanho é variável e possui informação específica do tipo de pacote em questão. Os campos fixos e *Information Elements* (IE) que podem ser encontrados no *Frame Body* dos diferentes pacotes de gestão estão sumarizados na tabela 2.21:

	Association Request	Association Response	Reassociation Request	Reassociation Response	Probe Request	Probe Response	Beacon	Disassociation	Authentication	Deauthentication
Authentication Algorithm Number									x	
Authentication Transaction Sequence Number									x	
Beacon Interval						x	x			
Current AP Address			x							
Listen Interval	x		x							
Reason Code								x		x
Association ID (AID)		x		x						
Status Code		x		x					x	
Timestamp						x	x			
Service Set Identifier (SSID)	x		x		x	x	x			
Supported Rates	x	x	x	x	x	x	x			
Extended Supported Rates	x	x	x	x	x	x	x			
FH Parameter Set						x	x			
DS Parameter Set						x	x			
CF Parameter Set						x	x			
Capability Information	x	x	x	x		x	x			
Traffic Indication Map (TIM)							x			
IBSS Parameter Set						x	x			
Challenge Text									x	
ERP Information						x	x			

Figura 2.21: Constituição do *Frame Body* para os diferentes pacotes de gestão, adaptado de [104]

Information Element: Encontra-se encapsulado nos pacotes de gestão do protocolo 802.11 MAC. É a forma usada pelos equipamentos para transmitir informação sobre si mesmos. É por isso comum serem enviados vários IEs em cada pacote de gestão, obedecendo o seu formato a uma estrutura pré-definida de tamanho dinâmico (figura 2.22).

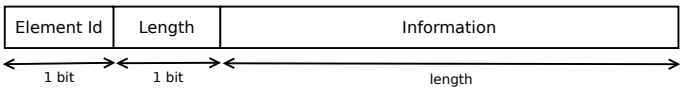


Figura 2.22: Estrutura de um *Information Element*, adaptado de [105]

Campo *Frame Check Sequence* Por vezes referenciado como *cyclic redundancy check* (CRC), permite às estações verificar a integridade dos pacotes recebidos. Caso a FCS seja positiva, é enviado um *acknowledge* pelo receptor. Caso seja negativa, o pacote deve ser retransmitido.

2.6 Conclusões

No presente capítulo começaram por se apresentar as abordagens tradicionais à gestão de redes. Da análise às mesmas pode concluir-se que:

- a gestão "centralizada" apresenta problemas sérios ao nível da escalabilidade, da falta de comunicação entre agentes e do ponto de falha central do sistema;
- a gestão "fracamente distribuída" tenta mascarar o problema da escalabilidade, introduzindo agentes intermédios de gestão, mas em rigor mantém as limitações da gestão centralizada;
- a gestão "fortemente distribuída" introduz a colaboração entre os agentes intermédios de gestão, mantendo, no entanto, o gestor central e, por isso, o ponto de falha central do sistema;

- a gestão "cooperativa" elimina a necessidade do gestor central, aumenta o nível de cooperação e torna as ligações dinâmicas entre agentes de gestão. Para isso exige implementações mais complexas e maior *overhead*.

Após identificadas estas limitações, e devido à crescente investigação no capítulo da gestão autonómica e distribuída, foram identificados os requisitos e desafios que são esperados das redes de próxima geração. Como possível solução foi apresentado o paradigma de INM que visa alcançar uma gestão distribuída da rede, incorporando capacidades de gestão em cada um dos elementos que a compõem. Neste sentido, é necessário implementar diversos mecanismos que permitam a comunicação entre elementos de rede, possibilitando a colaboração entre estes de modo a alcançar uma gestão mais eficiente em termos de escalabilidade, robustez e tolerância a falhas.

Após descritas as limitações de algumas abordagens a estes mecanismos encontradas na literatura, pretende-se na presente Dissertação desenvolver processos de *bootstrapping*, mecanismos de descoberta de entidades e de disseminação de informações na rede, bem como efectuar a eleição dinâmica de nós para atribuição de funções especiais. Todos estes mecanismos terão de cumprir os requisitos exigidos pelas NGNs (escalabilidade, integração com os restantes módulos, inexistência de pontos de falha críticos, adaptação a topologias dinâmicas, prescindir da sincronização global de informação da rede, etc).

As soluções desenvolvidas pretendem ser adaptáveis em termos de número de nós e dinâmica da rede, fazendo uso em simultâneo de conceitos inovadores como as métricas sociais [52, 106, 107], de forma a dar resposta aos desafios apresentados pelas NGNs.

Capítulo 3

Mecanismos de Cooperação

3.1 Introdução

De forma a implementar uma gestão distribuída e autonómica da rede é necessário que exista comunicação colaborativa entre os vários elementos que a constituem. O modelo de comunicação desenvolvido no âmbito da presente Dissertação está dotado dessa mesma colaboração, no qual se inserem os mecanismos que permitem a partilha de conhecimento entre os nós. Contudo, inúmeros desafios têm de ser levados em conta quando se pretende uma gestão sem recurso a elementos centralizadores.

No presente capítulo é dada uma visão geral dos mecanismos implementados, bem como de alguns conceitos e processos associados.

Em 3.2 são indicados os principais objectivos da solução implementada.

Na secção 3.3 são abordadas as redes sem fios e na secção 3.4 são abordadas as redes com fios.

O capítulo termina na secção 3.5 onde é apresentado um breve resumo do mesmo.

3.2 Objectivos

De modo a que a comunicação entre as várias entidades conduza a uma gestão autonómica e distribuída da rede, os vários mecanismos e funcionalidades desenvolvidas têm por objectivo conseguir realizar:

- *Bootstrapping* dos vários nós;
- Descoberta de outros nós na rede;
- Troca de informações entre os nós;
- Disseminação a múltiplos saltos de informações específicas;
- Recolha de informações necessárias para a eleição de líderes.

A estes objectivos transversais juntam-se ainda dois específicos das redes sem fios:

- Associação ponderada entre nós, com recurso a métricas sociais;
- Agregação dos nós em comunidades.

Ambas as redes com e sem fios têm os seguintes objectivos:

- Minimizar o número de mensagens usadas para realizar a descoberta;

- Minimizar o tempo de convergência da informação desse processo.

3.3 Arquitectura global

Os mecanismos desenvolvidos inserem-se numa arquitectura global que visa a gestão totalmente distribuída da rede através de mecanismos de cooperação e colaboração entre as diversas entidades que a constituem. Na figura 3.1 é apresentada a arquitectura global, onde se inserem os mecanismos desenvolvidos que servem como base a todas as interacções entre as entidades INM. De uma forma geral, cada entidade terá funcionalidades que lhe permitirão iniciar o processo de gestão por si só e, em seguida, descobrir e contactar os vizinhos. As informações recolhidas destes contactos serão guardadas em repositórios locais e servirão de base às decisões de gestão que serão tomadas, sendo necessária a disseminação destas através dos múltiplos domínios.

Em destaque no topo da figura 3.1 encontra-se a parte da arquitectura geral em que se inserem os mecanismos desenvolvidos, tanto para as redes com fios como sem fios.

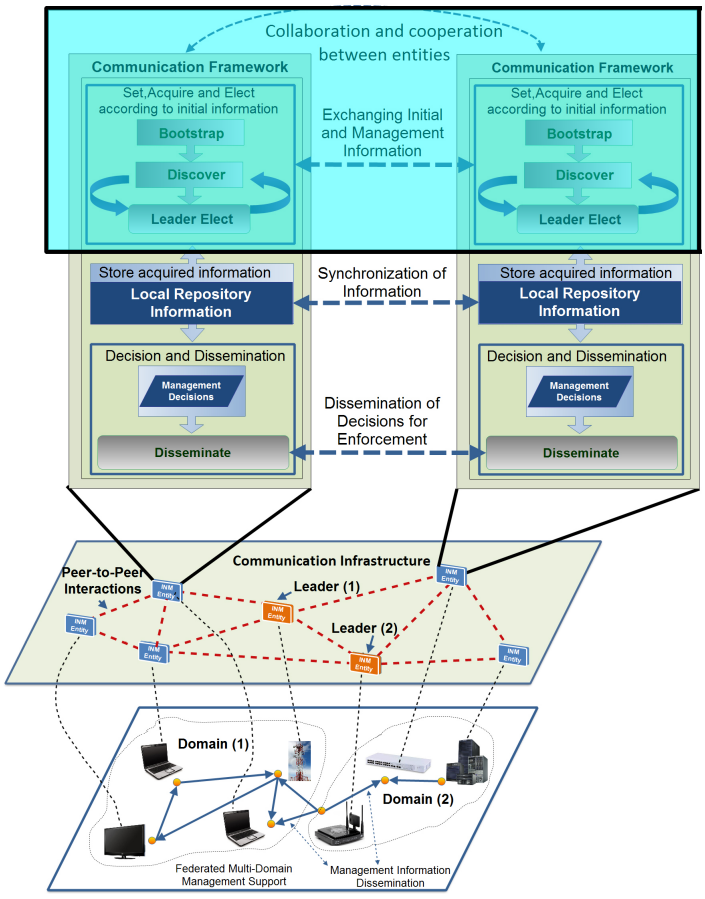


Figura 3.1: Arquitectura do modelo de comunicações segundo o conceito de INM [22]

3.4 Redes sem fios

3.4.1 Interligação das funcionalidades e mecanismos

A figura 3.2 resume a interligação entre as várias funcionalidades e mecanismos que serão descritos neste capítulo.

Ao nível do plano de comunicação, os nós interagem de forma directa com aqueles que estão ao seu alcance, dando origem a uma comunicação distribuída, ou seja, sem recurso a entidades centralizadoras de informação. O facto dos nós possuirem mobilidade traduz-se num maior dinamismo do cenário, implicando, porém, maior complexidade ao nível do nó devido às constantes mudanças no meio envolvente.

O plano de gestão local a cada nó, devido à arquitectura distribuída do sistema, é constituído por dois blocos principais: os repositórios locais e os mecanismos de cooperação. Do primeiro fazem parte as tabelas dinâmicas de informação local *Partial View* e *KnownNodes* (descritas em 4.2.2.8). No segundo inserem-se os mecanismos de *bootstrapping*, descoberta e eleição. Através destes são recolhidas informações relevantes da rede, que posteriormente são inseridas nos repositórios locais de modo a aumentar e actualizar constantemente o conhecimento do nó acerca dos restantes. Esta aquisição de conhecimento através da colaboração entre entidades é necessária mas simultaneamente complexa, devido à inexistência de pontos comuns para sincronização de informação, originando problemas de consenso em sistemas distribuídos. Os desafios que a solução implementada enfrentou são descritos na secção seguinte 4.2.3.

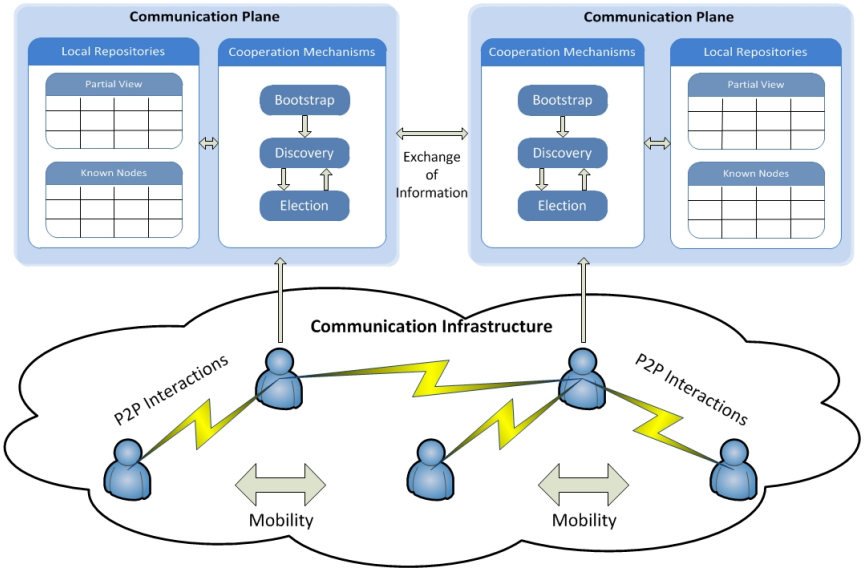


Figura 3.2: Interligação entre as várias funcionalidades e mecanismos implementados

3.4.2 Mecanismos e sua função

A função dos mecanismos implementados é permitir a comunicação sem fios entre entidades com diferentes padrões de mobilidade e a gestão distribuída da rede onde se inserem, de forma a cumprir os requisitos exigidos pelas NGNs.

A arquitectura foi construída de forma *bottom-up*, isto é, desde a implementação do modelo *ad hoc* até ao desenvolvimento das funcionalidades de mais alto nível. Os mecanismos em seguida apre-

sentados visam alcançar uma gestão eficiente, escalável, robusta, com baixo *overhead* de mensagens e complexidade reduzida do ponto de vista externo, mesmo considerando ambientes dinâmicos. Aqueles que se consideram mais relevantes são:

- *Bootstrapping*: é o processo de inicialização de cada entidade. É constituído por diversas chamadas de funções que configuram diferentes parâmetros fundamentais à existência daquela entidade na rede, de modo a que funcione segundo o protocolo usado (802.11). Como exemplo temos a atribuição do identificador único, endereço MAC, BSSID, conhecimento das suas capacidades de *hardware*, parâmetros temporais, etc. A partir desta fase o nó está preparado a iniciar a sua actividade na rede.

- Descoberta: é o mecanismo que permite a uma entidade descobrir outros nós na rede, de forma a que seja possível posteriormente criar ligações de associação. No presente trabalho a descoberta pode ser feita por contacto directo (informação recolhida directamente desse nó) ou indirecto (informação transmitida por outro nó). Assim, quando uma entidade recebe um *Beacon* pode ficar não só a conhecer o nó que o enviou, como também todos os nós que este já conhecia e que pertencem à sua comunidade. É necessariamente um processo contínuo já que, num cenário com mobilidade, as condições vão variando ao longo do tempo.

- Troca de informação entre vizinhos: corresponde ao contacto directo, ou seja, entre entidades a 1-salto. O mecanismo usado para este efeito é a troca de pacotes de gestão do padrão IEEE 802.11 MAC modificados. A estes foram acrescentados novos IEs que visam permitir uma gestão colaborativa e mais eficiente da rede, possibilitando por exemplo, a associação ao melhor nó ao alcance, deixando de ser feita de forma indiferenciada.

- Disseminação de conhecimento: corresponde ao contacto indirecto entre entidades, ou seja, à troca de informações a múltiplos saltos. Os mecanismos para a partilha de conhecimento de forma distribuída têm como objectivo aumentar a eficiência e rapidez na convergência da informação. Este conhecimento relaciona-se sobretudo com os nós que, num determinado momento, pertencem a um domínio e algumas características associadas aos mesmos. Para isso há que ter em consideração o elevado grau dinâmico dos cenários considerados, nos quais os nós podem entrar ou sair de uma comunidade em qualquer momento.

- Suporte à criação de multi-domínios: ter a rede dividida em domínios auto-organizados é vantajoso na medida em que cada entidade não necessita de ter uma visão global da rede, mas apenas do domínio em que se encontra, diminuindo a quantidade de informação necessária a recolher e a armazenar. No entanto, esta divisão não pode isolar partes da rede, porque se pretende que exista comunicação e colaboração não só dentro como entre domínios. Esta poderá ser realizada através dos líderes eleitos em cada domínio, alcançando assim uma gestão distribuída e comunicação entre toda a rede.

- Eleição: paralelamente ao processo de descoberta ocorre o mecanismo de eleição. Através deste são escolhidas as melhores entidades de cada domínio, com base nas métricas sociais, às quais serão atribuídas funcionalidades específicas. Contudo, este processo é dinâmico, ou seja, a eleição não é definitiva, podendo novos líderes serem eleitos caso as condições assim o exijam. É necessário encontrar a melhor solução tanto para a sinalização dessas entidades, como para o

consenso das entidades eleitas, já que o processo de eleição é totalmente distribuído. Acresce ainda a estes desafios o elevado grau de dinâmismo dos cenários, fazendo com que as condições se alterem rapidamente. É por isso importante definir os critérios para que não ocorram mudanças constantes de líder, o que levaria a períodos de interrupção das comunicações, degradando parâmetros ao nível da camada IP, como o *throughput* ou o atraso fim-a-fim.

3.4.3 Funcionamento dos principais mecanismos

O *bootstrapping* é o processo que todas as entidades têm de executar para que a sua configuração inicial seja realizada. Após esta fase garante-se que essa entidade possui as configurações necessárias para que o seu comportamento na rede seja coerente com a dos restantes elementos (e.g. parâmetros relativos ao protocolo de gestão da rede, identificador único, inicialização de repositórios locais, etc). Uma possível representação visual do processo de *bootstrapping* pode ser ilustrado pela figura 3.3.

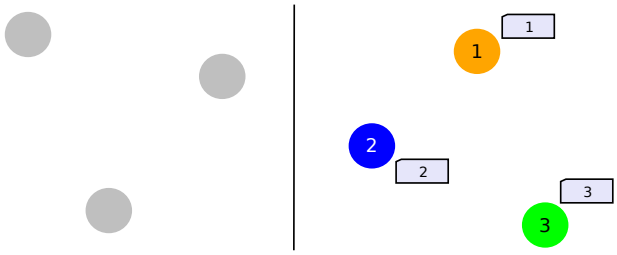


Figura 3.3: Representação genérica do processo de *bootstrapping*

O mecanismo de descoberta implementado baseia-se tanto no contacto directo como indirecto entre nós, de forma a estes conhecerem determinadas características associadas a cada um. O processo de decoberta é constante, não só pela busca contínua por novos nós, como pela actualização das informações dos já conhecidos.

Inicialmente os nós apenas se conhecem a eles próprios (etapa inicial da figura 3.4) e, através do contacto directo (e.g. recepção de *Beacons*), os nós passam a conhecer também o nó de origem do pacote. As informações relevantes são recolhidas e armazenadas na tabela local *Partial View*, correspondente às entidades na vizinhança (etapa intermédia da figura 3.4). No entanto, devido ao raio de alcance limitado, o contacto directo entre nós apenas permite um conhecimento muito reduzido da rede, isto é, em cada instante o nó apenas consegue "visualizar" a sua vizinhança.

Surge então a necessidade de introduzir cooperação entre entidades ao nível da descoberta, permitindo aos nós propagar o seu conhecimento pelas várias entidades (etapa final da figura 3.4). Isto resulta num conhecimento mais abrangente da rede, permitindo ao nó ter mais informação na qual se pode basear para tomar decisões. Para possibilitar este mecanismo, algum do conhecimento do nó é inserido ao nível do *Community-Based Beacon IE*, e quando o pacote é recepcionado, a informação é desencapsulada e adicionada ao repositório de informação local de nós conhecidos: tabela *Known Nodes*. O mecanismo de conhecimento indirecto de nós é abordado em mais detalhe na secção 4.2.2.10.

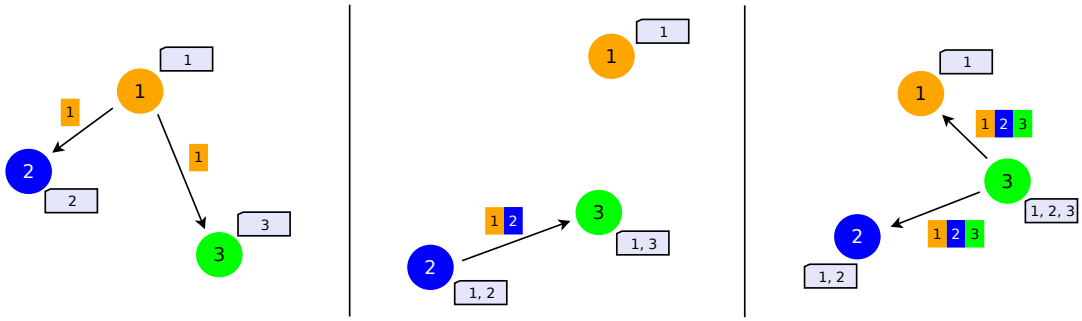


Figura 3.4: Etapas genéricas do processo de descoberta

Ao nível do processo de eleição, é necessário que cada entidade envie as informações associadas ao nó que considera ser o melhor (e.g. ID e métrica), tal como representado na etapa inicial da figura 3.5. Como o processo é realizado de forma distribuída, é necessário que todas as entidades recolham essa informação e a propaguem (etapa intermédia da figura 3.5), de forma a que seja antigida a convergência/consenso da eleição (etapa final da figura 3.5). Na abordagem proposta considera-se que o nó com maior métrica é considerado o líder dentro de uma determinada comunidade. No entanto, como os cenários são muito dinâmicos, quer em termos da mobilidade dos nós, quer do valor da própria métrica de cada um, o processo de sinalização dos nós eleitos requer ainda bastantes desenvolvimentos. Também as características e funções específicas que os líderes terão de desempenhar não estão ainda concretamente definidas, prevendo-se porém que o seu papel será fundamental ao nível do controlo do envio de informação entre nós para minimizar o tráfego na rede.

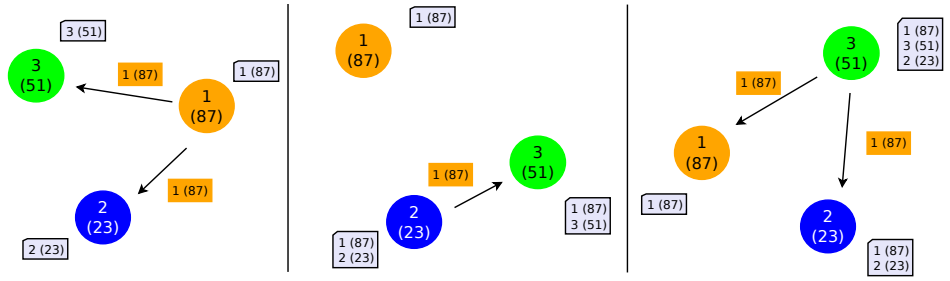


Figura 3.5: Representação genérica do processo de eleição

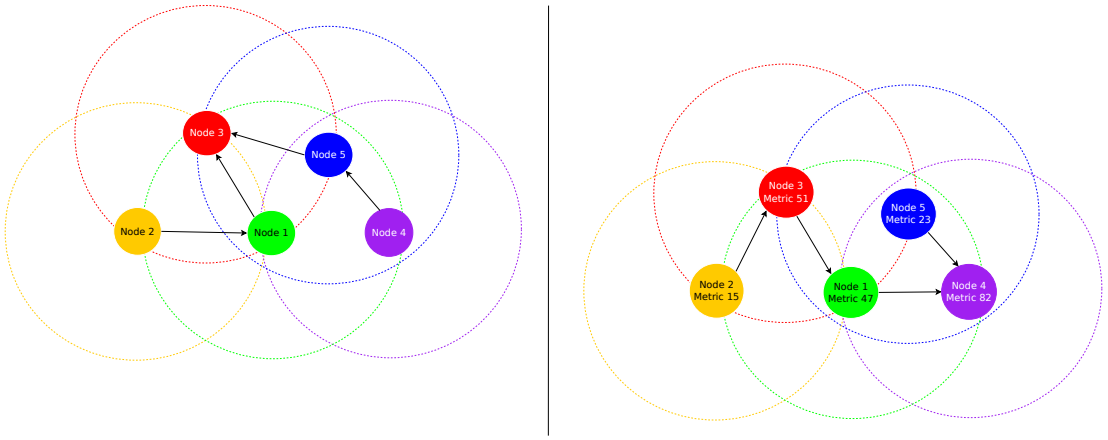
3.4.4 Processos e conceitos associados

3.4.4.1 Associação ao melhor nó

Uma das alterações importantes introduzidas relativamente ao padrão do protocolo 802.11 MAC foi a associação entre nós deixar de ser feita indiferenciadamente (ao primeiro nó detectado) e passar a ser feita ao nó ao alcance cuja métrica social seja a mais elevada (figura 3.6). Assim, a comunidade é criada de forma distribuída e optimizada, já que cada nó se associa àquele que considera ser o melhor na sua proximidade.

Paralelamente decorre o processo de manutenção da comunidade, uma vez que as associações podem não ser definitivas. De cada vez que um nó já associado receba contacto de outro cuja métrica social seja superior, é necessário que sinalize o nó onde mantém a ligação actual, indicando que pretende terminar essa associação.

Desta forma conseguir-se-á ter, em qualquer momento, o grafo de associação estabelecido com as melhores ligações de acordo com os parâmetros definidos.



(a) (b)

Figura 3.6: *Standard Association versus Social-based Association*

Para evitar mudanças constantes na associação por diferenças ligeiras da métrica social, foi definido um limiar superior até ao qual os valores de métricas sociais recebidas não despoletam um novo processo de associação.

Está ainda definido que este limiar é adaptativo conforme o valor do peso imposto a cada um dos parâmetros de entrada da métrica social. Deste modo consegue-se que o valor a partir do qual é realizada uma nova associação seja proporcional aos pesos impostos (w_1, w_2 e w_3). Assim, o limiar definido pela equação 3.1 actua de forma semelhante em cenários distintos.

$$threshold = \beta * (w1 + w2 + w3) \tag{3.1}$$

onde, β é uma percentagem do somatório dos pesos considerados (e.g. 10% nas simulações realizadas).

3.4.4.2 Perda de ligação ao nó associado

Após ser criada uma associação entre dois nós, estes podem afastar-se pelo facto de possuirem mobilidade, resultando na quebra de ligação entre ambos. Esta situação ocorre devido ao alcance limitado dos nós, originando situações de interrupção das comunicações. Este tipo de fenómeno implica que a associação seja terminada, sendo necessário que, uma vez detectada esta situação, o nó seja capaz de despoletar um novo processo de associação assim que se aperceber de outro nó no seu raio de alcance (figura 3.7).

Neste sentido foi implementado um mecanismo do tipo *break-before-make* [108]. Para tal usou-se um *watchdog timer* com escala igual a cinco vezes o intervalo entre *Beacons* indicado pelo nó onde a associação foi estabelecida. A cada recepção de *Beacons* vindos desse nó o *timer* é reiniciado. Ao atingir o fim de escala, o nó detecta a perda de ligação, permitindo-lhe iniciar uma nova associação, ou seja, nesta situação a máquina de estados é reiniciada.

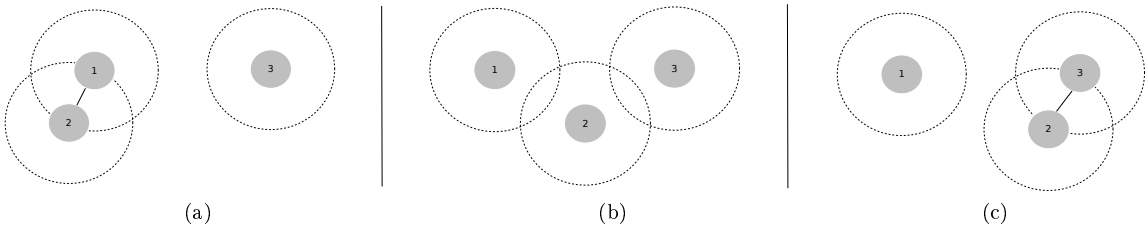


Figura 3.7: Fases desde a perda de ligação até à nova associação

3.4.4.3 Identificador da comunidade

Uma comunidade não é mais que um conjunto de nós associados entre si, ou seja, numa comunidade há a garantia que existe sempre um caminho de ligações de associação entre quaisquer dois nós. Este conceito é importante do ponto de vista da gestão dado que, numa rede cuja arquitectura seja distribuída, caso as entidades estejam organizadas numa única comunidade, o encaminhamento de dados entre quaisquer dois nós é garantido. Caso existam múltiplas comunidades na rede, a transferência de informação entre comunidades é complexa: terão de ser definidas entidades responsáveis por realizar essa função; e também será necessário introduzir maior conhecimento ao nível do nó, dado que este necessitará não só de conhecer os nós da sua comunidade, como também os que pertencem a cada uma das restantes, de modo a poder realizar o encaminhamento correcto.

A formação de comunidades ocorre quando o canal de comunicação deixa de ser ideal e passa a apresentar perdas e/ou atrasos devido à propagação no meio livre. Nesta situação os nós deixam de se conseguirem ver todos uns aos outros, ou seja, passa a existir um raio de alcance limitado em cada um. Assim sendo, os nós irão associar-se a outros que estejam na sua proximidade, podendo haver a formação de múltiplos domínios, isto é, conjuntos distintos de nós associados entre si. Cada um destes domínios designa-se por comunidade, que é identificada com um ID único[1].

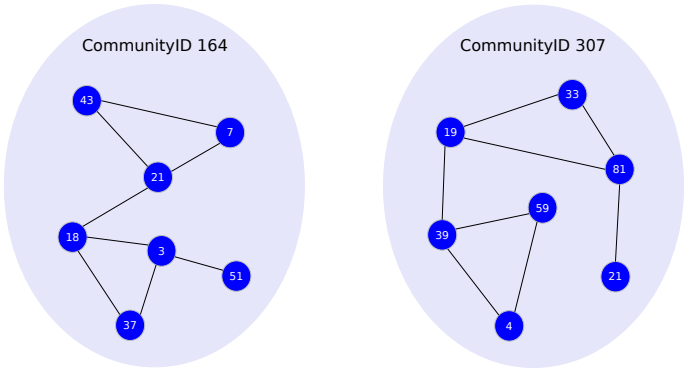


Figura 3.8: Conceito genérico do identificador de uma comunidade

Uma vez que tudo é realizado de forma distribuída, a gestão do identificador de cada comunidade terá de ser feita pelos vários nós que o constituem. Portanto terá de ser garantida a concordância desse mesmo identificador entre os vários elementos.

[1] Inteiro aleatório de 32 *bits*

Coerência do identificador da comunidade Um nó que se associe a outro adoptará o identificador da comunidade que lhe for enviado (comunidade onde estabeleceu a ligação) pelo nó onde efectuou a associação. No entanto, além da entrada e saída de nós individuais numa comunidade, existem situações especiais como a união e a separação de comunidades, que exigem a adopção de medidas adequadas para que seja evitada a inconsistência deste identificador.

União de comunidades Pode acontecer que as associações ocorram não só dentro da mesma comunidade, mas também entre nós de diferentes comunidades, dando origem a uma única comunidade de maior dimensão. Para garantir a coerência do identificador da comunidade é necessário que os nós adoptem apenas um deles. Para isso, o nó que efectua a associação adopta o identificador da nova comunidade e propaga-o ao(s) nó(s) a ele associado(s) e assim por diante, até que todos os elementos da comunidade concordem no identificador actual (figura 3.9).

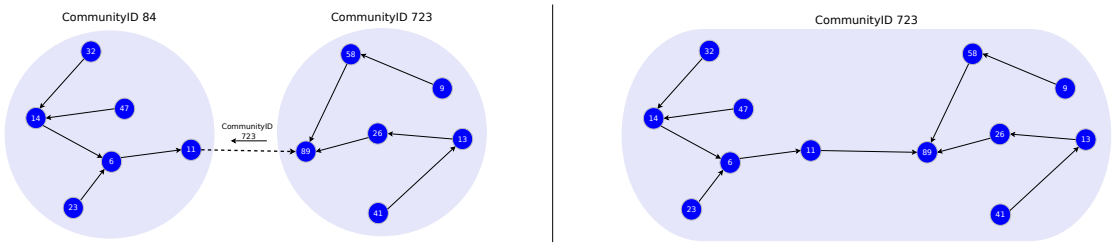


Figura 3.9: União de comunidades

Separação de comunidades O fenómeno oposto pode também acontecer, ou seja, a perda de ligação pode ocorrer não só nos extremos da comunidade, como também no interior da mesma. Neste caso, a comunidade separa-se em duas de menores dimensões. Uma das comunidades manterá o seu identificador e a outra terá de adoptar um novo ID. Para tal, o nó que perdeu a ligação será o responsável por gerar um novo identificador da comunidade e propagá-lo-á ao(s) nó(s) a si associado(s) e assim por diante, até que todos concordem no identificador actual da comunidade em que se encontram (figura 3.10).

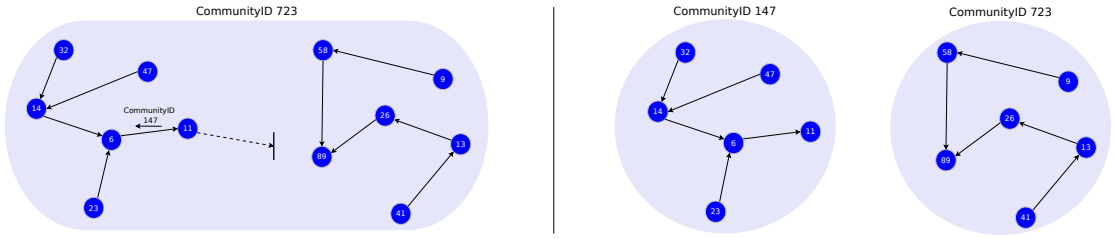


Figura 3.10: Separação de comunidades

3.4.5 Métrica Social

3.4.5.1 Objectivo

A métrica social é o conceito definido para avaliação do nó com base nas suas relações com os vizinhos, nas suas próprias características e nas particularidades da comunidade em que está inserido. Pretende-se desta forma suportar a capacidade de escolha nos nós, em vez do tipo de associação que é realizada

pelo padrão MAC do protocolo 802.11, no qual a associação é realizada ao primeiro nó detectado, independentemente das suas características. O principal objectivo é possibilitar a criação de comunidades nas quais as ligações estão optimizadas segundo determinados critérios, potenciando o desempenho global da rede. A aplicação deste conceito não está limitada à associação, sendo igualmente usado ao nível da eleição e do intervalo adaptativo entre *Beacons*.

Na essência da métrica social estão as relações existentes entre os nós num determinado momento e que podem potenciar não só as interacções directas como a disseminação mais rápida, eficiente e com menos interrupções da informação na rede. Como referido em [59], a correcta ponderação ao nível da associação pode potenciar o desempenho da rede na camada IP. Em [88, 89] mostra-se que é possível melhorar o desempenho de alguns parâmetros da rede recorrendo à eleição através de fórmulas ponderadas. No entanto, o facto dos cenários serem bastante dinâmicos será um dos desafios a enfrentar, nomeadamente ao nível da variação do valor da métrica social de cada nó, já que este depende do conhecimento e das características que cada nó tem sobre a rede em cada instante.

3.4.5.2 Fórmula ponderada

Em termos matemáticos, a métrica social de um nó é calculada de acordo com a equação 3.2:

$$SocialMetric = w_1 * FIN + w_2 * FQN + w_3 * CQE \tag{3.2}$$

onde w_1, w_2 e w_3 são os pesos dos diferentes parâmetros de entrada.

3.4.5.3 Parâmetros de entrada

***Friendship Indicator of Nodes* (FIN)** Este parâmetro tem como objectivo medir o grau médio de "amizade" entre um nó n_i e os seus vizinhos n_j. Como n_i entenda-se o nó central a verde da figura 3.11, n_j os nós azuis em volta do nó central e os vizinhos de n_j como os nós a vermelho. Como grau de "amizade" entenda-se:

- A relação entre o número de *Beacons* recebidos por n_i vindos dos seus vizinhos n_j e o número total de *Beacons* recebidos recentemente por n_i (*frequency factor*: $ff_{(n_j,n_i)}$);
- A relação entre o número de vizinhos dos vários n_j e o número de vizinhos de n_i.

Em termos matemáticos, podemos expressar o cálculo da FIN para cada n_i como:

$$FIN = \frac{\sum_j \left(ff_{(n_j,n_i)} + \frac{\frac{|PView_{n_j}|}{|PView_{n_i}|}}{N} \right)}{\sum n_j} \tag{3.3}$$

onde,

Terminologia	Significado
n_i	nó onde está a ser calculada a FIN
n_j	nó vizinho a n_i
$\|PView\|$	número de vizinhos de um dado nó
N	número total de nós em simulação

Tabela 3.1: Significado dos parâmetros da fórmula 3.3

A fórmula 3.3 encontra-se normalizada, tal que: $FIN \epsilon [0;1]$.

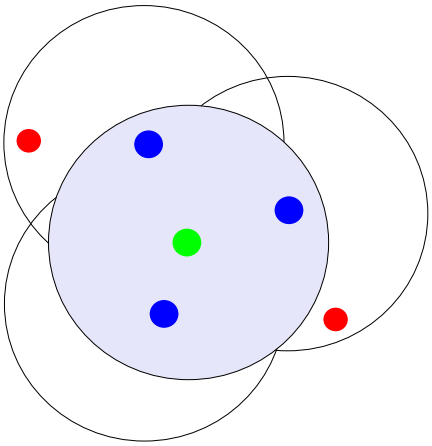


Figura 3.11: Interpretação visual da FIN

***Friendship Quality of Nodes* (FQN)** Este parâmetro tem como objectivo medir a qualidade da "amizade" entre um nó n_i e o nó n_a onde está associado. Como n_i entenda-se o nó a verde da figura 3.12 e n_a o nó cinza para onde aponta a seta (representação da associação). Como qualidade de "amizade" entenda-se:

- As características (processador, memória, disco, etc) e estado actual (bateria restante, interfaces livres, etc) do *hardware* de n_i;
- A estabilidade da ligação entre n_i e o nó n_a a que está associado.

Em termos matemáticos podemos expressar o cálculo da FQN para cada n_i como:

$$FQN = R.e^{-\triangle t} - \left(\frac{RecBeacons_{n_i}^{n_a}}{SentBeacons_{n_a}} \right)_{\Delta association} \tag{3.4}$$

onde,

Terminologia	Significado
R	recursos de *hardware* de n_i
$e^{-\triangle t}$	recursos disponíveis em função do tempo
n_i	nó onde está a ser calculada a FQN
n_a	nó ao qual n_i está associado
$RecBeacons_{n_i}^{n_a}$	número de *Beacons* recebidos por n_i vindos de n_a
$SentBeacons_{n_a}$	número de *Beacons* enviados por n_a
$\Delta association$	tempo de associação actual entre n_i e n_a

Tabela 3.2: Significado dos parâmetros da fórmula 3.4

A fórmula 3.4 encontra-se normalizada, tal que: $FQN \epsilon [0;1]$.

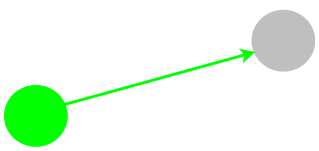


Figura 3.12: Interpretação visual da FQN

***Community Quality Estimation* (CQE)** Este parâmetro tem como objectivo avaliar o tamanho e a qualidade média das "amizades" da comunidade em que o nó n_i se encontra. Os vários nós n_i pertencentes a uma comunidade terão de estar interligados entre si através de associações, como representado na figura 3.13:

$$CQE = \frac{|Community_{n_i}|}{N} + \frac{\sum_c FQN_{n_c}}{\sum n_c} \tag{3.5}$$

onde,

Terminologia	Significado
n_i	nó onde está a ser calculada a CQE
$\|Community_{n_i}\|$	número de nós da comunidade de n_i
N	número total de nós em simulação
n_c	nó pertencente à mesma comunidade de n_i
FQN_{n_c}	valor da FQN de n_c

Tabela 3.3: Significado dos parâmetros da fórmula 3.5

A fórmula 3.5 encontra-se normalizada, tal que: $CQE \epsilon [0; 1]$.

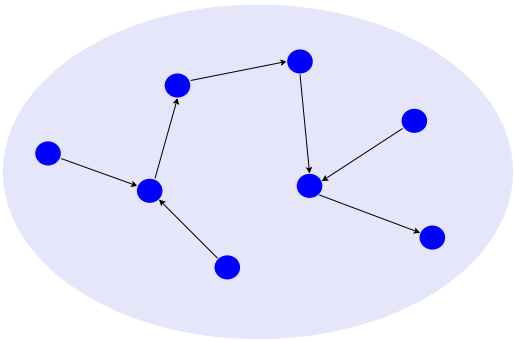


Figura 3.13: Interpretação visual da CQE

3.4.5.4 Intervalo adaptativo entre *Beacons*

Na presente solução foi também analisado o impacto que o intervalo entre *Beacons* tem na rede. Para tal implementou-se um intervalo adaptativo baseado em relações entre parâmetros da métrica social, comparando-se estes resultados com os obtidos com intervalo fixo.

Quanto maior for o número de vizinhos de um determinado nó, maior a probabilidade de este já ser conhecido (i.e. alguém já ter recebido um *Beacon* seu) e de o seu ID já estar a ser propagado pela comunidade. Ainda assim, se a comunidade for de dimensões consideráveis, a disseminação do seu ID demorará mais tempo a chegar a todos os nós. Neste caso, tendo em conta que o nó possui mobilidade e que por isso pode atravessar a comunidade e transmitir o seu ID, o intervalo entre *Beacons* deverá ser mais curto.

Esta primeira análise mais simples serve como base para o desenvolvimento do raciocínio, sabendo à partida que muitos outros factores podiam entrar em consideração para o cálculo do intervalo adaptativo. Assim, foram reaproveitados dois dos parâmetros de entrada da métrica social, uma vez que fornecem dados directamente relacionáveis com os factores apresentados anteriormente mas de forma mais robusta e completa.

Uma relação possível é o intervalo entre *Beacons* ser proporcional à FIN e inversamente proporcional à CQE. Uma vez que, por análise experimental, o valor médio da FIN é bastante inferior ao da CQE, optou-se por adicionar um peso a esta de forma a balancear o resultado final:

$$BeaconInterval = \delta \left(\frac{FIN}{CQE} \right) \tag{3.6}$$

Devido à oscilação do valor dos parâmetros no início de simulação, correspondente à aquisição inicial de conhecimento, definiu-se uma zona desde os 0s aos 0.5s de simulação (correspondente a cinco vezes o intervalo por defeito entre *Beacons*), na qual a frequência de *Beacons* se mantém fixa, para que a relação 3.6 não oscile bruscamente. Ainda assim, por uma questão de segurança, o intervalo dinâmico pode variar apenas dentro de uma zona entre 0.1s e 1s. Através de análise experimental, obteve-se um valor de δ que resulta em valores satisfatórios do intervalo adaptativo entre *Beacons* independentemente do número de nós em simulação (e.g. $\delta = 2$).

É necessário ter em atenção que a alteração deste intervalo implicará problemas cruciais no desempenho da rede se aplicada sem qualquer alteração ao nível estrutural do nó, uma vez que:

- a monitorização da associação é feita através de um intervalo máximo esperado entre *Beacons* a serem recebidos vindos desse mesmo nó. Caso o nó altere o seu intervalo para um valor muito elevado, a associação pode ser dada como perdida quando na realidade isso não aconteceu;
- a parcela da FQN que mede a estabilidade da ligação é calculada com base nos *Beacons* que supostamente devem ser recebidos vindos do nó onde está associado. Caso o nó altere o seu intervalo para um valor muito elevado, a estabilidade pode ser subavaliada quando na verdade se mantém na mesma.

Como forma de contornar os problemas expostos, cada nó terá de actualizar o intervalo esperado entre *Beacons* vindos do nó a que se encontra associado de cada vez que receba um pacote deste tipo cuja origem seja esse nó, de acordo com o *BeaconInterval* encapsulado. Como foi definida uma zona inicial de aquisição de informação e limites para o valor do intervalo dado pela relação 3.6, a evolução dos valores produzidos por esta tenderá para um valor médio dependente do cenário simulado, contribuindo para a não ocorrência de falsas detecções de perda de associação.

3.4.6 Complementos ao protocolo 802.11 MAC

Apesar de ter sido utilizado um ambiente de simulação, este apresenta-se bastante próximo de uma implementação real, nomeadamente ao nível das camadas protocolares. Neste trabalho houve a preocupação de tentar seguir com rigor esse compromisso, alterando o protocolo 802.11 MAC em pontos essenciais de modo a que, implementações futuras em *testbeds* reais não sejam impraticáveis com o protocolo em questão. Considerando o padrão IEEE referente a este protocolo [93] no modo *ad hoc*, foi necessário incluir alguns complementos como forma de cumprir os vários requisitos exigidos pelos mecanismos que se pretendem implementar:

- ao nível dos pacotes de gestão *Beacon* e *Association Response*, a introdução de novos IEs no campo *Frame Body*, inserido-os juntamente aos já existentes. Com as novas informações inseridas pretende-se possibilitar a cooperação e partilha de conhecimento entre as várias entidades da rede;

- ao nível da associação, a introdução de novas informações nos pacotes de gestão do protocolo 802.11 MAC permitirão realizar associações ponderadas entre nós. Pretende-se que, com a introdução de decisão ao nível local, a associação passe a ser baseada em determinados parâmetros e não de forma determinística ao primeiro nó ao alcance.

- ao nível do espaçamento temporal entre *Beacons*, passar do intervalo fixo e pré-estabelecido para um intervalo adaptativo, calculado dinamicamente através dos parâmetros da métrica social. Pretende-se desta forma garantir a gestão igualmente eficiente da rede com menor *overhead* de mensagens.

3.4.7 Desafios inerentes

A arquitectura dos mecanismos propostos para a gestão distribuída da rede traz inúmeros desafios, nomeadamente ao nível do sincronismo e consenso da informação entre as várias entidades. Sem a existência de nós centralizadores, torna-se difícil saber, em cada entidade, qual da informação recebida é efectivamente a mais próxima do contexto real da rede em cada instante. Estes desafios ocorrem devido ao raio de alcance limitado dos nós, que lhes permite apenas escutar o meio numa área limitada em seu redor. No entanto, é necessário que estes tenham uma visão mais abrangente da rede, o que é apenas alcançado através da colaboração entre as várias entidades. Por outro lado, é exigido que os nós sejam capazes de tomar decisões tendo apenas uma visão parcial da rede.

A estes podem ainda acrescentar-se os seguintes desafios com que o trabalho se deparou ao longo da sua execução:

- Inexistência de pacotes de gestão específicos para as funcionalidades acrescentadas: os mecanismos implementados para solucionar os desafios inerentes à gestão distribuída da rede foram alcançados mesmo sem a existência de pacotes de gestão específicos para essas funções. No presente trabalho, a base da colaboração ao nível MAC assenta sobretudo na informação inserida, através de novos IEs, nos pacotes de gestão do padrão 802.11 MAC. Pode afirmar-se que o *Beacon* é o pacote de gestão fundamental devido à sua periodicidade, uma vez que todos os outros são despoletados por eventos;

- Escassez de informação ao nível MAC: a definição e implementação do conceito da métrica social foi outro dos desafios. O objectivo final seria construir comunidades cujas ligações entre nós fossem adaptativas e optimizadas de acordo com determinados parâmetros. Contudo, na camada MAC, o nível de informação é limitado, sendo ainda mais escasso no ambiente de simulação. Assim, teve de haver alguma ponderação nos parâmetros para o cálculo da métrica social, bem como a implementação de mecanismos para a recolha da informação necessária para os mesmos e ainda a simulação dos parâmetros de *hardware* de cada entidade;

- Dinâmismo dos cenários: um dos pontos críticos dos cenários simulados é a constante mudança nas condições do cenário, devido não só à mobilidade dos nós como à constante recolha de informação. A justificação para os cenários não convergirem, isto é, não tenderem para um ponto estável por mais que se prolongue a simulação, está relacionada com o facto de existirem sempre perdas/aquisições de associação, entrada/saída de nós nas comunidades, etc. Assim sendo, a possibilidade de ocorrência destes eventos implica a partilha de informação e a actualização constante do conhecimento que cada nó tem da rede;

- Detecção da perda de associação: um nó tem de ter a capacidade de monitorizar o estado da ligação de associação, isto é, saber se o nó a que está associado ainda se encontra dentro do raio de alcance. Quando os nós estiverem fora do alcance um do outro serão incapazes de fazer a sinalização, indicando a quebra da ligação de associação. Para contornar este problema foi usado um *watchdog timer*[2] com escala igual a cinco vezes o intervalo entre *Beacons* indicado por esse nó de associação. A cada recepção de *Beacons* vindos desse nó o *timer* é reiniciado. Caso atinga o fim de escala, o nó detecta a perda de ligação, permitindo-lhe iniciar uma nova, ou seja, a máquina de estados é reiniciada;

- Conceito/abstracção de comunidade (geração, aceitação, rejeição e coerência): a agregação dos nós em comunidades de forma a criar partições na rede, visa permitir que cada entidade não necessite de manter conhecimento de toda a rede mas apenas de pequenos blocos desta. Porém, este conceito é uma abstracção lógica, na medida em que não existe uma circunscrição física nem nenhuma entidade centralizadora que represente a comunidade em questão. Sendo assim, numa gestão distribuída, cada entidade será responsável por conhecer e actualizar o identificador da comunidade em que se encontra. Para responder a este desafio, ao ser estabelecida uma associação, o nó requerente deve adoptar o identificador da comunidade em que se está a inserir. Se ocorrer a perda de associação, o identificador da comunidade deve ser eliminado do repositório local do nó. Notar ainda que uma nova comunidade é gerada quando, pelo menos, dois nós se associam. Caso haja separação ou união de comunidades, a coerência do identificador é garantida, dado que os nós aceitaram o novo identificador recebido pela ligação de associação;

- Conhecimento do número de nós numa comunidade: devido aos parâmetros da métrica social, nomeadamente da CQE, é exigido que se saiba qual o número de nós de uma determinada comunidade. Pelo facto de não existir uma entidade controladora central onde os nós de cada comunidade se registem, é necessária a colaboração entre as várias entidades de modo a partilharem o seu conhecimento. No problema considerado, esta cooperação deve ocorrer na partilha do conhecimento do número de nós que, em determinado instante, cada entidade sabe que existem na sua comunidade;

- Conhecimento da qualidade de uma comunidade: pelo mesmo motivo referido anteriormente, é também exigido que se saiba a qualidade de uma determinada comunidade. O desafio é basicamente o mesmo que foi referido na contagem de nós da comunidade, embora neste caso seja necessário trocar informação sobre a qualidade, ou seja, a FQN de cada um dos nós que pertence à comunidade. Desta forma, localmente poderá ser determinada a qualidade média que cada entidade tem da sua comunidade.

[2] é um mecanismo genérico que permite reactivar o sistema ou executar um determinado mecanismo de correcção quando o fim de escala é atingido. Para isso é necessário executar sucessivos *resets* ao *timer* sempre que ocorrem determinados eventos. No contexto actual, o *timer* está programado para reiniciar a máquina de estados do nó, ou seja, fazer com que este inicie um novo processo de associação.

3.5 Redes com fios

3.5.1 Mecanismos e sua função

O *bootstrapping*, a descoberta e a eleição são mecanismos essenciais para garantir a troca de informação base que permita assegurar a gestão distribuída da rede.

O *bootstrapping* corresponde à fase de arranque de uma nova entidade na rede, na qual são conhecidas as suas características estáticas (e.g. recursos locais). A descoberta refere-se ao processo contínuo de recolha de informação, não só para descoberta de novas entidades, como para a actualização da informação das já conhecidas. Por fim, a eleição é o processo através do qual é escolhido o nó com as melhores características, sendo-lhe atribuídas funcionalidades e tarefas específicas (e.g. disseminação de informações e decisões de gestão).

Os mecanismos referidos têm por base uma extensão da técnica *Hide & Seek* [16] e apresentam uma forte interligação, uma vez que a descoberta e a eleição se encontram num ciclo repetitivo, dado que o conhecimento de novas entidades ou a actualização da informação pode levar à eleição de um novo líder.

Na figura 3.14 é apresentada a interligação entre os vários mecanismos implementados, bem como a interacção entre entidades da rede através da troca de mensagens específicas e as acções daí resultantes. É possível verificar que inicialmente todas as entidades configuram os seus identificadores únicos (compostos pelo endereço MAC e um número aleatório). A partir desse momento, os repositórios locais são inicializados e é configurado o intervalo inicial entre pacotes *Hello*, passando as entidades para a fase da troca de pacotes. O papel desempenhado em cada instante por cada entidade é dinâmico, ou seja, pode ser alterado conforme as circunstâncias da rede (e.g. contacto de outras entidades).

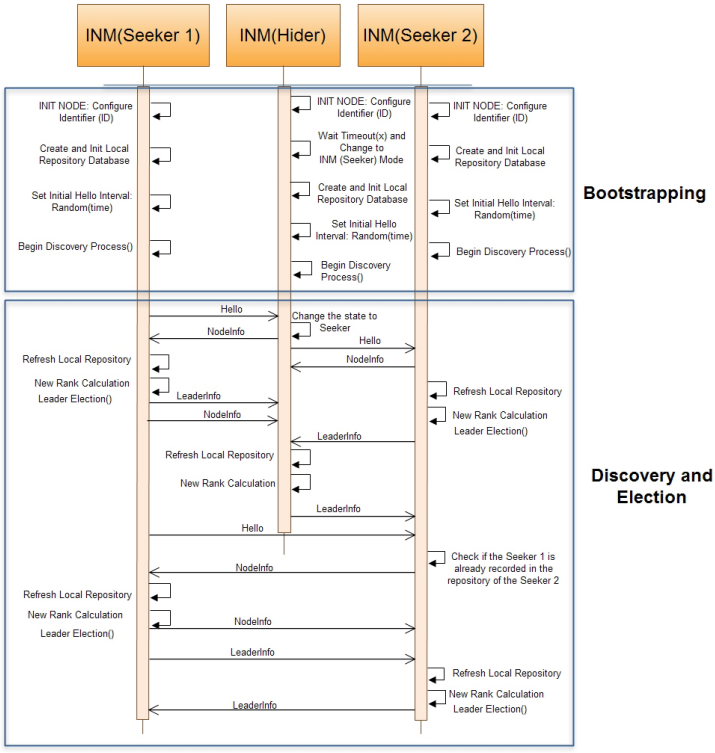


Figura 3.14: Processo de sinalização envolvida nos vários mecanismos do *INM-Discovery* [22]

3.5.1.1 *Bootstrapping*

Este processo corresponde à configuração inicial de uma nova entidade da rede, através do qual são inicializados, por exemplo, os repositórios e os identificadores locais. Estes identificadores internos são gerados localmente e compostos pelo endereço MAC e por um número aleatório (e.g. 00:35:FF:48:B3:12-349870).

Pode ser definido um de dois papéis para cada entidade: *INM_Seeker* e *INM_Hider*. Se inicialmente for configurada como *INM_Seeker,* então o processo de descoberta inicia-se de imediato. Caso a entidade seja configurada como *INM_Hider*, e se não for entretanto contactada por outra entidade, então terá de aguardar um tempo aleatório (e.g. 1s a 60s) até poder mudar o seu papel para *INM_Seeker*. Caso já tenha sido contactada, então o seu papel é automaticamente alterado e o processo continua.

É visível na figura 3.14 os dois tipos de entidades (*INM_Seeker* e *INM_Hider*), bem como a configuração do identificador único, a inicialização dos repositórios locais e a determinação o intervalo inicial entre pacotes *Hello* durante a fase de *bootstrapping*.

3.5.1.2 Descoberta

Para o mecanismo de descoberta implementado, o *INM_Seeker* enviará pacotes *Hello multicast* para a sua vizinhança de maneira a recolher informação acerca dos seus vizinhos. Como forma de definir a profundidade destes pacotes na vizinhança, o campo *Time-To-Live* (TTL) pode ser especificado. Em seguida, a informação recolhida é armazenada em repositórios locais, designados *Partial Views*. O outro tipo de entidade, designado *INM_Hider*, adopta uma postura passiva, aguardando o contacto por parte das outras entidades e, após ser detectado, muda o seu papel para *INM_Seeker,* contribuíndo activamente no processo de descoberta. Cada entidade possui os seus repositórios locais, designados *Partial Views*, onde armazena a informação recolhida através dos seus vizinhos. Nestes podem ser guardadas informações como IP origem e destino do pacote *Hello*, % livre de RAM e CPU, número de interfaces de rede, endereço MAC, *Round-Trip-Time* (RTT), largura de banda das ligações.

De acordo com a figura 3.15, as entidades *INM_Seekers* trocam pacotes *Hello* usando TTL de 1 (e.g. para evitar ciclos longos de mensagens) como forma de realizar o processo de descoberta, sendo o papel de cada entidade dinâmico. Note-se que, nesta abordagem, cada entidade não necessita de conhecer toda a rede, diminuindo assim a quantidade de informação necessária a guardar localmente. É importante referir também que o intervalo entre pacotes *Hello* é adaptativo, variando conforme o número de *INM_Seekers* na *Partial View* de cada entidade. Este repositório local contém informações como identificadores (e.g. Endereço MAC e identificador interno), endereços IP (e.g. origem e destino do *Hello* recebido) e papel (*INM_Seeker* e *INM_Hider*).

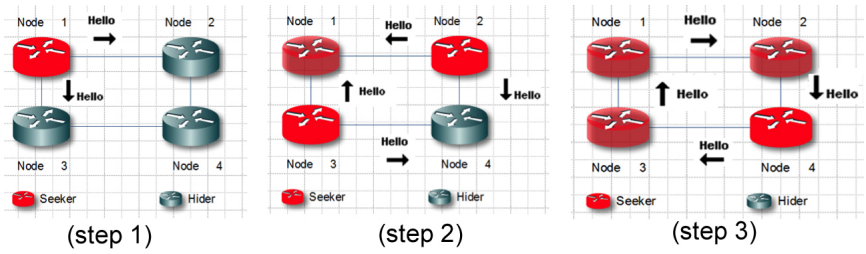


Figura 3.15: Etapas genéricas da descoberta pelo *INM-Discovery* [22]

Os pacotes *Hello* enviados pelos *INM_Seekers* contêm os campos (*MsgType, Hello ID*), sendo enviada a resposta da entidade contactada no pacote *NodeInfo,* que contém os campos (*MsgType, Hello ID, nNodes, Interface, pFreeRam, pFreeCPU, bandwidth, nInterfaces, type*). O impacto do mecanismo em termos de *overhead* de mensagens é reduzido devido à colaboração entre *INM_Seekers*, nomeadamente através das suas *Partial Views*.

Na comunicação entre *INM_Seekers* é verificado se aquela entidade já foi previamente contactada (i.e. está presente na *Partial View*), sendo em seguida a informação sincronizada entre os repositórios. É importante referir que, quando se formam grupos *multicast*, as mensagens são propagadas de imediato aos restantes *INM_Seekers*, de forma a que a informação entre eles esteja sempre actualizada. Ao nível da comunicação, o protocolo tem suporte para IPv4 e IPv6, bem como para mensagens de grupos *multicast*.

Na zona intermédia da figura 3.14 é possível observar o envio de um pacote *Hello* por parte do *INM_Seeker-1*, e após ser recebido pelo *INM_Hider*, o seu papel é alterado para *INM_Seeker*, passando a colaborar activamente no processo de descoberta. As respostas são enviadas na mensagem *NodeInfo*, nas quais é encapsulado o conhecimento de cada entidade, permitindo a colaboração e a partilha de informações contidas nos repositórios locais.

3.5.1.3 Eleição

Após o nó recolher informação sobre a sua vizinhança, inicia-se o processo de eleição, através do qual é escolhida a entidade considerada ser a melhor. Para iniciar este processo é necessário o cálculo do *ranking* dos vários nós, onde se tem em conta várias características previamente recolhidas dos diversos nós:

$$Rank\,(n) = (w_1 * Bw + w_2 * (Free_{RAM} + Free_{CPU} + RTT) + w_3 * (N_n + N_i)) \tag{3.7}$$

onde,

Designação	Significado
Bw	Largura de banda (Mbps)
$Free_{RAM}$	RAM livre (%)
$Free_{CPU}$	CPU livre (%)
RTT	*Round-Trip-Time* (ms)
N_n	Número de nós na vizinhança
N_i	Número de interfaces
w_1, w_2, w_3	Pesos das várias parcelas

Tabela 3.4: Significado das variáveis da equação 3.7

De acordo com a equação 3.7, o nó que possuir maior *ranking* é considerado líder localmente. De forma a evitar o problema de consenso de sistemas distribuídos, cada entidade envia o seu líder e *ranking* respectivo para a entidade com maior ID e número de interfaces. Esta entidade será a responsável por reunir todas as eleições locais e definir qual dos nós eleitos é efectivamente o melhor líder, notificando em seguida todos os nós sobre essa decisão. Este processo é contínuo, uma vez que a descoberta e a actualização de informação podem conduzir a novos líderes.

Na figura 3.14 é possível verificar o envio de um pacotes *LeaderInfo* onde estará colocado o ID do nó com maior *ranking*, de acordo com a equação 3.7.

3.5.2 Desafios inerentes

Os principais desafios associados à solução proposta para as redes com fios são:

- Sincronização e consistência da informação entre diferentes entidades;
- Processo de descoberta realizado de forma distribuída exige colaboração entre os vários elementos de rede;
- A escalabilidade da solução deve ser garantida, implicando que:
  - Disseminação da informação seja realizada recorrendo a um número reduzido de mensagens;
  - Convergência da informação de descoberta entre as várias entidades seja realizada num intervalo de tempo reduzido.

3.6 Conclusões

Neste capítulo foi descrita de forma conceptual a arquitectura geral e as soluções para os diferentes mecanismos propostos de um ponto de vista macroscópico, apresentando-se no início de cada secção um diagrama geral com a interacção entre entidades da rede e a interligação entre os principais mecanismos e funcionalidades desenvolvidos.

Concretamente nas redes sem fios, são descritas as funções dos principais mecanismos implementados, bem como alguns processos e conceitos associados aos mesmos. São apresentados alguns complementos incluídos ao protocolo 802.11 MAC, necessários para que a comunicação entre entidades conduza a uma gestão distribuída da rede. É também introduzido o conceito de métrica social e os objectivos que se pretendem alcançar com este. São ainda descritos alguns desafios inerentes ao processo de gestão autonómica da rede e aos mecanismos de cooperação distribuída apresentados.

Nas redes com fios foram abordados os mecanismos de *bootstrapping*, de descoberta baseado na técnica *Hide & Seek* e de eleição de forma a introduzir os conceitos abordados em detalhe na implementação dos mesmos.

Capítulo 4

Implementação

4.1 Introdução

Com o intuito de avaliar os mecanismos propostos no capítulo 3, procedeu-se à implementação das soluções correspondentes a redes sem fios em ambiente de simulação e às correspondentes a redes com fios em *testbed* virtual.

A secção 4.2 é relativa às redes sem fios, onde se apresenta em detalhe o protocolo 802.11 MAC e PHY, ao nível do simulador NS-3 e pormenores relativos à implementação das soluções anteriormente apresentadas.

A secção 4.3 trata das redes com fios, na qual são apresentados detalhes relativos à implementação dos mecanismos de *bootstrapping*, descoberta e eleição na *testbed* virtual.

Em 4.4 são abordados alguns desafios enfrentados especificamente ao nível da implementação.

Em 4.5 é elaborado um breve resumo do capítulo.

4.2 Redes sem fios

4.2.1 Protocolo 802.11 MAC e PHY em NS-3

4.2.1.1 Visão geral

O simulador de redes NS-3 oferece um conjunto de modelos bem definidos de modo a obter-se uma implementação precisa ao nível físico e MAC segundo o protocolo IEEE 802.11.

Na versão 3.9 [109, 110], os modelos implementados podem subdividir-se em 4 níveis:

- Modelos do nível físico (PHY *layer models*);
- Modelos MAC de baixo nível que implementam DCF e EDCAF (MAC *low level models*);
- Modelos MAC de alto nível: *Ad hoc*, STA e AP (MAC *high level models*);
- Algoritmos de controlo de taxas de transmissão usados pelos modelos MAC de baixo nível (*Rate control algorithms*).

4.2.1.2 Camada física

Ao nível físico, o modelo PHY é baseado na implementação descrita em [111]. O modelo físico impõe uma única interface de rádio que opera em modo *half-duplex*[1]. Possui ainda uma máquina de estados bem definida, cujos estados possíveis são:

- *IDLE* - quando não se encontra ocupada;
- *CCA_BUSY* - quando detecta o meio ocupado através do mecanismo *Clear Channel Assessment* (CCA);
- *TX* - quando se encontra a enviar um pacote;
- *RX* - quando está numa operação de recepção de um pacote;
- *SWITCHING* - quando se encontra a trocar para outro canal.

O modelo PHY depende das perdas e atrasos introduzidos pelo canal de comunicação, modelados pelas classes *PropagationLossModel* e *PropagationDelayModel* respectivamente.

4.2.1.3 Camada MAC de baixo nível

Os modelos MAC de baixo nível podem ser separados em três componentes:

- *MacLow* que trata da troca de mensagens RTS, CTS, DATA e ACK;
- *DcfManager* e *DcfState* que implementam a DCF para acesso ao meio;
- *DcaTxOp* e *EdcaTxOpN* que gere as filas de espera, fragmentação e retransmissão de pacotes. As classes MAC de alto nível sem QoS usam a *DcaTxOp*, enquanto aquelas que têm QoS activo usam *EdcaTxOpN*.

4.2.1.4 Camada MAC de alto nível

Nos modelos MAC de alto nível podemos encontrar os três modelos correspondentes às duas topologias sem fios mais comuns:

- Infra-estrutura: AP e STA;
- *Ad hoc*: STA num IBSS, também designada de rede *ad hoc*.

4.2.1.5 Interligação de camadas ao nível 2

A tentativa de aproximação fiel à realidade do simulador NS-3 é conseguida, não só pelo uso de modelos rigorosos, como pela interligação (recorrendo por vezes ao uso de *callbacks*[2] e *smart pointers*[3]) entre as várias camadas.

[1] Um dos primeiros problemas encontrados foi a não recepção de pacotes pelos nós. Apesar de estar referenciado que existe, por defeito, o mecanismo de *backoff time*, foi necessário inserir um atraso entre o início da transmissão dos vários nós, de modo a evitar as designadas *startup storms*.

[2] Mecanismo de programação cujo objectivo é permitir que num dado módulo seja chamada uma determinada função/método sem que haja uma dependência específica entre o módulo onde a função é chamada e aquele onde está definida.

[3] Tipo abstrato de dados que permite, entre outras funcionalidades, gestão de memória mais eficiente através da automática libertação de recursos.

Em ambientes sem fios, a interligação entre as várias camadas ao nível 2 pode ser ilustrada pelo fluxograma da figura 4.1:

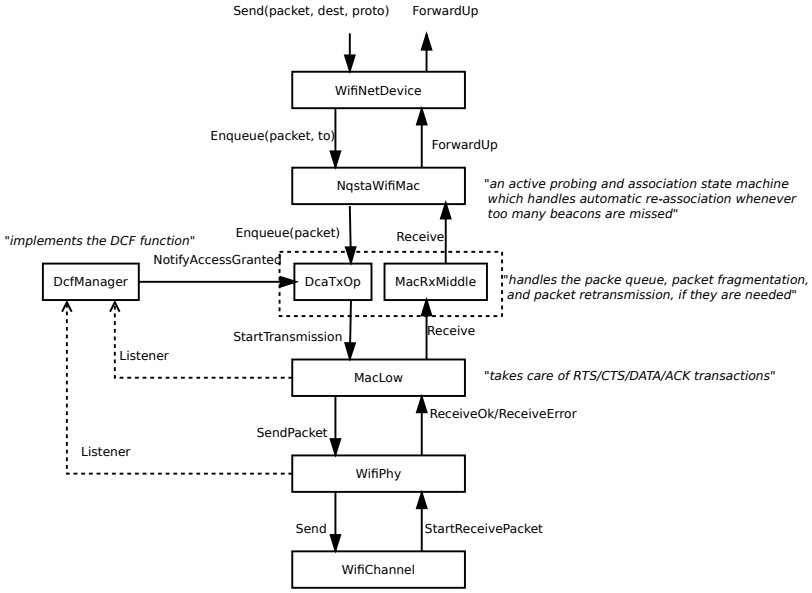


Figura 4.1: Sub-camadas atravessadas ao nível 2 para transmissão e recepção de pacotes em redes sem fios [109]

Assim, quando ocorre o envio de um pacote, é notificado o controlador do dispositivo (num exemplo real corresponderia ao *device driver* da interface de rede) que coloca o pacote na fila de espera do nó sem fios.

O gestor DCF encarrega-se de retransmitir o pacote se o envio não for bem sucedido e de gerir a fila de espera do nó.

Assim que seja possível iniciar a transmissão, o pacote é enviado à camada MAC de baixo nível, que tratará das negociações RTS, CTS, DATA, ACK.

Ultrapassada esta fase, o pacote é entregue à camada física (corresponde à interface de rede) que o coloca no canal de comunicação, cujos atrasos e perdas podem ser modelados (num exemplo real, o meio de propagação seria o ar).

Um processo análogo e simétrico ocorre quando se dá a recepção de um pacote.

4.2.1.6 Algoritmos de controlo de taxas de transmissão

O simulador possui um vasto conjunto de algoritmos para controlo de taxas de transmissão:

- ***ArfWifiManager*:** implementa o algoritmo *Auto Rate Fallback* (ARF) descrito inicialmente em [112]. Esta implementação difere da descrição inicial pois usa um temporizador baseado em pacotes (*packet-based timer*) em vez do original baseado em tempo (*time-based timer*);
- ***AarfWifiManager*:** implementa o algoritmo *Adaptive Auto Rate Fallback* (AARF), inicialmente descrito em [113];
- ***IdealWifiManager*:** implementa um controlo de taxa "ideal" semelhante à ideia do algoritmo *Receiver-Based AutoRate* (RBAR) descrito em [114]. Todas as estações guardam o *Signal-to-Noise Ratio* (SNR) de todos os pacotes recebidos e enviam de volta este SNR ao transmissor

original através de um mecanismo fora-de-banda (*out-of-band mechanism*). Cada transmissor guarda o último SNR enviado de volta pelo receptor e usa-o para conseguir um modo de transmissão baseado num conjunto de limites de SNR construídos através de um valor de *Bit Error Rate* (BER) pretendido e de um modo de transmissão específico de acordo com a relação SNR/BER;

- ***OnoeWifiManager*:** é muito popular por ser usado como algoritmo de controlo por defeito do *driver "madwifi"* [115]. A sua implementação foi desenvolvida por Atsushi Onoe;

- ***AmrrWifiManager*:** inicialmente descrito em [113];

- ***CaraWifiManager*:** implementa o algoritmo *Collision-Aware Rate Adaptation* (CARA) de[116], originalmente implementado por Federico Maguolo numa versão protótipo do ns-3;

- ***AarfcdWifiManager*:** inicialmente descrito em [117], a sua implementação em ns-3 foi feita por Federico Maguolo numa versão protótipo do ns-3.

No presente trabalho, foi utilizado o modelo simples *ConstantRateWifiManager* que usa taxas de transmissão constantes para o envio de pacotes de dados e controlo, uma vez que o que se pretende implementar é ao nível da troca de pacotes de gestão da camada MAC.

4.2.2 Solução implementada

4.2.2.1 Modelos existentes

Em termos de modelos já existentes no simulador utilizado e com possibilidades de reaproveitamento, destacam-se os modelos da designada camada MAC de alto nível.

O mais simples destes modelos existentes é o nó *ad hoc* uma vez que não efectua qualquer geração de pacotes de gestão de acordo com o padrão 802.11 MAC, mas sim uma troca de simples *sockets*. Também não é possível extrair uma máquina de estados, ainda que lógica, devido à simplicidade da troca de pacotes entre nós deste modelo. Facilmente se percebe que, apesar de se inserir no modo de comunicação pretendido (*ad hoc*), no contexto deste trabalho o modelo existente não terá grande utilidade.

Os modelos do modo infra-estrutura são bastante mais complexos. A classe STA é capaz de implementar *active probing*[4] e possui uma máquina de estados bem definida que efectua re-associação quando deixa de receber um determinado número *Beacons*. Ainda assim, esta recepção é independente do nó origem, que neste caso deveria ser o AP associado. A classe AP gera *Beacons* periodicamente e aceita quaisquer tentativas de associação por parte das STAs. Como facilmente se percebe, não existe qualquer tipo de comunicação entre STAs, apenas destas com os APs, demonstrando que o modo infra-estrutura é centralizado.

A abordagem para desenvolvimento do nosso modelo *ad hoc* foi, de um modo geral, englobar as funcionalidade tanto do AP como da STA num único modelo e efectuar a comunicação e associação entre nós deste tipo, tornando a arquitectura da rede distribuída.

[4] Apesar de ter sido reportado por nós um *bug* relativo a esta funcionalidade [118]

4.2.2.2 Características gerais do modelo *ad hoc* implementado

As funcionalidades gerais do modelo referente ao nó *ad hoc* implementado são:

- Capacidade de comunicação entre nós sem recurso a entidades centralizadas;
- Envio de *Beacons* em *broadcast* periodicamente;
- Escolha entre modo de *scanning* activo ou passivo;
- Escolha entre padrão de mobilidade aleatória ou estática;
- Escolha entre modo de associação indiscriminada ou baseada na métrica social;
- Detecção, quando associado, do limite máximo de *Beacons* perdidos, valor para o qual inicia automaticamente um novo processo de associação.

4.2.2.3 Canal de Comunicação

Por defeito, o ambiente de simulação não impõe nenhuma restrição ao nível do canal de comunicação, ou seja, considera-o ideal. Isto implica que quaisquer dois nós conseguem contactar directamente, independentemente da distância relativa entre ambos. No contexto da solução implementada significaria que a colaboração de conhecimento entre nós seria redundante, já que qualquer nó receberia informação de todos os outros, eliminando todas as dificuldades inerentes à sincronização e consenso da informação. Assim, de modo a tornar a simulação o mais semelhante possível ao ambiente real, configurou-se o canal de comunicação quer em termos de atrasos, quer em termos de perdas de propagação.

O atraso do canal é modulado pela classe *PropagationDelayModel*, da qual herdam os modelos *ConstantSpeedPropagationDelayModel* e *RandomPropagationDelayModel*. Devido à inexistência de modelos baseados, por exemplo, em coordenadas geográficas, optou-se pelo modelo de propagação com velocidade constante dado que o raio de alcance dos nós é reduzido, não impondo atrasos muito significativos em vez do que poderia acontecer ao se impor um atraso aleatório em todos os pacotes.

As perdas são modeladas pela classe *PropagationLossModel*, da qual herdam os modelos *RandomPropagationLossModel*, *FriisPropagationLossModel* e *PathLossPropagationLossModel*. A escolha recaiu sobre o modelo que segue a fórmula de Friis, dado que esta entra em consideração com diversos parâmetros importantes de telecomunicações. Pretende-se desta forma limitar o raio máximo até ao qual um nó consegue comunicar com outros.

Para definir o raio de cobertura das interfaces de rádio é necessário definir o limiar de detecção dos receptores, através da manipulação matemática da fórmula de Friis:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{4.1}$$

onde P_r e P_t são as potências de recepção e transmissão respectivamente (em Watts), G_r e G_t são os ganhos das antenas do receptor e transmissor respectivamente (em unidades lineares e não em decibéis), λ é o comprimento de onda e d a distância entre as antenas (ambas em metros).

Utilizando a equação anterior, é possível calcular o valor da potência que deve ser detectada a uma determinada distância d. Os parâmetros utilizados na configuração da interface rádio são:

- $P_t = 50mW = 16.990\,dBm$
- $G_t = G_t = 0dB = 1$
- $\lambda = \frac{c}{f} = \frac{3*10^8}{2.4*10^9} = 0.125\,m$
- $d = 100\,m$

Logo, pela equação 4.1:

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi d}\right)^2 \cong 4.947 * 10^{-7} mW \cong -63.056\,dBm$$

Assim, o limiar de detecção dos nós da rede deve ser configurado para apenas receber pacotes cuja potência do sinal seja superior a -63.056 dBm. O valor para o limiar de detecção encontra-se na gama comum das potências de sinal recebidas em redes sem fios das variantes 802.11 (-60 a -80 dBm [119]). Isto significa que o raio de alcance é adequado à realidade.

Porém, num cenário real poderia ser necessário encontrar uma modulação digital adequada, cujo ganho de processamento permitisse atingir as especificações de fidelidade de transmissão (SNR em função da probabilidade de erro), algo que no ambiente de simulação utilizado não acontece (figura 4.2).

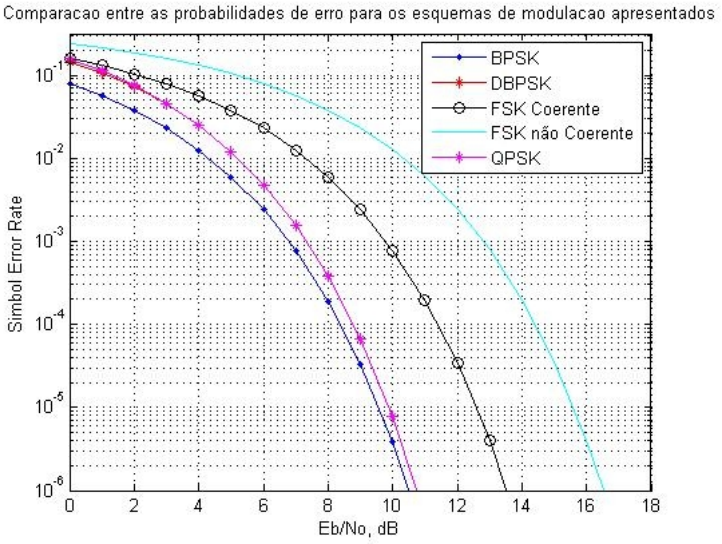


Figura 4.2: Relação entre a probabilidade de erro e a SNR para diferentes tipos de modelação [120]

4.2.2.4 Máquina de estados finitos

O modelo do nó *ad hoc* implementado possui uma máquina de estados interna (figura 4.3), da qual fazem parte os seguintes estados:

- ***Wait Probe Response* / *Wait Beacon*:** significa que já foi previamente enviado um *Probe Request* (modo *active scanning*), portanto é necessário esperar por uma resposta ou pelo *timeout* para reenviar o pacote. Este estado é partilhado pelo modo *passive scanning*, significando que o nó não está ainda associado e portanto aguarda a recepção de *Beacons* para iniciar o processo.

- ***Wait Association Response*:** já foi previamente enviado um *Association Request*, portanto é necessário esperar por uma resposta (concedendo ou recusando a associação) ou pelo *timeout* para reenviar o pacote.

- ***Refused*:** foi recebido um *Association Response* recusando o pedido de associação. Será iniciado um novo processo de associação assim que possível.

- ***Associated*:** foi recebido um *Association Response* concedendo a associação requerida. É gravado o endereço do nó a que se associa e o intervalo entre *Beacons* é alargado.

- ***Beacon Missed*:** caso seja detectada a perda de um número pré-determinado de *Beacons* do nó a que está associado, a ligação entre ambos é eliminada. O motivo para que tal possa acontecer está relacionado com a mobilidade dos nós. Será iniciado um novo processo de associação assim que um nó entre no raio de alcance daquele cuja associação foi perdida.

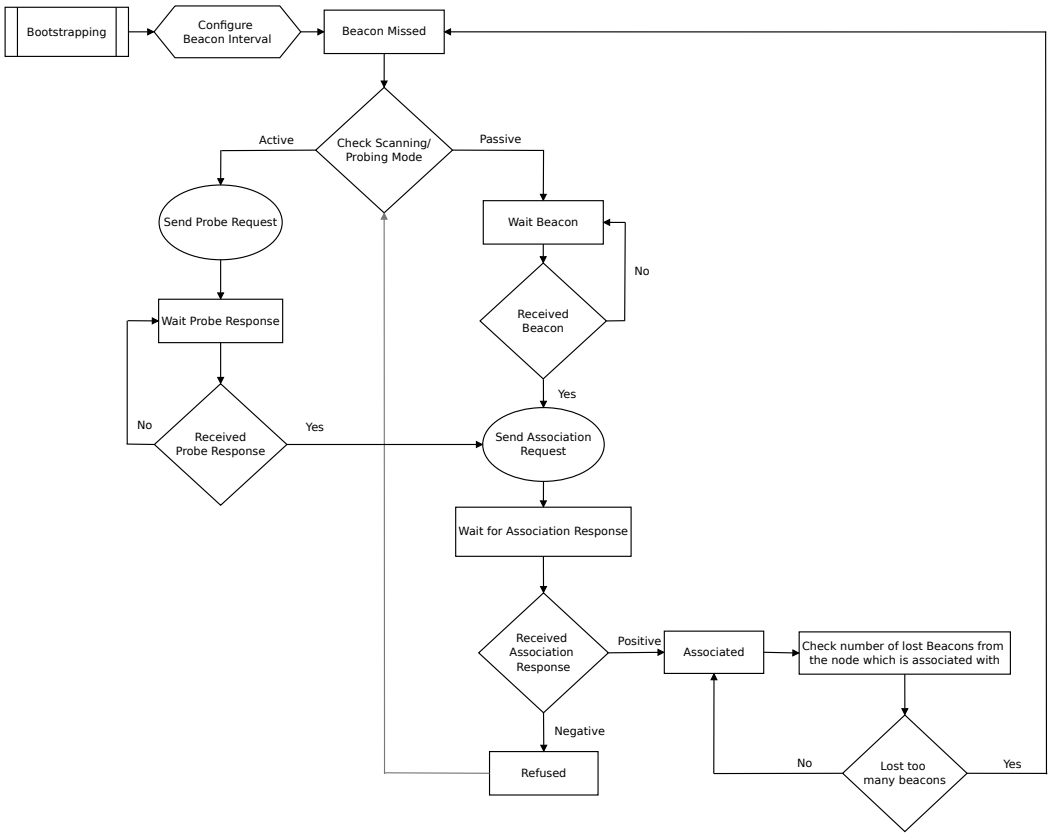


Figura 4.3: FSM implementada no nó ad hoc

4.2.2.5 *Bootstrapping*

O *bootstrapping* pode ser decomposto nas várias funções de configuração que o constituem ao nível MAC. Muitas destas são herdadas da classe *WifiMac* existente no simulador NS-3, nomeadamente as de configurações *standard* das camadas PHY e MAC de acordo com o protocolo 802.11.

Método	Descrição
Configuração de parâmetros temporais relativos ao gestor DCF	
`SetSifs (Time sifs)`	Define a duração do intervalo SIFS
`SetEifsNoDifs (Time eifsNoDifs)`	Define a duração do intervalo EIFS subtraindo a duração do intervalo DIFS
`SetSlot (Time slot)`	Define a duração da janela de transmissão
Configurações características do nó	
`SetAddress (Mac48Address address)`	Atribui o endereço MAC à entidade em questão
`SetSsid (Ssid ssid)`	Atribui o SSID à entidade em questão
`SetBssid (Mac48Address bssid)`	Atribui o BSSID à entidade em questão
`SetNodeId (uint32_t id)`	Define o identificador único da entidade
`SetLiderId (uint32_t id)`	Define o identificador do líder
`SetActiveProbing (bool enable)`	Define o modo de *scanning* (activo ou passivo)
`SetBeaconInterval (time interval)`	Atribui o intervalo de tempo entre *Beacons*
`SetBeaconGeneration (bool enable)`	Define se o nó tem capacidade de enviar *Beacons*
`SetSocialBasedAssoc (bool enable)`	Define o modo de associação (*Standard* ou *Social-based*)
`SetSocialMetric (uint32_t value)`	Define o valor da métrica social
`SetSocialWeights (uint8_t w1, uint8_t w2, uint8_t w3)`	Define o valor dos pesos dos parâmetros da métrica social
`SetMaxMissedBeacons (uint32_t missed)`	Define o número máximo de *Beacons* perdidos até se considerar que ocorreu quebra de associação
`SetProbeReqTimeout (Time timeout)`	Atribui o tempo máximo que um nó aguarda por um *Probe Response* até reenviar um *Probe Request*
`SetAssocReqTimeout (Time timeout)`	Atribui o tempo máximo que um nó aguarda por um *Association Response* até reenviar um *Association Request*
Termina a configuração da camada física, configurando as filas de espera da DCF de acordo com o padrão 802.11 escolhido	
`FinishConfigureStandard (enum WifiPhyStandard standard)`	
Cria o gestor associado ao endereço MAC	
`SetWifiRemoteStationManager (Ptr<WifiRemoteStationManager> stationManager)`	
Callback* relativa ao encaminhamento de pacotes no sentido ascendente da *stack	
`SetForwardUpCallback (Callback<void, Ptr<packet>, Mac48Address, Mac48Address> upCallback)`	
***Callbacks* relativas à activação e desactivação de determinada ligação**	
`SetLinkUpCallback (Callback<void> linkUp)`	
`SetLinkDownCallback (Callback<void> linkDown)`	

Tabela 4.1: Métodos envolvidos no processo de *bootstrapping*

4.2.2.6 Recepção de pacotes de gestão

Nesta secção são ilustradas as decisões que são tomadas ao nível do nó para cada tipo de pacote de gestão.

Beacon É o pacote fundamental para a troca de informações entre os nós a nível MAC. Para tal, foram encapsuladas nos novos IEs algumas das informações que se querem transmitir entre os vários nós de forma a determinar, por exemplo, os parâmetros necessários ao cálculo das métricas sociais. Aproveitando ainda o facto do seu envio ser feito em *broadcast* e em intervalos de tempo regulares, desempenha também o papel de pacote *Hello*, de forma a que se saiba quando a ligação de associação se perde devido ao afastamento entre nós.

A figura 4.4 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote. Após ser desencapsulada a informação contida no pacote, realiza-se o processo de eleição, ilustrado em mais detalhe na figura 4.16. Em seguida adiciona-se a origem do pacote à tabela *Partial View*, dado que foi recebido um *Beacon* de um nó vizinho. Posteriormente, caso o tipo de associação da simulação em questão for *Social-based*, poder-se-à realizar uma nova associação ao nó de origem do pacote dependendo do valor da sua métrica social. Independentemente do tipo de associação, o *watchdog timer* responsável por monitorizar a ligação de associação poderá ser reiniciado dependendo da origem do pacote. Por fim, caso o nó não esteja associado ou decida alterar a sua associação, será enviado um *Association Request*, ficando-se a aguardar pela respectiva resposta (*Association Response*).

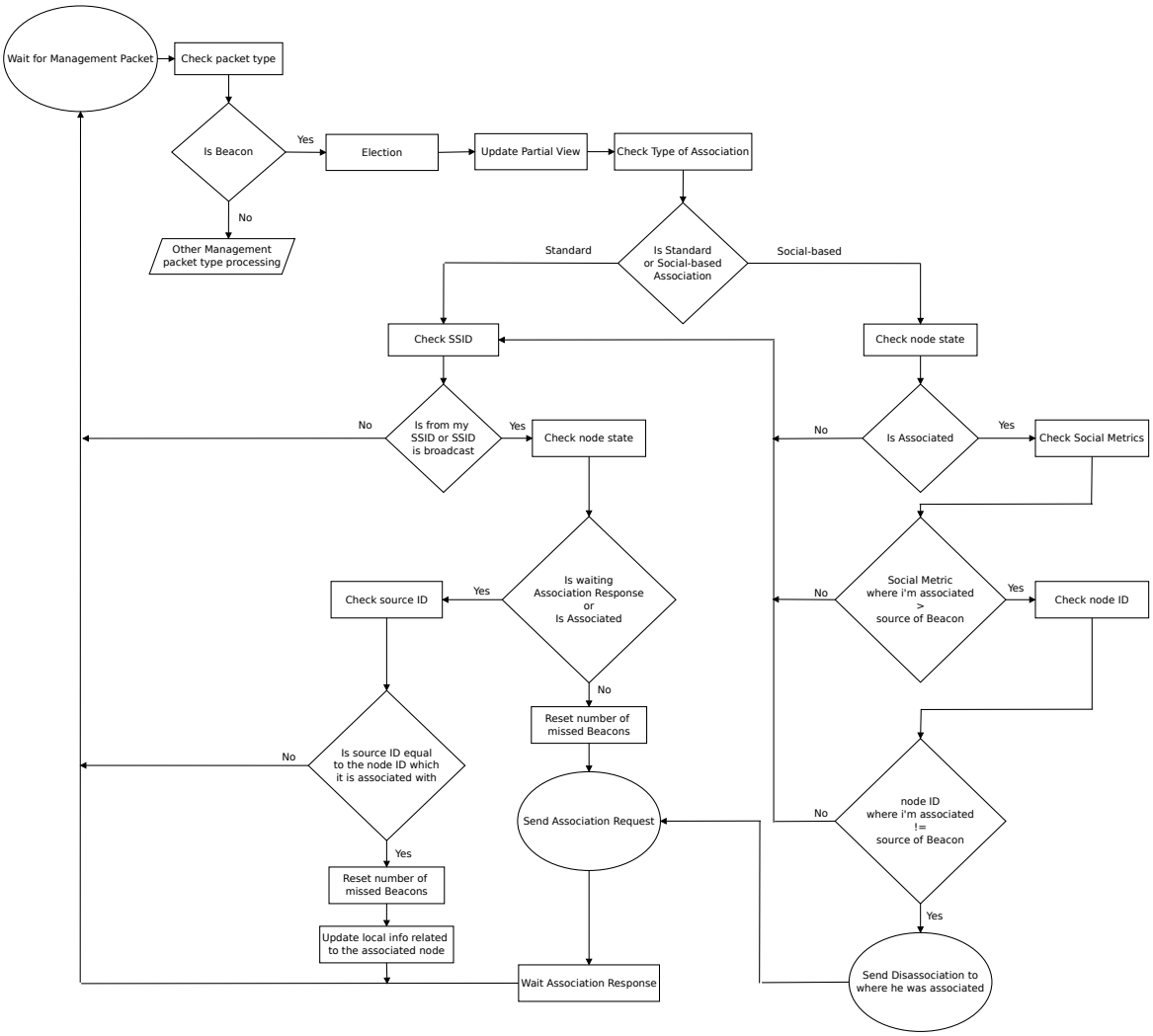


Figura 4.4: Recepção de *Beacon*

Probe Request É usado apenas no modo *active scanning*. É enviado em *broadcast* quando é detectada perda de ligação ao nó associado. Assim, em vez de aguardar o envio de *Beacons* por parte de um nó nas imediações, anuncia a sua presença para iniciar um processo de associação. A figura 4.5 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote.

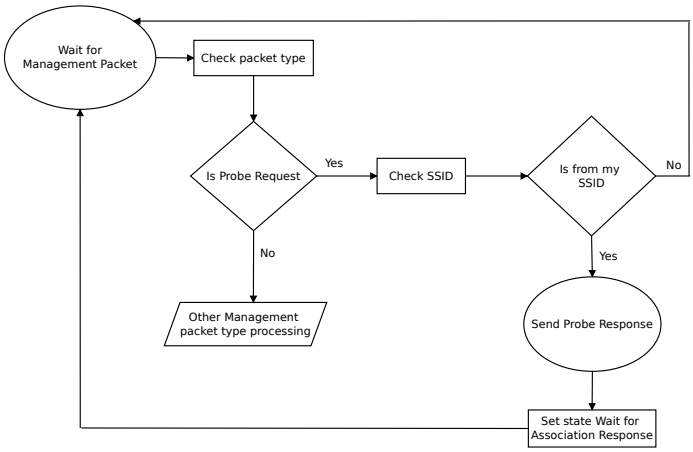


Figura 4.5: Recepção de *Probe Request*

Probe Response É a resposta ao pacote anterior e, consequentemente, só é usado no modo *active scanning*. Será enviado por todos os nós que recebam o *Probe Request*, indicando a sua presença e disponibilidade para iniciar o processo de associação. A figura 4.6 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote.

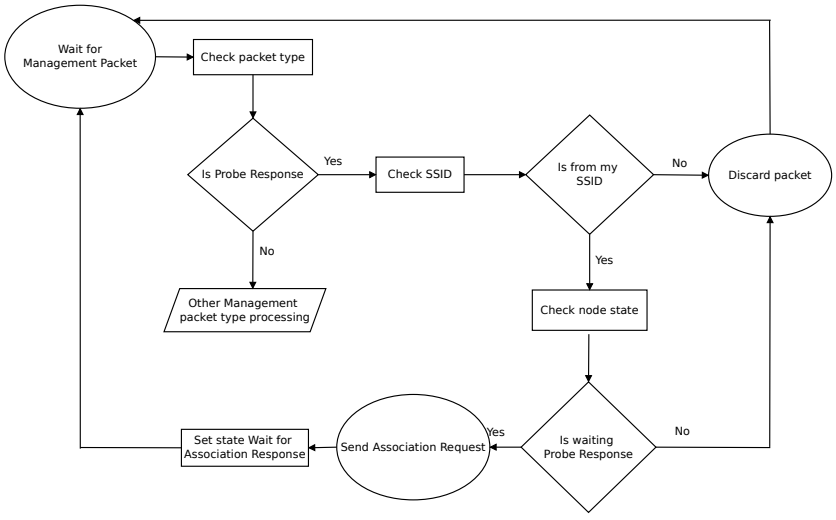


Figura 4.6: Recepção de *Probe Response*

Association Request Após a recepção de um *Beacon* no modo passivo ou de um *Probe Response* no modo activo por parte de um nó não associado, dá-se início ao processo de associação entre dois nós. A figura 4.7 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote, na qual se pode observar que, após ser recebido o pedido de associação, são verificados se os *Supported Rates* do nó correspondem aos do nó que requer a associação. Dependendo do resultado desta verificação, é enviada uma resposta positiva ou negativa relativamente à realização da associação.

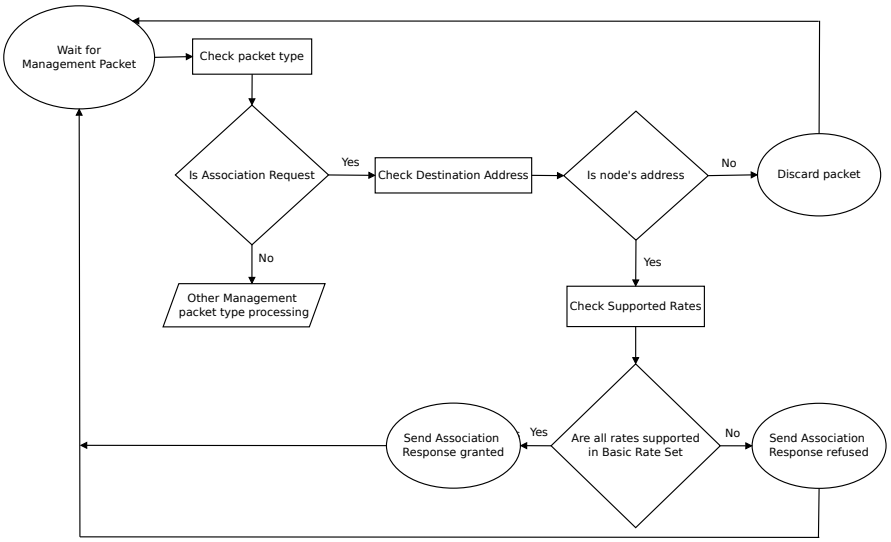


Figura 4.7: Recepção de *Association Request*

Association Response É a resposta ao pacote anterior e pode ser positiva ou negativa. Caso o nó esteja a aguardar a conclusão do processo de associação, e se a resposta conceder a ligação de associação, então irão ser guardadas no repositório local as informações relativas ao nó em que a associação passa a ser estabelecida. A figura 4.8 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote.

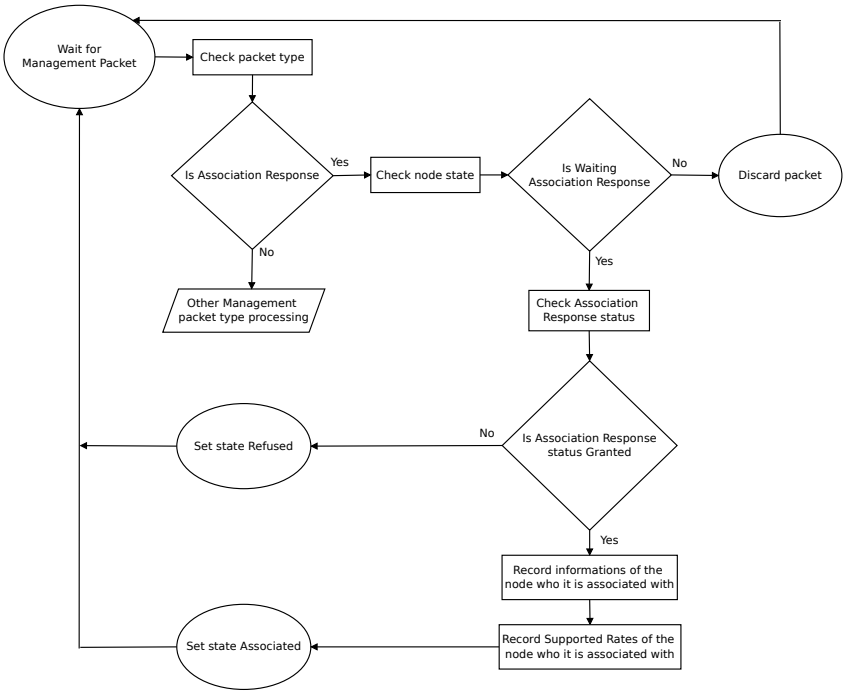


Figura 4.8: Recepção de *Association Response*

Disassociation É apenas usado no modo *Social-based Association* (figura 3.6). Uma vez que neste modo cada nó se encontra sempre associado àquele que esteja ao alcance e com melhor métrica social, sempre que chegue um nó melhor, é necessário notificar o antigo que a ligação de associação vai deixar

de existir (fluxo do lado direito da figura 4.4). A figura 4.9 ilustra o processamento efectuado quando se dá a recepção deste tipo de pacote.

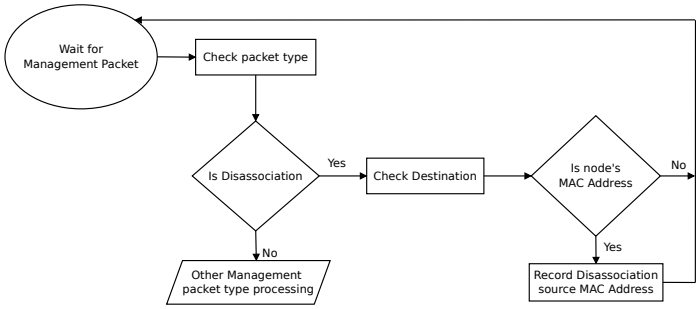


Figura 4.9: Recepção de *Disassociation*

Information Element Foram encapsulados IEs tanto ao pacote de gestão *Beacon* como ao *Association Response*. Estes contêm informações relevantes que vão ser trocadas entre os vários nós e que permitem, por exemplo, a associação ao melhor nó ao alcance. O encapsulamento destes ao nível do pacote de gestão 802.11 MAC em NS-3 é ilustrado na figura 4.10.

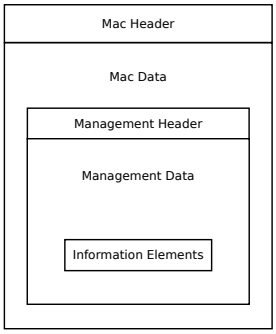


Figura 4.10: Estrutura genérica de um pacote 802.11 MAC em NS-3

Community-Based Beacon IE É a principal fonte de informação e tira partido dos *Beacons* serem gerados periodicamente por cada um dos nós para a propagar. Este IE está encapsulado no *Beacon* e, quando o pacote de gestão é recebido por um determinado nó, é desencapsulado e a informação contida é guardada numa tabela local, de acordo com critérios pré-estabelecidos. A sua estrutura é apresentada na figura 4.11.

A necessidade de enviar também um vector com os IDs dos nós conhecidos da comunidade onde o nó se encontra prende-se com a necessidade de estimar o tamanho da mesma (ver secção 4.2.2.10).

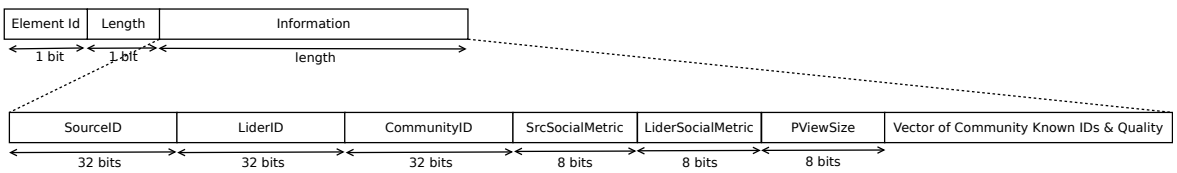


Figura 4.11: Estrutura do *Community-Based Beacon IE*

Community-Based Association Response IE A sua importância está relacionada com a manutenção e consistência da informação relativamente às características do nó e à comunidade onde se vai estabelecer a associação. A sua estrutura é apresentada na figura 4.12.

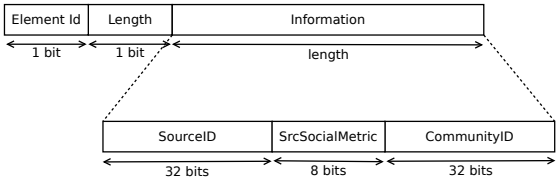


Figura 4.12: Estrutura do *Community-Based Association Response IE*

4.2.2.7 Identificador da comunidade

Neste trabalho definiu-se que uma comunidade é um conjunto de dois ou mais nós associados entre si, ou seja, existe um conjunto de ligações de associação entre nós tal que existe um caminho entre um qualquer nó A e um qualquer nó B. Neste trabalho uma comunidade é distinguida de outra através de um identificador único[5]. Num sistema distribuído o processo de manutenção desse identificador é realizado pelos vários nós pertencentes à comunidade. Devido à ausência de elementos centrais que possam garantir, em cada instante, a coerência do identificador, é necessário desenvolver mecanismos que garantam, de forma distribuída, não só a aquisição como a manutenção do ID de cada comunidade, nas situações já referidas na secção 3.4.4.3.

O processo é ilustrado na figura 4.13, na qual é possível observar que há dois tipos de pacotes envolvidos: o *Association Response* e o *Beacon*. O primeiro é responsável por garantir que, um nó que entre numa comunidade irá adoptar o identificador da mesma no momento da associação. O segundo permite realizar o processo de manutenção do ID da comunidade, dado que a mobilidade dos nós pode fazer com que as comunidades se fragmentem ou se unam. Para que a coerência do mesmo seja garantida, sempre que é recebido um *Beacon* cuja origem é o nó a que está associado, o identificador é actualizado, garantindo que mesmo nas situações referidas, o ID é propagado aos vários nós da comunidade. Além disso, um novo ID é gerado sempre que um nó perde (fragmentação da comunidade) ou lhe é requisitada uma associação (geração de uma nova comunidade).

[5] Inteiro aleatório de 32 *bits*

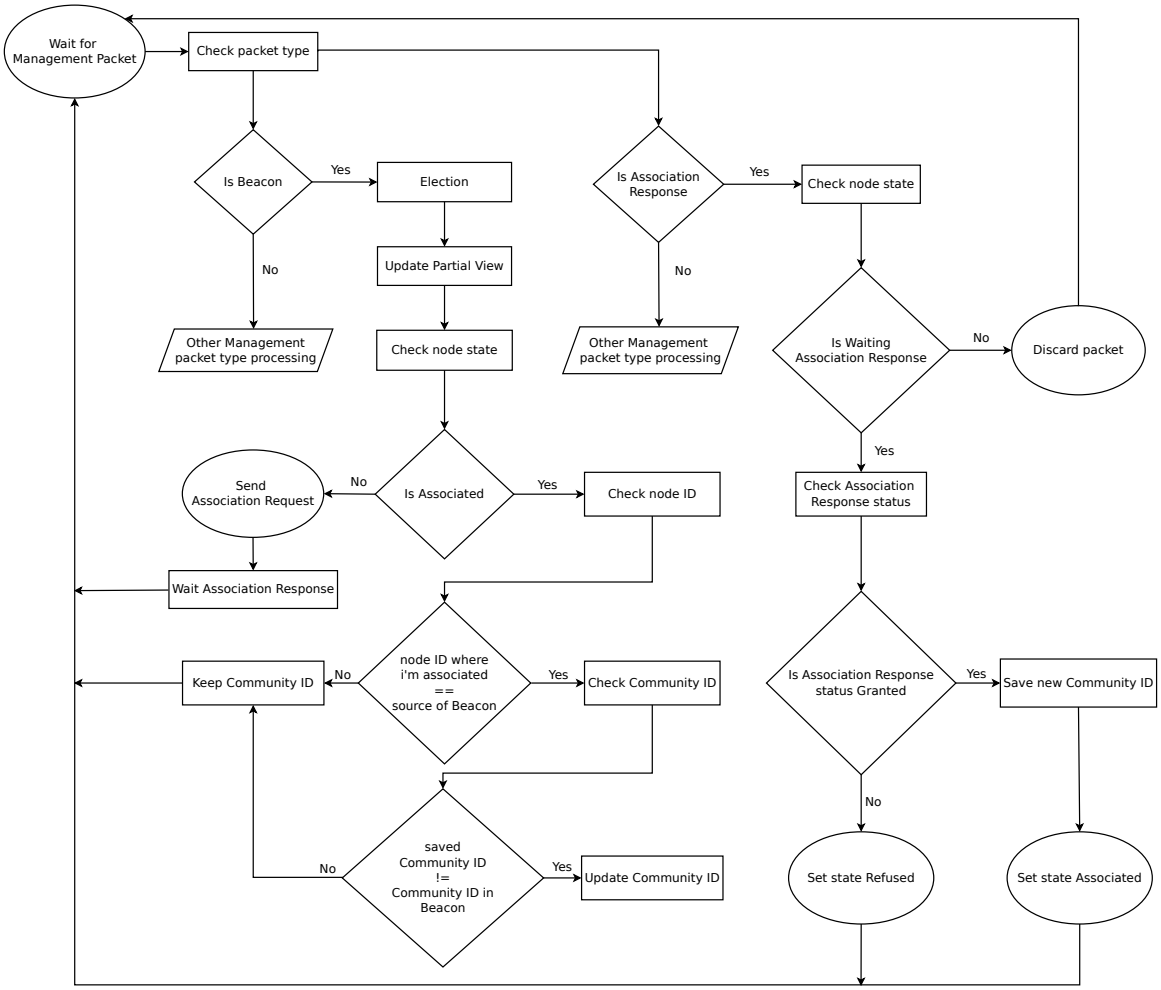


Figura 4.13: Processos envolvidos na aquisição e manutenção do identificador da comunidade

4.2.2.8 Tabelas dinâmicas de informação local

Partial View Esta tabela representa a vista parcial que cada nó tem da rede, ou seja, o conhecimento relativamente aos vizinhos. Nela encontram-se as informações recebidas recentemente pelos nós ao alcance.

Esta tabela local é dinâmica, isto é, o seu tamanho varia conforme seja inserida nova informação ou apagada aquela que se encontra desactualizada, ou seja, nós dos quais não foi recebido nenhum contacto há algum tempo (e.g. mais de 1s) e portanto se deduz que já não se encontram no raio de alcance.

A sua estrutura está definida da seguinte forma (figura 4.14):

- Cada entrada da tabela contém campos onde estão informações obtidas através dos IE recebidos através dos pacotes de gestão;
- Cada IE recebido só dá origem a uma nova entrada na *Partial View* se não existir nenhuma relacionada com esse ID;
- Se já existir, apenas os campos relevantes são actualizados;

- Caso não seja recebida informação vinda de um nó durante um determinado período de tempo, a entrada relativa a esse ID é removida.

uint32_t	uint32_t	uint32_t	uint32_t	uint32_t	uint32_t	uint32_t	uint32_t
Neighbor ID	Social Metric	Community ID	Lider ID	LiderSocialMetric	Timestamp	PView Size	NumRecBeacons
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

Figura 4.14: Estrutura da tabela *Partial View* de cada nó

Known Nodes Esta tabela contém os nós que, directa ou indirectamente, foram conhecidos. Ou seja, não contém apenas os nós no raio de alcance mas também aqueles que foram conhecidos através de outros. Esta tabela local também é dinâmica, isto é, o seu tamanho cresce conforme seja inserida nova informação, sendo a sua estrutura definida na figura 4.15.

A necessidade da existência desta tabela prende-se com o facto de um dos parâmetros de entrada da métrica social (CQE) ser a estimativa do tamanho e qualidade da comunidade em que o nó se encontra. A tabela é preenchida conforme a informação presente nos vectores *Known Nodes* e *Quality of Nodes* encapsulados no *Community-Based Beacon IE* .

Foi estabelecido um esquema de códigos (ver secção 4.2.2.10) para sinalizar a entrada e saída de nós da comunidade.

uint32_t	uint32_t	uint32_t	uint32_t
NodeID	CommunityID	Code	FQN
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

Figura 4.15: Estrutura da tabela de *Known Nodes* de cada nó

4.2.2.9 Eleição

O processo de eleição da presente solução tem como critério base o valor da métrica social. Assim, o nó eleito numa determinada comunidade será aquele que possuir a métrica social mais elevada. Este é um processo contínuo e dinâmico, sendo neste momento feita apenas a decisão local, isto é, ainda não se encontram implementados os mecanismos para a sinalização e a atribuição de funções específicas a essas entidades.

O processo começa por, no *bootstrapping*, cada nó se considerar líder. À medida que se vão recebendo informações dos outros nós (e.g. através do *Community-Based Beacon IE*), o líder vai sendo localmente actualizado. A decisão da eleição apenas considera nós que pertencem à mesma comunidade, ou seja, em cada instante, cada comunidade terá pelo menos um líder. Caso sejam cumpridas estas premissas, o critério de eleição é o valor da métrica social. Assim, se o líder for o mesmo, apenas será realizada a actualização do valor da sua métrica social. Caso o identificador recebido seja referente a outro nó, acontecerá uma de duas situações: o líder é mantido se a métrica social recebida for menor que a existente; o líder é alterado caso a métrica social recebida seja superior à existente. Este processo é ilustrado na figura 4.16.

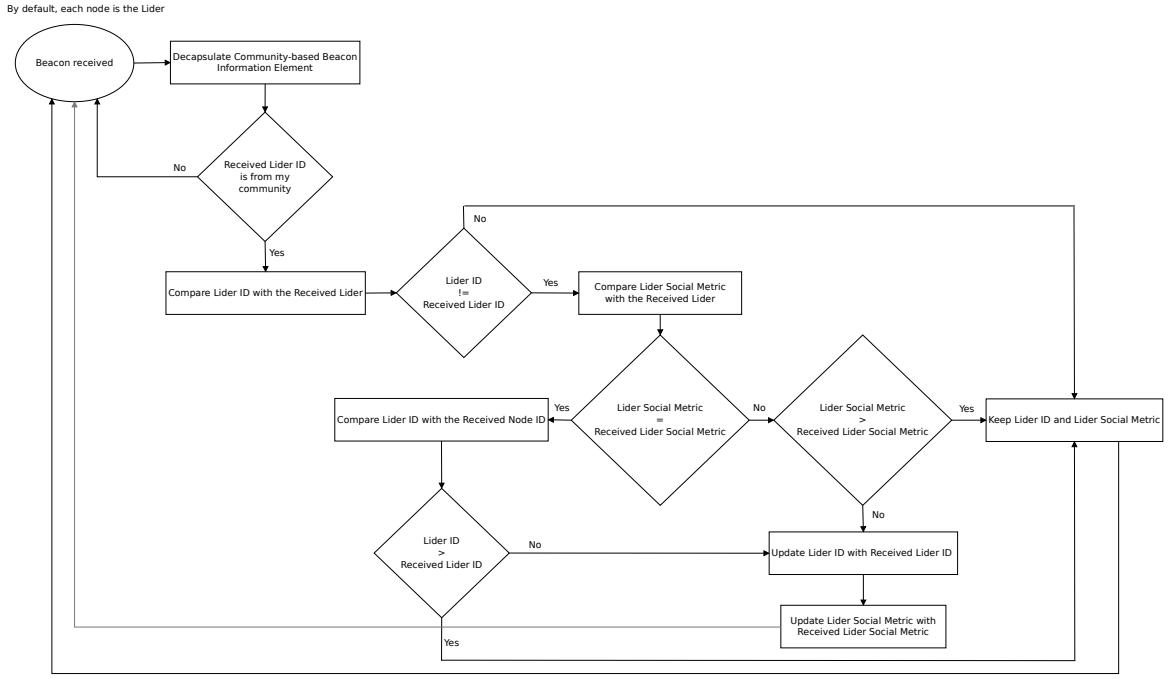


Figura 4.16: Processo de eleição

Outros critérios, como o tempo enquanto líder, podem complementar a indicação dada pela métrica social e serem também considerados na eleição. Contudo, soluções para os problemas de consenso e sincronização de sistemas distribuídos terão sempre de ser definidas (e.g. eleição em janelas intervalos de tempo regulares). É importante definir também critérios de desempate que garantam o consenso do líder em toda a comunidade (e.g. através do ID), bem como a escolha ou manutenção de ambos os líderes quando comunidades se separam ou se unem, são desafios a ter no desenvolvimento do processo de eleição.

4.2.2.10 Estimativa de parâmetros da comunidade

Um dos parâmetros de entrada da métrica social, a CQE, necessita de saber o tamanho e a qualidade média das "amizades" dentro de uma comunidade. Devido não só às limitações do simulador (tipos de dados permitidos ao nível dos IEs), como às características dinâmicas do sistema, é extremamente difícil assegurar, de forma distribuída, o número exacto de nós existentes e a qualidade das "amizades" numa determinada comunidade.

Uma vez que o alcance de cada nó é limitado, o seu conhecimento directo será constituído pelos nós que, em cada instante, estão na sua vizinhança (tabela local *Partial View*). Para ser possível que, localmente, cada nó consiga estimar parâmetros relativos ao estado da sua comunidade, é necessária a colaboração entre as várias entidades.

É necessário que este mecanismo se consiga adaptar às condições dinâmicas do sistema, isto é, que seja possível sinalizar a entrada e saída de nós de uma comunidade, bem como distinguir as informações mais recentes das mais antigas (o *timestamp* do pacote refere-se ao momento da sua geração e não ao da recolha da informação nele contida).

Para possibilitar a partilha de conhecimento entre as várias entidades, inseriram-se dois vectores nos IEs criados (*Community-Based Beacon IE*): um que transportará os IDs que pertencem a uma

dada comunidade e outro que terá o valor da FQN de cada um desses nós.

Tamanho da Comunidade Para que seja possível estimar localmente o tamanho da comunidade em que cada nó está inserido, é necessário não só propagar quais os nós que cada entidade conhece dentro da sua comunidade, como também sinalizar quando ocorre a saída e entrada de nós. Caso contrário, nos cenários com mobilidade considerados, o tamanho estimado por cada nó acerca da sua comunidade seria sempre crescente.

Para tal foi definido um esquema de códigos baseado no tipo de conhecimento que cada nó tem sobre determinado ID. Os códigos pretendem distinguir os vários tipos de conhecimento possíveis sobre os diversos nós, de forma a melhorar a estimativa do tamanho da comunidade. A necessidade de existir um código '0' para assinalar os IDs desconhecidos tem como única justificação o facto de, em NS-3, os IEs não poderem ser dinâmicos, ou seja, é exigido um tamanho pré-definido no momento da sua declaração. Assim, o índice do vector em questão corresponde ao ID do nó respectivo.

Código	Significado
0	ID desconhecido
1	ID conhecido indirectamente
2	ID conhecido directamente
3	ID com o qual se perdeu ligação

Tabela 4.2: Esquema de códigos para estimativa do tamanho da comunidade

Embora localmente seja mantida informação relativa a nós pertencentes a outras comunidades (para facilitar a aquisição de conhecimento caso o nó mude de comunidade), no envio apenas são colocados os IDs daqueles que pertençam à comunidade actual do nó. Na recepção dos códigos apresentados na tabela 4.3, a decisão de qual o código associado ao ID a inserir na tabela local *Known Nodes* terá de respeitar os seguintes critérios:

Código recebido	ID pertence à *Partial View*?	Código a inserir na *Known Nodes*
0	-	0
1	não	1
1	sim	2
2	não	1
2	sim	2
3	não	3*
3	sim	2

*imediatamente após o envio do próximo *Beacon*, os IDs cujos códigos da tabela *Known Nodes* sejam '3' passam a '0'.

Tabela 4.3: Decisão local baseada no esquema de códigos da tabela 4.2

Através da decisão ilustrada na tabela 4.3 consegue-se não só detectar e propagar a entrada e saída de nós na comunidade como garantir que a informação de um nó que tenha saído de uma comunidade não seja propagada indefinidamente, o que originaria incoerência da informação no caso de ocorrer reentrada desse nó na comunidade.

A figura 4.17 ilustra o envio de informação de um nó que, da sua comunidade, conhece o vizinho (forma directa) com o ID 3 e os nós com os IDs 1 e 5 (de forma indirecta). Tem ainda conhecimento que o nó com ID 6 abandonou a comunidade:

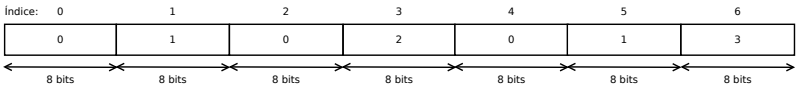


Figura 4.17: Exemplo de um vector de nós conhecidos

Qualidades das "amizades" na comunidade Para a partilha da informação relativa à qualidade da "amizade" (FQN) de cada nó o esquema seguido foi semelhante, optando-se também aqui por fazer corresponder o índice do vector ao IDs do nó correspondente pelas limitações do simulador anteriormente referidas. O conteúdo corresponde directamente ao valor da FQN. Embora localmente seja mantida informação relativa a nós pertencentes a outras comunidades (para facilitar a aquisição de conhecimento caso o nó mude de comunidade), no envio apenas são colocados os IDs dos nós pertencentes à comunidade actual do nó que está a transmitir. Na recepção estabeleceu-se que o último pacote recebido contém a informação mais actual, sabendo-se à partida que nem sempre é verdade. No entanto, o *timestamp* do pacote apenas fornece uma referência relativa ao tempo em que foi enviado e não de quando foi recolhida a informação nele contida. Além disso, como localmente será calculada uma média dos valores recebidos, não é exigido demasiado rigor a este nível.

4.2.2.11 Cenário de simulação

O cenário de simulação pode ser visto como o patamar de mais alto nível, ou seja, onde se pode ter a maior abstracção. Este foi desenvolvido com o intuito de executar a avaliação das várias soluções propostas. De forma a obter resultados em diferentes situações sem necessidade de alterar o código do mesmo, foi definido um conjunto de parâmetros de entrada no cenário genérico. Assim o utilizador pode definir o tipo de simulação de uma maneira muito simples.

Em termos genéricos o cenário é constituído por quatro métodos:

- `AdhocTest`: construtor por defeito que inicializa as variáveis da simulação com os valores por defeito;
- `Configure:` configura a simulação de acordo com os parâmetros indicados pelo utilizador*;*
- `Run:` executa a simulação;
- `Report`: mostra os resultados ao utilizador.

Ao nível da passagem de parâmetros de entrada como forma de definir algumas variáveis da simulação, existem as seguintes opções:

- `--verbose`: reporta as actualizações das tabelas de informação local;
- `--pcap`: exportação dos PCAP *traces;*
- `--useCChange`: reporta as actualizações das mudanças de rota;
- `--size`: número de nós por linha da grelha de alocação inicial ;
- `--nWifi`: numero de nós *ad hoc* sem fios;
- `--step_x`: distância inicial entre nós no eixo X;

- `--step_y`: distância inicial entre nós no eixo Y;
- `--simTime`: tempo total de simulação;
- `--mobile`: modelo de mobilidade (*random walk* / estático);
- `--activeScan`: modo de *scanning* (activo / passivo);
- `--adaptiveInterv`: intervalo entre *Beacons* (adaptativo / fixo);
- `--optimalAssoc`: modo de associação (com / sem uso da métrica social);
- `--w1`: peso do parâmetro FIN da métrica social;
- `--w2`: peso do parâmetro CQE da métrica social;
- `--w3`: peso do parâmetro FQN da métrica social.

É importante referir que, variando o número de nós, a janela de simulação é dinamicamente adaptada, de forma a que o comportamento das várias simulações seja directamente comparável.

Sendo assim, para executar uma simulação o utilizador apenas terá de introduzir um comando do tipo:

```
NS_GLOBAL_VALUE="RngRun=valA" ./waf --run "scratch/adhoc-test --param1=valB
    --param2=valC ... --paramN=valN" > log.dat 2>&1
```

onde,

Designação	Significado
valA	número da semente da simulação a usar
param1, param2 até *paramN*	parâmetros de entrada do cenário
valB, valC até *valN*	valores a atribuir a cada um respectivamente
log.dat	ficheiro para redireccionamento do *output*

Tabela 4.4: Significado das variáveis do comando para execução de uma simulação

4.2.3 Desafios enfrentados

A implementação do modelo de comunicação *ad hoc*, integrado com os vários mecanismos e funcionalidades descritas ao longo do capítulo trouxe vários desafios. Em termos conceptuais, e como já referido em parte na secção 3.4.7, o facto da rede não possuir nós centrais que sirvam como pontos de sincronização e consenso da informação torna o trabalho muito aliciante. Também o facto das entidades que constituem a rede possuirem apenas uma visão parcial da mesma e terem de tomar decisões globais (e.g. eleição de líderes) é mais um factor de motivação, uma vez que as funcionalidades a acrescentar não devem introduzir demasiada complexidade na rede, mantendo a solução escalável.

A estes podem ainda ser acrescentados os seguintes desafios mais relacionados com a implementação dos conceitos e mecanismos anteriormente descritos:

- Comunicação *half duplex*: este foi um dos primeiros problemas após ser criado o modelo do nó *ad hoc*. Apesar de não ser referenciado em [111], o modelo PHY utilizado é *half duplex*, ou seja, é possível haver tanto transmissão como recepção, no entanto não é possível que ambas ocorram em simultâneo. O que acontecia era que, como o envio de *Beacons* por parte dos vários nós era periódico, todos transmitiam exactamente no mesmo instante, criando as designadas *startup storms*. A solução adoptada passou por, mantendo o intervalo fixo, introduzir um ligeiro atraso entre o início de transmissão de cada nó.

- ID aleatório *vs* ID sequencial: como forma de garantir que cada nó possui um identificador único, inicialmente optou-se por atribuir um ID aleatório a cada um. Contudo, devido às limitações do simulador ao nível dos IEs, abordadas em 4.2.2.10, foi necessário recorrer a IDs sequênciais de forma a ser possível seguir a estratégia descrita na mesma secção. Sabe-se ainda que, num contexto real, este facto implicaria o recurso a um servidor central (à semelhança do DHCP) que fornecesse os identificadores conforme fossem recebidos os pedidos. Contudo, se a dimensão dos IEs adicionados pudesse ser dinâmica, o identificador do nó poderia ser gerado localmente e de forma aleatória;

- Modo *active probing*: existe um *bug* associado à activação deste modo de *scanning*, o qual foi reportado por nós em [118]. Foi posteriormente proposta uma solução que no entanto não funciona quando o modo é activado em múltiplas STAs. Como forma de ultrapassar o problema, definiu-se que os nós entravam em simulação usando o modo passivo e a partir desse momento, de cada vez que perdessem associação, o modo em funcionamento seria o activo (i.e. com recurso a *Probe Request/Response*).

4.3 Redes com fios

4.3.1 Módulos funcionais

O protocolo implementado ao nível das redes com fios é constituído por vários módulos com funções distintas: *Repository*, *Proxy*, *MmMsg*, *Throughput*, *TimeSampling*, *InetAddrv6, HelloReceivedLocal*, *AuxFunctions* e *LnDiscover*.

4.3.1.1 *Repository*

O módulo *Repository* é constituído pelas principais classes que compõem um nó:

- Classe *Node*: a sua função é guardar informação característica do nó, nomeadamente o identificador único, o IP, o RTT da troca de mensagens, o número de interfaces, a interface em que o protocolo está a correr e a largura de banda da mesma, a percentagem de CPU e de RAM livre, o número de nós já descobertos pela entidade e o seu próprio *ranking*.

- Classe *Repository*: é onde está armazenada toda a informação da rede recolhida pela entidade. É garantida exclusão mútua dado poderem ocorrer tentativas de acessos simultâneos. O repositório local é constituído pela *Partial View* que o nó tem da rede, onde constam as informações das entidades já conhecidas. Está ainda presente informação acerca do líder actual.

- Classe *Rank*: é onde estão armazenados os *ranking* (equação 3.7) dos vários nós conhecidos, de forma a que haja uma correspondência entre a posição do nó nesta classe e no repositório.

- Classe RTT: auxiliar para proceder ao cálculo do RTT na troca de mensagens.

4.3.1.2 *Proxy*

O módulo *Proxy* é o responsável pela troca correcta de mensagens entre os nós, sendo que, por cada pacote recebido é lançada uma nova *thread* que irá realizar todo o processamento e envio da resposta. Dado que o mecanismo de descoberta se baseia na troca de pacotes *Hello*, este módulo é considerado a base desse mecanismo. Como o protocolo suporta tanto IPv4 como IPv6, é necessário que existam funções para processamento dos pacotes na recepção segundo ambos. Informações como quais as interfaces em a resposta vai ser enviada, a sua largura de banda e o porto de destino segundo o protocolo usado são também aqui definidas.

4.3.1.3 *MmMsg*

O módulo *MmMsg* contém a definição de todas as mensagens que são trocadas no protocolo: *Hello*, *NodeInfo* e *LeaderInfo*.

- Classe *Hello*: representa o pacote *Hello* que é enviado por cada entidade *INM_Seeker* de forma a descobrir os outros nós na rede. Apenas contém dois campos (*MsgType, Hello ID*) de forma a que seja possível o cálculo do RTT. O intervalo entre o envio destes pacotes é adaptativo, sendo dependente do número de *INM_Seekers* presentes na *Partial View* de cada nó.

- Classe *NodeInfo*: representa o pacote *NodeInfo* que é enviado por cada entidade após ter sido contactada, ou seja, após a recepção de um *Hello*. Neste pacote de resposta serão enviados os recursos locais do nó (e.g. percentagem de CPU e RAM livre), informações da interface de resposta (e.g. nome e largura de banda) e informações complementares como o número de nós conhecidos, o endereço MAC e o identificador do nó que envia a mensagem.

- Classe *LeaderInfo*: representa a mensagem *LeaderInfo* que apenas é enviada pelo nó eleito a todos os seus vizinhos. Neste pacote apenas será enviado o identificador dessa mesma entidade líder.

4.3.1.4 *Throughput*

Este módulo existe para que seja possível calcular o parâmetro de *throughput* em relação ao número de mensagens enviadas e recebidas durante um determinado tempo de observação.

4.3.1.5 *TimeSampling*

A função deste módulo é unicamente calcular o tempo entre dois eventos consecutivos, de modo a se saber o tempo que um nó demora a construir a sua vista parcial. Desta forma é possível, posteriormente, saber o tempo de convergência da informação na rede.

4.3.1.6 *InetAddrv6*

O módulo *InetAddrv6* é o responsável pela criação de endereços IPv6. Neste estão definidos métodos para a definição do *scope* e do IP do endereço.

4.3.1.7 *HelloReceivedLocal*

A função deste módulo é apenas registar os identificadores dos nós dos quais já foram recebidos pacotes *Hello*, ou seja, os nós já conhecidos. O objectivo é que, caso seja posteriormente recebido um *Hello*, não seja necessário realizar o seu processamento.

4.3.1.8 *AuxFunctions*

Neste módulo encontra-se definido um conjunto de métodos auxiliares com o intuito de facilitar a tarefa de implementação. Podem-se destacar os métodos de obtenção do endereço MAC a partir do nome da interface, da percentagem de CPU e RAM livre, de uma lista de todas as interfaces de rede de um nó, de uma lista dos endereços IPv6 de uma interface, etc.

4.3.1.9 LnDiscover

Neste módulo encontra-se a função *main* do algoritmo, na qual são inicializadas todas as estruturas de dados necessárias. Após o início da sua execução, é lançada uma *thread* para cada interface existente que aguardará a recepção de novas mensagens. Sempre que for recebido um pacote será lançada uma nova *thread* do tipo *Proxy* que executará o processamento e respectivo envio de mensagem de resposta. Caso a entidade seja configurada como *INM_Seeker*, irá enviar pacotes *Hello* em *broadcast* (IPv4) ou em multicast para o endereço *All-Hosts* (IPv6) em intervalos de tempo adaptativos. Caso seja configurada como *INM_Hider,* então aguardará o contacto por parte de outra entidade, mudando nesse instante o seu papel.

4.3.2 *Scripts* de inicialização

O protocolo permite a definição de alguns parâmetros de entrada de modo a ser facilmente configurável:

- `-a`: inicia o protocolo em modo IPv6;
- `-d [profundidade]`: define o TTL dos pacotes *Hello* e consequentemente limita a profundidade da vista parcial do nó;
- `-h`: mostra o menu de ajuda;
- `-s`: inicia a entidade como *INM_Seeker* (por defeito inicia como *INM_Hider*);
- `-t [número de nós]`: faz a contagem do tempo até sejam descobertos `[número de nós]` pela entidade;

Para que a interacção com o utilizador seja simples, foram ainda definidos quatro *scripts* de inicialização: `auto_seeker.sh`, `auto_hider.sh`, `auto_seeker_test.sh` e `auto_hider_test.sh`.

- `auto_seeker.sh`: permite iniciar um nó como *INM_Seeker* que fará a busca de outros nós presentes na rede;

- `auto_hider.sh`: permite iniciar um nó como *INM_Hider* que aguardará ser contactado até mudar o seu papel;
- `auto_seeker_test.sh`: permite iniciar um nó como *INM_Seeker*, saber o tempo de descoberta e o número de mensagens recebidas e enviadas. Para tal deverá ser seleccionada a opção "`w`" enquanto se iniciam os nós e a opção "`s`" no último nó, de forma a iniciar o processo de forma síncrona;
- `auto_hider_test.sh`: permite iniciar um nó como *INM_Hider*, saber o tempo de descoberta e o número de mensagens recebidas e enviadas. Para tal deverá ser seleccionada a opção "`w`" enquanto se iniciam os nós e a opção "`s`" no último nó, de forma a iniciar o processo de forma síncrona;

4.3.3 *Testbed*

A *testbed* utilizada é constituída por 25 máquinas em grelha 5x5, cada uma com 558 MHz CPU, 512 MB RAM, 1 GB de disco e OS Debian Lenny 2.6.26, sendo criadas sobre um servidor através de virtualização (Xen). O atraso extremo-a-extremo (i.e. da máquina 1 à 24) é 0.745 ms e a largura de banda de cada ligação virtual de 1 Mbps. O esquema geral da ligação entre máquinas é apresentado na figura 4.18.

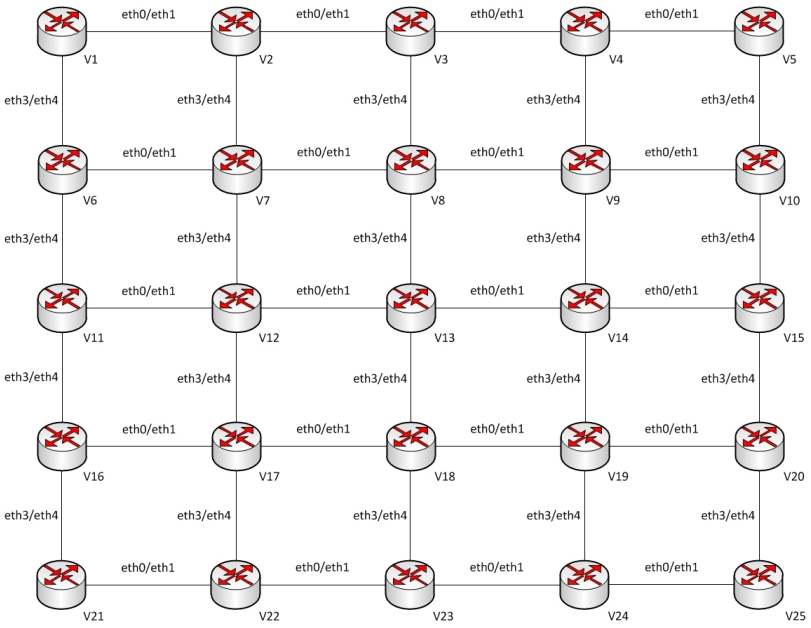


Figura 4.18: Esquema geral da *testbed*

O desenvolvimento e teste da solução numa *testbed* virtual implica algumas dificuldades de carácter mais técnico, como:

- A capacidade limitada do servidor onde está colocada a *testbed*, dificultando a análise da escalabilidade da solução;
- A gestão dos terminais de cada máquina virtual, devido à necessidade de configuração de cada uma com parâmetros específicos.

4.4 Conclusões

Neste capítulo foi apresentada a implementação das soluções propostas no capítulo 3.

Para as redes sem fios, o desenvolvimento dos vários mecanismos descritos foi realizado no simulador NS-3. Neste ambiente de simulação foi necessário, antes de mais, implementar o modelo de comunicação *ad hoc* ao nível MAC do protocolo IEEE 802.11. A justificação deve-se ao facto do objectivo último ser atingir a gestão distribuída e autonómica da rede. É sobre este modelo que assentam todos os mecanismos de cooperação e funcionalidades desenvolvidas, nomeadamente o *bootstrapping*, a descoberta e a eleição, cujos requisitos de informações exigem a inclusão de novos IEs nos pacotes de gestão do protocolo 802.11 para possibilitar a partilha de conhecimento entre entidades.

Foi ainda apresentado o cenário de simulação que aceita vários parâmetros de entrada de forma a facilitar a interacção com o utilizador.

Para as redes com fios, o desenvolvimento foi realizado ao nível de uma *testbed* virtual. Neste capítulo são apresentados detalhes relativos aos módulos implementados e que servem de base aos mecanismos de *bootstrapping*, descoberta e eleição.

Capítulo 5

Resultados

5.1 Introdução

O presente capítulo prentende avaliar as características das soluções implementadas, recorrendo à análise de diferentes cenários. À semelhança dos capítulos anteriores, também este se encontra dividido em duas secções principais: redes sem fios e redes com fios.

Em 5.2 são apresentados os resultados das redes sem fios através dos quais se pretende avaliar os seus mecanismos e a influência da métrica social e dos seus parâmetros individuais.

Em 5.3 é realizada uma análise aos resultados obtidos na *testbed* de redes com fios, como forma de avaliar o desempenho do protocolo desenvolvido comparativamente a outros protocolos de descoberta mais populares.

5.2 Redes sem fios

5.2.1 Cenários usados

Para a realização dos resultados em seguida apresentados, foram configurados vários cenários nos quais se introduziram diferentes parâmetros de entrada. Os parâmetros transversais a todas as simulações são o tempo de simulação de 300s, o padrão de mobilidade *random walk* e a separação inicial entre nós de 50m (correspondente a metade do seu raio de alcance). Os parâmetros específicos de cada simulação são o número de nós, o uso de *scanning* activo ou passivo, o intervalo entre *Beacons* fixo ou adaptativo, o uso ou não de métrica social e o peso de cada um dos parâmetros que a constitui (w_1, w_2 e w_3).

Um outro parâmetro importante em simulação é o valor da semente usada na simulação, a qual foi diferente em cada simulação. Esta é responsável por alterar o valor de parâmetros aleatórios, ou seja, simulações do mesmo cenário produzem resultados diferentes com valores de sementes distintos.

Os resultados apresentados foram obtidos através da média de 5 repetições independentes com intervalos de confiança a 90%.

Parâmetro	Possibilidades usadas
Tempo de simulação	300 s
Alocação inicial dos nós	em grelha
Padrão de mobilidade	*random walk* (velocidade e trajectória aleatórias)
Separação inicial	50 m
Número de nós	4, 9, 16, 25, 36, 49, 64, 81 ou 100
Dimensão da janela de simulação	proporcional ao número de nós
Tipo de *scanning*	activo ou passivo
Intervalo entre *Beacons*	fixo ou adaptativo
Uso de métrica social	com ou sem
Pesos dos parâmetros da métrica social	igual ou diferente de 0
Semente	1, 2, 3, 4 ou 5 (dependendo da repetição da simulação)

Tabela 5.1: Cenários usados para a obtenção de resultados

5.2.2 Influência da métrica social e dos seus pârametros individuais

Os resultados apresentados nas secções 5.2.2.1 e 5.2.2.2 pretendem mostrar a influência, individual e conjunta, dos parâmetros da métrica social para determinadas características do sistema. Na secção 5.2.2.1 é realizada uma análise global à simulação, variando o número de nós em simulação, apresentando os resultados de cada parâmetro analisado. Em 5.2.2.2, a análise é apenas feita às comunidades para as quais os nós tendem a agrupar-se, isto é, aquelas que existem no final de simulação, sendo a apresentação dos resultados feita em função do tempo. As figuras apresentadas estão subdivididas em (a) e (b) correspondendo às simulações onde foi usado o intervalo fixo e adaptativo entre *Beacons* respectivamente, de modo a apresentar também o impacto deste na rede.

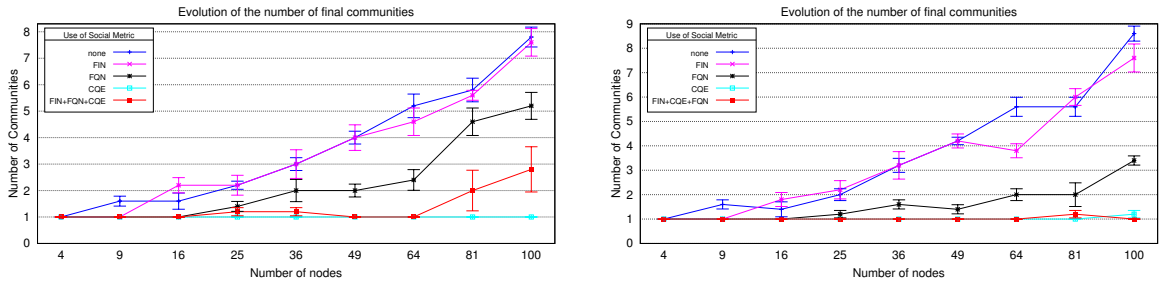
5.2.2.1 Variação do número de nós em simulação

Na figura 5.1 é apresentado o número de comunidades que existem no final da simulação, ou seja, aquelas onde os nós tendem a agrupar-se nos diferentes cenários simulados.

Em 5.1a mostra-se que na curva FIN+FQN+CQE, em simulações até 64 nós, o número de comunidades é fortemente influenciado pelo valor da CQE. Para simulações com maior número de nós, a rede tende a fragmentar-se em comunidades de menores dimensões, sendo que o número final de comunidades na rede obedece a uma influência idêntica entre a FQN e a CQE. Caso não seja usada a métrica social, ou seja, caso as associações sejam realizadas sem critério de decisão, a rede tende a fragmentar-se, numa relação "nºcomunidades:nºnós" que tende a ser constante no valor de 1:10.

Em 5.1b, na curva FIN+FQN+CQE, os nós tendem a juntar-se numa única comunidade, independentemente do número de nós em simulação. O facto da menor influência da FQN neste tipo de simulações é justificado por ter sido retirado o parâmetro correspondente aos recursos físicos do nó, deixando-se apenas a estabilidade da associação. Esta opção tem como justificação o facto de se estar a impor algo que não corresponde às características do nó em simulação, dado que no ambiente de simulação NS-3, os recursos de todos os nós são iguais. Assim, apesar da informação fornecida pela estabilidade da ligação ser relevante, não resulta num valor diferenciador entre nós por variar em períodos muito curtos de tempo (diminui na iminência da perda de associação e volta a recuperar o valor máximo quando ocorre uma nova associação). Já que a FIN exerce uma fraca influência a este

nível, assemelhando-se ao comportamento da simulação sem métrica social, o número de comunidades será maioritariamente imposto pela CQE.



(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

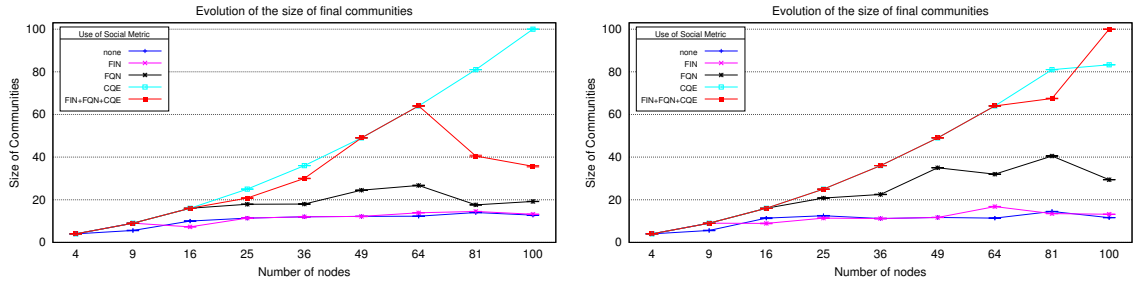
Figura 5.1: Evolução do número de comunidades finais

Na figura 5.2 é apresentado o tamanho médio das comunidades existentes no final da simulação, ou seja, a quantidade de nós que essas comunidades tendem a agrupar nos vários cenários simulados.

O gráfico 5.2a é complementar ao apresentado em 5.1a, mostrando de forma explícita o número médio de nós em cada comunidade final. As ligeiras discrepâncias no tamanho de comunidades finais nas simulações da CQE e FIN+FQN+CQE com 25 e 30 nós são mais evidentes neste gráfico, sendo a contribuição principal o valor da FQN.

Em 5.2b é também destacada a discrepância entre as simulações da CQE e FIN+FQN+CQE com 81 nós, novamente a ser maioritariamente influenciada pelo valor da FQN.

É relevante referir ainda que, pela análise das figuras 5.1 e 5.2, se pode concluir que o mecanismo distribuído de estimação do tamanho das comunidades 4.2.2.10 conduz, em média, a um resultado acertado, dado que $n^o comunidades * tamanho\,médio\,comunidades \approx n^o nós\,simulação$.



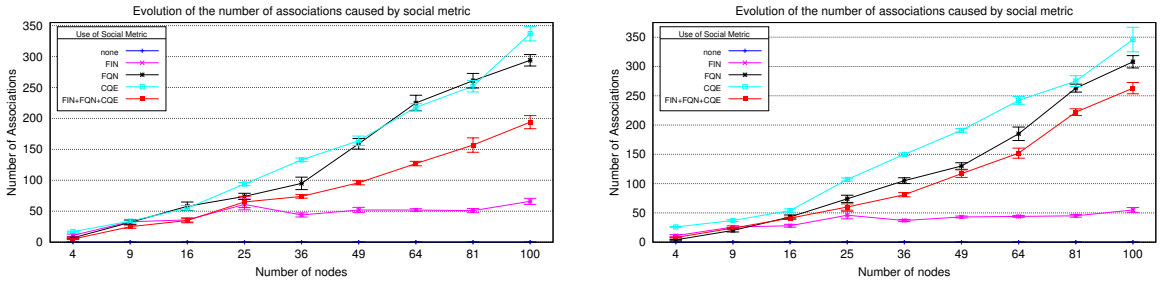
(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

Figura 5.2: Evolução do tamanho das comunidades finais

Na figura 5.3 é apresentado o número de associações forçadas pela métrica social, ou seja, devido à entrada de novos nós no raio de alcance de cada um, haverá a possibilidade de descoberta de métricas sociais superiores àquela onde cada um está associado. Este facto implicará a realização de uma nova associação, que pode ser provocada não só pela mobilidade como também pela recolha contínua e partilha de conhecimento. Assim, o melhor nó ao alcance poderá não ser sempre o mesmo e, se o valor da métrica social recebida passar um determinado limiar superior (equação 3.1), será realizada uma nova associação.

No gráfico 5.3a, o número total de associações na simulação FIN+FQN+CQE tende a situar-se entre as simulações com cada um dos parâmetros individualmente. Este aspecto é justificado pelo facto do limiar (equação 3.1) ser adaptativo em função dos pesos usados no cálculo da métrica. Na simulação sem recurso à métrica social, as associações são feitas indiscriminadamente, daí no gráfico a sua linha se situar sobre o valor 0.

Na figura 5.3b, devido à menor influência individual da FQN na métrica social já explicada anteriormente, o resultado da simulação FIN+FQN+CQE está situada entre o número de associações forçadas nos cenários individuais FIN e CQE. No entanto, há uma tendência para que a curva da FIN+FQN+CQE siga a curva da simulação CQE, devido à grande influência desta no valor final da métrica social, apresentando aproximadamente menos 100 associações nas simulações de 25 a 100 nós.



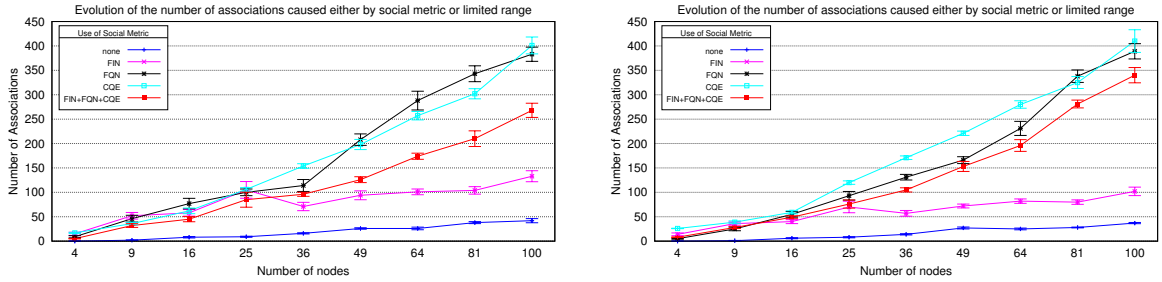
(a) Intervalo fixo entre *Beacons*

(b) Intervalo adaptativo entre *Beacons*

Figura 5.3: Evolução do número total de associações provocadas pela métrica social

Na figura 5.4 é apresentado o número total de associações, isto é, aquelas que são forçadas pela métrica social (figura 5.3) e aquelas que são provocadas pelo raio de alcance limitado de cada nó. Caso seja detectada esta última situação, é necessário proceder a uma nova associação. Uma vez que o padrão de mobilidade e o raio de alcance usados são sempre os mesmos, é esperado que o número de associações provocadas pela perda de conectividade entre os nós seja semelhante para todas as simulações, algo que pode ser confirmado através da comparação entre as figuras 5.4 e 5.3.

Como a simulação *none* não apresenta associações devido à metrica social, os valores apresentados na figura 5.4 representam o número de associações provocadas pela saída de nós do raio de alcance daquele a que estão associados. Somando cada um destes valores às curvas da figura 5.3 verifica-se que o resultado é aproximadamente o apresentado em 5.4. Desta forma demonstra-se que o padrão de mobilidade usado e o raio de alcance do nó são factores que afectam de igual forma os cenários com e sem métrica social ao nível do número de associações provocadas pela perda de conectividade.



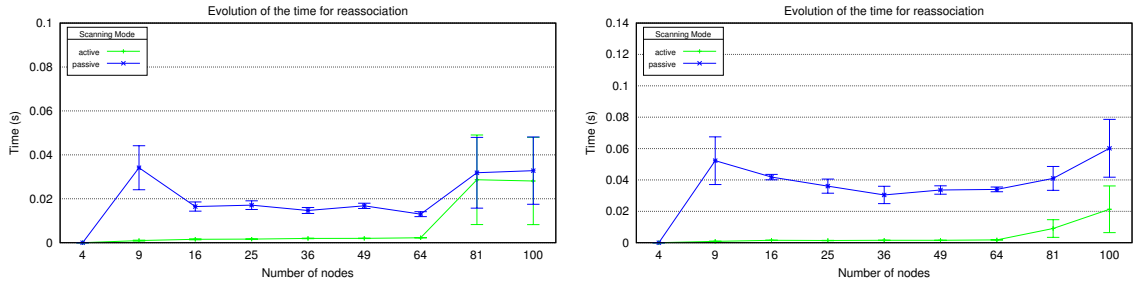
(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

Figura 5.4: Evolução do número total de associações provocadas pela métrica social e pelo alcance limitado

Nos gráficos da figura 5.5 é feita uma comparação entre os dois modos possíveis de *scanning/probing* implementados de acordo com protocolo 802.11.

Com o intervalo entre *Beacons* fixo em 0.1s (figura 5.5a), o valor médio para readquirir associação é aproximadamente 0.025s no modo passivo. Ainda assim, usando o modo activo este tempo é praticamente nulo em simulações até 64 nós. Em simulações com 81 e 100 nós, o número de colisões de pacotes é bastante superior, daí que o modo activo deixe de ser tão eficiente. A acrescentar a isto, há ainda o facto de que, em cenários com elevado número de nós, existe uma elevada densidade de *Beacons* em cada instante, fazendo com que os dois modos de *scanning/probing* se tornem semelhantes.

Com o intervalo entre *Beacons* adaptativo (figura 5.5b), o valor médio para readquirir associação é aproximadamente 0.04s no modo passivo, valor ligeiramente superior ao do gráfico 5.5a. Usando o modo activo este tempo volta a ser praticamente nulo em simulações até 64 nós, à semelhança da figura 5.5a. Em simulações com 81 e 100 nós, o modo activo continua a ser bastante mais eficiente que o modo passivo, demorando aproximadamente menos 0.035s a readquirir associação. Um dos factores que contribui para a diferença relativamente à figura 5.4b é o facto da densidade de *Beacons* ser menor, o que implica menor número de colisões e, por isso, o modo activo apresentar melhor desempenho.



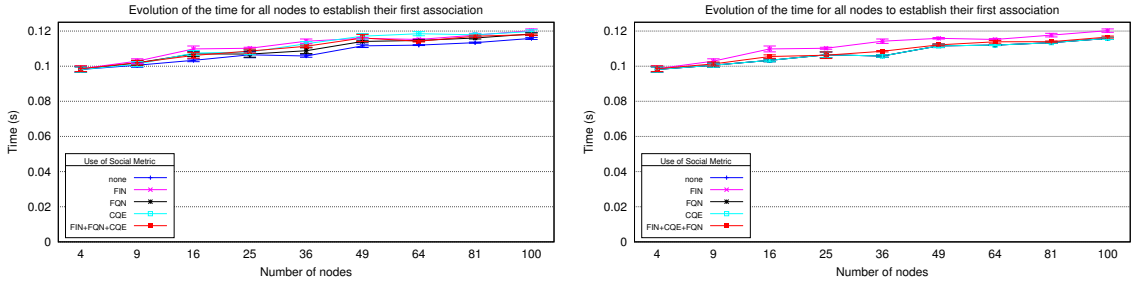
(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

Figura 5.5: Evolução do tempo para restabelecimento de associação

Os gráficos da figura 5.6 têm como objectivo demonstrar que a introdução da métrica social não tem implicações ao nível do tempo para associação.

Através dos gráficos 5.6a e 5.6b mostra-se que o tempo médio que todos os nós demoram para adquirir a sua primeira associação é aproximadamente 0.011s, tanto para simulações com intervalo entre *Beacons* fixo como adaptativo. Isto acontece porque em 5.6b está definida uma zona inicial de

aquisição de informação na qual o intervalo entre *Beacons* se mantém fixo e com o seu valor por defeito (0.1s). Assim os gráficos da figura 5.6 correspondem a zonas iniciais em que o comportamento do intervalo entre *Beacons* é semelhante. A informação apresentada demonstra também que a introdução da métrica social não degrada o tempo necessário ao estabelecimento da associação de todos os nós. Caso a comparação não fosse relativa à primeira associação mas ao tempo de reassociação (análise da figura 5.5), o resultado seria semelhante (curvas sobrepostas mas deslocadas verticalmente em relação à figura 5.6).



(a) Intervalo fixo entre *Beacons*

(b) Intervalo adaptativo entre *Beacons*

Figura 5.6: Evolução do tempo para estabelecimento da primeira associação em todos os nós

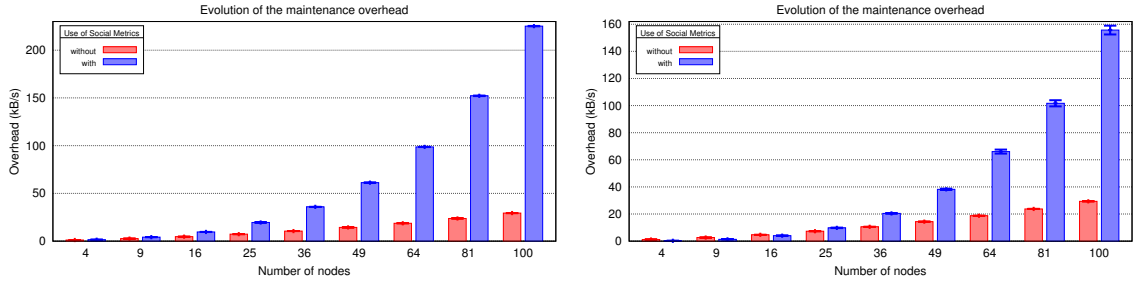
A figura 5.7 mostra o *overhead* necessário à manutenção da rede. Designou-se como *overhead* a relação entre o tamanho em *kilobytes* do *frame body* dos pacotes de gestão em que foram adicionados IE (*Beacons* e *Association Responses*) e a diferença de tempo ilustrado na figura 5.6 e o final de simulação:

$$overhead = \frac{sizeof\left(FrameBody\left(Beacon\right) + FrameBody\left(AssocResp\right)\right)}{time\left(total\right) - time\left(allGetFirstAssoc\right)} \tag{5.1}$$

O *overhead* de manutenção com recurso à métrica social é, em média, reduzido em aproximadamente 35% quando usado intervalo adaptativo entre *Beacons* (gráfico 5.7a) relativamente à situação em que é usado intervalo fixo (gráfico 5.7b). O aumento do *overhead* comparativamente ao cenário que não usa métrica social é fortemente influenciado pela estratégia adoptada em 4.2.2.10, daí que a componente principal do seu aumento seja o número de *Beacons* enviados. Assim, no caso do gráfico 5.7b, como o intervalo entre *Beacons* é maior que em 5.7a, o *overhead* na rede diminui.

Ao nível individual dos pacotes, o tamanho do *frame body* quando usada a métrica social nos *Beacons* e *Association Responses* é, no pior caso (i.e. simulação com 100 nós), 131 bytes e 20 bytes respectivamente, contra os 30 bytes e 14 bytes do padrão. O tamanho do *frame body*, em bytes, respeita a relação expressa na equação 5.2:

$$framebody = num_{Beacons} * \left(30 + nodes_{total} + \frac{nodes_{total}}{4}\right) + num_{AssocResp} * \left(14 + 6_{added}\right) \tag{5.2}$$



(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

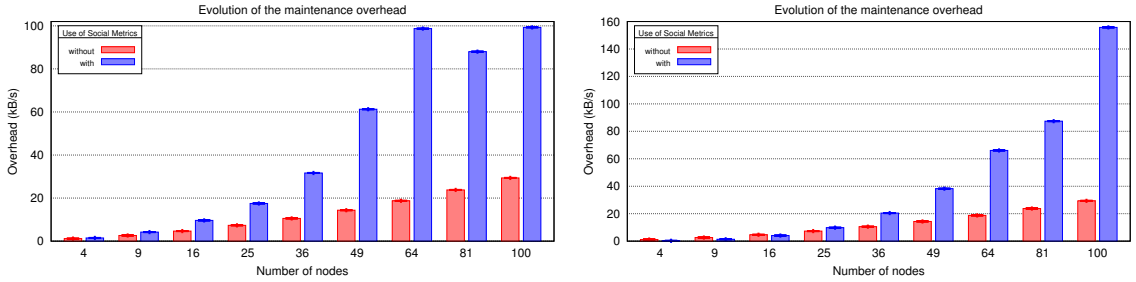
Figura 5.7: Evolução do *overhead* de manutenção

A figura 5.8 mostra o resultado em termos de *overhead* caso a estratégia implementada para descoberta de nós descrita na secção 4.2.2.10 pudesse evitar a sinalização dos nós ainda não conhecidos (índices do vector *KnownNodes* com valor '0'). Contudo, como já descrito, o simulador NS-3 impõe a criação de tipos de dados de tamanho pré-definido ao nível dos *Information Elements*, ou seja, a reserva de memória máxima necessária a este nível não pode ser feita dinamicamente. O *overhead* manteve a mesma relação da equação 5.1, porém, o tamanho do *frame body* passou apenas a ter em consideração o tamanho final das comunidades geradas por cada cenário, uma vez que este será o número de nós que cada elemento terá de disseminar pela sua comunidade (valor obtido através da figura 5.2):

$$framebody = num_{Beacons} * \left(30 + nodes_{total} + \frac{|communities_{final}|}{4}\right) + num_{AssocResp} * (14 + 6_{added}) \tag{5.3}$$

Em 5.8a é observável uma tendência para um valor final de *overhead* de aproximadamente 100kB/s, que representa menos de metade do *overhead* apresentado em 5.7a. Além de ser bastante inferior, o facto de estabilizar em torno de um valor final é muito importante em termos de escalabilidade com a rede. Nesta abordagem, o factor que mais influencia o *overhead* é o tamanho para o qual as comunidades tendem. Ou seja, de acordo com o gráfico 5.2a, o tamanho das comunidades no final de simulação tende a estabilizar, ou seja, o aumento de nós na rede implica que se gerem mais comunidades e não comunidades maiores.

Uma vez que no gráfico 5.8b, o intervalo entre *Beacons* é dinâmico e igual ou maior ao de 5.8a, poder-se-ia esperar que o *overhead* diminuisse. No entanto, como foi referido, na abordagem ilustrada pela equação 5.3, o factor que mais influencia o *overhead* é o tamanho das comunidades finais. De acordo com 5.2b, os nós tendem a agregar-se todos numa única comunidade, implicando que tenham de enviar informações relativamente a um maior número de nós, gerando um aumento do *overhead* da rede. É devido a este factor que os gráficos 5.7b e 5.8b são idênticos.



(a) Intervalo fixo entre *Beacons*

(b) Intervalo adaptativo entre *Beacons*

Figura 5.8: Evolução do *overhead* de manutenção

Na figura 5.9 é apresentado o número enviado de pacotes de gestão de dois tipos: *Beacon* e *Association Response*. Foram seleccionados estes tipos de pacotes por duas razões: serem aqueles em que foram introduzidos novos IEs, e por serem os pacotes de gestão fundamentais. A segunda razão é justificada pelo facto de que, enquanto os pacotes *Probe Request/Response* apenas são usados no modo de *scanning* activo, os tipos de pacotes apresentados na figura 5.9 são comuns aos dois modos (activo e passivo). Além disso, por cada *Association Request* há um *Association Response* (positivo ou negativo), pelo que apenas há necessidade de apresentar um deles.

Os valores apresentados nos gráficos 5.9a e 5.9b correspondem à média de pacotes de gestão enviados por cada nó durante a simulação.

Em 5.9a verifica-se que o número de *Beacons* é igual para as simulações com ou sem métrica social. A justificação é simples: o envio de *Beacons* é periódico e com o intervalo fixo entre os mesmos (e.g. o valor por defeito é 0.1s) não há qualquer impacto resultante do uso de métrica social.

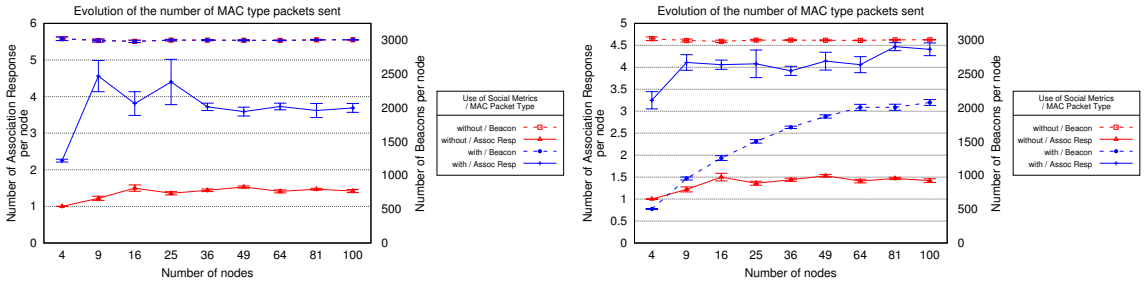
Ao nível do número de *Association Request*, o recurso à métrica social aumenta em aproximadamente 2 associações por nó relativamente ao padrão. As associações acrescentadas têm em vista o melhoramento da interligação dos nós ao nível MAC, possibilitando o melhor desempenho da rede ao nível da camada IP. No entanto, o impacto gerado ao nível das interrupções de conectividade provocadas pela mudança de associação dos nós não é muito relevante devido ao baixo número de ocorrências. Além disso, o facto deste fenómeno ter uma duração muito curta (i.e. o tempo entre a terminação da ligação anterior e a execução da nova é muito reduzido), contribui também para que o impacto da métrica social ao nível 3 melhore o desempenho global da rede.

Em 5.9b verifica-se que o número de *Beacons* é substancialmente inferior no cenário com métrica social, apresentando um comportamento logarítmico que tende para aproximadamente 2100 *Beacons* por nó. Este comportamento era esperado uma vez que, com o aumento do número de nós, estes tendem a agrupar-se numa única comunidade (figura 5.1b). O facto contribui, por um lado, para o aumento do valor da CQE (comunidade maior), e por outro, para a diminuição do valor da FIN (devido ao menor número de vizinhos em relação ao número total de nós na rede) que, de acordo com a equação 3.6, implica um menor espaçamento entre *Beacons*.

Ao nível do número de *Association Request*, o recurso à métrica social aumenta em média aproximadamente 3.5 associações por nó relativamente ao padrão. O facto de existir um ligeiro aumento no número de associações relativamente à figura 5.9a está relacionado com a diferente organização dos nós na rede (diferentes dimensões das comunidades de (a) para (b) - figura 5.2 - que influencia o próprio valor da métrica social). Tal como referido anteriormente, o número de associações forçadas

pela métrica social têm como objectivo o melhoramento da interligação dos nós ao nível MAC. A escolha ponderada dessa associação irá potenciar o desempenho da rede ao nível IP. Ainda assim, as mudanças de ligação entre nós irão provocar períodos de desconectividade local, sendo contudo muito curtos, devido à mudança de associação ser preparada antes de ser terminada. Além disso, o número destas ocorrências adicionais é apenas ligeiramente mais elevado que o padrão, pelo que é esperado que o impacto benéfico suplantará os instantes de conectividade intermitente.

Em ambos os gráficos 5.9a e 5.9b, é ainda importante considerar que o limiar definido para que ocorra a realização de uma nova associação devido a uma métrica social superior, terá uma forte contribuição no número total de associações executadas. Este factor aproximará ou afastará a curva '*with / Assoc Resp*' da curva '*without / Assoc Resp*' conforme seja maior ou menor respectivamente.



(a) Intervalo fixo entre *Beacons* (b) Intervalo adaptativo entre *Beacons*

Figura 5.9: Evolução do número de *Beacons* e *Associations*

5.2.2.2 Evolução das comunidades finais no tempo

Para complementar o estudo anterior que reflecte a análise completa da simulação para um determinado número de nós, foi elaborado um outro conjunto de resultados que mostra a evolução de determinados parâmetros das comunidades ao longo do tempo de simulação. Uma vez que há comunidades que são geradas e que, devido às posteriores associações entre nós, deixam de existir, esta nova análise foi apenas realizada às comunidades que estão presentes no final da simulação. Ainda assim, nas várias repetições, haverá simulações que poderão ter diferentes números de comunidades no final, o que implica que o processo de cálculo de médias e intervalos de confiança se adapte de forma autónoma a essa situação.

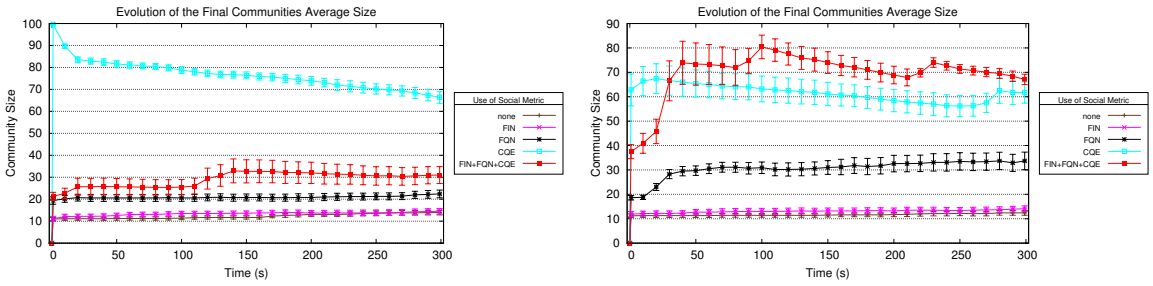
Neste caso apenas são apresentados os gráficos das simulações de 100 nós, existindo porém, o mesmo conjunto de resultados para simulações com diferentes números de nós (4, 9, 16, 25, 36, 49, 64 e 81 nós). Contudo, a sua discussão seria semelhante à que em seguida se apresenta.

Na figura 5.10 são apresentados os resultados da evolução ao longo do tempo do tamanho das comunidades que existem no final da simulação.

Do gráfico 5.10a é importante analisar a curva decrescente da simulação CQE. O pico inicial obtido no primeiro segundo é justificado por três factores: o intervalo entre *Beacons* ser fixo e suficiente para que até este instante de tempo tivessem sido enviados 10 *Beacons* por cada nó; ainda não ter decorrido muito tempo para que a mobilidade dos nós se tivesse feito reflectir ao nível das associações; e ainda o facto do parâmetro CQE ser fortemente agregador. Assim, até ao instante referido, a informação presente em todos os nós é coerente, mas começa a degradar-se à medida que a dinâmica dos nós

se faz sentir, uma vez que a propagação da informação numa comunidade de dimensão considerável é lenta. As restantes curvas tendem para um valor estável, correspondente ao tamanho médio das comunidades finais. O valor considerável dos intervalos de confiança da simulação FIN+FQN+CQE reflecte a diferente dinâmica de crescimento das comunidades obtidas no final de simulação para as várias repetições executadas.

No gráfico 5.10b, devido ao intervalo dinâmico entre *Beacons*, o pico inicial da simulação CQE desaparece, passando a sua curva a variar em torno de um valor médio em vez de decrescer constantemente como em 5.10a. Isto acontece porque o intervalo entre actualizações de informação consecutivas (envio de *Beacons*) é maior. A simulação com a FQN apresenta um crescimento médio de aproximadamente mais 15 nós nas comunidades finais relativamente a 5.10a, ou seja, passou a ter um maior poder de agregação dos nós, facto justificado fundamentalmente pela eliminação do factor aleatório que simulava as características de *hardware* na equação 3.4. As simulações *none* e FIN mantêm-se semelhantes, reforçando a ideia da fraca influência da FIN no valor da métrica social. O valor considerável dos intervalos de confiança, especialmente nas simulações CQE e FIN+FQN+CQE, reflecte a diferente dinâmica de crescimento das comunidades obtidas no final de simulação para as várias repetições executadas. A simulação FIN+FQN+CQE cresce de forma mais lenta mas atinge um valor final mais elevado comparativamente a 5.10a, facto justificado tanto pela baixa contribuição da FIN no valor final da métrica social, como pela menor distinção entre nós através da FQN (agora só quantifica a qualidade da ligação de associação) e, por isso a CQE passa a ser o factor de maior influência na métrica social.



(a) Intervalo fixo entre *Beacons* (b) Intervalo dinâmico entre *Beacons*

Figura 5.10: Evolução do tamanho das comunidades finais em simulações com 100 nós

É importante comparar também estes resultados com os da figura 5.2, nomeadamente nos valores do *eixo* y : $x = 100\,nós$, dado que a figura 5.2 também trata apenas das comunidades finais. Desta forma, é possível obter informação acerca da influência directa do tamanho da comunidade na propagação da informação em cenários dinâmicos, ou seja, conseguir de alguma forma quantificar a coerência da informação entre os vários nós. Nesta análise, a informação indicada pela figura 5.2 deve ser considerada como o tamanho real das comunidades no final da simulação, e a figura 5.10 como sendo a média da informação que cada nó indica acerca do tamanho da comunidade em que se encontra em cada instante.

Verifica-se que, comparando 5.2a com 5.10a, as simulações em que o número de nós em cada comunidade é baixo (até aproximadamente 20 nós - simulações *none*, FIN e FQN), há uma elevada coerência entre a informação média de cada nó e a real. Na simulação FIN+FQN+CQE as comunidades tendem a ter um tamanho médio final de aproximadamente 35 nós, existindo uma diferença de

aproximadamente 5 nós entre o tamanho real e aquele que é, em média, reportado por cada nó. Este é um valor aceitável dada a mobilidade constante dos nós, a possibilidade de mudanças de associações baseadas em melhores métricas sociais, e a perda destas devido ao alcance limitado. Quando o tamanho da comunidade é muito elevado (os 100 nós da simulação tendem a agrupar-se numa mesma comunidade), como acontece no cenário em que apenas é usada a CQE, e devido também à elevada dinâmica dos cenários já referida, a diferença média entre a informação do tamanho real e o que os vários nós reportam aumenta para aproximadamente 35 nós. Este valor não pode ser desprezado pois representa uma grande parte dos nós da comunidade, concluindo-se que a informação demora muito tempo a ser propagada, e portanto, a actualização e sincronização do conhecimento entre todos os nós é lenta relativamente à dinâmica do próprio sistema. Deste ponto de vista, a gestão de comunidades de tamanho considerável é mais difícil.

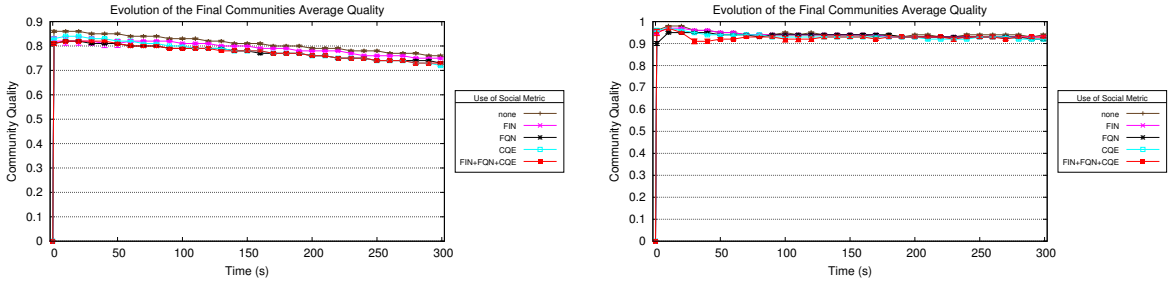
A análise comparativa das figuras 5.2b e 5.10b é semelhante, podendo-se acrescentar que na simulação com apenas a CQE, a diferença média entre o valor real e aquele que é reportado pelos vários nós é aproximadamente 20 nós e que, quando os nós tendem a ficar todos na mesma comunidade (simulação FIN+FQN+CQE) a diferença aumenta para aproximadamente 35 nós, tal como acontece na análise realizada para a simulação com a CQE das figuras 5.2a e 5.10a. É possível acrescentar ainda que, apesar do intervalo entre actualizações da informação ser ligeiramente maior (intervalo dinâmico entre *Beacons*), não há relação directa com o nível de coerência da informação na comunidade.

Na figura 5.11 são apresentados os resultados da evolução ao longo do tempo do valor médio das FQN nas comunidades existentes no final de simulação. É um facto que, sem os mecanismos implementados, não existe forma de medição da "qualidade" das associações. Ou seja, a medição desta característica deve-se aos requisitos do parâmetro FQN da métrica social. No entanto, a funcionalidade foi deixada activa na simulação *none* de forma a ser possível ter um ponto de comparação.

No gráfico 5.11a é importante salientar que a diminuição ao longo do tempo da "qualidade" das associações nas comunidades se deve à existência de um factor de decréscimo constante presente na equação 3.4, que apesar de pequeno, se reflecte no valor final das FQNs. O motivo para não se verificarem quebras abruptas nas curvas do gráfico é justificado por dois factores: o facto de ser apresentada uma média das FQNs dos vários nós e o período de amostragem ser 1 segundo. Ainda que o período de amostragem pudesse ser diminuído, pelo gráfico 5.5a verifica-se que o tempo que um nó demora a readquirir associação é aproximadamente 0.03s para 100 nós. Esse intervalo de tempo vai desde o valor mínimo da FQN até esta voltar ao valor máximo. Ou seja, um observador exterior terá dificuldades em detectar estas variações e, ainda assim, estas serão atenuadas pela média com as restantes FQNs da comunidade.

No gráfico 5.11b a interpretação é semelhante; porém, o decréscimo constante desaparece porque, mais uma vez, a parte da equação 3.4 que simulava os recursos do nó foi retirada de modo a provar que este era o factor responsável pela diminuição constante da "qualidade" da comunidade. Observa-se ainda que todas as simulações tendem para o mesmo valor médio; contudo, a simulação FIN+FQN+CQE apresenta uma ligeira oscilação num período inicial (até aproximadamente 80s) devido a dois factores em simultâneo: o maior número de desassociações/associações provocadas pela evolução da métrica social, e a mais lenta propagação da informação devido à agregação dos nós numa única comunidade (note-se a tendência das curvas FIN+FQN+CQE e CQE se seguirem em 5.11b e em 5.1b).

Uma vez que, pelo gráfico 5.11b, o valor final é semelhante para os vários tipos de cenários, a diferença entre a simulação *none* e a FIN+FQN+CQE ($\triangle \approx 0.05$) em 5.11a apenas pode ser justificada pelo parâmetro aleatório usado em 3.4 como forma de simular os recursos do nó.



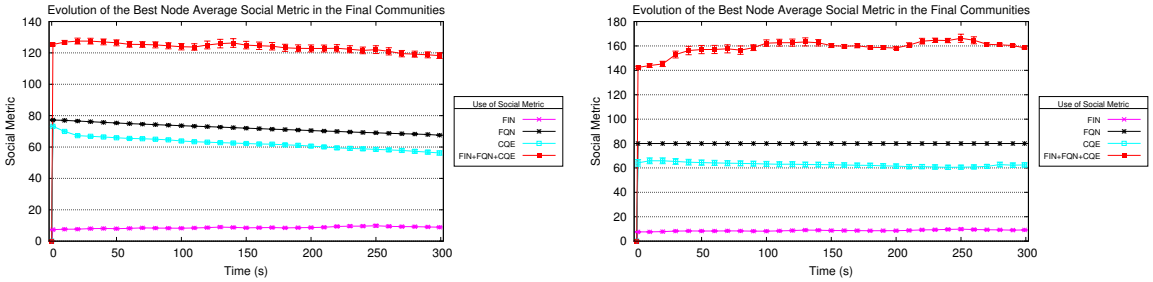
(a) Intervalo fixo entre *Beacons* (b) Intervalo dinâmico entre *Beacons*

Figura 5.11: Evolução da média das FQN nas comunidades finais em simulações com 100 nós

Na figura 5.12 são apresentadas informações relevantes para a implementação do processo da eleição, através da evolução do valor mais elevado da métrica social, ou seja, do nó potencialmente líder em cada instante. Através da indicação sobre o valor e sobre a curva de tendência deste parâmetro é possível antecipar mecanismos para que o processo de eleição não provoque mudanças constantes de líder (e.g. definir limites relativamente ao valor da métrica social ou aos seus parâmetros individuais (FIN, FQN e CQE) que provocam mudança de líder, duração do tempo como líder, etc).

No gráfico 5.12a confirma-se a fraca influência da FIN no valor da métrica social e uma diminuição praticamente constante nas simulações FQN e CQE, uma vez que a FQN tem um factor de decréscimo ao longo do tempo (já referido) e há uma relação de influência directa desta na CQE. Na simulação com a métrica social completa (FIN+FQN+CQE), a curva varia em torno de aproximadamente 125, não apresentando variações bruscas. Este facto é bastante importante porque demonstra que, independentemente do instante de tempo de simulação, haverá pelo menos um nó (potencial líder) que terá os valores de métrica social apresentados no gráfico.

A principal diferença no gráfico 5.12b é o facto de a FQN e a CQE se manterem praticamente constantes (o factor de perdas constantes da FQN foi eliminado como já referido). Devido não só ao maior espaçamento entre o envio de *Beacons* consecutivos, como ao maior nível de agregação dos nós (comparativamente a 5.12a), a simulação FIN+FQN+CQE tende mais lentamente para o valor final, ou seja, é necessário mais tempo até que os nós reúnam o conhecimento suficiente para que atinjam o valor final apresentado para a métrica social (≈ 160). É importante justificar que a diferença entre o valor médio da métrica social em 5.12a e em 5.12b se deve ao número final de comunidades (ver figura 5.1). Ou seja, dependendo do nível de agregação dos nós, a métrica social evolui de formas distintas e tende para valores ligeiramente diferentes.



(a) Intervalo fixo entre *Beacons* (b) Intervalo dinâmico entre *Beacons*

Figura 5.12: Evolução da melhor métrica social nas comunidades finais em simulações com 100 nós

A figura 5.13 é complementar à 5.12, apresentando a evolução individual dos parâmetros (FIN, FQN e CQE) que constituem o valor da métrica social. Tal como referido na secção 3.4.5, estes encontram-se normalizados no intervalo [0;1], uma vez que são apresentados antes da multiplicação pelos pesos w_1, w_2 *e* w_3 respectivamente.

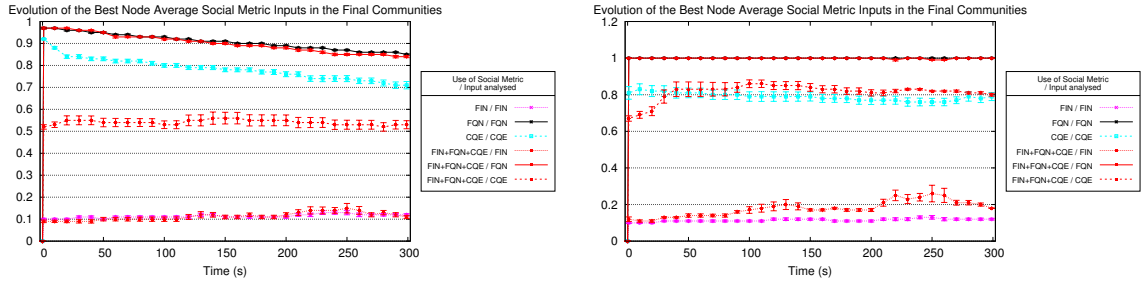
No gráfico 5.13a é possível perceber a influência individual e relativa de cada um dos parâmetros da métrica social. Analisando a simulação FIN+FQN+CQE, a FIN é o parâmetro que apresenta menor contribuição, confirmando as análises efectuadas aos resultados anteriores, pelo que pode ser necessário ajustar o peso correspondente (w_1). O parâmetro CQE tem uma influência intermédia e a FQN é a parcela com maior contribuição; no entanto a sua relevância diminui com o tempo devido ao já referido factor de decréscimo simulado dos recursos do nó. O facto da FQN ser o parâmetro que mais contribui para o valor final da métrica social não implica necessariamente que seja o factor dominante, por exemplo, ao nível da eleição. A justificação é simples: como a FQN será constante (exceptuando os períodos de tempo em torno da perda de ligação) para os vários nós, a sua contribuição será no valor absoluto e não no valor relativo da métrica social.

É importante ainda analisar a simulação CQE / CQE, na qual a contribuição do parâmetro em análise é superior à observada em FIN+FQN+CQE / CQE, devido ao diferente número de comunidades gerado por cada uma das simulações. Ou seja, em CQE / CQE há uma acção mais agregadora, criando comunidades maiores relativamente a FIN+FQN+CQE / CQE, o que implica que o próprio valor do parâmetro em análise (CQE) aumente. Como a simulação FIN+FQN+CQE tende a dividir as comunidades a partir de um certo tamanho (ver 5.2a), a contribuição do parâmetro CQE neste cenário é menor.

No gráfico 5.13b, uma vez que a parcela da FQN que simulava o decréscimo dos recursos do nó ao longo do tempo foi retirada, a FIN+FQN+CQE ganhou poder de agregação e, tal como já discutido anteriormente, a métrica social passou a seguir o comportamento agregador da CQE. Ou seja, apesar da FQN se manter praticamente no valor máximo ao longo da simulação, isto ocorre para todos os nós (à excepção de intervalos de tempo curtos em que a ligação de associação se degrada ou se perde efectivamente). Portanto, neste caso não é um factor diferenciador, isto é, tem uma contribuição para o aumento do valor absoluto da métrica social mas não em termos relativos. Assim, através deste gráfico, é possível indicar que: a FIN mantém uma fraca influência no valor da métrica social; a FQN, apesar de ter uma forte contribuição, é praticamente idêntica para todos os nós; portanto, a maior influência da métrica social é a CQE.

A partir destas informações é possível antecipar a forma como evolui a métrica de um nó potenci-

almente líder e qual a contribuição individual dos vários factores.



(a) Intervalo fixo entre *Beacons* (b) Intervalo dinâmico entre *Beacons*

Figura 5.13: Evolução dos componentes da melhor métrica social das comunidades finais em simulações com 100 nós

5.3 Redes com fios

5.3.1 Apresentação e discussão de resultados

Com os resultados seguintes pretende-se comparar o mecanismo proposto (*INM-Discovery*) com as abordagens de descoberta do OSPF, CDP e Fing, em termos de *overhead* de mensagens e tempo de convergência. O *overhead* representado é a percentagem de pacotes do processo de descoberta no tráfego total da rede; o tempo de convergência é o tempo necessário até que todos os nós na rede sejam descobertos. O tráfego de rede em situação estacionária (i.e. quando não há mecanismos de descoberta em execução) é aproximadamente 900 pacotes.

Uma vez que os mecanismos de descoberta do OSPF, CDP e Fing usam uma frequência fixa de pacotes *Hello*, foram realizadas medições com diferentes intervalos entre pacotes (1, 5, 10 e 20 s). Já no mecanismo de descoberta *INM-Discovery*, o intervalo entre pacotes *Hello* é 5 s durante o *bootstrapping* e adaptativo no restante tempo, de acordo com o número de *INM_Seekers* na *Partial View*. O número inicial de entidades *INM_Seekers* e *INM_Hiders* foi aleatoriamente atribuído.

Os resultados 5.14, 5.15 e 5.16 foram recolhidos com a topologia de rede ilustrada na figura 4.18, usando tempos de observação de 60s e 5 repetições independentes. Os resultados apresentam intervalos de confiança a 90%.

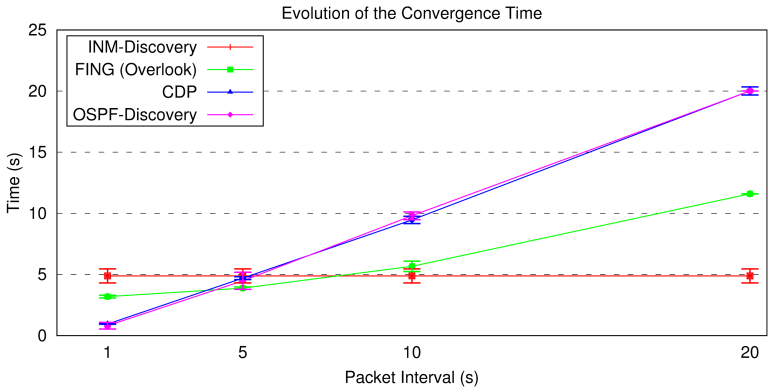


Figura 5.14: Evolução do tempo de convergência da descoberta

Pela figura 5.14 é visível que, nas situações em que o intervalo entre *Hello* é 1 e 5 s, o tempo de convergência da descoberta de todos os nós na rede é superior através do *INM-Discovery* comparativamente aos restantes protocolos. Ainda assim, enquanto que o tempo para o protocolo *INM-Discovery* se mantém constante nos vários cenários, nos restantes ele aumenta proporcionalmente com o intervalo entre pacotes *Hello*. No caso do CDP e do OSPF-Discovery, o tempo de descoberta tende a ser semelhante ao intervalo entre pacotes *Hello*. No caso do Fing, com 20 s de intervalo entre pacotes *Hello*, tende a demorar mais do dobro do tempo do *INM-Discovery*. O resultado é explicado pelo intervalo adaptativo do *INM-Discovery*, ou seja, independentemente do intervalo fixo que é colocado na fase de *bootstrapping* (e.g. 5 s), tenderá a adaptar-se automaticamente de acordo com os critérios estabelecidos.

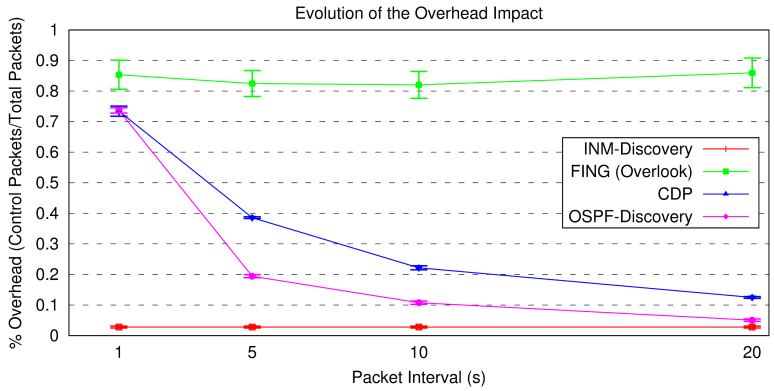


Figura 5.15: Evolução do *overhead* de mensagens na rede

Pela figura 5.15 é possível verificar que o *overhead* introduzido tanto pelo *INM-Discovery* como pelo Fing é praticamente constante para os vários cenários, enquanto que no CDP e no OSPF-Discovery é inversamente proporcional ao intervalo entre pacotes *Hello*. O elevado *overhead* do Fing resulta do seu mecanismo de descoberta se basear no protocolo ARP e, portanto, no envio de ARP-Requests em broadcast para todos os IPs da sub-rede. O protocolo *INM-Discovery* tem um impacto bastante inferior a todos os protocolos analisados, facto explicado tanto pela colaboração entre entidades *INM_Hiders* e *INM_Seekers*, como pelo papel dinâmico e despoletado por eventos que cada entidade desempenha.

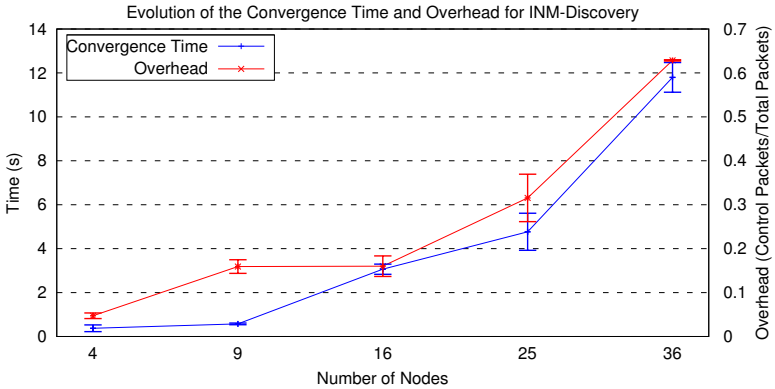


Figura 5.16: Evolução do comportamento do protocolo *INM-Discovery* com o número de nós na rede

Através da figura 5.16 pode-se afirmar que, em cenários com 4 e 9 máquinas virtuais, o tempo de convergência é semelhante, no entanto o *overhead* aumenta. Uma possível explicação pode ser o número de *INM_Seekers* existentes na rede numa fase inicial ser proporcional ao número máquinas virtuais em ambos os cenários, fazendo com que o tempo de convergência da informação se mantenha idêntico, à custa de um aumento do *overhead* devido ao maior número de mensagens trocadas (pacotes *Hello* e *NodeInfo*).

Acontece a situação inversa nos cenários com 9 e 16 máquinas virtuais, ou seja, o tempo de convergência aumenta mas o *overhead* mantém-se constante. A explicação possível é análoga à anterior: a quantidade de *INM_Seekers* atribuída probabilisticamente na fase inicial de cada cenário deve ser semelhante, fazendo com que o *overhead* se mantenha idêntico, aumentando com isso o tempo necessário para a convergência da informação.

Nos cenários com o número máquinas virtuais entre 16 e 36, o tempo de convergência e o *overhead* apresentam um crescimento idêntico, que pode ser justificado por uma relação constante entre o número de *INM_Seekers* e de máquinas virtuais de cada cenário.

De facto, a quantidade de *INM_Seekers* numa fase inicial da rede determina desempenho do protocolo, tendo de existir um compromisso entre *overhead* e tempo de convergência. É importante realçar também que o papel de *INM_Seeker* é atribuído probabilisticamente e que cada *INM_Hider* poderá converter-se em *INM_Seeker* após um determinado período de tempo, caso não seja entretanto contactado. Assim, em redes de pequena dimensão, os *INM_Hiders* serão rapidamente contactados, fazendo com que a convergência de informação ocorra num intervalo de tempo reduzido. No entanto, quando a dimensão da rede aumenta, passam a existir múltiplos saltos entre as entidades, exigindo-se por isso colaboração entre os nós para uma maior eficiência nas comunicações. Contudo, tanto em redes de pequena como de média dimensão, o número de *INM_Seekers* presentes numa fase inicial será sempre o factor com maior contribuição no desempenho do protocolo.

5.4 Conclusões

De forma sucinta, as conclusões que se podem retirar da discussão individual dos resultados obtidos anteriormente para as redes sem fios são:

- Em relação ao número e à dimensão das comunidades existentes na rede, terá de existir um compromisso, na medida em que:
  - Um número elevado de comunidades, ou seja, comunidades de dimensão reduzida implicam:
    - menor inconsistência da informação devido à rápida propagação (fig. 5.10);
    - menor *overhead* devido ao menor tamanho das comunidades (fig. 5.8);
    - no entanto, há possibilidade de constante re-aprendizagem devido às mudanças de comunidade.
  - Um número reduzido de comunidades, ou seja, comunidades de dimensão elevada originam:
    - vantagens do ponto de vista da gestão, uma vez que há garantia que todos os nós estão interligados entre si, ou seja, é possível chegar a qualquer nó partindo de qualquer outro. Este facto potenciará menores atrasos no encaminhamento de pacotes de dados;

    - reaprendizagens menos recorrentes dado que o conhecimento de cada nó sobre a rede já foi previamente adquirido, ocorrendo apenas actualizações de informações específicas. A separação dos nós em domínios poderá ser deixada para o gestor ao nível IP;
    - maior impacto em termos de *overhead*, uma vez que existem mais nós a serem conhecidos e, portanto, uma maior quantidade informação a ser partilhada entre os nós. No entanto, abordagens probabilísticas na disseminação do conhecimento podem ser vantajosas a este nível.
  - Em suma, o número ideal de comunidades depende das características que se pretendam privilegiar na rede, uma vez que várias comunidades apresentam melhores resultados a este nível mas podem ser penalizadas ao nível do encaminhamento de pacotes de dados (terá de ser o nó a perceber que o destino não pertence à sua comunidade e a enviar o pacote para outra).

- Em relação ao número de associações é importante referir que:
  - Um número baixo de associações implicam maior estabilidade na rede, contribuindo para a convergência mais rápida dos protocolos de encaminhamento ao nível IP. Contudo, de acordo com o protocolo 802.11 MAC, as associações são realizadas ao primeiro nó no alcance sem qualquer outro critério, ou seja, as ligações não são potencializadas, ficando a ideia que o desempenho da rede poderá ser melhorado;
  - O número elevado de trocas de associação que a métrica social provoca está directamente relacionado com o valor do limiar (+10%) definido para a realização da mudança de associação quando uma melhor métrica é detectada num vizinho. Ou seja, a afinação deste parâmetro é fundamental para a maximização do desempenho global da rede. Apesar do maior esforço na camada IP devido às reconfigurações das tabelas de encaminhamento, principalmente na fase inicial, os períodos de desconectividade ao nível MAC são mínimos. Assim, a introdução de decisão ao nível da escolha da associação recorrendo a critérios que avaliam determinadas condições e características do nó e da sua comunidade, poderão melhorar o desempenho da rede.

- Relativamente ao impacto na rede, este foi analisado de várias perspectivas:
  - Em termos de *overhead*, considerando apenas o tamanho do campo *Frame Body* dos *Beacons* e *Association Responses* (no qual se introduziram os novos IEs):
    - na abordagem implementada, a qual está limitada devido à definição estática imposta pelo simulador ao nível dos IEs, o *overhead* é fundamentalmente imposto pelos *Beacons*, nomeadamente ao nível das listas de nós para determinação do tamanho e da FQN média da comunidade de forma distribuída. Pode ainda afirmar-se que, nesta abordagem, o *overhead* introduzido é independente do número de comunidades formadas;
    - caso fosse possível introduzir alguma dinâmica no tamanho destas listas, enviando apenas IDs pertencentes à comunidade do nó, o *overhead* passaria a ser proporcional ao tamanho da comunidade, ou seja, tendo em conta apenas este aspecto, o *overhead* introduzido pela solução proposta tende para a abordagem sem métricas sociais quanto menor for a comunidade.

  - Em termos de números de pacotes de gestão enviados (considerando apenas *Beacons* e *Association Responses* pela mesma razão apresentada anteriormente):
    * Com o intervalo fixo entre *Beacons*, tanto na solução com métrica social como no cenário sem métrica, o número de *Beacons* enviados por cada nó é o mesmo, uma vez que estes são periódicos. Ao nível de *Association Responses*, recorrendo à métrica social há mais ≈ 2 associações por cada nó, devido à escolha ponderada e dinâmica da ligação de associação;
    * Com o intervalo adaptativo entre *Beacons*, a solução com métrica social apresenta um comportamento logarítmico ao nível do número de *Beacons* enviados, que tende para ≈ 2100 por nó, contra o comportamento constante da abordagem sem métrica social, que apresenta ≈ 3000 *Beacons* por nó. Ao nível de *Association Responses*, recorrendo à métrica social há mais ≈ 3.5 associações por cada nó. O ligeiro aumento relativamente ao cenário com intervalo fixo entre *Beacons* deve-se à diferente organização dos nós na rede, nomeadamente ao nível do número de comunidades para o qual os nós tendem.

- Em relação à análise realizada ao parâmetro de entrada do processo de eleição (i.e. o valor da métrica social mais elevada), verifica-se que a sua evolução ao longo do tempo tende a manter-se numa faixa estreita de valores. Esta constatação é extremamente importante, na medida em que permite antecipar vários mecanismos que evitem a mudança constante de líder que resultaria certamente numa degradação do desempenho da rede. Por exemplo, definição de limites relativamente ao valor da métrica social ou aos seus parâmetros individuais (FIN, FQN e CQE) que provocam a mudança de líder, período mínimo de tempo como líder após ser eleito, histórico de tempo que o nó já esteve como líder, etc. Em termos de valor absoluto, o valor máximo da métrica social está directamente dependente dos pesos individuais w_1, w_2, w_3 atribuídos a cada uma das suas parcelas, pelo que todos os mecanismos relacionados com a eleição se deverão adaptar automaticamente ao valor destes, de modo a que diferentes cenários sejam directamente comparáveis.

Para as redes com fios, pode concluir-se de forma resumida que:

- O mecanismo de *bootstrapping* e de descoberta proposto apresenta características fundamentais em sistemas distribuídos como (1) intervalo adaptativo entre pacotes *Hello* de acordo com a quantidade de entidades *INM_Seekers* presentes na rede; (2) cooperação entre entidades na partilha dos seus repositórios locais; (3) papel desempenhado por cada entidade é alterado dinamicamente, de acordo com as condições da rede;

- O protocolo de descoberta apresenta-se mais eficiente comparativamente aos protocolos de descoberta CDP, OSPF e Fing, tanto ao nível de tempo de convergência, como de *overhead* de mensagens. Demonstra-se, desta forma, que o seu impacto na rede é bastante reduzido;

- O comportamento do protocolo em termos de *overhead* e tempo de convergência é fortemente influenciado pelo número de *INM_Seekers* existentes numa fase inicial da rede. Existe um compromisso entre estes parâmetros, ou seja, a quantidade inicial de *INM_Seekers* é inversamente proporcional ao tempo de convergência e proporcional ao *overhead*. No entanto, o papel de cada entidade é atribuído inicialmente de forma probabilística, e é dinâmico, ou seja, um *INM_Hider* pode passar a *INM_Seeker* dependendo de factores específicos já referidos anteriormente.

Capítulo 6

Conclusão e linhas futuras de investigação

A presente Dissertação pretende, através dos conceitos implementados e resultados alcançados, dar o seu contributo ao nível do desenvolvimento de novas abordagens que permitam uma gestão eficiente da rede.

Através do trabalho apresentado foi possível implementar o modelo de comunicação *ad hoc*, fundamental à gestão distribuída da rede, bem como vários mecanismos autonómicos (e.g. *bootstrapping*, descoberta, propagação de informações, etc), que constituem uma base indispensável ao desenvolvimento futuro de funcionalidades que introduzam maior capacidade de gestão na rede (e.g. decisão).

Ao nível das redes sem fios foi necessário incluir alguns complementos ao protocolo 802.11 MAC para cumprir os requisitos dos mecanismos propostos, nomeadamente a introdução de novos *Information Elements* ao nível de alguns pacotes de gestão (*Beacon* e *Association Response*), permitindo a associação baseada em métricas sociais estabelecidas.

Ao nível das redes com fios, o protocolo *INM-Discovery* apresenta papéis dinâmicos, colaboração entre as entidades da rede e intervalo adaptativo entre pacotes *Hello*, características que contribuem para um desempenho superior em termos de *overhead* e de tempo de convergência comparativamente aos restantes protocolos de descoberta analisados.

Como forma de analisar as soluções propostas, elaboraram-se alguns cenários teste dos quais se retiraram os principais resultados. Outras variações dos mesmos poderiam ser realizadas, no entanto os cenários apresentados foram definidos após alguma ponderação e observação de resultados preliminares de outras simulações. A discussão e conclusão dos mesmos reflecte a compreensão dos conceitos envolvidos e os compromissos que são necessários definir.

Em termos de linhas futuras de investigação nas redes sem fios, os principais pontos são:

- Finalização do processo de eleição:
  - consenso, sinalização e funcionalidades específicas dos líderes.
- Comunicação entre líderes de domínios:
  - cooperação e disseminação de informações entre domínios.

- Implementação da camada IP:
  - geração de tráfego na rede e avaliação da influência dos mecanismos implementados em termos de *overhead*, atraso fim-a-fim e *throughput;*
  - análise à variação de parâmetros específicos (e.g. limiar do valor da métrica social que origina mudança de associação e pesos w_1, w_2, w_3 da métrica social) no desempenho da rede.
- Implementação de decisões locais para gestão global da rede:
  - mecanismos de decisão (*reinforcement learning*), consenso, impacto das mesmas na rede, etc.
- Algoritmo (já modelado) que exige menor *overhead* para conseguir:
  - a contagem de nós e estimativa da qualidade média em cada comunidade.
- Interface gráfica para mais fácil interacção com o utilizador;
- Estudo de mecanismos para garantir requisitos de QoS, segurança e poupança de energia em redes *ad hoc.*

Ao nível das redes com fios, algumas funcionalidades não discutidas nesta Dissertação encontram-se já em funcionamento, como:

- *Cross-compile* do protocolo *INM-Discovery* em equipamentos reais;
- Avaliação do protocolo numa *testbed* mista, isto é, com equipamentos reais e máquinas virtuais.

Contudo, podem destacar-se as seguintes linhas futuras com as principais funcionalidades a implementar:

- Avaliação do protocolo ao nível da escalabilidade comparativamente a outros protocolos de descoberta;
- Implementação de mecanismos de disseminação de decisões e sincronização da informação de gestão;
- Implementação da funcionalidade para a detecção da saída de nós;
- Introdução de tolerância a falhas na solução proposta.

Bibliografia

[1] J. D. J. M. Fedor, M. Schoffstall, "Rfc 1157 - simple network management protocol (snmp)," tech. rep., 1990.

[2] U. Warrier and et al., "Rfc 1189 - common management information services and protocols," tech. rep., 1990.

[3] A. M. Barotto and et al., "Distributed network management using snmp, java, www and corba," *Journal of Network and Systems Management*, vol. 8, no. 4, pp. 483–497, 2000.

[4] T. M. Chen and et al., "A model and evaluation of distributed network management approaches," *IEEE Journal on Selected Areas in Communications*, vol. 20, p. 8, May 2002.

[5] A. Modarressi and S. Mohan, "Control and management in next-generation networks: challenges and opportunities," *Communications Magazine, IEEE*, vol. 38, no. 10, pp. 94–102, 2000.

[6] I. Akyildiz, J. Mcnair, J. Ho, H. Uzunalioglu, and W. Wang, "Mobility management in next-generation wireless systems," *Proceedings of the IEEE*, vol. 87, no. 8, pp. 1347–1384, 1999.

[7] M. Salehie and L. Tahvildari, "Autonomic computing: emerging trends and open problems," *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–7, 2005.

[8] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. Foghlu, W. Donnelly, and J. Strassner, "Towards autonomic management of communications networks," *Communications Magazine, IEEE*, vol. 45, pp. 112–121, oct 2007.

[9] I. G. B. Yahia and E. Bertin, "Towards autonomic management for next generation services," pp. 38–38, 16-18 July 2006.

[10] S. Schmid, M. Sifalakis, and D. Hutchison, "Towards autonomic networks," *In proceedings of 3rd Annual Conference on Autonomic Networking, Autonomic Communication Workshop (IFIP AN/WAC), Paris France*, September 2006.

[11] S. White, J. Hanson, I. Whalley, D. Chess, , and J. Kephart, "An architectural approach to autonomic computing," *In Proceedings of the International Conference on Autonomic Computing - ICAC, IEEE Computer Society*, pp. 2–9, 2004.

[12] M. Aldinucci and et al., "Towards hierarchical management of autonomic components: A case study," pp. 3–10, 18-20 Feb. 2009.

[13] M. Franzke and G. Hasslinger, "D4.2 - in-network management concept," *FP7 ICT 2007 1 216041 4WARD Architecture and Design for the Future Internet*, 2009.

[14] D. Dudkowski and et al., "Architectural principles and elements of in-network management," pp. 529–536, 1-5 June 2009.

[15] M. Flood, "The hide and seek game of von neumann," p. 107 to 109, 1972.

[16] L. Guardalben, V. Mirones, P. Salvador, and S. Sargento, "A cooperative hide and seek discovery over in network management," *2nd IFIP/IEEE International Workshop on Management of the Future Internet (ManFi 2010), IEEE/IFIP Network Operations and Management Symposium*, January 2010.

[17] M. Subramanian, T. Gonsalves, and U. Rani, *Network Management: Principles and Practice*. Pearson Education India, 2010.

[18] A. Clemm, *Network Management Fundamentals: A Guide to understanding how network management technology really works*. Cisco Press, 2007.

[19] ITU-T, "M.3010 principles for a telecommunications management network," 1996.

[20] ITU-T, "M.3400 tmn management functions," 1997.

[21] J. Martin-Flatin, S. Znaty, and J. Hubaux, "A survey of distributed enterprise network and systems management paradigms," *Journal of Network and Systems Management*, vol. 7, no. 1, pp. 9–26, 1999.

[22] L. Guardalben, "Communication between nodes and domains over in-network management paradigm." PhD thesis plan, 2011.

[23] S. Dharwadkar and N. Masood, "Next generation network," in *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*, pp. 1–4, IEEE.

[24] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing: degrees, models, and applications," *ACM Comput. Surv.*, vol. 40, pp. 7:1–7:28, August 2008.

[25] J. Ding, *Advances in Network Management*. CRC Press, 2010.

[26] H. de Meer, P. Wuchner, and A. Houyou, "Self-organizing systems: New trends in architectures and performance modeling," in *International Workshop on Self-Organizing Systems (IWSOS)*, Stembro 2006.

[27] G. Sawma, R. Ben-El-Kezadri, I. Aib, and G. Pujolle, "Autonomic management for capacity improvement in wireless networks," 2009.

[28] M. Serrano, S. van Der Meer, V. Holum, J. Murphy, and J. Strassner, "Federation, a matter of autonomic management in the future internet," pp. 845–849, 19-23 April 2010.

[29] B. Jennings, R. Brennan, W. Donnelly, S. N. Foley, D. Lewis, D. O'Sullivan, J. Strassner, and S. van der Meer, "Challenges for federated, autonomic network management in the future internet," pp. 87–92, 1-5 June 2009.

[30] A.-L.-G. Ramy Farha, Myung Sup Kom and J. W.-K. H. Yu Cheng, "A generic architecture for autonomic service and network management," *Computer Communications*, vol. 29, pp. 3691–3709, November 2006.

[31] A. Project, "Autonomic network architecture (ana)," *Project funded by the European Union Information Society Technologies Framework Programme 6 (EU IST FP6)*, 2010.

[32] A. Manzalini and F. Zambonelli, "Towards autonomic and situation-aware communication services: the cascadas vision," *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS)*, pp. 383–388, 2006.

[33] HAGGLE, "A novel communication paradigm for autonomic opportunistic communication." Janeiro 2006.

[34] J. Strassner, N. Agoulmine, and E. Lehtihet, "Focale: A novel autonomic networking architecture," 2006.

[35] "Autonomic internet project." `http://www.ist.autoi.eu/autoi/`.

[36] S. Schuetz, K. Zimmermann, G. Nunzi, S. Schmid, and M. Brunner, "Autonomic and decentralized management of wireless access networks," *Network and Service Management, IEEE Transactions on*, vol. 4, no. 2, pp. 96–106, 2007.

[37] H. Derbel, N. Agoulmine, and M. Salaün, "Anema: Autonomic network management architecture to support self-configuration and self-optimization in ip networks," *Computer Networks*, vol. 53, no. 3, pp. 418–430, 2009.

[38] 4WARD, "4ward project http://www.4ward-project.eu/," 2008.

[39] R. Roth and F. Wolff, "D4.1 - definition of scenarios and use cases," *FP7 ICT 2007 1 216041 4WARD Architecture and Design for the Future Internet*, 2009.

[40] M. Achemlal and P. Aranda, "D2.1 - technical requirements," *FP7 ICT 2007 1 216041 4WARD Architecture and Design for the Future Internet*, 2008.

[41] G. Lanzhi, Z. Chunhong, J. Yang, and L. Lichun, "An o(1) lookup and decentralized bootstrapping peer to peer sip system," pp. 1–2, 10-13 Jan. 2009.

[42] C. G. Dickey and C. Grothoff, "Bootstrapping of peer-to-peer networks," pp. 205–208, July 28 2008-Aug. 1 2008.

[43] A. Mawji and H. Hassanein, "Implementation of bootstrapping for p2p overlays in manets," pp. 14–17, 12-14 May 2010.

[44] D. Doval and D. O'Mahony, "Overlay networks: A scalable alternative for p2p," vol. 7, pp. 79–82, July–Aug. 2003.

[45] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[46] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," 2001.

[47] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 1, pp. 41–53, 2004.

[48] M. Knoll, A. Wacker, G. Schiele, and T. Weis, "Bootstrapping in peer-to-peer systems," in *Parallel and Distributed Systems, 2008. ICPADS'08. 14th IEEE International Conference on*, pp. 271–278.

[49] G. W. Cox, W. L. Johnson, J. Strassner, and D. Raymer, "Bootstrapping device state in managed systems," pp. 767–770, 7–11 April 2008.

[50] M. Knoll, A. Wacker, G. Schiele, and T. Weis, "Decentralized bootstrapping in pervasive applications," in *Proc. Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops PerCom Workshops '07*, pp. 589–592, 19–23 March 2007.

[51] S. Milner, J. Llorca, A. Anibha, and U. Vishkin, "A bootstrapping model for directional wireless networks," vol. 10, pp. 840–842, dec 2006.

[52] Y. Jiang, C.-H. Lung, and N. Goel, *A Tree-Based Multiple-Hop Clustering Protocol for Wireless Sensor Networks*, vol. 49 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 371–383. Springer Berlin Heidelberg, 2010.

[53] A. Abbasi and et al., "Restoring connectivity in wireless sensor-actor networks with minimal topology changes," pp. 1–5, 23-27 May 2010.

[54] J. Wang, M. Li, and Y. Liu, "Fractured voronoi segments: Topology discovery for wireless sensor networks," pp. 137–145, 8-12 Nov. 2010.

[55] F. Yasir, "Maximal weight topology discovery in ad hoc wireless sensor networks," vol. 0, pp. 715–722, 2010.

[56] C. Peiwei and et al., "The implementation of a new topology discovery algorithm in mobile ad hoc network," pp. 1–4, 29-31 Oct. 2010.

[57] C. Yonghui, Z. Chunfeng, and L. Zhiqin, "Energy efficient routing protocol for ad hoc networks," vol. 5, pp. V5–320–V5–323, 25-27 June 2010.

[58] S. Hariharan and et al., "Secure neighbor discovery through overhearing in static multihop wireless networks," pp. 1–6, 21-21 June 2010.

[59] L. Raju and et al., "Beacon assisted discovery protocol (bead) for self-organizing hierarchical ad-hoc networks," vol. 3, pp. 1676–1680 Vol.3, 29 Nov.-3 Dec. 2004.

[60] K. M. Konwar and et al., "Node discovery in networks," *Journal of Parallel and Distributed Computing*, vol. 69, no. 4, pp. 337 – 348, 2009.

[61] M. Jelasity and et al., "T-man: Gossip-based fast overlay topology construction," *Computer Networks*, vol. 53, no. 13, pp. 2321 – 2339, 2009.

[62] Y. Liu and et al., "A practical hybrid mechanism for peer discovery," in *Proc. International Symposium on Intelligent Signal Processing and Communication Systems ISPACS 2007*, pp. 706–709, Nov. 28 2007–Dec. 1 2007.

[63] S. A. Borbash and et al., "An asynchronous neighbor discovery algorithm for wireless sensor networks," vol. 5, pp. 998 – 1016, 2007.

[64] P. Dutta and et al., "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," 2008.

[65] P. He, H. Pan, X. Li, and Q. Zheng, "Physical topology discovery based on spanning tree protocol," vol. 14, pp. V14–308–V14–311, 22-24 Oct. 2010.

[66] J. L. Guangyu Dong, "The spanning tree protocol," tech. rep., MNG Group, 2005.

[67] S. Vasudevan, "Self-organization in largescale wireless networks," Master's thesis, University of Massachusetts, Dept Computer Science, sep 2006.

[68] B. T. Nassu, T. Nanya, and E. P. Duarte, "Topology discovery in dynamic and decentralized networks with mobile agents and swarm intelligence," in *Proc. Seventh International Conference on Intelligent Systems Design and Applications ISDA 2007*, pp. 685–690, 20–24 Oct. 2007.

[69] J. Moy, "Rfc2328: Ospf version 2," *RFC Editor United States*, 1998.

[70] A. Siddiqi and B. Nandy, "Improving network convergence time and network stability of an ospf-routed ip network," *NETWORKING 2005*, pp. 469–485, 2005.

[71] inc Cisco Systems, "Cisco discovery protocol (cdp)." `http://www.cisco.com/en/US/docs/ios/12_1/configfun/configuration/guide/fcd301c.html`. [Acedido em 05.03.2011].

[72] inc Overlook, "Fing discovery protocol." http://www.overlook.com/site/index.php/documentation/fing-manual. [Acedido em 13.02.2011].

[73] C. Chi, D. Huang, D. Lee, and X. Sun, "Lazy flooding: A new technique for information dissemination in distributed network systems," vol. 15, pp. 80–92, Feb. 2007.

[74] V. Drabkin, R. Friedman, G. Kliot, and M. Segal, "Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks," in *Proc. 26th IEEE International Symposium on Reliable Distributed Systems SRDS 2007*, pp. 13–22, 10–12 Oct. 2007.

[75] H. Sabbineni and K. Chakrabarty, "Location-aided flooding: an energy-efficient data dissemination protocol for wireless-sensor networks," vol. 54, pp. 36–46, Jan 2005.

[76] Z. Genc and O. Ozkasap, "Eramobile: Epidemic-based reliable and adaptive multicast for manets," *Wireless Communications and Networking Conference, 2007.WCNC 2007.IEEE*, pp. 4395–4400, March 2007.

[77] C. Chen, J. Ma, and J. Salomaa, "Simulation study of cluster based data dissemination for wireless sensor networks with mobile sinks," in *Proc. 10th International Conference on Advanced Communication Technology ICACT 2008*, vol. 1, pp. 231–236, 17–20 Feb. 2008.

[78] N. Damianou, N. Dulay, E. Lupu, M. Sloman, and T. Tonouchi, "Tools for domain-based policy management of distributed systems," pp. 203–217, 2002.

[79] Z. Zhou, "Distribution policy based information dissemination management system," vol. 4, pp. 325–328, 26-27 Dec. 2009.

[80] M. L. Sbodio and W. Thronicke, "Ontology-based context management components for service oriented architectures on wearable devices," pp. 129–133, 10-12 Aug. 2005.

[81] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 32–40, ACM, 2007.

[82] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 299–308, ACM, 2009.

[83] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," *IEEE Transactions on Mobile Computing*, 2010.

[84] P. Hui, E. Yoneki, S. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, p. 7, ACM, 2007.

[85] M. Gerla and J. Tzu-Chieh Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[86] E. J. Stefano Basagni, "Distributed and mobility-adaptive clustering for ad hoc networks," 1998.

[87] M. Elhdhili, L. Azzouz, and F. Kamoun, "Lowest weight: Reactive clustering algorithm for adhoc networks," in *Personal Wireless Communications*, pp. 135–146, Springer, 2006.

[88] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[89] M. Brust, A. Andronache, and S. Rothkugel, "Waca: A hierarchical weighted clustering algorithm optimized for mobile hybrid networks," in *Wireless and Mobile Communications, 2007. ICWMC'07. Third International Conference on*, pp. 23–23, IEEE.

[90] E. H. Kim and J. K. Kim, "A leader election algorithm in a distributed computing system," 1995.

[91] H. Yih and P. K. McKinley, "Group leader election under link-state routing," pp. 95–104, 28-31 Oct 1997.

[92] K. Nakano and S. Olariu, "Uniform leader election protocols for radio networks," pp. 240–247, 3-7 Sept. 2001.

[93] I. C. Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Strandard 802.11-1997, New York, NY, 1997,.

[94] `http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx`. [Acedido em 03.10.2010].

[95] `http://posit.hfoss.org/research`. [Acedido em 20.04.2011].

[96] `http://www.juliobattisti.com.br/tutoriais/paulocfarias/redeswireless020.asp`. [Acedido em 15.11.2010].

[97] P. Brenner, "A technical tutorial on the ieee 802.11 protocol." `http://www.sss-mag.com/pdf/802_11tut.pdf`, 1997.

[98] P. Wireless, *CWAP Certified Wireless Analysis Professional Official Study Guide (Exam PW0-205)*. Planet3 Wireless Series, McGraw-Hill/Osborne, 2004.

[99] `http://www.zytrax.com/tech/wireless/802_mac.htm`. [Acedido em 11.12.2010].

[100] `http://www.wi-fiplanet.com/tutorials/article.php/1447501/Understanding-80211-Frame-Types.htm`. [Acedido em 09.10.2010].

[101] M. S. Gast, *802.11 wireless networks - the definitive guide: creating and administering wireless networks*. O'Reilly, 2002.

[102] `http://www.wifi-doc.com/O.Reilly-802.11.Wireless.Netwo/0596100523/wireless802dot112-chp-3-sect-5.html`. [Acedido em 10.10.2010].

[103] `http://www.slidefinder.net/m/mobile_communications_chapter_wireless_lans/14368166`. [Acedido em 29.09.2010].

[104] `http://www.wireless-net.org/McGraw.Hill-CWAP.Certified.Wir/8156final/LiB0062.html`. [Acedido em 11.10.2010].

[105] S. Institute, "Ieee 802.11 pocket reference guide." `http://www.willhackforsushi.com/papers/80211_Pocket_Reference_Guide.pdf`.

[106] M. A. A. Martin May, Christophe Diot, "Peoplerank: Social opportunistic forwarding," *EEE INFOCOM 2010*, 2010.

[107] E. Y. P. Jon Crowcroft, "Bubble rap: Social-based forwarding in delay tolerant networks," *MobiHoc 08*, 2008.

[108] K. Ramachandran, S. Rangarajan, and J. Lin, "Make-before-break mac layer handoff in 802.11 wireless networks," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 10, pp. 4818–4823, IEEE, 2006.

[109] NS-3, "Manual ns-3.9." http://www.nsnam.org/docs/release/3.9/manual.pdf, 2010.

[110] NS-3, "Tutorial ns-3.9." http://www.nsnam.org/docs/release/3.9/tutorial.pdf, 2010.

[111] M. Lacage and T. Henderson, "Yet another network simulator," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, pp. 12–es, ACM, 2006.

[112] A. Kamerman and L. Monteban, "Wavelan®-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs technical journal*, vol. 2, no. 3, pp. 118–133, 1997.

[113] M. Lacage, M. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 126–134, ACM, 2004.

[114] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 236–251, ACM, 2001.

[115] http://www.thinkwiki.org/wiki/Madwifi. [Acedido em 02.11.2010].

[116] J. Kim, S. Kim, S. Choi, and D. Qiao, "Cara: Collision-aware rate adaptation for ieee 802.11 wlans," in *IEEE INFOCOM*, pp. 1–11, Citeseer, 2006.

[117] F. Maguolo, M. Lacage, and T. Turletti, "Efficient collision detection for auto rate fallback algorithm," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pp. 25–30, IEEE.

[118] https://www.nsnam.org/bugzilla/show_bug.cgi?id=1060. [Acedido em 22.02.2011].

[119] H. P. Pfeifer, "On the validation of radio propagation models - analytical validation of network simulator used propagation and bit error rates models." http://www.jauu.net/data/pdf/propagation-models.pdf.

[120] http://www.eetimes.com/design/signal-processing-dsp/4017668/Modulation-roundup-error-rates-noise-and-capacity?pageNumber=1. [Acedido em 17.01.2011].