Universidade do Minho
Escola de Engenharia

Hélder Tavares de Lemos

Descoberta Passiva de Estações em Redes 802.11

Hélder Tavares de Lemos Descoberta Passiva de Estações em Redes 802.11

UMinho | 2011

outubro de 2011

Universidade do Minho
Escola de Engenharia

Hélder Tavares de Lemos

Descoberta Passiva de Estações em Redes 802.11

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Adriano J. C. Moreira

outubro de 2011

Agradecimentos

Este trabalho não seria possível sem o apoio de algumas pessoas às quais quero prestar os meus agradecimentos.

Começo por agradecer à minha família e principalmente aos meus pais, Casimiro Machado de Lemos e Maria de Lurdes Peixoto Tavares pelo suporte que me deram durante o meu longo percurso académico e por tudo o resto.

Agradeço ao Professor Doutor Adriano J. C. Moreira por toda a dedicação, paciência e disponibilidade com que efectuou a orientação deste trabalho, por todos os ensinamentos transmitidos e pelas valiosas indicações prestadas durante a realização deste trabalho.

Aos meus colegas Ilídio Silva e Ângelo Conde pelo trabalho de equipa realizado sobre o Epi 2.0 e ao colega Carlos Sousa pela disponibilidade que demonstrou para tirar dúvidas sobre o Epi 1.0.

Aos meus colegas de curso João Quintas, André Gomes, Tiago Gomes, Jorge Miranda e Diogo Ribeiro por todos os momentos.

Agradeço também a todos aqueles que colaboraram no projecto Epi 2.0 ao mostrarem disponibilidade para instalar a aplicação, bem como todos os comentários e sugestões que fizeram sobre ela.

Resumo

Hoje em dia, em qualquer zona urbana em que nos encontremos, podemos encontrar dispositivos *WiFi*, sejam estes pontos de acesso ou estações. Estes dispositivos podem ser utilizados para caracterizar o ambiente em que se encontram através da recolha de dados sobre os sinais de rádio que emitem.

Uma das funções básicas do protocolo de acesso ao meio em redes *WiFi* é a descoberta de redes na vizinhança da interface de rede através de uma procura activa ou passiva. Esta função permite detectar apenas os pontos de acesso sendo que, na especificação do IEEE 802.11, não existe nenhuma função que permita detectar outras estações que estejam presentes na vizinhança da *interface* de rede, excepto durante a formação de uma rede *ad-hoc*.

Com este projecto foi desenhada uma solução para redes *WiFi* que permite a uma estação descobrir outras estações na sua vizinhança, obtendo-se uma visão mais completa da vizinhança de cada estação. Esta funcionalidade está disponível em estações que se encontrem em modo infra-estrutura ou *ad-hoc*, uma vez que é o modo em que se encontra a maioria dos dispositivos.

O projecto desenvolvido no âmbito desta dissertação está inserido num outro projecto maior, que aborda a caracterização da dinâmica dos espaços habitados pelos utilizadores de redes *WiFi* e Bluetooth. A solução encontrada foi integrada numa aplicação já existente, destinada à recolha de dados, como um módulo adicional de *software*, permitindo adicionar à recolha periódica dos APs (Access Points) e dispositivos Bluetooth vizinhos a recolha de estações *WiFi* presentes na vizinhança da estação colectora.

Neste documento é discutido todo o processo de exploração da solução para o problema de detecção de outras estações e descrito o *software* de testes desenvolvido. É discutido também o desenvolvimento do módulo de *software* para a aplicação atrás referida, os testes efectuados e os resultados obtidos.

Abstract

Nowadays, wherever we go inside an urban area, we can find a large number of WiFi devices that can be access points or stations. Those devices can be used to characterize the environment where they can be found by collecting the radio signal information they transmit.

One basic function available in WiFi networks MAC (Media Access Control) protocol is the discovery of existing networks in the neighborhood area through a passive or active scanning. This function only allow us to detect access points and the standard IEEE 802.11 doesn't have a function that allows the discovery of stations in the neighborhood area, except during the formation of a ad-hoc network.

This project aim was to design a solution for WiFi networks that allow any station to discover other stations in the neighborhood, allowing to get a more complete vision of the neighborhood of each station. This new feature is available for stations connected to a network in infrastructure and ad-hoc modes as they are the most commonly used modes by most devices.

The project developed under this thesis is embedded in another larger project, that approaches the dynamic characterization of areas that are inhabited by WiFi and Bluetooth networks users. The solution found was embedded as a additional software module into an existing application, with data collecting purpose, allowing in addition to the existing periodic collecting of APs and Bluetooth devices in the neighborhood, the capability to collect WiFi stations in the neighborhood of the station that does the discovery.

In this document the exploration process for finding a solution to the problem of discovering other stations is discussed, and all the testing software developed is described. It is also discussed the development of the software module to embed in the previous existing application, the tests that have been made and the obtained results.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ACK Acknowledgment. 18, 20, 22, 23, 26, 34–41, 136

ACM Auto Configuration Module. 62

AID Association ID. 30, 40, 45, 46

AP Access Point. v, 2, 10–17, 19, 25, 26, 29–31, 34–37, 40, 41, 43–48, 50, 51, 54, 57, 60–63, 75–79, 122, 123, 125–129, 131, 136–138, 140, 141, 143–147, 155–158, 160, 161, 183, 184, 186–188

API Application Programming Interface. 55, 60–62, 71, 129, 131, 134, 170, 174, 176, 177, 180, 182, 190–192, 211, 212

ARP Address Resolution Protocol. 67, 89, 90, 195

ATIM Announcement Traffic Indication Map. 45

BA Block Acknowledgement. 41

BLOB Basic Large Objects. 44

BSA Basic Service Area. 11–13, 16, 45, 46

BSS Basic Service Set. 11–14, 16, 29, 30, 35, 36, 40, 43–45, 61, 136

BSSID Basic Service Set ID. 31, 35–37, 40, 43, 44, 77, 123–125, 132, 133, 136–138, 140–144, 173, 177, 183–189

CCA Clear Channel Assessment. 52

CIFS Common Internet File System. 110, 114

COM Component Object Model. 134

CRC Cyclic Redundancy Code. 33

CSMA Carrier Sense Multiple Access. 17, 18

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 17, 19

CSMA/CD Carrier Sense Multiple Access with Collision Detection. 8, 17

CTS Clear to Send. 18–24, 37–40, 136

DA Destination Address. 31, 135, 138, 140, 143, 144, 187, 188

DCF Distributed Coordination Function. 19, 21, 23, 24

DHCP Dynamic Host Configuration Protocol. 88, 89, 91

DHCPv6 Dynamic Host Configuration Protocol for IPv6. 85, 88

DIFS DCF Inter-Frame Space. 20–22, 24

DiS Distribution System. 11, 14–17, 28, 31, 35, 36, 46–48, 61, 122, 138, 140, 143, 145, 184, 187

DMT Discrete Multi-Tone. 53

DNS Domain Name System. 64–67, 69, 91–94, 105, 106

DSAP Destination Service Access Point. 24, 120

DSL Digital Subscriber Line. 53

DSSS Direct Sequence Spread Spectrum. 8, 10, 45, 53

EIFS Extended Inter-Frame Space. 21, 22

ESA Extended Service Area. 14, 16, 17, 45, 46, 50

ESS Extended Service Set. 13–17, 61, 63

EUI Extended Unique Identifier. 123

FCS Frame Check Sequence. 25, 33

FHSS Frequency Hopping Spread Spectrum. 8, 10, 45, 51, 53

GPS Global Positioning System. 54, 150, 151

GUID Globally Unique Identifier. 134

HR/DSSS High Rate Direct Sequence Spread Spectrum. 8

HT High Throughput. 41

HTML HyperText Markup Language. 72, 132, 170, 174, 181

HTTP Hypertext Transfer Protocol. 64–66, 69, 95, 97, 161, 166

IANA Internet Assigned Numbers Authority. 65, 103

IAPP Inter-Access Point Protocol. 15, 17

IBSS Independent Basic Service Set. 12, 13, 21, 31, 35, 37, 44–47, 51, 61, 122–125, 133, 136, 138, 140, 141, 143, 144, 184, 187, 188, 190

ICMP Internet Control Message Protocol. 78, 79, 86, 195

ICMPv6 Internet Control Message Protocol Version 6. 79–86

IEEE Institute of Electrical and Electronics Engineers. 2, 7, 8, 16, 17, 26, 31, 41, 42, 59, 120, 123, 124, 211

IGMP Internet Group Management Protocol. 86, 88, 195

Infrastructure BSS Infrastructure Basic Service Set. 12–14, 16, 19, 21, 28, 29, 31, 33, 35–37, 45–47, 50, 51, 61, 122, 136–138, 140–144, 187, 188, 190

IOCTL Input/Output Control. 129

ISM Industrial, Scientific and Medical. 9, 18, 52

LAN Local Area Network. 7, 8, 14–16

LLC Logic Link Control. 8, 24, 120, 195

LLMNR Link-Local Multicast Name Resolution. 67, 94

MAC Media Access Control. vii, 2, 3, 5, 8–11, 14–17, 19, 24–28, 30, 31, 36, 37, 39, 41, 43, 44, 49, 51, 52, 54, 55, 57, 59, 63, 64, 69, 70, 72–78, 80, 85, 88, 89, 91, 94, 102, 104, 108, 110, 118, 119, 121–127, 129–134, 136–138, 140–147, 159, 161–163, 165, 170, 173, 176, 180, 181, 184, 188–190, 195, 208, 211, 212

MDNS Multicast Domain Name System. 67, 91, 94, 95, 105, 106

MIB Management Information Base. 49

MIMO Multiple-Input and Multiple-Output. 10

MIT Massachusetts Institute of Technology. 56, 155

MLD Multicast Listener Discovery. 86

MLME MAC Sublayer Management Entity. 49

MTU Maximum Transmission Unit. 85, 87

NAV Network Allocation Vector. 20–23, 25, 30, 33, 34, 41

NBDD NetBIOS Datagram Distribution. 104, 105, 109, 110

NBDS NetBIOS Datagram Service. 65, 69

NBF NetBIOS Frames protocol. 104

NBNS NetBIOS Naming Service. 104–106, 108

NBSS NetBIOS Session Service. 104

NBT NetBIOS over TCP/IP. 104

NDIS Network Driver Interface Specification. 60, 62, 129, 131, 147

NDP Neighbor Discovery Protocol. 85

NetBIOS Network Basic Input/Output System. 104–110, 115, 116, 118

OFDM Orthogonal Frequency Division Multiplexing. 8, 10, 53

OSI Open Systems Interconnection. 7, 8

OUI Organizationally Unique Identifier. 25, 124

PCF Point Coordination Function. 19, 20, 41, 45

PID Process identifier. 111

PIFS PCF Inter-Frame Space. 20

PLCP Physical Layer Convergence Procedure. 8, 9, 51, 52

PLME Physical Sublayer Management Entity. 49

PMD Physical Medium Dependent. 9, 51, 52

QoS Quality of Service. 1, 30, 32, 33, 36, 37, 42, 47

RA Receiver Address. 31, 38–40, 42, 136, 137, 141, 142, 177, 183–187

RR Resource Record. 91–95, 105, 106, 108

RSSI Received Signal Strength Indicator. 162

RTS Request to Send. 18–21, 23, 24, 34, 37–39, 136

SA Source Address. 31, 136, 138, 140, 143, 144, 183, 184, 187, 188

SFX SelF eXtracting. 193, 197, 198

SIFS Short Inter-Frame Space. 20–22, 24, 25, 34, 38–41

SMB Server Message Block Protocol. 65, 110–113, 118

SME System Management Entity. 49

SNAP Sub-Network Access Protocol. 24, 120, 195

SNMP Simple Network Management Protocol. 145

SO Sistema Operativo. 54–56, 59–61, 63, 79, 94, 98, 110, 115, 117, 119, 120, 129, 131, 134, 147, 169, 170, 174–177, 179, 192, 194, 196–198

SOAP Simple Object Access Protocol. 97, 98

SP Service Pack. 59, 61

SSAP Source Service Access Point. 25, 120

SSDP Simple Service Discovery Protocol. 66, 95

SSID Service Set Identity. 45, 132, 133, 173

SUM Sensing and Understanding human Motion dynamics. 156

TA Transmitter Address. 31, 39, 40, 42, 136, 137, 141, 142, 177, 185, 186

TBTT Target Beacon Transmission Time. 51

TCP Transmission Control Protocol. 62–64, 68, 74, 78, 86, 94, 104, 166, 195

TID Traffic Identifier. 32

TIM Traffic Indication Map. 26, 45

TSF Timer Synchronization Function. 51

TTL Time to Live. 78

TU Time Units. 44

UDP User Datagram Protocol. 62, 64, 68, 74, 78, 86, 88, 94, 95, 97, 98, 103–105, 165, 195

URI Uniform Resource Identifier. 95–97

USN Unique Service Name. 95, 97

UUID Universally Unique Identifier. 134

WDS Wireless Distribution System. 36

WEP Wired Equivalent Privacy. 10

WiFi Wireless Fidelity. 1

WMI Windows Management Instrumentation. 134, 179

WPA Wi-Fi Protected Access. 10

WS-Discovery Web Services Dynamic Discovery. 66, 97

XML Extensible Markup Language. 61, 98, 102

Zeroconf Zero configuration networking. 91

Capítulo 1

Introdução

1.1 Enquadramento

As redes WiFi (Wireless Fidelity) nasceram em 1997 com o lançamento da primeira norma 802.11, sendo esta rapidamente substituída pela norma 802.11b em 1999. Os primeiros dispositivos apareceram no mercado em 2000.

Assistiu-se a seguir a uma massificação destes dispositivos e a uma evolução dos mesmos, acompanhando também a norma ao longo dos anos até à actualidade, aumentando a largura de banda dos acessos, mecanismos de qualidade de serviço (QoS, Quality of Service), tipos de redes suportados e mecanismos de segurança, mantendo sempre a retro compatibilidade com os dispositivos mais antigos.

Ao longo dessa evolução podemos destacar algumas fases da mesma, primeiro nos computadores portáteis, em que os fabricantes começaram a incluir placas *WiFi* internas. Mais tarde, com a evolução da banda larga, os operadores de Internet fixa começaram a incluir nas suas ofertas equipamentos com suporte a esta tecnologia, o que fez com que praticamente todas as casas que possuem Internet fixa possuíssem igualmente uma rede *WiFi* activa, ainda que muitas das vezes estas não sejam sequer usadas.

Na actualidade vemos uma massificação de dispositivos móveis com suporte a esta tecnologia, entre os quais telemóveis, *smartphones*, *media players* portáteis, etc.

Hoje em dia, em qualquer zona urbana em que nos encontremos podemos encontrar dispositivos *WiFi*, sendo estes pontos de acesso ou clientes da rede (estações). Podemos usar esses dispositivos que detectamos para explorar o ambiente em que estes se encontram fazendo uma recolha da informação disponibilizada pelos mesmos.

Uma nova área surgiu devido a popularidade destes dispositivos (*Collaborative Sensing*) que explora a informação recolhida permitindo analisar, avaliar e caracterizar os ambientes urbanos em que estes dispositivos se encontram, principalmente em zonas que exibem uma grande concentração dos mesmos.

1.2 Objectivos

Uma das funções básicas do protocolo de acesso ao meio (MAC, Media Access Control) em redes *WiFi* é a função de associação, que consiste na descoberta de redes na vizinhança da *interface* de rede, através de procura activa ou passiva. Após descobertas as redes uma delas é seleccionada, manual ou automaticamente, e completa-se o processo de associação.

A função de descoberta passiva de redes vizinhas permite detectar pontos de acesso (APs, Access Points) através da recepção de *beacons* que são transmitidos periodicamente pelos APs, no entanto, na especificação IEEE (Institute of Electrical and Electronics Engineers) 802.11, não existe nenhuma função que permita detectar outras estações que estejam na vizinhança da *interface* de rede.

Pelo contrário, na tecnologia Bluetooth existe um processo de descoberta que permite detectar quais os dispositivos que se encontram na vizinhança.

Com este projecto pretende-se desenhar uma solução para redes *WiFi* que permita a uma qualquer estação descobrir outras estações na sua vizinhança, à semelhança do que acontece na tecnologia Bluetooth, permitindo obter uma visão mais completa da vizinhança de cada estação, que inclui os APs próximos e também as estações próximas, constituindo uma espécie de radar. Esta funcionalidade deverá estar disponível em estações que se encontrem em modo de infra-estrutura, uma vez que é este o modo em que se encontra a maioria dos dispositivos.

Esta solução será depois integrada numa aplicação que irá efectuar a recolha periódica das estações e APs vizinhos, destina-se à caracterização da dinâmica dos espaços habitados pelos utilizadores de redes *WiFi* e à caracterização do movimento humano.

1.3 Estrutura da Dissertação

No capítulo 1 é feita uma breve introdução ao tema desta dissertação. É descrita a motivação para a realização deste trabalho juntamente com os respectivos objectivos a alcançar. Este capítulo termina com a descrição da estrutura adoptada para esta dissertação.

O capítulo 2 apresenta o estado de arte relativamente ao protocolo de comunicações em redes sem fios 802.11, incidindo principalmente na camada MAC e tramas MAC. Este capítulo apresenta também algumas ferramentas de monitorização de redes *WiFi* e descreve alguns trabalhos relacionados com este.

No capítulo 3 é descrito o processo de exploração da solução com a abordagem seguida, com a primeira tentativa de solução de captura e a análise de tráfego capturado na camada Ethernet. Segue-se a descrição da solução que foi adoptada que passa pela captura de tráfego na camada MAC 802.11 usando o Network Monitor e o processamento dos endereços nos cabeçalhos 802.11.

O capítulo 4 começa com uma descrição do conceito de *Collaborative Sensing* e a arquitectura das *Collaborative Sensing Networks* bem como algum trabalho relacionado com este tipo de redes. Depois é apresentado a aplicação Epi e a descrição das novas funcionalidades introduzidas ao nível da recolha de novas assinaturas de rádio, novo *interface* gráfico de utilizador e alterações a nível de servidor e protocolos de comunicações.

No capítulo 5 é descrita a implementação do módulo desenvolvido (o WifiRadar) e a sua integração na aplicação Epi. O capítulo termina com a integração do Network Monitor no pacote de instalação da aplicação Epi.

No capítulo 6 são apresentados os testes que foram realizados com a aplicação Epi e o WifiRadar integrado, bem como a análise dos resultados obtidos durante o teste realizado durante um período de 3 semanas.

Por fim, o capítulo 7 apresenta as conclusões desta dissertação, resumindo os objectivos cumpridos e discutindo o trabalho futuro sobre o WifiRadar e a aplicação Epi.

Capítulo 2

Norma 802.11 e Trabalho Relacionado

O módulo a ser desenvolvido envolve um conjunto prévio de conhecimentos. Neste capítulo é feito uma introdução às redes sem fios 802.11 e à sua arquitectura. É também feito um estudo sobre a norma do 802.11, que irá incidir principalmente sobre a sua camada MAC e o formato e uso das tramas MAC.

Também foi efectuada uma pesquisa sobre ferramentas que tenham sido desenvolvidas para monitorização de redes *WiFi*.

O capítulo termina com a descrição de alguns trabalhos relacionados com este.

2.1 Redes sem fios

As redes sem fios surgiram devido a um conjunto de necessidades que as redes fixas não conseguiam satisfazer: "People move. Networks don't."[1]. Estas duas simples frases explicam essas necessidades e a razão da explosão no surgimento de *hardware* com suporte para redes sem fios nas duas últimas décadas. Devido a essa explosão os preços caíram e, nos dias de hoje, as redes sem fios fazem parte de uma diversidade de dispositivos electrónicos com o qual vivemos diariamente.

As redes sem fios oferecem várias vantagens relativamente às redes fixas:

Mobilidade

Os utilizadores movem-se mas os dados normalmente são armazenados de forma centralizada. As redes sem fios permitem aos utilizadores acesso aos dados

enquanto estes se encontram em movimento, o que leva a grandes ganhos na produtividade.

Facilidade e velocidade de implantação

Existem locais com estruturas (ex: estruturas antigas e históricas) em que é difícil senão mesmo impossível a colocação de cabos de rede. Passar cabos por paredes de pedra de edifícios antigos e que normalmente não têm plantas, ou já foram perdidas, é um desafio. Em outros locais as leis de preservação histórica tornam difícil passar cabos para a criação de uma nova rede. Mesmo em edifícios modernos a instalação de cabos pode ter um custo elevado.

Flexibilidade

Uma rede sem cabos não precisa de actualização dos cabos. As redes sem fios permitem a formação rápida de espaços de trabalho temporários, uma rede para um pequeno grupo numa reunião, ou o movimento entre os compartimentos de trabalho e os escritórios. A expansão da rede é muito fácil uma vez que o meio usado é o ar, bastando para isso ficar apenas no raio de cobertura da rede. Não existem cabos para puxar, ligar ou tropeçar por cima. A flexibilidade é o ponto mais importante para o mercado dos "hot spots", disponíveis em hotéis, aeroportos, estações de comboio, bibliotecas e cafés.

Custo

Em muitos casos o custo pode ser reduzido usando as tecnologias de redes sem fios. Como exemplo, a interligação de dois edifícios, usando equipamento 802.11 para criar uma ponte sem fios. O custo inicial será maior devido ao equipamento que será necessário comprar, pontos de acesso e placas de rede sem fios. No entanto, depois do investimento inicial, estas redes sem fio com linha de vista apenas têm um custo de operação mensal muito baixo e com o passar do tempo o custo é menor do que alugar um cabo a uma companhia telefónica.

Apesar do número de vantagens que as redes sem fios oferecem estas não substituem as redes fixas. As redes sem fios também têm desvantagens:

- Menor largura de banda em comparação a redes por cabo devido ao limite do espectro de frequência livre disponível.
- O débito é partilhado entre todas as estações que partilham o mesmo meio.

- Maiores risco de segurança devido ao meio estar disponível para toda a gente na area de cobertura do transmissor.
- O meio sem fios usado é incerto e não fiável, o que leva à necessidade da validação dos dados recebidos devido a perdas.

No entanto essas desvantagens tem vindo a ser cada vez mais minimizadas com o aumento da segurança através do uso de técnicas e algoritmos criptográficos mais elaborados. O aumento do débito binário das redes sem fios também tem aumentado com o surgimento de novas normas que vão sendo adoptadas pelos fabricantes no lançamento de novos equipamentos, mantendo a compatibilidade com os equipamentos antigos.

Inicialmente as ofertas de redes sem fios estavam limitadas ao fabricante, ou seja, todos os equipamentos que iam fazer parte da rede sem fios tinham que ser do mesmo fabricante. Cada fabricante tinha uma especificação própria dos protocolos e *hardware* usados e isto levava a soluções com um custo muito elevado e pouco flexíveis. Foi então que surgiu a normalização de uma solução que fosse adaptada pelos fabricantes, de forma a tornar os equipamentos de diferentes fabricantes compatíveis e mais baratos.

2.2 Norma IEEE 802.11

Nesta secção apresenta-se a norma IEEE 802.11, dando ênfase aos aspectos relevantes para o trabalho descrito nesta dissertação.

2.2.1 Introdução

As redes sem fios 802.11, também conhecidas como redes *WiFi*, viram a primeira versão da norma aprovada em 1997 pelo comité de normalização do IEEE. Em 1999, depois de uma revisão à norma, é aprovada a nova versão que é adoptada pelos fabricantes e dá origem aos primeiros equipamentos *WiFi*. Até à actualidade, foram surgindo novas versões da norma que permitiram lançar novos equipamentos com maior largura de banda e compatíveis com os mais antigos já existentes.

O 802.11 é um membro da família IEEE 802, que é uma série de especificações para as tecnologias de redes de área local (LAN, Local Area Network). Os diferentes componentes da família 802 estão relacionados entre si e cada um tem o seu lugar no modelo OSI (Open Systems Interconnection), como mostra a Figura 2.1. As

especificações IEEE 802 estão focadas nas camadas mais baixas do modelo OSI porque incorporam componentes da camada física e da camada de ligação. A sub camada MAC é um conjunto de regras que determina como é feito o acesso ao meio e o envio de dados, os detalhes de como a transmissão e recepção é efectuada e feita pela camada física.

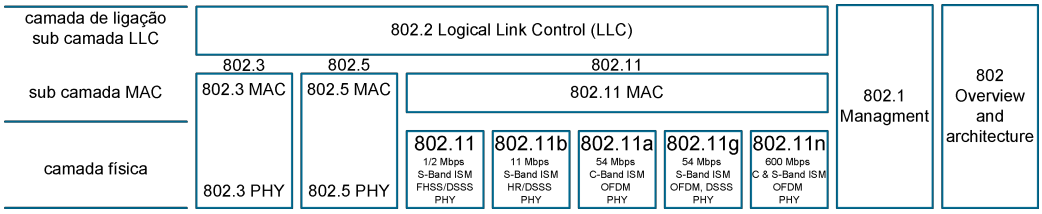


Figura 2.1: Relacionamento dos componentes da família IEEE 802.

As especificações individuais na família 802 são identificadas pelo segundo número. O 802.3, por exemplo, é a especificação CSMA/CD (Carrier Sense Multiple Access with Collision Detection) (normalmente conhecido como Ethernet). Outras especificações descrevem partes da pilha de protocolos 802, o 802.2 especifica a sub camada de ligação comum (LLC, Logic Link Control) que é usado pelas tecnologias de redes LAN das camadas inferiores. As funções de gestão do 802 são especificadas no 802.1.

O 802.11 é apenas mais uma camada de ligação que usa o encapsulamento do 802.2/LLC. A especificação base das redes 802.11 define a sub camada MAC, a gestão MAC dos protocolos e serviços, duas camadas físicas com modulações FHSS (Frequency Hopping Spread Spectrum) e DSSS (Direct Sequence Spread Spectrum). As revisões feitas mais tarde ao 802.11 foram adicionando mais camadas físicas, 802.11b com modulação HR/DSSS (High Rate Direct Sequence Spread Spectrum), 802.11a com modulação OFDM (Orthogonal Frequency Division Multiplexing), 802.11g com modulação OFDM e DSSS e a mais recente 802.11n com modulação OFDM.

O que torna o 802.11 muito importante é o simples detalhe de "ser apenas mais uma camada de ligação do 802.2". O 802.11 permite o acesso móvel às redes 802 que já existiam, como a rede Ethernet. No entanto, para conseguir este objectivo, foi necessário incorporar na camada MAC um conjunto de novas funcionalidades, o que resultou numa camada MAC bastante complexa relativamente às outras especificações MAC do IEEE 802.

A camada física também é relativamente complexa devido ao uso de transmissão rádio. Nas redes 802.11, esta camada divide-se em dois componentes genéricos: o PLCP (Physical Layer Convergence Procedure) que faz o mapeamento das tramas

MAC ao meio, e o PMD (Physical Medium Dependent) que transmite essas tramas. Nas redes 802.11, o PLCP encontra-se no limite superior da camada física colado à parte inferior da sub camada MAC. Este acrescenta um número de campos adicionais à trama recebida da sub camada MAC que são necessários para a transmissão dessa trama sobre o meio.

As redes 802.11 necessitam de um recurso chave para poder transmitir usando o ar como meio de transmissão, que é uma banda de frequências no espectro de rádio. Cada banda não é mais do que uma gama de frequências alocada para uma aplicação específica que tem associado uma largura de banda. A atribuição dessas bandas de frequências requer uma licença que é atribuída e controlada rigorosamente por autoridades reguladoras de forma a evitar a sobreposição das ondas de rádio.

Existem bandas que geralmente são livres para dispositivos com baixa potência de transmissão, não necessitando de uma licença explícita. Existe um conjunto de três dessas bandas (Tabela 2.1) que são conhecidas como bandas ISM (Industrial, Scientific and Medical), e duas dessas bandas (S-Band ISM e C-Band ISM) são usadas pelos dispositivos 802.11. Essas bandas são divididas em canais de forma a permitir que várias redes 802.11 possam coexistir no mesmo espaço.

Banda	**Gama de frequências**
UHF ISM	902 - 928 MHz
S-Band ISM	2.4 - 2.5 GHz
C-Band ISM	5.725 - 5.875 GHz

Tabela 2.1: Bandas ISM.

O uso de ondas de rádio como meio de transmissão do 802.11 trouxe uma série de desafios na especificação, uma vez que estas podem sofrer problemas de propagação que podem interromper a ligação de rádio, como por exemplo, interferência provocada por múltiplos caminhos e sombras.

Outra das preocupações principais na especificação do 802.11 é a segurança. A transmissão de dados sobre o meio está disponível para quem está na area de cobertura do transmissor com a antena apropriada. Isto torna mais fácil o *sniffing*[1], uma vez que as transmissões de rádio foram desenhadas para serem processadas por qualquer receptor que se encontre na área de cobertura.

A norma 802.11 inclui mecanismos de segurança que têm vindo a ser melhorados a

[1]Consiste em interceptar o tráfego de dados numa rede de computadores.

cada nova versão da especificação, como o WEP (Wired Equivalent Privacy) e o WPA (Wi-Fi Protected Access). Assim, o acesso a uma rede 802.11 pode ser condicionado a apenas alguma estação com o conhecimento prévio de uma chave de acesso. Permite também a cifra dos dados numa ligação 802.11 para que alguém que faça *sniffing* dessa ligação à rede não tenha acesso directo aos dados.

As diferenças entre as especificações 802.11 que foram lançadas desde 1997 até hoje são apresentadas na Tabela 2.2.

802.11	**Data lançamento**	**Débito (Mbps)**	**Freq. (GHz)**	**Notas**
-	Jun 1997	1,2	2.4	Primeira norma. Especificava três camadas físicas DSSS, FHSS e infravermelhos.
a	Set 1999	até 54	5	Usa a mesma camada de ligação da especificação original mas com modulação OFDM na camada física a operar nos 5 GHz.
b	Set 1999	5.5, 11	2.4	Usa a camada MAC original e introduziu melhorias na modulação da especificação original. Este tornou-se um dos mais populares porque foi rapidamente adoptado devido ao débito e a uma baixa de preço dos equipamentos.
g	Jun 2003	até 54	2.4	A camada física opera nos 2.4 GHz (como no 802.11b) e usa a modulação OFDM (como no 802.11a).
n	Out 2009	até 600	2.4, 5	Acrescenta o suporte para múltiplas entradas, múltiplas saídas (MIMO, Multiple-Input and Multiple-Output), múltiplas antenas na transmissão e recepção. Opera nas bandas de 2.4 e 5 GHz com modulação OFDM e suporte para canais de 40 MHz.

Tabela 2.2: Comparação de algumas das especificações 802.11.

A especificação original do 802.11 foi rapidamente substituída pela 802.11a e 802.11b, as quais deram origem à primeira vaga de equipamentos. Mais tarde apareceu o 802.11g, que deu origem a uma nova vaga de equipamentos que suportavam banda dupla e mesmo modo triplo suportando a, b e g na mesma placa de rede e AP. As especificações que até 2007 se encontravam em 8 documentos separados, foram agrupadas num único documento (IEEE Std 802.11-2007[2]) com as especificações 802.11a, b, d, e, g, h, i e j. A última especificação 802.11n (IEEE Std 802.11-2009[3]), lançada em 2009, já era usada em muitos dispositivos sendo estes baseados num rascunho de uma proposta lançada em 2007.

2.2.2 Arquitectura

Uma rede 802.11 é constituída por quatro componentes físicos. A presença de alguns destes componentes não é obrigatória para todos os tipos de rede 802.11. Os componentes são os seguintes:

Ponto de acesso (AP) - estes dispositivos fazem a ponte entre o meio sem fios e o fixo, de forma a converter as tramas 802.11 recebidas para outro tipo de tramas, normalmente Ethernet. Também suportam um conjunto de outras funções, como por exemplo interligar um conjunto de estações 802.11, no entanto a função de ponte é de longe a mais importante.

Meio sem fios - para mover as tramas de estação para estação o 802.11 usa o meio rádio. São definidas na norma um conjunto de camadas físicas em que a arquitectura permite o suporte das mesmas por uma única camada MAC.

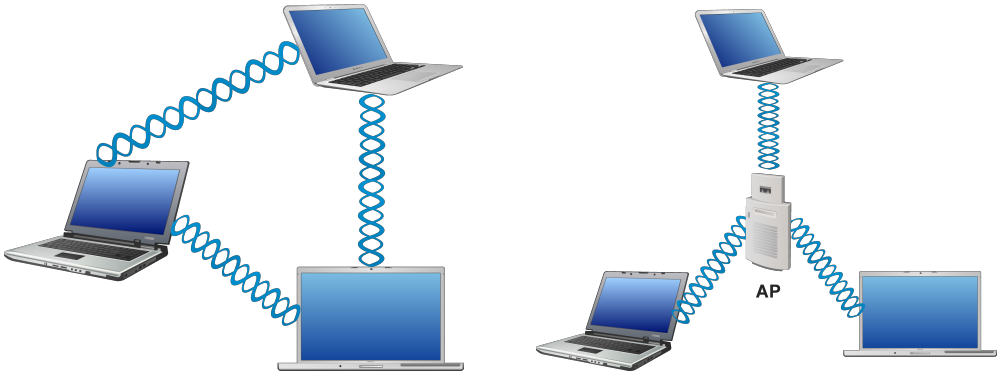
Estações - são os dispositivos com capacidade de computação que ligam à rede através de uma placa de rede sem fios 802.11. Estes dispositivos usam a rede para trocar dados. Normalmente estes dispositivos são alimentados a baterias (computadores portáteis, *smarthphones*, etc). Em alguns casos podem ser computadores de secretária que por algum motivo não podem ser ligados à rede por cabo. Neste projecto o módulo desenvolvido destina-se a detectar estes dispositivos.

Sistema de distribuição (DiS, Distribution System) - quando vários APs se encontram ligados para formar uma área maior de cobertura. Estes comunicam entre si para acompanhar o movimento de estações. Este é o componente lógico do 802.11 que envia as tramas para o destino. Normalmente a ponte de comunicação entre os APs é feita por uma rede Ethernet.

Tipos de Redes

O bloco básico de uma rede 802.11 consiste simplesmente num conjunto de estações que comunicam entre si denominado conjunto de serviços básico (BSS, Basic Service Set). A comunicação entre as estações ocorre dentro de uma área denominada área de serviços básico (BSA, Basic Service Area). Esta área é definida pelas características de propagação do meio sem fios. Quando uma estação se encontra dentro dessa área pode comunicar com outros membros dessa BSS.

Uma BSS pode ser de dois tipos:



(a) IBSS. **(b)** Infrastructure BSS.

Figura 2.2: Tipos de redes 802.11.

Independente (IBSS, Independent Basic Service Set)

As estações comunicam directamente entre si, como mostra a Figura 2.2a. Neste tipo de rede as estações têm que se encontrar na área de cobertura que permita a comunicação directa. A rede mais pequena que é possível formar é constituída por duas estações. Normalmente este tipo de rede é composto por um número pequeno de estações que se juntam para um propósito específico, e durante um curto espaço de tempo. Por exemplo, os participantes criam a rede no início de uma reunião para partilhar dados. Quando a reunião acaba a rede é dissolvida. Devido à sua curta duração, tamanho pequeno e propósito específico este tipo de rede também é conhecido como rede *Ad-hoc*.

Infra-estrutura (Infrastructure BSS, Infrastructure Basic Service Set)

Este tipo de rede distingue-se por usar um AP, como mostra a Figura 2.2b. Toda a comunicação é feita através do AP, incluindo a comunicação entre estações na mesma BSA. Assim, uma estação quando quer comunicar com uma segunda estação na mesma Infrastructure BSS, fá-lo em dois saltos. Primeiro a trama é transferida da estação de origem para o AP. Depois o AP transmite essa trama para a estação de destino. Com as comunicações todas a serem efectuados pelo AP, a BSA é definida pelos pontos de onde o AP pode receber tramas. Necessitar de diversos saltos para a comunicação requer mais capacidade do transmissor e receptor do que a transmissão directa, como nas IBSS. No entanto, têm duas grandes vantagens:

- Uma Infrastructure BSS é definida pela distância máxima até ao AP, todos as estações têm que estar dentro dessa distância para ter a cobertura do AP. No entanto, não é imposta distância máxima entre estações, o que pode aumentar a distância de comunicação entre estações. A comunicação directa usada nas IBSS permite uma maior capacidade de transmissão, mas à custa do aumento da complexidade da camada física que é necessária para manter uma relação com as estações vizinhas na mesma BSA.
- Os APs numa Infrastructure BSS têm um posição que permite assistir as estações na tentativa de poupar energia. O AP pode armazenar tramas destinadas às estações enquanto estas se encontram em modo de poupança de energia. Quando uma estação voltar ao modo normal, este entrega as tramas armazenadas à estação. Normalmente as estações sem fios são alimentadas a baterias e é importante desligar o transmissor sem fios para apenas voltar a ligar para transmitir tramas ou receber as tramas armazenadas no AP, de forma a poupar energia.

Numa Infrastructure BSS as estações têm que se associar com o AP para ter acesso aos serviços da rede. A associação é o processo usado para uma estação se juntar a uma rede 802.11, que equivale a ligar o cabo numa rede Ethernet. Neste processo a iniciativa parte sempre das estações e o AP decide se permite ou não o acesso, baseado no conteúdo do pedido de associação. Uma estação só pode estar associada com um único AP de cada vez. A norma 802.11 não impõe um limite para o número de estações que estejam associadas com o mesmo AP, no entanto, o desempenho baixo da rede numa situação com muitas estações associadas irá impor esse limite.

A solução deste projecto foi desenvolvida de forma a detectar as estações presentes na vizinhança nos dois tipos de redes 802.11.

Áreas alargadas de serviço (ESA)

Uma BSS consegue fornecer cobertura de rede para um pequeno espaço mas não para espaços mais alargados, como o campus de uma universidade. O 802.11 permite a criação de redes grandes através da ligação de várias BSSs de forma a formar um conjunto de serviços alargado (ESS, Extended Service Set). Uma ESS é criada através da interligação das várias BSSs usando uma rede de *backbone*. O 802.11 não

específica nenhuma tecnologia específica para a rede de *backbone*mas é um requisito esta suportar um conjunto de serviços específicos.

As estações que fazem parte da mesma ESS podem comunicar entre si mesmo quando se encontram em BSSs diferentes. Inclusive podem mover-se entre BSSs. Na comunicação entre estações na mesma ESS o meio sem fios tem que actuar como uma ligação da camada 2. Os APs funcionam como pontes e a comunicação entre estações na ESS requer uma rede de *backbone* também a actuar como uma ligação da camada 2. Assim bastará uma qualquer ligação de camada 2. Os APs podem ser ligados todos ao mesmo comutador (*switch*) ou usar LANs virtuais para o caso de a ligação de camada 2 se estender a uma área grande.

A mobilidade numa ESS no 802.11 é apenas possível se a rede de *backbone* pertencer toda a um único domínio da camada de ligação.

Uma área alargada de serviço (ESA, Extended Service Area) é o nível mais alto de abstracção suportado pelas redes 802.11. Os APs numa ESS operam de forma a que seja apenas usado um único endereço MAC para que alguém que aceda de fora consiga comunicar com um estação dentro da ESS. Um router não sabe a localização de uma estação na ESS, confiando nos APs para fazer a entrega das tramas às estações.

Sistema de distribuição

O DiS é o mecanismo que permite a mobilidade através da comunicação entre os APs. Quando uma trama é entregue ao DiS, é entregue ao AP correcto e esse entrega a trama à estação de destino.

O DiS é responsável por seguir onde uma estação se encontra fisicamente e entregar as tramas que lhe são destinadas. Quando uma trama é enviada para uma estação, o DiS é encarregue de entregar a mesma ao AP ao qual a estação se encontra associada. Uma parte do mecanismo de distribuição é a rede de *backbone*, no entanto, esta sozinha não têm forma de escolher qual o AP apropriado. No 802.11, a rede de *backbone* é apenas considerada o meio do DiS.

A outra parte dos DiS são os APs, mais concretamente no motor que faz a ponte que existe normalmente entre os *interfaces* sem fios e Ethernet. Este interage directamente com os dois *interfaces* e completa o DiS.

A comunicação entre estações numa Infrastructure BSS têm que usar o DiS, uma vez que não se encontram directamente ligadas entre si. Mesmo as estações dentro da mesma BSS têm que usar o DiS.

Existem características nas redes 802.11 que são parte da interacção nos DiS:

Comunicação entre APs - o DiS tem um método para gestão das associações. Uma estação só pode estar associada num único AP num dado momento, se uma estação está associada a um AP os outros APs na ESS precisam de o saber. Assim, um AP tem que informar os outros APs das estações que têm associadas. Os APs normalmente usam o protocolo IAPP (Inter-Access Point Protocol) sobre o meio do *backbone* para comunicar entre si. Este ainda não é uma norma do 802.11 mas apenas uma recomendação no grupo de trabalho do 802.11f.

Pontes sem fios - normalmente o DiS tem uma rede cablada mas o 802.11 suporta o uso do meio sem fios para o DiS. Este permite ligar duas LANs pela camada de ligação, ligando rapidamente duas localizações fisicamente distintas.

Operações e Serviços de Rede

O 802.11 foi desenhado para ser apenas mais uma das camadas de ligação para protocolos das camadas superiores. A herança que trás do Ethernet é grande e as estações são identificadas por endereços MAC de 48 bits. As tramas também são entregues baseadas no endereço MAC, no entanto a entrega de tramas não é confiável como nas redes Ethernet mas o 802.11 incorpora alguns mecanismos básicos que ajudam de certa forma a ultrapassar isso. Do ponto de vista de utilizador as redes 802.11 são a mesma coisa que as redes Ethernet. No entanto para os administradores de rede são necessárias maiores competências sobre o 802.11.

Uma forma de definir uma tecnologia de rede é através dos serviços que esta fornece. No caso do 802.11 são nove serviços, três usados nas operações de transmissão de dados, os restantes para operações de gestão. Os serviços encontram-se resumidos na Tabela 2.3.

Serviços de estação - os serviços de estação são obrigatórios e fazem parte de todas as estações 802.11. Estes serviços estão presentes nas estações móveis e nas *interfaces* sem fios dos APs. As estações providenciam o serviço de entrega de tramas mas para suportar esta tarefa também são necessários os serviços de autenticação para estabelecer a associação. Estas também podem beneficiar ao usar as funções do serviço de privacidade para protecção das mensagens.

Serviços	Estação ou DiS	Descrição
Distribution	DiS	Este serviço é usado numa Infrastructure BSS para determinar o endereço de destino de uma trama.
Integration	DiS	Entrega as tramas numa rede IEEE 802 LAN, fora da rede sem fios.
Association	DiS	Usado para uma estação se registar com o AP, que vai servir de passagem para as tramas da estação.
Reassociation	DiS	Usado para uma estação se registar num AP diferente dentro da mesma ESS.
Disassociation	DiS	Usado para remover o registo de uma estação do AP.
Authentication	Estação	Usado para autenticar uma estação perante o AP, antes de ocorrer a associação.
Deauthentication	Estação	Usado para terminar a autenticação perante o AP.
Privacy	Estação	Providencia mecanismos de protecção contra uma leitura não autorizada das tramas.
MSDU delivery	Estação	Entrega de dados ao receptor.

Tabela 2.3: Serviços disponíveis numa rede 802.11.

Serviços de DiS - os serviços DiS ligam os APs ao DiS. A função principal do AP é extender os serviços das redes fixas às redes sem fios. Isto é feito providenciando a distribuição e integração de serviços para a parte sem fios. A gestão da associação das estações sem fios é uma das funções principais do DiS. De forma a manter os dados de associação e localização das estações, o DiS providência os serviços de associação, re-associação e dissociação.

Mobilidade

Uma das motivações para o desenvolvimento do 802.11 foi também a mobilidade. As estações podem-se mover enquanto se encontram ligadas na rede sem fios a transmitir tramas. A norma reconhece três cenários de transições na mobilidade das estações:

Sem transição - quando uma estação se encontra dentro da BSA de um AP ao qual se encontra associada sem se mover ou movendo-se mas sem sair dessa BSA. Desta forma não existe nenhuma transição entre APs.

Transição de BSS - quando uma estação se encontra dentro de uma ESA, o 802.11 fornece mobilidade ao nível da camada MAC. As estações fazem a monitorização

continua da potência e qualidade do sinal dos APs que constituem essa ESA e fazem a associação ao AP mais apropriado enquanto se movimentam. O DiS é depois capaz de entregar as tramas destinadas ao endereço MAC de uma estação deixando os APs encarregues do último salto. Os APs têm que cooperar entre si de forma a informar os outros APs quando um estação faz essa transição. O 802.11 não especifica como essa comunicação é feita, no entanto normalmente é usado o protocolo IAPP.

Transição de ESS - o 802.11 não suporta a transcrição de uma ESS para outra ESS distinta sem a perda de ligação para as camadas superiores. De forma a manter essa ligação na transcrição de ESS, no caso do TCP/IP, seria necessário o uso de Mobile IP[1]. No entanto, o 802.11 permite que, com relativa facilidade, uma estação faça a associação com um AP numa nova ESA.

2.2.3 Camada MAC

A camada MAC é a chave da especificação do 802.11. Esta encaixa em todas as camadas físicas, controla a transmissão dos dados do utilizador para o meio, as principais operações com as tramas e a interacção com as redes fixas de *backbone*. As diferentes camadas físicas fornecem apenas diferentes velocidades de transmissão que inter-operam entre si.

A norma 802.11 não muda muito radicalmente relativamente às outras normas IEEE 802. Este adapta o típico funcionamento das redes Ethernet às ligações rádio. As redes 802.11 também usam CSMA (Carrier Sense Multiple Access) para controlar o acesso ao meio de transmissão. No entanto, para evitar o desperdício de capacidade de transmissão em detecção de colisões como o CSMA/CD nas redes Ethernet, este usa CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) de forma a evitar as colisões. O acesso ao meio também é feito de forma distribuída e não centralizada.

No desenvolvimento da camada MAC do 802.11 foi necessário ultrapassar alguns desafios e impor algumas regras para acesso ao meio que não existiam nas redes fixas.

[1]Protocolo que permite aos utilizadores moverem-se entre redes distintas, mantendo um endereço IP permanente.

Mecanismo de reconhecimento positivo

Nas comunicações sem fios não se pode assumir que uma trama transmitida chega ao seu destino correctamente como nas redes fixas Ethernet. O meio sem fios é diferente, especialmente porque usa frequências não licenciadas ISM. As redes 802.11 têm que assumir que existe interferência e foram desenhadas de forma a contornar isso. Foram consideradas interferências de fornos micro-ondas e outras fontes de rádio frequência. Adicionalmente existe o ruído, interferência multi-caminhos e situações em que uma estação se move para um ponto morto (zona de sombra) que impede a transmissão de tramas.

A especificação 802.11 incorpora um mecanismo em que para cada trama transmitida, tem que existir uma resposta de reconhecimento positivo que confirma a recepção da trama através da trama de ACK (Acknowledgment). No caso de alguma das partes falhar a trama é considerada perdida. As tramas de ACK são transmitidas sem contenção. Quando os dados são corrompidos, o destinatário não faz o reconhecimento negativo.

Mecanismo RTS/CTS

Numa rede sem fios existem pontos onde algumas estações se podem encontrar fora do alcance de cobertura rádio de outras estações na mesma rede. Isto pode provocar problemas uma vez que as estações dependem da recepção de transmissões para fazer as funções de *carrier-sensing* do CSMA e detectar transmissões de forma a evitar colisões. Uma colisão que ocorra entre duas estações escondidas que transmitam durante a transmissão da outra são muito difíceis de detectar, porque um transceptor rádio não consegue transmitir e receber ao mesmo tempo.

Para evitar colisões, o 802.11 permite que as estações usem as tramas de RTS (Request to Send) e CTS (Clear to Send) de forma a "limpar" a área antes da transmissão.

Uma estação que pretenda enviar uma trama, inicia o processo enviando uma trama de RTS. A trama RTS serve para reservar o meio para a transmissão da trama e para silenciar as estações que o recebem. A estação de destino que recebe o RTS responde com uma trama CTS. A trama CTS, assim como a trama RTS, serve para silenciar as estações que a recebem. Quando a troca do RTS/CTS está completa, a estação que a iniciou pode transmitir as tramas sem a preocupação de interferências das estações escondidas. As tramas transmitidas usando este processo

também precisam de reconhecimento positivo.

Este processo consome capacidade de transmissão, principalmente no atraso introduzido antes do início da transmissão, justificando-se o seu uso apenas em ambientes com alta capacidade e contenção significativa.

Modos de acesso ao meio

O acesso ao meio sem fios é controlado por funções de coordenação. O modo idêntico ao usado nas redes Ethernet é fornecido pelo DCF (Distributed Coordination Function). Quando é necessário um serviço livre de contenção, este é fornecido pelo PCF (Point Coordination Function) que é construído em cima do DCF e apenas é fornecido em Infrastructure BSS.

DCF - é a base do acesso ao meio CSMA/CA. Este verifica, como nas redes Ethernet se o meio rádio se encontra livre antes de transmitir. Para evitar colisões, as estações esperam um tempo aleatório (*backoff*) antes da transmissão de cada trama. O primeiro a transmitir irá tomar conta do canal. Em algumas circunstâncias o DCF usa o mecanismo de RTS/CTS para reduzir a possibilidade de acontecerem colisões.

PCF - fornece o serviço livre de contenção. Este necessita de uma estação que faça a coordenação de forma a garantir que o meio é usado sem contenção. Os pontos de coordenação estão nos APs, o que restringe o PCF apenas às Infrastructure BSS. De forma a ganhar acesso aos serviços livres de contenção, no PCF existe um período de contenção que permite às estações transmitir tramas após um curto intervalo. O PCF nunca foi extensamente implementado e usado logo não será abordado em detalhe.

Funções de Carrier-Sensing

O *carrier-sensing* é usado para determinar se o meio se encontra livre. O 802.11 usa duas funções de *carrier-sensing*, uma física e outra virtual. Para qualquer uma dessas funções, quando o meio está ocupado, o MAC reporta às camadas superiores.

O *carrier-sensing* físico é fornecido pela camada física em questão e depende do estado do meio e modulação usados, no entanto, não consegue fornecer toda a informação necessária, uma vez que o transceptor não consegue transmitir e receber simultaneamente.

O *carrier-sensing* virtual é fornecido pelo NAV (Network Allocation Vector). As tramas 802.11 contém um campo de duração que é usado para reservar o meio por um período de tempo. O NAV é um temporizador que indica o tempo que o meio está reservado. As estações definem no NAV o tempo que prevêem usar o meio, incluindo as tramas necessárias para completar a operação actual. Outras estações fazem a contagem do NAV até 0. Enquanto o NAV não estiver a zero, o *carrier-sensing* indica que o meio está ocupado.

Ao usar o NAV as estações conseguem garantir a não interrupção de sequências de operações, como a sequência de RTS/CTS.

Espaço Inter-tramas

Assim como nas redes Ethernet, o espaço inter-trama no 802.11 também é importante na coordenação do acesso ao meio de transmissão. Este usa quatro espaços diferentes em que três são usados para determinar o acesso ao meio.

De forma a evitar colisões, no 802.11 a transmissão das tramas é atrasada até o meio se encontrar livre. Ao variar o tempo inter-tramas, pode-se criar uma diferença de prioridade para diferentes tipos de tráfego. Tramas com prioridade elevada não atrasam tanto tempo depois de o meio ficar livre, assim ocupam o meio antes de tramas com prioridade mais baixa. De forma a lidar com os diferentes taxas de transmissão do 802.11, os tempos de atrasos foram fixados, independentemente da velocidade de transmissão.

SIFS (Short Inter-Frame Space) - é o tempo de atraso usado nas transmissões de prioridade mais elevada. As tramas como RTS/CTS e ACK apenas esperam SIFS depois de o meio ficar livre, assim ocupam o meio e ganham prioridade sobre a transmissão das tramas com atraso maior.

PIFS (PCF Inter-Frame Space) - é o atraso usado no PCF apenas durante o período livre de contenção. Um estação atrasa PIFS antes de transmitir uma trama no período livre de contenção.

DIFS (DCF Inter-Frame Space) - é o atraso mínimo em que o meio se encontra livre usado nos serviços com contenção. As estações podem ter acesso ao meio se este se encontra livre por um período maior que DIFS.

EIFS (Extended Inter-Frame Space) - é o atraso usado apenas quando ocorrem erros na transmissão de tramas e não têm um tempo de intervalo fixo.

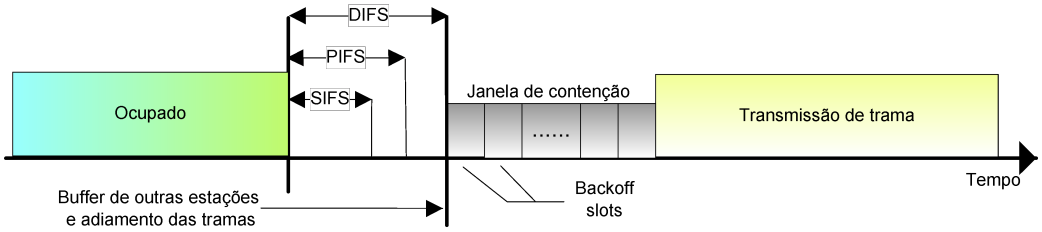


Figura 2.3: Relação entre os atrasos inter-tramas no 802.11.

A relação entre os atrasos encontra-se ilustrada na Figura 2.3, com excepção do EIFS.

Numa transmissão de uma sequência de operações como o RTS/CTS, para ganhar acesso ao meio têm que usar o DIFS como qualquer transmissão normal. Depois de ganhar acesso ao meio, esta sequência de operações usa SIFS para as restantes tramas da sequência. Assim, a sequência de operações vai ocupar o meio antes que outro tipo de tramas possam ser transmitidas. Com o uso de SIFS e NAV uma estação pode prender o meio pelo tempo necessário.

Acesso com contenção usando o DCF

As redes 802.11 usam na maioria dos casos o DCF, fornecendo um acesso idêntico ao das redes Ethernet baseado em contenção. Este pode ser usado em IBSS ou Infrastructure BSS, permitindo o acesso independente de múltiplas estações ao meio sem um controlo central.

Cada estação, antes de uma tentativa de transmissão, verifica se o meio se encontra livre. Caso não esteja, adia a transmissão usando um algoritmo de *backoff* de ordem exponencial para evitar colisões.

Existe um conjunto de regras que são sempre usadas e outras adicionais que dependem das circunstâncias. O DCF têm duas regras aplicadas a todas as transmissões:

1. Se o meio se encontra livre mais de DIFS, a transmissão pode começar imediatamente. O *carrier-sensing* é feito usando o meio físico e o NAV.

   (a) Se a trama recebida anteriormente não contém erros, o meio tem que ficar livre DIFS.

   (b) Se a trama recebida anteriormente contém erros, o meio tem que ficar livre EIFS.

2. Se o meio está ocupado, a estação tem que esperar até o meio ficar livre. Quando o acesso ao meio é adiado, a estação espera o meio livre durante DIFS e adicionalmente um período de *backoff* aleatório entre 0 e um máximo que é calculado usando o algoritmo de *backoff* de ordem exponencial.

As regras adicionais podem ser aplicadas em certas situações. Algumas dependem de situações particulares do meio e são específicas do resultado de transmissões anteriores.

1. A recuperação de erros é da responsabilidade da estação que enviou a trama. Por cada trama transmitida é esperado um ACK e cabe à estação que transmitiu a trama voltar a tentar até receber o reconhecimento positivo de que foi recebida.

   (a) O reconhecimento positivo indica sucesso. Nas sequências de operações, estas tem que ser completadas na sua totalidade para ter sucesso. Quando um ACK esperado não é recebido toda a sequência é considerada perdida.

   (b) Todos os dados enviados em *unicast* têm que ter reconhecimento positivo.

   (c) A falha na transmissão de uma trama provoca o incremento de um contador de tentativas e uma nova tentativa. É considerada uma falha quando falhou o ganho no acesso ao meio e não houve um reconhecimento positivo. No entanto, apenas as tentativas de transmissão provocam um aumento da janela de congestão.

2. As sequências de operações podem actualizar o NAV em cada passo da transmissão. Uma estação quando recebe um NAV maior que o actual, actualiza-o.

3. Apenas as tramas de ACK, CTS e tramas de uma sequência de fragmentos podem ser transmitidas usando o atraso SIFS, recebendo máxima prioridade.

   (a) Uma estação quando transmite a primeira trama de uma sequência de operação, ganha acesso ao meio. As restantes tramas podem ser enviadas com um atraso de SIFS, bloqueando as outras estações.

(b) As tramas adicionais da sequência actualizam o NAV para o tempo em que o meio vai permanecer em uso.

4. A extensão das sequências de tramas são necessárias para pacotes de camadas superiores, que podem ser maiores que os *thresholds* configurados.

   (a) Um pacote maior que o *threshold* RTS obriga ao uso do mecanismo de RTS/CTS.

   (b) Um pacote maior que o *threshold* de fragmentação tem que ser fragmentado.

Para recuperação de erros no DCF cada trama tem um contador de tentativas associado. O contador de tentativas pode ser de dois tipos, um para tramas com tamanho maior que o RTS *threshold* (*long retry count*) e outro para as tramas menores (*short retry count*). Estes começam a 0 e são incrementados a cada tentativa falhada de transmissão.

O *short retry count* é reiniciado a 0 quando:

- Uma trama de CTS é recebido em resposta ao RTS transmitido.
- Um ACK é recebido após a transmissão de uma trama com dados não fragmentados.
- Uma trama de *broadcast* ou *multicast* é recebida.

O *long retry count* é reiniciado a 0 quando:

- Um ACK é recebido após a transmissão de uma trama maior que o RTS *threshold*.
- Uma trama de *broadcast* ou *multicast* é recebida.

Às tramas que transportam dados fragmentados é associado um tempo de vida máximo. Este começa a contar quando é transmitido o primeiro fragmento. Se atingir o limite do tempo de vida e ainda houve fragmentos a serem transmitidos, a trama é descartada e não é efectuada nenhuma tentativa de enviar os restantes fragmentos.

Quando o contador de tentativas atinge o limite, a trama é descartada e é reportado às camadas superiores a sua perda.

No DCF é usado também uma janela de contenção ou janela de *backoff*, que é um atraso que as estações fazem após o meio ficar livre durante DIFS. Esta janela é divida em *slots* em que o seu tamanho depende do meio e da camada física que está a ser usada. As estações escolhem um *slot* aleatório e esperam por esse *slot* antes da tentativa de ganharem o acesso ao meio. A probabilidade de escolha do *slot* é igual para todos os *slots*. A estação que gerou o número aleatório mais baixo ganha.

O número de *slots* da janela aumenta numa potência de base 2 menos 1 por cada tentativa de retransmissão. Sucesso na transmissão reinicia o tamanho da janela. Existe também um tamanho máximo para a janela que quando é atingido mantém-se nesse valor até ser reiniciado ou o contador de tentativas atinge o limite e a trama é descartada.

Fragmentação

A camada MAC do 802.11 faz também a fragmentação de pacotes grandes. A fragmentação ajuda a combater a interferência nas transmissões, afectando apenas pequenos fragmentos e não tramas grandes. Isto reduz a quantidade de dados que podem ser corrompidos por interferência, além de resultar numa maior taxa de transferência.

As tramas com fragmentos contém todos o mesmo número de sequência mas diferente número de fragmento. Também é indicado se existem mais fragmentos a caminho. Normalmente os fragmentos são enviados numa sequência em *burst* com o mecanismo de RTS/CTS. Para cada fragmento é usado o atraso SIFS e reconhecimento positivo.

Encapsulamento dos protocolos de camadas superiores

O 802.11 pode transportar qualquer protocolo da camada de rede. Contrariamente ao Ethernet, o 802.11 depende do encapsulamento do LLC para transportar os protocolos para as camadas superiores.

Existem dois métodos diferentes usados para o encapsulamento de dados no LLC. Um descrito no RFC 1042[4] e outro no 802.1h. Ambos os métodos são muito idênticos e derivam do 802.2's SNAP (Sub-Network Access Protocol).

O LLC ao receber uma trama Ethernet da camada de cima, copia os endereços MAC de origem e destino para o cabeçalho de uma nova trama. Depois insere o cabeçalho SNAP que contém os campos DSAP (Destination Service Access Point),

SSAP (Source Service Access Point), um campo de controlo que é fixo a "0x03" e um campo com o OUI (Organizationally Unique Identifier). Da trama Ethernet original são ainda copiados os campos *Type* e os dados encapsulados, que é a trama IP. O FCS (Frame Check Sequence) é recalculado e colocado no final.

Ao construir a trama MAC 802.11, são usados os campos dos endereços MAC da trama obtida anteriormente para construir o cabeçalho 802.11. Os restantes campos são encapsulados nos dados da trama 802.11.

Na Figura 2.4[1] está ilustrado como é feito o encapsulamento usando o RFC 1042.

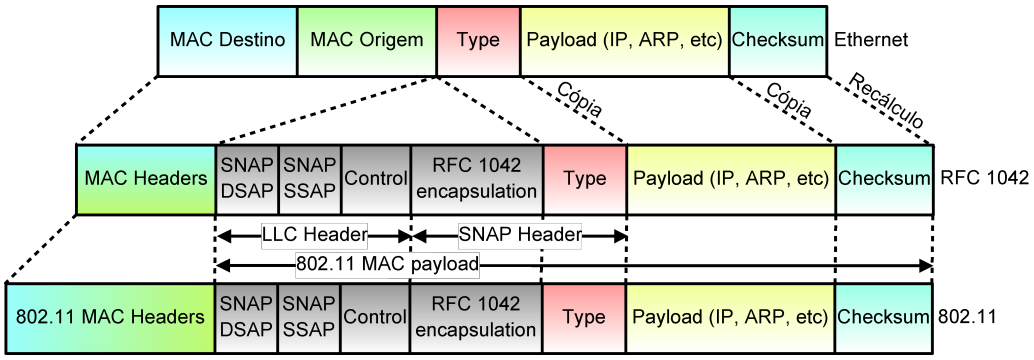


Figura 2.4: Encapsulamento do Ethernet em 802.11 usando o RFC 1042.

Poupança de energia

O 802.11 usa ondas de rádio frequência para comunicar e num sistema deste tipo existem amplificadores de sinal que consomem muita energia. O 802.11 permite às estações desligar ou adormecer o transceptor de rádio periodicamente de forma a maximizar a autonomia dessas estações. Nestes períodos o AP armazena as tramas *unicast* que chegam destinados às estações que se encontram a dormir. Essas tramas são anunciadas nas tramas de *Beacon* que são enviadas periodicamente pelo AP. Quando a estação acorda envia uma trama do tipo *PS-Poll* para receber as tramas que o AP tem armazenadas.

A resposta do AP a uma trama *PS-Poll* pode ser imediata ou adiada.

Resposta imediata - o AP espera um período SIFS e depois pode começar a transmitir a trama. A trama *PS-Poll* impõe também o NAV que contém o identificador da associação com o AP para que este possa determinar quais as tramas que estão armazenadas para a estação.

Resposta adiada - o AP responde com um ACK a indicar que recebeu o pedido mas que não actua imediatamente. As tramas armazenadas podem ser entregues a qualquer momento e a estação tem que permanecer acordada até as receber. Uma estação só pode voltar a adormecer quando a próxima trama de *Beacon* com o TIM (Traffic Indication Map) indicar que o AP não tem tramas armazenadas para a estação.

As tramas de *Beacon* indicam apenas se as estações têm ou não tramas armazenadas no AP que lhe são destinadas, não indicando o número de tramas.

2.2.4 Formato base da trama 802.11

O formato das tramas também faz parte da especificação da camada MAC do 802.11. No entanto, como é uma parte muito importante para este projecto, será abordado mais em detalhe nesta secção à parte da camada MAC em geral.

No 802.11 a camada MAC foi forçada a adoptar características únicas como o uso de quatro campos de endereços. Nem todos os tipos de tramas usam todos os campos de endereços e os valores designados para os campos com os endereços podem mudar consoante o tipo de trama MAC a ser transmitida.

A Figura 2.5 mostra uma trama MAC genérica do 802.11. As tramas aqui representadas seguem as convenções do IEEE. Os campos são transmitidos da esquerda para a direita, com o bit mais significativo a aparecer no fim.

Nem todos os campos são obrigatórios em todos os tipos e sub-tipos de tramas 802.11, no entanto, os primeiros 3 campos (*Frame Control*, *Duration/ID* e *Address 1*) e o último (*FCS*) constituem o mínimo e estão presentes em todas as tramas 802.11. Os restantes aparecem consoante o tipo e sub-tipo de trama e sempre pela mesma ordem como na Figura 2.5.

Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control
2 bytes	2	6	6	6	2

Address 4	QoS Control	HT Control	Frame Body	FCS
6	2	4	0-7955	4

Figura 2.5: Trama MAC genérica do 802.11.

A trama MAC do 802.11 não inclui no cabeçalho os campos típicos das tramas

Ethernet como o *type* e *length*. Estes são encapsulados nos dados transportados pela trama.

Campo *Frame Control*

Cada trama 802.11 começa com um campo de dois bytes *Frame Control*, representado na Figura 2.6. Os componentes deste campos são os seguintes:

Protocol version	Type	Sub-type	To DS	From DS	More Frag	Retry	Pwr Mgt	More Data	Protected Frame	Order
2 bits	2	4	1	1	1	1	1	1	1	1

Figura 2.6: Campo *Frame Control* da trama MAC do 802.11.

Protocol version - indica a versão do 802.11 MAC contida na trama. Actualmente apenas existe uma versão (valor 0). Novas versões só irão aparecer caso a camada MAC mude e se torne incompatível com a versão actual.

Type* e *Subtype - campos que identificam o tipo e sub-tipo de trama 802.11 usada. Os identificadores de tipo e sub-tipo são usados para criar diferentes classes de tramas, como mostra a Tabela 2.4. Na tabela os bits estão representados pelo bit mais significativo primeiro.

Tipo (b3b2)	**Sub-tipo (b7b6b5b4)**	**Descrição**
Management (00)	0000	Association request
	0001	Association response
	0010	Reassociation request
	0011	Reassociation response
	0100	Probe request
	0101	Probe response
	0110-0111	Reserved
	1000	Beacon
	1001	Announcement traffic indication message (ATIM)
	1010	Disassociation
	1011	Authentication
	1100	Deauthentication

continua na próxima página

	1101	Action
	1110	Action No Ack (norma 802.11n)
	1110-1111	Reserved
Control (01)	0000-0110	Reserved
	0111	Control Wrapper (norma 802.11n)
	1000	Block Acknowledgment Request (QoS)
	1001	Block Acknowledgment (QoS)
	1010	Power Save (PS)-Poll
	1011	RTS
	1100	CTS
	1101	Acknowledgment (ACK)
	1110	Contention-Free (CF)-End
	1111	CF-End+CF-Ack
Data (10)	0000	Data
	0001	Data+CF-Ack
	0010	Data+CF-Poll
	0011	Data+CF-Ack+CF-Poll
	0100	Null data (no data transmitted)
	0101	CF-Ack (no data transmitted)
	0110	CF-Poll (no data transmitted)
	0111	CF-Ack+CF-Poll (no data transmitted)
	1000	QoS Data
	1001	QoS Data + CF-Ack
	1010	QoS Data + CF-Poll
	1011	QoS Data + CF-Ack + CF-Poll
	1100	QoS Null (no data transmitted)
	1101	QoS CF-Ack (no data transmitted)
	1110	QoS CF-Poll (no data transmitted)
	1111	QoS CF-Ack+CF-Poll (no data transmitted)
Reserved (11)	0000-1111	Reserved

Tabela 2.4: Identificação dos tipos e sub-tipos de tramas 802.11.

ToDS* e *FromDS - os bits nestes campos indicam se a trama é destinada ou tem origem no DiS. Numa Infrastructure BSS pelo menos um dos bits vai estar a "1". A Tabela 2.5 mostra como esses bits são interpretados.

More Frag - quando uma trama foi fragmentada pela camada MAC, o primeiro fragmento e os que lhe seguem, excepto o último, tem este bit definido a "1". O bit a "1" indica que existem mais fragmentos. As tramas de dados e gestão podem ser grandes e necessitar de serem fragmentadas.

Valores *To DS* e *From DS*	**Significado**
$ToDS = 0$ $FromDS = 0$	Trama de dados enviada directamente de uma estação para outra dentro da mesma BSS (nunca tramas de dados enviadas dentro de uma Infrastructure BSS). Todas as tramas de gestão e controlo.
$ToDS = 1$ $FromDS = 0$	Trama de dados enviada de uma estação para o AP a que se encontra associada, dentro da mesma Infrastructure BSS.
$ToDS = 0$ $FromDS = 1$	Trama de dados recebida por uma estação enviada pelo AP a que se encontra associada, dentro da mesma Infrastructure BSS.
$ToDS = 1$ $FromDS = 1$	Trama de dados que usa os quatro endereços, normalmente enviada entre APs (pontes sem fios). No entanto, a norma não define procedimentos para usar esta combinação.

Tabela 2.5: Combinação dos campos *To DS*/*From DS* nas tramas 802.11.

Retry - quando uma trama é uma retransmissão de uma trama anterior, este bit é definido a "1". Na estação receptora, ajuda no processo de eliminar tramas duplicadas.

Power management - este bit é usado para indicar em que estado de gestão de energia vai ficar a estação após a sequência de operações de troca da trama. O bit "1" indica que a estação vai entrar no estado de poupança de energia. A "0" indica que vai permanecer activa. Um AP envia sempre este campo a "0", uma vez que nunca entra no estado de poupança de energia.

More data - este bit é usado quando existem tramas armazenadas no AP destinadas a uma estação que se encontra no estado de poupança de energia. O bit a "1" indica que existe pelo menos uma trama armazenada no AP que é destinada a essa estação.

Protected Frame - Quando os dados da trama 802.11 são protegidos por um algoritmo criptográfico este bit é "1". Apenas é posto a "1" nas tramas dos tipos que transmitem dados e para a trama de autenticação do tipo gestão.

Order - este bit é definido a "1" em duas situações:

- Uma trama de dados sem QoS transmitida por uma estação sem QoS indica que é transferida usando a classe de serviço "strict ordering".
- Uma trama de dados ou gestão com QoS transmitida que inclui o campo *HT Control*.

Campo *Duration/ID*

O campo *Duration/ID* é o que se segue depois do *Frame Control*. Aparece também em todas as tramas MAC do 802.11.

O conteúdo deste campo tem três formas distintas, sendo mais comum transportar o NAV:

- Quando o bit 15 é "0", o campo é usado para definir o NAV. Este representa o valor em micro-segundos que é esperado que o meio permaneça ocupado por uma transmissão em progresso. Todas as estações têm que fazer a monitorização dos cabeçalhos de todas as tramas que recebem e actualizar o NAV em conformidade. Qualquer valor que estenda o período de tempo que o meio está ocupado, o NAV é actualizado e bloqueia o acesso ao meio por um período adicional.
- Durante o período livre de contenção, o bit 14 é "0" e o bit 15 é "1". Os restantes são "0", assim o campo toma o valor "32768". Este valor é interpretado como o NAV e permite às estações que não receberam o *Beacon* a anunciar o período livre de contenção e actualizar o NAV com um valor grande o suficiente para evitar a interferir com as transmissões dentro desse período.
- Nas tramas *PS-Poll*, os bits 14 e 15 são "1". Os restantes bits incorporam o identificador de associação (AID, Association ID) dessa estação com o AP, indicando a que BSS pertence. O AID incluído na trama varia entre 1-2007, os valores entre 2008-16383 são reservados.

Campos de Endereços

Uma trama MAC 802.11 pode conter até quatro campos de endereços. Esses campos encontram-se numerados porque os campos são diferentes em diferentes situações, dependendo do tipo de trama. Regra geral, o campo *Address 1* é usado para

o receptor, o campo *Address 2* para o transmissor e o campo *Address 3* é usado para filtrar o receptor.

Os endereços usados são os mesmos das restantes redes IEEE 802, o MAC-48, constituído por 48 bits. Estes podem ser endereços individuais (*unicast*) ou de grupo (*broadcast* ou *multicast*). Dentro dos individuais, os endereços dividem-se em endereços administrados localmente (*locally administered*) ou universalmente (*universally administered*).

No contexto das redes 802.11 os endereços podem ser:

DA (Destination Address) - assim como na Ethernet, o endereço de destino corresponde ao endereço MAC que identifica o receptor final, a estação que vai usar a trama nas camadas superiores para processamento.

SA (Source Address) - endereço de origem que corresponde ao endereço MAC que identifica a origem da transmissão. A origem de uma trama apenas pode ser uma estação.

RA (Receiver Address) - endereço MAC que identifica, no meio sem fios, a estação que deve processar a trama. Se é uma estação sem fios, o endereço de receptor é o endereço de destino. Para tramas destinadas a um nó numa rede Ethernet ligada ao AP, o endereço de receptor é do *interface* sem fios do AP e o endereço de destino pode ser um router ligado na rede Ethernet.

TA (Transmitter Address) - endereço MAC que identifica a *interface* sem fios que transmitiu a trama no meio sem fios. Este endereço é usado apenas em pontes sem fios.

BSSID (Basic Service Set ID) - numa Infrastructure BSS indica endereço MAC que identifica a *interface* sem fios usada pelo AP. Numa IBSS indica o endereço MAC que é gerado aleatoriamente dos endereços administrados localmente, de forma a não entrar em conflito com os endereços atribuídos oficialmente (administrados universalmente).

O número de campos de endereços depende do tipo de trama. A maior parte das tramas de dados usa três campos de endereços para origem, destino e BSSID. É também por isso que existem três campos de endereços contíguos, porque a maior parte das transmissões usa estes três campos. O número e função dos campos de endereços depende como a trama de dados viaja relativamente ao DiS.

Campo *Sequence Control*

Este campo é usado na desfragmentação e rejeição de tramas duplicadas. O campo é dividido em dois sub-campos, um sub-campo de 4 bits que é usado para o número de fragmento e um outro sub-campo de 12 bits que é usado como número de sequência. Na Figura 2.7 encontram-se representados os campos.

Fragment number	Sequence number
4 bits	12

Figura 2.7: Campo *Sequence Control* da trama MAC do 802.11.

Sequence number - a cada trama de dados e gestão é atribuído um número de sequência. Em estações sem QoS, o número de sequência opera sobre um contador até 4095 das tramas transmitidas. Nas estações com QoS são mantidos vários desses contadores, um por cada identificador de tráfego (TID, Traffic Identifier) presente no campo *QoS Control* e um contador adicional para as tramas de gestão, dados QoS com destino a um grupos de estações e dados enviados sem QoS. O contador começa em 0 e incrementa 1 por cada pacote que passa e quando atinge o limite volta a 0. Em tramas fragmentados todos os fragmentos têm o mesmo número de sequência e em tramas retransmitidas o número de sequência não se altera.

Fragment number - indica o número de fragmento numa trama fragmentada. Ao primeiro fragmento é atribuído o número "0" a que depois é incrementado 1 por cada fragmento. O número de fragmento permanece constante nas retransmissões.

Campo *QoS Control*

Este campo é usado para identificar a categoria de tráfego a que uma trama pertence e outras informações relacionadas com a qualidade de serviço, que varia com o tipo e subtipo de trama. Este campo está presente em tramas de dados com o bit de QoS a "1" no subtipo.

Campo *HT Control*

Este campo foi adicionado na última emenda da norma (802.11n-2009[3]). Encontra-se presente na trama de controlo *Control Wrapper*, tramas de dados com QoS e

tramas de gestão com o *Order Bit* definido. Este define um conjunto de sub campos com informação para controlo de adaptação da ligação, calibração de posição e sequência, tipo de *feedback*, etc.

Campo *Frame body*

Neste campo são transportados os dados da trama. Aqui encontram-se encapsulados os protocolos das camadas superiores de rede. Caso seja uma trama protegida, poderá incluir *overhead* adicional.

Campo *FCS*

Assim como na Ethernet, as tramas 802.11 terminam com o FCS. Este campo é usado para verificar a integridade da trama no receptor. Neste campo é calculado o CRC (Cyclic Redundancy Code) de 32 bits sobre todos os outros campos presentes na trama.

2.2.5 Tramas de dados

As tramas de dados são usadas para transportar os dados das camadas superiores de rede. Os diferentes sub-tipos de tramas de dados podem ser categorizados de acordo com a função. Existem sub-tipos usados para serviços livre de contenção e serviços com contenção. Os sub-tipos usados nos serviços livre de contenção, apenas são usados nas Infrastructure BSS. Existe também a divisão dos sub-tipos que transportam dados com QoS, que incluem o campo *QoS Control*. Algumas tramas de dados não transportam dados, apenas são usadas para funções de gestão. Na Tabela 2.6 mostra-se como as tramas de dados são categorizadas.

Campo *Frame Control*

Todos os bits do campo *Frame Control* são usados de acordo com as regras já descritas. Estes afectam a interpretação dos outros campos do cabeçalho onde é mais notável os campos de endereços.

Campo *Duration/ID*

Este campo contém o tempo que o acesso ao meio é restrito, o NAV. Nas tramas de dados existem quatro regras que dizem respeito a este campo:

Sub-tipo	Serviço com contenção	Serviço livre de contenção	Dados	QoS
Data	√		√	
Data+CF-Ack		√	√	
Data+CF-Poll		AP only	√	
Data+CF-Ack+CF-Poll		AP only	√	
Null data	√	√		
CF-Ack		√		
CF-Poll		AP only		
CF-Ack+CF-Poll		AP only		
QoS Data	√		√	√
QoS Data + CF-Ack		√	√	√
QoS Data + CF-Poll		AP only	√	√
QoS Data + CF-Ack + CF-Poll		AP only	√	√
QoS Null	√	√		√
QoS CF-Ack		√		√
QoS CF-Poll		AP only		√
QoS CF-Ack+CF-Poll		AP only		√

Tabela 2.6: Categorias das tramas de dados 802.11.

1. Em tramas transmitidas durante o período livre, este campo tem valor "32768".

2. Em tramas transmitidas para um endereço de grupo *broadcast* ou *multicast*, este campo tem valor "0". Neste tipo de tramas não é necessário o reconhecimento positivo e após terminar a transmissão da trama o acesso ao meio com contenção pode começar imediatamente.

3. Se o bit *More Fragments* no *Frame Control* estiver a "0", não existe mais fragmentos para transmitir. O último fragmento apenas precisa de reservar o meio para o próprio ACK. O penúltimo fragmento faz a reserva para o último fragmento.

4. Se o bit *More Fragments* no *Frame Control* estiver a "1", existem mais fragmentos para transmitir. No campo *Duration* é definido o tempo necessário para dois ACKs, mais três SIFS, mais o tempo necessário para o próximo fragmento. Fragmentos não finais definem o NAV como um RTS faz, daí ser referido como um "RTS virtual".

Endereçamento e campos *To DS* e *From DS*

O número e função dos campos de endereços nas tramas de dados depende de como os bits nos campos *To DS* e *From DS* estão definidos no campo *Frame Control*.

O uso dos campos de endereços indirectamente depende do tipo de rede sem fios. A Tabela 2.7[5] sumaria o uso dos campos de endereços em tramas de dados.

Função	***ToDS***	***FromDS***	***Address 1***	***Address 2***	***Address 3***	***Address 4***
IBSS	0	0	RA=DA	SA	BSSID	N/A
Do AP	0	1	RA=DA	BSSID	SA	N/A
Para AP	1	0	RA=BSSID	SA	DA	N/A
WDS (ponte)	1	1	RA	TA	DA	SA

Tabela 2.7: Campos de endereços nas tramas de dados 802.11.

O campo *Address 1* indica o receptor da trama que em muitos casos é o destinatário, mas nem sempre. Caso o campo *Address 1* seja um endereço de grupo *broadcast* ou *multicast*, o BSSID é também verificado. As estações respondem apenas a endereços de grupo originados na mesma BSS e ignoram os de BSS diferentes. O campo *Address 2* é o endereço do transmissor e é usado para responder com os ACKs. O campo *Address 3* é usado para filtragem no AP e DiS, no entanto este depende do tipo particular de rede.

No caso das redes IBSS, em que não são usados APs e o DiS não está presente, o transmissor é a origem, e o receptor é o destino. Todas as tramas trazem o campo de *bssid* para a estações verificarem as tramas de *broadcast* e *multicast* e apenas se pertencerem à mesma BSS são processados. O BSSID é gerado aleatoriamente dentro dos endereços administrados localmente.

O 802.11 estabelece a diferença entre origem e transmissor e paralelamente entre destino e receptor. O transmissor envia a trama para o meio sem fios mas não foi necessariamente o criador da trama. Esta diferença é importante para o 802.11 uma vez que os ACKs são enviados para o endereço do transmissor e as camadas superiores respondem para o endereço de origem da trama. Um receptor pode ser um intermediário para o destino e as tramas apenas são processadas pelos protocolos das camadas superiores quando chegam ao destino.

Consideremos o uso dos campos de endereços numa Infrastructure BSS, onde existe uma estação sem fios que faz a ligação a um servidor ligado à rede por Ethernet através de um AP. As tramas enviadas da estação para o servidor usam os endereços especificados na terceira linha da Tabela 2.7. Quando as tramas são enviadas para um destino no DiS, a estação é ao mesmo tempo a origem e o transmissor da trama. O receptor da trama no meio sem fios é o AP que é apenas um intermediário para o destino. A trama é envia pelo AP através do DiS para o destino que é o servidor.

Nas Infrastructure BSS os APs criam a associação de BSSs com o endereço do seu *interface* sem fios, daí o endereço do receptor (*Address 1*) ser o BSSID.

Quando o servidor responde à estação as tramas são transmitidas através do AP, o que corresponde à segunda linha da Tabela 2.7. A trama é criada no servidor, logo o endereço de origem é o endereço MAC do servidor. A trama será recebida pelo AP através da *interface* Ethernet e transmitida depois pelo *interface* sem fios do AP para a estação. Como no caso anterior, o endereço da *interface* sem fio do AP é também o BSSID. Neste caso o AP é o transmissor, a estação é o receptor e destino da trama.

A quarta linha da Tabela 2.7 mostra o uso dos endereços numa ponte sem fios ou num sistema de distribuição sem fios (WDS, Wireless Distribution System). Quando duas redes fixas são unidas por APs, os endereços do transmissor e receptor nas tramas sem fios são os endereços das *interfaces* sem fios dos APs. Os endereço de origem e destino são de nós no meio Ethernet.

Variações nas tramas de dados

O 802.11 usa diferentes sub-tipos de tramas de dados. As variações dependem se é um serviço com contenção ou livre de contenção. No serviço livre de contenção as tramas incorporam várias funções que aumentam a eficiência. Os dados são transmitidos com um sub-tipo diferente e assim é usado ao mesmo tempo como um ACK, poupando assim espaço entre tramas e ACK separados. Na Tabela 2.6 estão identificados os sub-tipos.

Os tipos comuns mais usados nos serviços com contenção são:

Data - são transmitidas apenas durante o período com contenção. São tramas com o propósito de apenas transportar dados de uma estação para outra. Se for uma estação QoS em que a trama inclui o campo de *QoS Control*, o sub-tipo muda.

Null - consiste num cabeçalho MAC 802.11 com o campo *FCS*. Não contem dados e são usados pela estações para informar o AP de mudanças no seu estado de poupança de energia. Uma estação quando vai adormecer e não têm dados a enviar para o DiS, envia uma trama *Null* com o bit de *Power Management* do *Frame Control* a "1". O AP começa depois a armazenar tramas para a estação. Se for uma estação QoS em que a trama inclui o campo *QoS Control*, o sub-tipo muda.

Existem outros sub-tipos de tramas para uso nos serviços livres de contenção, no entanto o serviço livre de contenção foi pouco implementado e por isso estes sub-tipos não serão abordados.

Aplicação das tramas de dados

A forma das tramas de dados depende do tipo de rede sem fios. O sub-tipo de trama de dados é apenas determinado pelo campo sub-tipo no *Frame Control* e não pela presença ou não de campos.

Numa IBSS são usados três campos de endereços. O primeiro endereço identifica o receptor que é também o destino da trama. O segundo endereço identifica a origem. O terceiro endereço identifica o BSSID. Quando a camada MAC recebe uma trama, verifica o endereço de BSSID e apenas passa para as camadas superiores se é o endereço de BSSID a que a estação está actualmente associada. Nestas redes são usados os sub-tipos de tramas *Data* e *Null*.

Numa Infrastructure BSS em que a transmissão é efectuada do AP ou para o AP são usados três campos de endereços. Quando a transmissão é efectuada a partir do AP podem ser usados todos os sub-tipos com excepção do *Null*. Quando a transmissão é feita para o AP podem ser usados os sub-tipos *Data*, *Null*, *Data+CF-ACK* e *Null+CF-ACK*. Neste caso também pode ser adicionado QoS.

Numa ponte sem fios são usados os quatro campos de endereços. Dois dizem respeito aos endereços da ligação sem fios, o transmissor e receptor, que são usados para os ACKs e RTS/CTS. Os outros dois dizem respeito à ligação origem e destino, que diferem da ponte de ligação sem fios. Os sub-tipos de tramas usados neste caso são apenas *Data*.

Tramas que usam protecção com cifra tem apenas o bit de *Protected Frame* do *Frame Control* a "1" e não são um novo tipo. Os dados que transportam no campo *Frame Body* começam com um cabeçalho de protecção.

2.2.6 Tramas de controlo

As tramas de controlo fornecem funções que são usadas para assistir à entrega das tramas de dados. Administram o acesso ao meio (não o meio em si). Em junção com as tramas de dados permitem a entrega de dados de forma fiável entre estações.

Campo *Frame Control*

No campo *Frame Control* das tramas de controlo, entre diferentes tramas apenas alternam os bits dos campos *Subtype* e *Pwr Mgt*. Os restantes são fixos como mostra a Figura 2.8. As tramas de controlo não podem ser fragmentadas ($MoreFrag = 0$), não são retransmitidas ($Retry = 0$), não são cifradas ($ProtectedFrame = 0$) e não podem ser transmitidas fora de ordem ($Order = 0$).

Protocol	Type	Sub-type	To DS	From DS	More Frag	Retry	Pwr Mgt	More Data	Protected Frame	Order
2 bits	2	4	1	1	1	1	1	1	1	1
0 0	1 0		0	0	0	0		0	0	0

Figura 2.8: Campo *Frame Control* das tramas do tipo controlo 802.11.

Trama RTS

As tramas RTS são usadas para ganhar o controlo do meio para transmitir tramas "grandes", em que "grandes" é definido por um *threshold* RTS no controlador da placa de rede sem fios. O acesso ao meio apenas pode ser reservado para tramas *unicast*. O formato de uma trama RTS está representado na Figura 2.9.

Frame Control	Duration /ID	RA	TA	FCS
2 bytes	2	6	6	4

Figura 2.9: Trama RTS do 802.11.

A trama RTS é apenas o cabeçalho, como todas as tramas de controlo. O cabeçalho contém quatro campos, o *FCS* segue-se logo após o cabeçalho.

Frame Control - campo onde apenas é definido o sub-tipo de uma trama RTS.

Duration - uma trama RTS reserva o meio durante uma sequência completa de operações. O cálculo da duração terá que ser efectuado sobre a sequência completa que vai haver após o envio do RTS pelo transmissor. O cálculo inclui três períodos SIFS, a duração de um CTS, um ACK final e o tempo necessário para transmitir a trama ou fragmento. Os fragmentos seguintes actualizam o campo *Duration* neste caso.

***Address 1*/RA** - endereço da estação que é pretendida como receptor da trama "grande".

***Address 2*/TA** - endereço do remetente da trama RTS.

Trama CTS

Uma trama CTS é usada como resposta a uma trama RTS. O formato de uma trama CTS está representado na Figura 2.10.

Frame Control	Duration /ID	RA	FCS
2 bytes	2	6	4

Figura 2.10: Trama CTS do 802.11.

A trama CTS é constituída por um cabeçalho com três campos.

Frame Control - neste campo apenas é definido o sub-tipo de uma trama CTS.

Duration - o remetente da trama CTS usa a duração da trama RTS como base de cálculo para a duração. É subtraído à duração o tempo de uma trama CTS e um SIFS e o resultado é colocado no campo *Duration*.

***Address 1*/RA** - o receptor da trama CTS é o transmissor da trama RTS anterior. A camada MAC copia o endereço do transmissor da trama RTS para o endereço de receptor da trama CTS.

Trama ACK

As tramas ACK são usadas para enviar o reconhecimento positivo, que é requerido pela camada MAC e usado em qualquer transmissão. O formato de uma trama ACK está representado na Figura 2.11.

Frame Control	Duration /ID	RA	FCS
2 bytes	2	6	4

Figura 2.11: Trama ACK do 802.11.

A trama ACK é constituída por um cabeçalho com três campos.

Frame Control - neste campo apenas é definido o sub-tipo de uma trama ACK.

Duration - a duração pode ser definida de duas formas, dependendo da posição do ACK na troca de tramas. ACKs que completam tramas de dados ou o

fragmento final de um *burst* de fragmentos, o campo é definido "0". Na trama anterior recebida, o bit do campo *More Fragments* do *Frame Control* a "0", indicando que a transmissão está completa. Se o bit do campo *More Fragments* é "1", o *burst* de fragmentos encontra-se em progresso. Neste caso o campo *Duration* é calculado da mesma forma que na trama de CTS, sendo subtraído a duração do fragmento mais recente e um SIFS.

***Address 1*/RA** - o endereço do receptor é copiado da trama que foi transmitida e que está reconhecida positivamente. Tecnicamente, é a copia do campo *Address 2*/TA da trama que está ser reconhecida positivamente. Os ACKs são transmitidos em resposta a tramas de dados, tramas de gestão e tramas *PS-Poll* enviadas directamente.

Trama PS-Poll

Quando uma estação acorda do estado de poupança de energia, transmite a trama *PS-Poll* para o AP para recuperar as tramas que foram armazenadas no AP enquanto estava no estado de poupança de energia. O formato de uma trama *PS-Poll* está representado na Figura 2.12.

Frame Control	AID	BSSID (RA)	TA	FCS
2 bytes	2	6	6	4

Figura 2.12: Trama PS-Poll do 802.11.

A trama *PS-Poll* é constituída por um cabeçalho com quatro campos.

Frame Control - neste campo apenas é definido o sub-tipo de uma trama *PS-Poll*.

AID - em vez do campo *Duration*, este tipo de trama usa o terceiro e o quarto byte do cabeçalho para o AID. O AID é um valor numérico atribuído pelo AP para identificar uma associação. Ao incluir o AID na trama, o AP consegue encontrar as tramas armazenadas para a estação que acordou.

***Address 1*/BSSID (RA)** - este campo contém o BSSID da BSS criada pelo AP a que o remetente está actualmente associado.

***Address 2*/TA** - endereço do remetente da trama *PS-Poll*.

A trama *PS-Poll* não inclui informação de duração para actualizar o NAV, no entanto, quando as estações recebem uma trama *PS-Poll*, actualizam o NAV para uma duração de um SIFS mais o tempo necessário para transmitir um ACK. Esta actualização automática do NAV permite ao AP transmitir um ACK com uma pequena probabilidade de colisão.

Tramas CF-End e CF-End+CF-ACK

Os sub-tipos de tramas de controlo *CF-End* e *CF-End+CF-ACK* são usadas nos serviços livres de contenção PCF. Como este muito raramente é implementado nas redes 802.11, estes sub-tipos de tramas não são analisados.

Tramas de Block Acknowledgement (BA)

Este sub-tipo de tramas de controlo foi definido inicialmente na norma IEEE 802.11e como opção de forma a melhorar a eficiência da camada MAC. Foi depois rectificado na emenda IEEE 802.11n-2009[3] e tornou-se obrigatório em todos os dispositivos HT (High Throughput).

Em vez de ser transmitido um ACK individual para cada trama ou fragmento, múltiplas tramas podem ser reconhecidas positivamente usando uma trama de reconhecimento positivo de bloco (BA, Block Acknowledgement). A trama *BlockAck* contém uma mapa de bits de 64*16 bits. Esses 16 bits contém o número de fragmento que está a ser reconhecido positivamente. Cada bit do mapa representa o estado (sucesso ou falha) da trama.

O BA começa com uma fase inicial de configuração em que é trocada a informação sobre as capacidades e as políticas BA com o receptor através do uso de tramas *Action*. Após essa fase o envio das tramas pode ser efectuado sem esperar pelos ACKs e no fim efectuado reconhecimento positivo usando o BA. A sequência de operações termina o envio de uma trama *Action*.

São definidas duas tramas de BA. A trama *BlockAckReq* que faz o pedido à estação receptora para confirmar a recepção do bloco de tramas. O formato da trama *BlockAckReq* está representado na Figura 2.13.

A trama *BlockAck* é a resposta a esse pedido com o respectivo mapa de bits. O formato da trama *BlockAck* está representado na Figura 2.14.

Ambas as tramas são muitos idênticas e apenas são analisados os campos mais relevantes.

Frame Control	Duration /ID	RA	TA	BAR Control	BAR Information	FCS
2 bytes	2	6	6	2	variable	4

Figura 2.13: Trama BlockAckReq do 802.11.

Frame Control	Duration /ID	RA	TA	BA Control	BA Information	FCS
2 bytes	2	6	6	2	variable	4

Figura 2.14: Trama BlockAck do 802.11.

Frame Control - neste campo apenas é definido o respectivo sub-tipo de um trama *BlockAckReq* e *BlockAck*.

Duration/ID - o valor deste campo segue as regras definidas na especificação para uma estação com suporte para QoS.

Address 1**/RA** - endereço da estação que é pretendida como receptor da trama.

Address 2**/TA** - endereço do remetente da trama.

Trama Control Wrapper

Este sub-tipo de trama é usado para transportar qualquer trama do tipo controlo (excepto *Control Wrapper*) em conjunto com o campo *HT Control*. Foi acrescentado na emenda IEEE 802.11n-2009[3]. O formato da trama *Control Wrapper* está representado na Figura 2.15.

Frame Control	Duration /ID	Address 1	Carried Frame Control	HT Control	Carried Frame	FCS
2 bytes	2	6	2	4	variable	4

Figura 2.15: Trama Control Wrapper do 802.11.

O valor do campo *Duration* é gerado usando as mesmas regras da trama de controlo que transporta, assim como o campo *Address 1*. O campo *Carried Frame Control* contém o valor do campo *Frame Control* da trama de controlo que transporta. O campo *Carried Frame* transporta o conteúdo da trama de controlo com os campos que aparecem depois do *Address 1* e excluindo o *FCS* no fim.

2.2.7 Tramas de gestão

A gestão é um componente grande na especificação 802.11. Existem diferentes sub-tipos de tramas de gestão que são usados para providenciar serviços que normalmente são simples nas redes Ethernet. O 802.11 divide os procedimentos de gestão em três componentes. Uma estação que procura por conectividade tem que primeiro encontrar uma rede sem fios compatível. Depois a rede tem que autenticar a estação, verificando se é permitido à estação a ligação à rede sem fios. No fim, a estação tem que se associar com o AP de forma a ganhar acesso à rede sem fios.

Trama genérica de gestão

As tramas de gestão têm um formato que é partilhado por todos sub-tipos. O formato está representado na Figura 2.16.

Frame Control	Duration /ID	DA	SA	BSSID	Sequence Control	Frame Body	FCS
2 bytes	2	6	6	6	2	0-2312	4

Figura 2.16: Trama MAC de gestão do 802.11.

O cabeçalho MAC de todas as tramas de gestão é o mesmo e não depende do sub-tipo de trama. Algumas tramas de gestão usam o *Frame Body* para transmitir informação específica ao sub-tipo de trama de gestão. O campo *Duration* é calculado da mesma forma que nas tramas de dados.

Campos de endereços

O primeiro campo de endereços é usado para o destinatário da trama e o segundo campo de endereço para a origem. Algumas tramas de gestão são usadas para manter as propriedades dentro de uma BSS. De forma a limitar o efeito das tramas de gestão enviadas em *broadcast* e *multicast*, as estações verificam o BSSID depois de receber a trama de gestão. Apenas as tramas de *broadcast* e *multicast* enviadas pela BSSID a que a estação se encontra actualmente associada são passadas para a camada de gestão MAC, com uma única excepção para as tramas de *Beacon* usadas para anunciar a existência de uma rede 802.11.

Os APs usam o endereço MAC do *interface* sem fios como BSSID. As estações usam o endereço de BSSID do AP com que se encontram actualmente associadas.

As estações numa IBSS usam um BSSID gerado aleatoriamente dentro dos endereços administrados localmente quando é criada a BSS. Existe no entanto uma excepção, tramas enviadas por uma estação à procura de uma rede específica, podem usar o BSSID da rede que procuram ou usar o endereço de *broadcast* como BSSID para encontrar redes na vizinhança.

Campo *Frame Body*

Nas tramas de gestão este campo contém elementos de informação de tamanho variável e campos de tamanho fixo, chamados também *information elements* e *fixed fields* respectivamente. Os elementos de informação são objectos básicos grandes (BLOB, Basic Large Objects) de dados etiquetados com número do tipo, o seu tamanho e o conteúdo de um certo tipo é interpretado de certa maneira. Novos elementos de informação podem ser definidos em novas revisões da especificação do 802.11 e as implementações anteriores ignoram os novos elementos.

O 802.11 específica a ordem com que os elementos de informação aparecem, no entanto nem todos os elementos são obrigatórios. Estes elementos de informação variam também conforme o sub-tipo de trama de gestão.

Alguns dos campos de tamanho fixo encontram-se presentes na Tabela 2.8.

Nome	Descrição
Authentication Algorithm Number	Identifica o tipo de autenticação usado no processo de autenticação.
Authentication Transaction Sequence Number	Número de sequência usado para seguir o progresso na troca de tramas no processo de autenticação. Este processo tem vários passos, que consiste num desafio enviado pelo AP.
Beacon interval	Intervalo entre transmissões do *Beacon* em unidades de tempo (TU, Time Units).
Capability Information	Informação sobre as capacidades da rede sem fios. Usado no *Beacon*, *Probe Request* e *Probe Response*.
Current AP Address	Contém o endereço MAC do AP que a estação se encontra associada. Usado para associações e re-associações.
Listen interval	Número de intervalos de *Beacon* que uma estação permanece adormecida.
Association ID	Identificador de associação atribuído na associação com o AP.
Timestamp	Tempo que permite a sincronização entre estações.
Reason Code	Indica ao remetente o que fez incorrecto nas tramas de *Disassociation* e *Deauthentication*.
Status Code	Indica o sucesso ou falha de uma operação.

Tabela 2.8: Campos de informação fixos de uma trama de gestão 802.11.

Alguns dos elementos de informação encontram-se presentes na Tabela 2.9.

ID	Nome	Descrição
0	SSID (Service Set Identity)	É um nome que permite identificar uma BSS. Este nome é igual para as BSA que formam uma ESA. O tamanho varia entre 0 e 32 bytes.
1	Supported Rates	Contém a informação das taxas de transmissão suportadas, em que algumas são obrigatórias e outras opcionais.
2	FH Parameter Set	Contém os parâmetros necessários para se juntar a uma rede com modulação FHSS.
3	DS Parameter Set	Contém o parâmetro para se juntar a uma rede com modulação DSSS, que é apenas o número de canal usado.
4	CF Parameter Set	Contém elementos de informação para o serviço livre de contenção PCF.
5	TIM	É um mapa de bits virtual que indica os AIDs que têm tramas armazenadas no AP.
6	IBSS Parameter Set	Contém a janela com o ATIM (Announcement Traffic Indication Map) para as redes IBSS.
16	Challenge text	Contém o desafio enviado para autenticação em *shared-key*.

Tabela 2.9: Elementos de informação de uma trama de gestão 802.11.

Tipos de tramas de gestão

O 802.11 contém um conjunto de sub-tipos de tramas de gestão, que são:

Beacon - as tramas de *Beacon* anunciam a existência de uma rede sem fios e são uma parte importante em várias tarefas de manutenção da rede. São transmitidas em intervalos regulares, de forma a permitir às estações encontrar e identificar redes, e configurar certos parâmetros para se juntar à rede. Numa Infrastructure BSS, o AP é responsável por transmitir as tramas de *Beacon*. A área em que o *Beacon* aparece, define a BSA.

Probe Request - as estações usam a trama de *Probe Request* para encontrar numa área uma rede 802.11. As estações (geralmente um AP) que recebem o *Probe Request* usam a informação presente para determinar se as estações se podem juntar à rede. Isto acontece quando a estação suporta as taxas de transmissão requeridas pela rede e se quer juntar à rede com o SSID identificado. Normalmente as placas permitem ligar a uma qualquer rede e no *Probe Request* usam o endereço de *broadcast* SSID.

Probe Response - quando o *Probe Request* encontra uma rede com parâmetros compatíveis, a rede envia uma trama *Probe Response*. Numa Infrastructure BSS o AP que enviou o último *Beacon* é responsável por responder aos *Probes* que lhe chegam. Numa IBSS a responsabilidade é distribuída, onde uma estação após transmissão de um *Beacon*, assume a responsabilidade de enviar a trama *Probe Response* durante o proximo intervalo de *Beacon*.

ATIM - numa rede IBSS não existe um AP, por isso o armazenamento das tramas destinadas a uma estação em poupança de energia é efectuado em cada estação. É enviado uma trama *ATIM* durante o período de entrega a notificar que existem tramas armazenadas.

Disassociation e ***Deauthentication*** - tramas de *Disassociation* são usadas para terminar uma relação de associação, as tramas de *Deauthentication* são usadas para terminar uma relação de autenticação. Estas tramas incluem um campo *Reason Code*.

Association Request - assim que uma estação encontra uma rede compatível e se autentica com a mesma, pode tentar juntar-se à rede ao enviar um trama *Association Request*. O campo *Capability Information* presente na trama indica o tipo de rede a que a estação se pretende juntar. O AP antes de aceitar o pedido de associação, verifica se o *Capability Information*, *SSID* e *Supported Rates*, são compatíveis. O AP também anota o *Listen Interval* da estação.

Reassociation Request - estações que se movem entre BSA na mesma ESA necessitam de fazer re-associação com a rede antes de usar o DiS. Isto também pode acontecer quando uma estação se afasta temporariamente da área de cobertura do AP e se volta a juntar mais tarde. A diferença para o *Association Request* é que o *Reassociation Request* inclui o endereço do AP a que a estação se encontra actualmente associada. Isto permite ao novo AP contactar o AP antigo e transferir os dados de associação que podem incluir tramas armazenadas.

Association Response e ***Reassociation Response*** - quando uma estação tenta a associação com um AP, este responde uma trama de *Association Response* ou *Reassociation Response*. A diferença entre os dois é apenas o campo *Subtype* no *Frame Control*. Parte da resposta é o AID atribuído pelo AP.

Authentication - as estações têm que fazer a autenticação perante o AP através da troca de tramas de *Authentication*. O processo de autenticação pode involver um conjunto de passos que dependem do algoritmo de autenticação usado.

Action* e *Action No Ack - outras operações de gestão que não tenham sido definidas inicialmente usam as tramas *Action*. Contêm um campo *Action* que é um elemento de informação que determina a operação de gestão pretendida. Operações de gestão do espectro (802.11h) e QoS (802.11e) são alguns exemplos que usam este tipo de trama.

2.2.8 Transmissão de Tramas e Estados na Autenticação e Associação

Os tipos e sub-tipos de tramas que são permitidos variam com o estado em que uma estação se encontra. As estações podem estar autenticadas ou não autenticadas, associadas ou não associadas. Estas duas variáveis podem ser combinadas em três estados.

1. Estado inicial, não autenticada e não associada.
2. Autenticada e não associada.
3. Autenticada e associada.

Cada estado é sucessivamente mais avançado no desenvolvimento de uma ligação 802.11. Todas as estações começam no estado 1 e apenas no estado 3 estas podem transmitir dados através do DiS numa Infrastructure BSS. Numa IBSS, como não existe associação com um AP, apenas atinge o estado 2.

Na Figura 2.17 encontra-se representado o diagrama de estados. Podemos identificar que as tramas são dividas em classes, que as tramas de classe 1 podem ser transmitidas no estado 1, tramas classes 1 e 2 no estado 2 e classes 1,2 e 3 no estado 3. As tramas permitidas por classe encontram-se representados na Tabela 2.10 e as classes são divididas da seguinte forma:

Classe 1 - são usadas nas operações básicas pelas estações 802.11. Tramas de controlo recebidas e transmitidas de forma a manter o "respeito" no acesso ao meio e para transmitir tramas numa IBSS. As tramas que permitem a uma estação encontrar e fazer a autenticação com uma rede sem fios são também permitidas.

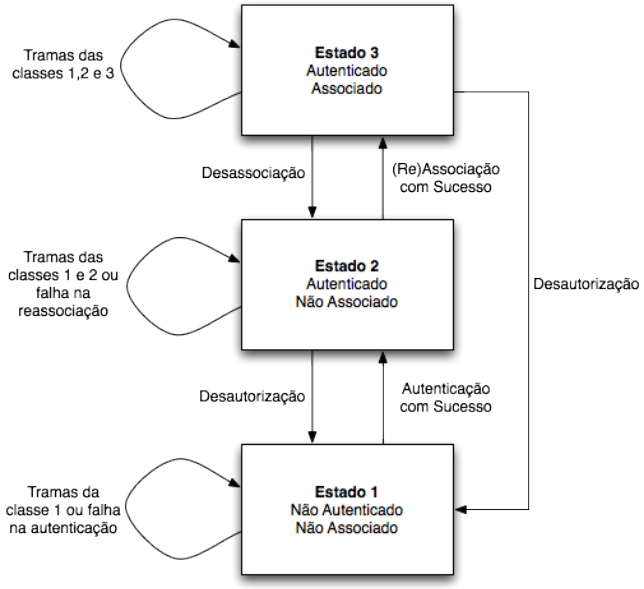


Figura 2.17: Diagrama dos estados das tramas permitidas do 802.11.

Classes	Controlo	Gestão	Dados
Classe 1	*RTS*, *CTS*, *ACK*, *CF-End*, *CF-End+CF-Ack*	*Probe Request*,*Probe Response*, *Beacon*, *Authentication*, *Deauthentication*, *ATIM*	Qualquer trama com os campos $ToDS = 0$ e $FromDS = 0$.
Classe 2	Nenhum	*Association Request*, *Association Response*, *Reassociation Request*, *Reassociation Response*, *Disassociation*	Nenhum
Classe 3	*PS-Poll*	*Deauthentication*	Qualquer tramas de dados.

Tabela 2.10: Divisão das tramas 802.11 em classes.

Classe 2 - tramas que podem ser transmitidas depois de uma estação efectuar a autenticação com sucesso. Podem ser usadas nos estados 2 e 3 para as operações de associação.

Classe 3 - tramas que são usadas quando uma estação efectuou com sucesso a autenticação e associação com o AP. Quando uma estação atinge o estado 3 pode usar os serviços do DiS e serviços de poupança de energia fornecidos pelo AP.

2.2.9 Operações de Gestão

Não estar preso a uma rede com um cabo pode ter vantagens mas também levanta alguns problemas. O meio sem fios não é confiável, utilizadores não autorizados têm vantagem porque não existe uma barreira física, consumo de energia em dispositivos que funcionam a baterias, etc. As características de gestão do 802.11 foram desenvolvidas para reduzir o efeito desses problemas.

Arquitectura de Gestão

O 802.11 é constituído por um conjunto de três entidades que fazem as gestão das suas camadas físicas e MAC. A relação entre essas entidades de gestão e as camadas do 802.11 encontra-se representada na Figura 2.18. Existe uma entidade de gestão da camada física (PLME, Physical Sublayer Management Entity), outra entidade para a gestão da camada MAC (MLME, MAC Sublayer Management Entity) e a entidade de gestão do sistema (acsme).

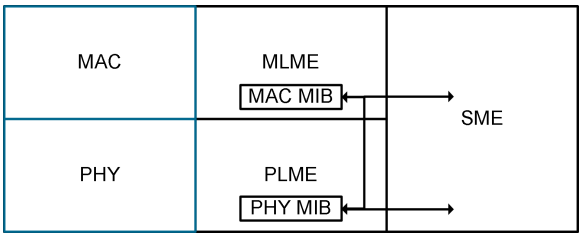


Figura 2.18: Arquitectura dos componentes de gestão do 802.11.

O 802.11 não especifica o SME, System Management Entity. Este é o método pelo qual os utilizadores e o controlador interagem com o *interface* de rede 802.11, fazendo a recolha de informação acerca do mesmo. A camada MAC e física do 802.11 tem acesso a uma MIB (Management Information Base)[1].

Existem três *interfaces* definidos entre os componentes de gestão. A entidade SME pode modificar a MIB da camada MAC e física através dos interfaces de serviço MLME e PLME. Algumas modificações na camada MAC podem requerer as modificações correspondentes na camada física, portanto, existe um *interface* adicional entre o MLME e o PLME.

[1]É uma base de dados virtual que contém objectos com informação que podem ser pesquisados e ao mesmo tempo provocar que certas acções ocorram.

Operações de gestão

Existe um conjunto de operações de gestão importantes do 802.11. Algumas das mais comuns usadas são:

Scanning - é o processo responsável pela descoberta de redes sem fios realizado antes do utilizador se juntar a uma delas. Este usa um conjunto de parâmetros que podem ser especificados pelo utilizador apesar de muitas implementações usarem parâmetros predefinidos no controlador. Este processo de descoberta pode ser passivo ou activo. Quando é passivo a estação espera pelas tramas de *Beacon*. No processo activo, a estação usa a trama *Probe Request* para solicitar respostas e encontrar as redes. Quando a descoberta é concluída, é criado um relatório com as redes encontradas e os seus parâmetros. A estação pode depois escolher uma dessas redes para se juntar com a intervenção do utilizador, ou não, caso preencha todos os requisitos necessários.

Autenticação - processo em que uma estação se autêntica perante a rede sem fios. A autenticação pode ser aberta ou protegida. Quando é aberta a autenticação pode ser feita por qualquer estação livremente. Quando é protegida normalmente existe uma chave guardada na estação a que o utilizador teve acesso e que foi configurada na estação. Essa chave é usada depois para a autenticação através de um processo que envolve técnicas e algoritmos criptográficos.

Associação - processo em que uma estação se associa a uma Infrastructure BSS depois de se ter autenticado com a mesma, para assim ganhar acesso completo. Uma estação não pode estar associada com mais do que um AP. O processo de associação parte sempre da estação. Uma estação também pode efectuar a reassociação, que consiste normalmente em mover a associação de um AP antigo para um novo dentro da mesma ESA. Este processo também parte da estação que faz a monitorização da qualidade do sinal dos APs da ESA e escolhe o que achar adequado.

Poupança de Energia - as estações sem fios normalmente são alimentadas a baterias e o processo de poupança de energia permite a estas ter uma melhor autonomia. Numa Infrastructure BSS, esta operação para os APs tem duas tarefas, determinar se uma trama é enviada ou armazenada e depois anunciar periodicamente quais estações que têm tramas armazenadas. As estações acordam

periodicamente para ouvir esse anúncio e eventualmente pedir para o AP enviar as tramas armazenadas. Uma estação pode mudar o modo de conservação de energia a qualquer altura e passar a transmitir com potência máxima, por exemplo, quando se liga um portátil à ficha eléctrica. Numa IBSS a gestão de energia não é tão eficiente como em Infrastructure BSS. Mais carga é posta nos remetentes para garantir que o receptor está activo, o receptor também não pode permanecer tanto tempo adormecido. Não existe um coordenador central, o que obriga a uma coordenação distribuída.

Sincronização Temporal - as redes 802.11 dependem muito da sincronização temporal, assim como outras redes. É especialmente importante nas redes que usam FHSS porque as estações têm que estar coordenadas ao mudar de canal e também nos mecanismos de reserva do meio que usam funções temporais. Numa Infrastructure BSS os APs mantém as estações associadas sincronizadas através do *Beacon*. O *Beacon* inclui a função de sincronização temporal (TSF, Timer Synchronization Function), que é colocada pelo AP imediatamente antes de ser enviada. As estações mantém depois localmente o TSF, assim mantendo a sincronização mesmo na falha de *Beacons*. Numa IBSS o processo do *Beacon* é distribuído. O TSF é mantido a partir da geração do *Beacon*, que deve ser transmitido exactamente no TBTT (Target Beacon Transmission Time), que é o guia neste tipo de redes.

2.2.10 Camada física

O segundo componente principal da arquitectura do 802.11 é a camada física. Esta divide-se em duas sub-camadas, PLCP e PMD, como mostra a Figura 2.19.

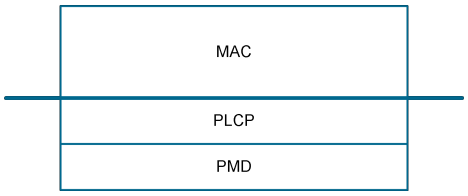


Figura 2.19: Arquitectura da camada física do 802.11.

O PLCP faz a troca das tramas com a camada MAC e acrescenta o seu cabeçalho antes da transmissão para o meio. Um preâmbulo pode ser requerido para sincronizar

as transmissões, no entanto depende da modulação usada. O PMD é responsável por transmitir os bits que recebe do PLCP para o ar usando a antena.

A camada física incorpora também um mecanismo que faz a avaliação do meio (CCA, Clear Channel Assessment) para indicar à camada MAC quando um sinal é detectado.

A norma lançada inicialmente contava com três camadas físicas. Com a evolução da norma, nos dias de hoje, já são sete camadas físicas disponíveis. Cada uma das normas suporta mais do que um tipo de modulação que varia conforme o débito binário pretendido.

- IR PHY - baseada em infra-vermelhos - 802.11
- FHSS PHY (2.4 GHz) - Frequency Hopping Spread Spectrum - 802.11
- DSSS PHY (2.4 GHz) - Direct Sequence Spread Spectrum - 802.11
- OFDM PHY (5 GHz) - Ortogonal Frequency Division Multiplexing - 802.11a
- HR/DSSS PHY (2.4 GHz) - High Rate DSSS - 802.11b
- ERP PHY (2.4 GHz) - Extended Rate - 802.11g
- HT PHY (2.4 GHz e 5 GHz) - High Troughput - 802.11n

A camada física baseada em infra-vermelhos nunca foi muito usada. As restantes camadas físicas são quase todas retro-compatíveis entre si quando funcionam à mesma frequência.

Alocação de espectro

O 802.11 usa as bandas ISM que não carecem de licença para utilização na maior parte dos países, desde que cumpridas as regras dos sinais transmitidos como níveis de potência, tipos de antenas, etc.

As gamas de frequências variam ligeiramente em alguns países mas no geral são usados as gamas 2.400-2.4835 GHz e 5.725-5.850 GHz.

2.400-2.4835 GHz - esta banda é dividida em 13 canais espaçados por 5 MHz, com o primeiro canal centrado nos 2.412 GHz e o último nos 2.472 GHz. O

802.11b necessita de 22 MHz que resulta em apenas 3 canais não sobrepostos. O 802.11g/n usa 20 MHz, o que resulta em 4 canais não sobrepostos. O 802.11n pode usar também canais de 40MHz, sendo apenas possível 2 canais não sobrepostos.

5.725-5.850 GHz - na banda dos 5 GHz existem mais canais, no entanto a variação de país para país é maior, nesta banda em Portugal existem 4 canais de 20 MHz e 2 de 40 MHz não sobrepostos. Estão em andamento, na União Europeia, processos para libertar mais espectro dentro dos 5 GHz, que vão poder ser usados para as redes 802.11.

Espalhamento espectral

As camadas físicas do 802.11 usam a tecnologia de espalhamento espectral (*Spread Spectrum*). Consiste numa técnica de modulação em que a largura de banda usada na transmissão de um sinal é maior que a banda mínima necessária para a transmissão da informação. O sinal a ser transmitido é espalhado no domínio das frequências antes da transmissão recorrendo a um código independente dos dados a transmitir. O mesmo código é usado na recepção, correlacionando-o com o sinal recebido e assim recuperando a informação.

Esta técnica traz algumas vantagens, com destaque para a imunidade contra interferências, distorções e desvanecimento de banda estreita e pode ser partilhada na mesma banda de frequência com baixa interferência.

Nas diferentes camadas físicas do 802.11 podem-se encontrar vários tipos de espalhamento espectral:

FHSS - consiste em saltos de uma frequência para outra num padrão aleatório, transmitindo durante um curto espaço de tempo em cada sub-canal.

DSSS - a potência do sinal é espalhada por uma largura de banda ainda mais larga usando uma função matemática para codificar o sinal.

OFDM - o canal é dividido em diversos sub-canais. Em cada sub-canal é codificado uma porção do sinal em paralelo. Esta técnica é parecida com o DMT (Discrete Multi-Tone) usado no DSL (Digital Subscriber Line).

2.3 Ferramentas de monitorização de redes WiFi

Inicialmente foi realizado um levantamento de algumas ferramentas para monitorização de redes *WiFi*. A procura pelas ferramentas não se restringiu a um SO (Sistema Operativo) específico, *software* não comercial ou *software* open-source.

Após o levantamento das ferramentas, estas foram testadas de forma a efectuar um levantamento das funcionalidades e de alguma semelhança com a solução que se pretende para este projecto.

inSSIDer 2

Esta ferramenta permite obter uma lista com as características das redes *WiFi* que estão presentes na vizinhança e seguir a potência de sinal de cada uma ao longo do tempo. Com o uso auxiliar de um dispositivo de posicionamento (GPS, Global Positioning System) permite associar coordenadas geográficas a cada um dos APs detectados. Permite também exportar todos os dados recolhidos para ficheiro. Esta ferramenta é open-source e não comercial e apenas se encontra disponível para os SOs Windows.

Aircrack-ng

Esta ferramenta é usada para quebrar chaves de redes *WiFi*. Permite a obtenção de APs e a captura de tráfego de forma a obter a chave de uma rede. A captura de tráfego está limitada a um grupo pequeno de placas *WiFi*. Esta ferramenta é open-source e não comercial e encontra-se disponível para os SOs Linux e Windows.

Commview for *WiFi*

Esta ferramenta permite a captura de tráfego ao nível da camada MAC do 802.11 num grupo pequeno de placas *WiFi* quando é instalado o controlador proprietário que acompanha a ferramenta. Também permite a análise do tráfego capturado e dos protocolos das diferentes camadas. Existe uma funcionalidade que faz o que é pretendido para a solução, no entanto de uma forma diferente que não se pretende. Este muda o modo da placa de rede sem fios e salta de canal em canal à procura de estações e APs, o que obriga a desligar de qualquer rede sem fios durante o processo. Esta ferramenta é comercial e está apenas disponível para os SOs Windows.

WildPackets OmniPeek

Esta ferramenta permite a captura e análise de tráfego. É idêntica ao Comm-View for *WiFi*. Nos testes efectuados só foi possível descer a captura ao nível da camada Ethernet. Esta ferramenta é comercial e está apenas disponível para os SOs Windows.

***WiFi* Scanner**

Uma ferramenta simples que permite obter uma lista com as características das redes *WiFi* presentes na vizinhança através do comando *netsh* disponível na família de SOs Windows Vista e 7. A ferramenta é open-source e não comercial.

Packetyzer

É outra ferramenta que permite a captura e análise de tráfego. Permite seleccionar o motor usado na captura. A novidade que traz é permitir construir diagramas temporais da troca de tramas. Esta ferramenta é livre e apenas disponível para os SOs Windows.

***WiFi* Locator**

Esta ferramenta permite obter uma lista com as características das redes *WiFi* presentes na vizinhança. Usa uma API (Application Programming Interface) open-source conhecida por ManagedWifi para a obtenção dessa lista. Como novidade, permite localizar a posição geográfica da rede *WiFi* através de uma base de dados de redes *WiFi* da Google. A ferramenta é livre e apenas disponível para os SOs Windows.

Wireless Network Watcher

Esta ferramenta permite a obtenção de uma lista com algumas características dos equipamentos ligados na rede. Com esta solução pretende-se uma lista das estações *WiFi* vizinhas à semelhança desta ferramenta, no entanto, sem os computadores ligados por rede fixa e os equipamentos de rede também ligados por rede fixa. Esta ferramenta é livre e apenas disponível para os SOs Windows.

Kismet

Esta ferramenta permite a captura de tráfego ao nível da camada MAC do 802.11 assim como detectar intrusões. No entanto é necessário um controlador da placa *WiFi* compatível. A ferramenta é livre e está disponível para os SOs Windows e Linux.

TekWiFi

Esta ferramenta permite obter uma lista com as características das redes *WiFi* presentes na vizinhança. Como novidade, permite efectuar testes e diagnósticos à rede ao qual a estação se encontra associada. A ferramenta é livre e está disponível apenas para os SOs Windows.

WirelessNetView

Esta ferramenta permite obter uma lista com as características das redes *WiFi* presentes na vizinhança. Não traz nada de novo. A ferramenta é livre e está disponível apenas para os SOs Windows.

Nas ferramentas analisadas apenas foi encontrada uma que faz o que é pretendido para esta solução, no entanto esta requer um controlador proprietário instalado na placa de redes sem fios. Este controlador proprietário apenas existe para um número limitado de placas de rede sem fios. A ferramenta ao procurar por estações na vizinhança, desliga-se de qualquer rede *WiFi* a que estejamos ligado, pois põe a placa de rede sem fios em modo de monitorização (*Monitor Mode*). Isto não é desejável na solução que queremos implementar.

2.4 Projectos relacionados com redes WiFi

Foram analisados alguns projectos que utilizaram as redes *WiFi*. Estes projectos ilustram sobretudo a utilização de redes *WiFi* na caracterização dos espaços. A fonte de dados principal nestes projectos é a infra-estrutura de rede.

Um dos primeiros projectos encontrados tinha como finalidade criar um mapa do uso da rede *WiFi* do campus do MIT (Massachusetts Institute of Technology) em tempo real[6] de forma a perceber melhor os padrões diários da vida académica. Neste projecto foram construídos mapas do uso da rede *WiFi* pelos utilizadores por um período de 24 horas. Nesses mapas foi possível observar certos padrões que se verificavam, como os picos de utilizadores em determinados locais do campus a determinadas horas.

Um outro projecto, consistia num estudo da rede *WiFi* do Google, em Mountain View na Califórnia[7]. Nesta rede é usado um servidor que faz a gestão de contas, que recebe em intervalos de 15 minutos a informação de todos os nós, com os clientes que fizeram a associação ou abandonaram a rede. Foi realizado uma captura de

tráfego na camada 3 que lhes permitiu identificar os utilizadores da rede através do endereço MAC. Durante um período de 28 dias em intervalos de 15 minutos foram feitas medições do tempo médio que os utilizadores permaneciam activos na rede. Realizaram também algumas classificações como o uso da rede por tipo de dispositivos, bytes transferidos, velocidade de transmissão por tipo de dispositivos, número de ligações por classe de aplicação, etc.

O próximo projecto é a análise a mais um campus, The Dartmouth College[8]. Foi efectuada a recolha de informação dos *logs* e da captura de tráfego nos *interfaces* Ethernet dos APs. Isto permitiu saber o local, número de cliente e a quantidade de informação transferida. O período de análise foi de 17 semanas. A captura de tráfego decorreu em 18 locais e permitiu capturar a maior parte do tráfego que passava pelos APs incluindo o tráfego entre APs. Com o tráfego capturado e o auxílio do utilitário p0f2[1] foi identificado os tipos de dispositivos associados aos respectivos endereços MAC. As estatísticas obtidas do uso da rede no campus são idênticas às observadas já em outros artigos, com a adição de permitir observar a mobilidade dos utilizadores na rede.

A finalidade da solução deste projecto é permitir a cada estação recolher informação das estações presentes na sua vizinhança. Mas contrariamente a estes projectos, pretende-se que cada estação faça a recolha de informação de forma individual sem recorrer ao uso da infra-estrutura de rede. Cada estação apenas deve depender de si própria para completar a tarefa.

[1]Identifica nos pacotes assinaturas características da implementação dos protocolos nos diferentes Sistemas Operativos.

Capítulo 3

Exploração de uma Solução

Neste capítulo vai ser descrito todo o processo de exploração de uma solução para o problema da descoberta de estações próximas, com a descrição de todos os passos efectuados e os problemas encontrados, tendo sempre em conta as restrições impostas.

3.1 Abordagem

A abordagem inicial neste projecto para exploração de uma solução, já depois de um estudo da norma IEEE 802.11 que incidiu principalmente na sua camada MAC e tramas MAC, bem como a análise de alguns trabalhos e ferramentas relacionados, foi realizar um levantamento dos requisitos e restrições essenciais a que a solução a desenvolver estava sujeita.

Alguns dos requisitos e restrições são impostos porque esta solução neste projecto faz parte de um outro projecto mais alargado, o Epi[1], em que a solução encontrada para este projecto será parte integrante do Epi, como um módulo de *software*.

O Epi[9] é uma aplicação desenvolvida para os SOs Windows onde as estações com *interface WiFi* armazenam informação do ambiente rádio que rodeia os utilizadores enquanto fornece funcionalidades de rede social, baseada na difusão epidémica de mensagens entre utilizadores próximos.

Os requisitos e restrições que se podem identificar nesta fase são os seguintes:

- Funcionar nos SOs Windows, versões XP SP (Service Pack) 3, Vista e 7. Assim, como este módulo será integrado na aplicação Epi, terá que ser desenvolvido para os mesmos SOs.

[1]Difusão Epidémica de Mensagens em Hotspots *WiFi*, `http://epi.dsi.uminho.pt/`.

- Ser desenvolvido usando a linguagem C#.net. O Epi foi desenvolvido em C#.net e apesar de isso não obrigar a desenvolver o módulo na mesma linguagem, decidiu-se usar a mesma. Esta é uma linguagem de alto nível muito actual e assim mantém-se o projecto todo na mesma linguagem.

- Actuar de uma forma passiva não gerando muito tráfego extra na rede. Numa situação em que a rede se encontre perto ou mesmo saturada, esse tráfego extra só vai agravar a situação sem acrescentar valor ao utilizador. Por isso pretende-se apenas usar o tráfego que já exista a circular na rede.

- Não depender de nenhum *hardware* específico como Airpcap[1]. O Epi é uma aplicação para os utilizadores usarem nas máquinas comuns, com uma variedade muito grande de *hardware*. Logo para este módulo não se pode impingir o funcionamento apenas em *hardware* específico.

- Não depender de nenhum controlador específico para a placa de rede sem fios, como faz o CommView[2]. Não é convidativo impingir o utilizador a mudar o controlador da sua placa de rede sem fios de forma a este módulo funcionar. Isto também levaria à necessidade de estudar cada placa de rede sem fios, o que é impensável devido a diversidade de placas existentes. Acrescentar outro tipo de controladores que não sejam específicos de algum *hardware*, como controladores de protocolos ou filtros de rede NDIS (Network Driver Interface Specification)[3], já é um caso aceitável.

- Ser transparente, ou seja, não interferir na ligação *WiFi*, permitindo às estações a transferência normal de dados e acesso à Internet quando associada a um AP, bem como a associação a um novo AP.

Depois deste levantamento de requisitos e restrições, podem-se excluir logo alguns aspectos relacionados com o modo de funcionamento de uma placa de rede sem fios. O *Monitor Mode* que inicialmente parecia útil para a solução viola alguns dos requisitos e restrições:

[1]Conjunto *hardware+software* para captura de tráfego num canal *WiFi*, `http://www.cacetech.com/products/airpcap.html`.

[2]Usa um controlador proprietário, para algumas placas de redes sem fios específicas.

[3]API para o *interface* das placas de rede usado principalmente no SOs Windows

- Não podemos transmitir enquanto a placa de rede sem fios permanecer neste modo, logo quebra a ligação de dados activa que possa existir entre uma estação e um AP e não permite a associação a um novo AP.

- Só permite a monitorização de um canal *WiFi* de cada vez, no entanto seria possível fazer *Channel Hopping*[1] para contornar isso, permanecendo uma porção de tempo em cada canal.

- Este modo pode não ser suportado em todas as placas de rede sem fios, sendo que depende do controlador e da implementação do mesmo.

Assim, o *Monitor Mode* fica para já excluído como parte da solução pretendida para este projecto.

Os modos que permitem a ligação a uma Infrastructure BSS e IBSS nesta altura parecem os mais indicados para a solução do problema, uma vez que não violam nenhuma das restrições e requisitos que foram impostos. A diferença entre os dois é que o primeiro é usado quando a comunicação entre as estações na mesma BSS é efectuada com recurso a um AP por onde passa toda a comunicação, criando assim uma Infrastructure BSS, que poderá vir a tornar-se uma ESS quando várias Infrastructure BSS se interligam entre si e formam um DiS. No caso do segundo modo as estações fazem a comunicação directamente entre si, formando uma IBSS, que será uma rede pequena com tempo de vida muito limitado e criada para um propósito específico.

Foi também realizada uma exploração de APIs que ajudem a encontrar uma solução tendo sempre em conta que estas não violem as restrições e requisitos impostos. Foram encontradas diversas APIs, no entanto nenhuma que resolva de forma directa o problema.

As APIs encontradas foram as seguintes:

***Native Wifi*[10]** é a *stack* nativa dos SOs Windows, que a partir da versão XP SP 2 dá suporte às redes *WiFi*. Permite obter a lista e as características das redes sem fios que são disponibilizas pelos APs na vizinhança. Permite ligar e desligar das redes sem fios de forma manual ou automática através de perfis com as configurações, sendo estes guardados em ficheiro na forma de um documento em XML (Extensible Markup Language). Permite também expor elementos

[1] Consiste em saltar de canal em canal.

lógicos através do ACM (Auto Configuration Module) para permitir extensões, permitindo aos programadores incorporar novas funcionalidades adicionais sem afectar a framework original.

***Winpcap*[11]** é a versão para Windows da biblioteca libpcap. Inclui um controlador para permitir a captura de tráfego na rede, usando o interface NDIS para ler os pacotes directamente da placa de rede. Inclui também uma API para comunicar com esse controlador, que é usada por algumas ferramentas conhecidas, tais como o Wireshark.

***Sharpcap*[12]** é uma biblioteca para a plataforma C#.net que permite a interacção com o controlador libpcap/winpcap. Inclui também a biblioteca Packet.Net[13], um *parser* de tramas Ethernet, IP, TCP (Transmission Control Protocol), UDP (User Datagram Protocol), etc. Nas versões mais antigas do Sharpcap, ambas as bibliotecas eram desenvolvidas dentro do mesmo projecto, no entanto agora são projectos em separado.

***Managed WiFi*[14]** é um conjunto de bibliotecas com classes para C#.net que permite controlar as placas de redes sem fios, instaladas no Windows, usando a API Native WiFi.

Com estas APIs, de forma directa, apenas é possível obter os APs na vizinhança, o que não resolve o nosso problema de detectar as estações presentes na vizinhança.

Nesta fase surge a ideia de que a solução poderá passar pela análise de tráfego que circula numa rede e que é descrita na secção 3.2.

3.2 Captura e análise de tráfego

Numa rede TCP/IP, existe uma quantidade grande de protocolos que são transmitido em *broadcast* e *multicast*, desta forma uma qualquer estação ligada na rede, seja com uma ligação física ou sem fios, irá receber este tráfego.

Estes protocolos são normalmente respostas a erros em pacotes IP, diagnóstico e encaminhamento, entre outros. Assim, de forma a identificar e perceber melhor estes protocolos, realizaram-se algumas experiências de captura de tráfego.

Para esta tarefa foi realizada uma captura de tráfego em dois locais diferentes de forma a obter-se padrões de tráfego diferentes:

- uma rede grande constituída por vários APs que pertencem à mesma ESS.
- uma pequena rede pública constituída por um único AP.

Estas recolhas decorrem durante aproximadamente 1 hora durante um dia da semana à tarde.

Um dos locais foi a biblioteca da Universidade do Minho no pólo de Azurém, onde normalmente se concentra um grande número de estações que se ligam à rede *WiFi* "eduroam", sendo esta uma rede grande, alargada ao nível de todo o campus do pólo, usando diversos APs que se interligam através de uma rede de *backbone* Ethernet.

O outro local foi o Café Jardim situado em Guimarães, onde existe uma rede pequena de acesso público, constituída apenas por um único AP.

As capturas de tráfego foram realizadas usando a ferramenta Wireshark[1] em modo promíscuo[2]. As capturas decorreram no limite superior da camada 2, nível Ethernet, que é o nível mais baixo que foi possível descer para realizar a captura de tráfego com esta ferramenta nos SOs Windows.

Todas as tramas com origem e destino à estação onde foi efectuada a captura foram eliminadas através da aplicação de um filtro no Wireshark, para descartar qualquer pacote com endereço MAC de destino ou origem igual ao da estação. Desta forma é garantida também a eliminação de tráfego TCP de alguma sessão de comunicação que possa ter existido com esta estação devido a alguma aplicação a correr na estação durante a realização da captura. Este tráfego não tem interesse, uma vez que apenas se pretende realizar o estudo sobre o tráfego gerado pelas outras estações que se encontravam ligadas na rede.

3.2.1 Tráfego Biblioteca UM - Azurém

A captura de tráfego neste local foi efectuada no dia 4 de Novembro de 2010 com início às 17:02:35, tendo a duração de 01:00:49. Foram capturados um total de 243392 pacotes a que correspondem 64113947 bytes.

Na Tabela 3.1 encontram-se algumas estatísticas recolhidas do tráfego ao nível da camada de rede IP onde se distinguem as duas versões do protocolo IP, o número de pacotes e o tamanho em Bytes, bem como as respectivas percentagens.

[1] Usa o controlador Winpcap.

[2] Captura de todo o tráfego naquele segmento de rede, e não apenas endereçado ao próprio.

Protocolo	Pacotes(%)	Bytes(%)
IPv6	145496 (59,78%)	39770499 (66,04%)
IPv4	97896 (40,22%)	20449151 (33,96%)

Tabela 3.1: Estatísticas ao nível da camada de rede IP, Azurém

Observa-se que o maior número de pacotes recolhidos neste local são já IPv6, o tamanho dos pacotes IPv6 é também maior, uma vez que a fatia no número de bytes aumenta ainda mais relativamente ao IPv4.

Subindo para o nível acima, na camada de transporte, observa-se na Tabela 3.2 os protocolos encontrados da mesma forma que na Tabela 3.1.

Protocolo (Versão)	Pacotes(%)	Bytes(%)
ICMPv6(IPv6)	66474 (27,31%)	5909412 (9,81%)
UDP (IPv6)	79022 (32,47%)	33861087 (56,23%)
UDP (IPv4)	97510 (40,06%)	20425670 (33,92%)
IGMP (IPv4)	363 (0,15%)	21780 (0,04%)
ICMP (IPv4)	23 (0,01%)	1702 (0,00%)

Tabela 3.2: Estatísticas ao nível da camada de transporte, Azurém

Foram encontrados protocolos de controlo de erros e diagnóstico, bem como de controlo de acesso a grupos *multicast*, quer no IPv4 como no IPv6, no entanto a fatia maior de pacotes é UDP[1]. Como seria de esperar, não foi capturado nenhum pacote TCP, uma vez que foi aplicado o filtro para eliminar os pacotes com endereço MAC igual a máquina de captura. Os restantes protocolos neste nível são descritos na secção 3.2.5.

UDP é um protocolo simples da camada de transporte em que um pacote é enviado para o destino sem controlo de fluxo ou garantias de entrega, contrariamente ao TCP. Este é usado tipicamente para o envio de tráfego em *broadcast* e *multicast* e encapsula já protocolos da camada de aplicação. Na Tabela 3.3 pode-se observar os protocolos da camada de aplicação que foram encontrados no tráfego capturado neste local.

Dos protocolos identificados ao nível da camada de aplicação, Dados, HTTP (Hypertext Transfer Protocol) e DNS (Domain Name System) representam apenas o formato em que a informação contida dentro do pacote é transportada e não um protocolo específico. Assim, Dados para pacotes com dados genéricos que não são

[1]Apesar dos protocolos ICMP e IGMP usarem UDP como transporte, são considerados distintos do UDP uma vez que não transportam dados para as camadas superiores.

Protocolo (Versão)	Pacotes(%)	Bytes(%)
Dados(IPv6)	27252 (11,20%)	21659239 (35,97%)
HTTP (IPv6)	20529 (8,43%)	9351636 (15,53%)
DNS (IPv6)	31241 (12,84%)	2850213 (4,73%)
HTTP (IPv4)	26877 (11,04%)	6341858 (10,53%)
Bootp / DHCP (IPv4)	1111 (0,46%)	383896 (0,64%)
Dados (IPv4)	9686 (3,98%)	7443354 (12,36%)
Dropbox LAN sync Discovery Protocol (IPv4)	6396 (2,63%)	1191667 (1,98%)
NetBIOS Datagram Service (IPv4)	3793 (1,56%)	898335 (1,49%)
NetBIOS Name Service (IPv4)	19629 (8,06%)	1849248 (3,07%)
DNS (IPv4)	30007 (12,33%)	2315367 (3,84%)
TiVoConnect Discovery Protocol (IPv4)	10 (0,00%)	1740 (0,00%)
CUPS Browsing Protocol (IPv4)	1 (0,00%)	206 (0,00%)

Tabela 3.3: Estatísticas ao nível da camada de Aplicação, Azurém

identificados como outro tipo, HTTP para os pacotes em que a informação é transportada no formato HTTP e DNS em que a informação é transportada no formato típico de um pacote DNS.

Os restantes protocolos são bem conhecidos, no entanto alguns apresentam uma percentagem muito baixa de pacotes, e por serem casos isolados de aplicações a correr, não terão interesse para o pretendido.

Os pacotes do protocolo NBDS (NetBIOS Datagram Service) contém outros níveis protocolares encapsulados que fazem parte apenas da própria aplicação. Nos níveis superiores são encapsulados os protocolos, SMB (Server Message Block Protocol) que por sua vez encapsula o protocolo *SMS MailSlot Protocol* e por fim este encapsula o protocolo *Microsoft Windows Browser Protocol*.

Na Tabela 3.4 encontra-se detalhado os protocolos encontrados dentro dos pacotes que transportam informação de dados genérica. Os protocolos são identificados pelo seu nome na aplicação Wireshark ou pelo nome da aplicação da qual fazem parte. Neste último caso o protocolo foi identificado pelo conteúdo do pacote, uma vez que a porta usada na camada de transporte não está registada e associada a um protocolo pelo IANA (Internet Assigned Numbers Authority)[1]. É difícil identificar o protocolo apenas pelo conteúdo dos dados no pacote e este pode ser desconhecido ou proprietário. Na tabela é identificado a respectiva porta, o número de pacotes encontrados e a versão do protocolo IP.

O número de pacotes capturados na grande maioria dos protocolos é muito baixo,

[1]Autoridade responsável pela atribuição do endereçamento IP, recursos de protocolos da Internet e DNS Root `http://www.iana.org/`.

Protocolo(Versão)	Porta	NºPacotes
Gnutella-rtr(IPv4)	6347	24
ws-discovery(IPv4)	3702	9153
EPI(IPv4)	51127	74
desconhecido(IPv4)	6646	1
flukeserver(IPv4)	2359	3
futrix(IPv4)	2358	4
groove-dpp(IPv4)	1211	135
gxtelmd(IPv4)	2356	4
hlserver(IPv4)	3047	17
nati-logos(IPv4)	2347	2
nextorindltd(IPv4)	2360	2
plysrv-https(IPv4)	6771	117
psbserver(IPv4)	2350	4
psdbserver(IPv4)	2355	4
pslserver(IPv4)	2352	3
psprserver(IPv4)	2354	5
pspserver(IPv4)	2353	4
psrserver(IPv4)	2351	2
sentinelsrm(IPv4)	1947	122
tcpnethaspsrv(IPv4)	475	3
unihub-server(IPv4)	2357	3
WS-Discovery(IPv6)	3702	27252

Tabela 3.4: Protocolos em Dados UDP, Azurém

logo não são considerados para o estudo dos protocolos. Desta forma, apenas se considera o protocolo WS-Discovery (Web Services Dynamic Discovery) para estudo na secção 3.2.5, uma vez que o número pacotes é considerável, tanto em IPv4 como em IPv6.

Nesta lista foram encontrados alguns pacotes do protocolo usado pela aplicação Epi, que foram identificados pelo conteúdo dos pacotes. Este projecto será mais tarde integrado nessa aplicação.

Na Tabela 3.5, da mesma forma que foi feito anteriormente, são detalhados os protocolos encontrados dentro dos pacotes que transportam informação no formato genérico HTTP e, na Tabela 3.6, para os protocolos que transportam informação do tipo genérico DNS.

Nos pacotes HTTP foi encontrado apenas o protocolo SSDP (Simple Service Discovery Protocol) com um número de pacotes considerável relativamente ao total de

Protocolo(Versão)	Porta	NºPacotes
SSDP(IPv4)	1900	26877
SSDP(IPv6)	1900	20529

Tabela 3.5: Protocolos em HTTP UDP, Azurém

pacotes capturados, tanto em IPv4 como em IPv6, o que o torna interessante para o estudo descrito na secção 3.2.5.

Protocolo(Versão)	Porta	NºPacotes
LLMNR(IPv4)	5355	28504
MDNS(IPv4)	5353	1503
LLMNR(IPv6)	5355	31130
MDNS(IPv6)	5353	111

Tabela 3.6: Protocolos em DNS UDP, Azurém

Nos pacotes DNS foram encontrados dois protocolos distintos em ambas as versões do IP. O LLMNR (Link-Local Multicast Name Resolution) foi encontrado num número de pacotes considerável e o MDNS (Multicast Domain Name System) foi encontrado num número de pacotes bastante mais reduzido mas suficiente para ser alvo de estudo. Ambos os protocolos serão alvo de estudo na secção 3.2.5.

3.2.2 Tráfego Café Jardim - Guimarães

A captura de tráfego neste local foi efectuada no dia 29 de Outubro de 2010 com início às 15:06:01, tendo duração de 00:49:11. Foram capturados um total de 1364 pacotes a que correspondem 295260 bytes. O número de pacotes é bastante inferior ao capturado no caso anterior, o que é de esperar uma vez que se trata de uma rede de pequenas dimensões em que o número de utilizadores visível quando foi efectuado a captura não devia ultrapassar os 6.

Na Tabela 3.7, encontram-se algumas estatísticas recolhidas do tráfego ao nível IP. Distingue-se as duas versões do protocolo IP e o ARP (Address Resolution Protocol), o número de pacotes e o tamanho em Bytes, bem como as respectivas percentagens.

Observa-se neste caso um número maior de pacotes em IPv4, contrariamente ao caso anterior onde a fatia de pacotes IPv6 é muito reduzida. Foram encontrados também pacotes ARP, em número reduzido, mas que é interessante relatar uma vez que no caso anterior não se verificou.

Protocolo	Pacotes(%)	Bytes(%)
IPv4	1180 (86,51%)	257677 (94,24%)
IPv6	92 (6,74%)	11763 (4,30%)
ARP	92 (6,74%)	3972 (1,45%)

Tabela 3.7: Estatísticas ao nível da camada de rede IP, Café Jardim

Subindo para um nível acima, na camada de transporte, observa-se na Tabela 3.8 os protocolos encontrados, na mesma forma que na Tabela 3.2.

Protocolo (Versão)	Pacotes(%)	Bytes(%)
UDP (IPv4)	1132 (82,99%)	254867 (93,22%)
UDP (IPv6)	92 (6,74%)	11763 (4,30%)
IGMP (IPv4)	48 (3,52%)	2810 (1,03%)

Tabela 3.8: Estatísticas ao nível da camada de transporte, Café Jardim

Ao nível da camada de transporte foram encontrados pacotes de controlo de acesso a grupos *multicast* em IPv4 e pacotes UDP em IPv4 e IPv6. A maior fatia de pacotes é UDP em IPv4, os restantes têm um número bastante reduzido.

Relativamente ao caso anterior e através dos protocolos encontrados nesta camada pode-se concluir que a maior parte dos equipamentos presentes na rede não tinham IPv6 activado. Não foram capturados pacotes ICMPv6 de controlo de erro e diagnóstico. Como seria de esperar também não foi encontrado nenhum pacote TCP.

Na Tabela 3.9 pode-se observar os protocolos encontrados ao nível da camada de aplicação.

Protocolo (Versão)	Pacotes(%)	Bytes(%)
Http (IPv6)	9 (0,66%)	4483 (1,64%)
DNS (IPv6)	83 (6,09%)	7280 (2,66%)
Http (IPv4)	432 (31,67%)	150382 (55,00%)
Bootp / DHCP (IPv4)	17 (1,25%)	5842 (2,14%)
Dados (IPv4)	1 (0,07%)	62 (0,02%)
Dropbox LAN sync Discovery Protocol (IPv4)	345 (25,29%)	58995 (21,58%)
NetBIOS Name Service (IPv4)	196 (14,37%)	18644 (6,82%)
NetBIOS Datagram Service (IPv4)	33 (2,42%)	8133 (2,97%)
DNS (IPv4)	108 (7,92%)	12809 (4,68%)

Tabela 3.9: Estatísticas ao nível da camada de aplicação, Café Jardim

Nos protocolos identificados ao nível da camada de aplicação, em Dados apenas foi detectado um único pacote, identificado pelo Wireshark como *nati-logos*. Como é

um caso isolado, com apenas um único pacote, não é considerado para estudo.

No caso dos pacotes DNS e HTTP é necessário uma análise mais detalhada. Os protocolos restantes são conhecidos do caso anterior e são alvo de estudo na Secção 3.2.5.

Os pacotes NBDS, assim como no caso anterior, contêm uma hierarquia protocolar da própria aplicação.

Nas Tabelas 3.10 e 3.11, pode-se observar os protocolos que foram identificados para os pacotes DNS e HTTP.

Protocolo(Versão)	**Porta**	**NºPacotes**
SSDP(IPv4)	1900	432
SSDP(IPv6)	1900	9

Tabela 3.10: Protocolos em HTTP UDP, Café Jardim

Protocolo(Versão)	**Porta**	**NºPacotes**
LLMNR(IPv4)	5355	74
MDNS(IPv4)	5353	34
LLMNR(IPv6)	5355	83

Tabela 3.11: Protocolos em DNS UDP, Café Jardim

Os protocolos identificados são os mesmos que no caso anterior. O número de pacotes capturados também é um número considerável, relativamente ao total de pacotes, logo estes protocolos permanecem alvo de estudo na Secção 3.2.5.

3.2.3 Aplicação para detecção das estações

Com a análise do tráfego conseguiu-se perceber os protocolos mais relevantes, no entanto é necessário mais informação sobre as estações que originaram o tráfego capturado. Para isso foi desenvolvida uma nova aplicação que usa o tráfego capturado para extrair a seguinte informação:

- Lista total com os endereços MAC de todas a estações.
- Lista de endereços MAC detectados por intervalo de tempo.
- Lista de endereços IP associados a cada endereço MAC.

O endereço MAC é o endereço físico de 48 bits único e estático de cada *interface* de rede portanto cada estação é identificada pelo mesmo. Consegue-se extrair os endereços MAC dos pacotes Ethernet capturados na camada de ligação, que é o nível mais baixo em que foi possível efectuar a captura de tráfego.

Cada pacote Ethernet é constituído por um endereço MAC de origem e outro de destino. O endereço que tem interesse neste caso é o de origem, que pertence às estações que geram o tráfego, as quais se pretende detectar. O endereço MAC de destino é mais provável ser um endereço *broadcast* ou *multicast*, uma vez que recebemos o tráfego da mesma forma que uma qualquer outra estação na rede.

Em cada pacote capturado durante uma captura de tráfego, o Wireshark acrescenta *Meta* informação que aparece antes do nível da camada de ligação Ethernet. Nesta informação existe o *timestamp* em que chegou o pacote, que é usado para a informação temporal de cada endereço MAC e permite também separar a detecção dos endereços MAC por intervalos de tempo.

Para cada endereço MAC, que representa uma estação detectada, foi também associada uma lista de endereços IP. Um *interface* de rede pode ter associados um ou mais endereços IP, por exemplo um endereço IPv4 e outro endereço IPv6. Um *interface* de rede com IPv6 activado é normal ter vários endereços IPv6 associados. O endereço IP atribuído a um *interface* de rede também pode mudar ao longo do tempo, por exemplo, quando uma estação perde ligação de rede e se volta a ligar ou quando o tempo do aluguer de um endereço IP expira.

Adicionalmente, foi decidido incluir mais informação que estava presente no tráfego capturado e que podia vir a ser útil, complementando desta forma a informação essencial necessária para identificar cada estação.

- Protocolos detectados para cada endereço MAC em cada intervalo.
- Número de pacotes associado a cada protocolo em cada intervalo.
- Identificação do fabricante da placa associado a cada endereço MAC.
- Número de endereços MAC em cada intervalo.
- Número de pacotes capturados em cada intervalo.
- Número médio de endereços MAC por intervalo.
- Número médio de endereços IP por intervalo.

- Marcação se o IP detectado pertence à mesma subrede que a estação que fez a captura.

O número de estações que são detectadas por intervalo de tempo e a média de estações detectadas no conjuntos dos intervalos têm um relativa importância pois permite determinar o tempo de duração que a procura por estações na rede deverá durar. Quando integrado no Epi, o módulo será chamado para realizar recolhas periódicas das estações detectadas na vizinhança. Cada recolha será uma captura de tráfego com uma duração limitada de apenas alguns segundos.

Para implementar a aplicação com as funcionalidades necessárias para detectar as estações foi necessário recorrer a uma API. Das APIs estudadas na secção 3.1, foi determinado que a mais indicada para desenvolver a aplicação seria o conjunto de APIs Winpcap e Sharppcap porque:

- Winpcap é de instalação fácil e não é dependente do controlador da placa de rede sem fios.
- Winpcap permite a captura de pacotes Ethernet ao nível da camada de ligação e permite a filtragem de pacotes.
- É o controlador usado noutras soluções como o Wireshark, com desempenho e eficiência comprovada.
- A API original do Winpcap é para C++, no entanto o Sharppcap é um conjunto de bibliotecas para C#.net baseado nesta API.
- O Packet.Net permite o *parsing* de pacotes e está incluído junto com o Sharppcap.

Procedeu-se depois ao desenvolvimento da aplicação. Durante o seu desenvolvimento foram tomadas algumas decisões relativamente a algumas características que a aplicação deveria incluir.

Inicialmente a aplicação tinha o objectivo de ajudar na análise do tráfego capturado anteriormente, no entanto foi decidido acrescentar a captura e análise de tráfego em tempo real em qualquer placa de rede presente no computador. A escolha da placa de rede para efectuar a captura de tráfego seria feita pelo utilizador. A funcionalidade de abrir o tráfego que foi previamente capturado e realizar uma análise a partir de um ficheiro de tráfego do tipo pcap também está disponível.

Depois da análise do tráfego, quer a captura tenha sido efectuada em tempo real ou a partir do ficheiro, são gerados um conjunto de resultados. Estes são guardados em ficheiros HTML (HyperText Markup Language) para que possam ser facilmente consultados em qualquer navegador. Adicionalmente também é guardada alguma informação dos resultados em ficheiro de texto para permitir exportar facilmente para uma folha de cálculo. Os ficheiros de resultados que se obtém depois de uma análise são então os seguintes:

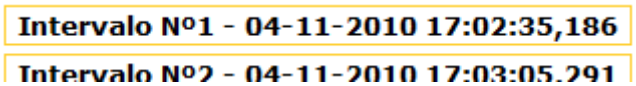
result.html contém a lista de estações detectadas por intervalo de tempo. No fim de cada intervalo é registado também o número de endereços MAC, endereços IP, pacotes e protocolos. A informação registada para cada estação em cada intervalo é a seguinte:

- *Timestamp*, quando foi detectada pela primeira vez no intervalo.
- Endereço MAC e fabricante da placa de rede.
- Lista de IPs associados ao endereço MAC, com marcação a **negrito** dos IPs que pertençam à mesma sub rede da máquina onde foi efectuada a captura.
- Lista de protocolos associados ao endereço MAC com contagem do número de pacotes usados por cada protocolo.

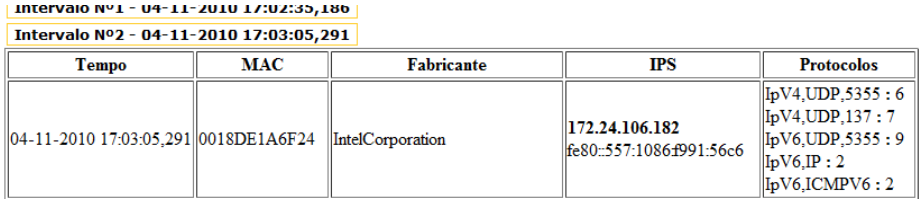
Os intervalos são numerados por ordem crescente, cada intervalo é identificado pelo seu número e o *timestamp* da primeira estação detectada dentro desse intervalo. No ficheiro de resultados cada intervalo aparece na forma *Expand/Collapse* para permitir fácil consulta, devido ao tamanho extenso que este poderá tomar. O ficheiro é escrito a cada intervalo permitindo a consulta durante uma captura. As estatísticas finais contém as médias de endereços MAC e IP, o número total de pacotes e o intervalo de captura e são escritas no ficheiro no final da captura. Nas Figuras 3.1 pode-se observar um exemplo de um ficheiro de resultados obtido no final de uma captura.

maclist.html apresenta a lista com todas a estações que foram encontradas na captura, para cada estação é apresentado o endereço MAC e o fabricante da placa de rede. O ficheiro só é escrito quando a captura termina.

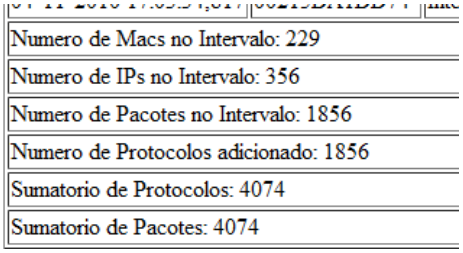
export.txt apresenta um conjunto de linhas de texto em que cada linha representa um intervalo, para cada intervalo é apresentado o número de estações, número



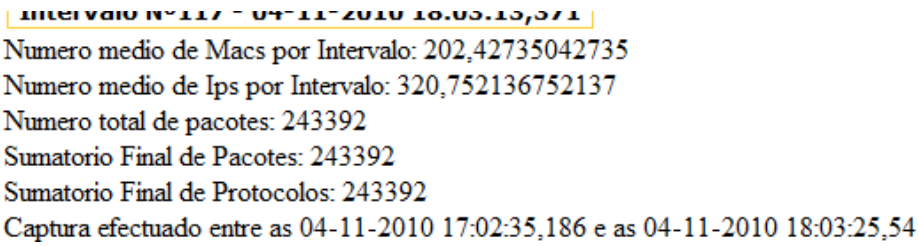
(a) Intervalos.



(b) Intervalo expandido.



(c) Dados estatísticos de um intervalo.



(d) Dados estatísticos finais.

Figura 3.1: Exemplo do ficheiro de resultados obtidos na captura efectuada na UM.

de IPs encontrados, o *timestamp* do início do intervalo e o número que identifica o intervalo. O formato deste ficheiro é em CSV (Comma-Separated Values)[1].

Toda a informação que é apresentada nos resultados é extraída processando pacote a pacote. Assim, ao receber um pacote é invocada uma função e o pacote é passado para dentro da mesma.

Dentro da função são retiradas as tramas pertencentes às respectivas camadas que se encontram encapsuladas no pacote. Retira-se primeiro a informação do *timestamp* dos *meta* dados. Depois na camada de ligação, encontra-se encapsulado a trama Ethernet onde se retira o respectivo endereço MAC de origem do cabeçalho. Na

[1]Cada valor separado por vírgula representa um valor de uma coluna numa tabela e as linhas, linhas da tabela.

próxima camada encontra-se encapsulado a trama IP que pertence à camada de rede IP, onde é retirado do cabeçalho os campos com o endereço IP de origem, que pode ser um endereço IPv4 de 4 bytes ou IPv6 de 16 bytes. Depois, encontra-se a camada de transporte onde são encapsuladas as tramas do tipo TCP ou UDP e é retirado do cabeçalho a porta de destino. Por fim segue-se a camada de aplicação onde são encapsulados os dados que são entregues às aplicações, no entanto nesta camada não é extraída informação.

Na Figura 3.2 está ilustrado como é feito o encapsulamento das tramas nas camadas de ligação, rede e transporte.

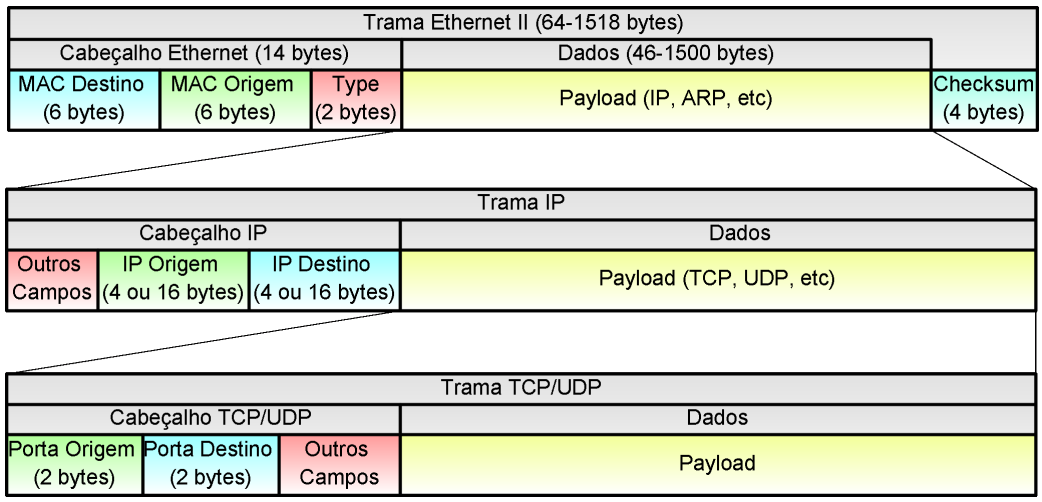


Figura 3.2: Encapsulamento do pacote Ethernet.

Com a aplicação pronta, o próximo passo foi o processamento dos ficheiros com o tráfego capturado que foram analisados nas secções 3.2.1 e 3.2.2 para diferentes tempos de intervalo.

3.2.4 Resultados da aplicação

Nos resultados obtidos observa-se que no Café Jardim são detectados 7 estações com endereços MAC distintos, o que é um número reduzido de estações mas já era de esperar uma vez que se trata de uma rede de pequenas dimensões. O número de estações detectadas à partida aparenta ser o mesmo número de estações presentes no espaço na data da captura, que eram as 6 que estavam visíveis.

Os resultados obtidos para Azurém foram diferentes, onde foram detectadas 632 endereços MAC distintos, um número consideravelmente grande que não reflecte o

número de estações na vizinhança que foi observado no local durante a captura. O número total de estações no local não devia ultrapassar as 50, contando com as estações novas que se iam ligando à rede no decorrer da captura.

Na Tabela 3.12 estão representadas as médias de endereços MAC e IP detectados por intervalo nos 2 locais, usando tempos de intervalo de 5, 10, 20 e 30 segundos.

Locais(Tempo Intervalo)	Média MACs	Média IPs	NºIntervalos
Azurém(30s)	202,42	320,75	117
Azurém(20s)	173,02	266,29	176
Azurém(10s)	126,85	185,76	346
Azurém(5s)	89,54	124,93	679
Café Jardim(30s)	1,90	2,01	89
Café Jardim(20s)	1,89	1,99	96
Café Jardim(10s)	1,49	1,58	144
Café Jardim(5s)	1,34	1,41	191

Tabela 3.12: Médias de endereços MAC e IP detectados por intervalo.

A diferença entre a média de endereços MAC e IP em Azurém mostra que existem muitas estações que têm mais do que um endereço IP. A rede em Azurém tem IPv6 activado e observa-se que já é usado por um grande número de estações, que têm um endereço IPv6 juntamente com um endereço IPv4.

No Café Jardim a diferença entre a média de endereços MAC e IP é muito pequena, o que leva a concluir que é apenas atribuído um único endereço IPv4 a cada estação.

No gráfico da Figura 3.3 está ilustrado o número de endereços MAC e IP detectados, distribuído ao longo de intervalos de 30 segundos, em Azurém. A média de endereços MAC acompanha a de endereços IP, o que reforça a ideia de que existem estações com dois endereços IP, com grande probabilidade de um ser IPv4 e outro IPv6.

No mesmo gráfico existem duas situações em que se verifica uma queda no número de endereços MAC e IP detectados:

Intervalos 55 a 58 - o número de endereços MAC e IP encontrados é muito reduzido e próximo de zero, o que leva a concluir que existiu várias quebras na ligação nestes intervalos, entre estação de captura e o AP. As estações detectadas durante esse período devem-se ao curto espaço de tempo em que a estação estabeleceu ligação até à próxima quebra.

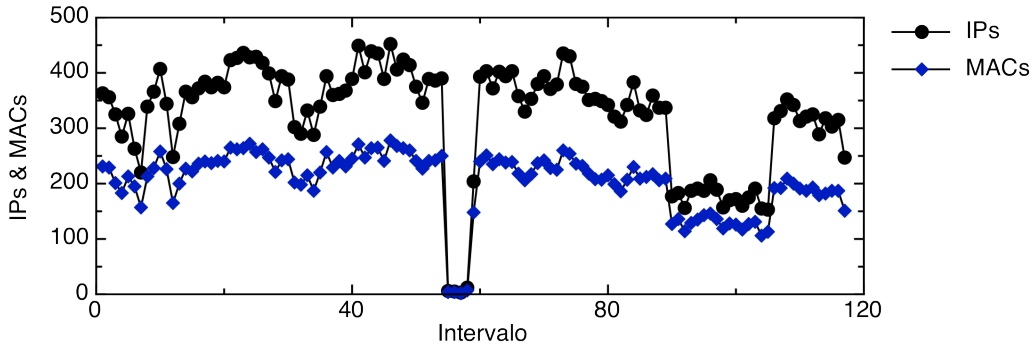


Figura 3.3: IPs e MACs detectados ao longo dos intervalos, Azurém (30s).

Intervalos 90 a 105 - o número de endereços MAC e IP encontrados tem uma quebra bastante acentuada que poderá ter sido provocado por uma quebra de ligação entre algum segmento da rede em algum ponto do campus.

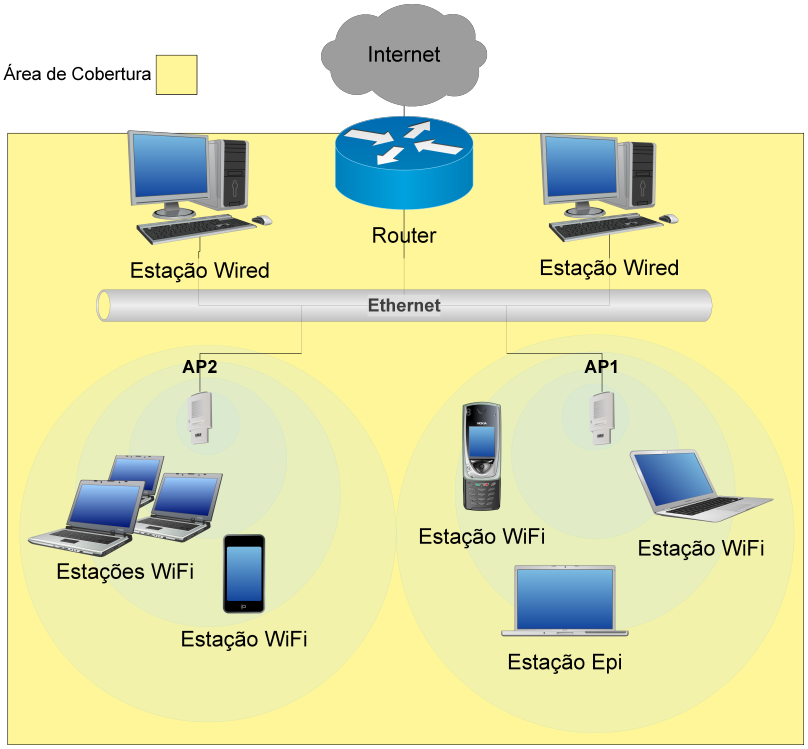


Figura 3.4: Cenário actual na detecção de estações de rede.

Voltando ao número de estações detectadas em Azurém, este prova que a rede de Azurém é uma rede alargada, constituída por vários APs, e que a estação de captura está a receber os pacotes transmitidos em *broadcast* e *multicast* por uma qualquer

estação ligada à rede *WiFi* do campus da Universidade. Também se detectaram estações que se encontravam ligadas à rede através de uma ligação Ethernet. Na lista de endereços MAC aparecem equipamentos Cisco que não são BSSIDs de algum dos APs presentes na vizinhança. Estes são com uma grande probabilidade equipamentos de rede, routers, etc. Os endereços IPv4 também se encontram todos na mesma sub rede que a estação que fez a captura de tráfego, ou seja, a rede estende-se a todo campus. Assim estamos perante um cenário idêntico ao ilustrado na Figura 3.4 em que a cobertura na detecção de estações engloba estações sem fios com ligação estabelecida aos APs, que fazem parte da mesma infra-estrutura de rede e que podem, ou não, estar na vizinhança da estação que realizou a captura e estações ligadas fisicamente através de um cabo de rede na mesma infra-estrutura de rede, sendo estes equipamentos de rede, computadores, etc.

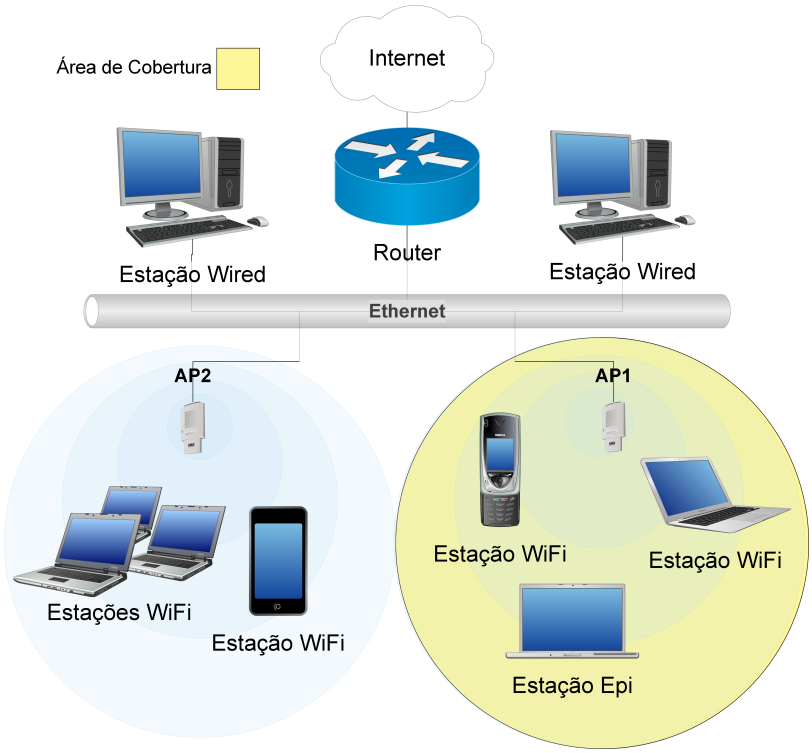


Figura 3.5: Cenário pretendido na detecção de estações de rede.

O cenário ilustrado na Figura 3.5 é o pretendido, onde na procura de estações sejam apenas detectadas as estações que se encontrem associadas ao mesmo AP ou na vizinhança da estação que faz a captura. No entanto, para conseguir o pretendido, é necessário alguma informação adicional que permita filtrar apenas os endereços

MAC vizinhos. Nos pacotes capturados existe ainda a informação dos protocolos que ainda não foram estudados e podem conter informação útil, portanto, foi realizado um estudo de cada um dos protocolos mais relevantes que foram identificados durante a análise do tráfego capturado nos dois locais, descrito nas secções 3.2.1 e 3.2.2.

3.2.5 Protocolos

Ao conjunto de protocolos que foram identificados durante as análises de tráfego foi realizado um estudo detalhado com o objectivo de encontrar alguma informação que permita isolar as estações vizinhas que se encontravam presentes no mesmo AP da estação onde foi realizada a captura de tráfego.

ICMP IPv4

O ICMP (Internet Control Message Protocol) é parte integrante do protocolo IP. É utilizado para fornecer resposta a erros em pacotes IP, diagnóstico e *routing*. Os equipamentos que utilizam IP precisam aceitar este tipo de mensagens e adaptarem-se aos tipos de erros, reportando sempre ao endereço de origem do pacote IP original.

Este difere dos protocolos de transporte TCP e UDP na medida que não é usado tipicamente para trocar dados entre sistemas. Geralmente também não é usado por aplicações de utilizador, com excepção das ferramentas de diagnóstico *ping* e *traceroute*.

As mensagens ICMP são encapsuladas num único pacote IP e este é enviado como um pacote UDP, não fornecendo garantias.

Apesar das mensagens ICMP estarem contidas num pacote normal IP, o seu processamento é efectuado de uma forma diferente. O seu conteúdo é inspeccionado e só depois é gerado a respectiva mensagem de erro, que é entregue à aplicação que gerou o pacote IP original.

O ICMP é usado em diversas situações, como por exemplo:

- Um pacote IP não consegue chegar ao seu destino, porque o seu tempo de vida (TTL, Time to Live) chegou a "0".
- O *gateway* não consegue retransmitir um pacote.
- O *router* indica uma nova e melhor rota para a máquina.

O pacote ICMP é constituído por um cabeçalho com 8 bytes e uma secção de dados de tamanho variável, como está ilustrado na Figura 3.6.

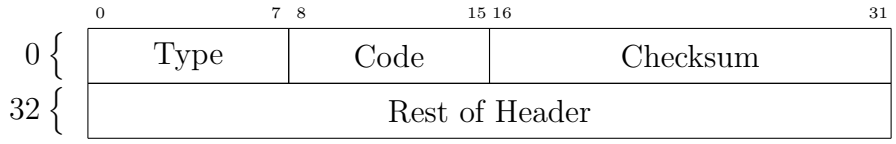


Figura 3.6: Cabeçalho ICMP.

Type - tipo de pacote ICMP.

Code - sub-tipo de um tipo de pacote ICMP.

Checksum - verificação de erros no pacote, este é calculado sobre o cabeçalho ICMP com valor do campo checksum a "0" mais os dados.

Rest of Header - campo de 4 bytes, que varia consoante o tipo e código de ICMP.

A descrição completa do ICMP pode ser encontrada no RFC 792[15].

No âmbito deste projecto, após a análise deste protocolo, não foi encontrado nenhuma informação que permita identificar numa rede sem fios outras estações ligadas no mesmo AP. Informação encontrada mostra que é possível identificar o SO que gera os pacotes de *ping*, a secção de dados de um pacote ICMP desse tipo é gerado através de *padding* e o tamanho do *padding* adicionado varia consoante o SO em questão, em Windows são 32 bytes e em Linux são 56 bytes.

No tráfego capturado, os pacotes ICMP encontrados foram todos do tipo *Ping Request*. O campo de dados tem tamanho de 32 bytes em todos os pacotes capturados e o conteúdo dos dados identifica a aplicação que gerou os pacotes "BitDefender Firewall Broadcast", em vez do *padding* normal que são zeros.

ICMPv6 IPv6

O protocolo ICMPv6 (Internet Control Message Protocol Version 6) é a implementação do ICMP para IPv6 e é parte integrante do IPv6. A sua função é reportar erros, diagnóstico, descoberta de vizinhos e controlo de acesso a grupos *multicast*. Este implementa uma *framework* de extensão para uso futuro.

As mensagens ICMPv6 dividem-se em duas categorias: mensagens de erro e mensagens de informação. Nos pacotes IPv6 as mensagens ICMPv6 são transportadas com o campo *Next Header* a 58.

O processamento das mensagens é feito de acordo com o tipo de pacote ICMPv6 recebido onde são tomadas as acções necessárias. Em resposta a um erro ICMPv6 nunca é enviado uma resposta com outro pacote de erro ICMPv6. O número de pacotes ICMPv6 de erro enviado para uma mesma estação é também limitado para não provocar sobrecarga na rede, sendo enviado um pacote de erro ICMPv6 inicial e depois este continua a sinalizar o erro periodicamente.

O cabeçalho de um pacote ICMPv6 tem tamanho 4 bytes e o seu formato está ilustrado na Figura 3.7.

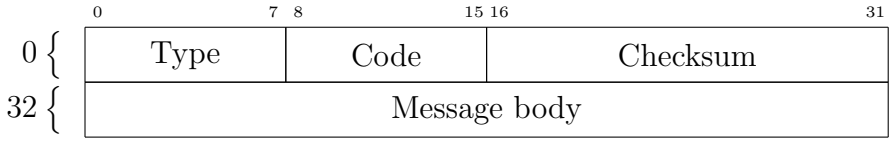


Figura 3.7: Cabeçalho ICMPv6.

Type - tipo de mensagem ICMPv6, de 0-127 é uma mensagem de erro e de 128-255, uma mensagem de informação.

Code - este campo depende do tipo de mensagem ICMPv6.

Checksum - verificação de erros no pacote ICMPv6. O cálculo é efectuado com este campo a "0" e engloba parte do cabeçalho IPv6 com endereço de origem e destino, tamanho do pacote e *Next Header*, dando assim mais integridade a um pacote IPv6, uma vez que este não tem verificação de erros.

Message body - campo com os dados da mensagem relativo ao tipo. Este é de tamanho variável.

A descrição completa do protocolo pode ser encontrada no RFC 4443[16].

Após o estudo do protocolo, no âmbito deste projecto, não foi encontrada informação útil que possa ser usada para a filtragem dos endereços MAC que se encontrem na vizinhança, no entanto, no tráfego capturado foi encontrado um número grande de alguns tipos de pacotes ICMPv6:

Ping Request

Este tipo (*type* = 128) de pacote ICMPv6 é usado para efectuar um *Ping* de forma a verificar a conectividade entre equipamentos. Consiste no envio de pacotes para o destino e depois na espera por uma resposta. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.8 e os campos que a constituem são so seguintes:

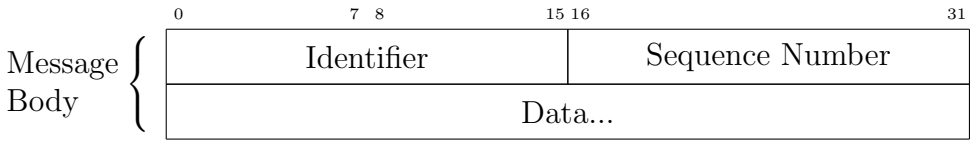


Figura 3.8: Mensagem Echo Request ICMPv6.

Identifier - identificador que ajuda a associar os pedidos às respectivas respostas de *Echo*.

Sequence Number - número de sequência que ajuda a associar os pedidos às respectivas respostas de *Echo*.

Data - dados arbitrários, pode ser vazio. Nos pacotes capturados deste tipo este campo contém "BitDefender Firewall Broadcast".

Multicast Listener Report

Este tipo (*type* = 131) de pacote ICMPv6 é descrito no RFC 2710[17]. É usado em duas situações: quando uma estação se junta a um grupo *multicast* e em resposta a um pedido de *Multicast Listener Query* do router. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.9 e é idêntico às mensagens dos tipos *Multicast Listener Query* e *Multicast Listener Done* sendo os campos que as constituem os seguintes:

Figura 3.9: Mensagem Multicast Listener Report ICMPv6.

Maximum Response Delay - atraso máximo em milisegundos até ao envio da resposta a um pedido. Este campo só tem significado na mensagem *Multicast Listener Query*.

Reserved - este campo é preenchido a zeros no envio e ignorado na recepção.

Multicast Address - para a mensagem *Multicast Listener Report* contém um endereço *multicast* específico que a estação que a envia está a ouvir.

Multicast Listener Report Message v2

Este tipo (*type* = 143) de pacote ICMPv6 é descrito no RFC 3810[18]. É usado pelos nós na rede para reportar o estado dos endereços *multicast* que estão a ouvir. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.10 e os campos que a constituem são os seguintes:

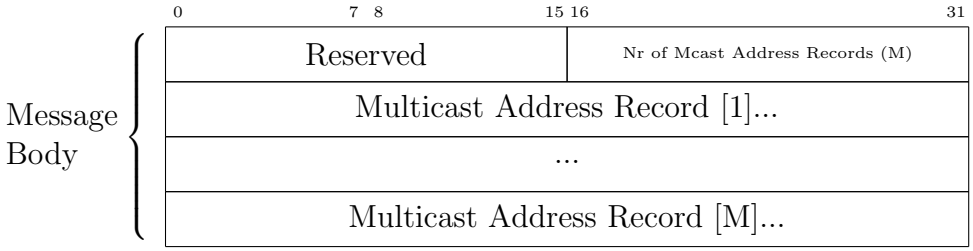


Figura 3.10: Mensagem Multicast Listener Report Message V2 ICMPv6.

Reserved - este campo é preenchido a zeros no envio e ignorado na recepção.

Nr of Mcast Address Records (M) - especifica o número de registos de endereços *multicast* presentes na mensagem.

Multicast Address Record [M] - cada bloco é um registo de endereço *multicast*.

O formato de um registo de um endereço *multicast* está ilustrado na Figura 3.11 e os campos que o constituem são os seguintes:

Record Type - tipo de registo de endereço *multicast*. Nos pacotes capturados foram encontrados os tipos 3 (*CHANGE_TO_INCLUDE_MODE*) e 4 (*CHANGE_TO_EXCLUDE_MODE*). Ambos os tipos significam uma alteração no modo de filtragem, para incluir ou excluir os endereços *unicast* de origem a um endereço *multicast* específico.

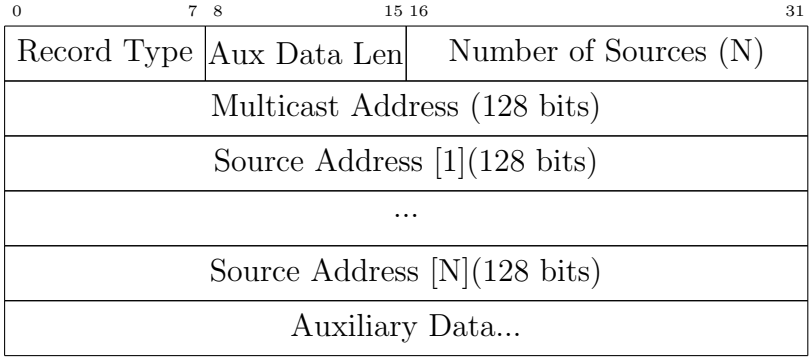


Figura 3.11: Multicast Address Record do MLDv2 no ICMPv6.

Aux Data Len - contém o tamanho do campo *Auxiliary Data*, a "0" indica que o campo não existe.

Number of Sources (N) - número de campos *Source Address* presente no registo.

Multicast Address - endereço *multicast* a que este registo pertence.

Source Address [N] - vector de N endereços *unicast* de origem.

Auxiliary Data - este campo existe apenas para uso futuro.

Neighbor Advertisement

Este tipo (*type* = 136) de pacote ICMPv6 é descrito no RFC 4861[19]. Apenas um único pacote desse tipo foi capturado. É usado pelos nós na descoberta de vizinhos como resposta a um *Neighbor Solicitation* ou é enviado sem solicitação de forma a propagar nova informação rapidamente. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.12 e os campos que a constituem são os seguintes:

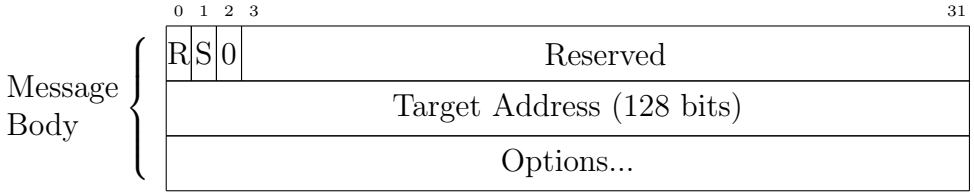


Figura 3.12: Mensagem Neighbor Advertisement ICMPv6.

R - *flag* que indica se o *host* é um router.

S - *flag* que indica se é uma resposta a uma solicitação.

O - *flag* que indica se deve substituir uma entrada já existente na *cache*.

Reserved - este campo é preenchido a zeros no envio e ignorado na recepção.

Target Address - quando é uma resposta a uma solicitação contém o endereço alvo. Quando não houve solicitação, contém o endereço de *link-layer* que foi alterado.

Options - contém opções, no entanto, a especificação apenas define a opção *Target link-layer Address*. Este contém o endereço de *link-layer* alvo.

Neighbor Solicitation

Este tipo (*type* = 135) de pacote ICMPv6 é também descrito no RFC 4861[19]. É usado pelos nós na descoberta de vizinhos para solicitar o endereço de *link-layer* de um nó alvo enquanto também informa do seu endereço de *link-layer*. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.13 e os campos que a constituem são os seguintes:

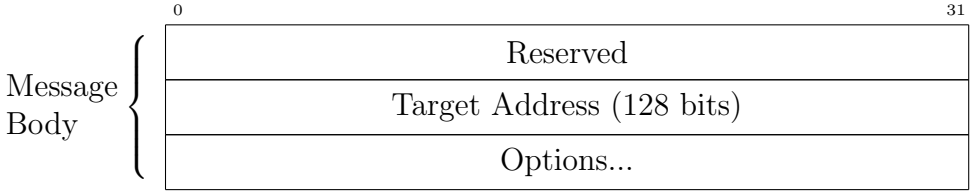


Figura 3.13: Mensagem Neighbor Solicitation ICMPv6.

Reserved - este campo é preenchido a zeros no envio e ignorado na recepção.

Target Address - endereço IP que é alvo da solicitação.

Options - contém opções, no entanto, a especificação apenas define a opção *Source link-layer address*. Este contém o endereço de *link-layer* de origem.

Router Advertisement

Este tipo (*type* = 134) de pacote ICMPv6 é também descrito no RFC 4861[19]. É usado pelos routers para se anunciarem periodicamente na rede ou quando são solicitados. O formato da mensagem que vai dentro do corpo de uma mensagem ICMPv6 está ilustrado na Figura 3.14 e os campos que a constituem são os seguintes:

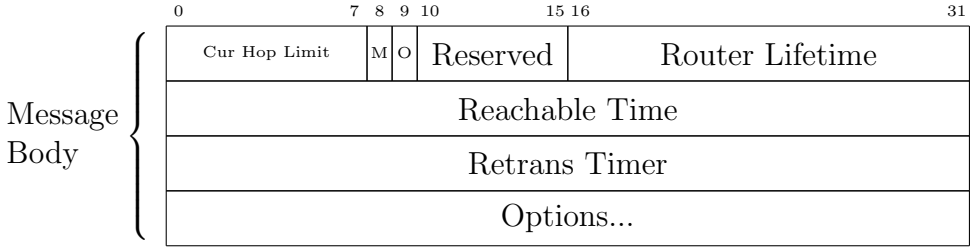


Figura 3.14: Mensagem Router Advertisement ICMPv6.

Cur Hop Limit - valor predefinido que deve ser colocado no campo *Hop Count* do cabeçalho de um pacote IP.

M - *flag* que indica que estão disponíveis endereços por DHCPv6 (Dynamic Host Configuration Protocol for IPv6).

O - *flag* que indica que estão disponíveis outras informações de configurações por DHCPv6.

Reserved - este campo é preenchido a zeros no envio e ignorado na recepção.

Router Lifetime - indica o tempo de vida do router predefinido na rede, em segundos. Este campo a "0" indica que não é o router predefinido.

Reachable Time - tempo em milisegundos que um nó demora a assumir que outro nó vizinho se encontra ao alcance depois de receber a confirmação.

Retrans Timer - tempo em milisegundos entre a retransmissão de uma solicitação de vizinhos.

Options - contém opções. São especificadas as opções: *Source link-layer address* que contém o endereço de *link-layer* da *interface* do router que envia a mensagem, *MTU* (MTU, Maximum Transmission Unit) que indica o tamanho máximo do pacote que pode ser transmitido e *Prefix Information* que especifica o prefixo que é usado para a auto-configuração *stateless*.

A análise dos tipos de mensagens ICMPv6 encontrados no tráfego capturado foi feita usando as diversas especificações diferentes que definem os mesmos tipos. A especificação NDP (Neighbor Discovery Protocol) define as mensagens do tráfego capturado para a descoberta de nós e routers na rede, no entanto, a descoberta acontece ao nível de rede local o que não acrescenta informação útil que possa ser usada para a filtragem dos endereços MAC que se encontram na vizinhança.

Os restantes tipos de mensagens encontrados fazem parte da gestão dos grupos *multicast*. No IPv6 este faz parte do ICMPv6, contrariamente ao IPv4 que usa o IGMP (Internet Group Management Protocol). Estas mensagens também não contêm informação útil para o pretendido.

IGMP IPv4

O IGMP é um protocolo de comunicações usado por clientes e routers adjacentes em redes IP para estabelecer grupos *multicast*, controlando os membros e a entrada e saída de *hosts* dos grupos.

Este é apenas usado em redes IPv4, sendo substituído pelo MLD (Multicast Listener Discovery) que é um componente do ICMPv6 em redes IPv6.

As mensagens IGMP são encapsuladas directamente dentro de um pacote IP e assim como no ICMP não necessitam de nenhum protocolo de transporte como UDP ou TCP.

As mensagens IGMP são enviadas pelos clientes para o router local como pedido para se juntar ou sair como membro de um grupo *multicast*, por sua vez, o router local envia mensagens IGMP para os routers adjacentes para determinar quais são os endereços dos grupos *multicast* de interesse. De forma a manter os membros dos grupos actualizados estes vão enviando mensagens periodicamente. Também podem ser efectuadas pedidos para grupos específicos para determinar o seu estado e efectuar pedidos para determinar se um sistema pretende receber mensagens enviadas para um grupo *multicast* de um endereço de origem *unicast* específico.

O IGMP existe em 3 versões, IGMPv1 descrito no RFC 1112[20], IGMPv2 descrito no RFC 2236[21] e IGMPv3 descrito nos RFC 3376[22] e RFC 4604[23].

Na Figura 3.15, está ilustrado o formato de pacote usado pela versão IGMPv3 e os campos que o constituem são os seguintes:

MaxRespCode - especifica o tempo máximo permitido para enviar a resposta a um *report*.

Checksum - verificação de erros do pacote IGMP, engloba a mensagem IGMP completa.

Group Address - endereço do grupo *multicast* pedido no caso de ser um pedido geral é preenchido a zeros.

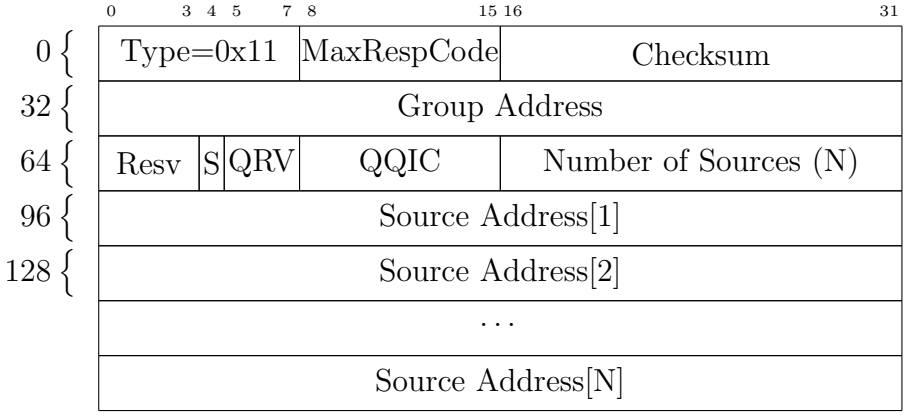


Figura 3.15: Estrutura do pacote IGMPv3.

Resv - campo reservado, enviado a zeros e ignorado quando recebido.

S - *Suppress Router Side Processing flag* indica aos routers que a recebem para ignorar o tempo normal de actualizações.

QRV - *Querier's Robustness Variable* quando este campo não é zero, quem envia este pacote espera que o router faça actualização à sua *Robustness Variable* para igualar o valor deste campo.

QQIC - *Querier's Query Interval Code* especifica o intervalo que vai ser usado para fazer queries.

Number of Sources (N) - número de endereços de origem presentes nos campos seguintes. Em pedidos a grupos específicos ou em geral, o valor deste campo é zero, sendo limitado o seu tamanho ao MTU em pedidos *group-and-source-specific*.

Source Address [i] - endereços de origem IP *unicast*.

Na Figura 3.16 está ilustrado o formato de pacote usado pela versão IGMPv2 e os campos que o constituem são os seguintes:

Type - tipo de pedido. Pode ser *Membership Query* (0x11), *Membership Report* (IGMPv1: 0x12, IGMPv2: 0x16) e *Leave Group* (0x17). O IGMPv3 adiciona um novo tipo *Membership Report* (0x22).

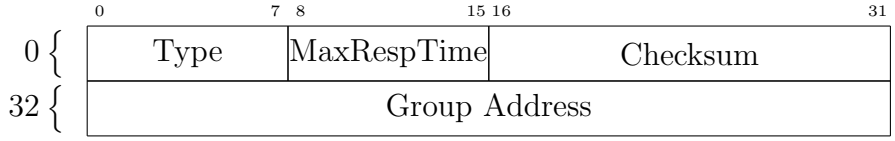


Figura 3.16: Estrutura do pacote IGMPv2.

Max Resp Time - tempo limite de espera pela resposta correspondente. O campo só tem significado no caso de o tipo ser "0x11", noutros tipos é "0" e ignorado.

Após o estudo deste protocolo, no âmbito deste projecto, não foi encontrada informação útil que possa ser usado na filtragem dos endereços MAC que se encontrem na vizinhança.

No tráfego capturado, os pacotes IGMP são IGMPv2 são todos do mesmo tipo "0x16" *Membership Report/Join*.

DHCP

O DHCP (Dynamic Host Configuration Protocol) é um protocolo de serviço TCP/IP que permite a uma estação obter a configuração automática do endereço de rede IP e outros parâmetros de configuração. A primeira versão do protocolo DHCP surgiu para redes IPv4, sendo descrito no RFC 2131[24], mais tarde surgiu outra versão, o DHCPv6, que é o DHCP para redes IPv6 descrito no RFC 3315[25]. Estes são considerados separados uma vez que os detalhes do protocolo são diferentes, no entanto, mantêm o mesmo propósito.

O DHCP usa as portas 67 em UDP para enviar mensagens para o servidor e a porta 68 em UDP para mensagens para o cliente. A operação do DHCP segue as seguintes fases: *IP discovery*, *IP lease offer*, *IP request* e *IP lease acknowledgement*.

Quando um cliente DHCP se encontra na mesma sub rede que o servidor a sua comunicação é feita em UDP *broadcast*. Se o cliente e o servidor estiverem em sub redes diferentes, as operações *IP discovery* e IP *request* usam UDP *broadcast* e as operações de *IP lease* e *IP lease acknowledgement*, mensagens para o endereço *unicast*.

O DHCP opera da seguinte forma:

1. *DHCP discovery* - o cliente envia uma mensagem em *broadcast* para descoberta de servidores DHCP disponíveis.

2. *DHCP offer* - o servidor ao receber um pedido de IP de um cliente reserva um endereço IP para o cliente e envia uma mensagem DHCP *offer* para o cliente. A mensagem contém: endereço MAC do cliente, endereço IP oferecido, máscara de rede, duração do aluguer e o endereço IP do servidor que envia a mensagem.

3. *DHCP request* - um cliente pode receber várias ofertas DHCP de múltiplos servidores mas apenas aceita uma. Envia depois uma mensagem de resposta em *broadcast* do tipo DHCP *request* e os servidores através do identificador que vai na mensagem sabem se a sua oferta foi aceite. Quando não é aceite o aluguer do IP é libertado. A mensagem é enviada em *broadcast* uma vez que o cliente nesta fase ainda não recebeu endereço IP e permite apenas numa mensagem informar os vários servidores DHCP.

4. *DHCP acknowledgement* - quando o servidor recebe o DHCP *request* do cliente, a configuração entra na sua fase final e é enviado para o cliente um DHCP *ack* com a duração do aluguer e alguma informação adicional requisitada. Aqui o processo de configuração do IP está completo, sendo esperado pelo protocolo que o cliente configure a rede com os parâmetros negociados. Algum conflito de endereços IP pode ser prevenido usando o ARP.

5. *DHCP information* - o cliente pode pedir informação adicional ao servidor em relação à enviada no DHCP *offer*. O cliente também pode pedir a repetição dos dados para uso particular de alguma aplicação, como por exemplo um navegador para obter a configurações de um *proxy*. No entanto estes pedidos não renovam o aluguer do IP.

6. *DHCP releasing* - o cliente envia um pedido ao servidor para libertar o aluguer do seu endereço IP e depois desactiva o seu endereço IP.

Um servidor DHCP pode fornecer parâmetros opcionais de configuração aos clientes. O cliente pode seleccionar, manipular ou sobrepor os parâmetros fornecidos pelo servidor DHCP.

O formato de pacote DHCP está ilustrado na Figura 3.17 e os campos que o constituem são os seguintes:

OP - tipo de mensagem.

HTYPE - tipo de endereço de *hardware*.

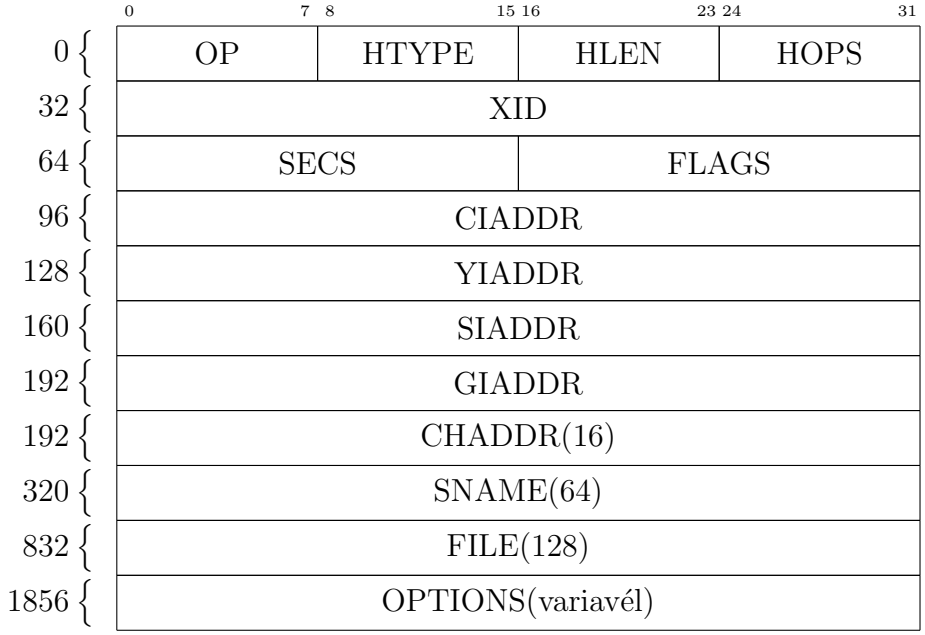


Figura 3.17: Estrutura do pacote DHCP.

HLEN - tamanho do endereço de *hardware*.

HOPS - usado opcionalmente pelos agentes *relay* quando o arranque é feito por outro agente *relay*.

XID - número aleatório gerado no cliente e usado depois nas mensagens trocadas entre o cliente e o servidor, servindo como um identificador de sessão.

SECS - preenchido pelo cliente, representa o tempo em segundos desde que começou o processo de requisição de endereço ou renovação.

FLAGS - campo usado para *flags*, onde o bit *B* mais à esquerda é usado como *broadcast flag*, os restantes são reservados para uso futuro e têm que estar a zeros. O bit *B* indica se o cliente é capaz ou não de receber tramas IP *unicast* antes de o TCP/IP estar configurado.

CIADDR - endereço IP do cliente, é usado nos estados em que o cliente responde a pedidos ARP.

YIADDR - endereço IP para o cliente.

SIADDR - endereço IP do servidor.

GIADDR - endereço IP do *gateway*.

CHADDR - endereço de *hardware* do cliente.

SNAME - nome do servidor.

FILE - ficheiro para arranque.

OPTIONS - opções adicionais.

Nos dois locais onde foi capturado tráfego foram encontrados pacotes DHCP dos tipos, *request*, *offer*, *NAK*, *inform*, *discover* e *ACK*. Observando a informação contida dentro dos mesmos, o tipo de *hardware* do cliente é sempre identificado como Ethernet. No campo com opções adicionais existe também uma identificação do cliente, no entanto aqui o tipo de *hardware* também é identificado como Ethernet e o resto dos campos não fornecem nenhuma outra informação relevante.

Assim, nos pacotes trocados no processo de obtenção automática de endereço IP, também não foi encontrada informação útil que possa ser usada para a filtragem dos endereços MAC que se encontrem na vizinhança.

Multicast DNS (MDNS)

O protocolo MDNS é o baseado no DNS e faz parte do Zeroconf (Zero configuration networking)[1] que ainda se encontra em desenvolvimento.

Sendo baseado no DNS, este protocolo mantém a estrutura de mensagens DNS com os mesmos códigos de operação, respostas e registos. A especificação do MDNS acrescenta apenas como o cliente deve enviar mensagens DNS em *multicast* e como as estações devem responder às mensagens colaborando de uma forma útil.

Para o estudo deste protocolo é usado o DNS, para procurar alguma informação que ajude a isolar estações na vizinhança numa rede sem fios. O DNS é descrito nos documentos RFC 1034[26], RFC 4033[27] e RFC 1035[28].

Um elemento básico do DNS é o recurso que contém o registo (RR, Resource Record) ilustrado na Figura 3.18. Os campos que constituem um RR são:

NAME - nome do dono a quem o RR pertence.

TYPE - código do tipo de RR.

[1]Consiste num conjunto de técnicas para criar uma rede IP usável de forma automática.

NAME	TYPE	CLASS	TTL	RDLENGTH	RDATA
variável	2 bytes	2	4	2	variável

Figura 3.18: Estrutura de um RR de DNS.

CLASS - classe do código RR.

TTL - tempo de vida que o RR permanece em *cache* até voltar a ser necessária a consulta da sua informação na origem. Este campo a zero significa que o RR nunca pode ser mantido em *cache*.

RDLENGTH - tamanho do campo *RDATA* em bytes.

RDATA - dados que descrevem o RR, o seu formato varia conforme os campos *TYPE* e *CLASS*.

A mensagem que é enviada dentro de pacote DNS encontra-se dividida em 5 secções podendo algumas das secções ser encontradas vazias para determinados casos. Estas secções estão identificadas e descritas na Tabela 3.13.

Secção	Descrição
Header	cabeçalho da mensagem.
Question	questão para o servidor de DNS.
Answer	RRs que respondem à questão.
Authority	RRs que apontam para uma autoridade.
Additional	RRs que contêm informação adicional.

Tabela 3.13: Secções de uma mensagem DNS.

O *Header* é uma secção obrigatória, uma vez que indica as restantes secções que se encontram presentes na mensagem DNS e se esta é uma pesquisa ou resposta. Este será estudado depois das restantes secções.

A secção *Question* está ilustrada na Figura 3.19 e descreve a pesquisa ao servidor através dos campos: tipo ($QTYPE$), classe ($QTYPE$) e domínio ($QNAME$).

QNAME	QTYPE	QCLASS
variável	2 bytes	2

Figura 3.19: Secção Question de uma mensagem DNS.

As restantes três secções são de formato idêntico pois cada uma contém uma lista de RRs, que pode ser vazia. No entanto a lista de RRs que cada uma dessas secções

contém são diferentes e para propósitos diferentes. A secção *Answer* contém RRs de resposta a uma pesquisa, *Authority* contém as RRs que apontam para um servidor DNS autoritário e *Additional* as RRs adicionais relacionados com uma pesquisa.

Na Figura 3.20 está ilustrado o cabeçalho (*Header*) de um pacote DNS.

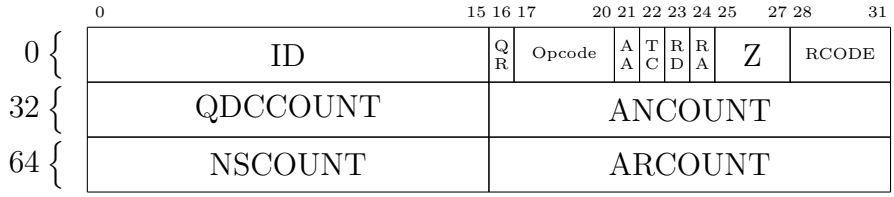


Figura 3.20: Cabeçalho de uma mensagem DNS.

ID - identifica o programa que gerou a pesquisa, este é copiado para as respostas.

QR - identifica se a mensagem é uma pesquisa ou uma resposta.

Opcode - tipo de mensagem de pesquisa, este é gerado pela origem e copiado depois para a resposta.

AA - *Authoritative Answer*, indica se o servidor que respondeu é um servidor DNS autoritário sobre o domínio em questão.

TC - *TrunCation*, indica se a mensagem foi truncada devido ao tamanho maior que o permitido no canal de transmissão.

RD - *Recursion Desired*, indica para efectuar uma pesquisa recursiva ao servidor.

RA - *Recursion Available*, indica se está disponível pesquisa recursiva.

Z - reservado para uso futuro.

RCODE - *Response Code*, faz parte da resposta e indica um erro.

QDCOUNT - número de entradas presente na secção *Question*.

ANCOUNT - número de elementos da lista de RRs presente na secção *Answer*.

NSCOUNT - número de elementos da lista de RRs presente na secção *Authority*.

ARCOUNT - número de elementos da lista de RRs presente na secção *Additional*.

Este protocolo usa o formato de mensagens idêntico ao DNS e neste não foi encontrada informação útil que possa de alguma forma ser usada para a filtragem dos endereços MAC que se encontrem na vizinhança.

Link-Local Multicast Name Resolution(LLMNR)

O protocolo LLMNR também é baseado no DNS. É usado para efectuar a resolução de nomes de estações presentes na mesma ligação de rede local, em redes IPv4 e IPv6. Este protocolo é incluído nas versões dos SOs Windows Vista, Server 2008 e 7.

Para receber mensagens com pesquisas, as estações ouvem em UDP na porta 5355 nos endereços *multicast*: 224.0.0.252 para IPv4 e FF02::1:3 para IPv6. Para as mensagens de resposta, as estações ouvem em TCP na porta 5355 no endereço *unicast*.

Assim como o MDNS, este protocolo também usa o formato de mensagens baseado no DNS, com as mesmas secções (Tabela 3.13) e elemento RR (Figura 3.18).

Na Figura 3.21 está ilustrada a secção do cabeçalho (*Header*) do LLMNR que apresenta diferenças relativas ao que foi estudado no protocolo MDNS.

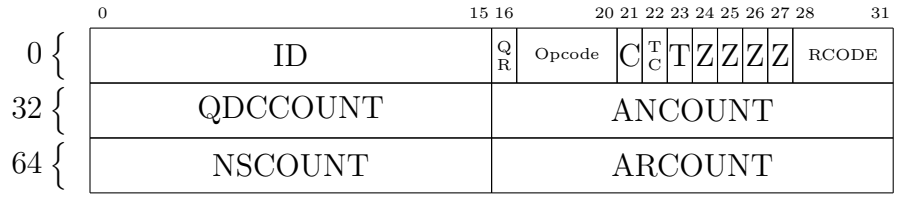


Figura 3.21: Estrutura do cabeçalho do pacote LLMNR.

ID - campo que identifica o programa que gerou a pesquisa.

QR - identifica se a mensagem é uma pesquisa ou uma resposta.

Opcode - especifica o tipo de mensagem de pesquisa, este é gerado pela origem e copiado depois para a resposta. A especificação define este valor para "0", deixando aberto para futuro o uso de outros valores neste campo.

C - conflito, indica que foram enviadas várias mensagens para esta pesquisa.

T - tentativa, é usado numa mensagem de resposta se quem respondeu tem nome competente mas ainda não foi verificada a unicidade do nome.

Z - reservado para uso futuro.

QDCOUNT - número de entradas presente na secção *Question*.

ANCOUNT - número de elementos da lista de RRs presente na secção *Answer*.

NSCOUNT - número de elementos da lista de RRs presente na secção *Authority*.

ARCOUNT - número de elementos da lista de RRs presente na secção *Additional*.

A descrição completa deste protocolo pode ser encontrada no RFC 4795[29].

Este protocolo é muito idêntico ao MDNS e também não acrescenta informação nova que seja útil para a filtragem das estações presentes na vizinhança.

Simple Service Discovery Protocol (SSDP)

O SSDP é um protocolo para descobrir e anunciar serviços de rede. Funciona sem uma configuração do tipo servidor ou sequer algum tipo de configuração. A especificação do SSDP[30] ainda se encontra em *Draft*. O formato de pacotes é texto, baseado em HTTP sobre UDP[31] como protocolo de transporte na porta 1900.

Os anúncios são feitos para um endereço *multicast*, sendo este 239.255.255.250 para IPv4 e FF0X::C nos diferentes *scopes* de rede IPv6.

Este usa um formato de mensagem *HTTP Notify*[32] para anunciar o estabelecimento ou remoção de serviços para o grupo *multicast*. Quando um cliente pretende fazer uma descoberta dos serviços disponíveis na rede usa o formato de mensagem *HTTP M-Search*[1]. As respostas são depois enviadas em *unicast* com o endereço e porta do pedido recebido por *multicast*. Os clientes também podem anunciar na rede com os serviços que podem fornecer.

Os serviços são sempre identificados pelo par único constituído pelo URI (Uniform Resource Identifier) e USN (Unique Service Name). O USN é sempre usado para diferenciar serviços do mesmo tipo. Adicionalmente os serviços tem associados a localização e um tempo de validade. A localização identifica como um serviço deve ser contactado, podendo ser incluído várias localizações para o mesmo URI. O tempo de validade informa quanto tempo o cliente deve manter a informação sobre determinado serviço em *cache*.

Na Listagem 3.1 está ilustrado o texto de um pacote SSDP *M-Search HTTP* capturado, com a descrição de cada linha:

[1]É uma mensagem *HTTP Search*[33] a que é aplicado uma extensão, segundo o RFC 2774[34].

```
1 M-SEARCH * HTTP/1.1
2 Host:239.255.255.250:1900
3 ST:urn:schemas-upnp-org:device:InternetGatewayDevice:1
4 Man:"ssdp:discover"
5 MX:3
```

Listagem 3.1: Texto contido num pacote SSDP M-Search HTTP capturado.

Linha 1 - método obrigatório do pedido *M-SEARCH*, seguido do URI que está a ser pedido,"∗" é sempre usado em pacotes enviado em *multicast*, por fim a versão "HTTP/1.1".

Linha 2 - endereço e porta para onde foi enviado o pacote.

Linha 3 - cabeçalho *ST* contém um único URI, este indica um tipo específico de serviço que o cliente pretende descobrir.

Linha 4 - representa uma extensão obrigatória, neste caso uma funcionalidade que identifica o protocolo.

Linha 5 - tempo máximo em segundos para o atraso do envio da resposta. Para balanceamento de carga na rede, a resposta vai ser atrasada um valor aleatório entre 0 e o valor no cabeçalho *MX* segundos.

O outro método usado por este protocolo é o *HTTP Notify*. Na Listagem 3.2 está ilustrado o exemplo de um pacote capturado. Existem dois tipos de anúncios, *sspd:alive* quando um serviço quer anunciar a sua presença na rede e o *sspd:byebye* quando um serviço pretende anunciar que vai deixar de estar presente na rede. O exemplo é relativo ao anúncio *sspd:alive*.

```
1 NOTIFY * HTTP/1.1
2 Host:239.255.255.250:1900
3 NT:urn:schemas-upnp-org:service:ConnectionManager:1
4 NTS:ssdp:alive
5 Location:http:// ... udhisapi.dll?content=uuid: ...
6 USN:uuid: ... ::urn:schemas-upnp-org:service:ConnectionManager:1
7 Cache-Control:max-age=900
8 Server:Microsoft-Windows-NT/5.1 UPnP/1.0 UPnP-Device-Host/1.0
```

```
9  OPT:"http://schemas.upnp.org/upnp/1/0/"; ns=01
10 01-NLS:3091d725becb21bf6a3669fbf4abe163
```

Listagem 3.2: Texto contido num pacote SSDP Notify capturado.

Linha 1 - método obrigatório do pedido *NOTIFY*, seguido do URI que está a ser pedido, "∗" é sempre usado em pacotes enviado em *multicast*, por fim a versão "HTTP/1.1".

Linha 2 - endereço e porta para onde foi enviado o pacote, uso obrigatório em ambos os tipos de anúncios.

Linha 3 - tipo de serviço, uso obrigatório em ambos os tipos de anúncios.

Linha 4 - tipo de anúncio, uso obrigatório em ambos os tipos de anúncios.

Linha 5 - informação de localização do serviço, uso obrigatório no tipo *sspd:alive*.

Linha 6 - USN do serviço, uso obrigatório em ambos os tipos de anúncios.

Linha 7 - tempo de vida do serviço, uso obrigatório no tipo *sspd:alive*.

Linha 8, 9, 10 - não são de uso obrigatório pelo protocolo, logo, vão ser ignoradas.

Este protocolo usa mensagens no formato de alguns métodos HTTP conhecidos. Estes podem incluir campos extra mas que são ignorados pelo protocolo quando não são reconhecidos. Considerando apenas os campos que são reconhecidos pelo protocolo, não foi encontrada informação útil nos mesmos que possa ser usada para a filtragem das estações presentes na vizinhança.

Web Services Dynamic Discovery (WS-Discovery)

O WS-Discovery[35] é um protocolo de descoberta da localização de serviços por *multicast* para uma rede local.

A comunicação entre os nós da rede é efectuada usando normas de serviços Web como o SOAP (Simple Object Access Protocol) em UDP[36]. Funciona na porta 3702 e nos endereços *multicast* 239.255.255.250 em IPv4 e FF02::C no *scope link-local* em IPv6.

Este é usado nos SOs Windows e está integrado como parte das tecnologias Windows Rally[1] e Devices Profile for Web Services[2].

Permite o anúncio e pesquisa de serviços e ainda localizar serviços anteriormente referenciados na sub rede local. As mensagens de pesquisa de serviços são enviadas para o grupo *multicast*. Caso o serviço exista em alguma estação na sub rede, a resposta é enviada em *unicast*. Em redes grandes este protocolo pode suprimir as mensagens em *multicast* se existir um *proxy* de descoberta de serviços. Um serviço novo é anunciado no grupo *multicast* para que os clientes existentes na rede o possam detectar sem necessitar de fazer uma pesquisa.

As mensagens definidas por este protocolo são *Hello*, *Bye*, *Probe* e *Resolve*. Existe outro tipo de mensagens definidas que são enviados em *unicast*, como nunca vão fazer parte do tráfego capturado no âmbito deste projecto, não fazem parte deste estudo. As mensagens são usadas quando:

Hello - mensagem enviada quando um serviço pretende anunciar que está disponível na rede. Este contém também a informação do serviço.

Bye - mensagem enviada quando um serviço se prepara para abandonar a rede.

Probe - mensagem enviada por um cliente que procura um serviço pelo seu tipo.

Resolve - mensagem enviada por um cliente que procura um serviço pelo seu endereço.

O conteúdo destas mensagem é enviado num envelope em SOAP em UDP que é basicamente um documento XML. Nas Listagens 3.3 para *Hello*, 3.4 para *Bye*, 3.5 para *Probe* e 3.6 para *Resolve* está ilustrado um exemplo dos conteúdos das respectivas mensagens, com uma descrição das linhas mais importantes.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <soap:Envelope
3       xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
4       xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
5       xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery"
6       xmlns:wsdp="http://schemas.xmlsoap.org/ws/2006/02/devprof">
7 <soap:Header>
```

[1] `http://msdn.microsoft.com/en-us/windows/hardware/gg463018`.
[2] `http://specs.xmlsoap.org/ws/2006/02/devprof/`.

```
8       <wsa:To>
9           urn:schemas-xmlsoap-org:ws:2005:04:discovery
10      </wsa:To>
11      <wsa:Action>
12          http://schemas.xmlsoap.org/ws/2005/04/discovery/Hello
13      </wsa:Action>
14      <wsa:MessageID>
15          urn:uuid:0f5d604c-81ac-4abc-8010-51dbffad55f2
16      </wsa:MessageID>
17      <wsd:AppSequence InstanceId="2"
18          SequenceId="urn:uuid:369a7d7b-5f87-48a4-aa9a-189edf2a8772"
19          MessageNumber="14">
20      </wsd:AppSequence>
21  </soap:Header>
22  <soap:Body>
23      <wsd:Hello>
24          <wsa:EndpointReference>
25              <wsa:Address>
26                  urn:uuid:37f86d35-e6ac-4241-964f-1d9ae46fb366
27              </wsa:Address>
28          </wsa:EndpointReference>
29          <wsd:Types>wsdp:Device</wsd:Types>
30          <wsd:MetadataVersion>2</wsd:MetadataVersion>
31      </wsd:Hello>
32  </soap:Body>
```

Listagem 3.3: Formato do documento XML numa mensagem Hello no WS-D.

Linha 8-10 - indica que é uma mensagem *multicast*.

Linhas 11-13 - indica que esta é uma mensagem do tipo *Hello*.

Linha 17-20 - contém um identificador e número da mensagem, permitindo à aplicação manter a sequência de mensagens.

Linha 25-27 - contém o endereço do *endpoint*, este é único do serviço que é constante em todas as *interfaces* de rede, endereços de transporte e IPv4/IPv6.

Linha 29 - contém o tipo de serviço que está a ser anunciado.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
4      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
5      xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery">
6  <soap:Header>
7      <wsa:To>
8          urn:schemas-xmlsoap-org:ws:2005:04:discovery
9      </wsa:To>
10     <wsa:Action>
11         http://schemas.xmlsoap.org/ws/2005/04/discovery/Bye
12     </wsa:Action>
13     <wsa:MessageID>
14         urn:uuid:193ccfa0-347d-41a1-9285-f500b6b96a15
15     </wsa:MessageID>
16     <wsd:AppSequence InstanceId="2"
17         SequenceId="urn:uuid:369a7d7b-5f87-48a4-aa9a-189edf2a8772"
18         MessageNumber="21">
19     </wsd:AppSequence>
20 </soap:Header>
21 <soap:Body>
22     <wsd:Bye>
23         <wsa:EndpointReference>
24             <wsa:Address>
25                 urn:uuid:37f86d35-e6ac-4241-964f-1d9ae46fb366
26             </wsa:Address>
27         </wsa:EndpointReference>
28     </wsd:Bye>
29 </soap:Body>
```

Listagem 3.4: Formato do documento XML numa mensagem Bye no WS-D.

Linha 7-9 - indica que é uma mensagem *multicast*.

Linhas 10-12 - indica que esta é uma mensagem do tipo *Bye*.

Linha 16-19 - contém um identificador e número da mensagem, permitindo à aplicação manter a sequência de mensagens.

Linha 24-26 - contém o endereço do *endpoint*, que pretende abandonar a rede.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <soap:Envelope
3       xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
4       xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
5       xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery"
6       xmlns:wsdp="http://schemas.xmlsoap.org/ws/2006/02/devprof">
7  <soap:Header>
8       <wsa:To>
9            urn:schemas-xmlsoap-org:ws:2005:04:discovery
10      </wsa:To>
11      <wsa:Action>
12           http://schemas.xmlsoap.org/ws/2005/04/discovery/Probe
13      </wsa:Action>
14      <wsa:MessageID>
15           urn:uuid:29cf10da-5c41-4d55-b184-5ee15e38ce23
16      </wsa:MessageID>
17 </soap:Header>
18 <soap:Body>
19      <wsd:Probe>
20           <wsd:Types>wsdp:Device</wsd:Types>
21      </wsd:Probe>
22 </soap:Body>
```

Listagem 3.5: Formato do documento XML numa mensagem Probe no WS-D.

Linha 8-10 - indica que é uma mensagem *multicast*.

Linha 11-13 - indica que esta é uma mensagem do tipo *Probe*.

Linha 14-16 - contém o identificador da mensagem que depois é referenciado na resposta em *RelatesTo*.

Linha 20 - contém o tipo de serviço que o cliente pretende localizar.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <soap:Envelope
3       xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
4       xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
5       xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery">
```

```
6  <soap:Header>
7      <wsa:To>
8          urn:schemas-xmlsoap-org:ws:2005:04:discovery
9      </wsa:To>
10     <wsa:Action>
11         http://schemas.xmlsoap.org/ws/2005/04/discovery/Resolve
12     </wsa:Action>
13     <wsa:MessageID>
14         urn:uuid:38d1c3d9-8d73-4424-8861-6b7ee2af24d3
15     </wsa:MessageID>
16 </soap:Header>
17 <soap:Body>
18     <wsd:Resolve>
19         <wsa:EndpointReference>
20             <wsa:Address>
21                 urn:uuid:37f86d35-e6ac-4241-964f-1d9ae46fb366
22             </wsa:Address>
23         </wsa:EndpointReference>
24     </wsd:Resolve>
25 </soap:Body>
26 </soap:Envelope>
```

Listagem 3.6: Formato do documento XML numa mensagem Resolve no WS-D.

Linha 7-9 - indica que é uma mensagem *multicast*.

Linha 10-12 - indica que esta é uma mensagem do tipo *Resolve*.

Linha 13-15 - contém o identificador da mensagem que depois é referenciado na resposta.

Linha 20-22 - contém o endereço do *endpoint* que se pretende resolver.

No conjunto de documentos XML transmitidos em *multicast* por este protocolo não foi identificado no seu conteúdo nenhuma informação útil que possa ser usada para filtrar os endereços MAC das estações sem fios que se encontrem na vizinhança.

Dropbox LAN sync Discovery Protocol

Este protocolo pertence à aplicação Dropbox e faz parte de uma das suas funcionalidades. Este permite que a sincronização entre várias estações, com o Dropbox

instalado e usando a mesma conta, seja efectuada directamente através da rede local, sem intervenção da *cloud*.

A aplicação envia pacotes em *broadcast* periodicamente para a rede local para anunciar a sua presença na rede. Ao receber os pacotes, uma estação na rede local descobre que uma outra estação também está a usar o Dropbox. O conteúdo da mensagem irá identificar a conta que está a ser usada.

Não foi encontrado documentação relativa a este protocolo, no entanto, a porta 17500 em UDP usada pelo protocolo está identificada no IANA.

Na Listagem 3.7 está ilustrado o conteúdo de uma mensagem que estava dentro de um dos pacotes capturados.

```
1 {
2     "host_int": 28237344,
3     "version": [1, 8],
4     "displayname": "bruno-castros-computer",
5     "port": 17500,
6     "namespaces": [20575370, 4861559]
7 }
```

Listagem 3.7: Mensagem num pacote capturado do Dropbox LAN sync Discovery.

Como não existe informação sobre o mesmo, pois trata-se de um protocolo proprietário, a análise será realizada observando o conteúdo do exemplo de um dos pacotes capturados onde os restantes são idênticos. No formato da mensagem pode ser identificado um conjunto de pares, parâmetro e valor.

Para cada parâmetro identificado, é realizado uma tentativa de identificar a sua função:

Linha 2 - parâmetro *host_int* que parece identificar a máquina ou a conta Dropbox que contém aquela instalação através de um número inteiro.

Linha 3 - parâmetro *version* identifica a versão do Dropbox.

Linha 4 - parâmetro *displayname* parece identificar o nome do computador.

Linha 5 - parâmetro *port* identifica a porta usada.

Linha 6 - parâmetro *namespaces* contém um conjunto de números inteiros e neste

momento não é identificado qual a sua função, sabe-se que o número de elementos deste conjunto varia.

So conjunto de parâmetros que foram identificados nas mensagens deste protocolo no tráfego capturado, nenhum contém informação útil que possa ser usada para filtrar os endereços MAC das estações sem fios que se encontrem na vizinhança.

NetBIOS (Network Basic Input/Output System)

O NetBIOS (Network Basic Input/Output System) é uma *interface* que corre na camada de sessão que permite a comunicação entre máquinas na mesma rede local usando pacotes NBF (NetBIOS Frames protocol). Actualmente, este corre sobre TCP/IP e usa o protocolo NBT (NetBIOS over TCP/IP) para efectuar as trocas dos pacotes. O NBT será o protocolo alvo deste estudo.

Cada computador na rede, além do endereço IP correspondente, tem um nome NetBIOS.

O protocolo NBT está descrito nos documentos RFC 1001[37] e RFC 1002[38].

Este fornece os seguintes serviços na rede:

- Registo e resolução do nome (NBNS, NetBIOS Naming Service), funciona na porta 137 em UDP.
- Comunicação orientada à ligação (NBSS, NetBIOS Session Service), funciona na porta 139 em TCP.
- Comunicação não orientada à ligação (NBDD, NetBIOS Datagram Distribution), funciona na porta 138 em UDP.

O serviço NBSS é para sessões de comunicação usando o TCP. Nas capturas que foram realizadas, os pacotes TCP foram filtrados por razões explicadas anteriormente, logo este serviço não será alvo de estudo, contrariamente aos restantes serviços.

NBNS - cada nó NetBIOS na rede tem que registar pelo menos um nome único mas pode ter mais nomes registados. Pode também registar um nome de grupo e que pode pertencer a vários nós. Cada nó mantém informação de estado de cada nome registado guardando:

- Nome quer seja único ou de grupo.

- Se o nome se encontra em conflito.
- Se o nome se encontra em processo de ser apagado.

Os nós para efectuar o registo do nome enviam em *broadcast* uma mensagem a clamar o nome e a solicitar defesa do mesmo caso já exista.

NBDD - é o serviço para troca de pacotes não orientado à ligação. Cada pacote contém um nome NetBIOS de origem e outro de destino. São efectuadas pesquisas para aprender o endereço IP e os atributos do nome NetBIOS de destino. Estes são transportados em UDP e podem ser fragmentados. Um pacote NetBIOS pode ser enviado em *unicast*, *broadcast* e *multicast*. No caso do tráfego capturado, todos os pacotes foram enviados em *broadcast*, ou seja, foram enviados usando a primitiva *Send Broadcast Datagram*.

O NBNS usa um formato de mensagens baseado no DNS, descrito no RFC 1035[28], no entanto, a especificação do NetBIOS acrescenta novos tipos e códigos. Na Tabela 3.13, já referenciada anteriormente, está ilustrado o formato geral de uma mensagem DNS. Na Figura 3.20, também já referenciada anteriormente, está ilustrado o respectivo cabeçalho onde a única modificação introduzida pelo NetBIOS é o campo *Z* que está agora ilustrado na Figura 3.22.

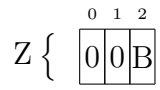


Figura 3.22: Modificação pelo NBNS no cabeçalho DNS.

Os dois primeiros bits ficam a "0" e o bit *B*, se for "0", a mensagem é enviada em *unicast*, se for "1", é enviado em *broadcast* ou *multicast*. Nas mensagens NetBIOS encontradas no tráfego capturado o bit *B* é sempre "1", o que já era de esperar uma vez que o tráfego em *unicast* não tem interesse e foi logo filtrado.

A secção *Question* também é idêntica à que foi analisada no MDNS, ilustrada na Figura 3.19, no entanto, os campos que se seguem tem um significado diferente:

QNAME - nome NetBIOS comprimido para o pedido.

QTYPE - tipo de pedido, que pode ser:

- *NB*, RR geral do NBNS.

- *NBSTAT*, estado do RR do nó NetBIOS.

Os RR também são usados, assim como no MDNS, que está ilustrado na Figura 3.18. O seu formato é também idêntico, no entanto os campos que se seguem também têm um significado diferente.

NAME - nome NetBIOS comprimido correspondente ao RR.

TYPE - código de tipo de RR NetBIOS que pode ser:

- *A*, endereço IP do RR.
- *NS*, nome de servidor do RR.
- *NULL*, RR vazio.
- *NB*, RR geral do NBNS.
- *NBSTAT*, estado do RR do nó NetBIOS.

RDATA - depende dos campos *CLASS* e *TYPE* e contém a informação de recurso para o nome NetBIOS.

Quando um RR é do tipo "NB", no campo *RDATA* aparecem dois novos campos, *NB_FLAGS* e *NB_ADDRESS*. O primeiro é um conjunto de campos e está ilustrado na Figura 3.23, o segundo representa o endereço IP que é dono do nome.

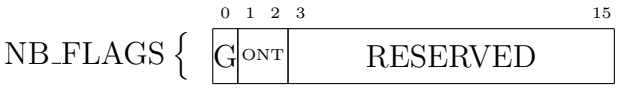


Figura 3.23: Campos de flags de um RR do tipo "NB" em RDATA, no NBNS.

G - indica se o nome NetBIOS é único ou de grupo.

ONT - tipo de nó NetBIOS.

RESERVED - reservado para uso futuro, devem ser colocados a "0".

Depois de identificadas as diferenças introduzidas pelo protocolo NetBIOS no formato de mensagens DNS, foram identificados os tipos de mensagens que foram encontrados no tráfego capturado, relativo ao NBNS, onde foram encontrados os seguintes tipos de mensagens:

- *Name Query Request*, enviado quando se pretende descobrir o endereço IP associado a um nome NetBIOS.
- *Name Registration Request*, enviado quando um nó pretende registar um nome único ou um nome de grupo.
- *Name Release Request*, enviado quando um nó pretende libertar o nome NetBIOS explicitamente.

Na Figura 3.24 está ilustrado uma mensagem do tipo *Name Query Request*, onde são representados campos com valor fixo que dizem respeito ao respectivo tipo de mensagem.

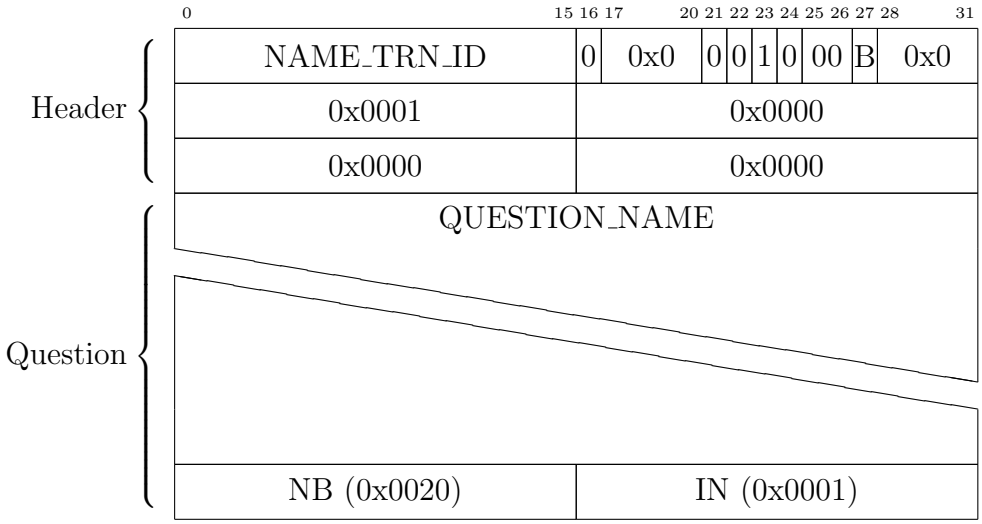


Figura 3.24: Mensagem Name Query Request do NBNS.

O campo *NAME_TRN_ID* representa o identificador da mensagem e varia de mensagem para mensagem, *QUESTION_NAME* é o campo que contém o nome NetBIOS de que se pretende descobrir o endereço IP. O bit *B* estará a "1" para o caso em estudo.

Na Figura 3.25 está ilustrada uma mensagem do tipo *Name Registration Request* onde também são representados campos com valor fixo que dizem respeito ao respectivo tipo de mensagem.

Os campos *QUESTION_NAME* e *RR_NAME* contém o mesmo valor, o nome NetBIOS que se pretende registar. O bit *B* também estará a "1" para o caso em estudo. O campo *TTL* contém o tempo de vida pretendido para o nome NetBIOS,

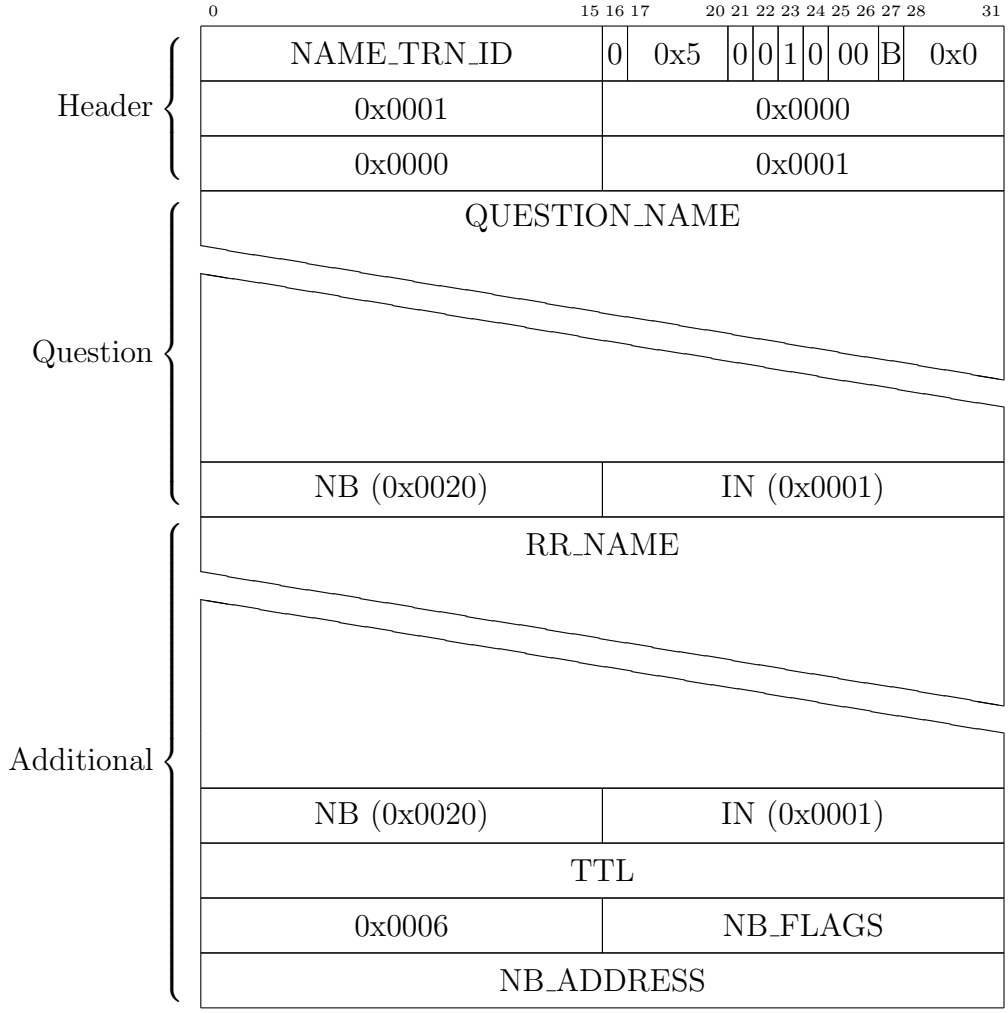


Figura 3.25: Mensagem Name Registration Request do NBNS.

NB_FLAGS o tipo de nó NetBIOS e se representa um nome único ou de grupo e por fim *NB_ADDRESS* que contém o endereço IP do nó associado ao nome NetBIOS. Os campos da secção *Additional* constituem o RR do nome NetBIOS que se pretende registar.

A mensagem do tipo *Name Release Request* é idêntica à anterior que se encontra ilustrada na Figura 3.25, no entanto, existem algumas diferenças. O campo *OPCODE* com o tipo de mensagem toma o valor "0x6" e o campo *TTL* com o tempo de vida do nome NetBIOS é "0".

Nos pacotes NBNS não foi encontrado nenhuma informação útil que possa ser usada para filtram os endereços MAC das estações sem fios que se encontrem na

vizinhança mas ainda falta analisar o formato de pacotes do serviço NBDD.

Na Figura 3.26 está ilustrado o cabeçalho de uma mensagem do NBDD.

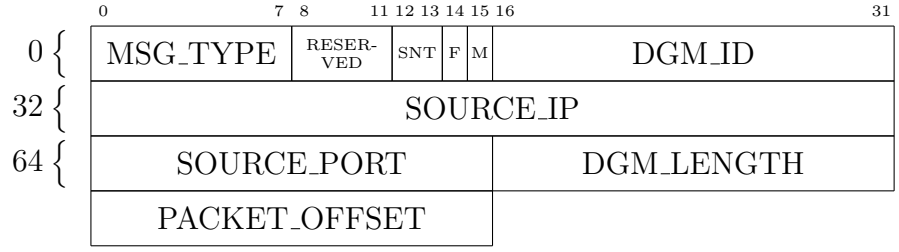


Figura 3.26: Cabeçalho da mensagem NetBIOS Datagram no NBDS.

MSG_TYPE - contém o tipo de mensagem NBDD. No tráfego capturado todas as mensagens foram do mesmo tipo "DIRECT_GROUP DATAGRAM", ou seja, foram enviados para um nome NetBIOS que é de grupo.

RESERVED - reservados, têm que estar a "0".

SNT - tipo de nó NetBIOS de origem.

F - indica que é a primeira mensagem de um possível conjunto de mensagens fragmentadas.

M - indica que virão mais mensagens que estão fragmentadas.

DGM_ID - identificador do datagrama.

SOURCE_IP - endereço IP da estação que enviou a mensagem.

SOURCE_PORT - número de porta da estação que enviou a mensagem.

DGM_LENGTH - tamanho da mensagem em bytes.

PACKET_OFFSET - se a mensagem faz parte de uma mensagem fragmentada, indica qual a parte da mensagem.

Neste serviço NetBIOS, para um pacote enviado em *broadcast* são acrescentados os seguintes campos logo a seguir ao cabeçalho:

SOURCE_NAME - nome NetBIOS do nó que enviou a mensagem.

DESTINATION_NAME - nome NetBIOS do nó a que a mensagem é destinada.

USER_DATA - dados de informação da mensagem.

Os dados de informação que foram encontrados dentro das mensagens capturadas são do protocolo SMB. Este é encapsulado no NetBIOS e também é alvo de estudo.

Nos pacotes do protocolo NBDD também não foi encontrada informação útil que possa ser usada para filtrar os endereços MAC das estações sem fio na vizinhança, no entanto, permitiu detectar um novo protocolo que funciona ao nível de aplicação que vai ser estudado a seguir.

SMB (Server Message Block)

O protocolo SMB também é designado como CIFS (Common Internet File System), uma vez que a Microsoft no lançamento de uma versão melhorada do protocolo mudou o nome ao mesmo. Este funciona ao nível de aplicação numa rede e é maioritariamente usado para partilhar ficheiros, impressoras, etc. Permite autenticação no processo de comunicações inter-processos e é usado principalmente nos SOs Windows.

No tráfego capturado foram detectados outros protocolos que são encapsulados dentro deste, o Browser Protocol e o Remote Mailslot Protocol.

As especificações dos mesmos são encontrados nos documentos MS-MAIL[39], MS-CIFS[40], MS-SMB[41] e MS-BRWS[42].

Nas mensagens deste protocolo encontra-se primeiro o cabeçalho SMB, que está ilustrado na Figura 3.27, e os campos que o constituem são:

PROTOCOL - identifica o protocolo, primeiro byte a "0xff" e os restantes "SMB".

COMMAND - código do comando SMB. No tráfego capturado apenas foi detectado um tipo de comando, o *SMB_COM_TRANSACTION* ("0x25").

STATUS - campo usado para comunicar erros do servidor para o cliente.

FLAGS - campo com um conjunto de flags de 1 bit. O primeiro bit deste campo só tem interesse para alguns comandos SMB e nenhum desses comandos foi encontrado nos pacotes do tráfego capturado. O último bit indica se a mensagem é um pedido ou resposta e os restantes bits estão obsoletos.

FLAGS2 - campos com um conjunto de flags de 1 bit. No tráfego capturado estes bits estão todos a "0".

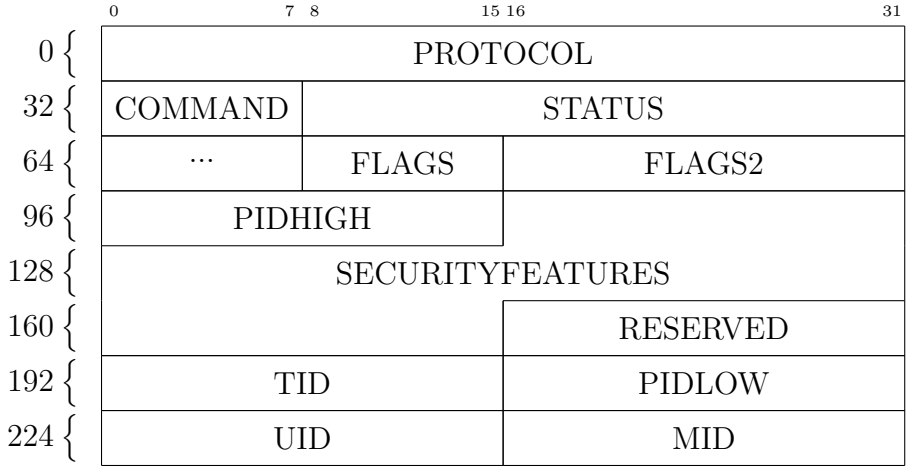


Figura 3.27: Cabeçalho SMB.

PIDHIGH - contém os bytes de maior ordem que formam o identificador do processo (PID, Process identifier), este é combinado com o campo *PIDLOW*.

SECURITYFEATURES - contém uma assinatura de segurança quando existe negociação de assinaturas de segurança.

RESERVED - reservado e tem que estar a zeros.

TID - identificador de árvore.

PIDLOW - contém os bytes de menor ordem do PID.

UID - identificador de utilizador.

MID - identificador de multiplexador.

Depois do cabeçalho encontram-se os campos que constituem o tipo de comando SMB, que estão ilustrados na Figura 3.28 para o único tipo de comando que foi identificado no tráfego capturado. O tipo é designado *Transaction* ("0x25") e no campo *flags* observa-se que não se trata de uma mensagem de resposta, mas sim de um pedido do cliente ao servidor. Este transporta uma comunicação inter-processos que opera em *Mailslots* entre dois *endpoints*.

Os campos que constituem o comando são:

TOTALPARAMETERCOUNT - número total de bytes com parâmetros de transacção que o cliente espera enviar no pedido.

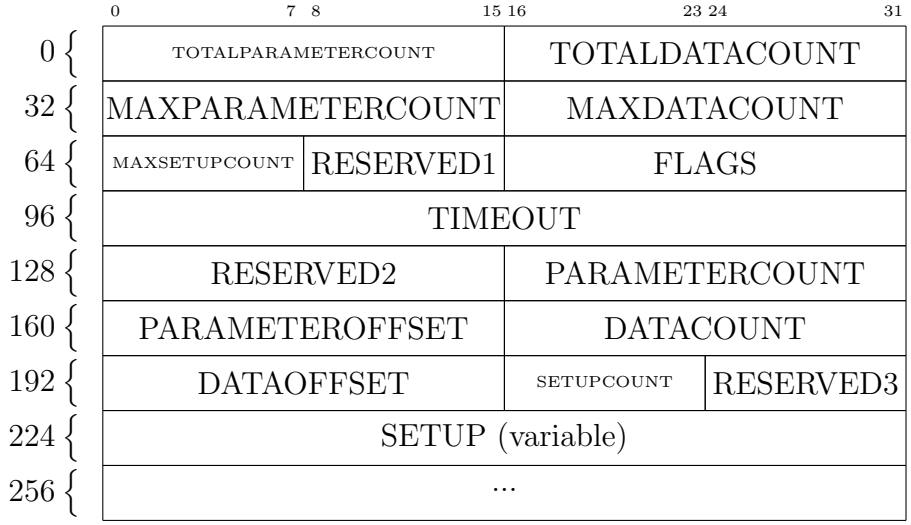


Figura 3.28: Comando SMB_COM_TRANSACTION (0x25) SMB.

TOTALDATACOUNT - número total de bytes com dados de transacção que o cliente tenta enviar no pedido.

MAXPARAMETERCOUNT - número máximo de bytes com parâmetros transacção que o cliente aceita na resposta.

MAXDATACOUNT - número total de bytes com dados de transacção que o cliente aceita na resposta.

MAXSETUPCOUNT - número máximo de bytes que o cliente aceita no campo *SETUP* na resposta.

RESERVED1 - byte com *padding*.

FLAGS - campo com *flags*, estas indicam se o servidor deve responder à transacção e se deve ou não desligar da árvore que vem no cabeçalho SMB.

TIMEOUT - tempo em milisegundos máximo que o servidor espera até a transacção ficar completa.

RESERVED2 - reservado, estes bits devem ser enviados a zero.

PARAMETERCOUNT - número de bytes de parâmetros de transacção que o cliente tenta enviar para o servidor neste pedido.

PARAMETEROFFSET - número de bytes que existe entre o início do cabeçalho SMB até ao início dos dados dos parâmetros de transacção.

DATACOUNT - número de bytes de dados de transacção que o cliente envia para o servidor neste pedido.

DATAOFFSET - número de bytes que existe entre o início do cabeçalho SMB até ao início dos dados de transacção.

SETUPCOUNT - número de palavras de configuração que são incluídas no pedido.

RESERVED3 - byte com *padding*.

SETUP - este campo inclui as palavras de configuração. Começa pelo número de bytes das palavras e depois as respectivas palavras. No caso dos pacotes capturados, este campo inclui a indicação que a comunicação inter-processos entre o cliente e servidor é efectuada usando o protocolo Mailslot.

O protocolo Mailslot consiste em criar no servidor um ficheiro em que os dados podem ser escritos por várias estações na mesma rede.

Na Figura 3.29 está ilustrado o formato da mensagem Mailslot, que é enviada dentro das palavras de configuração da mensagem anterior.

Os campos que constituem a mensagem são:

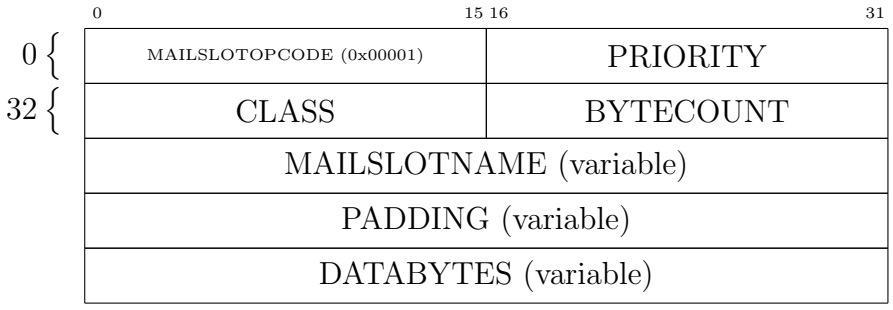


Figura 3.29: Mailslot Write Message do SMB.

MAILSLOTOPCODE - representa o código a indicar que é uma mensagem *Mailslot Write Message*.

PRIORITY - valor numérico que representa a prioridade da mensagem. Quanto maior este número, mais prioritária é a mensagem.

CLASS - classe da mensagem Mailslot. Existem duas classes, *first-class* que oferece garantias de entrega da mensagem e *second-class* que não oferece garantias de entrega da mensagem. No caso das mensagens do tráfego capturado são *second-class* uma vez que são mensagens enviadas em *broadcast* e assim nunca pode ser garantido a entrega da mensagem.

BYTECOUNT - número de bytes que se seguem a este campo.

MAILSLOTNAME - nome do Mailslot para onde a mensagem está a ser enviada.

PADDING - *padding* para alinhar o campo *DATABYTES* a 32 bits.

DATABYTES - mensagem Mailslot para ser entregue ao servidor.

Dentro das mensagens Mailslot foi detectado o protocolo CIFS Browser Protocol. Este protocolo define mensagens que são enviadas e recebidas por um servidor que funciona como um colector e distribuidor de serviços disponíveis na rede, como a partilha de ficheiros e impressoras. Os clientes acedem ao mesmo para obter informação de um serviço em particular.

Os tipos de mensagens deste protocolo que foram detectadas no tráfego capturado serão alvo de um estudo, no entanto, existem mais tipos na especificação do protocolo que não vão ser estudados.

O primeiro tipo de mensagem encontrado foi *BecomeBackup Browser* que está ilustrado na Figura 3.30. Esta mensagem é usada quando um servidor mestre pretende eleger um potencial servidor mestre de reserva.

Em todas as mensagens do protocolo o primeiro campo *COMMAND* contém o código que identifica o tipo de mensagem. A descrição deste campo será omitida na descrição dos campos presentes nas mensagens.

Os campos que constituem a mensagem *BecomeBackup Browser* são:

Figura 3.30: Mensagem BecomeBackup Browser do SMB.

BROWSERTOPROMOTE - nome NetBIOS que é eleito servidor de reserva.

O tipo de mensagem encontrado a seguir foi *RequestElection Browser* que está ilustrado na Figura 3.31. Esta é usada para começar a eleição de um novo servidor mestre. Isto acontece quando quando existe falha do servidor mestre e também na obtenção da lista de servidores de reserva por parte de um cliente. A mensagem é enviada em *broadcast* para o nome NetBIOS de grupo, todas as estações no grupo devem responder com uma mensagem do mesmo tipo. Os campos que constituem a mensagem são:

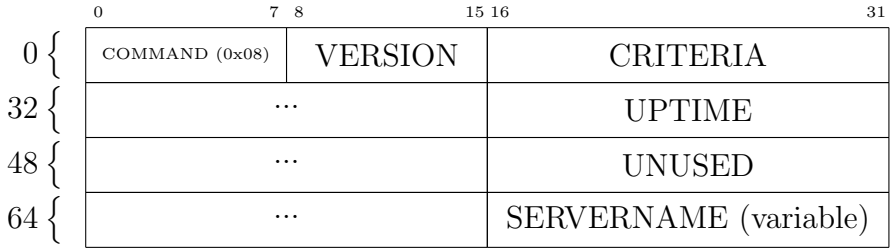


Figura 3.31: Mensagem RequestElection Browser do SMB.

VERSION - versão da eleição da mensagem, enviado com o valor "0x01".

CRITERIA - critério para eleição de quem envia a mensagem, no caso de um cliente é "0". Um servidor alternativo envia neste campo uma combinação do SO, versões do protocolo suportadas e papel pretendido.

UPTIME - tempo em segundos desde que o serviço se encontra activo.

UNUSED - não usado, tem que estar a zeros.

SERVERNAME - nome de quem envia a mensagem.

Depois foi encontrada a mensagem do tipo *DomainAnnouncement Browser*, que está ilustrada na Figura 3.32. Esta é usada por um servidor mestre para anunciar aos outros servidores mestre o grupo ou domínio que servem. A mensagem é enviada em *broadcast* para a sua subrede. Os campos que constituem a mensagem são:

UPDATECOUNT - enviado a "0x00", tem que ser ignorado.

PERIODICITY - periodicidade com que são enviados estes anúncios em milisegundos.

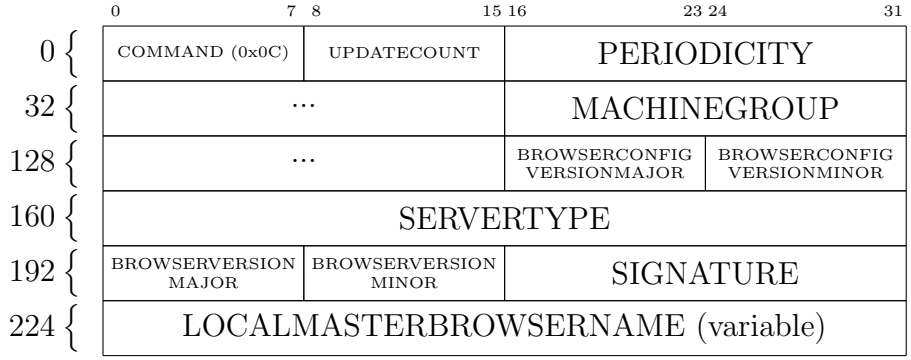


Figura 3.32: Mensagem DomainAnnouncement Browser do SMB.

MACHINEGROUP - nome do grupo de trabalho ou nome NetBIOS do domínio.

BROWSERCONFIG VERSIONMAJOR - versão maior do protocolo a correr no servidor.

BROWSERCONFIG VERSIONMINOR - versão mais baixa do protocolo a correr no servidor.

SERVERTYPE - tipo de servidor.

BROWSERVERSION MAJOR - este campo deve ter valor "0x0F".

BROWSERVERSION MINOR - este campo deve ter valor "0x01".

SIGNATURE - este campo deve ter valor "0xAA55".

LOCALMASTERBROWSERNAME - nome do servidor que envia a mensagem.

Depois foi encontrada a mensagem do tipo *GetBackupListRequest Browser* que está ilustrada na Figura 3.33. Esta é usada por um cliente que pretende obter uma lista de servidores de reserva. A resposta é enviada ao cliente pelo servidor mestre. Os campos que constituem a mensagem são:

REQUESTCOUNT - número de servidores de reserva que o cliente pede.

TOKEN - o conteúdo exacto deste campo terá que ser retornado na mensagem de resposta.

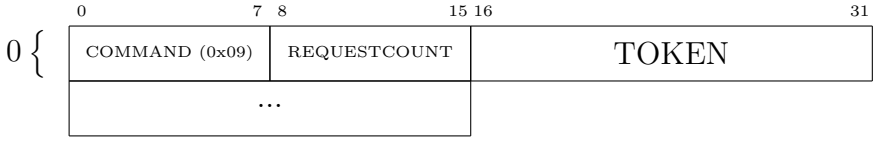


Figura 3.33: Mensagem GetBackupListRequest Browser do SMB.

Depois foi encontrado a mensagem do tipo *HostAnnouncement Browser* que está ilustrado na Figura 3.34. Esta é usada quando um servidor pretende anunciar a sua presença e os tipos de recursos e serviços que suporta. A mensagem só é enviada quando existe um pedido ou quando expira o temporizador da mesma. Os campos que constituem a mensagem são:

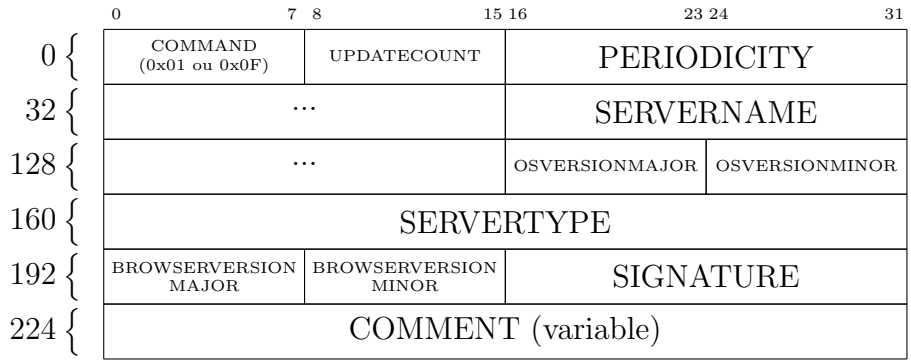


Figura 3.34: Mensagens HostAnnouncement Browser e LocalMasterAnnouncement Browser do SMB.

UPDATECOUNT - enviado a "0x00", tem que ser ignorado.

PERIODICITY - periodicidade com que são enviados estes anúncios em milisegundos.

SERVERNAME - nome do servidor que faz o anúncio.

OSVERSIONMAJOR - versão maior do SO que corre no servidor.

OSVERSIONMINOR - versão mais baixa do SO que corre no servidor.

SERVERTYPE - tipo de servidor.

BROWSERVERSION MAJOR - versão maior do CIFS Browser Protocol a correr no servidor.

BROWSERVERSION MINOR - versão mais baixa do CIFS Browser Protocol a correr no servidor.

SIGNATURE - este campo deve ter valor "0xAA55".

COMMENT - comentário associado ao servidor, é apenas um campo informal.

O penúltimo tipo de mensagem encontrado foi *LocalMasterAnnouncement Browser*. Este é usado por um servidor mestre quando pretende periodicamente anunciar a sua presença na rede a todos os servidores na mesma subrede. O formato desta mensagem é idêntico ao tipo anterior que foi ilustrado na Figura 3.34. O único campo modificado é o *COMMAND* que muda o seu valor para "0x0F".

O último tipo de mensagem encontrado foi *AnnouncementRequest Browser* que está ilustrado na Figura 3.35. Este é enviado de um nome único NetBIOS para um nome de grupo NetBIOS para forçar todas as estações, ou apenas o servidor mestre, no grupo de trabalho ou domínio para se anunciarem ao cliente. Os campos que constituem a mensagem são:

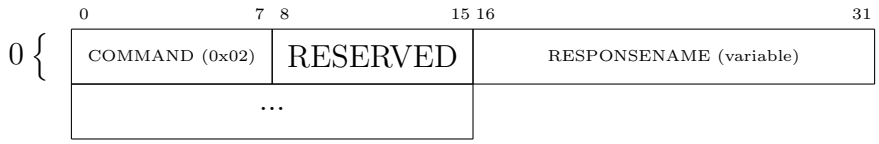


Figura 3.35: Announcement Request Browser do SMB.

RESERVED - reservado, enviado a zeros.

RESPONSENAME - nome de quem envia o pedido.

Com este conjunto de protocolos pertencentes ao SMB e usando os tipos de mensagem que foram detectados no tráfego capturado também não foi obtida nenhuma informação particularmente útil para este projecto que permita das estações todas detectadas no tráfego capturado, filtrar os endereços MAC das estações sem fios que se encontram presentes na vizinhança.

3.2.6 Conclusão da análise de tráfego

No decorrer da análise do tráfego, que foi capturado ao nível Ethernet em diversos locais, foram detectados com sucesso um grande número de endereços MAC que se

encontravam activos nas respectivas redes. Estes pertencem às placas de rede de estações que transmitiram pacotes em *broadcast* e *multicast* para a rede local através da ligação à rede que tinham activa. No entanto, para determinada estação detectada, não sabemos se:

- está ou não na vizinhança da estação que efectuou a captura do tráfego.
- o tipo de ligação usada para se ligar à rede, Ethernet ou *WiFi*.

Sem essa informação só podemos afirmar que as estações detectadas estão todas na mesma rede local.

De forma a filtrar os endereços MAC que se encontram na vizinhança e que estão ligados na rede usando uma ligação *WiFi*, foi elaborado um estudo dos protocolos encontrados no tráfego capturado que são usados com mais frequência, no entanto, este estudo não teve sucesso, porque não foi encontrada informação útil nos protocolos que possa ser usada para o pretendido.

Esta solução fica a um passo do pretendido, no entanto, foi abandonada para uma nova solução que passa pela análise de pacotes a um nível mais baixo na camada de ligação.

3.3 Captura de tráfego na camada MAC do 802.11

Nesta altura sabe-se que os pacotes obtidos através da captura do tráfego ao nível da camada Ethernet e que são transmitidos pelas estações em *multicast* e *broadcast* contém protocolos que vêm encapsulados para as diferentes camadas superiores de rede. Estes protocolos são usados de raiz nos diferentes SOs e através da informação que contêm permite detectar as estações presentes na rede, no entanto, as estações fisicamente podem-se encontrar a uma grande distância e não partilham o mesmo meio para acesso a rede, neste caso *WiFi*, logo, não podem ser consideradas vizinhas da estação que realiza a captura e que queremos que seja o radar para estações *WiFi*.

Assim, a captura de tráfego terá que de algum modo ser efectuada na camada MAC do 802.11, por onde passam as tramas 802.11 do tipo dados, controlo e gestão. Nesta camada todas as tramas ainda incluem os cabeçalhos 802.11.

As tramas do tipo controlo e gestão do 802.11 não são entregues às camadas de rede superiores, uma vez que apenas transportam informação relativa ao controlo e gestão do acesso ao meio 802.11. As tramas do tipo dados contém informação

para ser entregue às camadas superiores de rede, uma vez que transportam outros protocolos encapsulados como é o caso do IP. Ao passar os dados para a camada superior, alguns campos de informação presentes no cabeçalho 802.11 são copiados para um novo cabeçalho, como é o caso dos campos de endereçamento. Usando esses endereços é construído um novo cabeçalho, um "falso" cabeçalho Ethernet, e recalculado o *Checksum* que vai encapsular os dados para a camada superior.

Existe uma sub-camada que pertence à camada de ligação para efectuar esta função, conhecida como LLC e que está descrita na norma IEEE 802.2. O LLC é responsável por fazer a ponte entre as diferentes tecnologias de acesso ao meio e o respectivo meio físico para a camada superior de rede, assim permitindo a uniformização para as camadas superiores.

Existem duas especificações para fazer esta ponte que derivam do 802.2's SNAP, o 802.1H e o RFC 1042[4]. Nos SOs Windows foi predefinido o uso da especificação 802.1H para os protocolos AppleTalk e IPX e o RFC 1042 nos restantes. Sendo as redes TCP/IP mais comuns hoje em dia, a forma como a ponte é efectuada segundo a especificação do RFC 1042 tem mais interesse no âmbito deste projecto e por isso está ilustrada na Figura 3.36[1] a forma como a ponte é efectuada.

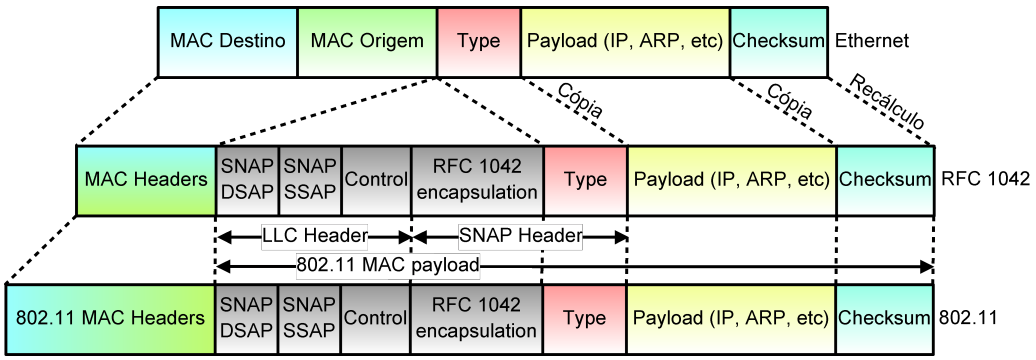


Figura 3.36: Ponte entre 802.11 e Ethernet usando o RFC 1042.

Nessa ponte é criado um cabeçalho que é dividido em dois grupos de campos. O cabeçalho LLC com o destino (DSAP) e origem (SSAP) do *service access point*, um campo de controlo que é definido com um valor decimal "3" e que representa *Unnumbered Information*. O segundo cabeçalho não é mais do que uma extensão do primeiro e é constituído pelos campos *RFC 1042 Encapsulation* que são postos a zeros, indicando também que o campo *Type* é uma copia do campo *EtherType* da trama Ethernet.

Percebe-se melhor agora o porquê de nas capturas de tráfego efectuadas e analisadas na secção 3.2 só se ter capturado pacotes Ethernet 802.3, mesmo com a captura a ser realizada numa placa de rede sem fios que abstraiu os cabeçalhos das tramas do 802.11. O mecanismo de captura do Winpcap, usado na solução anterior, não desce a um nível mais baixo que este.

A arquitectura do 802.11 é constituída por um conjunto de componentes que formam uma "caixa" que está inserida na camada de ligação e na camada física, como está ilustrado na Figura 3.37.

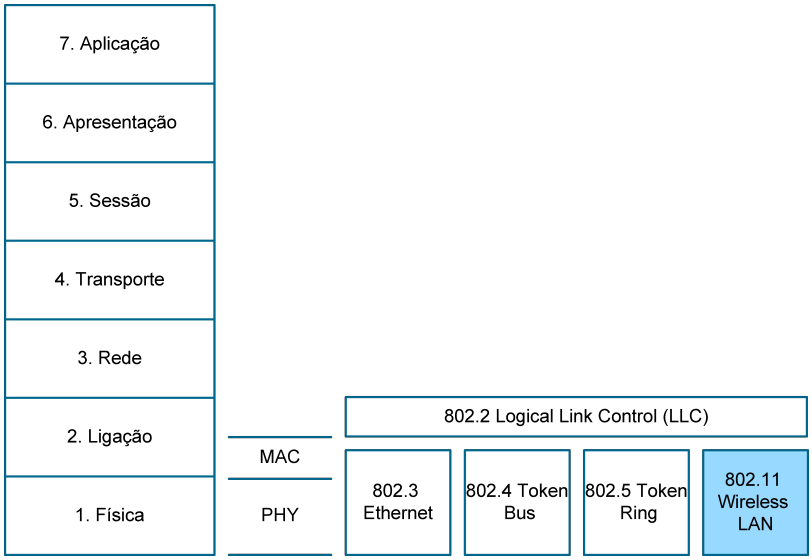


Figura 3.37: Modelo OSI para o 802.11.

Essas componentes são as diversas sub camadas físicas (802.11b, 802.11g, etc), que pertencem à camada física e à sub-camada MAC 802.11, que pertence à camada de ligação. A captura terá que ser efectuada num desses componentes do 802.11 ilustrados na Figura 3.38.

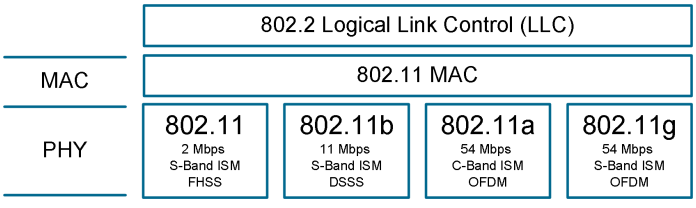


Figura 3.38: Componentes do 802.11.

Como já foi estudado na norma 802.11, sabe-se que as tramas que chegam à sub camada MAC são as que têm interesse para a captura, uma vez que não trazem já o

preâmbulo para aquisição de sinal e treino e sincronização do receptor, assim como o cabeçalho da camada física. A trama genérica 802.11 entregue à sub camada MAC tem o formato que está ilustrado na Figura 3.39.

Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control
2 bytes	2	6	6	6	2

Address 4	QoS Control	HT Control	Frame Body	FCS
6	2	4	0-7955	4

Figura 3.39: Trama MAC genérica do 802.11.

A trama contém toda a informação de endereçamento MAC e, com a sua captura, podemos extrair a informação que permite detectar as estações presentes na vizinhança. Com o campo *Frame Control* (Figura 2.6) consegue-se obter a seguinte informação que tem interesse para a solução:

- Tipo e Sub-Tipo de trama 802.11 com a qual se está a lidar (Tabela 2.4).
- Através dos bits *To DS* e *From DS* consegue-se saber se a trama de dados tem origem ou é destinada a um DiS numa Infrastructure BSS ou se pertence a uma IBSS (Tabela 2.5).

Com a informação do *Frame Control* sabe-se exactamente quais os endereços MAC a extrair segundo a Tabela 2.7.

Na solução anterior, foi usado apenas o endereço MAC de origem para detectar uma estação ao nível Ethernet mas agora neste cenário é diferente. Os cabeçalhos 802.11 podem trazer até 4 campos de endereços MAC que dizem respeito a ligações diferentes:

- ligação entre uma estação e o AP ao qual se encontra associada.
- ligação lógica entre uma estação e a rede local a que pertence, em que a ligação passa pelo AP ao qual se encontra associada.
- ligação entre diversas estações ligadas numa IBSS.
- ligação entre APs, que formam uma ponte.

Para cada um destes casos será necessário um processamento diferente dos endereços, segundo a Tabela 2.7 de endereços do 802.11, para os diferentes valores dos campos *To DS* e *From DS*.

Nas tramas do tipo dados os endereços MAC das estações presentes na vizinhança são detectadas e identificadas pelo endereço de receptor/destino (RA=DA) ou do endereço de origem (SA), caso a trama seja enviada do AP para a estação ($DS = 01$)[1] ou da estação para o AP ($DS = 10$). No caso de uma IBSS são vizinhos tanto o endereço de origem como o receptor/destino ($DS = 00$). Na ligação entre APs ($DS = 11$) nenhum dos 4 endereços é considerado vizinho, uma vez que há 2 endereços de BSSIDs e os outros 2 são endereços da ligação lógica para a rede local.

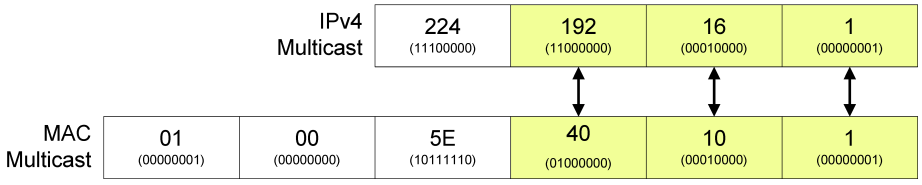
Dos endereços MAC, alguns terão de ser filtrados:

- Endereços MAC enviados para o grupo *broadcast*.
- Endereços MAC de grupos *multicast*, que tomam uma gama de endereços MAC diferentes para IPv4 e IPv6, sendo que para IPv4 os 23 bits de baixa ordem do endereço IP são copiados para o endereço MAC com um prefixo fixo (Figura 3.40a), segundo o RFC 1112[20]. Para IPv6 isto também acontece mas são os 32 bits de baixa ordem do endereço IPv6 que são copiados para o endereço MAC com outro prefixo fixo (Figura 3.40b), segundo o RFC 2464[43].
- Endereços MAC de protocolos específicos que usam endereços MAC de grupo *multicast* reservados e diferentes dos anteriores.

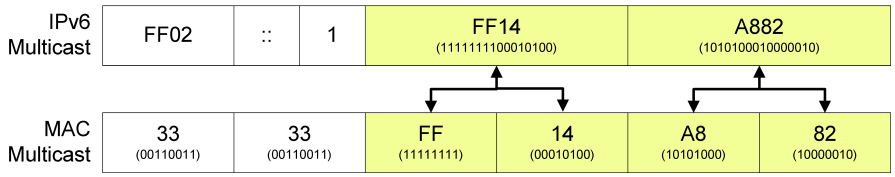
A lista completa de endereços MAC a ser filtrados pode ser muito grande se forem considerados individualmente, o que torna muito difícil a sua implementação, quer em recursos, quer na própria construção da lista, logo procedeu-se a um estudo para perceber os endereços MAC. Estes são geridos pelo IEEE e são usados nas tecnologias de rede mais comuns como todas as redes 802.

Os endereços MAC são conhecidos como MAC-48 por serem constituídos por 48 bits. Estes nem sempre representam um endereço de *hardware* físico escrito nas placas de rede. Podem também representar outros dispositivos e *software*, sendo neste caso designado como EUI-48 (EUI, Extended Unique Identifier). Este não se distingue do anterior em sintaxe e usa o mesmo espaço de endereçamento. Os endereços MAC podem ser divididos em vários tipos:

[1] DS é constituído pelos bits dos dos campos *To DS* e *From DS*



(a) IPv4 *multicast*



(b) IPv6 *multicast*

Figura 3.40: Conversão endereços IP *multicast* para MAC *multicast*.

universally administered addresses - são os endereços gravados pelos fabricantes nas placas de rede que produzem. Em cada endereço MAC deste tipo, os primeiros 24 bits do endereço identificam o fabricante (OUI)[1], os restantes 24 bits são geridos pelo fabricante. Nesta solução, apenas os endereços deste tipo são considerados válidos como endereços de estações vizinhas, porque representam o *hardware* da placa de rede de uma estação.

locally administered addresses - são endereços atribuídos por administradores de rede e não contêm OUI. No âmbito desta solução, segundo a especificação 802.11, uma IBSS tem como BSSID um endereço deste tipo escolhido aleatoriamente quando é criada a rede.

A distinção entre os dois tipos de endereços é feita num único bit. Esse é o bit $b2$ do *byte*6 (mais significativo) do endereço MAC, como mostra a Figura 3.41. O bit a "0" representa um endereço do tipo *universally administered* enquanto que a "1" representa um endereço do tipo *locally administered*.

Num endereço MAC existe também o bit $b1$ do *byte*6 (mais significativo) que permite distinguir se um endereço é *unicast* ou *multicast* (Figura 3.41). Este bit a "0" indica que é um endereço *unicast* e representa apenas uma estação, se estiver a "1" indica que é um endereço *multicast* e representa um grupo de estações.

Com esta informação sobre os endereços MAC-48, pode-se agora filtrar os endereços MAC olhando só para alguns bits específicos dos 48 bits do endereço e assim

[1]Número de 24 bits único comprado ao IEEE que identifica um fabricante universalmente.

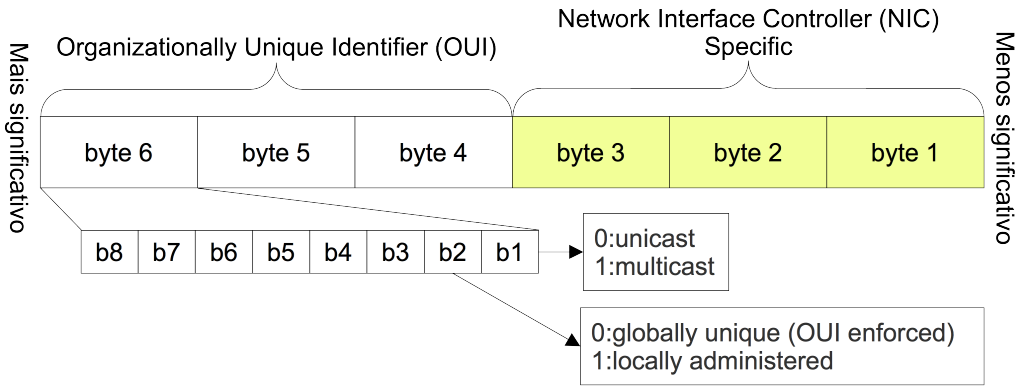


Figura 3.41: Endereço MAC-48.

saber se se trata de um endereço real gravado numa placa de rede ou de um endereço administrado localmente, bem como se o endereço é *multicast* ou *unicast*. A filtragem será feita de forma a que na lista de estações vizinhas apenas se obtenha os endereços de placas de rede reais.

Das tramas 802.11 são extraídos e usados os endereços MAC de origem e destino, no entanto existem endereços que só podem aparecer como endereço de destino. Os endereços *multicast* e *broadcast* são usados para enviar pacotes destinados a um grupo de estações, logo só são considerados válidos quando usados no endereço de destino. Os endereços *universally administered* e *locally administered* são considerados válidos tanto como endereço de origem como de destino, uma vez que são *unicast*.

Os endereços MAC são normalmente representados em 6 conjuntos de dois caracteres cada em hexadecimal e separados pelo sinal de dois pontos entre conjuntos, assim, de uma forma que torne mais fácil identificar os diferentes tipos de endereços MAC, foi construída a Tabela 3.14, onde o asterisco representa um qualquer caracter em hexadecimal de 0 a F.

A gama de endereços MAC *globally unique* representam os endereços que são gravados nas placas de rede das estações, logo os endereços das estações sem fios vizinhas estão dentro desta gama. Todas as outras gamas de endereços não representam estações na vizinhança e são filtrados. Os endereços MAC na gama *locally administered* vão ser úteis para identificar se determinado BSSID de uma rede pertence a uma IBSS.

Esta solução baseia-se em princípios diferentes da anterior uma vez que o meio sem fios funciona sobre um princípio de *broadcast*. Um AP quando tem uma trama para entregar a uma estação envia-a pelo meio sem fios, que é partilhado por todas

Tipo Endereço	Gama de Endereços
Globally Unique	*0:**:**:**:**:**
	*4:**:**:**:**:**
	*8:**:**:**:**:**
	*C:**:**:**:**:**
Locally Administered	*2:**:**:**:**:**
	*6:**:**:**:**:**
	*A:**:**:**:**:**
	*E:**:**:**:**:**
Multicast	*1:**:**:**:**:**
	*3:**:**:**:**:**
	*5:**:**:**:**:**
	*7:**:**:**:**:**
	*9:**:**:**:**:**
	*B:**:**:**:**:**
	*D:**:**:**:**:**
	*F:**:**:**:**:**
Broadcast	FF:FF:FF:FF:FF:FF
Nulo	00:00:00:00:00:00

Tabela 3.14: Tipos de endereços MAC

as estações na vizinhança, e todas as estações serão capazes de receber essa trama, se tiverem sintonizadas no mesmo canal. No entanto, nas placas de rede sem fios, existem mecanismos que ignoram as tramas que não lhe são destinadas. Para o caso da placa de rede sem fios se encontrar em *Monitor Mode*, é realizada a captura de todos os pacotes recebidos num determinado canal, incluindo os pacotes recebidos com erros. Mas este modo não tem interesse, uma vez que viola as restrições que foram impostas para este projecto.

As tramas 802.11 do tipo controlo fornecem funcionalidades que assistem à entrega de tramas de dados entre estações, de forma a manter um fluxo de dados controlado. As tramas 802.11 do tipo gestão fornecem os serviços de descoberta, associação e autenticação, e assim estabelecem a ligação inicial entre as estações e os APs. Estes tipos de tramas também podem ser usados para identificar as estações que se encontram na vizinhança, não sendo necessária nenhuma filtragem nos endereços MAC de grupo como nas tramas de dados uma vez que não transportam protocolos para camadas superiores, apenas informação para a camada de acesso ao meio 802.11 e relativa ao meio de acesso, que as estações que se encontrem na vizinhança partilham. A informação vai em tramas que têm como destino as estações já autenticadas e associadas, ou que se encontram a realizar este processo no mesmo AP ao qual a estação de captura está também autenticada e associada. Outras tramas são respos-

tas destinadas a estações que se encontrem na vizinhança, que não estão autenticadas e associadas no mesmo AP, mas fizeram de uma descoberta activa de redes 802.11 na sua vizinhança.

Com estas tramas MAC do 802.11 e toda a informação de endereçamento presentes nas mesmas, consegue-se extrair os endereços MAC das estações sem fios presentes na vizinhança. Este processo não quebra a ligação existente entre a estação que realiza a captura e o AP ao qual está associada, permanecendo assim transparente para quem está a usar essa estação. A área que é coberta pela descoberta de estações *WiFi* não é a área de cobertura da estação que realiza a descoberta, mas sim a área de cobertura do AP, ao qual a estação se encontra associada. Este cenário está ilustrado na Figura 3.42.

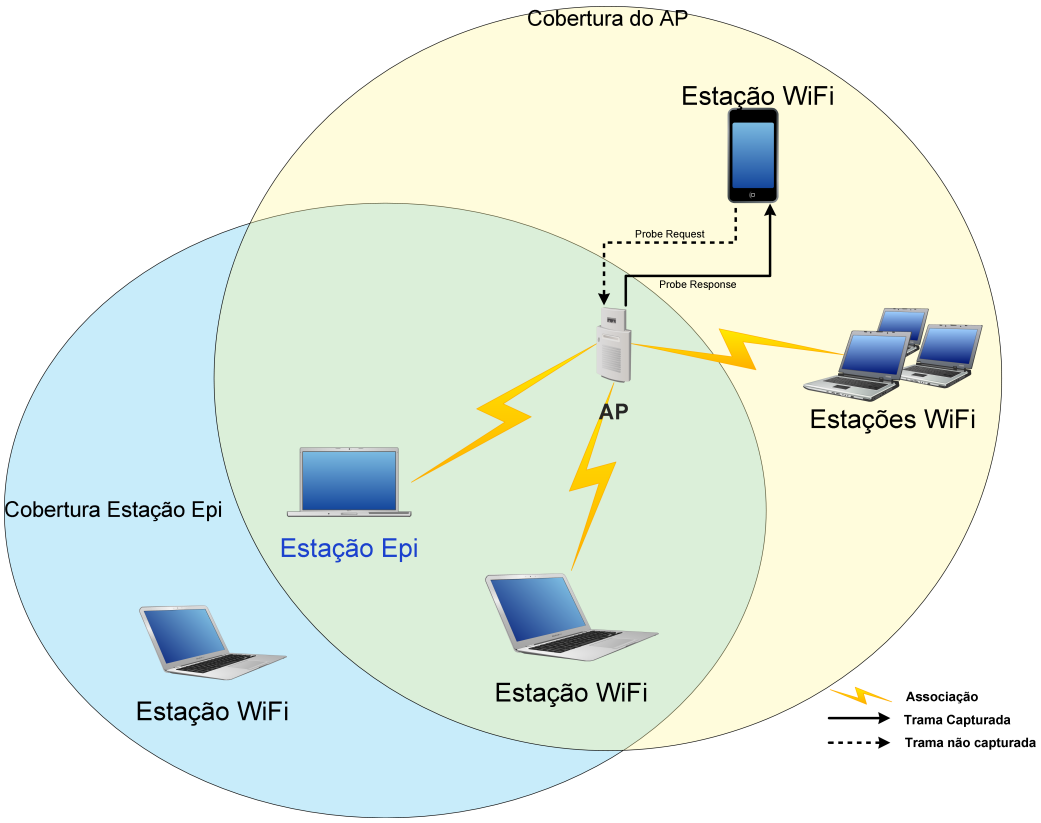


Figura 3.42: Cobertura na detecção de estações.

Isto verifica-se porque na estação que faz a captura apenas são capturadas as tramas que vão no sentido AP para as estações. Para as tramas do tipo dados apenas são capturadas as que são destinadas à própria estação de captura. Existem várias razões porque isto acontece, os mecanismos que ignoram tramas que não lhe

são destinadas, implementados na placa de rede sem fios ou no controlador. Os mecanismos de gestão de energia, que colocam o transmissor de rádio a dormir, acordando apenas periodicamente, isto quando a placa se encontra a funcionar em modo normal.

Na recepção de tramas do tipo controlo e gestão na estação de captura, os mecanismos funcionam de forma diferente. Mesmo as tramas que não tenham como destino a estação de captura normalmente são recebidos e capturados, permitindo assim detectar na mesma as estações que se encontrem na vizinhança do AP ao qual a estação que realiza a captura também se encontra associada. Na Figura 3.43 está ilustrado um cenário de exemplo.

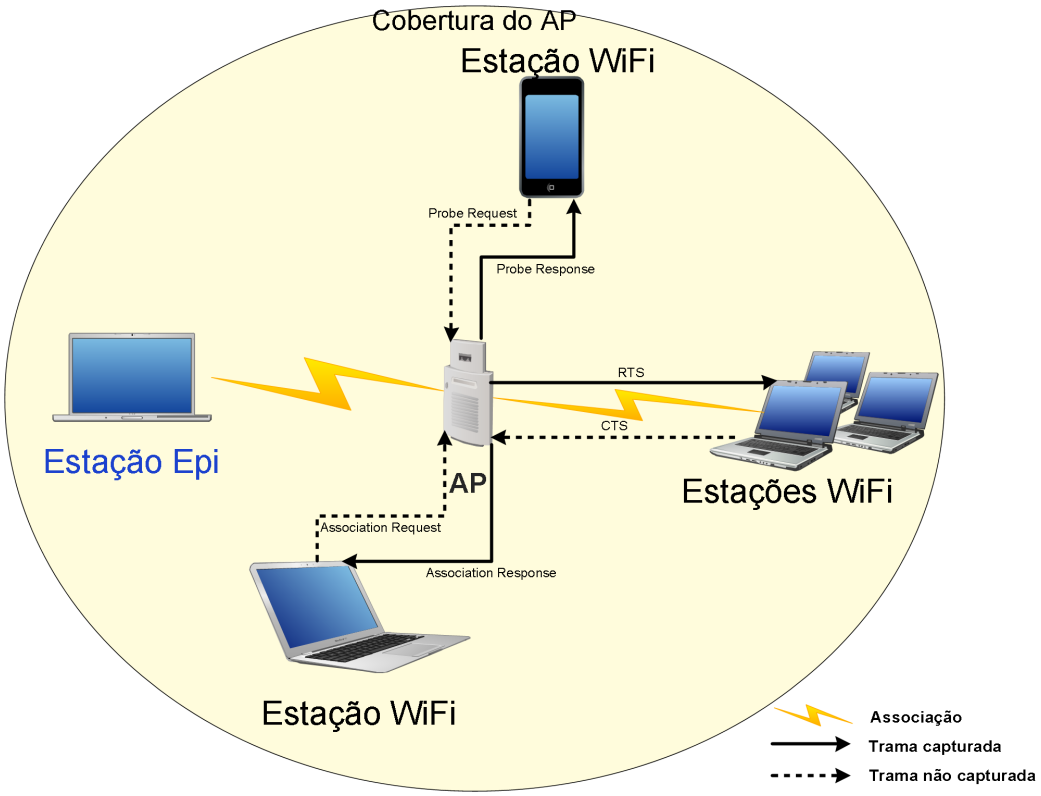


Figura 3.43: Troca de tramas entre APs e estações.

Este comportamento nas tramas de controlo e gestão verificou-se em diversas placas de rede diferentes mas não em todas. Para verificar este facto, foi usado um *software* e controlador de captura que está descrito na secção seguinte.

O cenário representado na Figura 3.42 é um exemplo das estações que vão ser detectadas. Todas as que estejam na área de cobertura do AP associadas e auten-

ticadas com o mesmo e outras que não estejam associadas e autenticadas no mesmo mas que realizem uma descoberta activa de redes *WiFi*. Estações que sejam vizinhas da estação de captura e não do AP não são detectadas.

Resta agora a captura de tráfego na camada MAC do 802.11, que é providenciado pelo *software* Microsoft Network Monitor que está descrito na próxima secção.

3.3.1 Microsoft Network Monitor

O Microsoft Network Monitor é um *software* de captura e análise de tráfego de rede, assim como o Wireshark. Este apenas foi desenvolvido para funcionar nos SOs Windows, que é uma das razões para incluir características únicas que são muito úteis para a solução que se pretende implementar.

As principais características do Network Monitor são:

- Inclui um controlador LightWeight Filter NDIS Driver, que permite capturar pacotes na camada NDIS. Trata-se de um controlador de SO que se situa acima do 802.11 *WiFi* Miniport Driver[1] e que permite expor o dispositivo através de IOCTL (Input/Output Control)[2] às aplicações do utilizador. Este permite a qualquer placa de rede sem fios, cujo controlador tenha sido desenvolvido para suportar a API Native *WiFi*, capturar tráfego ao nível da camada MAC do 802.11.

- Permite captura de tráfego em tempo real.

- Inclui um motor de *parsing* de pacotes, permitindo a análise do tráfego capturado.

- Inclui uma API que permite o acesso à captura e motor de parsing em C++ e inclui um *Wrapper* numa classe C#.net que permite acesso às suas funções.

Inclui outras características também interessantes, mas que não são tão relevantes para esta solução:

- Permite a captura em vários interfaces de rede em simultâneo.

[1] Controlador que permite ao NDIS comunicar com o *hardware* da placa de rede e fornece um *interface* para os controladores de protocolos configurarem a placa de rede e enviar e receber pacotes pela rede.

[2] *System calls* para dispositivos específicos que não podem ser expressas nas *system calls* normais.

- Inclui um utilitário para funcionar em linha de comandos.
- Inclui um conjunto grande de *parsers* de protocolos públicos e proprietários Microsoft.
- Funciona também em *Monitor Mode* e inclui um utilitário que permite colocar a placa sem fios nesse modo e controlar o canal da mesma.
- Acrescenta um cabeçalho em cada pacote *WiFi* com *meta* dados, que incluem informação sobre a transmissão do pacote. A informação disponibilizada pode variar nas diferentes placas de rede sem fios, no entanto, alguns campos são comuns como o tipo de camada física, canal, força do sinal e taxa de transmissão.

O Network Monitor também tem algumas desvantagens para a solução, mas ainda assim continua a ser a melhor solução, tendo em conta os requisitos e restrições que foram impostos para este projecto e as soluções já analisadas anteriormente. As desvantagens encontradas são as seguintes:

- Pode não ser possível, em todas as placas de rede sem fios, efectuar uma captura de tráfego na camada MAC do 802.11, uma vez que o controlador da placa de rede sem fios tem que ser desenvolvido para suportar Native *WiFi*. No entanto, o suporte para Native *WiFi* é o mais comum em placas de rede sem fios recentes, o que permite à solução suportar uma maior diversidade de *hardware*.
- O comportamento na captura de tráfego para as tramas 802.11 do tipo controlo e gestão pode não acontecer devido ao controlador da placa de rede sem fios instalado, que assim como nas tramas 802.11 do tipo dados, ignora as tramas que não lhe são destinadas. A tramas são deitadas fora pelo controlador da placa antes de serem entregues ao nível onde é efectuado a captura do tráfego. Este comportamento foi verificado, por exemplo, na placa de rede sem fios Intel *WiFi* Link 5100.
- É necessário instalar o Network Monitor no sistema para a solução funcionar e a Microsoft não permite que este seja incluído num pacote de instalação, sendo apenas permitido obter este de uma origem fidedigna da Microsoft. No entanto, esta situação foi contornada e os componentes necessários do Network Monitor foram integrados no pacote de instalação do Epi. Estes são instalados

ou removidos juntamente com o Epi de forma transparente para o utilizador. Os detalhes de como a integração foi efectuada estão descritos na Secção 5.4.

- O Network Monitor existe em versões 32 bits e 64 bits, para as respectivas versões dos SOs, logo é preciso também instalar a versão do Network Monitor adequada ao SO que vai correr.
- A captura de tráfego na camada MAC do 802.11 apenas funciona nas versões 6 do NDIS, que está presente nos SOs Windows Vista, Server 2008, 7 e 8. O Windows XP usa o NDIS 5.1, logo o seu funcionamento neste fica excluído da solução.

O Network Monitor permite então ultrapassar os principais requisitos e restrições impostos e minimizar outros. Permite abranger o maior número de sistemas e possibilita a captura de tráfego em tempo real das tramas MAC do 802.11 dos tipos dados, gestão e controlo. Com essas tramas, realizar a análise de tráfego usando o seu motor de *parsing* em tempo real para extrair do cabeçalho 802.11 os campos tipo, subtipo, direcção da trama e endereços. Com esses campos, efectuar depois o processamento e filtragem dos endereços MAC válidos para estações sem fios presentes na vizinhança, dentro do raio de cobertura do AP.

3.3.2 Aplicação para detecção das estações

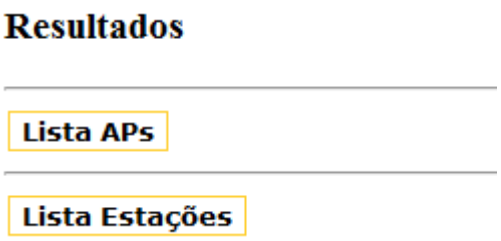
O próximo passo foi testar na prática toda a teoria analisada até agora para esta solução, que passou pelo desenvolvimento de uma aplicação em C#.net para detectar as estações sem fios na vizinhança. Foi usada a API do Network Monitor na aplicação com o respectivo *Wrapper* para C#.net.

A aplicação permite escolher um dos *interfaces* de rede disponíveis na máquina e efectuar uma captura em tempo real com a duração que o utilizador pretender ou abrir um ficheiro com tráfego de rede armazenado que foi capturado previamente. No fim do processamento dos pacotes será obtido um conjunto de ficheiros de saída com os resultados.

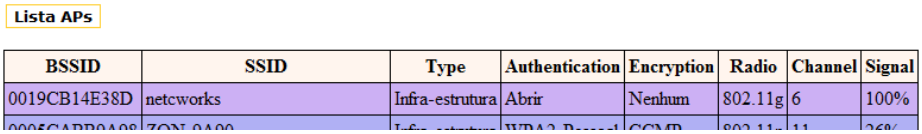
A informação chave que se pretende obter é a lista de endereços MAC das estações presentes na vizinhança do AP. No entanto, a aplicação tem como objectivo testar e, sendo assim, achou-se por bem adicionar mais informação aos resultados que possam ser úteis. Os ficheiros de resultados obtidos na saída e os ficheiros usados na entrada da aplicação são os seguintes:

result.html é o ficheiro de saída com os resultados principais. O seu formato é HTML e no seu conteúdo é apresentada uma lista com as estações presentes na vizinhança, juntamente com uma lista com as redes *WiFi* presentes na vizinhança a fazer *broadcast* do seu SSID para depois se associar com que redes as estações estão a comunicar. Na Figura 3.44 encontra-se representado um exemplo de resultados obtidos. A informação mais em detalhe contida no ficheiro é a seguinte:

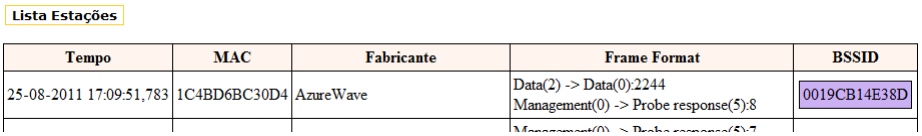
- Na lista com as estações presentes na vizinhança, para cada estação é apresentado o *timestamp* em que foi detectada a estação pela primeira vez, o endereço MAC e o respectivo fabricante, contagem dos tipos e subtipos de tramas MAC 802.11 em que a estação foi detectada, por fim o BSSID detectado nas tramas destinadas a esta estação.



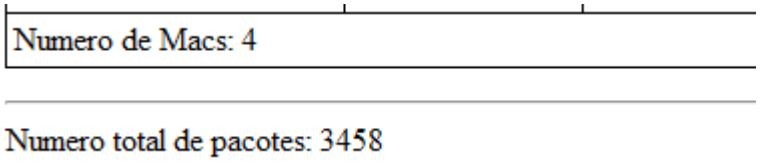
(a) Resultados.



(b) Lista APs expandido.



(c) Lista estações expandida.



(d) Estatísticas finais.

Figura 3.44: Exemplo do ficheiro de resultados obtidos numa captura.

- Na lista de redes *WiFi* são apresentadas as redes detectadas no processo de procura de redes do Windows. Para cada rede é apresentado o BSSID, SSID, *Type*, *Authentication*, *Encryption*, *Radio*, *Channel* e *Signal*. Se durante a captura e análise do tráfego é encontrada uma rede nova, o seu BSSID é adicionado à lista.
- Número total de estações detectadas.
- Número total de pacotes capturados e processados.

log.txt ficheiro de saída em formato texto que guarda alguma informação sobre todos os pacotes processados. Para cada linha deste ficheiro é apresentado:

- número de pacote.
- instante de tempo em que foi capturado o pacote (*timestamp*).
- tipo e sub tipo de pacote *WiFi* e os campos *To DS* e *From DS*, retirados do campo *Frame Control* das tramas 802.11.
- Endereço BSSID. No caso do tipo de pacote 802.11 não ter este campo, este é preenchido a zeros.
- Endereço MAC da estação presente na vizinhança.
- Endereço MAC de outra estação presente na vizinhança, no caso das IBSS.
- Quando é detectado um endereço MAC de uma estação nova, é criado uma linha adicional a indicar.

tmp.cap é um ficheiro de saída criado pelo motor de captura do Network Monitor que contém uma espécie de *buffer* com os últimos pacotes capturados. Nesta aplicação o tamanho máximo definido do *buffer* foi de 10 MB.

maccomp.txt é um ficheiro de entrada que contém a identificação dos fabricantes de placas de rede a partir dos primeiros 24 bits do seu endereço MAC. Cada linha contém os 24 bits em hexadecimal e a identificação do respectivo fabricante separados por um espaço em branco.

Para obter a informação presente nos ficheiros de saída, a aplicação começa por obter uma lista de redes *WiFi* com a informação sobre as mesmas através do método *getaps_at_range_netsh*. Nesta função é criado um novo processo que vai executar o seguinte comando e respectivos argumentos:

```
1 netsh wlan show networks mode=bssid
```

Listagem 3.8: Comando para obter a lista de rede *WiFi*.

Obtém-se assim um conjunto de linhas de texto no *output* com a lista de redes *WiFi* detectadas no Windows, que depois é extraído para uma lista de objectos do tipo *AP*.

É inicializada depois a API do Network Monitor através do método *InitNetMon* onde se começa por obter a lista de *interfaces* de rede dadas pela API, que contém as *interfaces* com o controlador do Network Monitor activado. As *interfaces* estão identificadas por uma referência GUID (Globally Unique Identifier)[1].

Para obter uma descrição mais amigável para os *interfaces* de rede, é efectuada uma pesquisa no WMI (Windows Management Instrumentation)[2] para obter a lista de *interfaces* de redes presentes no Windows com a descrição mais amigável através do método *GetNICs*. Quando é apresentado ao utilizador a lista com o índice e a descrição amigável de cada um dos *interfaces* de rede, as referências GUID que identificam cada *interface* presente nas duas listas são associadas.

Nesta altura é inicializado o motor de *parsing* e depois a sua configuração através do método *LoadNPL*. Na configuração são inseridas propriedades, campos e filtros que mais tarde vão ser aplicadas às tramas do tráfego. Nesta altura apenas são gerados e obtidos os identificadores dos mesmos, que são guardados num conjunto de variáveis inteiras positivas, usadas mais tarde para obter o conteúdo dos respectivos campos e propriedades.

As propriedades são identificadas por um nome, os campos por um caminho e o filtro é uma *string*. Na Tabela 3.15 estão representadas todas as propriedades, campos e filtros usadas na aplicação, o nome da variável que guarda o seu identificador e a respectiva descrição. As propriedades, campos e filtros representam:

Property.WiFiPktLen - propriedade que representa o tamanho dos dados contidos na trama MAC 802.11, incluindo os protocolos de camadas superiores. Actualmente esta propriedade não é usada para nenhuma função, uma vez que não é necessário aceder aos dados de camadas superiores, no entanto pode ser necessária para uso futuro.

[1] É a implementação nos SOs Windows do UUID (Universally Unique Identifier), que é uma referência usada internamente para identificar classes e interfaces COM (Component Object Model).

[2] É a infra-estrutura para gestão de dados e operações nos SOs Windows.

Nome/Caminho	Variável com ID	Tipo
Property.WiFiPktLen	mWiFiPkLen_ID	Propriedade
WiFi.FrameControl.Type	mWiFiFrameControlType_ID	Campo
WiFi.FrameControl.SubType	mWiFiFrameControlSubType_ID	Campo
WiFi.FrameControl.DS	mWiFiFrameControlDS_ID	Campo
WiFi.Management.DA	mWiFiManagementDA_ID	Campo
WiFi.Management.SA	mWiFiManagementSA_ID	Campo
WiFi.Management.BSSID	mWiFiManagementBSSID_ID	Campo
WiFi.Control.RTS.RA	mWiFiControlRTSRA_ID	Campo
WiFi.Control.RTS.TA	mWiFiControlRTSTA_ID	Campo
WiFi.Control.CTS.RA	mWiFiControlCTSRA_ID	Campo
WiFi.Control.ACK.RA	mWiFiControlACKRA_ID	Campo
WiFi.Control.CFEnd.BSSID	mWiFiControlCFEndBSSID_ID	Campo
WiFi.Control.CFEnd.RA	mWiFiControlCFEndRA_ID	Campo
WiFi.Control.PSPoll.BSSID	mWiFiControlPSPollBSSID_ID	Campo
WiFi.Control.PSPoll.TA	mWiFiControlPSPollTA_ID	Campo
WiFi.Control.CFEnd_CFAck.BSSID	mWiFiControlCFEnd_CFAckBSSID_ID	Campo
WiFi.Control.CFEnd_CFAck.RA	mWiFiControlCFEnd_CFAckRA_ID	Campo
WiFi.Control.BlockAck.RA	mWiFiControlBlockAckRA_ID	Campo
WiFi.Control.BlockAck.TA	mWiFiControlBlockAckTA_ID	Campo
WiFi.Control.BlockAckReq.RA	mWiFiControlBlockAckReqRA_ID	Campo
WiFi.Control.BlockAckReq.TA	mWiFiControlBlockAckReqTA_ID	Campo
WiFi.Data.APToClient.BSSID	mWiFiDataAPToClientBSSID_ID	Campo
WiFi.Data.APToClient.DA	mWiFiDataAPToClientDA_ID	Campo
WiFi.Data.APToClient.SA	mWiFiDataAPToClientSA_ID	Campo
WiFi.Data.ClientToAP.BSSID	mWiFiDataClientToAPBSSID_ID	Campo
WiFi.Data.ClientToAP.DA	mWiFiDataClientToAPDA_ID	Campo
WiFi.Data.ClientToAP.SA	mWiFiDataClientToAPSA_ID	Campo
WiFi.Data.AdHoc.BSSID	mWiFiDataAdHocBSSID_ID	Campo
WiFi.Data.AdHoc.DA	mWiFiDataAdHocDA_ID	Campo
WiFi.Data.AdHoc.SA	mWiFiDataAdHocSA_ID	Campo
WiFi	mWiFiFilterID	Filtro

Tabela 3.15: Propriedades, campos e filtros.

WiFi.FrameControl.Type - representa o caminho para o campo do *Frame Control* que contém o tipo de trama 802.11.

WiFi.FrameControl.SubType - representa o caminho para o campo do *Frame Control* que contém o sub-tipo de trama 802.11.

WiFi.FrameControl.DS - representa o caminho para os campos *To DS* e *From DS* do *Frame Control* da trama 802.11, onde os bits com a informação dos campos são extraídos como se apenas de um único campo se tratasse.

WiFi.Management.DA - representa o caminho para o campo com o endereço DA

de uma trama de gestão 802.11. Este é o endereço MAC da estação a que a trama é destinada.

WiFi.Management.SA - representa o caminho para o campo com o endereço SA de uma trama de gestão 802.11. Este é o endereço MAC de origem da estação que envia a trama.

WiFi.Management.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de gestão 802.11. Este é o endereço que identifica a BSS, pode ser o endereço MAC físico do *interface* de rádio do AP numa Infrastructure BSS ou o endereço MAC *locally administered* caso se trate de uma IBSS.

WiFi.Control.RTS.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo RTS do 802.11. Este é o endereço MAC do receptor, a estação a que a trama é destinada.

WiFi.Control.RTS.TA - representa o caminho para o campo com o endereço TA de uma trama de controlo RTS do 802.11. Este é o endereço MAC do transmissor, a estação que envia a trama.

WiFi.Control.CTS.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo CTS do 802.11. Uma trama CTS é transmitida como resposta a uma trama RTS e o campo de endereço RA da trama CTS é uma cópia do campo de endereço TA da trama RTS, recebida imediatamente antes. Quando não é uma trama CTS de resposta, representa o endereço MAC da estação que transmitiu a trama.

Wifi.Control.ACK.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo ACK do 802.11. Este campo é uma cópia do campo *address 2* de uma tramas dos tipos dados, gestão, *BlockAckReq*, *BlockAck* e *PS-Poll* anteriormente enviada à estação.

WiFi.Control.CFEnd.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de controlo *CF-End* do 802.11. Este é o endereço que identifica a Infrastructure BSS, o endereço MAC físico do *interface* de rádio do AP.

WiFi.Control.CFEnd.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo *CF-End* do 802.11. Este é o endereço MAC de destino, para este tipo é o endereço de *broadcast*.

Wifi.Control.PSPoll.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de controlo *PS-Poll* do 802.11. Este é o endereço que identifica a Infrastructure BSS, o endereço MAC físico do *interface* de rádio do AP.

Wifi.Control.PSPoll.TA - representa o caminho para o campo com o endereço TA de uma trama de controlo *PS-Poll* do 802.11. Este é o endereço MAC do transmissor, a estação que envia a trama.

Wifi.Control.CFEnd_CFAck.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de controlo *CF-End+CF-Ack* do 802.11. Este é o endereço que identifica a Infrastructure BSS, o endereço MAC físico do *interface* de rádio do AP.

Wifi.Control.CFEnd_CFAck.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo *CF-End+CF-Ack* do 802.11. Este é o endereço MAC de destino, para este tipo é o endereço de *broadcast*.

WiFi.Control.BlockAck.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo *Block Ack* do 802.11. Este é o endereço MAC do receptor, a estação a que a trama é destinada.

WiFi.Control.BlockAck.TA - representa o caminho para o campo com o endereço TA de uma trama de controlo *Block Ack* do 802.11. Este é o endereço MAC do transmissor, a estação que envia a trama.

WiFi.Control.BlockAckReq.RA - representa o caminho para o campo com o endereço RA de uma trama de controlo *Block Ack Req* do 802.11. Este é o endereço MAC do receptor, a estação a que a trama é destinada.

WiFi.Control.BlockAckReq.TA - representa o caminho para o campo com o endereço TA de uma trama de controlo *Block Ack Req* do 802.11. Este é o endereço MAC do transmissor, a estação que envia a trama.

WiFi.Data.APToClient.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de dados do 802.11 enviada do DiS para as estações. Este é o endereço MAC físico do *interface* de rádio do AP que identifica a Infrastructure BSS.

WiFi.Data.APToClient.DA - representa o caminho para o campo com o endereço DA de uma trama de dados do 802.11 enviada do DiS para as estações. Este é o endereço MAC da estação a que a trama é destinada.

WiFi.Data.APToClient.SA - representa o caminho para o campo com o endereço SA de uma trama de dados do 802.11 enviada do DiS para as estações. Este é o endereço MAC de origem da trama, a estação que criou a trama.

Wifi.Data.ClientToAP.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de dados do 802.11 enviada das estações para o DiS. Este é o endereço MAC físico do *interface* de rádio do AP que identifica a Infrastructure BSS.

Wifi.Data.ClientToAP.DA - representa o caminho para o campo com o endereço DA de uma trama de dados do 802.11 enviada das estações para o DiS. Este é o endereço MAC da estação a que a trama é destinada.

Wifi.Data.ClientToAP.SA - representa o caminho para o campo com o endereço SA de uma trama de dados do 802.11 enviada das estações para o DiS. Este é o endereço MAC de origem da trama, a estação que criou a trama.

Wifi.Data.AdHoc.BSSID - representa o caminho para o campo com o endereço BSSID de uma trama de dados do 802.11 enviada dentro de uma IBSS. Este é o endereço MAC *locally administered* que identifica a IBSS.

Wifi.Data.AdHoc.DA - representa o caminho para o campo com o endereço DA de uma trama de dados do 802.11 enviada dentro de uma IBSS. Este é o endereço MAC da estação a que trama é destinada.

Wifi.Data.AdHoc.SA - representa o caminho para o campo com o endereço SA de uma trama de dados do 802.11 enviada dentro de uma IBSS. Este é o endereço MAC de origem da trama, da estação que criou a trama.

WiFi - representa o filtro que vai ser aplicado para apenas permitir o processamento de pacotes com cabeçalho 802.11.

Terminado o carregamento do motor de *parsing* e a obtenção dos identificadores, é apresentando ao utilizador a lista de *interfaces* de redes disponíveis para a captura de tráfego, com a opção de abertura de tráfego também a partir de ficheiro. Para qualquer dos casos é inicializado o motor de captura do Network Monitor e a captura de tráfego começa.

Cada pacote capturado é processado individualmente e em tempo real. A captura termina até o utilizador pressionar "Enter", no caso da captura ser a partir de um *interface* de rede. Caso seja a partir de ficheiro, o processamento decorre até este chegar ao fim.

O processamento dos pacotes é efectuado em *background* numa nova *thread* que foi iniciada e imediatamente bloqueada antes do motor de captura. Esta *thread* permanece bloqueada enquanto não receber sinalização para desbloquear, o que só acontece quando é capturado um novo pacote. Quando é capturado um novo pacote, é chamado pelo motor do Network Monitor uma função de *callback* que adiciona o novo pacote a um *buffer* de pacotes e envia o sinal para desbloquear a *thread*. A *thread*, que se encontrava até agora bloqueada, chama um método onde é iniciado o processamento do pacote ou pacotes recebidos e armazenados no *buffer*.

O método *ParseFrames()* realiza o processamento dos pacotes pela mesma ordem que foram recebidos. É pedido ao motor do Network Monitor o próximo pacote armazenado no *buffer* através de um número de pacote. O pacote é filtrado chamando o método *GetWifi* que vai aplicar e validar o filtro para verificar se se trata de um pacote *WiFi*. O processamento do pacote prossegue se este passar na validação, ou termina e avança para o pacote seguinte. Ao prosseguir, é aplicado o *parsing* configurado com os campos e propriedades que foram definidos anteriormente. É obtido o novo pacote resultante com o *parsing* efectuado que é armazenado num novo pacote.

A partir daqui é extraída a informação presente no pacote, começando pelos *timestamp*, depois os campos tipo, sub-tipo e *DS* do *Frame Control*. Os próximos campos que se pretende obter dizem respeito ao endereçamento mas estes não são fixos como os campos anteriores e variam consoante o tipo, sub-tipo e direcção do pacote.

Para retirar os campos de endereçamento, começa-se por verificar primeiro o tipo

de pacote 802.11. As operações seguintes variam, sendo para os diferentes tipos efectuadas da seguinte forma:

Tipo *Management*

O endereçamento é sempre efectuado da mesma forma, logo não necessita que o processamento dos sub-tipos tenha que ser efectuado individualmente. A direcção, campo *DS*, também é sempre igual a "0" como mostra a Tabela 2.5. Assim quando um pacote 802.11 é de gestão, os campos com os endereços são extraídos usando os identificadores *mWiFiManagementDA_ID*, *mWiFiManagementSA_ID* e *mWiFiManagementBSSID_ID* para variáveis *address1*, *address2* e *address3*, respectivamente. Depois é verificado o campo *DS* para garantir que este campo se encontra a "0", caso contrário o processamento do pacote termina. Os pacotes de gestão são usados apenas entre o AP e as estações e não são entregues para o DiS, uma vez que só dizem respeito à gestão da ligação entre os dois e não transportam dados para camadas superiores. Portanto, os endereços MAC presentes nos campos do pacote são de estações na vizinhança e APs locais. Para extrair esses endereços MAC de forma correcta é necessário fazer algumas verificações, uma vez que os pacotes de gestão podem ser *request* e *response*. Podem ser enviados entre duas estações numa IBSS, de um AP para uma estação ou na direcção contrária numa Infrastructure BSS. Assim podemos separar em 3 situações diferentes, usando como referência a distribuição de endereços na Tabela 2.7:

- O pacote é enviado de uma estação na vizinhança para o AP, isto verifica-se se o endereço de BSSID for igual ao endereço de destino (DA) do pacote. Assim temos como estação vizinha o endereço MAC de origem (SA).
- O pacote é enviado do AP para uma estação na vizinhança, isto verifica-se se o endereço de origem (SA) for igual ao endereço de BSSID do pacote. Assim temos como estação vizinha o endereço MAC de destino (DA).
- Um pacote enviado entre duas estações na vizinhança que se encontrem na mesma IBSS, neste caso verifica-se se o endereço de BSSID é um endereço MAC do tipo *locally administered*. Se for, temos duas estações vizinhas com o endereços MAC de origem (SA) e de destino (DA).

Depois dessas verificações temos o endereço MAC de uma das estações presentes

na vizinhança, ou no caso de uma IBSS, os endereços MAC de duas estações presentes na vizinhança.

Tipo *Control*

Contém endereçamento diferente para cada sub tipo de pacote, logo cada um é processado individualmente. A direcção, campo *DS*, é também sempre igual a "0" para este tipo de pacote como mostra a Tabela 2.5, contudo, este campo é sempre verificado se está a "0", caso contrário o processamento deste pacote termina. Este tipo de pacotes são usados para assistir a entrega de pacotes de dados entre estações e AP que usam o mesmo meio de acesso por isso o endereçamento presente no pacote será apenas local. Assim, para cada sub tipo, o processamento é efectuado da seguinte forma:

Block Acknowledgment Request - Um pacote deste sub-tipo tem dois campos de endereços que são representados pelos identificadores *mWiFiControlBlockAckReqRA_ID* e *mWiFiControlBlockAckReqTA_ID*, o primeiro representa o endereço MAC do receptor (RA) e o segundo o endereço MAC do transmissor (TA) do pacote. Como nenhum desses campos é identificado como um endereço de BSSID, não se consegue concluir nada quanto às estações na vizinhança, no entanto, usando a lista de redes *WiFi*, é verificado se os endereços MAC do receptor ou transmissor são uma BSSID, em caso de o ser, o outro endereço MAC pode ser considerado uma estação que está presente na vizinhança.

Block Acknowledgment - Um pacote deste sub-tipo tem dois campos de endereços que são representados pelos identificadores *mWiFiControlBlockAckRA_ID* e *mWiFiControlBlockAckTA_ID*, o primeiro representa o endereço MAC do receptor (RA) e o segundo o endereço MAC do transmissor (TA) do pacote. O processamento é efectuado como o sub-tipo anterior, *Block Acknowledgment Request*.

PS-Poll - Um pacote deste sub-tipo tem dois campos de endereços que são representados pelos identificadores *mWiFiControlPSPollBSSID_ID* e *mWiFiControlPSPollTA_ID*, o primeiro representa o endereço MAC que identifica a BSSID ao qual o pacote é destinado, o segundo é o endereço MAC do transmissor (TA) do pacote. Estes pacotes são sempre enviados por uma estação para uma Infrastructure BSS, logo, pode-se concluir que o en-

dereço MAC do transmissor é uma estação presente na vizinhança. Para validar esta situação o BSSID do pacote é verificado se existe na lista de redes *WiFi*.

RTS - Um pacote deste sub-tipo tem dois campos de endereços que são representados pelos identificadores *mWiFiControlRTSRA_ID* que representam o endereço MAC de receptor (RA) e *mWiFiControlRTSTA_ID* o endereço MAC do transmissor (TA) do pacote. O processamento é efectuado tal como o sub-tipo anterior, *Block Acknowledgment Request*.

CTS - Um pacote deste sub-tipo tem apenas um campo de endereço que é representado pelo identificador *mWiFiControlCTSRA_ID*. Este é ignorado, uma vez que apenas com um único endereço nada se pode concluir quanto às estações presentes na vizinhança. Poderia ser possível retirar alguma conclusão se a análise dos pacotes fosse feita por fluxo de pacotes.

Acknowledgment - Um pacote deste sub-tipo tem apenas um campo de endereço que é representado pelo identificador *mWiFiControlACKRA_ID* e este é simplesmente ignorado pelos mesmos motivos que o sub-tipo anterior, CTS.

CF-End - Um pacote deste sub-tipo tem dois campos de endereços, que são representados pelos identificadores *mWiFiControlCFEndBSSID_ID* e *mWiFiControlCFEndRA_ID*, o primeiro representa o endereço BSSID que identifica a Infrastructure BSS em que o pacote tem origem e o segundo o endereço MAC de receptor (RA) do pacote. Este pacote é enviado sempre de uma BSSID para todas a estações, o endereço de receptor é o endereço de *broadcast*, logo este pacote também será ignorado porque não fornece nenhuma informação sobre as estações presentes na vizinhança.

CF-End+CF-Ack - Um pacote deste sub-tipo tem dois campos de endereços, que são representados pelos identificadores *mWiFiControlCFEnd_CFAckBSSID_ID* e *mWiFiControlCFEnd_CFAckRA_ID*, o primeiro representa o endereço BSSID que identifica a Infrastructure BSS em que o pacote tem origem e o segundo o endereço MAC de receptor (RA) do pacote. Este pacote é idêntico ao sub-tipo anterior, *CF-End*, sendo processado da mesma forma.

Depois da verificação dos sub-tipos, alguns pacotes contêm informação que per-

mite extrair o endereço MAC de estações presentes na vizinhança, verificando em alguns casos a lista de BSSIDs. Outros sub-tipos são ignorados porque o pacote porque não contém informação que seja útil.

Tipo *Data*

Contém endereçamento sempre efectuado da mesma forma para todos sub-tipos, no entanto a direcção, campo *DS*, já não é "0" como os tipos anteriores. Este varia, o que o torna importante para o processamento dos endereços, uma vez que este é que vai ditar o que cada endereço representa, segundo a Tabela 2.7. Nos pacotes de dados vêm encapsuladas mensagens para serem entregues às camadas superiores, por isso no endereçamento contém endereços MAC de origem e destino da rede local ou Internet, que podem não pertencer à vizinhança. Dos pacotes capturados, têm os que são destinados à rede local e que são entregues ao DiS através do AP numa Infrastructure BSS, e o inverso, que têm origem na rede local e que são entregues ao AP pelo DiS, com destino a uma estação ligada nesse AP. Existe também o caso da troca de pacotes de dados entre duas estações na mesma IBSS. Assim o processamento procede consoante o campo *DS*, da seguinte forma:

DS=0 Trata-se de um pacote enviado dentro de uma IBSS. Existem três campos de endereços representados pelos identificadores *mWiFiDataAdHocBSSID_ID*, *mWiFiDataAdHocDA_ID* e *mWiFiDataAdHocSA_ID* onde depois são extraídos os endereços *address3*, *address1* e *address2* que segundo a Tabela 2.7, representam os endereços BSSID, DA e SA, respectivamente. Verifica-se primeiro se o endereço de BSSID é um endereço MAC do tipo *locally administered* para confirmar que se trata mesmo de um pacote de uma IBSS. Caso se confirme, temos duas estações presentes na vizinhança que trocam pacotes de dados, por isso considera-se os endereços MAC de origem (SA) e de destino (DA) como estações presentes na vizinhança.

DS=1 Trata-se de um pacote enviado de uma estação para o AP numa rede Infrastructure BSS. Existem três campos de endereços representados pelos identificadores *mWiFiDataClientToAPBSSID_ID*, *mWiFiDataClientToAPDA_ID* e *mWiFiDataClientToAPSA_ID* onde depois são extraídos os endereços *address1*, *address3* e *address2* que segundo a Tabela 2.7, representam os endereços BSSID, DA e SA, respectivamente. É primeiro

verificado se o endereço de BSSID é um endereço MAC do tipo *universally administered*, que confirma que este é de um AP real de uma rede em Infrastructure BSS. Caso se confirme, temos uma estação presente na vizinhança que enviou este pacote para o AP e por isso considera-se o endereço MAC de origem (SA) como estação presente na vizinhança.

DS=2 Trata-se de um pacote enviado de um AP para uma estação numa rede Infrastructure BSS. Existem três campos de endereços representados pelos identificadores *mWiFiDataAPToClientBSSID_ID*, *mWiFiDataAPToClientDA_ID* e *mWiFiDataAPToClientSA_ID* onde depois são extraídos os endereços *address2*, *address1* e *address3* que segundo a Tabela 2.7, representam os endereços BSSID, DA e SA, respectivamente. É primeiro verificado se o endereço de BSSID é um endereço MAC do tipo *universally administered*, que confirma que este é de um AP real de uma rede em Infrastructure BSS. Caso se confirme, temos uma estação presente na vizinhança que recebeu este pacote enviado pelo AP e por isso considera-se o endereço MAC de destino (DA) como estação presente na vizinhança.

Depois dessas verificações temos o endereço MAC de uma das estações presentes na vizinhança numa Infrastructure BSS, ou no caso de uma IBSS, os endereços MAC de duas estações presentes na vizinhança.

Para finalizar o processamento de um pacote, com o endereço MAC da estação presente na vizinhança obtido, endereço BSSID, *timestamp* e tipo de pacote *WiFi*, chama-se o método *ProcessAddress*. Este é responsável por processar o endereço MAC obtido, verificando se é um endereço válido. O endereço MAC de uma estação vizinha só é considerado válido se for do tipo *universally administered*. Após ser validado, é adicionado à lista de estações na vizinhança juntamente com o tipo de pacote *WiFi* em que foi encontrado, *timestamp* e BSSID. Caso o endereço MAC já exista na lista, soma uma unidade à contagem do respectivo tipo de pacote *WiFi* e adiciona o BSSID caso não exista. Quando se obtém um endereço de BSSID, este é adicionado à lista de redes *WiFi*, caso não exista.

Quando termina a captura de tráfego, para cada endereço MAC da lista das estações presentes na vizinhança, é determinado o fabricante da respectiva placa de rede através de uma lista de fabricantes. Por fim, é escrito para o ficheiro de resultados a lista de estações, a lista de redes *WiFi*, o número de estações detectadas e o número

de pacotes processados.

3.3.3 Resultados da aplicação

As capturas efectuadas anteriormente, na primeira solução, não podem ser usadas agora uma vez que os pacotes não incluem a camada MAC do 802.11 que é necessária uma vez que é de lá que se obtém os cabeçalhos com o endereçamento. Para obter resultados foi necessário efectuar novas capturas de tráfego, tanto em tempo real como para ficheiro.

A captura em tempo real foi efectuada novamente nos dois locais usados na solução anterior e adicionalmente em novos locais. Estes locais foram escolhidos devido à existência de redes *WiFi* de pequenas dimensões constituídas por apenas um AP ligado a uma rede local. Estas redes *WiFi* tinham suporte para diversos tipos de rádio do 802.11, incluindo, em algumas, o mais recente 802.11n.

Depois da captura e análise do tráfego, usando a aplicação desenvolvida, foram obtidos nos respectivos locais os seguintes resultados:

UM, Azurém

É o local que tem a maior rede *WiFi* em que a aplicação foi testada. O DiS alarga-se por diversos APs distribuídos pelo campus da universidade, usando a camada física até ao 802.11g. Foram efectuadas capturas em diversos locais, começando pela biblioteca, onde se obteve em pouco tempo de captura (cerca de 15 segundos) uma lista de estações vizinhas de aproximadamente 30 estações, o que parece ser um número válido para as estações que estavam no raio do AP. Numa captura mais longa foram aparecendo mais estações. Algumas dessas foram detectadas através dos pacotes do tipo gestão de *Probe Response*, o que significa que fizeram uma procura activa por redes *WiFi* na vizinhança ao qual o AP respondeu. Estas podem-se encontrar associadas ao AP ou apenas fizeram uma procura quando se encontravam na vizinhança. Outras estações ligaram-se à rede já depois da captura ter começado e foram também imediatamente detectadas através dos pacotes do tipo gestão de *Association response* e *Authentication*. Outra captura efectuada no laboratório do DSI, que também decorreu durante alguns segundos (cerca de 15 segundos), mostrou uma lista mais pequena de estações. Foi acedido depois ao AP mais próximo ao qual a estação de captura se encontrava associada por SNMP (Simple Network Management Protocol) e verificou-se a lista de estações que se encontravam associadas

ao AP. O resultado obtido foi uma lista de endereços MAC de estações em que quase todos se encontravam também na lista de estações que foi obtida pela aplicação. Houve também estações detectadas pela aplicação que não estavam associadas ao AP, isto porque também são detectadas as estações que fazem procura de rede activas na vizinhança do AP, através de pacotes do tipo *Probe Response*. Pode acontecer estações presentes na vizinhança não serem detectadas numa captura de curta duração quando se encontram associadas ao AP e a sua actividade na rede é muito baixa, não havendo nenhuma troca de pacotes no decorrer de uma captura. Por outro lado, quanto maior a actividade de uma estação na rede no decorrer de uma captura, maior é a probabilidade desta ser detectada.

Café Jardim

Este local tem uma pequena rede *WiFi* pública a funcionar com a camada física 802.11n. Aqui a captura de tráfego também se realizou durante pouco tempo (cerca de 15 segundos) tendo-se obtido uma lista pequena de estações presentes na vizinhança, que foram 8. As estações que estavam associadas ao AP no momento de captura foram todas detectadas, mediante a verificação no AP das estações presentes no local. Foram também detectadas outras estações, através dos pacotes de *Probe Response*, o que significa que existiram estações que efectuaram uma procura activa por redes *WiFi* na vizinhança do AP que não se encontravam associadas ao AP.

Outros Locais

Foram efectuados outros testes em diversos locais diferentes, onde existiam redes *WiFi* privadas constituídas por um único AP a funcionar com diversas camadas físicas do 802.11. Nestes locais os resultados foram idênticos aos anteriores, obteve-se uma lista de estações em que todas as estações que se encontravam associadas ao AP encontravam-se também nessa lista e também outras estações que foram detectadas ao efectuarem uma procura activa por redes *WiFi* na vizinhança do AP. Outro teste efectuado foi a criação de uma rede em modo *Ad-hoc* constituída por duas estações e o resultado foi a detecção das duas estações que constituiam a rede.

3.3.4 Conclusão da solução

Com os resultados obtidos na aplicação pode-se concluir que a solução funciona, na medida em que detecta quase todas as estações associadas e autenticadas no mesmo AP a que a estação que realiza a captura de tráfego está associada e outras estações na vizinhança que não se encontram associadas ao AP mas que fazem uma procura activa por redes *WiFi*. Existem estações que não são detectadas durante uma captura porque já se encontravam associadas ao AP e no decorrer da captura permaneceram inactivas na rede, gerando um número muito baixo de pacotes ou mesmo nulo. Estas poderiam eventualmente ser detectadas se a captura se prolongasse por mais tempo. Por outro lado, durante uma captura, as estações associadas ao AP que estão activas são detectadas. As estações que fazem a autenticação ou associação com o AP no decorrer da captura são também detectadas. Estações que efectuem uma procura activa de redes sem fios na vizinhança do AP são também detectadas, apesar de não se encontrarem associadas ao mesmo. A procura activa por redes sem fios é usada por todos os SOs actuais. As estações podem estar estacionárias no raio de cobertura do AP ou estarem apenas a passar na vizinhança do AP no momento em que se estava a realizar a captura de tráfego mas para todos os efeitos era uma estação presente na vizinhança naquele momento.

A aplicação irá conseguir detectar mais estações quanto mais tempo se prolongar a captura do tráfego, no entanto, apenas alguns segundos já permitem obter uma lista precisa da maior parte das estações presentes na vizinhança.

O Network Monitor, com o seu controlador NDIS e motor de *parsing*, permite capturar na camada MAC do 802.11 os pacotes 802.11, no entanto, a captura não funciona da mesma forma para todas a placas de rede sem fios existentes. Em algumas placas testadas todas os pacotes que não tinham como destino a estação de captura simplesmente eram descartados antes de serem capturados. Isto acontece porque o controlador da placa de rede sem fios descarta os mesmos antes de entregar ao nível superior dentro do NDIS, onde funciona o controlador do Network Monitor. Nestas placas não será possível obter os resultados pretendidos e não existe uma solução genérica para todas as placas. Seria necessário uma análise profunda para cada placa de rede sem fios existente.

Não há duvida que esta é a solução que permite abranger um maior número de placas de redes sem fios, o que se traduz num maior número de sistemas e menor número de desvantagens face às restrições e requisitos impostos inicialmente para

este projecto. Sendo assim, esta é a solução escolhida e mais adequada para integrar no Epi como um módulo de *software*.

Capítulo 4

Projecto Epi

A solução encontrada será integrada na aplicação Epi. Neste capítulo é feito uma pequena descrição da aplicação Epi e da sua arquitectura de sistema. Depois são descritas as novas funcionalidades introduzidas e as modificações que foram efectuadas na aplicação do Epi, no protocolo de comunicações e no servidor.

A aplicação Epi enquadra-se no contexto das *Collaborative Sensing Networks*. A secção seguinte introduz este conceito.

4.1 Collaborative Sensing

O módulo desenvolvido neste projecto foi integrado num outra aplicação já existente destinada a dispositivos com *interface WiFi*. Esta aplicação armazena informação do ambiente rádio que rodeia os utilizadores enquanto fornece funcionalidades de rede social baseada na difusão epidémica de mensagens entre utilizadores próximos[9]. Os seus utilizadores colaboram na recolha e armazenamento de informação sobre o comportamento das pessoas em ambientes urbanos e sobre as características que os rodeiam.

A informação do ambiente de rádio é recolhida através de um sensor presente no dispositivo, o *interface WiFi*, que é usado para a recolha das redes *WiFi* presentes na vizinhança desse dispositivo. O módulo desenvolvido neste projecto tem a finalidade de constituir um sensor adicional com a capacidade de recolher informação sobre as estações *WiFi* que se encontram presentes na vizinhança.

A recolha e partilha de informação neste cenário de estudo é feita de forma livre e espontânea por parte dos utilizadores que colaboram e isto é a essência principal

das redes conhecidas como *Collaborative Sensing Networks* ou *Collaborative Sensor Networks*.

4.1.1 Collaborative sensing networks

A presença ubíqua de dispositivos móveis ou portáteis com diferentes capacidades trouxe aos dias de hoje um conjunto de desafios e também oportunidades no que diz respeito à disseminação e recuperação de informação em ambientes urbanos. Esses dispositivos podem ser de uma vasta variedade que inclui telemóveis, *smartphones*, *tablets*, computadores portáteis, câmaras de vídeo ou fotográficas, leitores de música, veículos, etc.

Houve um progresso tremendo nas tecnologias de redes de sensores, sobretudo para aplicações militares. Nos dias de hoje as necessidades civis para esse tipos de redes está a crescer, assim como as suas aplicações. Por exemplo, muitas cidades estão a considerar a distribuição de sensores para monitorização do ambiente (temperatura, humidade, qualidade do ar, etc.). No entanto, uma abordagem dessas requer que um grande número de sensores fixos estejam colocados em posições geográficas seleccionadas. Seria necessário um grande número de dispositivos para cobrir uma grande área, assim como uma infra-estrutura de rede complexa para ligar todos os sensores fixos aos centros de processamento de dados[44].

Um operador de redes móveis pode também querer recolher informações relativas à qualidade da sua cobertura de rádio e dos serviços. O que normalmente faz é enviar técnicos para determinadas áreas de interesse geográfico e efectuar medições, o que é uma solução cara e pouco eficaz. Os dispositivos móveis hoje em dia contêm uma infinidade de sensores (câmaras, acelerómetros, GPS, *interfaces WiFi*, Bluetooth, GSM, etc.) que ao serem usados de modo colaborativo podem efectuar tarefas de *sensing*. Os telemóveis podiam recolher periodicamente informação acerca da qualidade de cobertura rádio da rede de um operador, uma vez que estes são um poderoso sensor para obter informação acerca da rede sem a implementação de novo *hardware* ou *software*. O operador depois recolhia essa informação que era partilhada pelos telemóveis que colaboraram para a monitorização da qualidade da rede.

As aplicações deste tipo de redes podem ser imensas, medições de temperatura, poluição do ar ou sonora, prevenção de acidentes, alertas relacionados com pessoas, tráfego automóvel, etc. São múltiplas variáveis que podem ser medidas a toda a hora através dos biliões de dispositivos de comunicação que as pessoas possuem e que são

usados em ambiente pessoal, social e urbano.

Estas redes não se limitam a determinado espaço ou área como as normais redes de sensores. A mobilidade do *sensing* está associada à mobilidade dos utilizadores com os seus dispositivos durante as actividades diárias. Isto permite obter informação útil acerca dos utilizadores e do ambiente que os rodeia. A recolha dessa informação abre portas a vários cenários de estudo pessoal, social e urbano.

As redes *WiFi* e móveis possuem um papel importante uma vez que fornecem a mobilidade dos "sensores", podendo cobrir diferentes áreas geográficas em instantes de tempo diferentes. Assim é efectuada a monitorização de uma área maior do que nas normais redes de sensores.

4.1.2 Arquitectura do sistema

Numa *Collaborative Sensing Network* geralmente podemos identificar um conjunto de componentes que vão realizar determinadas tarefas:

Sensor Móvel - é um dispositivo móvel com capacidade para suportar tarefas de *sensing* para uma aplicação específica. Estes dispositivos contêm um conjunto de sensores como receptores GPS, *interfaces WiFi*, microfones, câmaras, acelerômetros e módulos auxiliares ligados nos *interfaces* disponíveis (como Bluetooth). Estes dispositivos são questionados para efectuar determinada tarefa de *sensing* e depois enviar os dados recolhidos para o Sistema Central. O envio da informação recolhida pode ser feito imediatamente se o dispositivo tiver conectividade através de uma rede *WiFi* ou móvel, ou ser atrasado. Neste caso vai armazenar a informação até ter conectividade com o Sistema Central.

Infra-estrutura - para que seja possível questionar os dispositivos para realizar determinada tarefa de *sensing* e depois recolher os dados resultantes dessa tarefa, é necessária uma infra-estrutura de rede. Essa infra-estrutura de rede é normalmente uma rede *WiFi* ou móvel já existente. Hoje em dia em qualquer ambiente urbano existe cobertura de uma rede *WiFi* e móvel. As redes *WiFi* ainda são as preferíveis uma vez que geralmente são grátis, no entanto está a tornar-se normal a conectividade permanente à Internet em muitos dispositivos móveis usando a rede móvel celular.

Sistema Central - o Sistema Central é normalmente um computador ou conjunto de computadores com um endereço permanente e com poder de processamento,

largura de banda e armazenamento suficiente para a tarefa que vai realizar. Este vai recolher a informação que é enviada pelos sensores móveis e, recorrendo a um algoritmo de análise de dados específico ao estudo que se pretende efectuar, formular os resultados. O Sistema Central pode também realizar um conjunto de tarefas administrativas como o registo dos sensores móveis, validação dos dados submetidos das tarefas de *sensing*, construção de directorias de tarefas de *sensing* válidas para questionar os sensores móveis, criação de relatórios dos sensores móveis e disponibilização de dados para aplicações.

Relatórios dos resultados - os relatórios com os resultados obtidos após o processamento da informação podem ser apresentados e partilhados com os utilizadores do sistema através de aplicações. Se o modelo for participativo os resultados devem ser apresentados também aos sensores móveis, uma vez que as pessoas que utilizam os sensores móveis também têm interesse nos resultados.

A colaboração dos sensores móveis na recolha de dados nestes sistemas pode ser efectuada de forma participativa ou oportunística [45][46].

Colaboração partitipativa - a recolha de dados requer que o portador do dispositivo de forma consciente e explícita escolha quais as tarefas de *sensing* a realizar pelo sistema, decidindo qual a informação que irá ser partilhada. O portador nestes sistemas também tem interesse na informação, logo, é preciso focar no desenvolvimento de ferramentas que permitam partilhar, publicar, procurar, interpretar e verificar as informações recolhidas. Nestes casos a informação obtida é mais ampla e objectiva e requer o desenvolvimento de uma aplicação mais complexa e apelativa, de forma a cativar a comunidade à sua utilização, ou mesmo de dispositivos adaptados para o sistema em específico. Este modelo deve ser usado quando existe um conjunto de participantes interessados nos resultados a serem formulados pelo sistema.

Colaboração oportunística - a recolha de dados é feita sem interacção do portador do dispositivo, este pode ou não estar consciente de como e quando o seu dispositivo se encontra a recolher informação. O portador não alterna o estado do seu dispositivo para satisfazer as necessidades das tarefas de *sensing*. Nestes casos, de forma a manter a transparência, a tarefa a realizar não deve causar um impacto na experiência de utilização no dispositivo do portador. O principal

desafio na recolha oportunística é determinar quando o estado do dispositivo tem os requisitos necessários para efectuar a recolha de informação, podendo mesmo a recolha não ser efectuada. As aplicações não necessitam de ser tão complexas e apelativas, em alguns casos não são mesmo necessárias, o que pode resultar num número maior de sensores com menos informação útil ou mesmo insuficiente. Este modelo deve ser usado quando a recolha é feita de forma automática sem a necessidade de intervenção directa dos participantes.

Os movimentos dos dispositivos móveis podem ser feitos de forma controlada ou não, para realizar uma tarefa de recolha. Nestes sistemas o movimento de um dispositivo pode ser classificado de duas formas[44]:

Controlado - os movimentos dos dispositivos é controlado de forma a realizar uma tarefa de *sensing*. Por exemplo, autocarros, aviões, balões de ar quente que viajam por percursos fixos ou controlados.

Não controlado - os movimentos destes dispositivos é aleatório e não controlável de forma a realizar uma tarefa de *sensing*. Por exemplo, táxis, carros polícia, pessoas que se movem aleatoriamente numa área. O uso de dispositivos não controlados têm algumas vantagens. Um número grande destes dispositivos em táxis, por exemplo, normalmente estão disponíveis em locais de interesse numa cidade. Podiam carregar sensores para ajudar na monitorização de uma área, podendo reduzir significativamente ou mesmo eliminar a necessidade de enviar veículos e pessoas dedicadas para realizar a mesma tarefa. Desta forma, usar veículos não controlados com sensores pode ajudar a reduzir o número de sensores necessários para a cobertura de determinada área, relativamente ao uso de sensores fixos.

Num sistema com sensores controlados e não controlados podem coexistir também sensores fixos. Um sensor móvel pode ser usado para a monitorização de uma área frequentada pelos portadores desses dispositivos. Sensores fixos podem ser colocados em locais pouco frequentados e de difícil acesso. Os sensores controlados podem depois seguir uma situação anormal detectada por um sensor não controlado.

4.1.3 Privacidade e Segurança

No desenvolvimento destes tipos de redes, principalmente em ambientes urbanos, são enfrentados desafios relacionados com a privacidade e segurança. Alguns desses

problemas estão muito relacionados com os problemas de segurança existentes nas próprias redes móveis hoje em dia.

Existem investigações realizadas que se focam na privacidade dos portadores humanos dos dispositivos móveis[47], exploração de mecanismos de protecção de privacidade através do anonimato e agregação dos dados recolhidos[45].

A protecção da privacidade dos dados recolhidos é de vital importância porque existem dados sensíveis contidos na localização, gravações áudio e vídeo, fotografias, sinais *WiFi*, Bluetooth e em muitos mais dados recolhidos em tarefas de *sensing*. A privacidade de espectadores terceiros também deve ser considerada, uma vez que informação sensível relacionada com os mesmos pode ser gravada por uma aplicação de *sensing* activa na área, desta forma violando os seus direitos de privacidade. O *sensing* sem notificação não é muito diferente da gravação de uma pessoa sem o seu consentimento.

Adicionar transparência aos sistemas de *sensing* urbano pode permitir às pessoas excluírem-se dessas recolhas de informação, ao evitar áreas onde estas actividades estejam a decorrer. No Google Street View, por exemplo, foi criado um sistema de notificação que consiste em publicar o percurso dos seus carros. Outra abordagem foi o anonimato ao desfocar a cara de espectadores, no entanto, este não é completamente eficaz.

Essas questões de privacidade são importantes a ter em conta no desenvolvimento de um sistema de *sensing* urbano de forma a que a aceitação social destes sistemas seja extensa.

A segurança também é importante nas recolhas de informação de um dispositivo móvel. As recolhas são feitas nos dispositivos móveis das pessoas, que são alvos geralmente mais "fáceis" e "apetecíveis" por parte de alguém mal intencionado que pretenda capturar e adulterar a informação recolhida. Desta forma, o uso de redes e encaminhamento seguro poderia suavizar o problema, mas o melhor mesmo nestas redes é arranjar maneira de garantir a integridade da informação recebida.

4.1.4 Projectos relacionados

Foi realizada uma pesquisa por diversos projectos relacionados com *Collaborative Sensing* e redes 802.11. Actualmente estes projectos são desenvolvidos maioritariamente em ambientes académicos e muitos envolvem a utilização de redes 802.11 como infra-estrutura e como fonte de informações diversas.

No Brasil, na Universidade Federal do Rio de Janeiro, existe um projecto que explora as infra-estruturas de rede 802.11 já existentes para a monitorização do trânsito[48]. Os veículos fazem a recolha da informação dos *Beacons* recebidos dos APs que já existiam ao longo de uma avenida, por exemplo. A informação é enviada para um Sistema Central que vai efectuar o processamento dos dados, disponibilizando depois informação que seja útil para os condutores, como a localização do veículo, direcção, velocidade, condições de trânsito baseado no tempo que os veículos demoram a percorrer uma distância.

Aqui os veículos são os sensores móveis que recolhem toda a informação fundamental e que depois a enviam para ser processada. Os utilizadores são participativos e necessitam de um *interface* 802.11 e da aplicação desenvolvida que vai usufruir desses serviços.

Um outro projecto idêntico é o CarTel[49] desenvolvido pelo MIT. Este permite recolher informação de sensores localizados em dispositivos móveis, como carros, e depois processar, distribuir e visualizar os dados. Cada carro contém um dispositivo semelhante a um pequeno computador com um conjunto de sensores. O objectivo é obter informação acerca do trânsito de forma a permitir definir rotas alternativas no caso de serem detectados problemas. O processamento da informação recolhida é efectuado localmente antes de ser enviado, utilizando uma rede 802.11, para o Sistema Central onde são guardados para mais tarde serem analisados e visualizados.

Existe um grande número de projectos que usam este tipo de redes em veículos. Um projecto um bocado diferente é o Wifi-Reports[50].

Neste projecto, os participantes que se ligam a *hotspots* comerciais recolhem algumas informações sobre os mesmos, como medições de performance e suporte de aplicações. Com essas informações são construídos relatórios que são enviados para um Sistema Central que vai processar, armazenar e disponibilizar os dados. A visualização dos resultados é a informação histórica da performance e suporte de aplicações dos APs, que irá permitir aos clientes de um *hotspot* comercial a escolha do melhor AP antes de efectuar a compra do acesso. No desenvolvimento do projecto foi tido em conta a privacidade dos relatórios enviados ao Sistema Central e a segurança, através da cifra dos dados.

4.2 Aplicação Epi

O Epi é uma aplicação que se encontra actualmente em desenvolvimento no seio do grupo Ubicomp da Universidade do Minho, no âmbito do projecto SUM (Sensing and Understanding human Motion dynamics).

Esta aplicação proporciona aos utilizadores de redes *WiFi* a troca de mensagens de texto entre utilizadores que se encontrem próximos, mesmo sem conexão à Internet.

As mensagens trocadas num determinado local são armazenadas e novamente difundidas noutros locais para onde o Epi se desloque, criando assim uma espécie de difusão epidémica das mesmas. A Figura 4.1, retirada de [9], pretende ilustrar a arquitectura e o princípio de funcionamento da aplicação Epi.

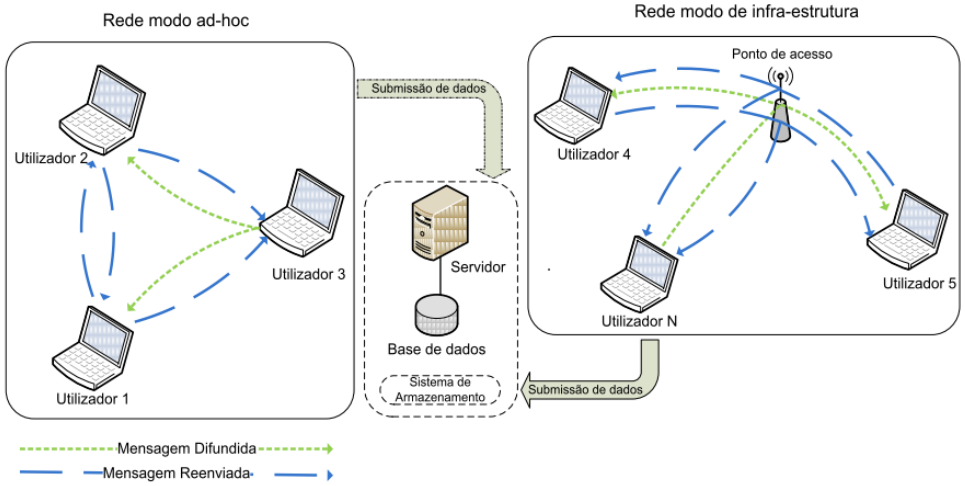


Figura 4.1: Arquitectura de sistema do Epi.

A aplicação não permite enviar mensagens a um utilizador em específico, dado que o envio é feito em *broadcast* na rede IP onde o utilizador se encontra ligado. No processo de troca de mensagens, um utilizador é considerado na vizinhança de outro se obedecer a uma condição estabelecida por uma função de proximidade.

Esta função é uma operação matemática efectuada sobre duas assinaturas[1] de rádio: a assinatura de rádio anexada à mensagem na origem e uma outra assinatura de rádio recolhida no instante de recepção da mensagem. A assinatura de rádio é a lista dos APs *WiFi* que se encontram na vizinhança quando é efectuada a recolha.

[1]Uma assinatura é o conjunto de informações que é recolhido do ambiente de rádio, num determinado instante ou intervalo de tempo. Por exemplo, uma assinatura de estações *WiFi* pode conter várias amostras.

Os utilizadores podem fazer uso das funcionalidades da aplicação quando se encontram ligados a uma rede *WiFi*. O tipo de rede *WiFi* que o utilizador usa é indiferente, podendo estar ligado a uma rede em modo *ad-hoc* (que pode ser criada pelo próprio) ou a uma rede em modo infra-estrutura (normalmente a um AP). Em qualquer um dos casos o utilizador não necessita de ter acesso à Internet.

Quando recebe ou envia uma mensagem, realiza a recolha de uma assinatura de rádio que é armazenada. Mais tarde, quando a aplicação se encontra na presença de uma ligação à Internet, a informação do ambiente de rádio armazenada é enviada para um servidor.

Relativamente à questão da privacidade da informação armazenada e enviada para o servidor, não são recolhidos dados relativos ao conteúdo das mensagens de texto trocadas entre utilizadores. Nem mesmo o nome de utilizador de quem envia uma mensagem é armazenado, no entanto, é recolhido um valor, recorrendo a um função de *Hash*, determinado pelo texto da mensagem. Este valor é usado para detectar a difusão sucessiva da mesma mensagem (efeito epidemia).

A restante informação enviada juntamente com a assinatura de rádio é a informação temporal e o endereço físico da placa de rede onde é efectuada a recolha da assinatura de rádio. Este é o único dado armazenado que pode ser associado ao utilizador.

4.3 Novas funcionalidades

O objectivo desta dissertação, no que concerne à aplicação, é a inclusão de um novo módulo que realize uma recolha (apenas periódica) de um novo tipo de assinatura de rádio. No âmbito desta dissertação, o novo módulo a incluir é um que realize a recolha de informação relativa às estações *WiFi* que se encontrem na vizinhança. A este novo módulo é dado destaque na próxima secção.

A par deste módulo foi também desenvolvido um outro módulo, no âmbito de outra dissertação, que realiza a recolha dos dispositivos Bluetooth.

As informações recolhidas por estes dois módulos, em conjunto com o módulo já existente (módulo de recolha de informação dos APs vizinhos), são periodicamente submetidas para um servidor onde são armazenadas numa base de dados para depois serem consultadas e analisadas.

Os detalhes da arquitectura original da aplicação Epi, da função de proximidade

e do módulo de recolha da informação dos APs na vizinhança, encontram-se no documento de dissertação *Difusão Epidémica de Mensagens em Hotspots WiFi*[9]. Os detalhes do módulo que realiza a recolha de informações relativas aos dispositivos Bluetooth, encontram-se no documento de dissertação *Avaliação da Tecnologia Bluetooth como Sensor da Mobilidade Urbana*[51].

Com a inclusão destes novos módulos na aplicação foi necessário realizar algumas alterações no código da mesma e consequentemente no protocolo de comunicação com o servidor, de forma a suportar os novos tipos de assinaturas de rádio.

No servidor foram também necessárias alterações para suportar as modificações introduzidas pelo novo protocolo e consequentemente na base de dados, para armazenar os dados referentes aos novos tipos de assinaturas de rádio recolhidas.

Uma vez que foram realizadas todas estas alterações na aplicação, achou-se que seria interessante desenvolver uma nova *interface* gráfica. Apesar de se incluírem novos módulos de recolha de informação (neste caso os módulos de recolha Bluetooth e estações *WiFi*), estes são transparentes do ponto de vista do utilizador. Com a nova *interface* gráfica, pretende-se oferecer uma melhor *user experience* e desta forma cativar mais utilizadores a instalar e usar a aplicação. O novo *interface* gráfico foi desenvolvido no âmbito de outra dissertação de mestrado. A janela principal da nova *interface* gráfica está ilustrada na Figura 4.2.

Figura 4.2: Novo ambiente gráfico do Epi.

Também se achou interessante adicionar uma nova funcionalidade que permita a um utilizador saber a lista de utilizadores com os quais interagiu mais vezes. Essa

lista é armazenada na estrutura de armazenamento de "vizinhos" (EAV).

Para suportar esta nova funcionalidade foi necessário adicionar um novo campo às mensagens do protocolo usado na comunicação entre clientes. Este novo campo é o endereço MAC da placa de rede dos utilizadores, pois só assim é possível identificar com que utilizadores interagiu cada utilizador.

Depois das alterações efectuadas ao projecto Epi, a aplicação apresenta a estrutura representada pelo diagrama de blocos da Figura 4.3, em que as alterações efectuadas estão assinaladas a **negrito**. Pode-se observar que a aplicação continua a ser controlada, tendo como "pivô", o Gestor principal. Este bloco é o responsável por fazer a ligação entre os diferentes *interfaces* da aplicação, o *interface* do utilizador, o *interface* de rede (*WiFi*) e ainda o novo *interface* Bluetooth. Este último foi o *interface* onde foi realizada a inclusão do novo módulo de recolha de assinaturas de rádio Bluetooth. O módulo de recolha de assinaturas de rádio de estações *WiFi* foi incluído no *interface* de rede, uma vez que usa o mesmo *interface* *WiFi* que os restantes.

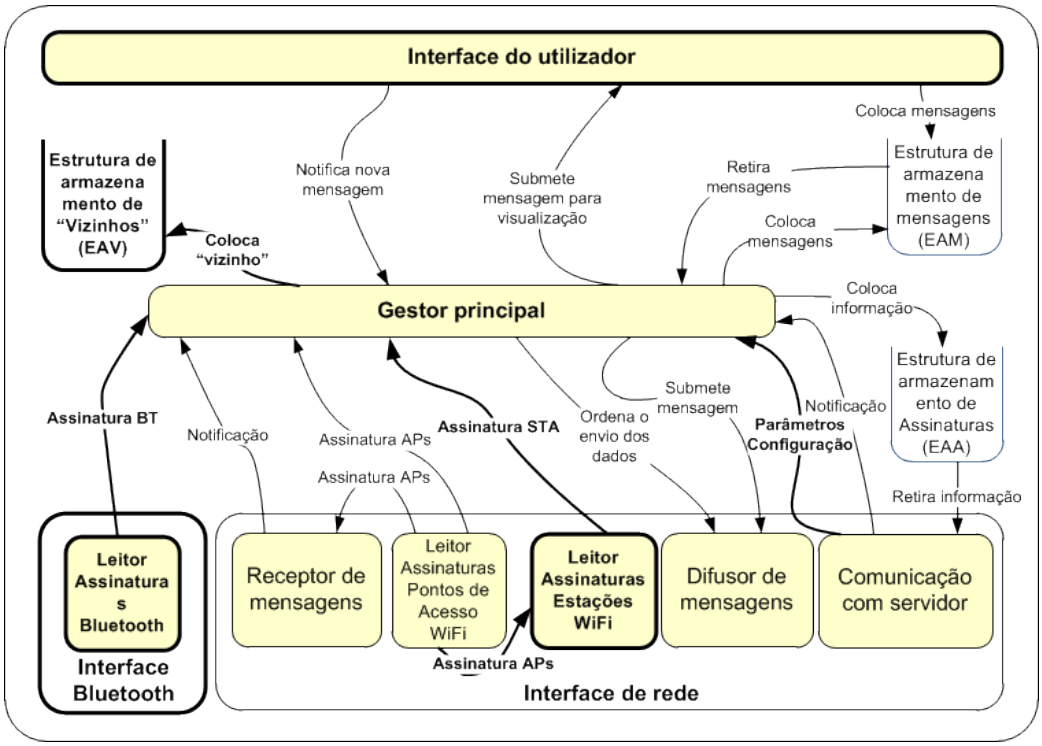


Figura 4.3: Diagrama de blocos da aplicação Epi.

Ao iniciar a aplicação, o Gestor principal é responsável por iniciar um conjunto de objectos, a *thread* que vai receber as comunicações de outros clientes e ainda três

temporizadores. Os temporizadores são responsáveis por periodicamente chamar as funções de difusão das mensagens armazenadas, de comunicação com o servidor e de recolha de assinaturas de rádio dos APs na vizinhança.

De forma a chamar os novos módulos foram adicionados dois novos temporizadores no início de execução da aplicação que periodicamente chamam uma função responsável pela recolha das novas assinaturas de rádio. A informação das recolhas é depois armazenada na estrutura de assinaturas (EAA).

O período de tempo predefinido para cada temporizador está representado na Tabela 4.1, no entanto, estes podem ser alterados numa das operações que acontecerem entre cliente e servidor. Sempre que um cliente comunica com o servidor efectua três operações distintas. Primeiro, verifica se existe uma nova versão da aplicação disponível. Segundo, verifica os parâmetros de configuração. Por fim, verifica se tem assinaturas de rádio armazenadas, que em caso positivo são imediatamente enviadas uma a uma para o servidor. As três operações são efectuadas pela ordem respectiva sempre que é aberta uma ligação ao servidor, se alguma delas falha, as seguintes já não acontecem. A comunicação com o servidor é realizada periodicamente ou quando o utilizador verifica se existe uma actualização da aplicação disponível.

Temporizador	**Primeira execução (seg.)**	**Período próximas execuções (seg.)**
Módulo AP	8	1800
Módulo Sta *WiFi*	15	1800
Módulo Bluetooth	10	1800
Comunicação Servidor	5	3600
Difusão Mensagens	900	900

Tabela 4.1: Valores de tempo predefinidos nos temporizadores.

A operação do cliente contactar o servidor para verificar os parâmetros de configuração impostos pelo servidor é também uma nova funcionalidade adicionada. Esta também é transparente para o utilizador. Os parâmetros de configuração incluem o período de tempo dos temporizadores, a activação e desactivação dos novos módulos e o valor de configuração da função de proximidade. Para a função de proximidade, os valores predefinidos mantêm-se a 20[9] e os novos módulos são activados.

4.4 Servidor

O servidor do Epi é um Servlet[1] a correr num servidor *Apache Tomcat* que recebe pedidos HTTP POST dos clientes para efectuar operações específicas. O servidor identifica a interacção que o cliente quer efectuar através do valor do campo de função ("f") no corpo do pedido POST.

De forma a suportar a recepção de novas assinaturas e a verificação dos parâmetros de configuração, foram adicionadas três novas funções ao servidor às duas funções "f1" e "f2" que já existiam[9]. As funções "f3" e "f4" para a recepção de assinaturas de rádio Bluetooth e assinaturas de rádio das estações *WiFi*, respectivamente. A função "f5" para efectuar a verificação dos parâmetros de configuração.

Os pedidos HTTP POST para cada uma das novas funções é processado da seguinte forma:

Função f3

O corpo POST neste pedido enviado ao servidor deverá incluir os campos:

f - identifica a função que se pretende invocar, que neste caso é "f3".

ats - instante de tempo no cliente em que o pedido HTTP POST é criado, o formato é do tipo "yyyy:MM:dd:HH:mm:ss;$\pm xx : zz$", em que "yyyy" representa o ano, "MM" representa o mês, "dd" representa o dia, "mm" representa os minutos, "ss" representa os segundos e "$\pm xx : zz$" representa o desfasamento horário existente em várias zonas do mundo.

sgt - identifica o tipo de assinatura de rádio a que se refere o pedido, "1" para a assinatura de rádio dos APs, "2" para a assinatura de rádio Bluetooth e "3" para a assinatura de rádio das estações *WiFi*.

sts - instante de tempo em que a assinatura rádio foi recolhida, o formato é do tipo "yyyy:MM:dd:HH:mm:ss", em que "yyyy" representa o ano, "MM" representa o mês, "dd" representa o dia, "mm" representa os minutos, "ss" representa os segundos.

clm - endereço MAC da máquina de onde a mensagem foi enviada, o formato é "XX:XX:XX:XX:XX:XX", onde "XX" representa um byte em formato hexadecimal (00 a FF).

[1] É um módulo que estende a funcionalidade de um servidor Web através de módulos de aplicação implementados em Java.

has - transporta o *Hash* da mensagem enviada ou recebida. É representado por uma *string* com 32 caracteres hexadecimais. No caso do pedido não se referir a uma mensagem este campo deve conter a string "nomsgnomsgnomsgnomsgnomsgnomsgno".

mts - instante de tempo em que a mensagem foi criada, o formato é do tipo "yyyy:MM:dd:HH:mm:ss;$\pm xx$: zz", em que "yyyy" representa o ano, "MM" representa o mês, "dd" representa o dia, "mm" representa os minutos, "ss" representa os segundos e "$\pm xx$: zz" representa o desfasamento horário existente em várias zonas do mundo.

mac - lista com os dispositivos Bluetooth. Este campo será representado por uma *string* com a lista de endereços MAC, separados pelo caracter ";". Cada endereço MAC deve ser transmitido na forma "XX:XX:XX:XX:XX:XX", onde "XX" representa um byte em formato hexadecimal (00 a FF).

dev - lista com o nome dos dispositivos Bluetooth, separados pelo caracter ";".

tds - lista com o instante de tempo de descoberta de cada dispositivo, separados pelo caracter ";". O formato de cada instante de tempo é do tipo "yyyy:MM:dd:HH:mm:ss", em que "yyyy" representa o ano, "MM" representa o mês, "dd" representa o dia, "mm" representa os minutos e "ss" representa os segundos.

cls - lista com o nome das classes dos dispositivos, separados pelo caracter ";".

srv - lista com o nome dos serviços disponibilizados por cada dispositivo, separados pelo caracter ";". Para cada dispositivo os serviços encontram-se separados pelo caracter ",".

rsi - lista com os valores de RSSI (Received Signal Strength Indicator) de cada dispositivo, separados pelo caracter ";". Este campo será representado por uma *string*, onde caso o seu valor seja igual a "Null" indica que esse dispositivo foi o que realizou o *scan* dos restantes.

A este pedido o servidor deve responder com uma das seguintes mensagens:

001;record_id - significa que o pedido foi aceite e que os respectivos dados foram armazenados na base de dados com o número do registo representado por *record_id*.

100;description - significa que o pedido não foi aceite devido a algum erro interno do servidor, o erro é descrito por *description*.

101 - significa que o pedido não respeita o formato definido e que por isso os dados não foram armazenados na base de dados.

102;description - significa que o pedido respeita o formato definido mas que não foi possível armazenar estes dados na base de dados, sendo o erro descrito por *description*.

Função f4

f - identifica a função que se pretende invocar, que neste caso é "f4".

ats*, *sgt*, *sts*, *clm*, *has* e *mts - estes campos nesta função são comuns aos campos da função "f3" descrita acima.

ngm - lista com as estações *WiFi* detectadas. Este campo será representado por uma *string* com a lista de endereços MAC, separados pelo caracter ";". Cada endereço MAC deve ser transmitido na forma "XX:XX:XX:XX:XX:XX", onde "XX" representa um byte em formato hexadecimal (00 a FF).

Para este pedido o servidor deve responder com mensagens que são iguais às da função "f3".

Função f5

O corpo POST neste pedido enviado ao servidor inclui apenas o campo de função "f" com o valor "f5". A este pedido, o servidor deverá responder um conjunto de parâmetros separados por "#", em que cada parâmetro é constituído pelo seu nome e valor, como ilustra a Figura 4.4. Os nomes dos parâmetros introduzidos nesta versão reconhecidos pelo cliente são os seguintes:

simthreshold - Valor da métrica associado à função de proximidade.

mod_wifista - Activação/Desactivação do módulo de recolha de assinaturas de rádio de estações *WiFi*, "enable" para activar e "disable" para desactivar.

mod_bt - Activação/Desactivação do módulo de recolha de assinaturas de rádio Bluetooth, "enable" para activar e "disable" para desactivar.

int_commsrv - Valor em segundos para o temporizador que chama a função para comunicação com o servidor.

intscan_wifista - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio de estações *WiFi*.

intscan_bt - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio Bluetooth.

intscan_wifiaps - Valor em segundos para o temporizador que chama a função de recolha de assinaturas de rádio das redes *WiFi*.

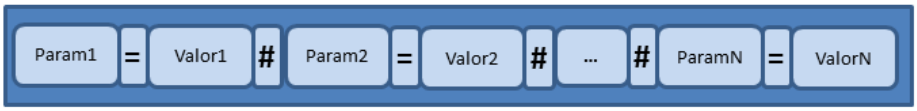


Figura 4.4: Resposta à função "f5".

Quando o cliente consegue estabelecer comunicação com o servidor, nas operações de verificação da versão da aplicação e dos parâmetros de configuração, o servidor faz consultas à base de dados onde é guardada essa informação. Na operação em que o cliente envia uma a uma as assinaturas de rádio que estão armazenadas na EAA, o servidor recebe a informação de cada assinatura, verifica-a e depois armazena-a na base de dados.

A base de dados também sofreu alterações para suportar o armazenamento das novas assinaturas e parâmetros de configuração. O modelo da base de dados encontra-se ilustrado na Figura 4.5.

Como se pode ver na figura, a base de dados é composta por um conjunto de seis tabelas. A tabela *clientversions* armazena informação relativa á versão da aplicação do cliente que os utilizadores usam. Já a tabela *epiconfig* armazena informação relativa aos parâmetros de configuração da aplicação de cliente. Esta foi construída de forma a manter um histórico das configurações introduzidas pelo administrador do lado do servidor. As tabelas *btsignatires*, *wifistasignatures* e *wifiapsignatures* armazenam as assinaturas de rádio que foram recolhidas pelos três módulos na aplicação cliente.

Por fim, *messages* é a tabela que guarda as informações relativas às mensagens trocadas pelos utilizadores, que servem como suporte para o envio das informações de recolha das assinaturas de rádio. Nesta tabela existem dois campos importantes:

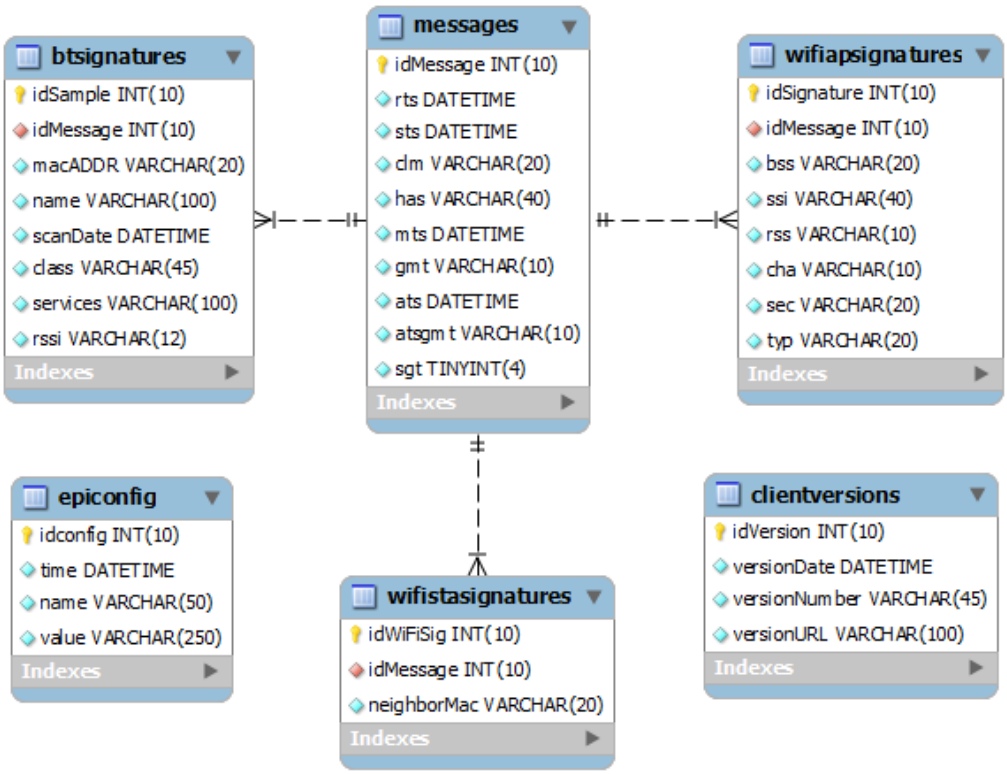


Figura 4.5: Tabelas e relacionamento na base de dados do servidor.

o *idMessage* e o *has*. O primeiro campo, para além de chave primária nos registos desta tabela, serve para referenciar nas tabelas *btsignatures*, *wifistasignatures* e *wifiapsignatures* qual a mensagem que enviou a respectiva assinatura. O campo *has* contém o *hash* do texto da mensagem que originou a recolha da assinatura de rádio. O conteúdo do campo também permite identificar se a assinatura de rádio é proveniente de uma mensagem ou de uma recolha periódica.

4.5 Protocolos de comunicação

Nas mensagens enviadas entre clientes, o endereço de destino dos pacotes é o endereço de *broadcast* e o protocolo de transporte usado é o UDP. Na Figura 4.6 está representado o formato de mensagem enviada ao nível de aplicação.

O campo *Sender MAC* contém o endereço MAC da placa de rede do utilizador que enviou a mensagem. Este campo é a única alteração efectuada ao protocolo original[9].

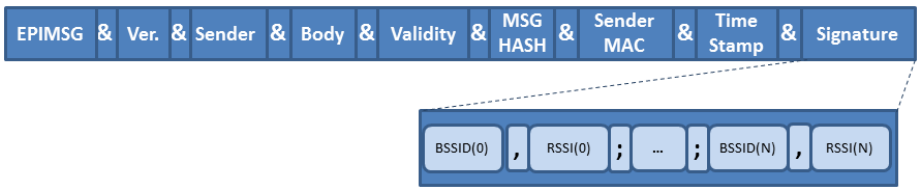


Figura 4.6: Mensagens enviadas entre clientes Epi.

A troca de mensagens entre cliente e servidor é diferente, o protocolo usado no nível de transporte é o TCP. No nível de aplicação, como se trata de um servidor web são usados pedidos HTTP POST. A Listagem 4.1 mostra o exemplo do envio de uma assinatura de rádio de estações *WiFi*, do cliente ao servidor.

```
1  Cliente -> Servidor:
2      POST /epiServer2/epiS HTTP/1.1
3      Content-Type: application/x-www-form-urlencoded
4      Host: epi.dsi.uminho.pt:8080
5      Content-Length: 389
6      Expect: 100-continue
7
8  Servidor -> Cliente:
9      HTTP/1.1 100 Continue
10
11 Cliente -> Servidor:
12     f=f4&
13     ats=2011%3a10%3a13%3a18%3a42%3a22%3b%2b01%3a00&
14     sgt=3&
15     clm=1c%3a4b%3ad6%3abc%3a30%3ad4&
16     has=nomsgnomsgnomsgnomsgnomsgnomsgno&
17     mts=2011%3a10%3a13%3a17%3a14%3a25%3b%2b01%3a00&
18     sts=2011%3a10%3a13%3a17%3a14%3a25&
19     ngm=90%3a4c%3ae5%3a8f%3a5d%3a48%3b00%3a23%3a6c%3a89%3a78%3a29
20     %3b00%3a17%3ac4%3a87%3a24%3a50%3b70%3a1a%3a04%3a07%3a16%3a5e
21     %3b00%3a25%3ad3%3a64%3a22%3a68%3b1c%3a4b%3ad6%3abc%3a30%3ad4
22
23 Servidor -> Cliente:
24     HTTP/1.1 200 OK
25     Server: Apache-Coyote/1.1
26     Content-Type: text/html
27     Content-Length: 10
28     Date: Thu, 13 Oct 2011 17:42:25 GMT
```

```
29      001;2684
```

Listagem 4.1: Envio de uma assinatura de rádio do cliente para o servidor.

A informação enviada pelo cliente ao servidor (linhas 12-21) não tem uma ordenação específica. Trata-se apenas de uma colecção de informação com pares nome - valor. A colecção de dados esperada pelo servidor varia consoante a função, como já foi descrito na Secção 4.4.

Capítulo 5

Módulo WifiRadar

Neste capítulo é descrita a especificação e desenvolvimento da solução que foi integrada na aplicação Epi como um módulo de *software*.

Também é descrito como foi criado o pacote de instalação da aplicação Epi e a integração do Network Monitor nesse pacote.

5.1 Considerações na implementação da solução

A solução encontrada para a detecção de estações *WiFi* na vizinhança, descrita na Secção 3.3, foi integrada como um módulo de *software* na aplicação Epi com o nome WifiRadar. Como base do módulo foi usada a aplicação de testes descrita na Secção 3.3.2.

Ao integrar o WifiRadar foi necessário ter em conta alguns aspectos importantes que na aplicação de teste não foram considerados. Esses aspectos foram os seguintes:

- Tem que funcionar de forma que seja completamente transparente para o utilizador que usa a aplicação. O utilizador também não têm, nem necessita de ter, qualquer interacção com o WifiRadar.

- A escolha do *interface* de rede na aplicação de teste era efectuado pelo utilizador, no WifiRadar a escolha é automática através da verificação de um conjunto de condições.

- Efectuar a verificação do SO que o utilizador está a usar, uma vez que esta solução apenas é válida para as versões Vista, 7, 8 e Server 2008 do Windows. A

aplicação Epi deverá funcionar em SOs Windows mais antigos, logo o WifiRadar foi desenhado com a capacidade de se desactivar quando o SO não é suportado.

- A inicialização da API do Network Monitor só deve ser realizada uma vez por cada vez que a aplicação é iniciada. Na documentação do Network Monitor fazem essa recomendação senão podem acontecer erros não previstos. Na aplicação de teste não havia essa consideração, em cada procura a aplicação era executada novamente. Na aplicação Epi pretende-se que a aplicação fique a correr e as procuras sejam efectuadas em intervalos regulares sem a necessidade de fechar e voltar a abrir a aplicação.

- A duração da procura por estações também deve ter um tempo fixo, na aplicação de teste também era controlado pelo utilizador manualmente.

- A solução necessita da lista de redes *WiFi* na vizinhança, o que na aplicação de teste era realizado através da execução do "netsh". A aplicação Epi já contém um módulo que faz a procura de redes *WiFi* na vizinhança e, para não criar redundância de funções, o WifiRadar tem a capacidade de receber a lista de redes *WiFi* no formato usado pelo módulo que faz essa função.

- Os resultados obtidos numa procura, na aplicação de teste, são escritos em HTML. No WifiRadar os resultados são retornados para a aplicação usando uma *string*. O conteúdo da *string* é a lista de endereços MAC detectados, que é a única informação necessária.

- O WifiRadar pode ser activado e desactivado pelo servidor de forma remota, o que é feito pelo Gestor Principal da aplicação Epi.

- O período entre descobertas das estações *WiFi* na vizinhança é controlado por um temporizador, que por sua vez é controlado pelo Gestor Principal da aplicação Epi.

- Todas as saídas para a consola foram eliminadas ou redireccionadas para um ficheiro de *log*. Com o *log*, pretendeu-se manter uma forma de observar a saída em pontos importantes e desta forma conseguir dar suporte a problemas relacionados com o WifiRadar.

- Foi necessário ter em conta verificações adicionais, dados de entrada e saída e controlo de erros de forma a não causar erros que façam a aplicação Epi deixar de responder.
- Optimizar de forma a consumir o mínimo de recursos disponíveis.

O módulo WifiRadar foi desenvolvido como uma biblioteca que pode ser integrada em qualquer aplicação. Desta forma permanece como uma solução separada, o que traz algumas vantagens. Pode ser integrado na aplicação Epi e noutras aplicações, no entanto, este foi desenvolvido tendo em conta a integração na aplicação Epi e algumas funções foram adaptadas especificamente para este. Pode mais tarde, se houver necessidade, ser actualizado de forma separada da aplicação.

5.2 Implementação do WifiRadar

A implementação da solução resultou no módulo WifiRadar e a sua arquitectura encontra-se ilustrada na Figura 5.1.

O WifiRadar é constituído pelo módulo principal *WifiMon*, que é o "core" do módulo e por onde as interacções com o Gestor Principal do Epi (podia ser outra aplicação) são efectuadas. Os módulos *AP*, *Station* e *Adaptor* definem os atributos e métodos do tipo de objecto que representam. O módulo *NetmonAPI* é o Wrapper que faz a interacção com o Network Monitor.

O módulo *WifiMon* pode ser divido em vários sub-módulos em que cada um é responsável por um conjunto de operações:

1. inicialização do *WifiMon*.
2. registo de eventos.
3. estado do *WifiMon*.
4. inicialização do Network Monitor.
5. recepção da assinatura do *LeitorAssinaturas*.
6. descoberta de estações e resultados.
7. análise sintáctica das tramas e processamento dos endereços.

Na Figura 5.1 também estão ilustradas as interacções que existem entre os métodos do *WifiMon*. Este ao ser inicializado também cria um conjunto de atributos que são usados que não estão ilustrados na figura (por questões de representação visual) e são também importantes na interacção entre os métodos.

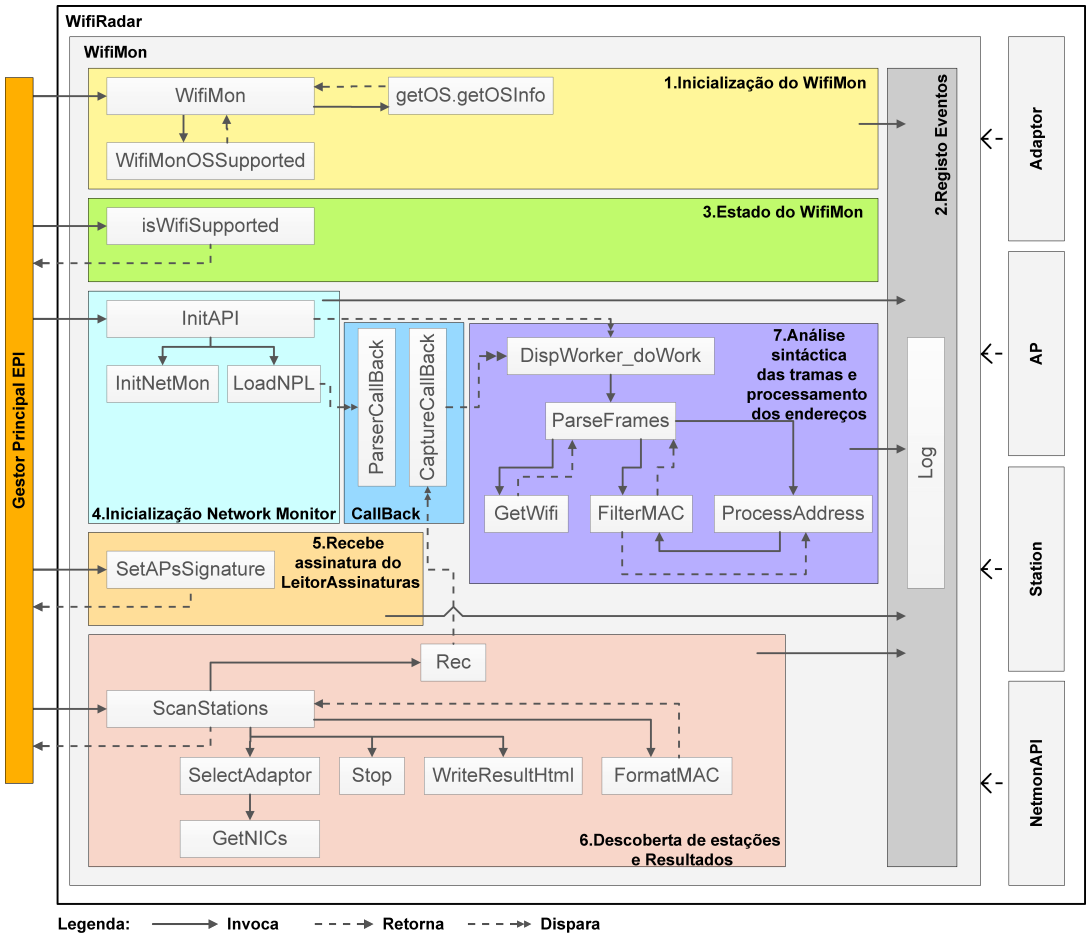


Figura 5.1: Esquema geral do WifiRadar.

Cada um dos sub-módulos é detalhado nesta secção. Também serão detalhados outros aspectos considerados importantes para perceber o funcionamento do WifiRadar.

5.2.1 Classes

A implementação do módulo WifiRadar envolveu um conjunto de classes de forma a satisfazer os requisitos pretendidos. Na Figura 5.2 está ilustrado, de forma básica

e simplificada, o diagrama de classes do WifiRadar. Os atributos e métodos de cada classe encontram-se representados em detalhe no Anexo A.

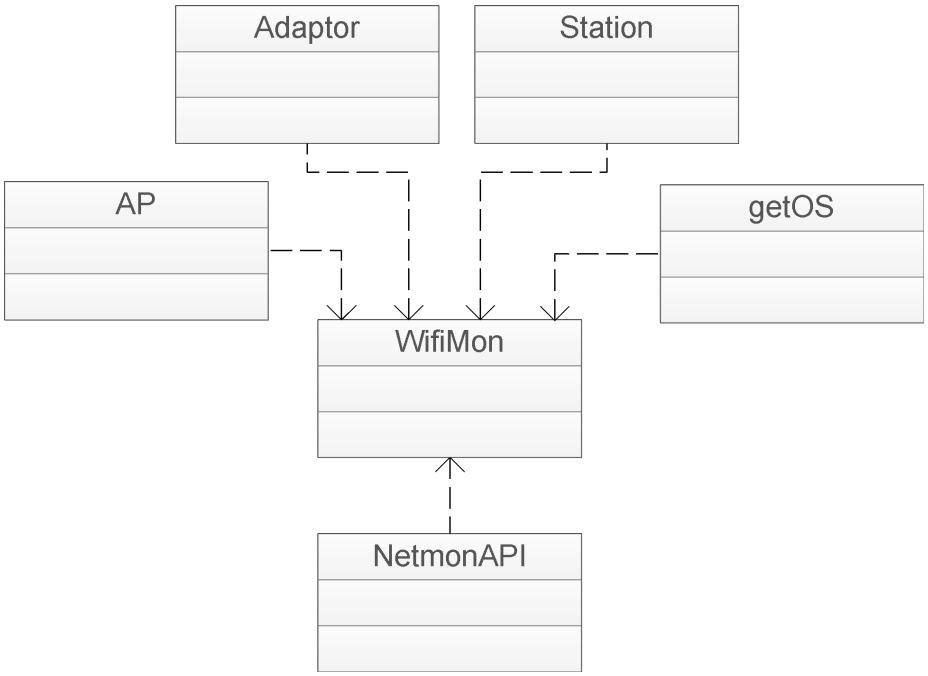


Figura 5.2: Diagrama de classes do WifiRadar.

As seis classes que constituem o módulo WifiRadar são as seguintes:

Classe *AP* - esta classe serve para definir os atributos e métodos de um objecto do tipo *AP*. No WifiRadar as propriedades de cada rede *WiFi* (BSSID, SSID, tipo de rede, etc) encontram-se armazenadas nos atributos de um objecto deste tipo. Os métodos desta classe são um par *Set*/*Get* para cada propriedade. Todos os atributos e métodos desta classe estão ilustrados na Figura A.3 do Anexo A.

Classe *Adaptor* - esta classe serve para definir os atributos e métodos de um objecto do tipo *Adaptor*. No WifiRadar as propriedades de cada placa de rede (descrição, índice, etc.) que pode ser usada para a captura de tráfego, encontram-se armazenadas nos atributos de um objecto deste tipo. Os métodos desta classe são *Get* para cada propriedade. Todos os atributos e métodos desta classe estão ilustrados na Figura A.1 do Anexo A.

Classe *Station* - esta classe serve para definir os atributos e métodos de um objecto do tipo *Station*. No WifiRadar as propriedades de cada estação detectada na vizinhança (endereço MAC, tempo, listas de tipos tramas *WiFi*, etc.)

encontram-se armazenadas nos atributos de um objecto deste tipo. Os métodos desta classe são *Set* e *Get* e alguns dos atributos simples, no caso das listas existem funções mais complexas para efectuar algumas operações sobre a mesma. Todos os atributos e métodos desta classe estão ilustrados na Figura A.4 do Anexo A.

Classe *getOS* - esta classe contém apenas métodos que são usados pelo WifiRadar para consultar informação acerca do SO. Todos os métodos desta classe estão ilustrados na Figura A.2 do Anexo A. Esta classe foi fornecida por terceiros[52].

Classe *WifiMon* - esta classe serve para definir os atributos e métodos de um objecto do tipo *WifiMon*. Esta é a classe principal do WifiRadar que armazena todas as suas propriedades e contém os métodos que controlam todas as suas operações. Os atributos e métodos desta classe estão ilustrados nas Figuras A.5 e A.6 do Anexo A.

Classe *NetmonAPI* - esta classe é o *Wrapper* fornecido pelo Network Monitor que é usado em todas as interacções com a sua API para C++.

5.2.2 Inicialização e estado do WifiRadar

A inicialização do WifiRadar acontece quando é construído um novo objecto *WifiMon*. Nesta altura são criados e inicializados todos os atributos que fazem parte da classe.

O construtor do *WifiMon* tem como requisito que lhe sejam fornecidos três parâmetros:

path - *string* que representa o caminho para a pasta onde estão os ficheiros de dados da aplicação.

loglevel - inteiro que representa o nível para o registo de eventos.

writerfile - indica se os resultados devem ser escritos também para um ficheiro HTML.

Depois de definir os parâmetros que recebeu e o ficheiro de saída do registo de eventos, começa por escrever no registo de eventos informação relativa à sua inicialização e

chama o método *getOSInfo* da classe *getOS* para obter a *string* com a informação do SO para também usar no registo de eventos.

Ao prosseguir vai chamar o método *WifiMonOSSupported* que avalia as versões do SO e decide se o WifiRadar é suportado. O método retorna "0" quando é suportado, "1" quando não é suportado e "2" quando não foi possível determinar. Na Tabela 5.1 estão representadas as versões dos SOs suportados. A avaliação é realizada pela plataforma e depois pelo valor do *MajorVersion*.

Plataforma	**MajorVersion**	**Sistema Operativo**	**Suportado**
Win32Windows	-	Windows 95, 98, 98SE and Me	
Win32NT	3	Windows NT 3.51	
	4	Windows NT 4.0	
	5	Windows 2000, XP e Server 2003	
	6	Windows Vista, Server 2008, 7 e 8	√

Tabela 5.1: Sistemas Operativos suportados pelo WifiRadar.

O estado em que se encontra o WifiRadar pode ser consultado através do método *isWifiSupported*, que retorna "true" se está activado ou "false" caso contrário. Os métodos públicos do WifiRadar não realizam nenhuma operação quando este se encontra desactivado, pelo que é recomendado a consulta deste método antes de prosseguir (como é efectuado no Epi).

A fase de inicialização termina depois da verificação do SO e do registo deste evento.

5.2.3 Registo de eventos

O registo de eventos serve para escrever para um ficheiro eventos relevantes que acontecem no WifiRadar. O objectivo é utilizar depois o ficheiro obtido para auditoria e diagnóstico de problemas que possam acontecer dentro do WifiRadar.

O registo de eventos usa o ficheiro "wifistalog.txt" para escrever as entradas. Para cada entrada também se escreve a data e hora em que aconteceu (ex: "17:23:29 terça-feira, 27 de Setembro de 2011").

Durante a execução do WifiRadar, o tamanho do ficheiro do registo de eventos podia tornar-se muito grande se todos os eventos fossem guardados. Em algumas situações, como na integração no Epi, não fazia sentido registar todos os eventos uma vez que era informação não relevante para diagnosticar algum problema e também por uma questão de poupar recursos.

Desta forma, o registo de evento é feito por níveis, em que cada nível inclui o anterior. Por exemplo, ao activar o WifiRadar para o nível 2 (como predefinido no Epi) este vai incluir também o nível 1. Na Tabela 5.2 estão representados os níveis e que eventos são registados.

Nível	Eventos Registados
0	desactivado
1	erros
2	funções, acções chave e detecção estações
3	reservado para uso futuro
4	resultado da análise sintáctica de todas tramas
5	resultado da análise de cada endereço MAC

Tabela 5.2: Níveis do registo de eventos.

O registo de eventos é feito pelo método *Log*, que tem como parâmetro uma *string*, que é o texto referente ao evento, e um inteiro que representa o nível da mensagem. O método é usado para registar eventos em quase todos os outros métodos no *WifiMon*.

5.2.4 Inicialização do Network Monitor

A inicialização da API do Network Monitor é feita pelo método *InitAPI*, que é idêntica à aplicação de teste descrita na Secção 3.3.2. É criada uma configuração da API onde vai entrar o modelo de *Threading*, que é "Multi-threaded" por predefinição no C#.net e nos SOs onde o WifiRadar é suportado.

Ao prosseguir, chama o método *InitNetMon* que abre o motor de captura de tráfego e vai buscar as placas de rede em que o *driver* do Network Monitor está activo.

A seguir, chama o método *LoadNPL* que vai realizar um conjunto de operações relacionadas com a API:

1. indica à API o apontador para o método *ParserCallback* que vai disparar quando acontecem avisos, erros e mensagens no motor de análise sintáctica das tramas (*parsing*).

2. faz o carregamento do ficheiro NPL ("sparser.npl") para o motor de *parsing*, que vai estar na pasta de dados da aplicação.

3. cria a configuração para a análise sintáctica das tramas pelo motor de *parsing* e adiciona os campos, propriedades e filtros à configuração. Estes já foram

representados anteriormente na aplicação de teste na Tabela 3.15. A descrição dos campos, propriedades e filtros também já foi realizada e está na Secção 3.3.2.

4. cria o motor de *parsing* com a configuração efectuada no passo anterior.

A última operação do *InitAPI* é configurar a *thread* dedicada às operações de processamento de cada trama de dados capturada em *background* de forma assíncrona. O método *DispWorker_doWork* é configurado para ser executado nessa *thread*. O método dispara a execução da operação em *background* imediatamente, no entanto o método *DispWorker_doWork* bloqueia a própria *thread* e espera até receber sinal para prosseguir.

A variável que guarda o estado da inicialização do Network Monitor é actualizada para "0" ao chegar a este ponto, terminando aqui esta operação.

Durante todo este processo de inicialização do Network Monitor são enviados para o registo de eventos erros e eventos relevantes que permitam diagnosticar problemas com a API, *driver* e *parsing*.

5.2.5 Lista de redes

O WifiRadar ao efectuar o processamento das tramas que são capturadas, precisa de manter uma lista das redes *WiFi* presentes na vizinhança no momento de captura. A informação das BSSIDs, permite determinar para alguns tipos de tramas do 802.11 que apenas contêm dois endereços, o endereço da estação que se encontra na vizinhança através da filtragem dos endereços de BSSID dos endereços TA ou RA.

No processamento das tramas, novas BSSID são reconhecidas e adicionadas à lista, no entanto, esse processo pode demorar mais tempo a convergir, principalmente em descobertas de curta duração, como na aplicação Epi. Por isso, o WifiRadar fornece o método *SetAPsSignature* que permite receber a lista de redes *WiFi* no formato usado pelo *LeitorAssinaturas* do Epi[9].

A aplicação de teste incluía um método para ir buscar ao SO as redes *WiFi* através do "netsh", no entanto, em versões mais antigas dos SOs suportados pelo WifiRadar causava alguns problemas e por isso optou-se por implementar este método.

Para perceber melhor o funcionamento deste método, o seu fluxograma está ilustrado na Figura 5.3.

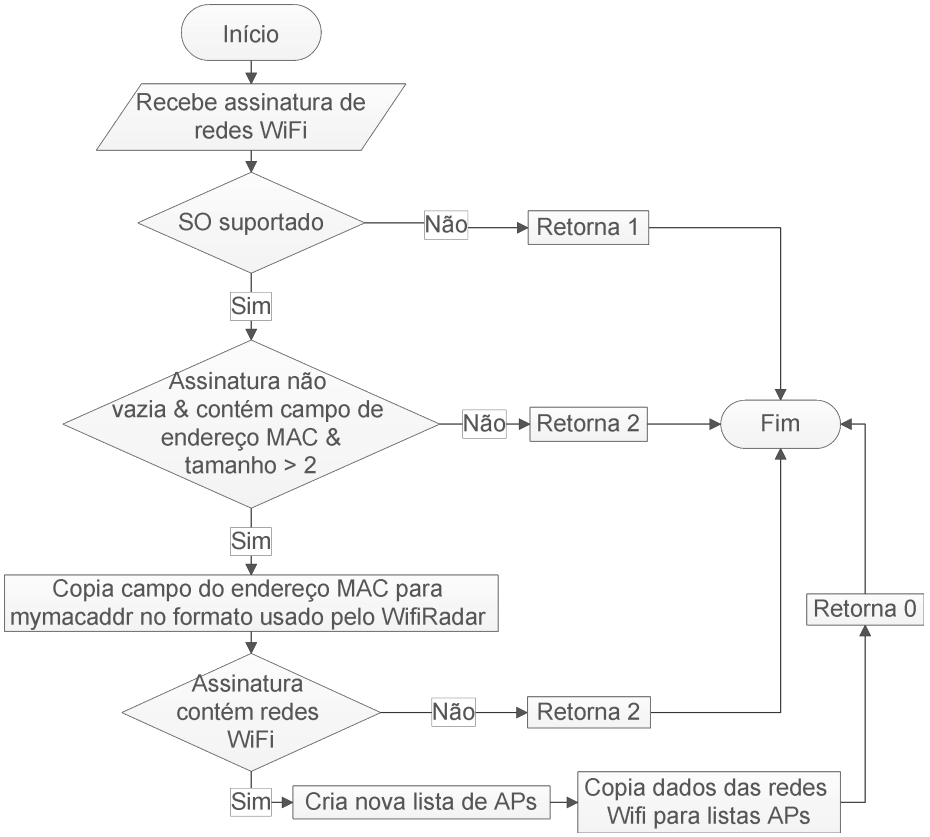


Figura 5.3: Fluxograma do método que recebe a assinatura com a lista de redes.

O método recebe a assinatura e verifica se está no formato correcto. Se está tudo correcto, cria uma nova lista de objectos *AP* e copia os dados de cada uma das redes *WiFi* para um objecto *AP* e adiciona à lista.

Durante este processo também são enviados para o registo de eventos erros e eventos relevantes.

5.2.6 Selecção da placa de rede

A selecção da placa de rede sem fios no qual é efectuada a captura de tráfego é feita de forma automática mediante algumas condições. Essas condições dependem do número de placas de rede disponíveis e de que tipo são.

O método *SelectAdaptor* é responsável por esta operação. Este é chamado dentro do método *ScanStations* quando é pedido o começo da descoberta de estações. Como este processo é vital para a descoberta de estações, este é descrito aqui fora do sub-módulo a que pertence.

O fluxograma que está ilustrado na Figura 5.3 e mostra como o processo de selecção da placa de rede sem fios é realizado.

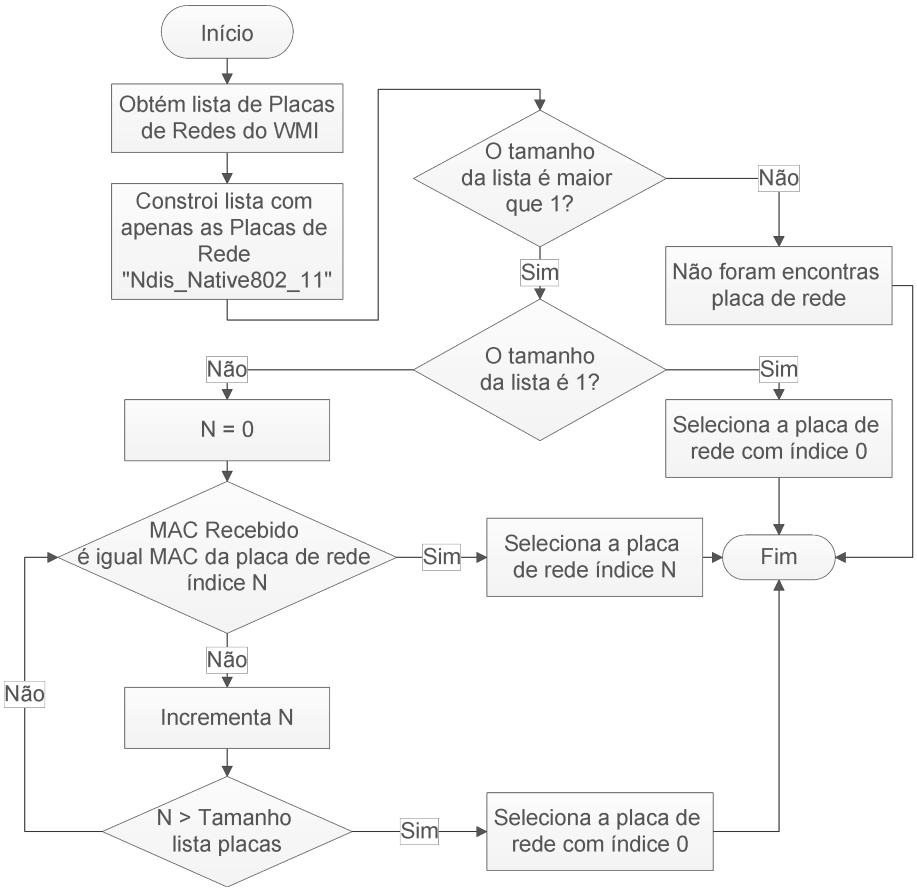


Figura 5.4: Fluxograma do método de selecção da placa de rede.

Este processo posso ser descrito pelos seguintes passos:

1. Começa por chamar o método *GetNICs*, que é o mesmo método usado na aplicação de teste. Este vai buscar a listas das placas de redes ao WMI do SO.

2. Com a lista das placas de redes obtidas no processo de inicialização do Network Monitor e a lista obtida no passo 1, é realizada a convergência das duas listas em apenas uma. Depois são filtradas as placas de redes sem fios que sejam *NativeWiFi* ("Ndis_Native802_11").

3. Selecciona uma placa ou não da seguinte forma:

- caso não existam placas de redes sem fios ou placas compatíveis, o método termina.
- se apenas existe uma placa de rede sem fios compatível, selecciona essa placa.
- caso existam múltiplas placas de rede sem fios compatíveis, selecciona a que tem o mesmo endereço MAC que foi usado pelo *LeitorAssinaturas* na descoberta de redes *WiFi*.
- caso não tenha sido obtido o endereço MAC usado no *LeitorAssinaturas*, é seleccionada a primeira placa de rede sem fios encontrada na lista.

Durante este processo também são enviados para o registo de eventos erros e eventos relevantes.

5.2.7 Descoberta das estações

A operação de descoberta das estações *WiFi* começa ao chamar o método *ScanStations*, que recebe como parâmetros a duração da descoberta (em segundos) e a referência para a *string* ao qual se pretende atribuir os resultados no final do método.

O fluxograma que está ilustrado na Figura 5.5 descreve o funcionamento deste método.

Começa por ser chamado o método *SelectAdaptor* para realizar a selecção da placa que já foi descrita antes na Secção 5.2.6.

Depois é necessário realizar três condições para que a descoberta de estações possa continuar:

- o WifiRadar tem que estar activado.
- a API do Network Monitor tem que se encontrar inicializada com sucesso (estado a "0").
- tem que existir uma placa de rede sem fios seleccionada para efectuar a captura de tráfego.

Com estas condições satisfeitas, é criada uma nova lista de objectos *Station* que vai armazenar todos os dados referentes às estações sem fios detectadas na vizinhança. O contador do número de estações também é igualado a "0".

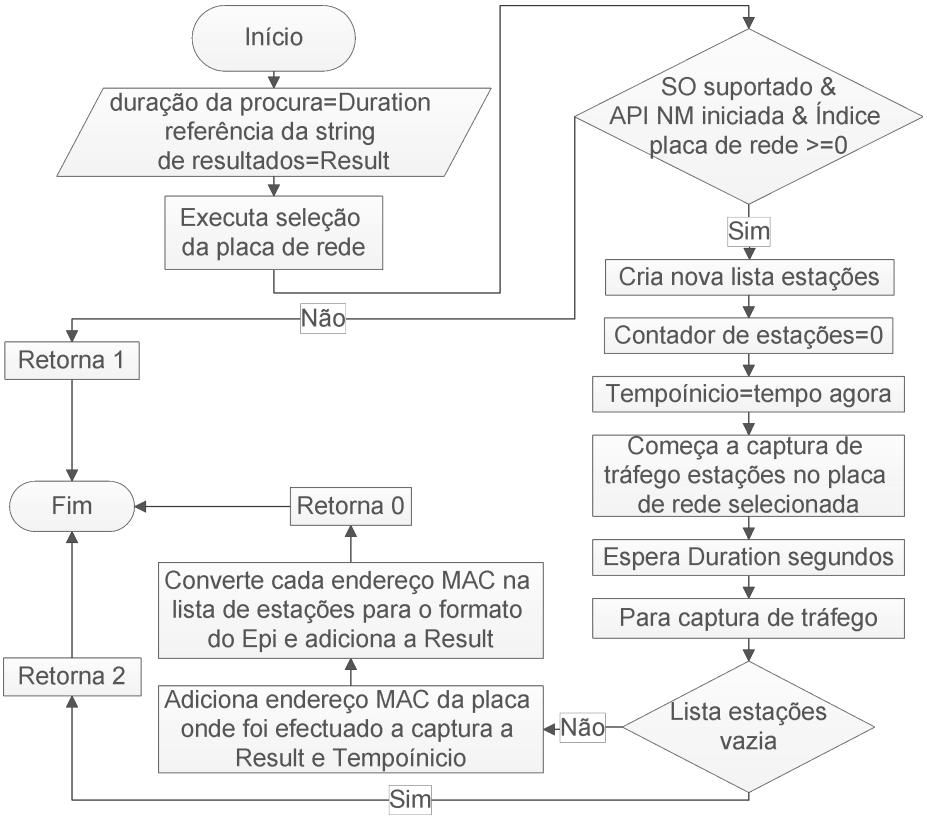


Figura 5.5: Fluxograma do método de descoberta de estações.

É registado também o *timestamp* imediatamente antes de chamar o método *Rec* que vai inicializar a captura de tráfego.

A *thread* actual adormece durante o tempo que é pretendido realizar a descoberta que foi passado como parâmetro. Durante este tempo irá decorrer a captura de tráfego e o processamento das tramas que são capturadas.

Quando a *thread* volta a acordar, chama o método *Stop* que pára a captura de tráfego. A *thread* adormece novamente durante mais um segundo para esperar que termine o processamento de eventuais tramas ainda no *buffer* (como segurança).

Os resultados depois podem ser escritos para um ficheiro HTML ao chamar o método *WriteResultHtml*, caso tenha sido pretendido ao inicializar o WifiRadar. Na integração no Epi foi desligado.

Para finalizar, a *string* de resultados é construída usando diversos campos separados por "&". O primeiro campo é o endereço MAC da placa onde foi efectuada a captura. O segundo é o instante de tempo que ficou registado no início da descoberta. O terceiro e último campo é a lista com endereços MAC das estações detectadas,

usando o separador ";" entre endereços. Os endereços são copiados dos resultados armazenados na lista de *Stations*.

Durante este processo também são enviados para o registo de eventos erros e outros eventos relevantes.

5.2.8 Processo de captura de tráfego

O método *Rec* começa o processo de captura de tráfego que termina quando se chama o método *Stop*. Ao chamar o método *Rec* este vai indicar à API do Network Monitor o ficheiro temporário onde é guardado o tráfego, ficheiro "tmp.cap" que é guardado na pasta de dados da aplicação e que serve como uma espécie de *buffer* para o tráfego capturado. Este ficheiro foi definido para o tamanho máximo de 10 MB e quando atinge o limite funciona da forma FIFO (First-In-First-Out) a descartar as tramas.

O método *Rec* vai também configurar a captura indicando à API do Network Monitor a placa de rede onde vai ser efectuada a captura de tráfego e o apontador para o método *CaptureCallBack*, que irá disparar a execução do método quando é capturada uma trama pelo *driver* do Network Monitor na placa de rede. Depois põe dois contadores a "0", um que conta o número de tramas capturadas e outro que conta o número de tramas processadas. A captura é depois iniciada e o método *Rec* termina.

Ao ser capturada uma trama, o método *CaptureCallBack* é disparado. Este método vai adicionar a trama capturada ao *buffer*, que é o ficheiro que guarda o tráfego e depois envia o sinal para desbloquear a *thread* *DispWorker_doWork* que se encontrava bloqueada.

A *thread* é o método *DispWorker_doWork*, que vai permanecer em ciclo até à sua operação ser cancelada. Em cada ciclo o método bloqueia-se, como aconteceu na sua inicialização. Quando recebe o sinal para desbloquear invoca o método *ParseFrames* que vai realizar a análise sintáctica das tramas que tem armazenadas uma a uma.

O método *ParseFrames* ao ser invocado processa todas as tramas que existem armazenadas no *buffer* (ficheiro de tráfego temporário) e termina. Assim que chegam mais tramas o método *CaptureCallBack* volta a desbloquear a *thread* do método *DispWorker_doWork* que volta a invocar o método *ParseFrames*.

Ao receber novas tramas enquanto está a decorrer o processamento na *thread* do método *DispWorker_doWork*, o sinal enviado não terá efeito nenhum mas a trama

recebida é adicionada ao *buffer* e depois processada pelo método. Apenas existe uma instância do método *DispWorker_doWork* a correr ao mesmo tempo.

Este processo repete-se até a captura de tráfego terminar quando o método *Stop* é invocado.

5.2.9 Análise Sintáctica das tramas

Esta operação acontece dentro no método *ParseFrames* que é invocado quando existem tramas armazenadas que foram capturadas. As tramas armazenadas vão sendo retiradas e processadas uma a uma.

Quando uma trama é retirada do *buffer* é invocado o método *GetWifi* onde é passado como parâmetro o apontador para a trama e um parâmetro de saída, que é um apontador para uma outra trama vazia. O método *GetWifi* vai aplicar o *parsing* na trama original usando o motor de *parsing* configurado anteriormente que depois é retornada numa nova trama com o *parsing* feito. O processamento continua depois sobre a nova trama obtida.

Se a nova trama é retornada vazia é porque não passou a validação do motor de *parsing*. Não é portanto uma trama com interesse, por isso o processamento da trama é interrompido e avança-se para a próxima.

O processo daqui para a frente é idêntico ao da aplicação de teste descrito na Secção 3.3.2. Os identificadores usados pelo Network Monitor para os campos também já foram explicados nessa secção, logo não serão novamente descritos aqui.

Usando os identificadores, é extraído da trama o *timestamp* dos *meta* dados e os campos *Type Subtype* e *DS* do *Frame Control* presente no cabeçalho de uma trama 802.11. O processamento continua depois de acordo com o valor do campo *Type*, que se divide nas seguintes operações:

Tramas de Gestão(0)

O endereçamento é sempre efectuado da mesma forma, logo não é necessário o processamento individual do campo *Subtype*. O campo *DS* é sempre "0" como especifica a norma (Tabela 2.5), no entanto esta condição é verificada para o processamento prosseguir. O fluxograma da Figura 5.6 ilustra como é efectuado a extracção do endereço de uma estação vizinha. Os campos *address1*, *address2* e *address3* da trama são usados para extrair os endereços BSSID, RA e SA, respectivamente. As tramas de gestão são transmitidas apenas entre o AP e

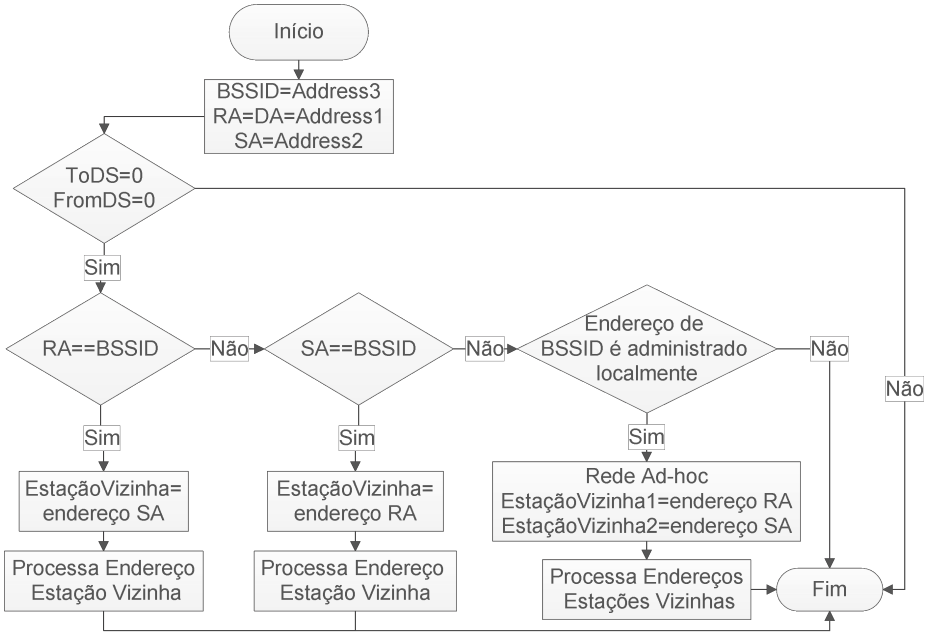


Figura 5.6: Fluxograma do processamento das tramas de gestão.

as estações e não são entregues para o DiS pois só dizem respeito à gestão da ligação entre os dois e apenas transportam dados referentes a camada MAC do 802.11. Os endereços presentes na trama são de estações na vizinhança e do AP. Para extrair o endereço da estação vizinha de forma correcta é necessário fazer verificações. Uma trama de gestão pode ser enviada entre duas estações numa IBSS, de um AP para uma estação ou na direcção contrária. Portanto são três casos diferentes que precisam de ser verificados:

- a trama é enviada de uma estação vizinha para o AP, o endereço de BSSID é igual ao RA, logo a estação vizinha tem endereço SA.
- a trama é enviada do AP para a estação vizinha, o endereço SA é igual ao endereço de BSSID, logo a estação vizinha tem endereço RA.
- a trama é enviada de uma estação vizinha para outra estação vizinha na mesma IBSS, o endereço de BSSID é um endereço administrado localmente (verificado usando o método *FilterMac*), logo temos duas estações vizinhas com os endereços SA e RA.

No final, nos dois primeiros casos, é invocado o método *ProcessAddress* para processar o endereço da estação vizinha juntamente com o BSSID, *timestamp* e

string do tipo de trama 802.11, que são passados como parâmetro. No terceiro caso é invocado o método *ProcessAddress* uma vez para cada endereço, com os parâmetros BSSID, *timestamp* e *string* do tipo de trama 802.11 iguais.

Tramas de Controlo(1)

O endereçamento é diferente para cada sub-tipo, logo é necessário o processamento individual de cada valor do campo *Subtype*. O campo *DS* é sempre "0" como especifica a norma (Tabela 2.5), no entanto, esta condição é verificada para o processamento prosseguir. O processamento continua depois de acordo com o valor do campo *Subtype*:

BAR(8)*, *BA(9)* e *RTS(11) - Estes sub-tipos de tramas têm endereçamento idêntico. Os dois campos de endereços presentes, *address1* e *address2* são usados como endereços de receptor e transmissor, respectivamente. O fluxograma da Figura 5.7 ilustra como é efectuado a extracção do endereço de uma estação vizinha para estes sub-tipos de tramas. São extraídos os

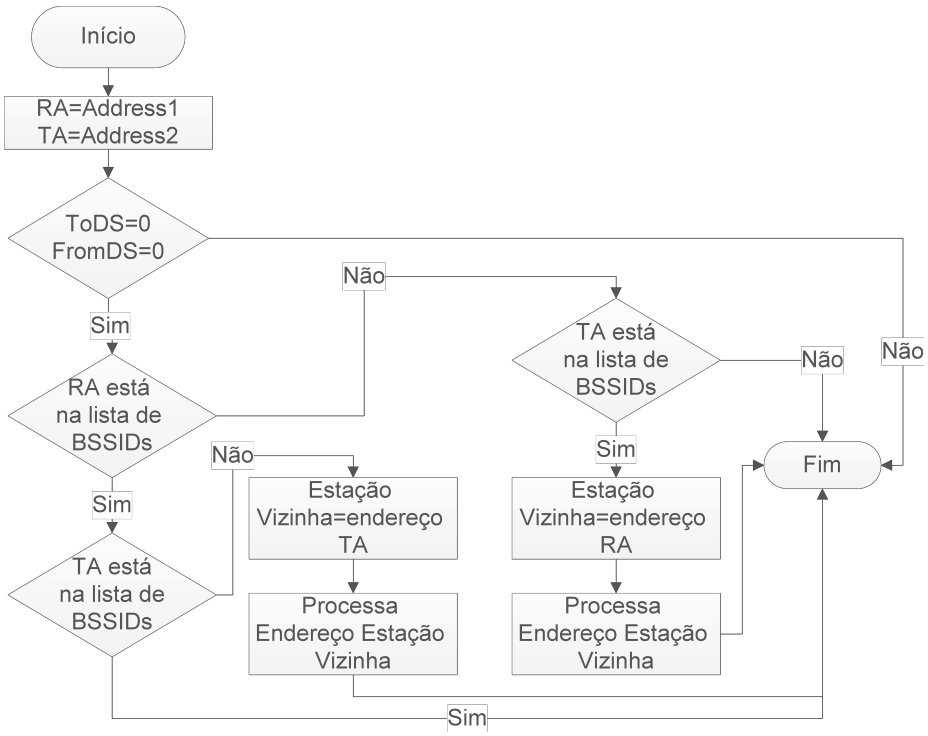


Figura 5.7: Fluxograma do processamento das tramas de controlo BAR, BA e RTS.

endereços RA e TA dos campos *address1* e *address 2*, respectivamente.

Verifica-se se o campo *DS* é "0". Como não existe endereço de BSSID, é usada a lista de redes *WiFi* para verificar dois casos diferentes:

- O endereço RA é uma BSSID e o endereço TA não é BSSID, logo a estação vizinha tem endereço TA.
- O endereço TA é uma BSSID e o endereço RA não é BSSID, logo a estação vizinha tem endereço RA.

No final é invocado o método *ProcessAddress* para processar o endereço da estação vizinha juntamente com o *timestamp* e *string* do tipo de trama 802.11 que são passados como parâmetro. O BSSID neste caso vai a zeros nos parâmetros.

PS-Poll(10) - Este sub-tipo de trama tem dois campos de endereços. Os dois campos de endereços presentes, *address1* e *address2* são usados como endereços de BSSID e transmissor, respectivamente. O fluxograma da Figura 5.8 ilustra como é efectuado a extracção do endereço de uma estação vizinha para este sub-tipo de trama. São extraídos os endereços BSSID e TA,

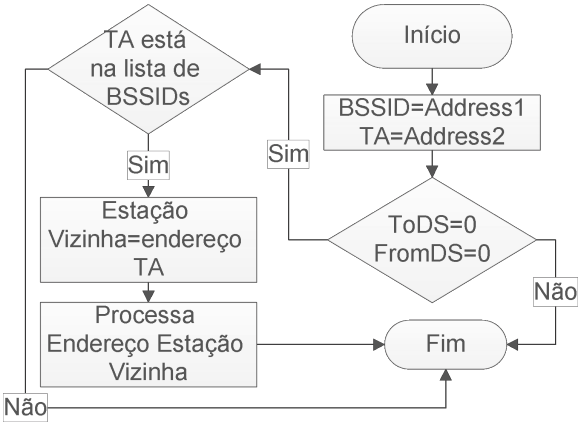


Figura 5.8: Fluxograma do processamento das tramas de controlo PS-Poll.

dos campos *address1* e *address 2*, respectivamente. Verifica-se se o campo *DS* é "0". Este sub-tipo de trama é sempre transmitido de uma estação para um AP, logo pode-se concluir que o endereço TA é uma estação vizinha, no entanto o endereço de BSSID é verificado se existe na lista de BSSIDs. Em caso positivo, é invocado o método *ProcessAddress* para processar o endereço da estação vizinha juntamente com o BSSID, *timestamp* e *string* do tipo de trama 802.11, que são passados como parâmetro.

CTS(12), Ack(13)* e *ControlWrapper(7) - Estes sub-tipos têm apenas o campo de endereço *address1*. Não se pode concluir que o endereço é de uma estação vizinha, logo o processamento destes sub-tipos termina.

CF-End(14)* e *CF-End+CF-Ack(15) - Estes sub-tipos de trama tem dois campos de endereços. Os dois campos de endereços presentes *address1* e *address2* são usados como endereços de receptor e BSSID, respectivamente. As tramas destes sub-tipos são sempre transmitidas de uma BSSID para todas as estações. A norma especifica que o endereço RA é o endereço de *broadcast*, assim o processamento destes sub-tipos termina porque não existe um endereço de uma estação vizinha presente.

Tramas de Dados(2)

O endereçamento é sempre efectuado da mesma forma, logo não é necessário o processamento individual de cada sub-tipo. O campo *DS* varia como especifica a norma (Tabela 2.5) por isso o processamento dos endereços deste tipo de trama terá que ser realizado de acordo com a Tabela 2.7. O fluxograma da Figura 5.9 ilustra como é efectuada a extracção do endereço de uma estação vizinha para este tipo. Nestas tramas existem as que são destinadas ao DiS, que o destino é a rede local ou a Internet e que passam através do AP numa Infrastructure BSS. Existe também o inverso, que têm origem na rede local ou na Internet e que são entregues ao AP pelo DiS, com destino a uma estação associada nesse AP. Também existem as tramas que são trocadas entre duas estações na mesma IBSS. O processamento continua de acordo com o valor do campo *DS*[1]:

DS=0 É uma trama enviada dentro de uma IBSS. Os campos da trama *address1*, *address2* e *address3* são usados para extrair os endereços RA, SA e BSSID, respectivamente. Se o endereço de BSSID é um endereço administrado localmente (verificado usando o método *FilterMac*), confirma-se que é uma IBSS e temos duas estações vizinhas com os endereços RA e SA.

DS=1 É uma trama enviada de uma estação para o AP numa Infrastructure BSS. Os campos da trama *address1*, *address2* e *address3* são usados para extrair os endereços BSSID, SA e DA, respectivamente. Se o endereço

[1]Este campo é a combinação dos campos *ToDS* e *FromDS* da trama 802.11

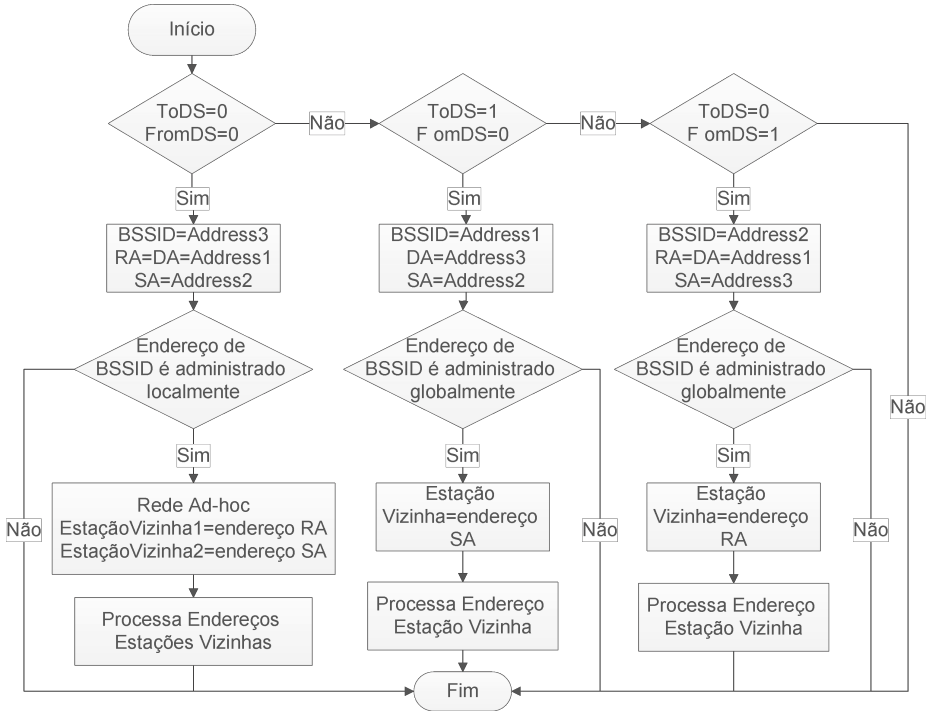


Figura 5.9: Fluxograma do processamento das tramas de dados.

de BSSID é um endereço administrado globalmente (verificado usando o método *FilterMac*), confirma-se que é um AP real de uma Infrastructure BSS e temos uma estação vizinha com o endereço SA.

DS=2 É uma trama enviada de um AP para uma estação numa Infrastructure BSS. Os campos da trama *address1*, *address2* e *address3* são usados para extrair os endereços DA, BSSID e SA, respectivamente. Se o endereço de BSSID é um endereço administrado globalmente (verificado usando o método *FilterMac*), confirma-se que é um AP real de uma Infrastructure BSS e temos uma estação vizinha com o endereço DA.

Para finalizar é invocado o método *ProcessAddress* para processar o endereço vizinho, juntamente com o BSSID, *timestamp* e *string* do tipo de trama 802.11. Numa IBSS são duas estações vizinhas, por isso o método é invocado duas vezes. Numa Infrastructure BSS é uma única estação vizinha, logo o método é invocado uma única vez.

Durante este processo são enviados para o registo de eventos erros, eventos relevantes, resultado da análise sintáctica de todas tramas e da análise de cada endereço MAC.

5.2.10 Processamento dos endereços

No final da análise sintáctica de uma trama pode-se obter o endereço MAC de uma estação vizinha (em alguns casos dois). Depois de obtido o endereço é necessário processar o mesmo para verificar se este já existe na lista de estações e se é válido. Considera-se válido se for um endereço MAC administrado globalmente, verificado usando o método *FilterMac*.

O processamento do endereço é efectuado no método *ProcessAddress*, que tem como parâmetros o endereço MAC da estação vizinha, endereço da BSSID, *timestamp* e uma *string* do tipo de trama 802.11. Na Figura 5.10 está ilustrado o fluxograma que descreve o seu funcionamento.

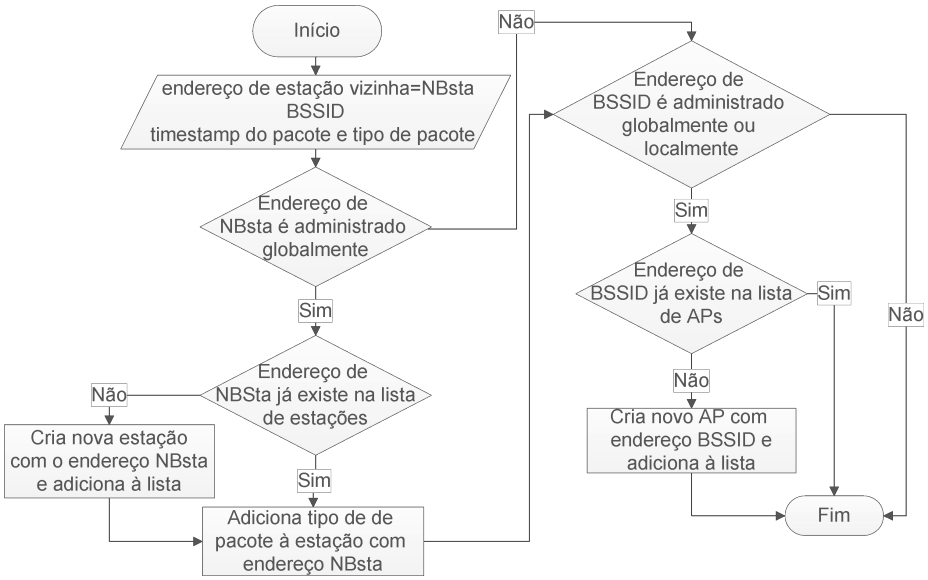


Figura 5.10: Fluxograma da função que processa os endereços.

O endereço MAC da estação vizinha vai ser usado para criar uma nova *Station* na lista de *Stations*, caso ainda não exista. Ao ser criada é usado o *timestamp* para registar quando foi detectada pela primeira vez.

A *string* com tipo de trama 802.11 é adicionada à lista dos tipos de tramas 802.11 que existe em cada *Station* para registar em cada estação vizinha as tramas nas quais foi detectada. Cada um dos tipos de trama 802.11 tem também um contador associado. O somatório destes contadores permite obter em quantas tramas foi detectada determinada estação.

O endereço de BSSID é usado para criar um objecto AP e para o adicionar à lista de redes *WiFi*, caso ainda não exista e seja um endereço válido. Considera-se

válido se for um endereço MAC administrado globalmente (Infrastructure BSS) ou localmente (IBSS), que é verificado usando o método *FilterMac*.

Durante este processo são enviados para o registo de eventos a detecção de novas estações vizinhas e a análise de cada endereço MAC.

5.3 Integração no Epi

Para integrar na aplicação Epi foi adicionada a referência para a biblioteca do módulo WifiRadar. Desta forma todas as funcionalidades do módulo ficaram disponíveis para serem usadas pela aplicação.

No Gestor Principal foi criado um conjunto de variáveis globais para dar suporte à execução das funcionalidades disponíveis pelo WifiRadar:

wifimodule_enable - permite activar ou desactivar a execução das funcionalidades do WifiRadar. A variável é do tipo booleano. O módulo está activado com a variável a "true" e desactivado quando está a "false". O valor desta variável será controlado pelos parâmetros de configuração e em alguns casos, apenas para desactivar, pelo método *recolhaocasional_stawifi*.

wifimodule - o WifiRadar para ser usado tem que criar um objecto que irá guardar toda a informação relativa à sua execução. Esta variável é do tipo *WifiMon* que é um tipo único do WifiRadar.

wifimodule_api_state - guarda o estado em que se encontra a API do Network Monitor. A variável é do tipo inteiro e inicializada a "0". Quando a API se encontra inicializada este valor está a "1", qualquer outro valor a API não se encontra inicializada.

trecolhastawifi - temporizador responsável por chamar periodicamente o método *recolhaocasional_stawifi* que executa o código relacionado com a recolha das estações *WiFi* usando o WifiRadar. O temporizador chama pela primeira vez o método após alguns segundos da sua inicialização e depois periodicamente em intervalos regulares. O temporizador é do tipo *System.Threading.Timer*. A primeira vez que chama o método ocorre passados 15 segundos, o tempo entre as chamadas seguintes é variável.

intervalofsampling_stawifi - contém o tempo (em segundos) entre intervalos para o temporizador. O valor desta variável será controlado pelos parâmetros de configuração da aplicação. A variável é do tipo inteiro.

No Gestor Principal foi criado o método *recolhaocasional_stawifi* que é responsável por toda a interacção com a biblioteca do WifiRadar e que é chamada periodicamente pelo temporizador. Na Figura 5.11 está ilustrado o fluxograma do método.

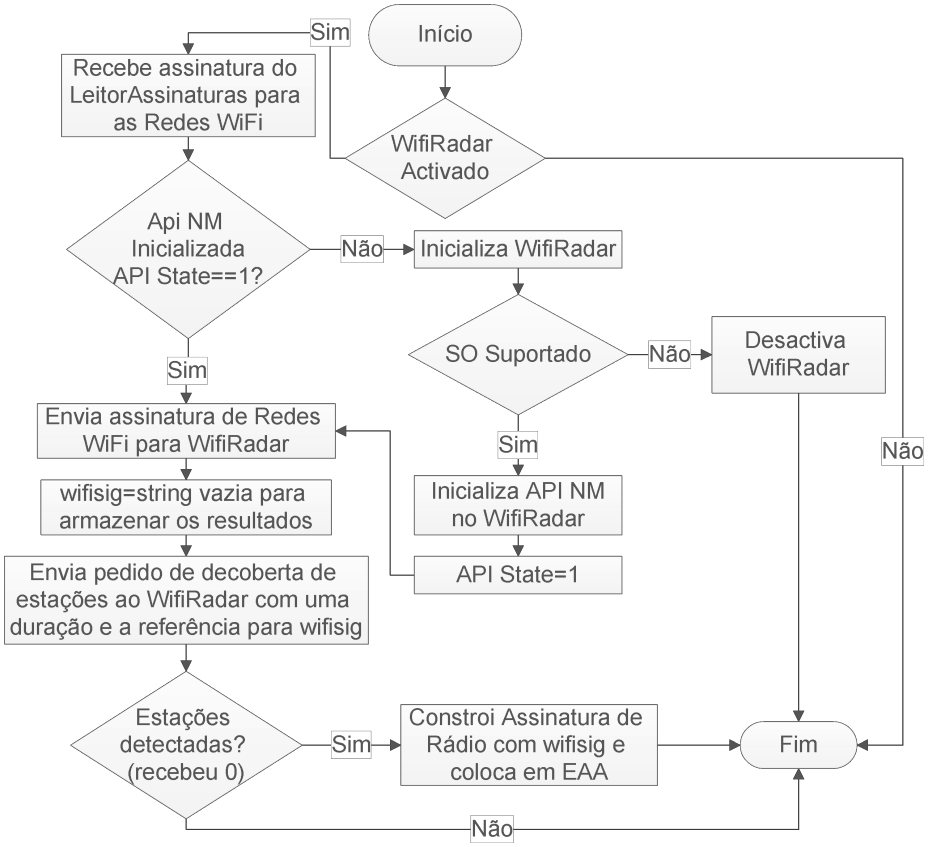


Figura 5.11: Fluxograma do método *recolhaocasional_stawifi*.

No método *recolhaocasional_stawifi* a primeira operação é verificar se o módulo está activado ou desactivado, terminando neste último caso. Ao prosseguir, chama o módulo do leitor de assinaturas de rádio para obter a lista das redes *WiFi* na vizinhança. Depois verifica se a API do Network Monitor já se encontra inicializada (se $wifimodule\_api\_state == 1$) que na primeira execução nunca vai estar.

Antes de inicializar a API, é chamado o construtor para criar o novo objecto do tipo *WifiMon* para depois chamar o método (*isWifiSupported*) dentro do WifiRadar

que verifica se o SO é suportado. Quando o SO não é suportado, o módulo é desactivado ($wifimodule\_enable = false$), caso contrário prossegue e chama o método (*InitAPI*) que vai inicializar a API do Network Monitor. É alterada depois a variável do estado da API ($wifimodule\_api\_state = 1$).

Os próximos passos são efectuados na primeira execução ou nas seguintes, quando o módulo está activado. A lista das redes *WiFi* é carregada para dentro do WifiRadar (método *SetAPsSignature*) e depois começa a procura por estações *WiFi* na vizinhança (método *ScanStations*).

Ao retornar resultados, é construída a assinatura de rádio das estações *WiFi* que foram detectadas na vizinhança e colocada dentro da estrutura de armazenamento de assinaturas (EAA). Caso não seja detectada nenhuma estação, a assinatura de rádio não é construída e o método termina.

5.4 Pacotes de instalação EPI

A criação do pacote de instalação para a aplicação Epi foi um desafio porque era necessária a integração do Network Monitor. Essa integração teve que ser realizada de forma a ser transparente para o utilizador durante a sua instalação.

Na construção dos pacotes de instalação do Epi começou-se por fazer um levantamento dos componentes necessários do Network Monitor. Depois procedeu-se à simplificação dos ficheiros de *parsing* que descrevem os protocolos. Por fim foram construídos os dois pacotes de instalação no Visual Studio com os componentes do Network Monitor integrados.

5.4.1 Identificação dos componentes do NM

Para começar foi realizado um levantamento dos componentes do Network Monitor que eram necessários para o módulo WifiRadar funcionar. Esse levantamento foi efectuado sobre as versões 32 bits e 64 bits do Network Monitor. Antes de começar o levantamento dos componentes foram removidas todas as versões deste *software* do sistema.

Os passos para identificar esses componentes foram os seguintes:

1. Foram executados os comandos que estão na Listagem 5.1. Desta forma obtiveram-se duas pastas com todos os ficheiros da aplicação para as versões 32

bits e 64 bits. Havia no entanto um *NetworkMonitor_Parsers.msi* que não foi possível extrair.

```
1 C:\NNM34\NM34_x64.exe /T:"C:\NM34\64" /C
2 C:\NNM34\NM34_x86.exe /T:"C:\NNM34\32" /C
3 msiexec /a C:\NM34\64\netmon.msi /qb TARGETDIR=C:\NM34\64\app
4 msiexec /a C:\NM34\32\netmon.msi /qb TARGETDIR=C:\NM34\32\app
```

Listagem 5.1: Extracção dos pacotes de instalação do Network Monitor.

Linha 1 e 2 - Executa o pacote SFX (SelF eXtracting) de instalação do Network Monitor com os argumentos a indicar para apenas extrair os ficheiros contidos no pacote, para as pastas "C:/NM34/64" no caso da versão 64 bits (Linha 1) e "C:/NM34/32" para a 32 bits (Linha 2). Nestas pastas foram obtidos os ficheiros MSI usados para instalar o Network Monitor usando o Windows Installer.

Linha 3 e 4 - Executa o Windows Installer indicando para extrair os ficheiros contidos no pacote de instalação MSI, obtidos antes, para as pastas "C:/NM34/64/app" na versão 64 bits (Linha 3) e "C:/NM34/32/app" na versão 32 bits (Linha 4). Estas pastas contêm os ficheiros do Network Monitor nas respectivas versões.

2. Em ambas as pastas foram encontradas as sub-pastas *windir* e *PFiles*. A primeira contém um par de ficheiros, *nm3.sys* e *netnm3.inf*, que constituem o LightWeight Filter NDIS Driver. A segunda contém os ficheiros da aplicação em que imediatamente foi reconhecido o ficheiro *NMAPI.dll* que é chamado pelo *Wrapper* usado no desenvolvimento do WifiRadar.

3. Ao ler o ficheiro *ReleaseNotes.txt* que contém ajuda para resolver problemas de instalação do Network Monitor, foi encontrada referência ao ficheiro *nmconfig.exe* que serve para instalar automaticamente o LightWeight Filter NDIS Driver.

4. Ao colocar o *NMAPI.dll* na pasta da aplicação e instalar o controlador no sistema já conseguimos correr a aplicação. No entanto acontecia um erro ao carregar os ficheiros NPL do motor de parsing. Nesta altura percebeu-se que o ficheiro que não se conseguiu extrair no Passo 1 tinha este componente.

5. O passo seguinte foi voltar a instalar o Network Monitor no sistema e encontrar os ficheiros NPL. Estes foram facilmente encontrados ao consultar as opções dos perfis de *parsing* da aplicação, como ilustrado na Figura 5.12. A pasta contém um conjunto de ficheiro com extensão NPL, em que cada um é descrição de um protocolo de rede na linguagem NPL. Nesta altura já tínhamos todos os componentes necessários.

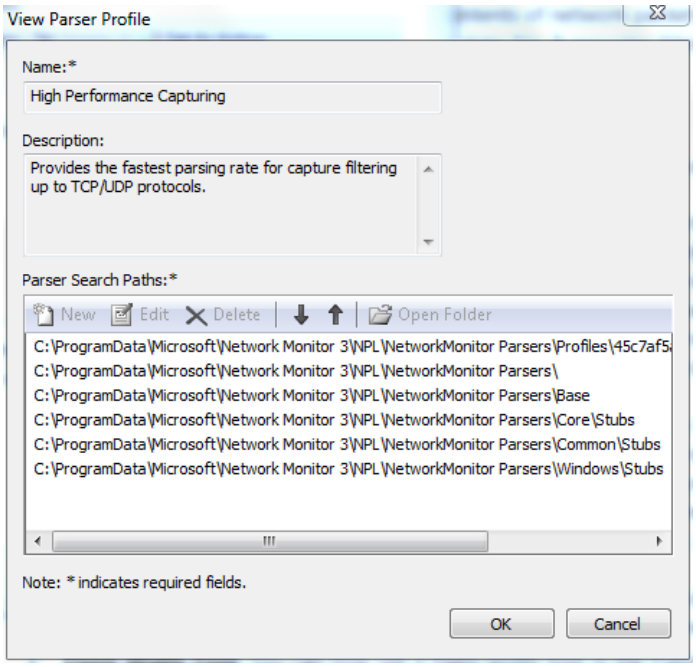


Figura 5.12: Opções dos perfis de *parsing* no Network Monitor.

Nos componentes que foram identificados, encontraram-se vários que são comuns nas versões 32 bits e 64 bits e outros que não. Isto obrigou à criação de dois pacotes de instalação, um destinado a SOs 32 bits e outro ao 64 bits, assim como acontece com os pacotes de instalação do Network Monitor.

Na Tabela 5.3 encontram-se todos os componentes que foram identificados, os respectivos ficheiros que constituem o componente e a arquitectura. A pasta de destino de alguns componentes na instalação é muito específica no caso do *driver*, uma vez que o instalador do *driver* (*nmconfig.exe*) apenas o consegue instalar se ele já estiver na pasta de controladores do SO.

Componente	Ficheiros	Arquitectura	Destino
API 32 bits	*NMAPI.dll*	x86	*Application Folder*
API 64 bits	*NMAPI.dll*	x64	*Application Folder*
Driver 32 bits	*nm3.sys*	x86	*SystemFolder*/drivers
Driver 64 bits	*nm3.sys*	x64	*System64Folder*/drivers
Driver Inf	*netnm3.inf*	x86 e x64	*Windows Folder*/inf
Instalador Driver 32 bits	*nmconfig.exe*	x86	*Application Folder*
Instalador Driver 64 bits	*nmconfig.exe*	x64	*Application Folder*
Ficheiro de parsing NPL	diversos	x86 e x64	*AppData*/Epi

Tabela 5.3: Componentes principais do Network Monitor.

5.4.2 Simplificação dos ficheiros de parsing

Os ficheiros NPL usados pelo motor de *parsing* formam um conjunto de 420 ficheiros de texto que descrevem tabelas, estruturas, tipos de dados e protocolos na linguagem NPL. Na pasta principal desses ficheiros encontra-se o *NetworkMonitor_Parsers_sparser.npl*, que não é mais do que um ficheiro de texto principal que chama os restantes através do *include* de cada um. Estes são incluídos pela ordem da camada de rede a que pertencem. Os restantes ficheiros estão separados em pastas, *Base*, *Core*, *Common* e *Windows*.

Os ficheiros NPL contidos na pasta *Base* são obrigatórios de forma a que seja possível executar o *parsing* dos protocolos de rede básicos, Ethernet, IPv4, IPv6, *WiFi*, LLC, SNAP, UDP, TCP, etc. A pasta *Core* inclui os ficheiros NPL relativos aos protocolos que funcionam imediatamente em cima da camada de transporte, ICMP, IGMP, ARP, etc. As restantes pastas são relativas a protocolos ao nível de aplicação.

O *parsing* é mais dispendioso em recursos consoante os protocolos incluídos. A captura efectuada pelo WifiRadar apenas necessita do *parsing* ao nível da camada MAC do 802.11, logo apenas é necessário os protocolos básicos que estão na pasta *Base*.

As restantes pastas contêm uma sub-pasta com Stubs[1] que serão usados em vez da descrição completa, como faz o Network Monitor ao definir os perfis de *parsing*.

O ficheiro *NetworkMonitor_Parsers_sparser.npl* que inclui os NPL todos trata-se apenas de um formalismo que não é necessário para a integração do Network Monitor no pacote de instalação do Epi. O conteúdo de todos os ficheiro NPL foi inserido

[1]É um termo usado no desenvolvimento de software em que um pedaço de código é usado para substituir algumas outras funcionalidades de programação.

dentro de um único ficheiro pela mesma ordem que eram incluídos no *NetworkMonitor_Parsers_sparser.npl*, com os ficheiros NPL das pastas *Core*, *Common* e *Windows* a incluírem apenas os Stubs.

No final do processo obtemos um único ficheiro (*sparser.npl*) que contém toda a informação necessária para o motor de *parsing*. O destino do ficheiro na instalação permanece o mesmo (*AppData*/Epi).

5.4.3 Construção dos pacotes de instalação

No Visual Studio foram criados dois projectos de *Setup* do *Visual Studio Installer*, um destinado para SOs Windows 32 bits ($TargetPlatform = x86$) e outro para 64 bits ($TargetPlatform = x64$).

Em cada um dos projectos foram adicionados pré-requisitos que serão instalados antes da aplicação, uma vez que a aplicação necessita dos mesmos para correr. Esses pré-requisitos são:

- Windows Installer 3.1
- Microsoft .NET Framework 4 Client Profile (x86 and x64)

Estes componentes, caso não sejam detectados no sistema ao iniciar a instalação, serão descarregados da Internet e depois instalados automaticamente.

Foi adicionado depois em ambos os projectos o *Primary Output* que é a aplicação Epi. As bibliotecas referenciadas na aplicação Epi foram automaticamente adicionadas uma vez que são componentes essenciais para a aplicação.

A seguir foi construída a respectiva árvore de ficheiros para cada projecto, com os componentes do Network Monitor para as respectivas plataformas. No Anexo B encontram-se as tabelas com a árvore de ficheiros para os dois projectos.

Durante a instalação é necessário instalar o *driver* do Network Monitor, assim como remover quando o utilizador desinstalar a aplicação. Como o Visual Studio permite a execução de *Custom Actions*, foram adicionadas as seguintes, que são iguais em ambos os projectos:

- Na instalação executar: “nmconfig.exe /Install”.
- Na desinstalação executar: “nmconfig.exe /Uninstall”.

Também foram adicionadas *Launch Conditions* que tinham que ser verificadas para que a instalação prosseguisse. Algumas das condições são diferentes, enquanto outras são comuns a ambos os projectos. As condições são então as seguintes:

1. O .NET Framework 4 Client Profile tem que se encontrar instalado no sistema. É comum a ambos os projectos.

2. A versão 2 do *driver* do Network Monitor não pode existir no sistema caso o SO seja o Windows Vista, Server 2008, 7 e 8. O ficheiro *nmnt.sys* não pode existir na pasta *SystemFolder/drivers* para 32 bits e *System64Folder/drivers* para 64 bits. A condição é "NOT (OLDDRIVEREXISTS AND VersionNT >= 600)" onde "OLDDRIVEREXISTS" é verdadeiro caso o ficheiro do *driver* exista.

3. Verifica se o SO é 32 bits para o projecto 32 bits, condição "Intel AND NOT VersionNT64". Se é 64 bits para o projecto 64 bits, condição "Msix64 and VersionNT64".

4. A versão do SO tem que ser Windows XP ou Server 2003 ou ainda Vista, Server 2008, 7 e 8 com a *build* maior que 5520. A condição usada é a seguinte "VersionNT = 501 OR VersionNT = 502 OR (VersionNT >= 600 AND WindowsBuild >= 5520)"

Nesta altura ambos os projectos encontram-se prontos. Após a construção dos dois pacotes de instalação obteve-se na saída dois ficheiros para cada pacote:

- os ficheiros *setup.exe* e *SetupEpi_x64.msi* para a versão 64 bits.
- os ficheiros *setup.exe* e *SetupEpi_x86.msi* para a versão 32 bits.

O ficheiro *setup.exe* faz a verificação e instalação dos pré-requisitos. O ficheiro MSI é usado para instalar a aplicação através do Windows Installer[1].

Como se pretende simplificar o processo de instalação e impingir o utilizador a correr em primeiro lugar o *setup.exe*, foi criado um executável SFX para os pacotes de instalação 32 bits e 64 bits. Assim, apenas um único executável contém os dois ficheiros de cada pacote e executa automaticamente a instalação da aplicação através do *setup.exe*.

[1]Programa da Microsoft usado para instalar aplicações no computador nos seus Sistemas Operativos.

O SFX foi criado usando o Winrar[1] com as opções de extrair automaticamente para uma pasta temporária e executar automaticamente o *setup.exe*.

Nesta altura os dois pacotes de instalação finais para a aplicação Epi encontram-se prontos a serem distribuídos. O utilizador terá que instalar a versão adequada ao seu SO, *Epi32bit.exe* para 32 bits e *Epi64bit.exe* para 64 bits.

[1]É um *software* que permite comprimir e descomprimir ficheiros em diversos formatos.

Capítulo 6

Testes e resultados

Neste capítulo são descrito os testes que foram realizados com a aplicação Epi e com o WifiRadar integrado.

Também é realizada uma análise aos resultados obtidos.

6.1 Testes realizados

A nova versão da aplicação Epi com o WifiRadar integrado e outros novos módulos e funcionalidades foi disponibilizada ao público em geral. Ao mesmo tempo foi posto online um novo servidor, com a base de dados vazia, para armazenar as novas assinaturas de rádio enviadas pelos utilizadores da nova versão da aplicação.

Queríamos obter um maior número possível de utilizadores, que equivale a obter o maior número de assinaturas de rádio possível, no espaço curto de tempo que tínhamos disponível para realizar a recolha. Por isso, a nova aplicação Epi foi distribuída entre pessoas conhecidas dentro do âmbito universitário e foi levado a cabo uma pequena distribuição publicitária através do correio electrónico da Universidade do Minho e das redes sociais Facebook e Twitter.

O servidor foi posto *online* no dia 27 de Setembro de 2011 por volta das 16:00, a nova aplicação Epi foi disponibilizada pouco depois. No dia 18 de Outubro de 2011 por volta das 17:00, foram descarregados os dados que se encontravam na base de dados do servidor.

Durante o período de recolha os parâmetros de configuração para o módulo WifiRadar permaneceram iguais aos predefinidos. O período entre recolhas de assinaturas no cliente foi de 30 minutos e o módulo permaneceu sempre activado. A duração de

uma descoberta não é um parâmetro de configuração, no entanto é importante referir que foi de 15 segundos.

Estes testes realizados vão permitir saber se o módulo se encontra correctamente integrado na aplicação Epi e o seu comportamento em diferentes placas de redes sem fios, fora de um ambiente laboratorial controlado. Em ambiente laboratorial controlado os testes estão descritos na Secção 3.3.3.

A análise realizou-se sobre os dados recolhidos durante um período de tempo de aproximadamente 3 semanas. Os dados recolhidos são apenas ilustrativos, uma vez que o tempo de recolha foi relativamente curto.

6.2 Análise dos resultados

A análise dos dados realizou-se apenas sobre as assinaturas de rádio das estações *WiFi* detectadas na vizinhança, recolhidas pelo módulo WifiRadar.

Com os dados recolhidos foram realizadas duas análises:

Primeira análise - em que a própria estação que fez a recolha de assinaturas está contabilizada nas assinaturas como uma estação presente na vizinhança. Assim, uma assinatura contém sempre pelo menos a estação que realizou a recolha da mesma.

Segunda análise - em que a própria estação que fez a recolha de assinaturas não está presente nas assinaturas. Uma assinatura pode não conter estações que, neste caso, deixam de ser consideradas.

6.2.1 Análise com contabilização da própria estação

Os dados analisados referem-se a um período de aproximadamente 3 semanas, em que a primeira assinatura recebida foi recolhida no dia 27 de Setembro de 2011 às 16:24 e a última do dia 18 de Outubro de 2011 às 16:46.

Foram recebidas um total de 758 assinaturas, que foram submetidas por 16 utilizadores Epi diferentes, a que correspondem 2272 dispositivos detectados, onde 710 desses dispositivos são distintos.

No gráfico da Figura 6.1 está ilustrado o número de estações detectadas em cada assinatura recolhida durante o período de 3 semanas.

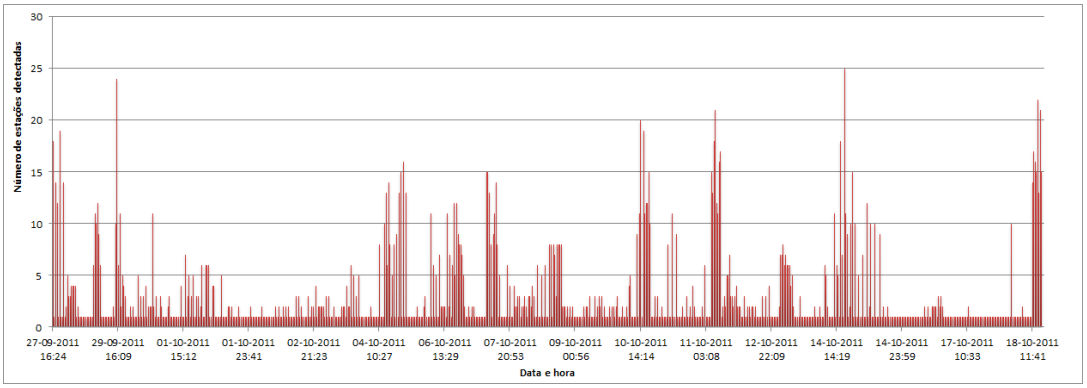


Figura 6.1: Número de estações detectadas por assinatura (com contabilização da própria).

No gráfico é possível observar que o máximo de estações detectadas numa assinatura é de 25. O valor médio do gráfico situa-se muito próximo das 3 estações detectadas por assinatura.

Observa-se uma grande percentagem de assinaturas onde o número de estações detectadas por assinatura é 1, o valor mínimo (porque, neste caso, é sempre contabilizada a própria estação na recolha de uma assinatura).

Também é possível observar um padrão nas detecções de estações. Existem períodos onde são detectadas um número grande de estações, seguido por um período com menor número de detecções e isto repete-se quase sempre ao longo das 3 semanas representadas no gráfico. Este comportamento deve-se às noites e dias. É normal as pessoas permanecerem mais activas durante o dia e menos activas durante a noite.

No histograma da Figura 6.2 é ilustrado o acumulado da detecção de estações pela hora do dia em que ocorreram ao longo das 3 semanas.

No histograma podemos observar o comportamento referido anteriormente através do número de detecções:

9h às 12h - o número de detecções começa a aumentar devagar durante a manhã.

12h às 14h - o número de detecções desce ligeiramente durante a hora do almoço.

14h às 16h - o número de detecções aumenta acentuadamente durante a tarde até atingir o pico às 16h.

16h às 19h - o número vai descendo ligeiramente até às 19h, normalmente quando as pessoas saem do trabalho ou aulas para ir para casa.

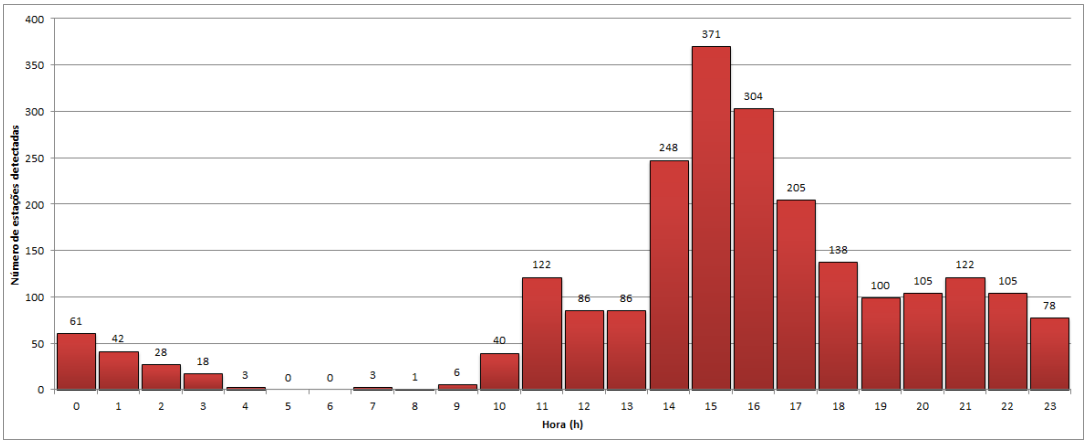


Figura 6.2: Detecções acumuladas por hora do dia (com contabilização da própria).

19h-23h - o número de detecções mantém-se estável.

23h-3h - o número de detecções cai progressivamente até as 3h.

3h-9h - permanece quase nulo.

Achou-se interessante também realizar um acumulado para cada dia durante as 3 semanas que está ilustrado no histograma da Figura 6.3.

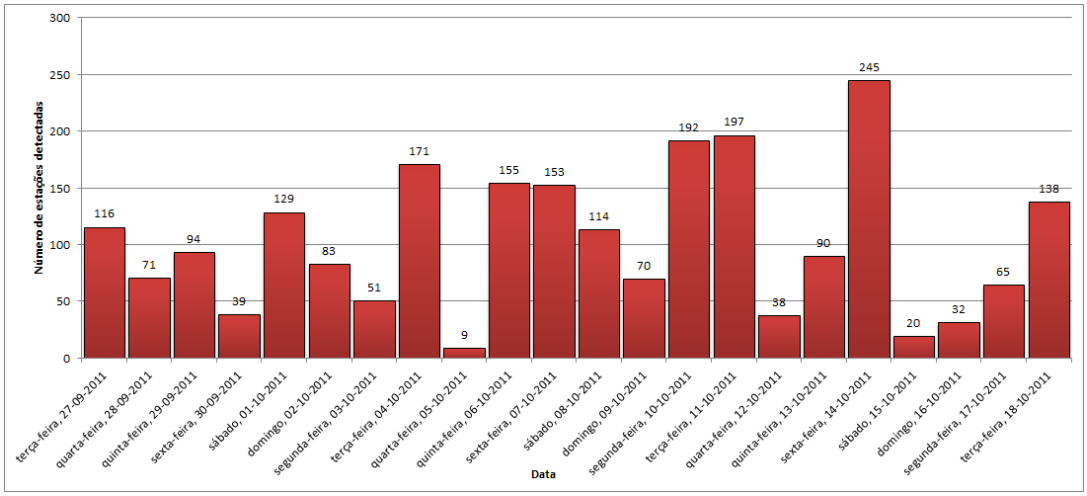


Figura 6.3: Detecções acumuladas por dia (com contabilização da própria).

Neste histograma observa-se que em alguns dias, principalmente ao fim de semana, o número de detecções diminui bastante. Outra observação é que o número de detecções no geral tem aumentado progressivamente desde o início, ou seja, o número de utilizadores da aplicação Epi está a aumentar.

No histograma da Figura 6.4 está ilustrado a frequência com que acontece o número de estações detectados em assinaturas.

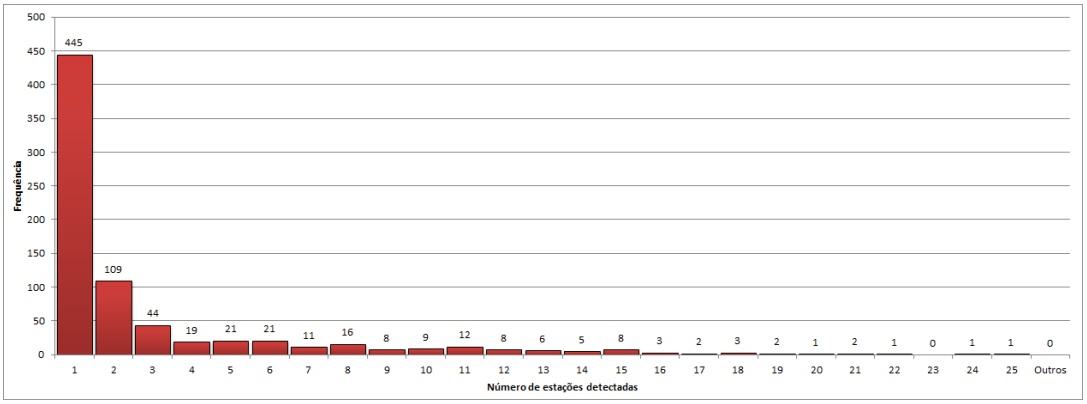


Figura 6.4: Frequência do número de estações detectados em assinaturas (com contabilização da própria).

Vemos que é mais frequente a estação realizar a recolha de uma assinatura e apenas detectar a própria estação. Volta-se a confirmar o número máximo de estações detectado numa assinatura.

Outra informação interessante está ilustrada no histograma da Figura 6.5 que mostra a frequência com que foram detectadas as estações nas assinaturas durante as 3 semanas.

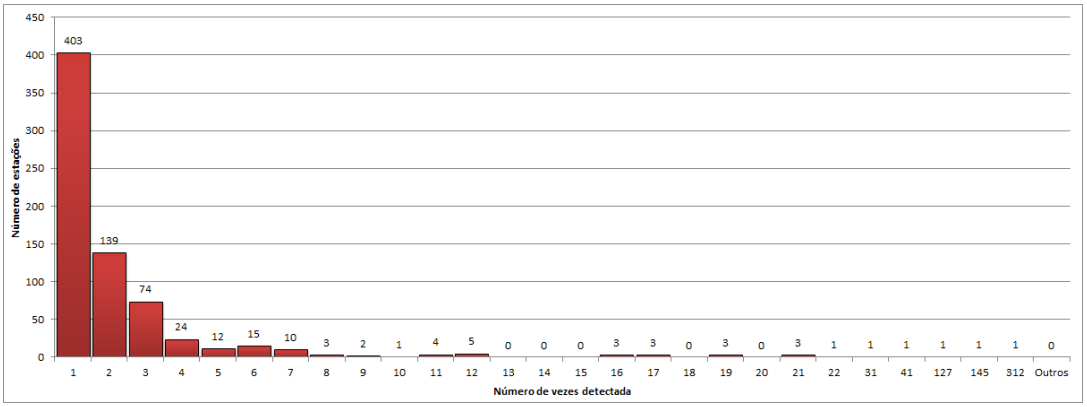


Figura 6.5: Frequência do número de vezes que as estações foram detectadas (com contabilização da própria).

Existe uma estação que foi detectada 312 vezes e um conjunto de duas outras estações que foram detectados um número elevado de vezes nas assinaturas. Estas estações pertencem às 3 pessoas envolvidas no desenvolvimento da aplicação Epi que

já têm a aplicação instalada desde o primeiro dia. Observa-se que a maioria das estações foram detectas poucas vezes ou mesmo uma única vez.

Este facto é positivo, uma vez que mostra que estamos de facto a detectar outras estações presentes na vizinhança. Uma estação que foi detectada 312 vezes neste espaço de tempo indica que está a realizar muitas recolhas em que apenas encontrou a própria estação. Na segunda análise a distância entre o número de vezes que foi detectada é muito menor.

6.2.2 Análise sem contabilização da própria estação

Os dados analisados referem-se ao mesmo período de aproximadamente 3 semanas, a primeira assinatura recebida foi recolhida no dia 27 de Setembro de 2011 às 16:24 e a última do dia 18 de Outubro de 2011 às 16:46, que é um acaso ser igual ao caso anterior.

Foram recebidas um total de 317 assinaturas que foram submetidas por 13 utilizadores Epi diferentes, a que correspondem 1535 dispositivos detectados, onde 700 desses dispositivos são distintos.

No gráfico da Figura 6.6 está ilustrado o número de estações detectadas em cada assinatura recolhida durante o período de 3 semanas, desta vez sem a contabilização da própria estação que recolheu a assinatura.

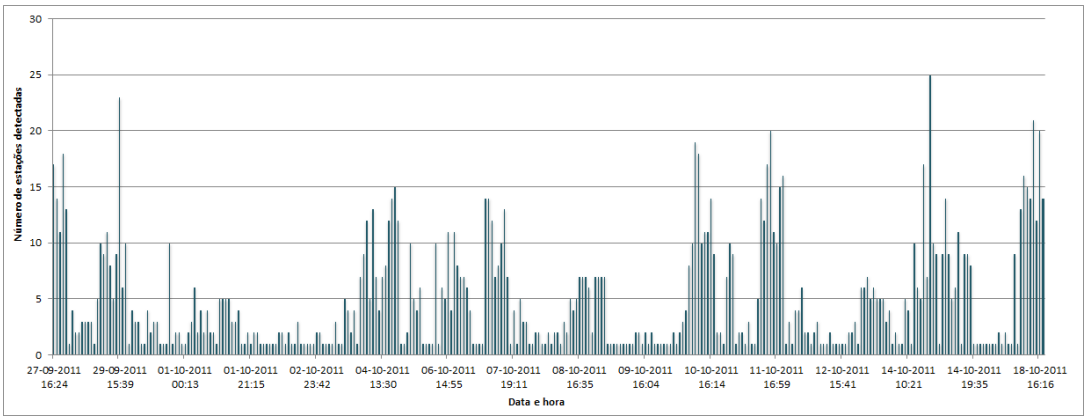


Figura 6.6: Número de estações detectadas por assinatura (sem contabilização da própria).

O gráfico permanece muito idêntico ao anterior na Figura 6.1, no entanto os valores mínimos que se observam agora são de facto detecções de outras estações na rede. O número de estações permanece quase o mesmo apesar do número de assinaturas

ter diminuído para menos de metade. O valor médio do gráfico agora situa-se muito próximo das 5 estações detectadas por assinatura.

Relativamente aos padrões observados antes no gráfico da Figura 6.1, estes permanecem os mesmos.

No histograma da Figura 6.7 está estabelecido um comparativo onde se pode observar as detecções acumuladas por hora do dia agora sem contabilização da própria estação (a azul) e com contabilização da própria estação (a vermelho), que está novamente representada.

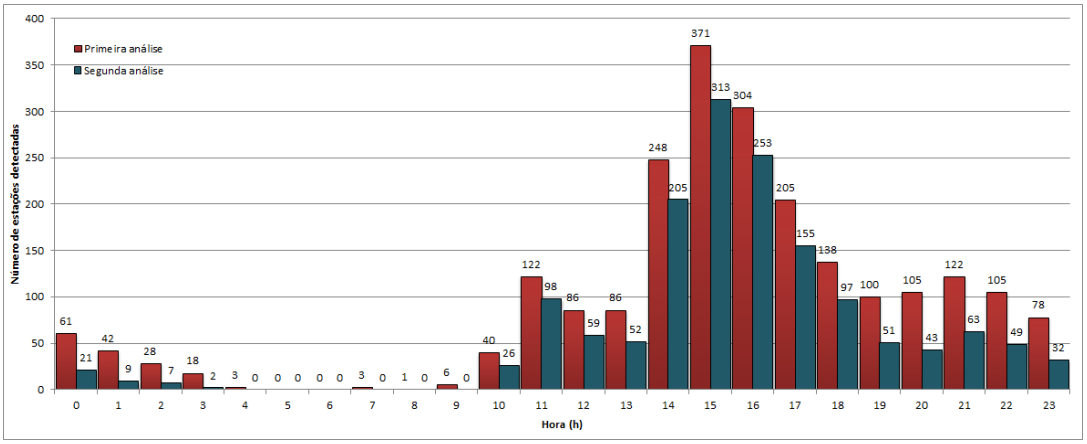


Figura 6.7: Comparativo das detecções acumuladas por horas do dia (com e sem contabilização da própria).

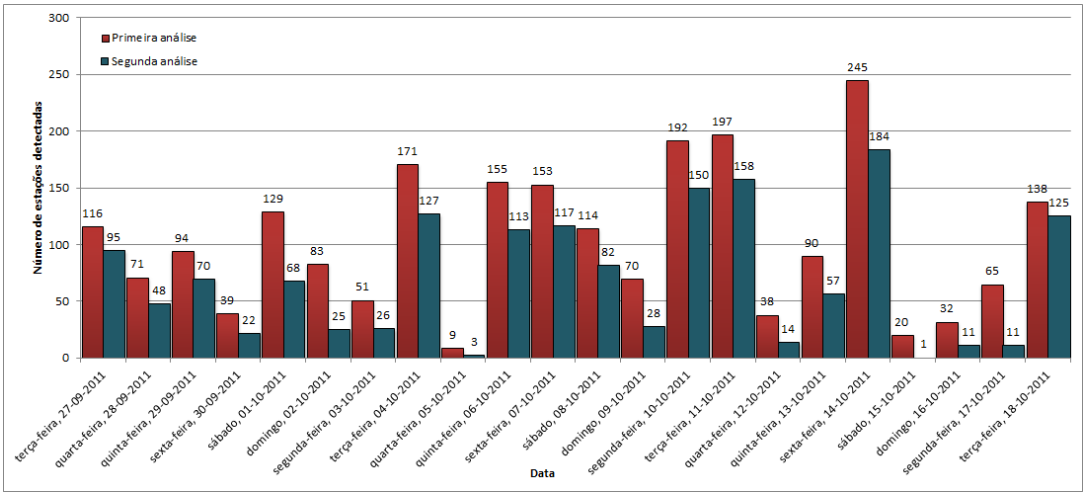


Figura 6.8: Comparativo das detecções acumuladas por dia (com e sem contabilização da própria).

Podemos observar que o comportamento também se manteve, assim como no

comparativo das detecções acumuladas por dia ilustrado na Figura 6.8, com a representação da contabilização sem a própria estação (a azul) e com a contabilização da própria estação (a vermelho).

No histograma da Figura 6.9 está ilustrada a frequência com que acontece o número de estações detectadas em assinaturas, agora sem a contabilização da própria estação.

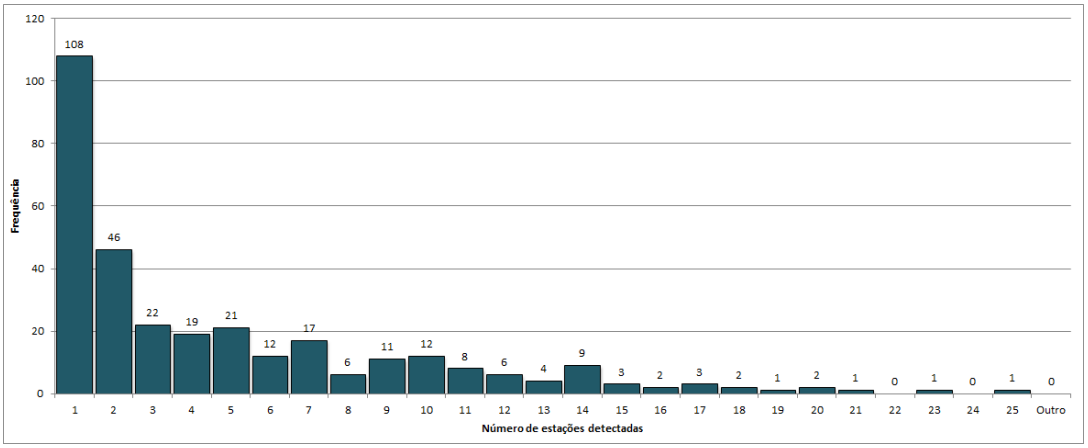


Figura 6.9: Frequência do número de estações detectados em assinaturas (sem contabilização da própria).

Comparativamente ao histograma do caso anterior, o número de vezes em que uma assinatura apenas continha uma única estação detectada diminuiu drasticamente. Isto já era de esperar, uma vez que já não são contabilizadas as assinaturas onde a própria estação aparecia sozinha.

A frequência com os restantes valores de número de estações detectadas em assinaturas mantêm-se muito parecidos ao caso anterior.

O próximo histograma, ilustrado na Figura 6.10, representa a frequência com que as estações foram detectadas nas assinaturas sem a contabilização da própria estação.

O número de vezes que uma estação foi detectada nas assinaturas, relativamente ao caso anterior que tinha uma estação detectada 312 vezes, diminuiu para 31, um valor 10 vezes menor. As restantes estações que também tinham um valor muito elevado também diminuíram drasticamente. As 3 pessoas envolvidas no desenvolvimento da aplicação Epi, que têm a aplicação a recolher assinaturas desde o primeiro dia e durante muito tempo, apenas detectavam a própria estação na maioria das recolhas.

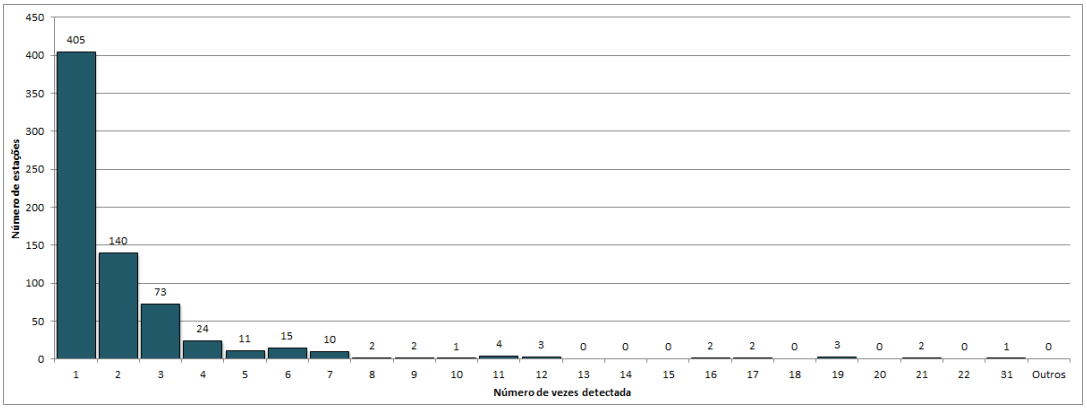


Figura 6.10: Frequência do número de vezes que as estações foram detectadas (sem contabilização da própria).

A frequência com que as restantes estações foram detectadas nas assinaturas permanece idêntico ao caso anterior.

6.2.3 Análise das recolhas dos utilizadores

Através das assinaturas submetidas ao servidor, durante o período de recolha de 3 semanas, foram detectados 16 utilizadores a usar a aplicação Epi. Estes utilizadores quando tinham a aplicação activa recolhiam e enviavam as assinaturas de rádio com as estações *WiFi* vizinhas para o servidor.

Para perceber como se está a comportar o módulo WifiRadar com as placas de rede sem fios agora num ambiente real é necessário analisar o número de assinaturas e número de estações detectadas por cada utilizador do Epi.

Para manter a privacidade dos utilizadores do Epi, estes foram associados a um número que os identifica.

Através do histograma da Figura 6.11, que ilustra o número de estações que cada utilizador Epi detectou, com e sem a contabilização do próprio, e do histograma da Figura 6.12, que ilustra o número de assinaturas que cada utilizador Epi recolheu, com e sem a contabilização do próprio, foi realizada uma análise de como está funcionar o WifiRadar.

Apesar de existirem 16 utilizadores, alguns têm um número muito baixo de assinaturas submetidas. Isto pode dever-se a serem utilizadores novos ou utilizadores muito ocasionais. Desses utilizadores não se pode tirar nenhuma conclusão.

Com o utilizador número 1, por exemplo, podemos observar que tem 1171 de-

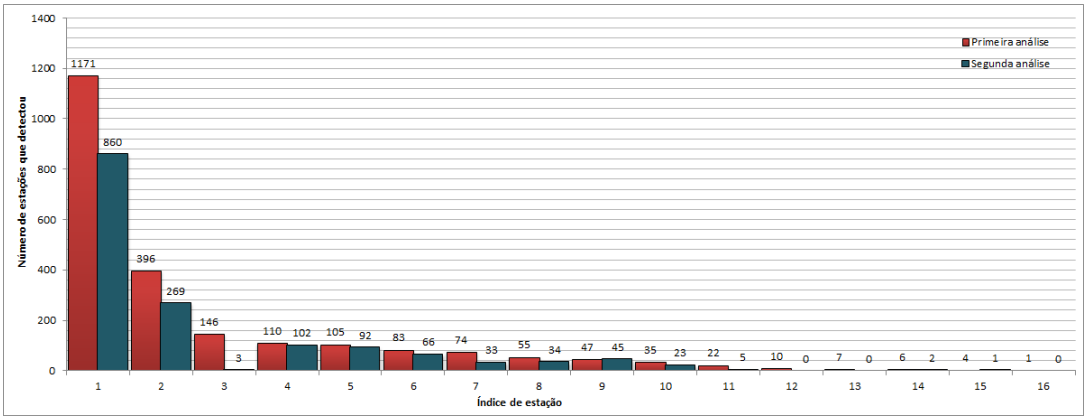


Figura 6.11: Número de estações detectadas pelos utilizadores Epi.

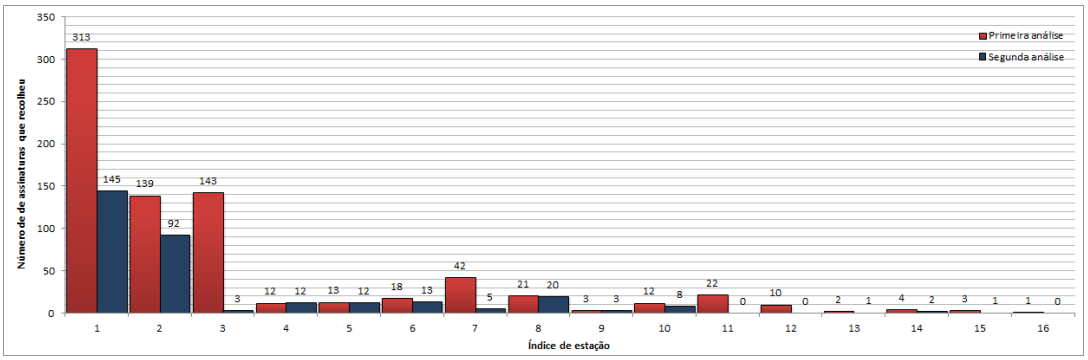


Figura 6.12: Número de assinaturas recolhidas pelos utilizadores Epi.

tecções de estações em 313 assinaturas submetidas o que corresponde a uma média de 3.74 estações por cada recolha que realiza. Se retirarmos as vezes que realizou uma recolha e apenas detectou ele próprio, a média sobe para 5.93. Ou seja, o módulo quase sempre que realiza uma recolha consegue detectar estações vizinhas.

Com o utilizador 2, a situação é a mesma mas com uma média menor de detecções.

Com o utilizador 3, detectou-se uma situação diferente em que a média de estações que detecta é 1. Este utilizador nas recolhas que realizou apenas se detectou a ele próprio. Para investigar mais o caso deste utilizador, através do endereço MAC, foi identificado o fabricante da placa de redes sem fios. Mais informação não seria possível obter sobre o sistema que era usado pelo utilizador.

A placa de rede sem fios do utilizador pertence ao fabricante Intel Corporation, que tinha sido identificado já na Secção 3.3.1 visto que, em pelo menos uma placa deste fabricante, não é possível realizar a captura de tráfego das tramas 802.11 necessárias para detectar as estações *WiFi* vizinhas.

Outro caso idêntico é o utilizador 10 e 13 que também possui uma placa do fabricante Intel Corporation. Nestes utilizadores o número de assinaturas submetidas é consideravelmente baixo.

Capítulo 7

Conclusão e trabalho futuro

Os objectivos propostos para esta dissertação foram cumpridos, uma vez que foram atingidos. O módulo está integrado na aplicação Epi que se encontra em distribuição e é capaz de realizar a descoberta de forma passiva das estações 802.11 vizinhas.

No decorrer desta dissertação foi estudado o protocolo de redes sem fios IEEE 802.11 que possui especial relevância uma vez que o objectivo era permitir a uma estação 802.11 descobrir outras estações 802.11 na sua vizinhança.

A principal dificuldade encontrada foi que a solução cumprisse os requisitos impostos, principalmente os necessários para a integração como um módulo de *software* na aplicação Epi. A abordagem inicial para satisfazer os requisitos levou à exploração de uma solução através da análise dos protocolos encontrados em tráfego capturado ao nível Ethernet. Esta solução acabou por consumir demasiado tempo e no final não serviu para cumprir os objectivos.

A solução final que acabou por ser adoptada passou pela captura de tráfego na camada MAC do 802.11, com recurso à ferramenta Microsoft Network Monitor. Esta disponibiliza uma API e apenas não satisfazia um requisito opcional.

Nesta dissertação foi implementado o protótipo de uma aplicação que recolhe as estações 802.11 vizinhas através da captura e processamento dos endereços nas tramas MAC 802.11 com recurso à API do Network Monitor.

Na aplicação Epi foi criada uma nova biblioteca baseada no protótipo, de forma a integrar na solução. Ao mesmo tempo foram integrados outros módulos, no âmbito de outras dissertações, realizados melhoramentos à aplicação Epi e Servidor e ainda foi integrado o Network Monitor na instalação da aplicação.

Para efectuar uma avaliação aos novos módulos, a aplicação Epi foi disponibilizada ao público novamente numa nova versão. Isto permitiu recolher as assinaturas de estações 802.11 vizinhas e obter alguns resultados sobre o módulo desenvolvido que foram ilustrados neste documento.

A privacidade na recolha de informação das assinaturas, ou em qualquer outra aplicação de recolha de informação, é uma questão delicada e que pode condicionar por vezes a participação de um maior número de pessoas nestas iniciativas. Nesta solução estamos conscientes que existem questões de privacidade, no entanto não as consideramos.

Como trabalho futuro existem algumas ideias no que diz respeito ao módulo desenvolvido:

- separar a solução do Network Monitor e desenvolver um *LightWeight Filter NDIS Driver* que copia as tramas 802.11 na camada MAC utilizando o *Windows Driver Kit* (WDK). Ao mesmo tempo desenvolver uma API para lidar com o *driver* e um motor de *parsing* para as tramas 802.11.

- processar o tráfego capturado na camada MAC do 802.11 por fluxo de tramas. Desta forma as tramas com apenas um endereço podem ser mais úteis.

- desenvolvimento para outras plataformas, MacOS X através do Wireshark, Linux, Android, iOS e Windows Phone 7.

No que diz respeito à aplicação:

- envio de imagens anexadas juntamente com as mensagens que são difundidas ao nível do utilizador.

- permitir saber os utilizadores Epi que se encontram por perto.

- envio de mensagens para um utilizador em específico.

- desenvolvimento para outras plataformas, MacOS X através do Wireshark, Linux, Android, iOS e Windows Phone 7.

Referências

[1] M. Gast, *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, Apr. 2002.

[2] IEEE, "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1 –1184, 12 2007.

[3] ——, "Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 5: Enhancements for higher throughput," *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pp. c1 –502, 29 2009.

[4] J. B. Postel and J. F. Reynolds, "Standard for the transmission of IP datagrams over IEEE 802 networks," Internet Engineering Task Force, RFC 1042, Feb. 1988. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1042.txt

[5] M. Ergen, "IEEE 802.11 Tutorial," *University of California Berkeley*, vol. 54, p. 70, 2002.

[6] A. Sevtsuk, S. Huang, F. Calabrese, and C. Ratti, "Mapping the MIT campus in real time using WiFi," *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, pp. 326–338, 2008.

[7] M. Afanasyev, T. Chen, G. M. Voelker, and A. C. Snoeren, "Analysis of a mixed-use urban wifi network: when metropolitan becomes neapolitan," in *Proceedings*

of the 8th ACM SIGCOMM conference on Internet measurement. ACM, 2008, pp. 85–98.

[8] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," *Computer Networks*, vol. 52, pp. 2690–2712, 2008.

[9] C. A. L. Sousa, "Difusão epidémica de mensagens em hotspots wifi," Master's thesis, Universidade do Minho, Guimarães, Portugal, Nov. 2010.

[10] (2011) Native wifi application programming interface. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms705945(v=VS.85).aspx

[11] (2011) Winpcap the industry-standard windows packet capture library. [Online]. Available: http://www.winpcap.org/

[12] (2011) Sharppcap is a cross platform .net assembly for interfacing with libpcap/winpcap. [Online]. Available: http://sourceforge.net/projects/sharppcap/

[13] (2011) Packet.net a high performance .net assembly for dissecting and constructing network packets. [Online]. Available: http://sourceforge.net/projects/packetnet/

[14] (2011) Managed wifi api .net class library allowing you to control wifi (802.11) network adapters. [Online]. Available: http://managedwifi.codeplex.com/

[15] J. Postel, "Internet control message protocol," Internet Engineering Task Force, RFC 792, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc792.txt

[16] A. Conta, S. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443 (Draft Standard), Internet Engineering Task Force, Mar. 2006, updated by RFC 4884. [Online]. Available: http://www.ietf.org/rfc/rfc4443.txt

[17] S. Deering, W. Fenner, and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6," RFC 2710 (Proposed Standard), Internet Engineering Task Force, Oct. 1999, updated by RFCs 3590, 3810. [Online]. Available: http://www.ietf.org/rfc/rfc2710.txt

[18] R. Vida and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6," RFC 3810 (Proposed Standard), Internet Engineering Task Force, Jun. 2004, updated by RFC 4604. [Online]. Available: http://www.ietf.org/rfc/rfc3810.txt

[19] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Standard), Internet Engineering Task Force, Sep. 2007, updated by RFC 5942. [Online]. Available: http://www.ietf.org/rfc/rfc4861.txt

[20] S. E. Deering, "Host extensions for IP multicasting," Internet Engineering Task Force, RFC 1112, Aug. 1989. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1112.txt

[21] W. Fenner, "Internet group management protocol, version 2," Internet Engineering Task Force, RFC 2236, Nov. 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2236.txt

[22] B. Cain, S. E. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet group management protocol, version 3," Internet Engineering Task Force, RFC 3376, Oct. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3376.txt

[23] H. Holbrook, B. Cain, and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast," RFC 4604 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4604.txt

[24] R. E. Droms, "Dynamic host configuration protocol," Internet Engineering Task Force, RFC 2131, Mar. 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2131.txt

[25] R. E. Droms, I. W. Ed., J. Bound, B. Volz, T. Lemon, C. E. Perkins, and M. Carney, "Dynamic host configuration protocol for IPv6 (DHCPv6)," Internet Engineering Task Force, RFC 3315, Jul. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3315.txt

[26] P. V. Mockapetris, "Domain names - concepts and facilities," Internet Engineering Task Force, RFC 1034, Nov. 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1034.txt

[27] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC 4033 (Proposed Standard), Internet Engineering Task Force, Mar. 2005, updated by RFC 6014. [Online]. Available: http://www.ietf.org/rfc/rfc4033.txt

[28] P. V. Mockapetris, "Domain names - implementation and specification," Internet Engineering Task Force, RFC 1035, Nov. 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1035.txt

[29] B. Aboba, D. Thaler, and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)," RFC 4795 (Informational), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4795.txt

[30] Y. Y. Goland, T. Cai, P. Leach, and Y. Gu, "Simple Service Discovery Protocol/1.0," Internet-Draft (work in progress), Apr. 1999, draft-cai-ssdp-v1-03.txt. [Online]. Available: http://tools.ietf.org/html/draft-cai-ssdp-v1-03

[31] Y. Y. Goland and C. Microsoft, "Multicast and unicast udp http messages," Internet-Draft (work in progress), Dec. 1999, draft-goland-http-udp-00.txt. [Online]. Available: http://tools.ietf.org/html/draft-goland-http-udp-00

[32] J. Cohen, S. Aggarwal, and Y. Y. Goland, "General event notification architecture base: Client to arbiter," Internet-Draft (work in progress), Jun. 1999, draft-cohen-gena-client-00.txt. [Online]. Available: http://tools.ietf.org/html/draft-cohen-gena-client-00

[33] S. Reddy, D. Lowry, S. Reddy, R. Henderson, J. Davis, and A. Babich, "Dav searching and locating, internet draft," IETF, Tech. Rep., June 1999, available at http://www.webdav.org/dasl/protocol/draft-dasl-protocol- 00.html.

[34] H. Nielsen, P. J. Leach, and S. Lawrence, "An HTTP extension framework," Internet Engineering Task Force, RFC 2774, Feb. 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2774.txt

[35] J. Beatty, G. Kakivaya, D. Kemp, T. Kuehnel, B. Lovering, B. Roe, C. S. John, J. S. (Editor), G. Simonnet, D. Walter, J. Weast, Y. Yarmosh, and P. Yendluri, "Web services dynamic discovery (ws-discovery)," Technical Documents, April 2005. [Online]. Available: http://msdn.microsoft.com/en-us/library/bb736562(v=VS.85).aspx

[36] H. Combs, M. G. (editor), J. Justice, G. Kakivaya, D. Lindsey, D. Orchard, A. Regnier, J. Schlimmer, S. Simpson, H. Tamura, D. Wright, and K. Wolf, "Soap-over-udp," Technical Documents, Sep. 2004, soap-over-udp.pdf. [Online]. Available: http://schemas.xmlsoap.org/ws/2004/09/soap-over-udp/

[37] N. W. G. in the Defense Advanced Research Projects Agency, I. A. Board, and E. to End Services Task Force, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods," Internet Engineering Task Force, RFC 1001, Mar. 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1001.txt

[38] ——, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications," Internet Engineering Task Force, RFC 1002, Mar. 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1002.txt

[39] C. Microsoft, "[ms-mail]: Remote mailslot protocol specification," Technical Documents, May 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/cc234511(v=PROT.13).aspx

[40] ——, "[ms-cifs]: Common internet file system (cifs) protocol specification," Technical Documents, May 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/ee442092(v=PROT.13).aspx

[41] ——, "[ms-smb]: Server message block (smb) protocol specification," Technical Documents, May 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/cc246231(v=PROT.13).aspx

[42] ——, "[ms-brws]: Common internet file system (cifs) browser protocol specification," Technical Documents, May 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/cc224428(v=PROT.13).aspx

[43] M. Crawford, "Transmission of IPv6 packets over ethernet networks," Internet Engineering Task Force, RFC 2464, Dec. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2464.txt

[44] S. Madhani, M. Tauil, and null Tao Zhang, "Collaborative sensing using uncontrolled mobile devices," *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, vol. 0, p. 8 pp., 2005.

[45] S. Pidcock, R. Smits, U. Hengartner, and I. Goldberg, "Notisense: An urban sensing notification system to improve bystander privacy," University of Waterloo.

[46] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *Proceedings of the 9th workshop on Mobile computing systems and applications*, ser. HotMobile '08. New York, NY, USA: ACM, 2008, pp. 11–16. [Online]. Available: http://doi.acm.org/10.1145/1411759.1411763

[47] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic Sensing: Security Challenges for the New Paradigm," in *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, Jan. 2009. [Online]. Available: http://www.cs.dartmouth.edu/~{}dfk/papers/kapadia-metrosec-challenges.pdf

[48] J. Júnior, M. Campista, and L. Costa, "Monitoramento colaborativo de trânsito utilzando redes ieee 802.11 em cidades inteligentes," Universidade Federal do Rio de Janeiro, 2007.

[49] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System," in *4th ACM SenSys*, Boulder, CO, November 2006.

[50] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan, "Wifi-reports: improving wireless network selection with collaboration," in *Proceedings of ACM MobiSys*. New York, NY, USA: ACM, Jun. 2009, pp. 123–136.

[51] I. M. P. Silva, "Avaliação da tecnologia bluetooth como sensor da mobilidade urbana," Master's thesis, Universidade do Minho, Guimarães, Portugal, Oct. 2011.

[52] A. Ensley. (2011) Csharp detect windows os version - part 2 wmi. [Online]. Available: http://andrewensley.com/2009/10/

[53] A. Peddemors and E. Yoneki, "Decentralized Probabilistic World Modeling with Cooperative Sensing," in *KiVS Workshop on Global Sensor Networks*. Citeseer, 2009.

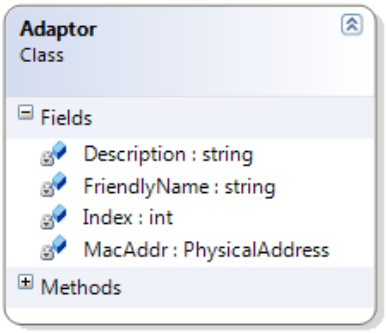
[54] V. Kostakos, T. Nicolai, E. Yoneki, E. O'Neill, H. Kenn, and J. Crowcroft, "Understanding and measuring the urban pervasive infrastructure," *Personal and Ubiquitous Computing*, vol. 13, pp. 355–364, 2008.

Anexos

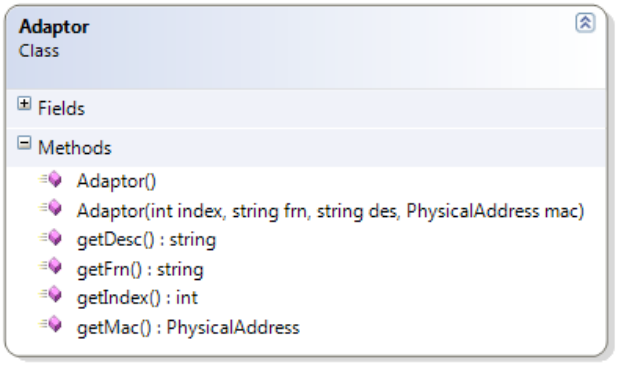
Anexo A

Classes do Módulo WifiRadar

A.1 Class Adaptor



(a) Fields da Class Adaptor do WifiRadar.



(b) Methods da Class Adaptor do WifiRadar.

Figura A.1: Class Adaptor do WifiRadar.

A.2 Class getOS

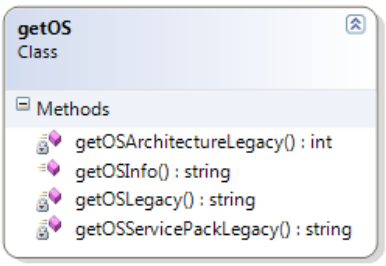
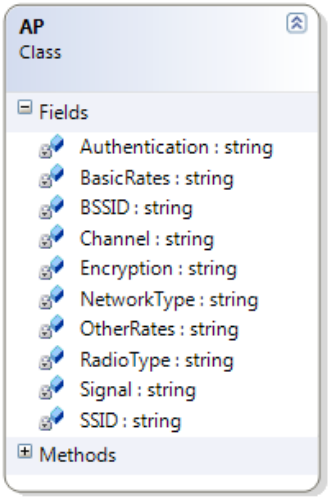
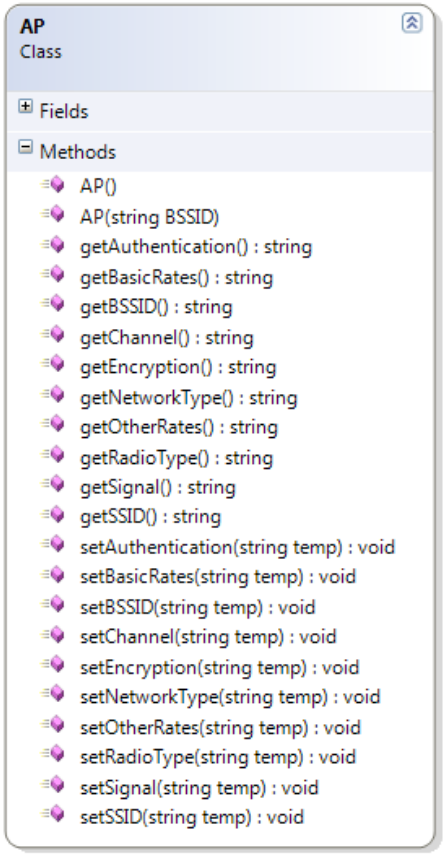


Figura A.2: Class getOS do WifiRadar.

A.3 Class AP



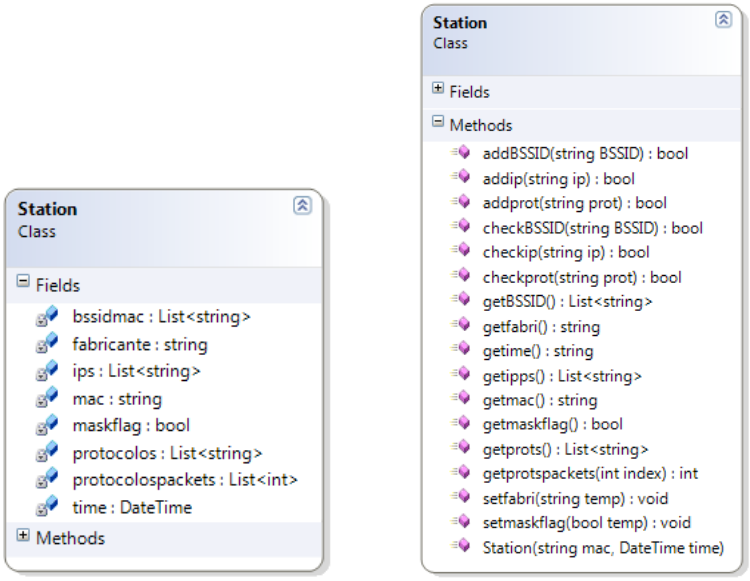
(a) Fields da Class AP do WifiRadar.



(b) Methods da Class AP do WifiRadar.

Figura A.3: Class AP do WifiRadar.

A.4 Class Station



(a) Fields da Class Station do WifiRadar.

(b) Methods da Class Station do WifiRadar.

Figura A.4: Class Station do WifiRadar.

A.5 Class WifiMon

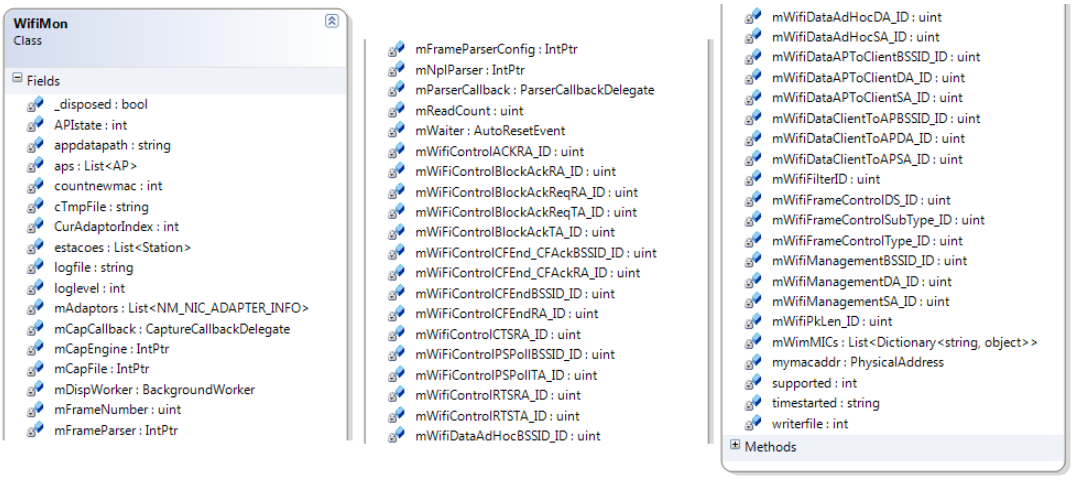


Figura A.5: Fields da Class WifiMon do WifiRadar.

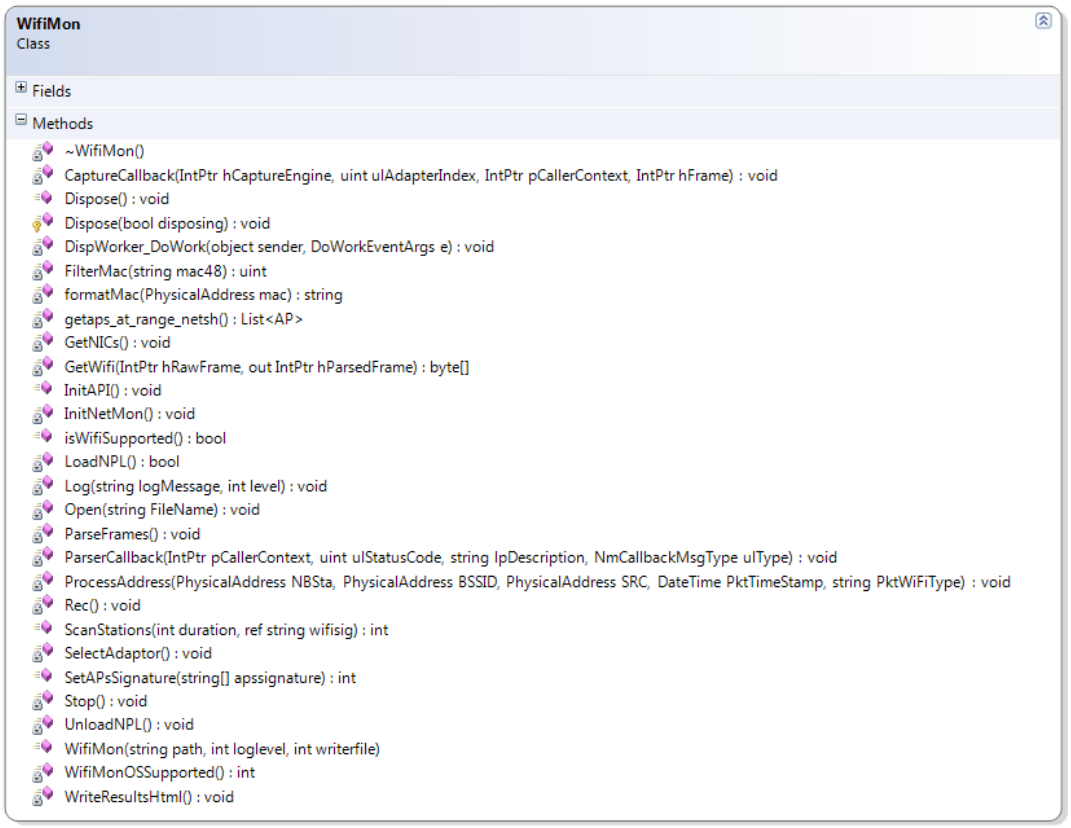


Figura A.6: Methods da Class WifiMon do WifiRadar.

Anexo B

Árvore de ficheiros da instalação do Epi

B.1 Árvore de instalação Epi 32 bits.

Caminho	Nome	Tipo
AppData/Epi	wifistalog.txt	ficheiro
	vizinhos.dat	ficheiro
	tmp.cap	ficheiro
	sparser.npl	ficheiro
	sparser.npb	ficheiro
	root.conf	ficheiro
	mensagens.dat	ficheiro
	configuracoes.dat	ficheiro
	assinaturas.dat	ficheiro
Application Folder	Wifiradar.dll	Assembly
	Uninstall.bat	ficheiro
	Epi.exe (Primary Output)	Output
	nmconfig.exe (32 bits)	ficheiro
	NMAPI.dll (32 bits)	ficheiro
	Metageek.IoctlNdis.dll	Assembly
	ManagedWifi.dll	Assembly
	JetPackWPFTheme.dll	Assembly
	InTheHand.Net.Personal.dll	Assembly

continua na próxima página

Caminho	Nome	Tipo
	Hardcodet.Wpf.TaskbarNotification.dll	Assembly
	epi.ico	ficheiro
	Epi Support.url	ficheiro
	BluetoothModule.dll	Assembly
SystemFolder/drivers	nm3.sys (32 bits)	ficheiro
ProgramMenuFolder/Epi	Uninstall	Atalho
	Epi Support	Atalho
	Epi	Atalho
StartupFolder	Epi	Atalho
WindowsFolder/inf	netnm3.inf	ficheiro

Tabela B.1: Árvore de instalação Epi 32 bits.

B.2 Árvore de instalação Epi 64 bits.

Caminho	Nome	Tipo
AppData/Epi	wifistalog.txt	ficheiro
	vizinhos.dat	ficheiro
	tmp.cap	ficheiro
	sparser.npl	ficheiro
	sparser.npb	ficheiro
	root.conf	ficheiro
	mensagens.dat	ficheiro
	configuracoes.dat	ficheiro
	assinaturas.dat	ficheiro
Application Folder	Wifiradar.dll	Assembly
	Uninstall.bat	ficheiro
	Epi.exe (Primary Output)	Output
	nmconfig.exe (64 bits)	ficheiro
	NMAPI.dll (64 bits)	ficheiro
	Metageek.IoctlNdis.dll	Assembly
	ManagedWifi.dll	Assembly
	JetPackWPFTheme.dll	Assembly
	InTheHand.Net.Personal.dll	Assembly
	Hardcodet.Wpf.TaskbarNotification.dll	Assembly
	epi.ico	ficheiro

continua na próxima página

Caminho	**Nome**	**Tipo**
	Epi Support.url	ficheiro
	BluetoothModule.dll	Assembly
System64Folder/drivers	nm3.sys (64 bits)	ficheiro
ProgramMenuFolder/Epi	Uninstall	Atalho
	Epi Support	Atalho
	Epi	Atalho
StartupFolder	Epi	Atalho
WindowsFolder/inf	netnm3.inf	ficheiro

Tabela B.2: Árvore de instalação Epi 64 bits.