Hand-off Automático em Redes WiFi

Versão Final

Por

Alexandre Mendes Xavier da Fonseca

Orientador: Doutor Pedro Miguel Mestre Alves da Silva

Co-orientador: Doutor Carlos Manuel José Alves Serôdio

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Eletrotécnica e de Computadores, de acordo com o disposto no
Regulamento Geral dos Ciclos de Estudo Conducentes ao Grau de Mestre na UTAD
DR, 2.ª série – N.º 133 – Regulamento n.º 658/2016 de 13 de julho de 2016

Hand-off Automático em Redes WiFi

Versão Final

Por

Alexandre Mendes Xavier da Fonseca

Orientador: Doutor Pedro Miguel Mestre Alves da Silva

Co-orientador: Doutor Carlos Manuel José Alves Serôdio

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Eletrotécnica e de Computadores, de acordo com o disposto no
Regulamento Geral dos Ciclos de Estudo Conducentes ao Grau de Mestre na UTAD
DR, 2.ª série – N.º 133 – Regulamento n.º 658/2016 de 13 de julho de 2016

Orientação Científica :

Doutor Pedro Miguel Mestre Alves da Silva

Professor Auxiliar do
Departamento de Engenharias da Escola de Ciências e Tecnologia da
Universidade de Trás-os-Montes e Alto Douro

Doutor Carlos Manuel José Alves Serôdio

Professor Associado c/Agregação do
Departamento de Engenharias da Escola de Ciências e Tecnologia da
Universidade de Trás-os-Montes e Alto Douro

"In the middle of difficulty lies opportunity."

Albert Einstein (1879 – 1955)

A quem dedico, ***Aos meus pais***

Hand-off automático em Redes Wifi

Alexandre Mendes Xavier da Fonseca

Submetido na Universidade de Trás-os-Montes e Alto Douro
para o preenchimento dos requisitos parciais para obtenção do grau de
Mestre em Engenharia Eletrotécnica e de Computadores

Resumo — Com as constantes exigências de qualidade nos serviços de Internet, o desenvolvimento e a otimização de técnicas e tecnologias *Wi-Fi*, torna-se preponderante para a satisfação dos seus utilizadores. Desde as suas versões primordiais, datadas de 1971, até às versões protocolares utilizadas nos dias de hoje, a utilização de *Wi-Fi* para aceder à Internet cresceu substancialmente nos últimos anos. Isto deve-se ao desenvolvimento de dispositivos móveis e também não-móveis, capazes de se ligar, sem-fios, à Internet e, consecutivamente, ao aumento exponencial de locais com cobertura *Wi-Fi*. Tendo em conta estas crescentes exigências, surge a necessidade de otimizar a metodologia de *handoff* utilizado presentemente nas redes *Wi-Fi*, visto que este tipo de processo, segundo os padrões IEEE802.11, se baseia maioritariamente na potência do sinal de cada AP (*Access Point*) que providencia acesso à *Internet*, fazendo com que a rede onde o utilizador está ligado possa não ser a mais adequada, ou seja, a que oferece melhor serviço. Isto deve-se ao facto de haver um grande número de variáveis numa rede *Wi-Fi* que podem fazer com que esta, apesar de poder ser a que apresenta uma potência de sinal superior, não corresponda ao melhor serviço de rede, pois outros fatores podem estar a afetar negativamente, deteriorando as suas capacidades. Foram feitos diversos ensaios, que provam que há um aumento de probabilidade de se ficar ligado a um AP que providencie um melhor serviço de *Internet* usando como parâmetro de escolha do AP, o atraso de rede.

Esta dissertação tem como objetivo desenvolver uma aplicação *Android* que tenha em conta o atraso da rede de modo a obter sempre o melhor serviço, a qualquer momento. A aplicação desenvolvida foi testada usando a *Eduroam* na UTAD (Universidade de Trás-os-Montes e Alto Douro).

Palavras Chave: IEEE802.11, Wi-Fi, Handoff, Android, Otimização.

Optimized handoff for Wifi Networks

Alexandre Mendes Xavier da Fonseca

Submitted to the University of Trás-os-Montes and Alto Douro
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computers

Abstract — With a worldwide constant service quality demands, regarding to the Internet, the development and optimization of Wi-Fi technologies is of extreme importance for the satisfaction of its users. Since its first versions, dated from 1971, until the latest protocol versions, the use of Wi-Fi to connect to the Internet skyrocketed in the recent years. This is due to the development of mobile and non-mobile devices, capable of connect wirelessly, to the Internet and, consequently, the exponential growth of spaces provided with Wi-Fi coverage. Taking into account these demands, came the need of optimize the handoff method that is used, nowadays, in Wi-Fi networks, because in this method, according to the IEEE802.11 standards, the AP switch is mainly based on the power of each AP that provides Internet access, and thus can make the network that the user is connected, not optimal. This is because there are a great number of variables in a Wi-Fi network that can make the network not optimal despite having higher received power, and that variables can negatively affect their capabilities. Several tests were made. Those tests proved that the probability of a device that is connected to an AP that provides the best Internet service using the network delay as a selection criteria, improves considerabily.
The goal of this thesis is to develop an Android application that takes into consideration the network delay for the choice of the best network access, at anytime. The developed application was tested in the Eduroam network in UTAD.
Keywords: IEEE802.11, Wi-Fi, Handoff, Android, Otimization.

Agradecimentos

Em primeiro lugar quero agradecer aos meus orientadores, Professor Doutor Pedro Miguel Mestre Alves da Silva do Departamento de Engenharia da Escola de Ciência e Tecnologia da Universidade de Trás os Montes e Alto Douro e Professor Carlos Manuel José Serôdio do Departamento de Engenharia da Escola de Ciência e Tecnologia da Universidade de Trás os Montes e Alto Douro. A sua disponibilidade e orientação ao longo de todo o processo de desenvolvimento desta dissertação revelou-se basilar para o sucesso da mesma, servindo também como guia para a acumulação de conhecimento e desenvolvimento pessoal que terá consequências positivas no meu futuro.

Quero agradecer aos meus pais, pelo apoio incondicional em todo o meu percurso académico bem como o seu esforço durante todo este tempo que permitiu que tivesse acesso a todas as ferramentas necessárias para ter a melhor educação possível.

Quero agradecer à minha namorada Raquel Almeida, por toda a paciência e apoio durante este percurso académico providenciando motivação e encorajamento.

Quero agradecer à minha amiga Alexandra Nunes, que fez valer a sua experiência e conhecimento, bem como motivação, de modo a acentuar o sucesso desta dissertação.

Por último, quero agradecer a todos os meus amigos que tornaram possível todo este

desenvolvimento a nível pessoal e académico, em especial àqueles que me acompanharam mais de perto durante estes anos.

UTAD/IPL, Alexandre Fonseca
Vila Real, fevereiro 2019

Índice geral

Resumo ix

Abstract xi

Agradecimentos xiii

Índice de tabelas xvii

Índice de figuras xix

Glossário, acrónimos e abreviaturas xxi

1 Introdução 1
- 1.1 Contexto . . . 1
- 1.2 Motivação e Objetivos . . . 2
- 1.3 Estrutura do Documento . . . 3

2 Enquadramento Tecnológico 5
- 2.1 Metodologias de *Handoff* . . . 5
  - 2.1.1 *Handoff* em Sistemas Celulares . . . 5
  - 2.1.2 Desempenho do *Handoff em Redes Wi-Fi* . . . 12
  - 2.1.3 Trabalhos Relacionados . . . 15
- 2.2 Redes *Wi-Fi* . . . 23
  - 2.2.1 Contexto e História . . . 23
  - 2.2.2 Padrão IEEE 802.11 . . . 24

2.2.3 Padrão IEEE 802.1X . . . 30
2.2.4 Eduroam . . . 31
2.3 Android . . . 34
2.3.1 Visão Geral do *Android* . . . 34
2.3.2 Ferramentas do Android . . . 36
2.4 Network Time Protocol . . . 38

3 Conceção **45**
3.1 Funcionamento da experiência . . . 46
3.2 Desenvolvimento da Aplicação *Android* . . . 54

4 Testes e Resultados **65**
4.1 Resultados da Experiência . . . 65
4.2 Demonstração do Funcionamento da Aplicação . . . 71

5 Conclusões e Trabalho Futuro **79**
5.1 Conclusões . . . 79
5.2 Trabalho Futuro . . . 80

Referências Bibliográficas **83**

Índice de tabelas

2.1 Informações sobre o modo de funcionamento de diferentes versões do protocolo 802.11 28

3.1 Tabela com exemplos de dados recolhidos. [1] 49

4.1 Tabela de resultados com os dados obtidos no cenário 1. 66
4.2 Tabela de resultados com os dados obtidos no cenário 2. 67
4.3 Tabela do cenário 3 para diferentes tamanhos de amostra do atraso de rede. 69

Índice de figuras

2.1 Handoff num sistema celular[2] 7
2.2 Handoff no protocolo 802.11 [3] 13
2.3 Atrasos de *handoff* [3] 14
2.4 Selective Scan [4] 19
2.5 Selective Cache [4] 21
2.6 Formato da Frame do padrão IEEE802.11 [5] 29
2.7 Funcionamento do sistema Radius. [6] 32
2.8 Utilização do sistema operativo Android e iOS. [7] 35
2.9 Estratos do NTP. [8] 40
2.10 Modo de implementação NTPv4. [9] 41
2.11 Troca de *timestamps* entre cliente e servidor. 42

3.1 Arquitetura da experiência. [1] 47
3.2 Velocidade de download em função do valor de RSSI para um dos cenários de teste. [1] 50
3.3 Velocidade de download em função do valor de atraso de rede, para um dos cenários de teste. [1] 51

3.4 Velocidade de download em função do valor do melhor atraso de rede em cada ponto, para um dos cenários de teste. [1] 52
3.5 Comparação da velocidade de download quando a escolha de AP é baseada no valor de RSSI ou no valor de atraso da rede. 53
3.6 Princípio de funcionamento da aplicação. 55
3.7 Funcionamento da aplicação. 56
3.8 *Layout* da primeira atividade de aplicação. 57
3.9 *Layout* da segunda atividade de aplicação. 58
3.10 *Layout* da terceira atividade de aplicação. 59
3.11 Configuração da rede para se ligar à Eduroam. 60
3.12 Fluxograma mais detalhado do funcionamento da aplicação. 62

4.1 Comparação entre *Best, Good* e *Worse* (Melhor, Bom e Pior, tradução para português) seleção de rede, para diferentes tamanhos de amostra. 70

Glossário, acrónimos e abreviaturas

Lista de acrónimos

Sigla	**Expansão**
AP	*Access Point*
UTAD	Universidade de Trás-os-Montes e Alto Douro
RSS	*Received Signal Strenght*
SSID	*Service Set IDentifier*
NTP	*Network Time Protocol*
MS	*Mobile Station*
BTS	*Base Station Transceiver*
BS	*Base Station*
BSC	*Base Station Controller*
BSS	*Base Station Subsystem*
MSC	*Mobile Switching Center*
VLR	*Visitation Location Register*
HLR	*Home Location Register*

Sigla	**Expansão**
AuC	*Authentication Center*
SIM	*Subscriber Identity Module*
EIR	*Equipment Identification Register*
NSS	*Network Switching Subsystem*
GMSC	*Gateway MSC*
PSTN	*Public Switched Telephone Network*
ISDN	*Integrated Services Digital Network*
GSM	*Global System for Mobile Communications*
UMTS	*Universal Mobile Telecommunications System*
LTE	*Long Term Evolution*
MCHO	*Mobile Controlled Handoff*
DECT	*Digital Enhanced Cordless Telecommunication*
NCHO	*Network Controlled Handoff*
MAHO	*Mobile Assisted Handoff*
CDMA	*Code Division Multiple Access*
TDMA	*Time Division Multiple Access*
FDMA	*Frequency Division Multiple Access*
OFDMA	*Orthogonal Frequency Division Multiple Access*
GPRS	*General Packet Radio Service*
WCDMA	*Wideband Code Division Multiple Access*
WiMAX	*Worldwide Interoperability for Microwave Access*
MAC	*Medium Access Control*
IAPP	*Inter Access Point Protocol*
WLAN	*Wireless Local Area Network*
NG	*Neighbour Graph*
VoIP	*Voice over Internet Protocol*
ISM	*Industrial, Scientific and Medical*

Sigla	**Expansão**
GPS	*Global Positioning System*
RF	*Radio Frequency*
ISP	*Internet service provider*
UHF	*Ultra High Frequency*
FCC	*Federal Communications Commission*
PHY	*PHYsical*
DCF	*Distributed Coordination Function*
PCF	*Point Coordination Function*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
DIFS	*DFC Inter-Frame Space*
ACK	*ACKnowledge*
SIFS	*Short Inter-Frame Space*
CSMA/CD	*Carrier Sense Multiple Access with Collision Detection*
PIFS	*PCF Inter-Frame Space*
RTS	*Request to Send*
CTS	*Clear to Sendo*
DSSS	*Direct Sequence Spread Spectrum*
FHSS	*Frequency Hopping Spread Spectrum*
OFDM	*Orthogonal Frequency Division Multiplexing*
MIMO	*Multiple Input-Multiple Output*
DS	*Distribution System*
BSSID	*Basic Service Set IDentifier*
EAP	*Extensible Authentincaion Protocol*
RADIUS	*Remote Authentication Dial-In User Service*
PC	*Personal Computer*
TLS	*Transport Layer Security*
TTLS	*Tunneled Transport Layer Security*

Sigla	**Expansão**
PAP	*Password Authentication Protocol*
MS-CHAP	*Microsoft Challenge Handshake Authentication Protocol*
CHAP	*Challenge Handshake Authentication Protocol*
PPP	*Point-to-Point Protocol*
PEAP	*Protected Extensible Authentication Protocol*
PWD	*Password*
FAST	*Flexible Authentication via Secure Tunneling*
PAC	*Protected Authentication Credential*
IDE	*Integrated Development Environment*
API	*Application Programming Interfaces*
IP	*Internet Protocol*
RSSI	*Received Signal Strength Indicator*
UTC	*Coordinated Universal Time*
VFO	*Variable Frequency Oscillator*
ICMP	*Internet Control Message Protocol*
UDP	*User Datagram Protocol*
UI	*User Interface*

Lista de abreviaturas

Abreviatura	**Significado(s)**
e.g.	por exemplo
et al.	e outros (autores)
i.e.	isto é, por conseguinte

1 Introdução

1.1 Contexto

A utilização de sistemas *Wi-Fi* tornou-se tão recorrente e difundida pelo mundo, que já é difícil imaginar a nossa vida sem o acesso a esta tecnologia. Em países considerados desenvolvidos, grande parte dos locais, privados e públicos nomeadamente cafés, restaurantes, hospitais e universidades, permitem acesso ao *Wi-Fi*. Cada vez que nos movimentamos numa rua principal de qualquer cidade ou vila, e se se fizer um *scan* com qualquer dispositivo móvel, é quase certo que se encontra dezenas de diferentes redes *Wi-Fi* disponíveis.

Atualmente, o desenvolvimento tecnológico ocorre a uma velocidade arrebatadora. O que é considerado bom, novo e topo de gama num dia, rapidamente passa a obsoleto e descartável. Visto que a população em geral gosta de se manter atualizada e de estar a par das novidades tecnológicas, tais como *smartphones* e computadores portáteis, é preciso desenvolver metodologias de apoio ao desempenho deste tipo de dispositivos.

Por estes motivos é importante estar sempre atualizado e na vanguarda da tecnologia, estudando maneiras de resolução de problemas que tornem os dispositivos mais

rápidos e eficientes, bem como métodos que permitam uma otimização no desempenho tanto desses dispositivos, como dos serviços que advêm da sua utilização, de modo a que seja garantida uma boa experiência do ponto de vista dos utilizadores.

No âmbito desta dissertação, é desenvolvida uma aplicação *Android* que permita otimizar o processo de *handoff* em redes *Wi-Fi*.

1.2 Motivação e Objetivos

O método atual de *handoff* em redes *Wi-Fi*, é maioritariamente baseado no RSS (*Received Signal Strenght*), ou seja, na potência do sinal proveniente do AP, desprezando assim, outro tipo de variáveis que podem ter influência no desempenho da rede e na escolha do melhor AP para se conectar [10].

Quando estamos perante uma rede com um determinado SSID (*Service Set IDentifier*), configurada em múltiplos APs, os dispositivos móveis conectam-se ao AP que apresente um maior valor de potência de sinal. Apesar de haver uma grande possibilidade deste método de escolha coincidir com o AP que fornece o melhor serviço de rede, mesmo quando todas as variáveis são consideradas, ensaios realizados ao longo desta dissertação mostram que isso nem sempre acontece. Caso o dispositivo se comece a mover, a potência de sinal do AP começa a decrescer mesmo quando há APs que se apresentem com um maior valor de potência de sinal. O dispositivo só muda de AP quando o valor da potência de sinal decresce abaixo de um certo nível de *threshold*. Se o dispositivo parar de se mover antes desse nível de *threshold* ser atingido, vai continuar conectado ao AP inicial que agora, apresenta níveis de potência não ótimos. Isto significa que os serviços providenciados pela rede poderão não estar a ser prestados em condições ótimas. Visto que a exigência em termos de qualidade de serviço, por parte dos utilizadores de redes *Wi-Fi* em dispositivos móveis é crescente, foi sugerido o desenvolvimento de uma aplicação com intuito de otimizar o método de *handoff* em redes *Wi-Fi*, em dispositivos *Android*.

O objetivo desta dissertação é desenvolver uma aplicação *Android* que tenha em

consideração a variável de atraso de rede para que, desta maneira, o dispositivo esteja sempre ligado ao melhor AP fazendo com que lhe seja providenciado o melhor serviço de *Internet*, estando este estacionário ou em mobilidade. A aplicação vai ser desenvolvida com intuito de ser utilizada em dispositivos com o sistema operativo *Android*, sendo que a rede que vai servir como caso de estudo onde vão ser realizados todos os ensaios é a rede *Eduroam* da UTAD e para vai ser usando o protocolo NTP *(Network Time Protocol)* para obter o atrasado de rede quando é pedido para que essa informação seja usando como parâmetro de escolha do AP.

1.3 Estrutura do Documento

A dissertação desenvolvida está dividida em seis capítulos. No presente capítulo 1, Introdução, está apresentado o contexto da dissertação, a motivação e os objetivos do trabalho bem como a descrição detalhada de todo o documento.

No capítulo 2, Enquadramento Tecnológico, é feita uma descrição detalhada das metodologias e características dos diferentes tipos de *handoff*, nomeadamente nos sistemas celulares e nas redes *Wi-Fi*. Estão também apresentados diversos trabalhos relacionados diretamente com o tema desta dissertação que são relevantes para entender os desafios que poderão surgir no desenvolvimento da mesma. É também feita uma descrição das redes *Wi-Fi* bem como uma abordagem aos padrões utilizados neste tipo de rede, nomeadamente os padrões IEEE802.11 e IEEE802.1X. É descrita a rede *Eduroam*, algumas das suas características nomeadamente o seu funcionamento e protocolos de segurança e de autenticação que são usados pela rede. Ainda neste capítulo, são abordadas as ferramentas *Android* que foram utilizadas na conceção da aplicação. Por último, é referido o protocolo NTP sendo descrito o seu funcionamento e a sua utilidade no âmbito desta dissertação.

No capítulo 3, Conceção, é descrito o estudo que levou à elaboração da aplicação bem como os passos da implementação da mesma. No estudo apresentado foram desenvolvidos vários cenários de teste que comprovam o melhoramento do desempenho

de *Internet* quando é usado o atraso de rede como parâmetro prioritário de escolha do AP. A partir das conclusões desse estudo é explicada a conceção da aplicação *Android*.

No capítulo 4, Testes e Resultados, são descritos detalhadamente os resultados do estudo referido no capítulo anterior. Posteriormente são demonstrados os passos da aplicação à medida da sua execução, referidos alguns problemas encontrados durante o seu desenvolvimento e, por fim, as soluções utilizadas para a sua resolução.

No capítulo 5, Conclusões e Trabalho Futuro, são apresentadas algumas conclusões sobre os resultados obtidos durante o desenvolvimento da dissertação bem como possíveis melhoramentos que possam vir ser implementados de modo a otimizar o seu funcionamento.

2 Enquadramento Tecnológico

2.1 Metodologias de *Handoff*

O *handoff*[1] ou *handover*[2] [11], utilizado em comunicações sem fios é o termo denominado para a passagem de um ponto de comunicação para outro [12]. Este tipo de procedimento é essencial para a continuação de qualidade de serviços quando existe mobilidade por parte do dispositivo. Muito antes de haver *handoff* em redes *Wi-Fi* já existia *handoff* para comunicações sem fios nomeadamente para sistemas celulares. Neste capítulo será abordado o funcionamento do *handoff* tanto em sistemas celulares, como em redes Wi-Fi.

2.1.1 *Handoff* em Sistemas Celulares

As redes de comunicações sem fios são basilares para o funcionamento de aplicações móveis. Os sistemas celulares permitem a utilização de serviços em grandes áreas, dividindo esse espaço em pequenos *clusters* de células. Nos sistemas primordiais, em vez de se utilizar arquitetura celular para garantir a cobertura de grandes áreas,

[1] Termo utilizado em inglês americano
[2] Termo utilizado em inglês britânico

eram necessárias estações que debitassem uma grande potência de maneira a garantir cobertura em localizações mais distantes. Com a introdução de sistemas celulares, a mudança no desenho da arquitetura veio trazer diversas vantagens, nomeadamente [13]:

- Aumento de capacidade, i.e aumento do número de utilizadores;
- A potência de transmissão necessária é muito inferior;
- Maior robustez;
- Topologia descentralizada, pois cada estação-base gere os problemas da sua célula;
- Reutilização de frequências, permitindo um conjunto de frequências seja utilizado em células diferentes, desde que obedeçam a uma razão de distância.

Com a divisão em células, ao invés de ser utilizada uma estação de rádio com uma potência muito elevada para cobrir uma grande área, é utilizada uma estação base em cada célula que vai gerir os problemas de logística que possam surgir, e.g. interferências e nível de potência. Contudo, a introdução deste tipo de mecanismo pode trazer algumas desvantagens:

- Interferências com outras células;
- Necessidade de haver uma rede fixa que controle as estação-base e as interligue.

Na Figura 2.1 está representada o processo de *handoff* num sistema celular.

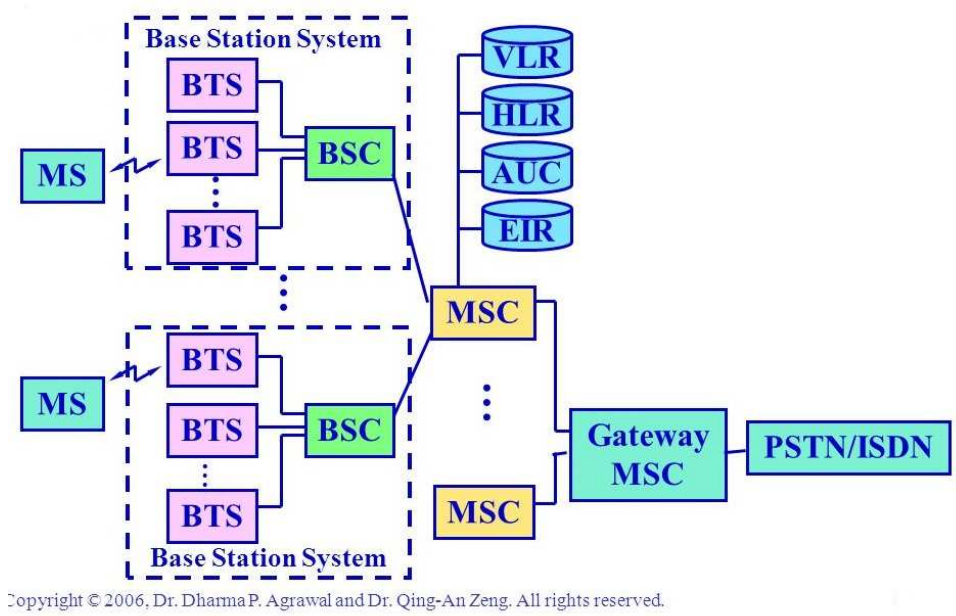

Figura 2.1 – Handoff num sistema celular[2]

Como se pode ver na Figura 2.1, em caso de mobilidade por parte do dispositivo móvel, que segundo a literatura tem a denominação de MS *Mobile Station*, surge a necessidade de trocar de célula. Os vários componentes da figura são os seguintes:

- MS (*Mobile Station*) - Estação ou terminal móvel, e.g telemóvel;
- BTS ou BS (*Base Transceiver Station* ou *Base Station*) - Estação base, que funciona como interface entre a estação móvel e a rede. Contêm equipamento para transmissão e receção de sinal;
- BSC (*Base Station Controller*) - Tem como funções principais controlar a BTS e gestão das frequências. Estes dois últimos componentes fazem parte da chamada BBS (*Base Station Subsystem*);
- MSC (*Mobile Switching Center*) - É o nó de comutação da rede móvel e tem como funções principais sinalizar a necessidade de *handoff* e tarifação e especificação de serviços;

- VLR (*Visitation Location Register*) - Base de dados onde são guardados os dados de todos as MS que estão registadas na sua MSC;

- HLR (*Home Location Register*) - Base de dados onde estão guardados os dados e as identidades de todos os utilizadores na área pertencente à MSC. A cada HLR está associado um AuC *(Authentication Center)* que é o centro de autenticação que possui uma cópia de código de cada cartão SIM *(Subscriber Identity Module)* para a autenticação e encriptação das comunicações;

- EIR (*Equipment Identification Register*) - Base de dados que contêm cópias dos números de identificação dos dispositivos. Com estas cópias define listas de validação dos mesmos. Estes últimos componentes (MSC, VLR, HLR, EIR) fazem parte da denominada NSS (*Network Switching Subsystem*);

- GMSC (*Gateway MSC*) - Serve de interface do sistema com a *PSTN (Public Switched Telephone Network)/ISDN (Integrated Services Digital Network).*

- PSTN/ISDN - A PSTN é a estrutura que providencia serviços de voz para estas redes. A ISDN é um conjunto de padrões que permite transmissões de dados, voz, vídeo, entre outros, em cima da estrutura PSTN. A grande diferença entre as duas é que as linhas PSTN são analógicas e no caso da ISDN da ISDN são digitais.

Ao longo dos anos, diferentes tipos de sistemas de comunicações celulares têm sido desenvolvidos, nomeadamente *GSM (Global System for Mobile Communications), UMTS (Universal Mobile Telecommunications System), LTE (Long Term Evolution)*, entre outros. Cada tipo de sistema utiliza o seu método e estratégia de *handoff*. De seguida são abordados os diferentes tipos e estratégias de handoff, que servirão como contexto para perceber o seu funcionamento nas diversas áreas tecnológicas.

Estratégias de *Handoff*

A decisão para a iniciação do processo de *handoff* é baseada através de medições. Estas medições têm como parâmetro de escolha de estação base a potência do sinal entre a estação móvel e a estação base atual, bem como as estações base vizinhas. Também existem outros parâmetros tais como a distância entre as anteriores e o volume de tráfego [14]. Dependendo destas medições mas também dos sistemas de comunicações que estão a ser utilizados, o processo de decisão de *handoff* pode ser feito a partir de três estratégias [11][15]. Essas estratégias são as seguintes:

- *MCHO (Mobile Controlled Handoff)* - Neste tipo de estratégia, a responsabilidade de monitorização recai no dispositivo móvel. Este, continuamente, monitoriza os sinais provenientes de estações bases circundantes e procede à troca quando vários critérios se juntam (e.g potência de sinal, interferências). É utilizado no padrão *DECT (Digital Enhanced Cordless Telecommunication)*;

- *NCHO (Network Controlled Handoff)* - Nesta estratégia, as estações base têm a responsabilidade de medir a potência de sinal vinda da estação móvel e todo o processo de decisão de *handoff* é feita por parte da rede. Esta estratégia era utilizada por sistemas analógicos de primeira geração;

- *MAHO (Mobile Assisted Handoff)* - Aqui há uma divisão de tarefas onde a rede pede à estação móvel para proceder à realização de medições dos sinais provenientes das estações base circundantes. É utilizado nas tecnologias de segunda geração tais como *GSM* e a primeira versão de *CDMA (Code Division Multiple Access)*.

Tipos de *Handoff*

Os tipos de *handoff* dividem-se maioritariamente em duas categorias: *Hard Handoff* e *Soft Handoff* [11].

O *Hard Handoff* é um tipo de troca que usa a tecnologia *break-before-make*. Isto quer dizer que, quando a estação móvel está entre estações base e é tomada a decisão de efetuar a troca, a ligação com a estação base atual é terminada antes de estabelecer contacto com a estação base a que se deseja ligar. Isto significa um corte de comunicações significando que com este tipo de *handoff*, uma estação móvel está ligada a uma, e uma só estação base em qualquer momento. O momento no qual se dá a transferência, é quando a potência baixa de um certo nível de *threshold* pré-definido. O *Hard Handoff* tem as seguintes características:

- Usa apenas um e só um canal de frequências;
- É usado em sistemas que usam técnicas de acesso ao meio tais como *TDMA (Time Division Multiple Access)*, *FDMA (Frequency Division Multiple Access)* e *OFDMA (Orthogonal Frequency Division Multiple Access)*. Esses sistemas são, por exemplo, GSM e *GPRS (General Packet Radio Service)*;
- Se houver chamada em curso por parte da estação móvel durante a troca, a mesma é terminada durante a troca, devido à quebra de ligação da estação onde estava ligada;
- É um tipo de *handoff* de barata e simples implementação;
- Sendo que há a possibilidade de a ligação ser terminada, este tipo de *handoff* é percetível ao utilizador.

O *Soft Handoff* é um tipo de troca que usa a tecnologia *Make-before-break*. Ao contrário do *Break-before-make*, quando a estação móvel está entre duas estações base, antes de se desconectar com aquela a que estava previamente ligada, estabelece ligação com uma ou várias estações base circundantes antes de efetuar a troca. As estações base são adicionadas a uma lista de potenciais pontos de ligação quando a sua potência sobe acima de um determinado valor de *threshold* e são retiradas quando o seu valor de potência baixa de outro determinado nível de *threshold*. Desta forma é garantida a continuidade das comunicações que poderão estar a ser

feitas durante o momento da troca. O *Soft handoff* tem as seguintes características:

- Usa vários canais de frequências;
- É usado em sistemas que usam técnicas de acesso ao meio tais como CDMA e WCDMA (*Wideband CDMA*). Esses sistemas são por exemplo o UMTS e o WiMAX (*Worldwide Interoperability for Microwave Access*);
- A fiabilidade de conexão é elevada;
- A probabilidade da ocorrência de falhas é reduzida;
- É um sistema bastante complexo.

Comparando estes tipos de *handoff* em sistemas celulares com o *handoff* em redes *Wi-Fi* pode-se dizer que o *soft handoff* não é possível em redes *Wi-Fi* visto que, segundos os padrões 802.11, não é possível para um dispositivo estar ligado a mais do que um AP em qualquer momento. A partir daqui, conclui-se que o tipo de *handoff* utilizado em redes *Wi-Fi* é o *hard handoff*. A transição entre APs de uma mesma rede acontece, maioritariamente, com base na potência de sinal desses mesmos APs, sendo que quando a potência de sinal de um dado AP baixa de um certo nível predefinido de *threshold*, o dispositivo associa-se a outro com um maior valor de potência de sinal.

Os tipos de *handoff* também podem ser divididos noutras duas categorias, vertical e horizontal.

- ***Handoff* horizontal** - É um tipo de *handoff* que ocorre quando os pontos de acesso aos quais o dispositivo se quer ligar pertencem ao mesmo tipo de rede, e.g., duas estações base de uma rede 3G [15]. Existem dois tipos de *handoff* horizontal:

  - ***Handoff* Intracélula** - É um tipo de *soft handoff* que ocorre dentro de uma célula quando a estação móvel muda de canal para minimizar a interferência entre canais dentro da mesma estação base;
  - ***Handoff* Intercélula** - É um tipo de *hard handoff* que ocorre quando a estação móvel passa para a célula adjacente transferindo-se de uma estação base para outra.

- ***Handoff* Vertical** - Ao contrário do *handoff* horizontal é um processo que ocorre quando há transferência da conexão da estação móvel entre diferentes tecnologias sem fios. Há dois tipos de *handoff* vertical:

  - ***Handoff* Vertical *Downward*** - É um tipo de *soft handoff* que ocorre quando a estação móvel troca para uma rede com uma maior largura de banda mas uma cobertura limitada;
  - ***Handoff* Vertical *Upward*** - É um tipo de *hard handoff* que, ao contrário do *Handoff* Vertical *Upward*, a estação móvel passa para uma rede com menor largura de banda mas uma cobertura mais abrangente.

2.1.2 Desempenho do *Handoff em Redes Wi-Fi*

Os autores [3] fizeram um estudo do desempenho do *handoff* em redes *Wi-Fi* na camada de ligação, observando que a maior causa de atraso no processo de *handoff* é a função de *probe*, na camada *MAC (Medium Access Control)*. Numa primeira abordagem, os autores definem o processo de *handoff* como uma função da camada física onde o dispositivo a efetuar a troca, fica com acesso à Internet a partir do AP posterior, depois de ter tido acesso a partir do AP anterior. No artigo apresentado, os autores dividiram os passos do *handoff* em dois:

1. **Descoberta** - Quando há mobilidade por parte do dispositivo móvel, a potência de sinal proveniente do AP pode-se começar a degradar, conforme o afastamento, sendo que, a certo ponto, a conetividade do dispositivo com esse

AP possa estar em risco. Nesta altura, o dispositivo precisar de encontrar um AP que ofereça melhores condições de ligação para efetuar a associação. Para isso acontecer, existe uma função da camada MAC, denominada de *scan*. É assim criada uma lista de APs dando prioridade àqueles que possuem um maior valor de potência de sinal;

2. **Reautenticação** - Após o passo anterior, o dispositivo tenta-se autenticar a um AP, de acordo com a lista. O processo de reautenticação envolve a transferência das credencias presentes no AP anterior para o AP onde se vai registar. Isto é possível através do *IAPP (Inter Access Point Protocol)* ou através de protocolos proprietários.

Na Figura 2.2, os autores mostram os passos do procedimento de *handoff*.

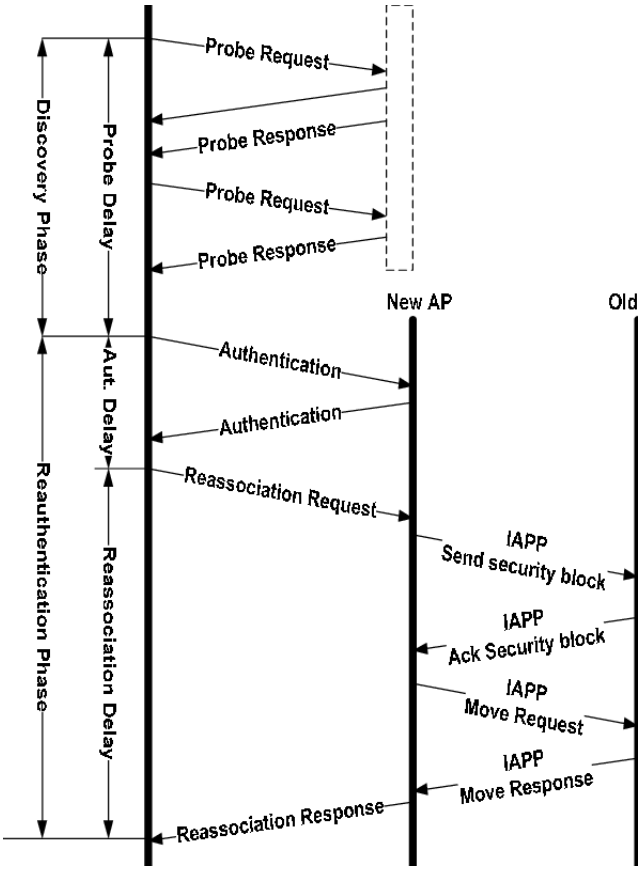

Figura 2.2 – Handoff no protocolo 802.11 [3]

Os autores definiram três atrasos em todo o processo:

1. Atraso de *Probe* - Este atraso deve-se às mensagem de *scan* trocadas com os diferentes APs;
2. Atraso de Autenticação - É o atraso provocado pela tentativa, por parte do dispositivo, em autenticar-se no novo AP. O número de mensagens trocadas com o AP depende do método de associação escolhido, o que pode ter influência no atraso geral de autenticação;
3. Atraso de Reassociação - É o atraso que decorre no processo de reassociação do dispositivo. Após haver uma autenticação efetuada com sucesso, são trocadas mensagens de pedido e resposta com o AP até para o processo de *handoff* fique completo.

A partir daqui os autores fizeram um grande número de experiências para tentar aferir o atraso de todo o processo de *handoff*. Foi usado um *sniffer* para discriminar os diferentes tipos de atrasos. A Figura 2.3 mostra um exemplo dos resultados obtidos com as experiências efetuadas.

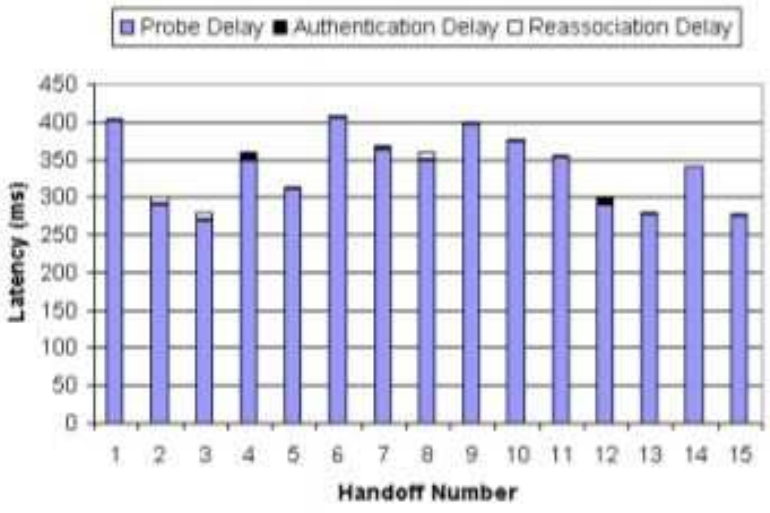

Figura 2.3 – Atrasos de *handoff* [3]

A partir da Figura 2.3, foi determinado o valor de atraso dos diferentes componentes do processo de *handoff* para cada troca efetuada. Os autores fizeram várias experiências onde usaram APs de diferentes fabricantes para tentar aferir a sua influência nos atrasos do processo. Após estas experiências, foram tiradas as seguintes conclusões:

1. O atraso de *probe* é o atraso predominante de todo o processo. Em todas as experiências efetuadas, o atraso de *probe* é responsável por, em média, 90% do atraso total do processo de *handoff*;
2. O tipo de *hardware* utilizado tem influência no atraso. Os autores deste trabalho fizeram testes com APs de diferentes marcas. Concluindo assim que pode haver variações até 50%, dependendo do tipo de *hardware*;
3. Há uma grande variação de atraso nas diferentes trocas efetuadas. Nos testes feitos, notou-se que, mesmo quando a experiência foi feita com as mesmas configurações, o tempo de atraso pode variar muito de um processo de *handoff* para outro.

A grande contribuição do estudo destes autores foi detetar os fatores que levam ao atraso no processo de *handoff* sendo que chegaram à conclusão que o fator que tem maior peso no atraso geral é o atraso de *probe*.

2.1.3 Trabalhos Relacionados

Neste capítulo vão ser abordados trabalhos e sistemas presentes na literatura, que se relacionam com o trabalho descrito nesta dissertação. Focam-se em metodologias de *handoff* em redes *Wi-Fi*, bem como sistemas de melhoramento deste mesmo procedimento, de modo a perceber as várias abordagens existentes nesta temática.

Como já foi referido anteriormente, as redes *Wi-Fi* têm tido um crescimento exponencial, tanto de utilização, como investigação com vista a melhorar o seu desempenho. Como foi visto no subcapítulo anterior, o processo de *handoff* tem uma

latência associada com relativa significância. Ao longo dos anos, foram feitos diversos estudos com o objetivo de otimizar o *handoff* em redes *Wi-Fi*. Este subcapítulo aborda alguns desses estudos.

SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks

Neste estudo, os autores [16] destacaram o problema de os métodos de monitorização disponíveis se basearem apenas na monitorização do AP atual, sendo que, só quando o sinal deste último diminui abaixo de um predefinido nível de *threshold*, é que é começado um *scan* ativo de outros APs para o dispositivo se conetar. O objetivo deste autores foi desenvolver um algoritmo, denominado de *SyncScan*, diminutivo de *Synchronized Scanning* que visasse monitorizar continuamente os APs circundantes.

O processo do algoritmo destes autores tira partido dos APs enviarem pacotes *beacon* a cada 100 ms e, a partir daí, foi criado um método de sincronização entre os clientes e a temporização desses pacotes em cada canal de transmissão. Basicamente, criaram um método onde os dispositivos conseguem monitorizar passivamente o meio mudando de canal de transmissão exatamente quando um pacote *beacon* está prestes a chegar. Assim, cada vez que um processo de *handoff* se aproxima, o atraso do mesmo é reduzido para a soma do atraso de autenticação com o atraso de reassociação. Na opinião destes autores, este método melhora também todo o processo de *handoff* na medida em que, visto que o dispositivo está sempre a monitorizar passivamente os sinais de APs circundantes, na altura de decisão, troca para o AP com melhor qualidade de sinal. Outro ponto positivo desta metodologia é que não influencia a estrutura dos algoritmos base definidos pelo padrão IEEE802.11, nomeadamente na vertente do *scan*, sendo garantida assim a compatibilidade com dispositivo IEEE802.11.

Para proceder à sincronização do sistema, os autores tiveram que ter em atenção à precisão dos relógios nos APs. Para isso, foi usado o NTP, pela sua simplicidade de implementação e pela sua disponibilidade na *Internet*. Contudo este tipo de sincronização trouxe a desvantagem de, caso haja APs no mesmo canal, podem tentar

gerar pacotes *beacon* ao mesmo tempo correndo o risco de haver interferência entre ambos. Para reparar este problema, os autores definiram uma janela de tempo de, por exemplo, 3ms, onde o envio desse será gerado aleatoriamente. Outro problema encontrado com esta implementação, foi o facto de apesar de remover o *overhead* de transição, acrescentar *overhead* regular. Isto significa que enquanto o dispositivo está à "escuta"de outros canais, não pode enviar informação para o próprio AP ou "escutar"o mesmo e, também pode perder pacotes que foram enviados durante a exploração de outros canais.

Finalmente, os autores deste algoritmo destacaram os seguintes benefícios na sua utilização:

- **Reduzir a latência**. Visto que 90 por cento do tempo de atraso que ocorre num processo de *handoff* ocorre durante o *scan* de outros APs e, sendo que esta implementação elimina a maior parte desse atraso, o algoritmo reduz a latência geral de 400ms, referidos noutros estudos, para apenas alguns milissegundos;
- **Melhores decisões de *handoff*.** Visto que o dispositivo está continuamente a fazer *scan*, na altura da decisão, consegue escolher inteligentemente o melhor AP;
- **Sistema de localização**. Em adição às vantagens já referidos, os autores referem que devido ao *scan* contínuo, o dispositivo estava sempre ciente das potências de sinal dos APs circundantes, providenciando assim um sistema de localização.

Improving the Latency of 802.11 Hand-offs using Neighbor Graphs

Nesta abordagem, dos autores [17] é destacado, mais uma vez, o tempo necessário na descoberta de novos AP para o dispositivo se ligar, sendo este classificado pelos autores como excessivo. É abordado também o problema do tamanho diminuto das células das redes WLAN *(Wireless Local Area Network)* causar um grande número de *handoffs* podendo levar a serviços não ótimos e até falha de serviço. O objetivo

deste estudo teve em vista diminuir o tempo de *probe*, ou seja, procura de novos APs, cortando ''escutas'' de canais desnecessárias que levaria a atrasos adicionais pois o dispositivo poderia estar à espera de resposta de AP não existentes.

Para a implementação de um método de otimização que tenha em conta os problemas já referidos, os autores propuseram dois algoritmos denominados de *NG* e *NG-pruning*. O primeiro algoritmo usa a estrutura de *neighbour graph*, ou grafo vizinho, sendo que o segundo melhora o processo de descoberta dos APs usando também, a estrutura de *non-overlap graph*, ou grafo não sobreposto. Um grafo vizinho é uma estrutura de dados que abstrai as relações de *handoff* entre dois APs. Se um dispositivo móvel consegue realizar *handoff* de um dado AP1 para outro AP2, então quer dizer que AP1 é vizinho de AP2.

Um grafo vizinho consegue saber os canais onde os APs vizinhos operam e também os APs vizinhos em cada canal [18]. O grafo não sobreposto é uma estrutura que abstrai relações não sobrepostas entre dois APs. É definido que dois APs não são sobrepostos quando um dispositivo móvel não consegue comunicar com ambos em boas condições.

Com estas estruturas de dados, os autores procuram reduzir o número de canais a procurar, reduzir o tempo de espera em cada canal e reduzir o número de APs a procurar.

Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs

Este estudo realizado por [4] também procura reduzir o tempo de atraso de *probe* visto que, segundo os autores, o tempo de atraso do *handoff* é muito elevado para permitir que serviços tipo *VoIP (Voice over Internet Protocol)* sejam feitos sem interrupções. Para este estudo os autores concentraram-se no padrão IEEE802.11b, mas referem o facto de todo o processo ser compatível com os padrões IEEE802.11a e IEEE802.11g. O padrão IEEE802.11b opera nos 2.4GHz na banda *ISM(industrial, scientific and medical)*, existindo 14 canais que vão desde os 2.402GHz até aos 2.483GHz, tendo cada canal 22MHz de largura. É referido que nos Estados Unidos

da América, local do estudo, são só usados os primeiros 11 canais e desses apenas os canais 1, 6 e 11 é que não se sobrepõem.

Como já foi referido, a grande parte do atraso do processo de *handoff* é o atraso de *probe*, constituindo 90 por cento de todo o atraso e, para reduzir este último, os autores focaram-se em melhorar o procedimento de *scan*, usando um algoritmo de *scan* seletivo, focando-se também em minimizar o número de vezes que esse processo de *scan* é necessário, usando para este último ponto, um mecanismo de *cache*.

O algoritmo de *scan* seletivo está representado na Figura 2.4.

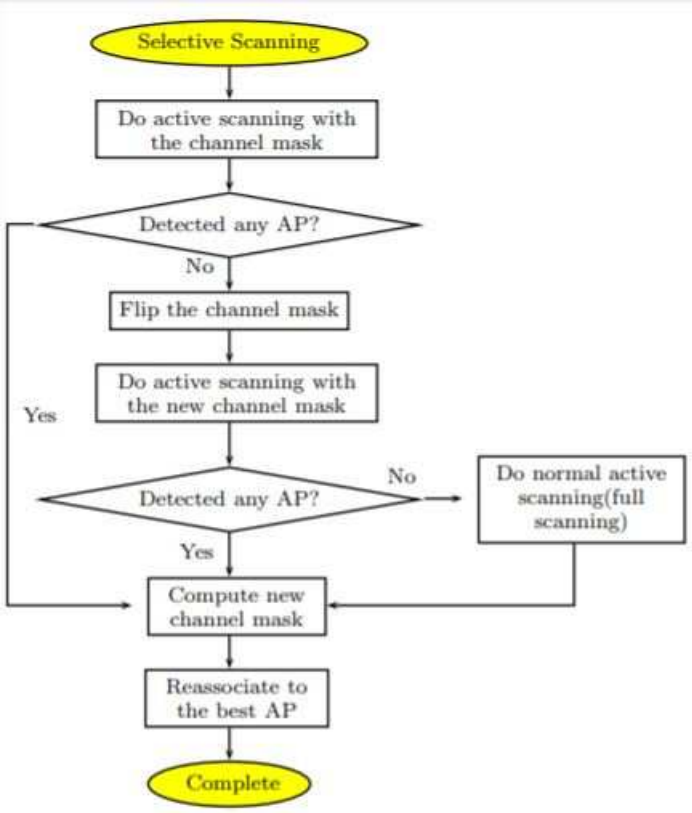

Figura 2.4 – Selective Scan [4]

Os passos deste algoritmo são os seguintes:

- Quando o *driver* é carregado, faz um *scan* completo, isto é, envia mensagens

de pedido de *probe* para todos os canais e escutas as respostas dos APs;

- A máscara de canal é definida ligando os *bits* de todos os canais onde a resposta de *Probe* foi ''escutada'' do resultado do passo anterior. Os *bits* dos canais 1,6 e 11 são também definidos visto que são estes canais onde a probabilidade da estação móvel se ligar, é maior;

- Seleciona o melhor AP, ou seja, aquele que tem melhor potência de sinal, e coneta-se a esse AP;

- O canal onde está o dispositivo é retirado da máscara. É definida uma fórmula para computar uma nova máscara, que é ''canais detetados (do passo 2) +1+6+11-canal atual'';

- Caso não sejam detetados APs com a máscara atual, esta é invertida e é feito um novo *scan*. Caso, mesmo assim, não seja detetado nenhum AP, é feito um novo *scan* completo a todos os canais.

Com vista a melhorar todo o sistema, foi implementado também um sistema de cache, neste caso uma cache do AP. Essa cache consiste numa tabela com uma lista de endereços MAC dos APs que estão adjacentes ao AP atual que usa como chave o endereço MAC deste último. A lista de endereços é gerada automaticamente aquando da mobilidade do dispositivo. O processo está representado na Figura 2.5.

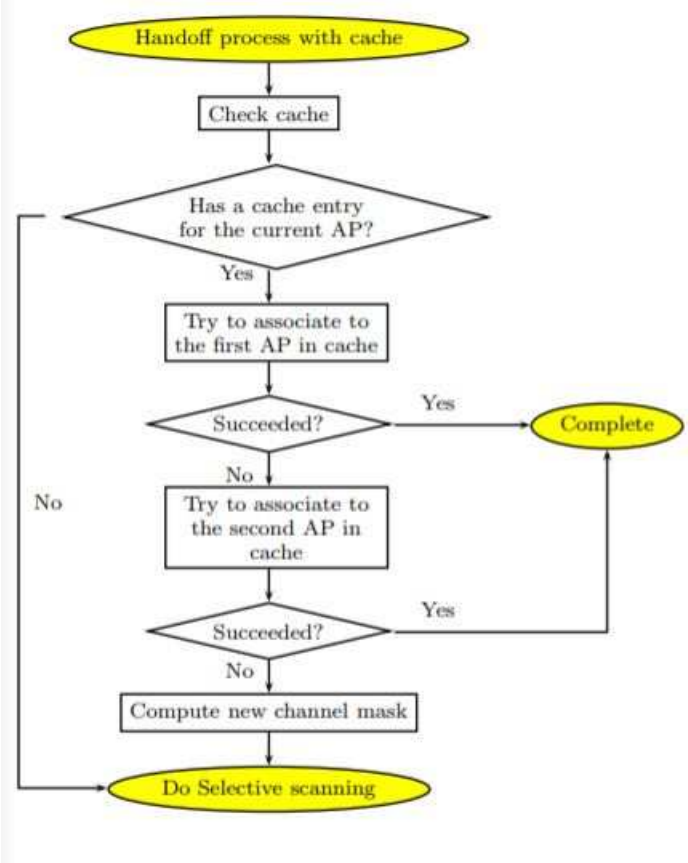

Figura 2.5 – Selective Cache [4]

Os passos do procedimento são os seguintes:

- Quando o dispositivo se associa a um AP, este é adicionado à cache como chave;
- Quando é necessário *handoff* são verificadas as entradas na cache correspondentes à chave atual.
- Se tal não for encontrado, o dispositivo faz um *scan* seletivo e os dois melhores APs encontrados (através da potência de sinal) são adicionado à cache com a chave do AP anterior;
- Caso seja encontrada entrada, o dispositivo é ordenado a associar-se ao novo AP e o processo de *handoff* está completo;

- Se o dispositivo falhar o registo no AP da primeira entrada, tenta associar-se ao da segunda entrada e caso haja uma nova falha, faz um *scan* seletivo.

Segundo os autores, o processo de *scan* seletivo reduziu a latência do processo de *handoff* para valores entre 30 e 60% dos valores de latência usando procedimentos padrão.

Predictive methods for improved vehicular Wifi access

No estudo representado a seguir, os autores [19] referem o problema de estabelecer ligação à rede em casos de grande mobilidade, neste caso, de veículos em circulação. Para reduzir os problemas de *handoff* para dispositivos dentro de veículos em elevada mobilidade, foi abordada a hipótese de tentar prever as melhores ligações a partir de informação guardada das ligações estabelecidas anteriormente.

Para isto, o método proposto funcionará quando o caminho percorrido pelo dispositivo dentro do veículo é feito com regularidade e, a partir da informação guardada das ligações feitas, é possível prever as melhores ligações a serem feitas. Esta estratégia tem como objetivo reduzir a latência das tentativas de ligação a um AP, a partir de *handoffs* predefinidos, bem como aumentar a velocidade de *download* através de transferência de dados de ligações feitas previamente.

Para a implementação deste método, um dispositivo dentro de um veículo acedia à rede através de APs que estariam montados ao longo da estrada. Os autores basearam-se nos estudos de [20] e [21] para provar que as pessoas usam regularmente as mesmas rotas, o que faz com que as suas viagens possam ser facilmente previsíveis, não só nas suas rotas, mas também nos locais de paragem e na velocidade de percurso. Assim é mais fácil aprender e guardar informação sobre os APs ao longo dos caminhos percorridos ajudando a estabelecer mais rapidamente uma ligação a esses mesmos APs quando o caminho é percorrido novamente.

Estas informações são marcadas com localizações de *GPS (Global Positioning System)*. Além de serem guardadas as informações dos APs, quando o dispositivo está

offline cria um *RF(Radio Frequency) fingerprint*, onde é guardada a potência de sinal dos APs circundantes e marcando-os com uma localização GPS. Assim, é feita uma estimativa do nível de conetividade dos APs ao longo do percurso e, é feito um pré-calculo onde o dispositivo necessita de *handoff* ao longo do caminho. Em adição aos conceitos já descritos, é realizada uma pré-busca do conteúdo que vai ser transferido. Com estas implementações, obtém-se um *handoff* e uma velocidade de *download* mais rápidos. Com esta implementação os autores referem que a performance de *download* foi melhorada num fator de dois.

2.2 Redes *Wi-Fi*

2.2.1 Contexto e História

A utilização de redes Wi-Fi está tão banalizada e mundialmente difundida que as novas gerações podem pensar que este tipo de tecnologia já existe há muito tempo, mas a maneira de como são utilizadas tem uma história muito recente.

O *Wi-Fi* é uma tecnologia que permite a comunicação entre diferentes dispositivos sem a necessidade de fios ou cabos para se estabelecer a ligação entre estes. Para se conseguir ter uma rede *Wi-Fi* que seja usada na distribuição de serviço de Internet é necessário um sistema de distribuição, como por exemplo, *router* e AP, que esteja ligado a um ISP (*Internet Service Provider*), sendo que, a zona onde se consegue aceder a essa rede *Wi-Fi* é denominada de *hotspot*.

A história do *Wi-Fi* remonta a 1971 quando a empresa *ALOHAnet*, pioneira no desenvolvimento de redes computacionais, desenvolveu um sistema sem fios de comunicação entre várias ilhas havaianas utilizando frequências UHF (*Ultra High Frequency*) [22]. Em 1985 surgiu o primeiro impulso para a criação comercial do *Wi-Fi*, quando a *FCC (Federal Communications Comission)*, organismo responsável pela regulação dos sistemas de telecomunicações nos Estados Unidos da América disponibilizou bandas no espetro de frequência, para o uso comunicacional, que

não precisavam de licença governamental. Foram disponibilizadas três bandas de frequência, chamadas de *"garbage bands"*, na ordem dos 900MHz, 2,4GHz e 5.8 GHz [23]. Estas faziam parte das bandas industriais, médicas e científicas, sendo que, estas frequências também eram utilizadas por outros equipamentos fora do contexto comunicacional, como por exemplo, os micro-ondas. Apesar destes avanços, só mais tarde é que foi criado um comité, denominado de 802.11, que permitiu desenvolver protocolos e padrões de utilização da tecnologia *Wi-Fi*. Este comité, composto por vários cientistas e engenheiros, foi inicialmente criado para permitir o uso deste tipo de tecnologia wireless em caixas registadoras e permitiam velocidades entre 1Mbit/s e 2Mbits/s [24].

Tendo em conta que estes acontecimentos remontam ao ano de 1997, pode-se dizer que o *Wi-Fi* tem um passado bastante recente e, foi a partir desta data que esta tecnologia teve um desenvolvimento exponencial, com a criação de novos padrões, permitindo o uso do *Wi-Fi* em vários tipos de equipamentos, com velocidades cada vez maiores. Apesar desta tecnologia ter começado a ser desenvolvida desde 1985, só no ano de 1999 é que a *Wi-Fi Alliance*, nome de uma organização sem fins lucrativos de várias grandes empresas, é que cunhou e registou o termo *Wi-Fi* que é conhecido nos dias de hoje, sendo que é também esta organização que é responsável pela interoperabilidade entre produtos. Desde essa altura até aos tempos que correm, o desenvolvimento e o lançamento de novos padrões tem acontecido quase anualmente, permitindo maiores velocidades de dados para serviços cada vez mais exigentes.

2.2.2 Padrão IEEE 802.11

Como dito anteriormente, o padrão IEEE802.11 é um conjunto de padrões que definem as características de comportamento das redes sem fios nomeadamente nas camadas MAC e PHY (camada física).

A camada MAC define dois tipos de mecanismo de acesso, DCF (*Distributed Coordination Function*) e PCF (*Point Coordinator Function*):

- DCF – Este tipo de mecanismo de acesso ao meio, de utilização obrigatória, baseia-se no protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) [25]. Este protocolo parte do princípio do *listen-before-talk*, ou seja, antes de transmitir a estação "escuta" o meio (*carrier sensing*) e, após esse tempo de escuta, denominado de DIFS, *DCF Inter-frame Space*, é inicializada a transmissão e as restantes estações ficam à escuta [26]. Após os dados terem sido enviados, o recetor confirma a chegada enviando um ACK (*Acknowledge*) depois de aguardar um curto espaço de tempo denominado de SIFS (*Short Inter-Frame Spaces*), sendo que este curto espaço de tempo impede que outras estações transmitam ao mesmo tempo, visto que as estações que recebem ACK têm prioridade sobre as outras. Caso, após o primeiro tempo de espera, DIFS, duas estações tentem transmitir ao mesmo tempo, há o risco de colisão e nenhuma das transmissões se realiza. Para resolver este problema, ao protocolo CSMA é adicionado o mecanismo de *Collision Avoidance*. Este método é escolhido ao invés do CD, *Collision Detection*, visto que as redes sem fios não permitem a sua implementação. O funcionamento deste, baseia-se no acrescento de um tempo de espera chamado fator de *Backoff*, que é escolhido aleatoriamente. Se se verificar que o meio não está livre, o valor de *Backoff* começa a decrescer até chegar a zero e volta a verificar a disponibilidade do meio até estar livre para transmitir [27];

- PCF – Este tipo de mecanismo é opcional neste tipo de redes e é usado por dispositivos e APs, pois são os que controlam o acesso ao meio. Visto que em PCF, são os APs e os dispositivos que têm prioridade, é definido um tempo de espera, denominado de PIFS (*PCF Inter Frame Space*), que é o tempo que os primeiros têm que esperar para escutar o meio e poder transmitir. Este tempo de espera é inferior ao DIFS [28];

- RTS (*Request to Send*) e CTS (*Clear to Send*) - Esta abordagem é um método opcional e adicional ao DCF [28]. Antes de ser transmitido um pacote, uma estação que tenha este mecanismo implementado, "reserva"o canal enviando uma frame RTS. O destino quando recebe essa frame envia outra denominada

de CTS e, a partir daí, o processo de transmissão de pacotes e respostas ACK ocorre normalmente. Este tipo de mecanismo ajuda a combater o problema do ''terminal escondido'' que ocorre quando duas estações móveis não se conseguem escutar.

Na vertente de modulação são usados os seguintes sistemas no padrão IEEE802.11:

Spread Spectrum - É uma técnica utilizada em telecomunicações onde um sinal é espalhado no domínio das frequências significando que ocupa uma maior largura de banda que aquela necessária para a transmissão de informação [29]. Este tipo de método tem como objetivo aumentar a segurança das comunicações aumentando a resistência ao ruído e interferências e, dificultando a sua deteção. Quando usado com utilizador único, não é um método muito eficiente no que toca à utilização da largura de banda, mas quando existem múltiplos utilizadores, estes conseguem alocar a mesma largura de banda do tipo *spread spectrum* sem existir interferência entre eles, por isso, estes tipos de sistemas são muito interessantes para o design de sistemas de comunicação sem fios. Devido à sua elevada segurança e dificuldade de terceiros em interferir deliberadamente, ou não, este tipo de técnica é muito utilizado em sistemas de comunicações militares. Existem dois tipos de técnicas *spread spectrum* que são utilizados no padrão IEEE802.11. Essas técnicas são:

- *DSSS (Direct Sequence Spread Spectrum)* - Com a técnica *Direct Sequence* a potência de transmissão do sinal é espalhado por toda a largura de banda do sistema e, assim, fica mais difícil para detetar o sinal [30];
- *FHSS (Frequency Hopping Spread Spectrum)* - Com esta técnica, a frequência da portadora é alterada e ''salta'' pelos canais através de um código pseudoaleatório apenas conhecido pelo transmissor e pelo recetor [31].

OFDM (Orthogonal Frequency Division Multiplexing) - Este método de codificação de sinal usa várias portadoras de frequências. O sinal é enviado para vários canais de banda estreita com frequências diferentes e, deste modo, reduz interferências e o fenómeno de *crosstalk*.

MIMO-OFDM (Multiple Input-Multiple Output OFDM) - Este método adiciona as vantagens do OFDM com o facto de a tecnologia MIMO usar múltiplas antenas no transmissor e no recetor. Assim conseguem ser codificados vários sinais ao mesmo tempo.

As diferentes versões do padrão 802.11 usam diferentes tipos de modulação e frequência bem como, têm diferentes larguras de banda e diferentes alcances. Na Tabela 2.1, estão representadas resumidamente essas características das diferentes versões. Existem muitas mais versões deste protocolo mas, ao longo dos tempos, foram desenvolvidas emendas onde eram englobados vários protocolos. Por exemplo na emenda 802.11 de 2007 foram englobadas 8 versões anteriores. De referir que a versão 802.11ax ou Wi-Fi 6 ainda não está disponível, tendo data estimada para lançamento ao público em dezembro de 2018.

Protocolo	Frequência (GHz)	Largura de Banda (MHz)	Modulação	Alcance(m) indoor| outdoor	Velocidade máxima (Mbps)
802.11 - 1997	2,4	22	DSSS, FHSS	20 | 100	2
802.11a	5 e 3,7	20	OFDM	35|120 para 5GHz e 5000 (out) para 3,7 GHz	54
802.11b	2,4	22	DSSS	38|140	11
802.11g	2,4	20	OFDM	38|140	54
802.11n Wi-Fi 4	2,4 e 5	20 e 40	MIMO - OFDM	70|40	72,2
802.11ac Wi-Fi 5	5 e de 0.054 até 0.79	20,40,80 e 160 e 6 até 8	MIMO - OFDM	35 indoor para 5GHz	866,7
802.11ad	60	2,160	MIMO portadora única	3,3	6912
802.11ah	0,9	1 até 16	MIMO-OFDM	até 1000	347
802.11aj	45 e 60	Sem info	Sem info	Sem info	Sem info
802.11ax Wi-Fi 6	2,4 e 5	Sem info	MIMO - OFDM	70|130	11000

Tabela 2.1 – Informações sobre o modo de funcionamento de diferentes versões do protocolo 802.11

A Frame IEEE802.11

Através do protocolo IEEE802.11 a comunicação entre os dispositivos e os APs é feita através da troca de *frames*, que são unidades de dados da camada de ligação de dados. Esta transmissão de informação permite saber as características dos intervenientes onde cada um tem a sua *frame* identificativa. O formato da *frame* do

padrão IEEE802.11 está presente na Figura 2.6:

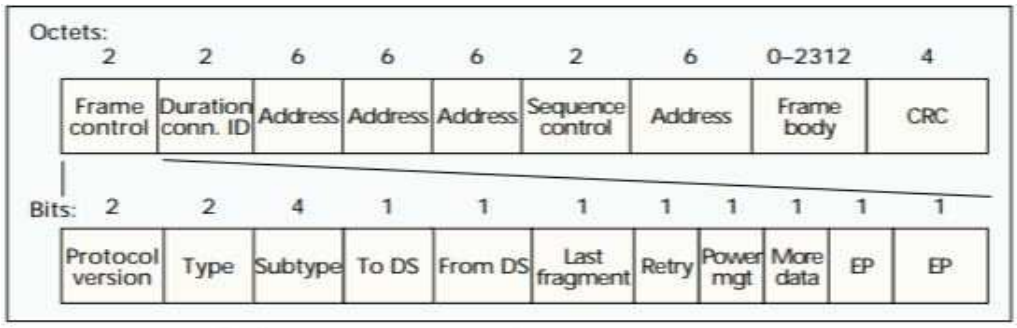

Figura 2.6 – Formato da Frame do padrão IEEE802.11 [5]

Os campos presentes na Figura 2.6 são os seguintes [32]:

- *Frame Control* - Este campo indica o tipo de *frame* que é e providencia informações de controlo. Essas informações são as seguintes:
  - *Protocol Version* - Identifica a versão do protocolo que está a ser usado;
  - *Type* e *subtype* - Identifica o tipo de *frame* que é (dados, gestão ou controlo);
  - *ToDS e FromDS* - Bits que indicam se a *frame* é destinada à *DS (Distribution System)* ou se vêm da DS;
  - *Last Fragment* - Informação sobre a fragmentação de *frames*;
  - *Retry* - Este bit é ativado no caso de haver uma retransmissão;
  - *Power Management* - Informação sobre a potência dos dispositivos;
  - *More Data* - Quando este bit é ativado, significa que o AP tem mais informação em espera para ser enviada;
  - EP - Informação sobre a encriptação do sistema.
- *Duration/connectionID* - Se for usado como campo de duração, indica o tempo (em ms) em que o canal vai ser alocado para a transmissão da *frame*. Pode ser usado como identificador da ligação;

- *Addresses* - Estes campos são, respetivamente em relação à Figura 3.1, o endereço do recetor (MAC do AP), endereço do transmissor (MAC do dispositivo de origem) e endereço de destino (MAC do endereço do dispositivo de destino);
- *Sequence Control* - Usado para fragmentar, reassociar e numerar *frames* enviadas entre o transmissor e o recetor;
- *Frame Body* - Contêm informações que são específicas de cada *frame*;
- *Cyclic Redundancy Check* - Controlo de redundâncias.

O envio de *frames* é essencial para as comunicações *wireless* nomeadamente em redes *Wi-Fi*. No contexto desta dissertação, é importante perceber como é que os APs e os dispositivos móveis comunicam e têm noção da existência uns dos outros. Esta comunicação é feita através do envio periódico de *beacons*, que são bocados de informação que são enviados pelos APs com intuito de anunciar a sua presença na área. Estes *beacons* são enviados ciclicamente de alguns em alguns segundos, dependendo do que foi definido pelo protocolo da rede. Em suma, o *beacon* tem como objetivo informar os dispositivos circundantes das suas características, e.g. BSSID *(Basic Service Set IDentifier)*, SSID e potência de sinal [33], de modo a que estes tenham noção das informações dos APs quando é feito um pedido de *scan*.

2.2.3 Padrão IEEE 802.1X

Estes padrões são referentes à segurança de autenticação. Foram introduzidos para melhorar a segurança providenciada pelos padrões 802.11 na vertente de autenticação, controlo de acesso e gestão de chaves [34], e são baseados no protocolo de autenticação *EAP (Extensible Authentication protocol)*. O EAP é uma framework de autenticação que tem três intervenientes no processo [35]:

- Supplicant/requerente – Utilizador que se deseja autenticar;
- Autenticador – No caso de redes sem fios *Wi-Fi*, são os APs;

- Servidor de autenticação – Centro de dados que valida o requerente. Apesar de não ser obrigatório, o servidor de autenticação normalmente utilizado é o RADIUS *(*Remote Authentication Dial-In User Service).

O funcionamento do processo de autenticação é o seguinte:

1. O requerente envia um pedido de autenticação;
2. O autenticador recebe o pedido e envia outro a pedir as credencias ao requerente, bloqueando o resto do tráfego;
3. O requerente envia os dados para o servidor de autenticação;
4. O servidor de autenticação processa os dados e envia uma resposta ao autenticador;
5. Se o pedido for aceite, o autenticador transita o requerente, i.e, passa o requerente para o estado de autenticado.

O requerente geralmente é um *software* num dispositivo, *PC (personal computer)*, *smartphone* ou computador portátil, o autenticador é um *switch Ethernet* ou APs e o servidor de autenticação é, na maior parte das vezes, servidor RADIUS [36].

2.2.4 Eduroam

Visão Geral

A *Eduroam (EDUcation ROAMing)* é uma rede direcionada para a vertente educativa, disponibilizada em 70 países, sendo utilizada principalmente em universidades, laboratórios, centros de pesquisa, entre outros.

A pioneira no desenvolvimento da *Eduroam* foi a GÉANT, uma rede de dados para a comunidade de educação e investigação, em 2003. Estruturada com os padrões IEEE 802.1X e com hierarquia baseada no RADIUS providencia acesso à *Internet*

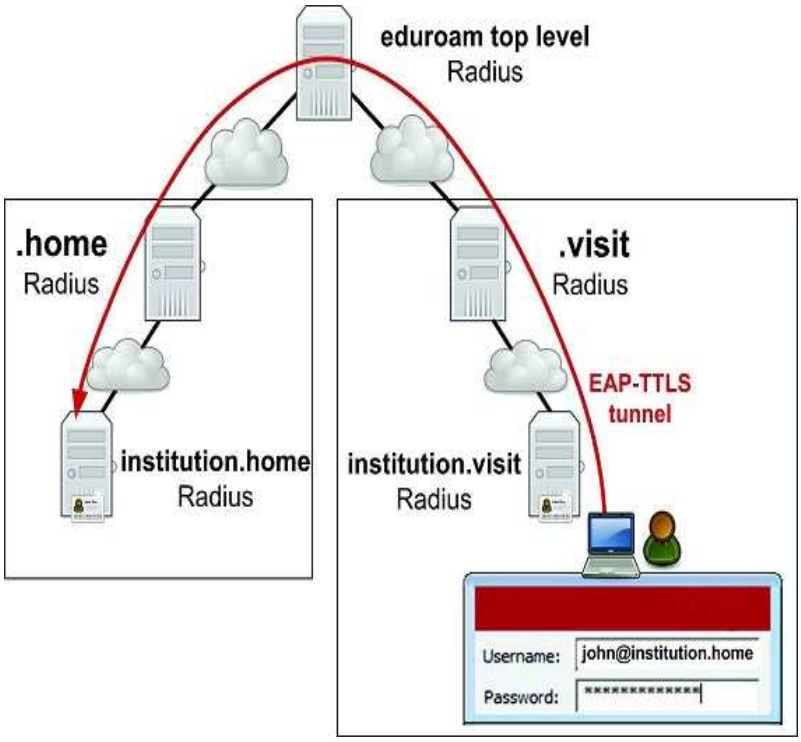

Figura 2.7 – Funcionamento do sistema Radius. [6]

ininterruptamente a utilizadores credenciados dentro de infraestruturas com suporte para tal [37]. O RADIUS é um protocolo/servidor que permite a autenticação remota de utilizadores de serviços que usem servidores *proxy* RADIUS e tem como objetivo alcançar maiores níveis de segurança, permitindo que os perfis de utilizador sejam guardados numa base de dados central e partilhados com todos os seus servidores. O modo de funcionamento do sistema RADIUS com a *Eduroam* está demonstrado na Figura 2.7.

Portugal foi um dos países pioneiros na fase de testes da *Eduroam*, a par da Holanda, Finlândia, Croácia e Reino Unido e após esses testes, as redes e organizações educativas de quase todos os países europeus começaram a juntar recursos para a instalação da *Eduroam*.

Segurança da Eduroam

Como já foi referido no subcapítulo anterior, a segurança de toda a estrutura na *Eduroam*, i.e., autenticação e encriptação, é assegurada pelos padrões 802.1X e 802.11, respetivamente, em conjunto com a hierarquia RADIUS. Como explicado anteriormente, o padrão 802.1X é baseado no protocolo EAP que, por sua vez, tem associado um grande número de módulos e protocolos de segurança e autenticação. A *Eduroam* suporta vários módulos do padrão 802.1X, mas a compatibilidade depende de cada dispositivo/sistema operativo. Os módulos suportados pela *Eduroam* são:

- TLS *(Transport Layer Security)* - É uma extensão EAP que suporta múltiplos métodos de autenticação que providencia autenticação mútua, isto é, cliente e servidor, com base em certificados e infraestrutura chave pública e chave privada [38];
- TTLS *Tunneled transport Layer Security*-PAP (*Password authentication Protocol)* – O TTLS-PAP é uma extensão EAP para a autenticação de cliente e servidor que tem como principal vantagem e diferença do EAP-TLS, o facto de ter um túnel que permite uma maior segurança na transição de dados. Este garante proteção contra ataques dos tipo *man in the middle*, que é um tipo de ataque onde os dados que são passados entre dois intervenientes são intercetados por uma terceira entidade que se pode fazer passar por qualquer um dos outros intervenientes e até adulterar os dados, sem que estes consigam ter perceção do ataque. Além disso, difere do EAP-TLS na vertente em que não necessita obrigatoriamente de certificados do lado do cliente para decorrer o seu funcionamento. A este protocolo é associado um outro, denominado de PAP, que é um simples protocolo de autenticação através de password, sendo que esta não é encriptada, ou seja, é enviada para o servidor de autenticação no formato de texto simples, tornando este protocolo vulnerável para comunicações point-to-point [39];
- TTLS-MSCHAPv2 – O MSCHAP (*Microsoft Challenge Handshake Authentication Protocol*) é um protocolo proprietário da *Microsoft* que é semelhante

ao *CHAP (Challenge Handshake Authentication Protocol)* na vertente em que também encripta informação antes de a transmitir através de uma ligação *PPP (Point-to-Point Protocol)* [40]. É utilizado na Eduroam como autenticação de segunda fase sendo que a primeira está a cargo do protocolo TTLS;

- PEAP *Protected EAP* – O PEAP é estruturalmente semelhante ao EAP-TTLS, na medida em que usa também um túnel para estabelecer comunicação entre cliente e servidor, para depois proceder à autenticação das credenciais. As diferenças entre estes dois métodos baseiam-se em motivos proprietários e de compatibilidade de dispositivos [41];

- *PWD (Password)* - Esta extensão do protocolo EAP que usa uma password partilhada para autenticação [35]. Esta password pode ser retirada de uma lista de passwords, tipo dicionário.

- *FAST (Flexible Authentication via Secure Tunneling)* - É uma extensão do protocolo EAP que permite a autenticação tanto do cliente como do servidor [42]. Funciona em duas etapas sendo que, na primeira é estabelecido um túnel TLS usando uma chave pré-partilhada, chamada de *PAC (Protected Authentication Credential)*. A seguir, dados codificados são usados para proceder à autenticação do utilizador.

2.3 Android

2.3.1 Visão Geral do *Android*

O *Android* é um sistema operativo e pilha de software *open-source*. Nos tempos que correm, é mundialmente conhecido por ser o sistema operativo mais utilizado em dispositivos móveis, mais propriamente *smartphones*. Estatísticas demonstram que 86.2% dos dispositivos usam sistema operativo *Android* contra apenas 12.9% do Sistema *iOS*, que vem em segundo lugar na lista dos sistemas operativos mais utilizados em dispositivos móveis. Esta estatística está demonstrada na Figura 2.8.

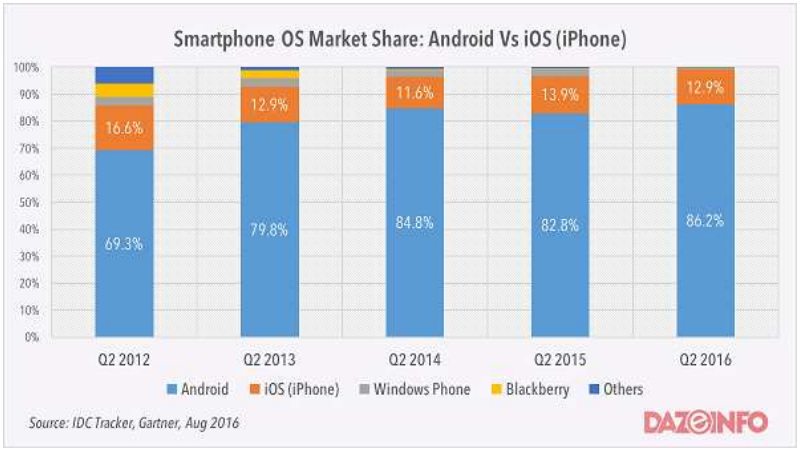

Figura 2.8 – Utilização do sistema operativo Android e iOS. [7]

O nome *Android* vem da empresa originalmente fundadora, Android Inc., que tinha como objetivo desenvolver dispositivos móveis "cientes da localização e das preferências do utilizador". Apesar desta definição ir ao encontro da atual descrição de um *smartphone*, foi cunhada na altura da fundação da empresa, datada de 2003, muito antes do termo *smartphone* ser conhecido pelo público em geral. Foi também revelado, anos mais tarde, por um dos fundadores, que a empresa não tinha como objetivo criar um sistema operativo que servisse como núcleo de funcionamento em *smartphones*, mas sim em câmaras digitais. Em 2005, a Google tomou a decisão de comprar a Android Inc., mantendo os fundadores originais, com o intuito de criar o sistema operativo Android. A partir dessa altura, tem sido desenvolvido continuamente, sendo que, em 2008 foi lançado o primeiro *smartphone* com o sistema operativo Android, T-Mobile G1, equipado com a versão 1.0. Após esta primeira versão, novas versões têm sido desenvolvidas e lançadas, onde a versão mais recente até à data de outubro de 2018, é a versão 9.0 [43].

Para o desenvolvimento de aplicações Android, a plataforma oficial é o Android Studio. Esta plataforma foi anunciada pela *Google* em meados de 2013 sendo oficializada e lançada ao público gratuitamente nos finais do ano de 2014 e veio com isto, substituir a plataforma *Eclipse* que era até então a plataforma oficial para o desenvolvimento de aplicações *Android*. É um *Integrated Development Environment*

(IDE) baseado num outro denominado de Intellij IDEA que foi desenhado maioritariamente para desenvolvimento de programas na linguagem de programação *Java*.

2.3.2 Ferramentas do Android

A plataforma *Android* disponibiliza ao programador um grande número de *Application Programming Interfaces (API)* que permitem o desenvolvimento de aplicações e serviços e o acesso aos recursos do *hardware* onde estes se encontram a ser executados. No contexto desta dissertação as APIs utilizadas foram a android.net.wifi e android.content, sendo que a primeira permite gerir as funcionalidades do *Wi-Fi* do dispositivo, e a segunda providencia categorias de partilha de conteúdo, gestão de pacotes e recursos que permitem o acesso e gestão dos dados que serão apresentados no dispositivo [44].

android.net.wifi

Quando é necessário ter acesso a informações sobre a rede *Wi-Fi*, esta é utilizada a API android.net.wifi. Esta API providencia um meio através do qual as aplicações conseguem comunicar com as zonas de baixo-nível da pilha *wireless* [45]. Com esta API é possível obter informações no que toca ao estado da ligação *Wi-Fi*, endereço IP *Internet Protocol*, velocidade de ligação, e.g, bem como, permite efetuar algumas operações em relação à rede tais como, fazer o scan das redes em volta do dispositivo, adicionar e remover rede e conectar-se e desconectar-se da mesma. Para se conseguir acesso às funcionalidades referidas, é necessários declarar as suas respetivas permissões, no ficheiro *manifest.xml* da respetiva aplicação. Essas permissões a ser declaradas são as seguintes [45]:

- `ACCESS_WIFI_STATE`;
- `CHANGE_ WIFI_STATE`;

• `CHANGE_WIFI_MULTICAST_STATE`;

O android.net.wifi providencia um número elevado de classes, no entanto, no contexto desta dissertação, só serão abordadas aquelas que foram utilizadas. Essas classes são as seguintes:

• *ScanResult* - esta classe descreve a informação de um AP que foi detetado. Dentro dessa informação estão incluídos o SSID, que é a identificação da rede do AP, e o *BSSID* , que é o endereço do AP;

• *WifiConfiguration* - classe que representa a configuração de uma rede *Wi-Fi*. Também se consegue ter acesso à configuração da segurança da rede;

• *WififEnterpriseConfig* - tal como a classe imediatamente em cima descrita, representa a configuração de uma rede *Wi-Fi*, só que na vertente *enterprise* sendo que, guarda informações e credencias do protocolo de autenticação EAP;

• *WifiInfo* - classe que descreve a informação da coneção *Wi-Fi* que está ativa no momento. Os seus métodos públicos permitem requerer informações tais como, BSSID, SSID, RSSI *Received Signal Strength Indicator*, frequência, endereço IP, identificação da rede, entre outros;

• *WifiManager* - esta classe providencia as ferramentas necessárias para a gestão de todos os aspetos relacionados com a conetividade de *Wi-Fi*.

AsyncTask

No desenvolvimento de aplicações *Android* por vezes a *MainThread*, onde o código é executado, pode ficar sobrecarregado com as tarefas definidas no mesmo. Caso isso aconteça, é aconselhável a implementação de uma *Asynchronous Task*, ou tarefa

assíncrona, que permite executar uma tarefa em *background* retirando trabalho ao *MainThread*. A classe *AsyncTask* permite um uso eficaz da *MainThread*, permitindo a execução de tarefas em *background* encapsulando a manipulação de *threads* e *handlers*, que são outros métodos de gestão de tarefas. A tarefa assíncrona usa três tipos genéricos [46]:

- Params - são os parâmetros enviados para a tarefa quando é executada;
- Progress - as unidades do progresso que poderão ser publicadas durante a computação em *background*;
- Result - que é o tipo dos resultados da computação em *background*.

De referir que nem todos os tipos mencionados são obrigatoriamente usados na tarefa. Podem ser declarados como *void*.

Quando é executada, a tarefa tem quatro passos:

- *onPreExecute()* - Passo usado para inicializar a tarefa;
- *doInBackGround(Params...)* - Invocado na *background thread* e é usado para fazer uma tarefa em *background*. É aqui que são passados os parâmetros para a execução da tarefa;
- *onProgressUpdate(Progress...)* - Método usando para mostrar ao utilizador alguma forma de progresso da tarefa, visto que o tempo de execução da mesma pode ser desconhecido;
- *onPostExecute(Result)* - Neste método, são passados os resultados da execução da tarefa em *background*.

2.4 Network Time Protocol

Em qualquer tipo de rede, privada ou pública, nas transações efetuadas na Internet, o tempo e a sincronização são características essenciais para que seja garantida a

precisão e a fiabilidade dessas mesmas transações. Os servidores de NTP oferecem servidores públicos facilmente encontrados na Internet e são usados globalmente para garantir a sincronização dos relógios dos computadores e coordenar a distribuição do tempo [47]. Este capítulo é referente à versão 4 do protocolo NTP, versão mais recente publicamente disponível, onde é explicado o seu funcionamento. A utilização do protocolo NTP no contexto desta dissertação, prende-se no facto de se conseguir obter os valores dos *timestamps* de pedidos feitos ao servidor NTP. Com estes valores é possível calcular o atraso da rede onde, a partir daí, tomar a decisão de escolher o AP que providencie a rede com menores valores de atraso de rede.

A rede NTP funciona de uma maneira hierárquica, sendo que cada nível da hierarquia é chamada de estrato (stratum). O estrato-0 é um relógio referência proveniente do padrão *UTC (Coordinated Universal Time)*, que tem muito pouco ou nenhum atraso. Este estrato não entra diretamente na rede, estando ligado a servidores primários que são chamados de dispositivos do estrato-1. Estes servidores recebem do estrato, via satélite ou rádio, o sinal com a informação de tempo. A sua função é propagar este sinal para estratos inferiores, fazendo com que o comportamento se repita de estrato para estrato. O NTP suporta até 16 estratos, onde o último indica inoperabilidade, sendo que a precisão da rede decresce de estrato para estrato. Contudo, clientes podem pedir informações de tempo diretamente a servidores do estrato-1 ou até receber essas informações de vários servidores, visto que servidores do estrato-2 e inferiores têm a capacidade de se sincronizarem entre eles, o que torna o sistema mais robusto. Esta topologia da divisão do sistema em estratos torna o sistema mais eficiente, pois existe um balanceamento de carga com vista a não sobrecarregar servidores do estrato-1. Na figura 2.9 estão apresentados os diferentes estratos do protocolo NTP.

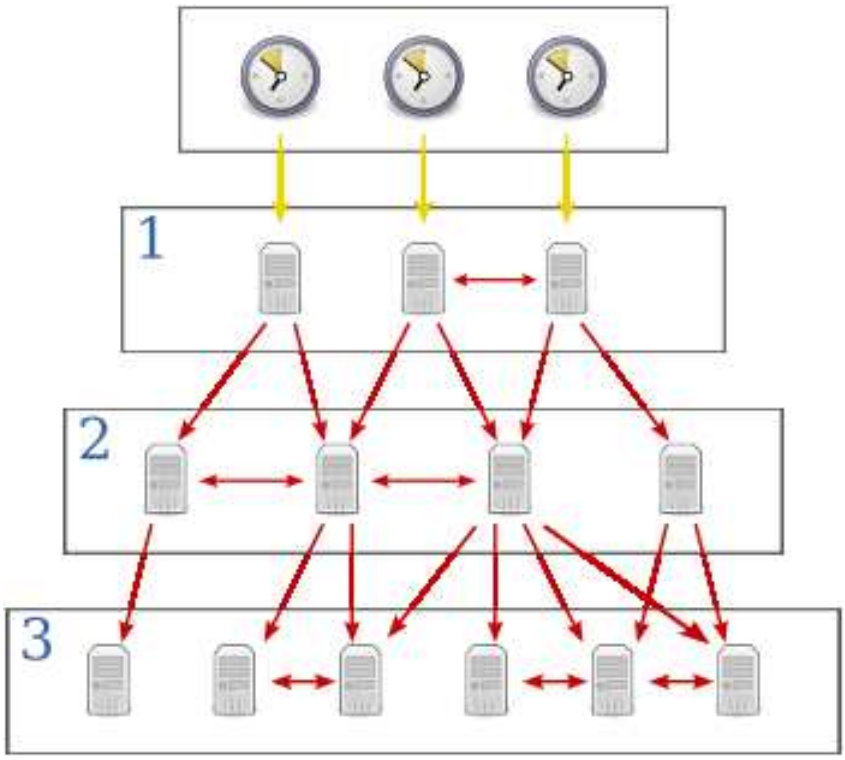

Figura 2.9 – Estratos do NTP. [8]

Como se pode ver na Figura 2.9, no estrato-0 são as referências externas que, ligadas diretamente, enviam os valores de sincronização aos servidores do estrato-1 onde a partir daí, procedem à partilha da informação de tempo com outros do mesmo estrato e de estratos inferiores, fazendo com que o comportamento se replique ao longo da rede.

Segundo [9], a arquitetura do protocolo NTP consiste em servidores remotos, processos *peer/poll*, algoritmos do sistema, processo de disciplina do relógio e processo de ajuste do relógio. Presente na Figura 2.10, esta arquitetura incluí dois processos dedicados a cada servidor, *peer*, para receber mensagens do servidor ou do relógio de referência e um outro, *pool*, para transmitir mensagens para o servidor ou relógio de referência.

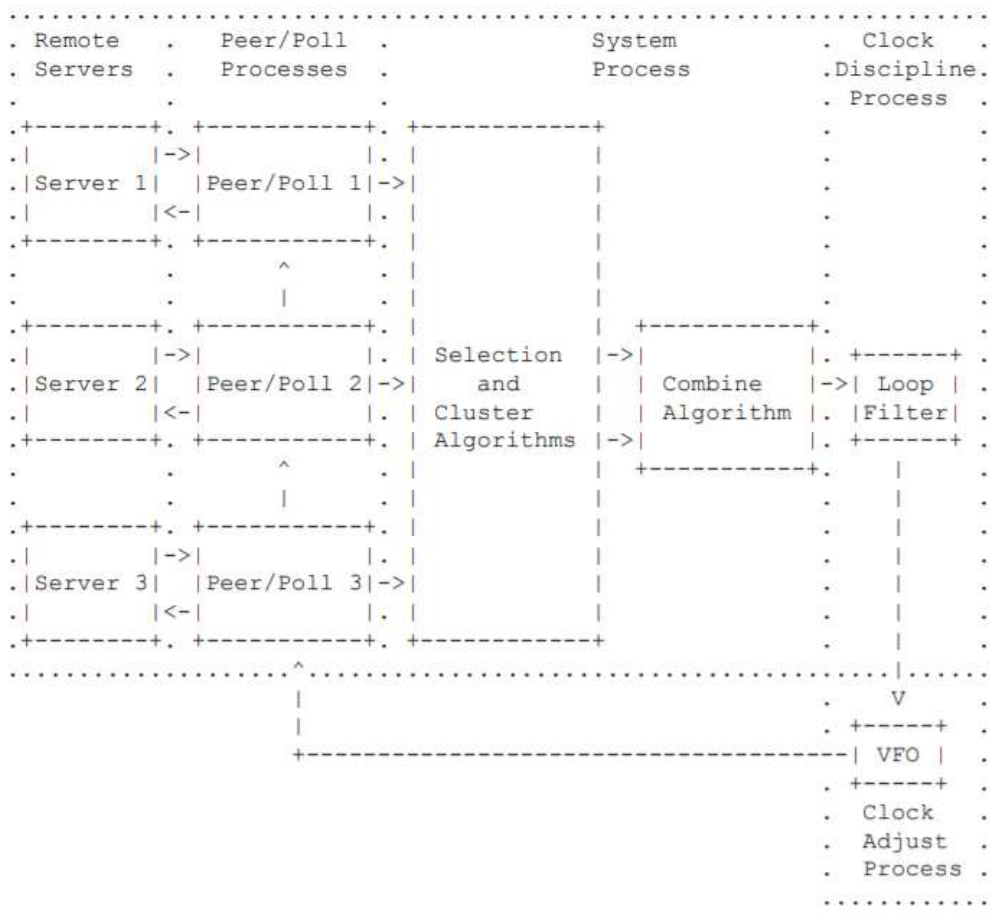

Figura 2.10 – Modo de implementação NTPv4. [9]

O modo de funcionamento desta implementação é o seguinte:

1. O cliente envia pedidos a um ou mais servidores de modo a obter coerência nos resultados. De seguida processa os dados da resposta dos servidores;

2. O servidor troca endereços de destino e origem e reescreve certos campos no pacote e retorna imediatamente ou passado algum tempo, dependendo se é no modo cliente/servidor ou simétrico, respetivamente;

3. Nos algoritmos do sistema há uma negociação entre os vários servidores e relógios de referência de modo a escolher os candidatos mais fiáveis para proceder

à sincronização;

4. O processo de disciplina do relógio representado pelo VFO (*Variable Frequency Oscillator*) tem como função controlar o tempo e a frequência do sistema. Os *timestamps* que atingem o VFO fecham o *loop* de *feedback* que mantém o tempo de relógio do sistema;

5. Associado ao processo de disciplina do relógio está o processo de ajuste do relógio que tem como função manter a frequência constante.

Na Figura 2.11 está representada a metodologia de sincronização de relógio.

Considere-se um servidor NTP e um cliente NTP. As trocas de dados entre ambos são marcadas com as informações de tempo (*timestamps*), onde a partir desses valores é possível calcular o *offset* e o tempo de ida e volta de um pacote, bem como estimar possíveis erros de uma forma fiável. Os valores *timestamp* estão representados por T1,T2,T3 e T4 e o tempo de ida e volta representado por d e o *offset* representado por o.

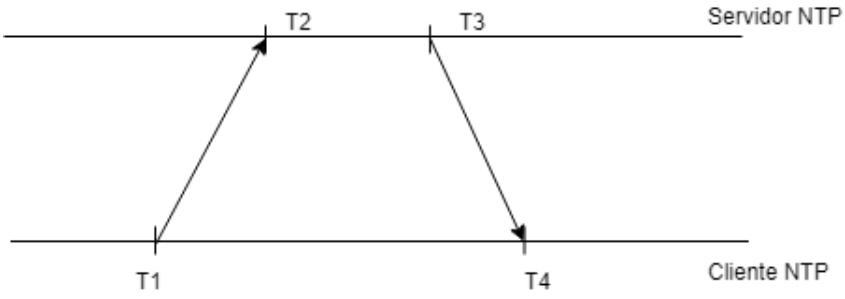

Figura 2.11 – Troca de *timestamps* entre cliente e servidor.

Este processo é utilizado para a sincronização do relógio onde o cliente faz, regularmente, *polls* a um ou mais servidores NTP de modo a manter-se sincronizado da maneira mais fiável possível. Com estes *timestamps*, é possível calcular os já referidos atrasos de ida e volta bem como o *offset*. O procedimento mostrado na Figura 2.11 caracteriza-se da seguinte forma:

- T1 - *Timestamp* do cliente no pacote do pedido de transmissão;
- T2 - *Timestamp* do servidor quando o pacote anterior chega ao destino;
- T3 - *Timestamp* do servidor no pacote de resposta;
- T4 - *Timestamp* do cliente quando o pacote anterior chega ao destino.

Com estes valores, o protocolo NTP permite calcular o atraso de ida e volta, representado por d, e o *offset* do relógio, representado por o, usando as seguintes equações:

Atraso de ida e volta:

$$d = (T4 - T1) - (T3 - T2) \tag{2.1}$$

Offset

$$o = \frac{(T2 - T1) + (T3 - T4)}{2} \tag{2.2}$$

3 Conceção

Para se conseguir definir um método de escolha para o melhor AP da rede, foi efetuado um estudo exaustivo do tipo de rede na qual pretendemos ligar o dispositivo móvel, bem como as variáveis que podem afetar a performance dessa mesma rede.

A rede utilizada foi a Eduroam que, como explicado no Capítulo 2, é uma rede usada em sessenta e uma instituições em Portugal, estando disponível em 70 países [37].

A Universidade de Trás-os-Montes e Alto Douro, mais precisamente, o Edifício da Escola de Ciências e Tecnologias Pólo 1, foi o local escolhido para o desenvolvimento desta dissertação, executando aqui todos os testes necessários. A grande quantidade de APs ao longo de todo o edifício possibilitou o acesso a um grande número de dados que se refletiu positivamente na performance da aplicação. Isto deve-se à heterogeneidade da utilização da rede Eduroam ao longo de todo o edifício.

Para o desenvolvimento deste projeto foi utilizado um *smartphone* provido com sistema operativo Android de versão 5.1.1. Foram também efetuados testes com diferentes dispositivos móveis equipados com diferentes versões do sistema operativo *Android*, de modo a testar a compatibilidade da aplicação em diferentes situações.

No presente capítulo é abordado o estudo que serviu como base para o desenvolvimento da aplicação *Android*. Neste estudo foi provado que usando o tempo de atraso de rede como parâmetro principal para a escolha do melhor AP, há uma maior probabilidade de obter um melhor serviço de *Internet*. É também referido o princípio de funcionamento da aplicação *Android*.

3.1 Funcionamento da experiência

Como dito anteriormente, o objetivo desta dissertação é desenvolver uma aplicação Android que otimize o método de *handoff*. Esta aplicação permite que o dispositivo móvel se conecte a APs da rede *Eduroam*, providenciando o melhor serviço, onde a escolha do AP se baseia no menor valor de atraso da rede.
Num período anterior ao desenvolvimento da aplicação propriamente dita, foi necessário estudar e avaliar a possibilidade de desenvolver a aplicação a partir da metodologia já referida. Isto quer dizer que foi desenvolvida um sistema, no qual foram feitos diversos testes, sendo que a partir dos seus resultados, se conseguiu concluir que os métodos que usam o valor de RSS como parâmetro de escolha do AP nem sempre oferecem o melhor serviço de *Internet*.

A imagem da Figura 3.1, ilustra a arquitetura do sistema que contém os seguintes componentes:

- Um cliente Android, i.e, um dispositivo móvel equipado com um sistema operativo Android e com uma aplicação pela qual se adquiria os dados desejados;
- Rede de APs;
- Servidor NTP, para realizar pedidos de tempo;
- Servidor iperf3, para fazer medições da velocidade da rede;
- Cliente iperf3, para simular condições reais de tráfego na rede.

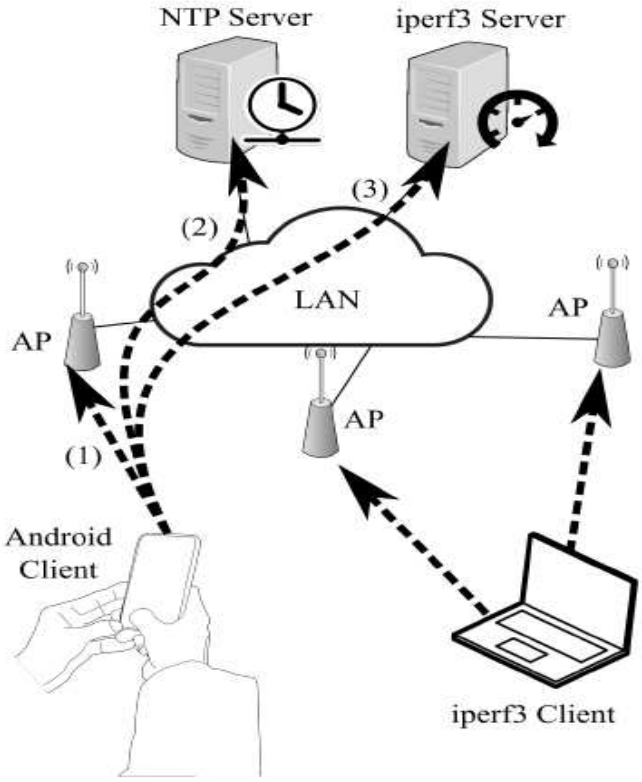

Figura 3.1 – Arquitetura da experiência. [1]

Para as experiências foram utilizados no máximo quatro APs, iguais aos normalmente utilizados na Universidade de Trás-os-Montes e Alto Douro e com estes, foi criada uma rede virtual a que estaria ligado um dispositivo móvel equipado com uma aplicação *Android* desenhada para retirar os dados desejados. A partir daí, retiraram-se esses dados, sendo que consistiam nos valores de RSSI (em dBm), atraso de rede (em ms) e velocidade de download (em Mbps). Para obter os dados de atraso de rede o dispositivo fazia pedidos ao servidor NTP. Para simular testes reais, também conectado a esta rede estaria um servidor iperf3, software que serviu para fazer medições de velocidade na rede. Este software foi utilizado como cliente, introduzindo tráfego na rede de modo a conseguir-se obter simulações aproximadas a situações reais. O software iperf3 foi instalado num computador portátil onde, a partir deste, estavam em funcionamento tanto o cliente como o servidor iperf3.

Para todos os diferentes cenários, que vão ser referidos mais à frente, os passos da aplicação eram os seguintes:

1. Obter a lista de todos os APs da rede virtual;
2. Para cada AP na lista:
   (a) Associar ao AP;
   (b) Conetar-se à rede;
   (c) Medir o atraso da rede.
3. Associar-se ao AP com o menor atraso de rede;
4. Conetar-se à rede.

Para a medição do atraso de tempo foi utilizado o servidor NTP. Como explicado no Capítulo 3, subsecção *Network Time Protocol*, o procedimento deste protocolo baseia-se em quatro tempos T1,T2,T3 e T4. A partir daqui consegue-se calcular o atraso de rede que tem a seguinte fórmula:

$$d = T4 - T1 - T3 + T2 \tag{3.1}$$

Com esta fórmula, é possível calcular o atraso de rede com exatidão mesmo que os relógios do servidor e do cliente não estejam sincronizados. Uma das vantagens de usar este protocolo em vez do protocolo *ICMP (Internet Control Message Protocol)*, com mensagens de *Echo Request* e *Echo Reply*, usado na aplicação *ping*, é que com o NTP conseguimos calcular o tempo de ida e volta da mensagem excluindo qualquer atraso que aconteça no sistema remoto [1]. Outra das vantagens prende-se no facto de os seus servidores serem públicos e de acesso livre, aliado à vasta utilização dos seus serviços em quase todos os sistemas de rede, o que leva a que o tráfego não seja bloqueado pelas *Firewall*.

Em relação às simulações propriamente ditas para a aquisição de dados foram montados três cenários:

1. Duas salas de aula com um corredor, cobertos com uma rede com quatro APs;
2. Duas salas de aulas com um corredor, cobertos com uma rede com três APs;
3. Uma sala de aula com um corredor, cobertos com uma rede com três APs.

Para estas simulações, os APs utilizados eram todos iguais e usam o padrão IEEE802.11g, da mesma marca e com a mesma versão de *firmware*. Foi usado um *smartphone Android* equipado com uma aplicação, de modo a listar os APs disponíveis, estabelecer a ligação entre todos, medir a velocidade de *download* e o atraso da rede, guardando assim todos estes dados num ficheiro .csv para análise futura. Na Tabela 3.1 é apresentada uma amostra dos dados recolhidos na experiência. Estes dados corroboram que escolhendo o AP que tem um maior valor RSSI, nem sempre providencia o melhor serviço, podendo não ser a melhor opção.

AP	RSSI(dBm)	Delay(ms)	Speed(Mbps)
AP1	-49	6,30	13,13
AP2	-49	7,50	5,41
AP3	-71	3,60	14,79
AP4	-62	4,00	9,01

Tabela 3.1 – Tabela com exemplos de dados recolhidos. [1]

Como se pode ver no exemplo da Tabela 3.1, não é o AP com um maior valor de potência de sinal, neste caso o AP1 e AP2, que dão uma velocidade de *download* mais elevada. O AP3 é o dispositivo que providencia uma velocidade de *download* maior, 14.79Mbps, no entanto é o que possui uma potência de sinal inferior, dos quatro exemplos recolhidos, -71 dBm. Este AP é também o que apresenta o atraso de rede mais baixo (3,60 ms). Consegue-se extrapolar a partir destes dados que a escolha de AP para proceder à ligação baseado na potência de sinal, pode levar a uma velocidade de *download* que não é a melhor.

Os resultados obtidos dos cenários acima descritos permitiram retirar algumas conclusões em relação à velocidade de rede em função tanto do valor de RSSI como do valor de atraso de rede. Os gráficos em baixo mostram resultados obtidos num dos cenários de teste.

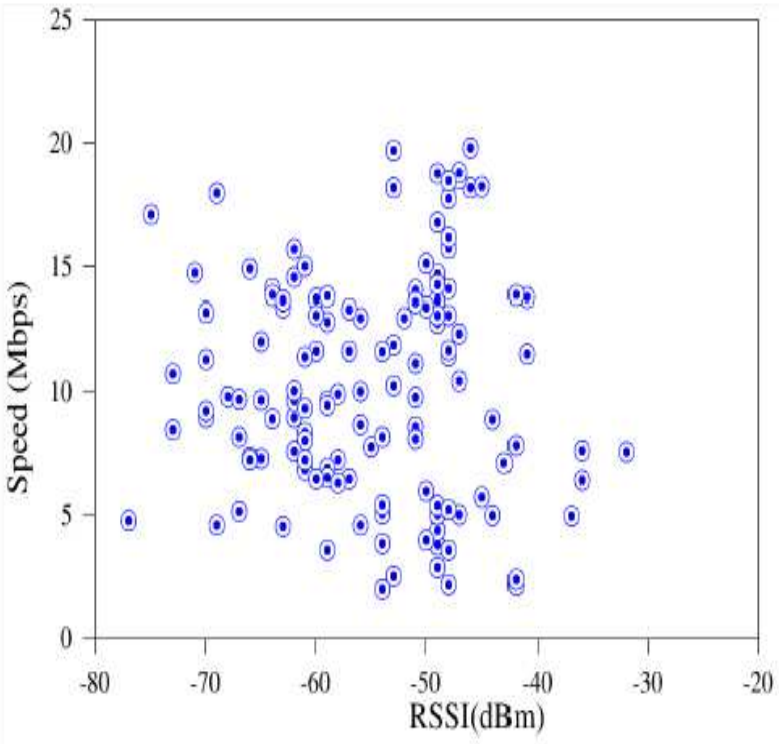

Figura 3.2 – Velocidade de download em função do valor de RSSI para um dos cenários de teste. [1]

Na Figura 3.2 está representado um gráfico com a velocidade de *download*, medido em Mbps, em função do valor de RSSI, medido em dBm. No gráfico não parece haver uma relação direta entre a velocidade de *download* e o valor de RSSI, visto que se consegue obter uma boa performance de rede tanto com valores baixos como com valores altos de RSSI, sendo assim evidente um espalhamento de pontos no gráfico. Consegue-se também observar que, além de se conseguir obter um bom desempenho de rede em toda a extensão de valores de RSSI, também há pontos onde se obtém um mau desempenho de rede nas mesmas condições, significando

assim que se consegue obter boa velocidade de download tanto com baixos valores de RSSI como com elevados. No entanto, também se consegue obter uma baixa velocidade de *download* com valores de RSSI altos e baixos.

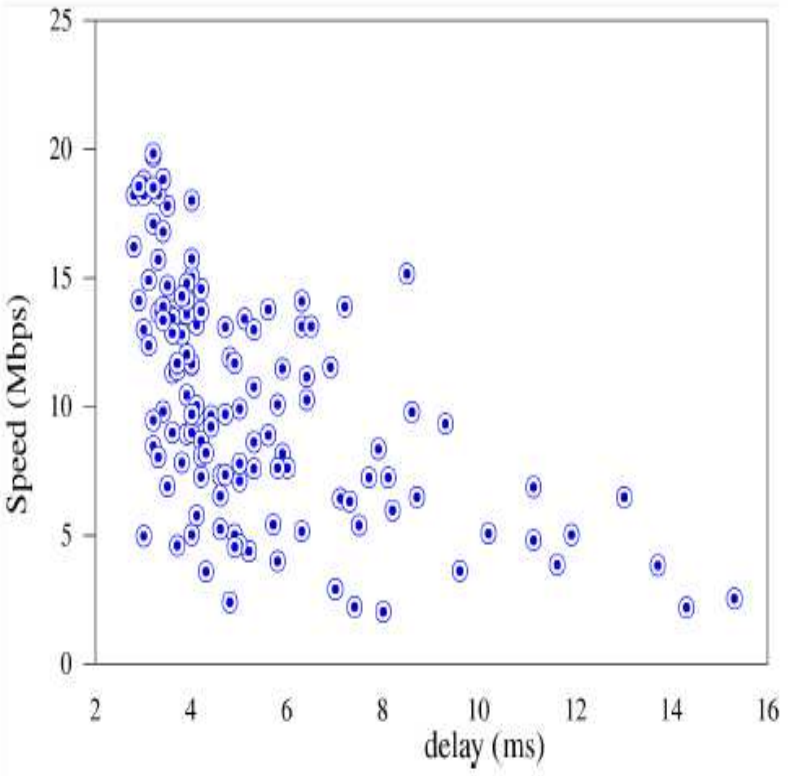

Figura 3.3 – Velocidade de download em função do valor de atraso de rede, para um dos cenários de teste. [1]

Na Figura 3.3 está representado um gráfico com a velocidade de *download* em função do valor de atraso de rede. Os valores representados no gráfico mostram uma clara discrepância na intensidade de pontos que revelam um melhor desempenho de rede quando o valor de atraso da mesma é baixo. Apesar de haver alguns pontos com o valor de atraso baixo, existe contudo um valor baixo de download. Apesar disso, pode-se verificar que, para esta experiência, com um grande atraso de rede as velocidades de *download* foram sempre também elas, baixas.

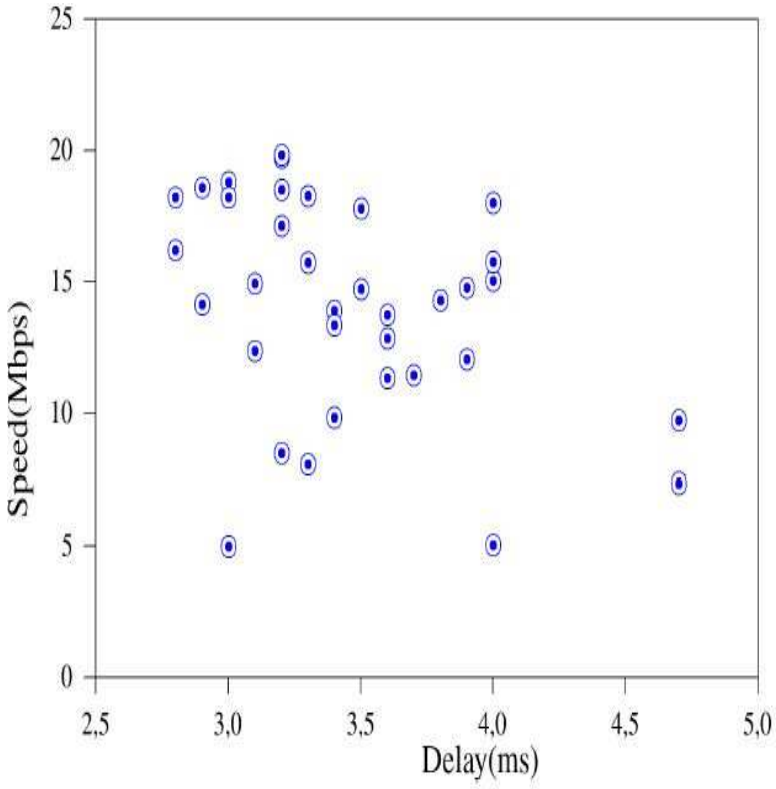

Figura 3.4 – Velocidade de download em função do valor do melhor atraso de rede em cada ponto, para um dos cenários de teste. [1]

Na Figura 3.4 está representado um gráfico com a velocidade de download em função do melhor valor de atraso de rede, ou seja, o mais baixo, para cada ponto de teste. Estes valores de atraso são um *subset* retirado da experiência apresentada na Figura 3.3.

Neste gráfico há um claro espalhamento de pontos por todo o espetro de valores levando a inferir que não há uma correlação direta entre o valor absoluto de atraso na rede com a velocidade de download. Contudo, e analisando o gráfico da Figura 3.3, podemos também inferir que se se tiver em conta o valor relativo do atraso de rede ao invés do valor absoluto, as probabilidades de se escolher uma rede que providencie uma melhor performance são superiores.

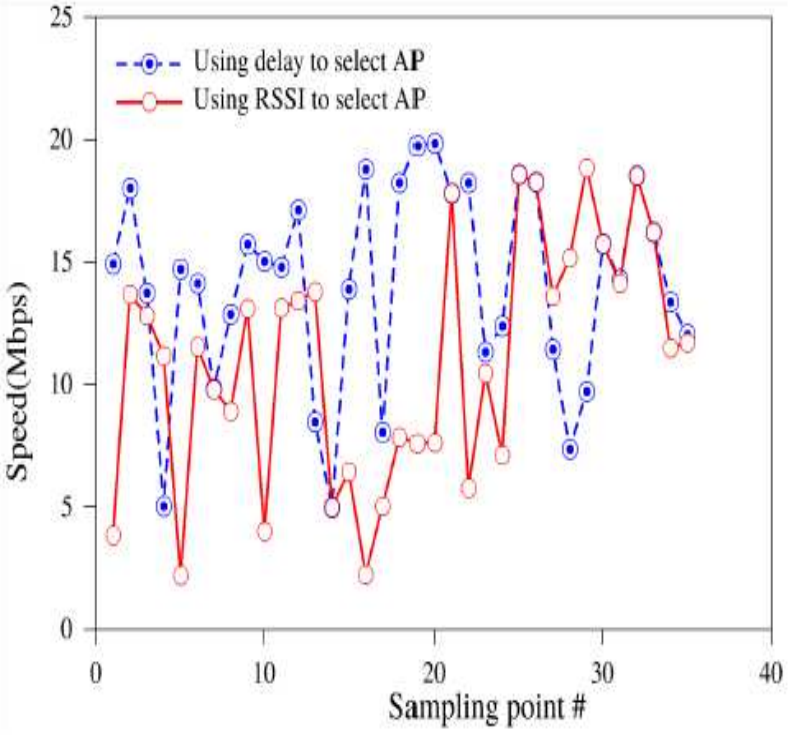

Figura 3.5 – Comparação da velocidade de download quando a escolha de AP é baseada no valor de RSSI ou no valor de atraso da rede.

Na Figura 3.5 é possível verificar o que foi dito sobre a Figura 3.3. O gráfico apresentado mostra uma comparação entre as velocidades de *download* quando o AP é escolhido com base no valor de RSSI, representado a vermelho, e com base no atraso de rede, representado a azul. De notar que os valor retirados para este gráfico fazem parte do mesmo conjunto de dados das figuras anteriores. Nestas condições de teste e para estes pontos de amostra, a escolha do AP a partir do valor de atraso de rede providenciou melhor performance da mesma em 65.71% dos casos, sendo que em 20% dos casos foram obtidas as mesmas performances usando tanto um método como o outro, e nos restantes casos, 14.29% o método que garantia melhor performance foi a escolha através do valor de RSSI.

A partir destas informações tornou-se possível o desenvolvimento da aplicação Android, que fará com que este processo se torne automático, onde o dispositivo escolherá o AP que providencie o melhor serviço de *Internet* com base no valor do atraso de rede.

3.2 Desenvolvimento da Aplicação *Android*

Após a recolha de resultados referenciados no subcapítulo anterior e respetiva análise, procedeu-se ao desenvolvimento da aplicação *Android* que consiga automaticamente ligar-se ao AP que providencia melhor serviço de Internet, através de informações do atraso de rede obtidas a partir de servidores NTP. Com essas informações é possível saber o atraso que a rede associada a cada AP tem e, com isso, escolher o AP que possui o valor de atraso de rede menor. A arquitetura do funcionamento da aplicação é muito semelhante com a arquitetura da experiência referida no subcapítulo anterior, sendo que as diferenças se prendem no facto de não ser preciso um cliente nem servidor *iperf3* pois a aplicação funcionará em situações reais de tráfego de rede. A Figura 3.6 mostra o esquema de funcionamento da aplicação.

Como se pode ver pela figura, o processo consiste em três grandes etapas:

1. Associação e autenticação ao AP;

2. Pedido de informação do atraso de rede ao servidor NTP;

3. Associar ao AP com um atraso de rede menor.

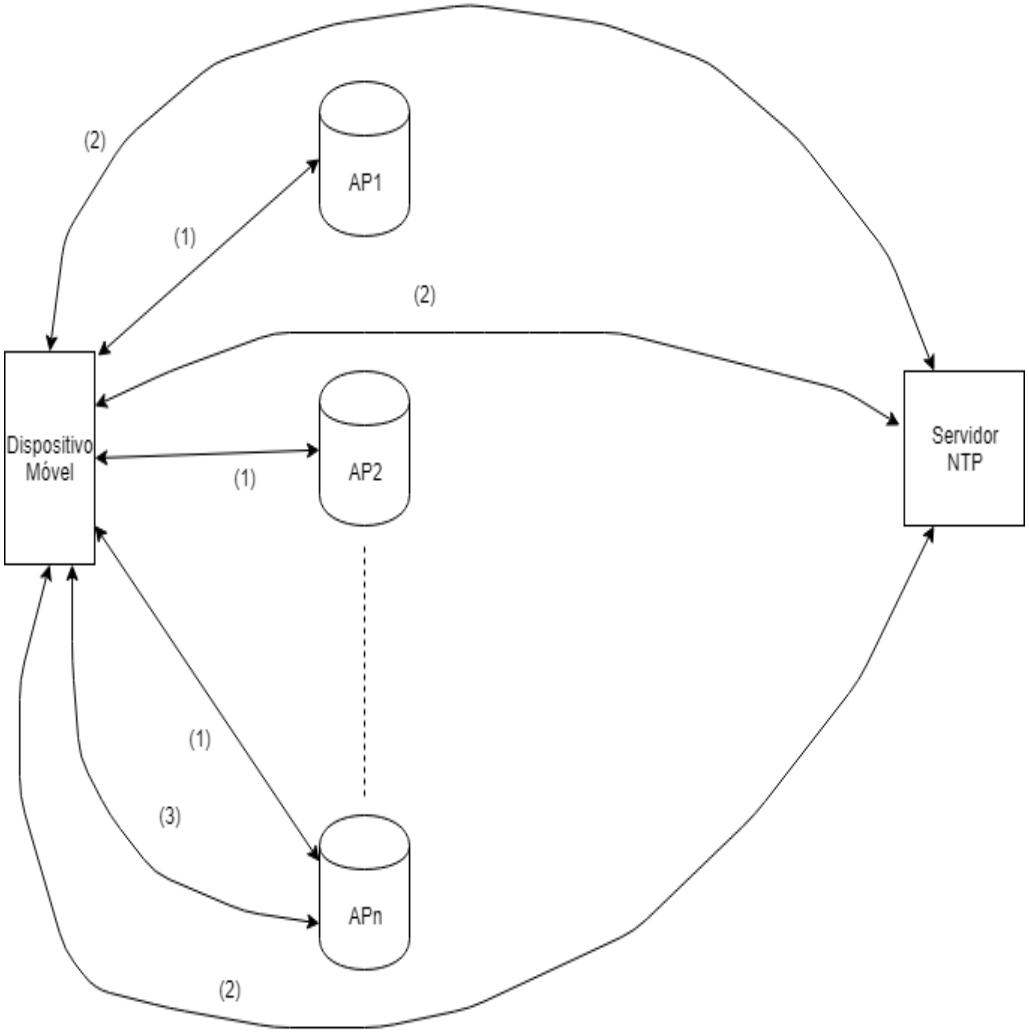

Figura 3.6 – Princípio de funcionamento da aplicação.

Com a inicialização da aplicação e antes de proceder aos passos referidos na Figura 3.6, são feitas algumas tarefas que contribuem para um funcionamento normal da aplicação. Antes de se ligar aos APs, o dispositivo precisa de saber que tipos de redes há disponíveis, se há rede Eduroam e quais os APs aos quais se pode ligar. A Figura 3.7 mostra o que acontece antes do dispositivo se ligar aos APs. Esta Figura é complementada pela Figura 3.12 que é apresentada posteriormente neste capítulo.

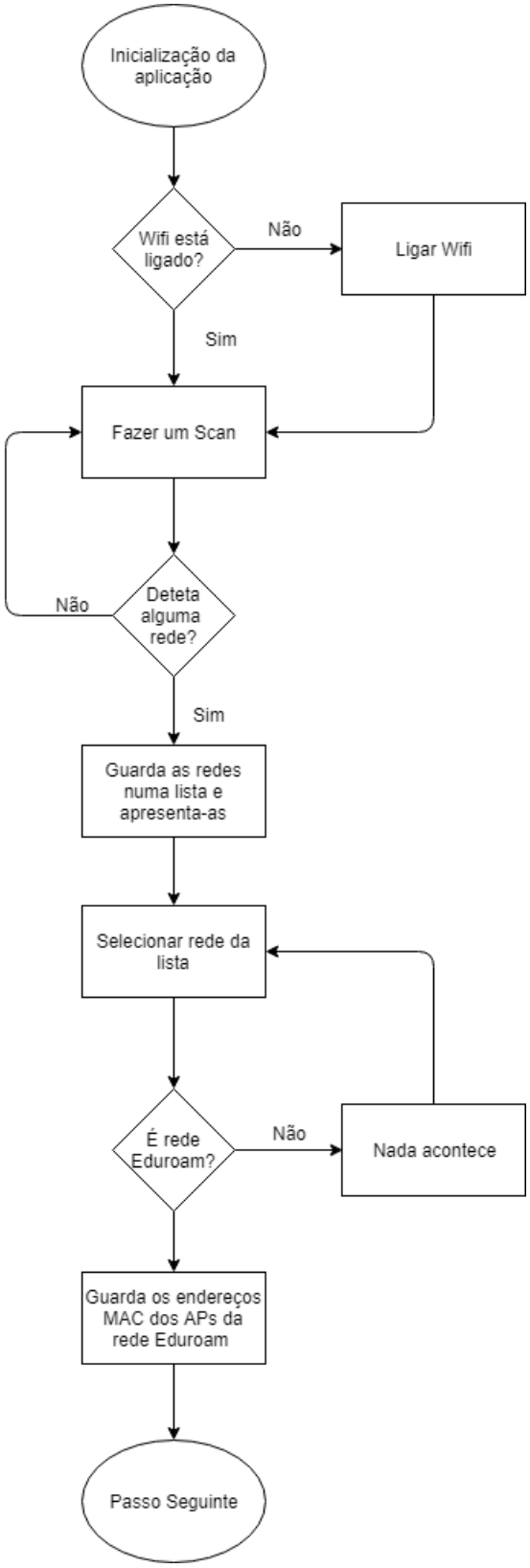

Figura 3.7 – Funcionamento da aplicação.

Ao ser iniciada a aplicação, o primeiro passo é verificar se o *Wi-Fi* está ativado, que permite funcionalidades como fazer *scan* e ligar-se a redes *Wi-Fi*. Caso já esteja ativo, a aplicação prossegue, se não estiver ativo, imediatamente essa funcionalidade é ativada. Na atividade da aplicação que é visível ao utilizador, aparece um *layout* simples que contém um botão chamado de *Scan*, e um caixa de texto no fundo da atividade onde, posteriormente, aparecerá o número de endereços MAC da rede *Eduroam* que foram encontrados, quando for premido o botão já referido. Esse *layout* está demonstrado na Figura 3.8.

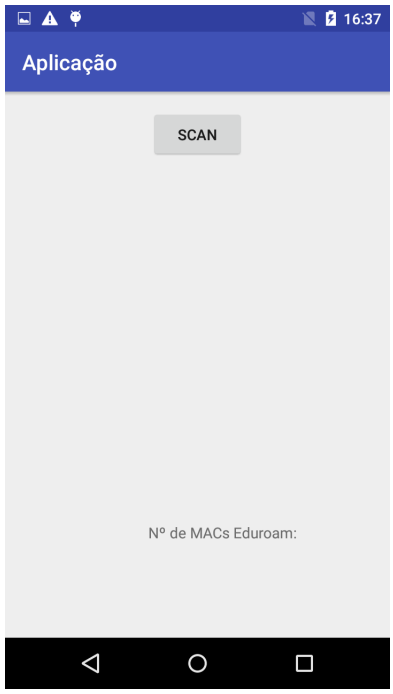

Figura 3.8 – *Layout* da primeira atividade de aplicação.

O botão *Scan* ao ser premido guarda e apresenta a lista de redes *Wi-Fi* disponíveis no momento. Caso não haja redes disponíveis, nada aparece na aplicação e o utilizador poderá premir as vezes que desejar até o dispositivo conseguir encontrar redes *Wi-Fi*, sendo que após serem encontradas e caso o botão *Scan* seja premido, serão apresentadas na aplicação. No fundo da lista também se pode ver o número de endereços MAC da rede *Eduroam* que foram detetados no momento. Também

ao ser pressionado o botão *Scan*, os endereços MAC dos APs que possuam rede *Eduroam*, são guardados, pelo que se conseguiu apurar, por ordem decrescente de potência, num vetor, querendo dizer que os APs que têm potência maior naquele momento vêm em primeiro lugar na lista. O passo de guardar os endereços MAC dos APs da rede *Eduroam* permite que aplicação tenha em sua posse uma lista de APs candidatos a serem escolhidos como melhor AP para o dispositivo de ligar. A aplicação percorre o vetor com o endereços MAC e, para cada um, procede aos testes que já foram referidos. A Figura 3.9 mostra um caso de redes encontradas.

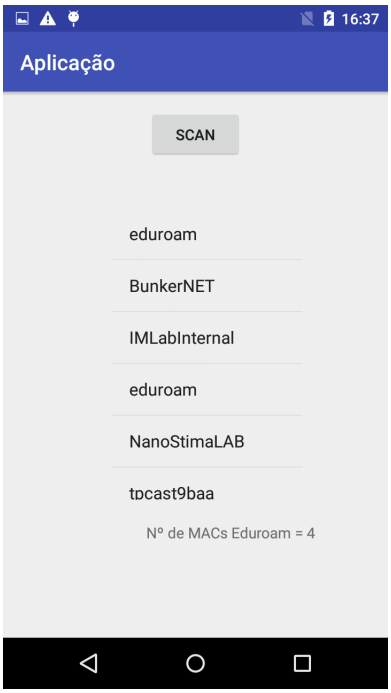

Figura 3.9 – *Layout* da segunda atividade de aplicação.

De referir que há repetição de redes, ou seja, há mais que uma rede com o mesmo nome, pois todos os APs que possuem uma rede com o mesmo SSID, são diferenciados e são todos listados. A lista que é apresentada na Figura tem a opção de ser "*Clickable*", ou seja, cada opção da lista pode funcionar como um botão. Nesta fase da aplicação, e visto que a aplicação é direcionada para funcionar na rede *Eduroam*, se for pressionado alguma opção que não seja a *Eduroam*, nada acontece. Caso seja

premida uma opção *Eduroam* é aberta uma outra atividade na aplicação onde aparecem dois campos de *login*, nomeadamente as credenciais que são necessárias para ser possível a autenticação na *Eduroam*. Também está presente um botão denominado de *Tempo* que deve ser pressionado quando as credenciais estão definidas de modo a proceder aos passos seguintes da aplicação. Caso não sejam preenchidos os campos de autenticação ou caso sejam mal preenchidos, i.e credenciais erradas, o dispositivo não se consegue autenticar na *Eduroam*. Com isto a aplicação fica parada pois está a tentar autenticar-se com credencias erradas e a aplicação tem que ser reiniciada pelo utilizador. Esse *layout* está presente na Figura 3.10.

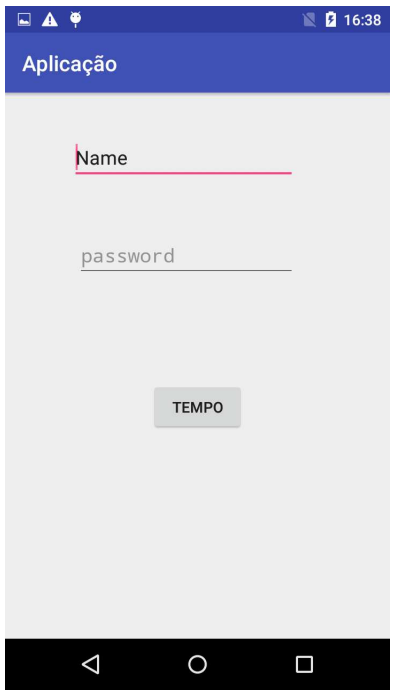

Figura 3.10 – *Layout* da terceira atividade de aplicação.

Após isto, o passo seguinte em desconetar o dispositivo da rede em que o dispositivo estaria ligado e esquecer a mesma. O objetivo deste passo consiste em fazer com que o dispositivo ''esqueça'' as informações da rede anterior, para que, desta maneira, as ligações seguintes sejam feitas de forma a que informações guardadas da rede anterior não tenham influência no comportamento dos passos seguintes da aplicação.

Após se desconetar e esquecer a rede anterior, o objetivo do passo seguinte da aplicação passaria por ligar-se ao AP com o primeiro endereço MAC que estaria no vetor. O problema com esta tarefa é que não há um método automático para ligar a um AP através do seu endereço MAC. Para superar este problema, foi criado um método onde os endereços MAC que estão guardados no vetor, são passados, um a um, como argumento. Este método cria uma rede Eduroam, com as definições de autenticação e segurança que estão definidas pelas entidades competentes da Universidade de Trás-os-Montes e Alto Douro. Além do endereço MAC, são passados também como argumento as credenciais de autenticação do utilizador do dispositivo. Assim, e para cada endereço MAC, é criada uma nova configuração da rede no dispositivo que se liga ao AP cujo endereço MAC foi passado como argumento. O procedimento da criação de uma rede é possível através das ferramentas disponibilizadas na API do *Android*. A partir da classe *Wi-FiConfiguration* é permitido criar, configurar e gerir redes *Wi-Fi* bem como as suas definições de segurança. A justificação da obrigatoriedade Na Figura 3.11 está representado como foi criada a rede Eduroam:

```
protected int createNetworkFirst(String mail,String pass, String mac, WifiManager mainWifi) {

    WifiConfiguration conf = new WifiConfiguration();
    conf.SSID = "\"eduroam\"";
    conf.BSSID = mac;
    conf.status = WifiConfiguration.Status.ENABLED;
    conf.allowedKeyManagement.set(WifiConfiguration.KeyMgmt.WPA_EAP);
    conf.enterpriseConfig.setEapMethod(WifiEnterpriseConfig.Eap.PEAP);
    conf.enterpriseConfig.setIdentity(mail);
    conf.enterpriseConfig.setPassword(pass);

    int a = mainWifi.addNetwork(conf);//Caso retorne valor negativo
    //a rede não foi criada, caso contrário, a rede foi criada com sucesso.
    mainWifi.enableNetwork(a, true);
    System.out.println("Criou a nova rede Eduroam");


    return a;
}
```

Figura 3.11 – Configuração da rede para se ligar à Eduroam.

Para se criar uma rede *Wi-Fi*, com os parâmetros desejados, são necessários os seguintes passos:

- Ativar a configuração de rede *Wi-Fi*. Com a classe *WifiConfiguration* é permitido ter acesso às configurações gerais e de segurança de uma rede *Wi-Fi*;

- Definir o nome da rede. Com o campo *SSID* é possível dar o nome à rede. Foi escolhido o nome *Eduroam*;

- Definir o endereço MAC do AP ao qual queremos estabelecer ligação. Com o campo *BSSID* é definido o endereço MAC do AP desejado, sendo que esse valor é passado como parâmetro para a função;

- Definir o estado da rede criada como ativa. Com o campo *.status* é possível definir o estado da rede;

- Definir o tipo de protocolo para manipulação de chaves de autenticação. Com o campo *allowedKeyManagement* define-se o tipo de protocolo que são suportados pela configuração. Neste caso foi escolhido `WPA_EAP`;

- Definir o protocolo EAP a ser utilizado. Com a escolha feita no ponto anterior, a partir da classe *entrepriseConfig* é permitido escolher o protocolo de segurança desejado. Foi escolhido o protocolo *PEAP*;

- Definir as credenciais do utilizador. Com a classe *entrepriseConfig* é permitido introduzir a identidade e a *password* do utilizador com os campos *setIdentity* e *setPassword*, respetivamente. Os dois campos são passados como argumento para o método.

Quando a aplicação já tem a lista dos endereços MAC, cria uma rede *Eduroam* com um endereço MAC da lista e contacta um servidor NTP de modo a solicitar os tempos de atraso de rede desejados. O servidor NTP foi escolhido de uma lista pública de servidores NTP que estão disponíveis na *internet*. Para a execução deste passo, foi usado uma *AsyncTask*, de modo a não sobrecarregar o *mainThread*. O processo está representado na Figura 3.12.

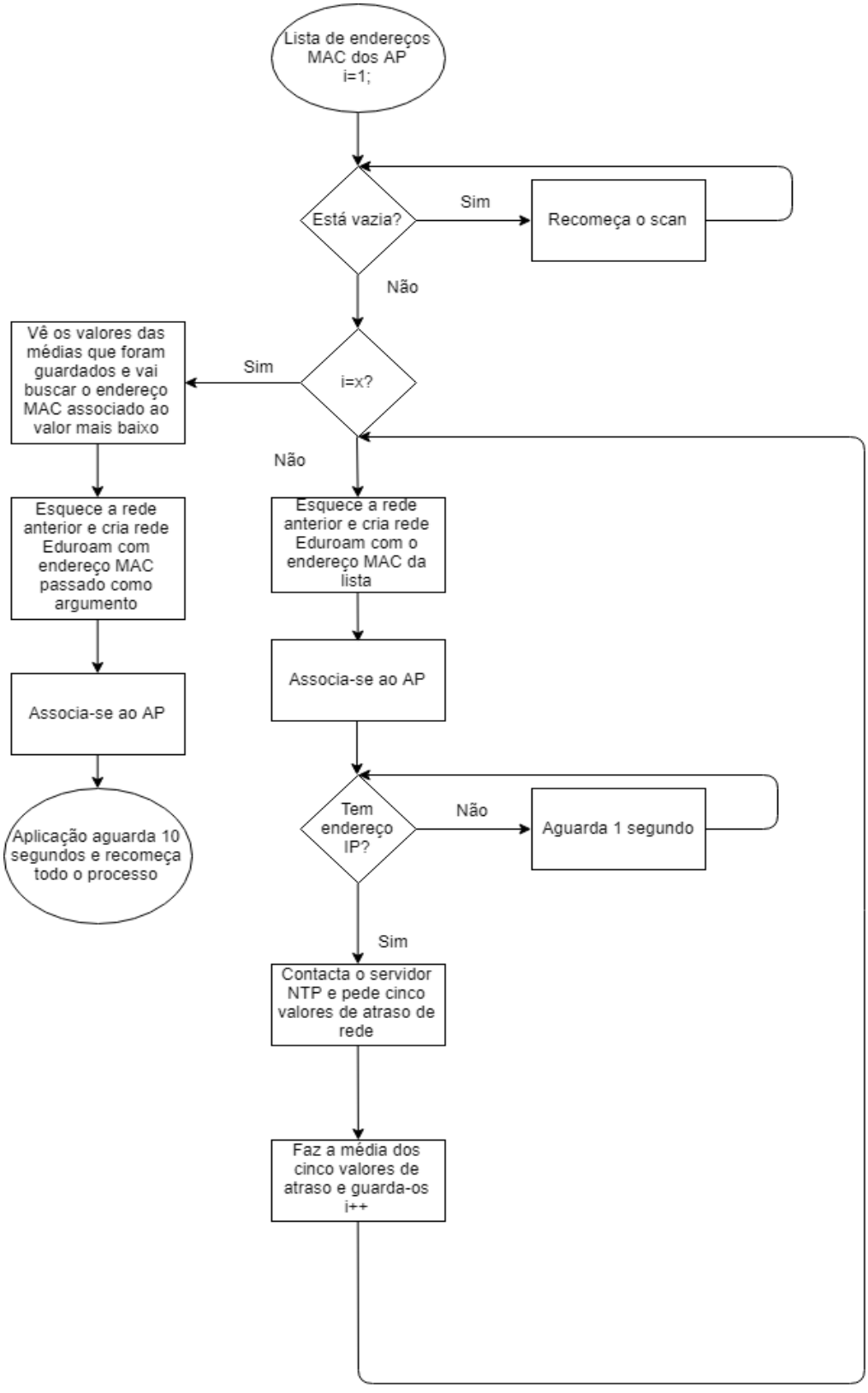

Figura 3.12 – Fluxograma mais detalhado do funcionamento da aplicação.

Após o *scan* e a aquisição dos endereços MAC dos APs com a rede *Eduroam*, a aplicação verifica se a lista está vazia, pois pode dar-se o caso de não haver APs disponíveis na altura. Se isso se verificar o *scan* é efetuado até encontrar APs da rede *Eduroam*. Para cada endereço MAC da lista são executados os passos necessários para obter o atraso de rede de modo a que o dispositivo fique ligado ao AP que ofereça um atraso de rede menor. Nem sempre todos os endereços MAC da lista são utilizados. A partir de testes feitos com a aplicação, chegou-se à conclusão que quando os APs têm uma potência de sinal reduzida, o dispositivo ou demora muito a associar-se a esses APs, ou nem sequer consegue fazer essa associação. Por isso, foi definido um *threshold* de potência -70 dBm, pois verificou-se que para esses níveis de potência há uma grande dificuldade para o dispositivo se conseguir associar. Isso leva a que a lista possa por vezes ter um número pequeno de endereços MAC aos quais se consegue ligar, levando a que o tempo total de execução da aplicação seja mais reduzido.

Enquanto a potência de sinal dos APs dos endereços MAC da lista for maior que o valor atribuído, para cada endereço é esquecida a rede *Eduroam* anterior e é criada uma nova com o endereço MAC que foi passado como argumento. De seguida, o dispositivo associa-se a esse AP e a aplicação aguarda que seja obtido o endereço IP. Quando isso acontece, é contactado o servidor NTP e são pedidos cinco valores de atraso de rede, para que se consiga fazer a média com vista a obter um valor coerente do atraso de rede desse AP. Após isso, esse valor é guardado num vetor onde, posteriormente, vai ser comparado com os restantes valores de média dos outros APs. Este procedimento é repetido para todos os APs que registaram um valor de potência de sinal superior a -70 dBm. Quando se chega ao final da lista ou quando a potência de sinal do AP é inferior a -70 dBm, a aplicação compara os valores da média do atraso de cada AP e verifica qual é o mais baixo. De seguida, a aplicação vai buscar o endereço MAC do AP que registou o valor de média do atraso mais baixo, esquece a rede *Eduroam* anterior e cria uma nova rede *Eduroam* com esse endereço MAC passado como argumento e assim, o dispositivo conecta-se ao AP.

Com este último passo e segundo os pressupostos teóricos já abordados, há uma probabilidade de 65.71% de o dispositivo estar ligado ao AP que providencia o melhor serviço de *Internet* naquele dado momento. De modo a ter um serviço de melhor rede contínuo, foi definido um intervalo de 10 segundos para que todo o processo da aplicação se volte a repetir. Assim, em casos de grande mobilidade, o dispositivo está sempre atualizado em relação ao AP que providencia o melhor serviço de *Internet*. Mesmo em casos de pouca mobilidade, o atraso de rede é volátil em curtos espaços de tempo, por isso é que também é aconselhável que o processo da aplicação se repita em intervalos regulares, não muito longos.

4 Testes e Resultados

Neste capítulo são apresentados resultados mais pormenorizados da experiência descrita no capítulo 3, na explicação do funcionamento da aplicação, bem como na apresentação de problemas e possíveis solução encontrados ao longo do seu desenvolvimento.

4.1 Resultados da Experiência

Neste subcapítulo vão ser apresentados, em mais detalhe, os resultados dos testes realizados, onde foi possível determinar a viabilidade do método que foi explorado. De referir que todos estes resultados, tanto deste subcapítulo, como do capítulo 3, estão apresentados no artigo [1]. No Capítulo 3, foram descritos alguns resultados das experiências que foram feitas com o intuito de corroborar o facto que usando o atraso de rede como fator principal para realizar o *handoff*, ao invés de usar a potência de sinal do AP, haver uma maior probabilidade de obter um melhor serviço de *Internet*. Como já foi referido e para relembrar, foram montados três cenários para a experiência:

1. Duas salas de aula com um corredor, cobertos com uma rede com quatro APs;
2. Duas salas de aulas com um corredor, cobertos com uma rede com três APs;
3. Uma sala de aula com um corredor, cobertos com uma rede com três APs.

Para a definição dos resultados foram definidos os seguintes parâmetros quanto à qualidade da rede selecionada:

- **Pior** - Quando a pior rede foi selecionada;
- **Melhor** - Quando a melhor rede foi selecionada;
- **Bom** - Quando nem foi a melhor nem a pior rede a ser selecionada.

Na tabela 4.1 estão representados os resultados sumariados do cenário 1, duas salas com uma rede de quatro APs. Em cada ponto de recolha dos dados foram recolhidas dez amostras de atrasos de rede, com um intervalo de 100ms entre cada amostra, sendo que foram coletadas pelo menos 35 amostras ao longo das duas sala de aula. Para este cenário, foram realizados dois testes, no Teste 1 estava presente uma estação que injetava na rede tráfego *UDP (User Datagram Protocol)* de 10Mbps, para simular tráfego normal na rede, e o Teste 2 foi realizado sem tráfego na rede. Não havia mais nenhuma estação ou dispositivo ligado à rede. Para cada um dos testes, foram obtidos os valores de RSS e de atraso de rede.

	Pior		Bom		Melhor	
	RSSI	Atraso	RSSI	Atraso	RSSI	Atraso
Teste 1	22.86%	0,00%	40,00%	31,43%	37,14%	68,57%
Teste 2	25,71%	20,00%	51,43%	45,71%	22,86%	34,29%

Tabela 4.1 – Tabela de resultados com os dados obtidos no cenário 1.

Como se pode ver na Tabela 4.1, a seleção da rede para o dispositivo se ligar baseado no atraso de rede nas condições descritas, leva a um baixo número de escolhas da

pior rede e a um aumento de escolhas da melhor rede. Tanto no Teste 1 como no Teste 2, houve uma menor percentagem de escolhas da pior rede usando o atraso de rede, 0,00% e 20,00% contra 22,86% e 25,71%, respetivamente, e uma maior percentagem de escolhas da melhor rede, 68,57% e 34,29% contra 37,14% e 22,86%.

Na Tabela 4.2 estão descritos os resultados do cenário 2, duas salas com uma rede de três APs. A localização deste cenário é a mesma do cenário anterior com a diferença de se ter reduzido o número de APs de 4 para 3, reduzindo a densidade de APs e consequentemente, obter um cenário mais próximo da realidade. Foram coletados 26 pontos de amostra e em cada ponto foram recolhidas 10 amostras do atraso de rede.

	Pior		Bom		Melhor	
	RSSI	Atraso	RSSI	Atraso	RSSI	Atraso
Teste 1	38,46%	15,38%	26,92%	19,23%	34,62%	61,54%
Teste 2	15,38%	11,54%	34,62%	26,92%	50,00%	65,38%

Tabela 4.2 – Tabela de resultados com os dados obtidos no cenário 2.

Para o cenário 2, os resultados mostram a mesma tendência. Usando o atraso de rede como parâmetro para a escolha do melhor AP tem, também neste cenário, um melhor desempenho do que a escolha baseada no RSSI, 15,38% e 11,54% contra 2«38,46% e 15,38% na escolha da pior rede e 61,54% e 65,38% contra 34,62% e 50,00% para a escolha da melhor rede.

Na Tabela 4.3 estão descritos os resultados do cenário 3, uma sala com uma rede de 3 APs. Foram recolhidas 100 amostras do atraso de rede em cada ponto de recolha de dados. O objetivo deste terceiro teste é avaliar o impacto do número de amostras no desempenho da arquitetura. É expectável que um número reduzido de amostras leve a um pior resultado mas, um elevado número de amostras também pode ter um impacto negativo na performance visto que o intervalo entre amostras é de 100ms. Isto significa que, para recolher 10 amostras, o procedimento demora cerca de 1 segundo, se forem recolhidas 100 amostras, demorará 10 segundos para cada AP.

Na Tabela 4.3 são apresentados os valores de ‘’Pior‘’, ‘’Bom‘’ e ‘’Melhor‘’ quando o parâmetro escolhido para a escolha do AP é o valor de RSSI e quando 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100 amostras do valor de atraso de rede são escolhidos para o mesmo efeito. Um terço das amostras foi recolhido sem tráfego na rede, um terço com duas estações a injetarem tráfego UDP a 10Mbps na rede e um terço com duas estações a injetarem tráfego de UDP a 20Mbps na rede. O objetivo foi ter um cenário parecido com as condições de rede em circunstâncias reais. Resultados com uma amostra usam o valor do primeiro valor de atraso, resultados com duas amostras usam a média do dois primeiros valores de atraso e assim sucessivamente.

Para ter uma melhor compreensão dos resultados, na Figura 4.1 está representado um gráfico que indica a percentagens de decisão (*% of decisions*), em função do número de amostras (*# of Samples*), usando os dados da Tabela 4.3.

	Pior	Bom	Melhor
RSSI	25,00%	33,33%	41,67%
1 amostra	25,00%	25,00%	50,00%
2 amostras	20,83%	16,67%	62,50%
3 amostras	20,83%	12,50%	66,67%
4 amostras	8,33%	16,67%	75,00%
5 amostras	12,50%	12,50%	75,00%
6 amostras	12,50%	16,67%	70,83%
7 amostras	16,67%	20,83%	62,50%
8 amostras	16,67%	20,83%	62,50%
9 amostras	25,00%	16,67%	58,33%
10 amostras	20,83%	20,83%	58,33%
20 amostras	16,67%	16,67%	66,67%
30 amostras	12,50%	12,50%	75,00%
40 amostras	12,50%	12,50%	75,00%
50 amostras	8,33%	8,33%	83,33%
60 amostras	4,17%	8,33%	87,50%
70 amostras	16,67%	16,67%	66,67%
80 amostras	12,50%	16,67%	70,83%
90 amostras	4,17%	16,67%	79,17%
100 amostras	16,67%	12,50%	70,83%

Tabela 4.3 – Tabela do cenário 3 para diferentes tamanhos de amostra do atraso de rede.

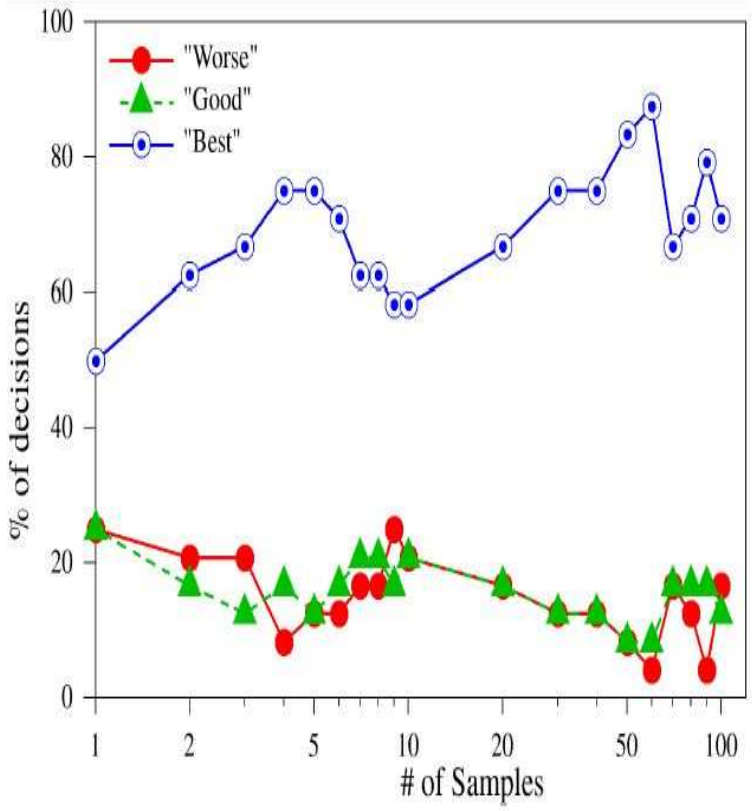

Figura 4.1 – Comparação entre *Best, Good* e *Worse* (Melhor, Bom e Pior, tradução para português) seleção de rede, para diferentes tamanhos de amostra.

Independentemente do número de amostras usado, os dados mostram a mesma tendência dos testes anteriores, querendo dizer que usando o atraso de rede, ocorre um melhor desempenho. Os resultados mostram que, mesmo havendo flutuações, com o aumento do número de amostras, tendencialmente aumenta o desempenho. De notar que mesmo com apenas uma amostra, o desempenho também é ligeiramente melhor. Mesmo com um número baixo de amostras, que leva a ter menos tempo para ''perceber'' a qualidade do AP, é possível obter resultados bastante positivos. Por exemplo, com apenas 5 amostras, foi possível ter 75% das decisões na ''Melhor'' rede. As variações que foram encontradas justificam-se com a sobreposição de canais, pois a rede para testes foi montada num local com outras redes *Wi-Fi* e por isso foi impossível reverter este problema. Outra questão foi o facto de poder ter havido interferências. Apesar de os testes terem sido feitos com um número reduzido

de pessoas no local, esse número não era zero e o seu tráfego na rede poderá ter causado essas mesmas interferências.

4.2 Demonstração do Funcionamento da Aplicação

A aplicação *Android* desenvolvida nesta dissertação foi desenhada para funcionar maioritariamente em *background* logo, para a demonstração da mesma, recorreu-se a resultados retirados do *logcat* do monitor *Android* à medida que a aplicação é executada no dispositivo. Nesta subsecção, serão apresentados e explicados esses mesmos resultados sequencialmente.

- **Função Wi-Fi Ativada** - Este primeiro passo acontece quando a aplicação é iniciada e ativa a função *Wi-Fi* do dispositivo;
- **Botão Scan premido** - Após o início da aplicação, como foi referido no Capítulo 4, aparece uma atividade com o botão *Scan*. Ao ser premido esse botão a aplicação inicia um *scan* das redes *Wi-Fi* circundantes;
- **[eduroam, BunkerNET, eduroam, IMLabInternal, eduroam, IMLabInternal, eduroam, NanoStimaLAB, ImersiveLabExternal, guest-utad, guest-utad, guest-utad]** - Este é o resultado do passo anterior. Como já foi referido, a questão de haver redes repetidas, deve-se ao facto de haver vários APs que têm essas redes configuradas e a aplicação diferencia-as;
- **Opção eduroam premida** - Visto que a aplicação só funciona para redes *eduroam*, para a aplicação seguir para o próximo passo, tem que ser premida a opção *eduroam* na lista de redes providenciada pelo passo anterior;
- **Lista de endereços MAC dos APs com rede Eduroam - [[], 80:2a:a8:5a:e0:61, 80:2a:a8:5b:e0:61, f0:9f:c2:f8:3a:3c, 00:3a:98:b0:54:b0]** - Após ser premido a opção *Eduroam*, a aplicação filtra e guarda os endereços MAC apenas dos APs da rede *Eduroam*;

- **Botão Tempo premido** - Após a opção *Eduroam* ser selecionada, aparece outra atividade no dispositivo onde está presente um botão denominado de Tempo. Ao ser premido esse botão o resto da aplicação funciona sem ser necessário mais alguma interação com o utilizador do dispositivo;
- **-58 dBm** - Visto que a aplicação só se associa a um AP caso a sua potência seja superior a -70 dBm, decidiu-se mostrar a potência de sinal do AP antes de proceder aos passos seguintes.
- **Esquece a rede anterior** - Após se ter verificado que a potência de sinal do AP é superior a -70 dBm, a aplicação procede a esquecer as informações de rede a que o dispositivo estava ligado anteriormente;
- **Criou a nova rede Eduroam** - A aplicação cria agora a rede *Eduroam* onde o dispositivo se vai ligar, tendo em conta o endereço MAC da lista;
- **80:2a:a8:5a:e0:61 Endereço MAC atual** - O dispositivo fica associado e autenticado ao endereço MAC da lista que foi passado como argumento;
- **192.168.128.140** - De modo à aplicação conseguir continuar a execução das tarefas seguintes, precisa de estar registado com endereço IP;
- **Antes de contactar o servidor NTP** - Para se conseguir aferir o atraso rede, é necessário contactar um servidor NTP, e pedir cinco valores de atraso de rede, em milissegundos:
  - Valor de Atraso 58;
  - Valor de Atraso 59;
  - Valor de Atraso 54;
  - Valor de Atraso 56;
  - Valor de Atraso 42.
- **Depois de ter contactado o servidor NTP** - Após terem sido adquiridos os cinco valores de atraso de rede, é feita a média dos mesmos e coloca o valor num vetor.

O procedimento anterior é executado para todos os APs da lista desde que cumpram os requisitos que foram definidos, nomeadamente o valor de potência de sinal. A restante execução da aplicação ocorreu da seguinte maneira:

- **-57 dBm**
- **Esquece a rede anterior**
- **Criou a nova rede Eduroam**
- **80:2a:a8:5b:e0:61 Endereço MAC atual**
- **192.168.128.140**
- **Antes de contactar o servidor NTP**
  - Valor de Atraso 41
  - Valor de Atraso 40
  - Valor de Atraso 40
  - Valor de Atraso 40
  - Valor de Atraso 40
- **Depois de ter contactado o servidor NTP**
- **-64 dBm**
- **Esquece a rede anterior**
- **Criou a nova rede Eduroam**
- **f0:9f:c2:f8:3a:3c Endereço MAC atual**
- **192.168.128.140**
- **Antes de contactar o servidor NTP**
  - Valor de Atraso 41

  - Valor de Atraso 40
  - Valor de Atraso 40
  - Valor de Atraso 40
  - Valor de Atraso 41

- **Depois de ter contactado o servidor NTP**
- **[53.0, 40.0, 40.0]** - À medida que os tempos de atraso de rede são recebidos, para cada AP é feita a média dos mesmos e são colocados num vetor por ordem. A partir daqui é escolhido o valor AP que apresentou o valor mais baixo da lista.
- **80:2a:a8:5b:e0:61 Endereço MAC escolhido** - Nesta fase, a aplicação escolheu o endereço MAC do AP ao qual corresponde o valor mais baixo de atraso de rede e é nesse AP que o dispositivo fica registado.

Estes resultados representam apenas um ciclo da execução da aplicação. Foi escolhido o AP onde o dispositivo fica conectado. Este processo é repetido automaticamente a cada 10 segundos de maneira a, como foi referido anteriormente, ter em conta a mobilidade do utilizador e também a volatilidade dos valores de atraso de rede. De notar que, nos resultados descritos acima, havia quatro endereços MAC configurados com a rede *Eduroam* e esses seriam os possíveis candidatos para o dispositivo ficar conectado. No entanto, o vetor onde se guarda a média dos valores de atraso só contém três médias, o que quer dizer que só em três APs é que foram efetuados os pedidos de tempo de atraso de rede. Isto significa, nestes resultados, que o último AP a ser testado apresentava uma potência de sinal inferior a -70 dBm, sendo que foi descartado de fazer os pedidos de tempo de atraso. Outro pormenor a ser referido é que, como se pode ver no vetor de médias apresentado, aparecem dois valores iguais, nomeadamente 40 ms. Como foi o valor mais baixo que foi registado e visto que há dois APs com o mesmo valor de média de atraso de rede, a escolha recai sobre a potência de sinal do AP. Entre os dois APs com a mesma média foi escolhido que apresentava um valor de potência de sinal superior.

Problemas encontrados

Devido às instabilidades das ligações entre os dispositivos e os APs, foi necessário adicionar algumas cláusulas de modo a garantir que a aplicação funcionasse sem interrupções. Os problemas encontrados foram os seguintes:

- Incapacidade de contactar o servidor NTP. Durante a implementação da aplicação surgiram dois problemas nesta vertente:
  - Sobrecarga do *mainThread* - Na execução da aplicação, numa primeira implementação, todo o processo era executado no *thread* principal da aplicação e quando tentava contactar com o servidor NTP de modo a obter os valores de atraso de rede, ou demorava um tempo inaceitável a obter resultados, ou podia mesmo fazer com que a aplicação deixasse de funcionar. O erro que era apresentado no *Android Monitor* referia a quantidade excessiva de processos a ser executados no *thread* principal. Este erro foi solucionado criando uma *asynchronous task* para executar o contacto com o servidor NTP numa outra *thread*. Assim, o processo de pedido de tempos de atraso e, no geral toda a aplicação, era executada mais rapidamente;
  - Rede inatingível - Após ser resolvido o problema anterior, apareceu uma outra questão no que toca ao contacto com o servidor NTP. Quando se obtinha os endereços MAC candidatos para se fazer o processo de *handoff*, a lista dos mesmos era percorrida sequencialmente, através de um ciclo, para depois serem executados os procedimentos seguintes. Acontece que com esta abordagem, o tempo atribuído para serem executados os passos todos para cada endereço MAC não era suficiente. Isto tinha como consequência que o passo que demorava mais tempo a ser processado, nomeadamente o estabelecimento de ligação com o servidor NTP, não fosse executado, reportando o erro de *Network Unreachable*. Para solucionar este problema, foi necessário mudar o modo de implementação,

criando-se cláusulas onde só se permitia que a execução do passo seguinte fosse feita caso o passo atual tivesse terminado.

- Incapacidade de obter endereço IP. Durante a execução de todo o processo, é necessário ter ligação à *Internet*, o que só se consegue quando se obtém endereço IP. Devido, muitas vezes, a questões alheias à aplicação, o dispositivo demorava muito tempo a obter o endereço IP ou até nem o conseguia obter. Acontecia que a aplicação tentava executar os passos seguintes: estabelecer contacto com o servidor NTP, o que é impossível sem ter acesso à *Internet*. Para resolver este problema, definiu-se um tempo de espera, antes de contactar com o servidor NTP, onde caso ainda não tivesse adquirido endereço IP, a aplicação aguardava esse tempo de espera;

- Incapacidade do dispositivo em associar-se a APs com baixa potência de sinal. Relacionado com o problema anterior, esta questão surge também quando o dispositivo não consegue obter endereço IP. No caso do problema anterior, as causas podem ser, por exemplo, problemas de autenticação. Neste caso, apesar de se conseguir obter endereços MAC de todos os APs configurados com a rede *Eduroam*, alguns desses APs podem apresentar um valor de potência de sinal demasiado baixo para o dispositivo se conseguir associar e autenticar. Através das várias experiências feitas ao longo do desenvolvimento desta dissertação, chegou-se à conclusão que APs que apresentam uma potência de sinal na ordem dos -70 dBm ou inferior, fazem com o que dispositivo demore muito a estabelecer ligação à *internet* ou que não consiga mesmo associar-se ao AP. Para solucionar esta questão, definiu-se um valor mínimo de potência de sinal, neste caso de -70 dBm para que a aplicação prossiga com a associação do dispositivo ao AP. Caso aparecesse um AP que apresente uma potência de sinal menor ou igual a -70 dBm, o mesmo é descartado da restante execução da aplicação.

- Incapacidade de funcionar em todos os dispositivos. Como já foi referido, a aplicação foi instalada e testada em diversos dispositivos com versões diferentes do sistema operativo *Android*, de modo a obter coerência nos resultados.

Apesar de em grande parte dos dispositivos testados, a aplicação tivesse funcionado conforme o esperado, deparou-se com certas versões onde não era permitida a configuração da rede *Eduroam* conforme foi desenvolvido. Julga-se que isto se deve à falta de permissões onde os dispositivos não permitem que certas tarefas sejam executadas. Este problema não foi resolvido, pois está dependente de características incluídas pelos fabricantes nos seus *firmwares*.

5 Conclusões e Trabalho Futuro

5.1 Conclusões

O crescimento de redes *Wi-Fi* tem sido uma constante ao longo dos anos e, com isso, a crescente exigência por parte dos utilizadores para ter um serviço mais rápido e sem falhas surge com naturalidade. Na vertente de mobilidade, os utilizadores de redes *Wi-Fi* querem cada vez mais ter acesso contínuo, ininterrupto e rápido à *Internet*. Para ir de encontro a essas mesmas exigências, melhoramentos no processo de *handoff* são necessários e importantes.

Com isto em vista têm surgido estudos, aplicações e algoritmos com o objetivo de otimizar esse mesmo processo. Como foi demonstrado anteriormente, grande parte dos estudos realizados definem a meta de reduzir o tempo de atraso provocado pelo *scan* de APs próximos do dispositivo. O sistema implementado nesta dissertação tem o objetivo de criar um método de *handoff* que garantisse que o dispositivo ficasse ligado ao AP garantindo uma velocidade de *download* maior. Através de pedidos de atraso de tempo a um servidor NTP consegue-se estimar o atraso da rede a um dado momento, sendo que, quanto menor o atraso, melhor será o serviço de *Internet*.

A aplicação foi testada em vários dispositivos móveis com o sistema operativo *Android* e os testes foram realizados na Universidade de Trás os Montes e Alto Douro, na rede Eduroam. Apesar de ser difícil perceber as alterações da velocidade de *Internet* do ponto de vista do utilizador, os resultados analisados anteriormente mostram que utilizando os valores de atraso de rede como parâmetro preferencial, a probabilidade de ter uma velocidade de *download* superior é bastante elevada. Isto leva à conclusão que, apesar de poder ser impercetível ao utilizador, o serviço de *Internet* providenciado é de melhor qualidade.

Apesar de ser necessário melhorar a aplicação em vários aspetos, numerados e explicados no subcapítulo *Trabalho Futuro*, ficou provado que o método desenvolvido nesta dissertação, usando os valores de atraso da rede, escolhe o AP que providencia o melhor serviço de *Internet*, em termos de velocidade de *download*. Apesar de haver a possibilidade dos APs escolhidos por este método poderem coincidir com os APs escolhidos na abordagem tradicional, através do valor RSS, os resultados apresentados provam que, usando este último método, nem sempre é escolhido o AP que oferece melhor serviço de rede.

Com isto, conclui-se que o método desenvolvido é mais adequado para garantir que os dispositivos ligados à *Internet* através de redes *Wi-Fi*, estejam ligados ao AP que providencie o melhor serviço de rede, mesmo em casos de mobilidade.

5.2 Trabalho Futuro

A aplicação desenvolvida atinge o objetivo principal desta dissertação. No entanto é necessária a continuação do estudo e desenvolvimento desta aplicação de modo a que a mesma seja melhorada. Desta forma, deverá ser necessário:

- Aumentar a rapidez de todo o processo pois foi abordado em estudos anteriores que o atraso de todo o processo de *handoff* se dá na altura da descoberta de novos APs. No entanto, no trabalho realizado nesta dissertação, a otimização

do processo de *handoff* deu-se na vertente de qualidade de serviço. Implementar medidas de redução de atraso com a metodologia desta dissertação poderá ser uma boa ideia de otimizar ainda mais o processo;

- Tornar a aplicação *seamless*. Um dos problemas desta aplicação é a lentidão e até o possível corte de ligação quando estão a decorrer os pedidos de atraso de rede aos APs. Isto dá-se porque, para se conseguir os valores dos atrasos, o dispositivo precisa de se conectar e associar a cada AP, sendo que, durante a troca de APs, o dispositivo fica desconectado da rede;

- Tornar a aplicação invisível para o utilizador. Esta versão da aplicação desenvolvida possuí um *UI(User Interface)*. Um dos possíveis melhoramentos a ser implementado, seria associar a inicialização da aplicação automaticamente quando o utilizador do dispositivo pretende associar-se a uma rede *Wi-Fi*;

- Adaptar a aplicação a outras redes. A aplicação está preparada para funcionar na rede *Eduroam* dentro da UTAD. Cada entidade que usa a rede *Eduroam* pode escolher os padrões de segurança e autenticação diferentes. Por isso, para a aplicação funcionar na rede *Eduroam* dentro de outras universidades será necessário implementar um método que obtenha os parâmetros pertinentes aquando da autenticação do utilizador para ser possível tornar a aplicação utilizável em todo o ambiente *Eduroam*.

Referências bibliográficas

[1] Pedro Mestre, Alexandre Fonseca, Carlos Serodio, and Paulo Salgado. Wifi access point selector based on the network delay. In *Lecture Notes in Engineering and Computer Science, Proceedings of World Conference of Engineering 2018*, pages 409–414. IAENG, 2018. xvii, xix, xx, 47, 48, 49, 50, 51, 52, 65

[2] Dharma P Agrawal and Qing-An Zeng. *Introduction to Wireless and Mobile Systems*. Cengage learning, 2015. xix, 7

[3] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. In *Computer Communication Review*, volume 33, 01 2003. xix, 12, 13, 14

[4] Sangho Shin, Andrea G Forte, Anshuman Singh Rawat, and Henning Schulzrinne. Reducing mac layer handoff latency in ieee 802.11 wireless lans. In *Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 19–26. ACM, 2004. xix, 18, 19, 21

[5] Brian P Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T Sakai. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications magazine*, 35(9):116–126, 1997. xix, 29

[6] Eduroam. How does Eduroam work? https://www.eduroam.org/how/, 2016. Last accessed 9 november 2018. xix, 32

[7] Gray Kennedy. IOS V/S ANDROID- A COMPARATIVE ANALYSIS. https://graykennedy.wordpress.com/2017/05/15/ios-vs-android-a-comparative-analysis/, 2017. Last accessed 11 december 2018. xix, 35

[8] Galsys. Stratum Levels NTP Explained. https://www.galsys.co.uk/news/ntp-stratum-levels-explained/, 2018. Last accessed 11 december 2018. xix, 40

[9] D. Mills, U. Delaware, J. Martin Ed, J. Burbank, and W. Kasch. Network Time Protocol Version 4: Protocol and Algorithms Specification. Rfc, 2010. xix, 40, 41

[10] WiFi Alliance. How does a client roam? https://www.wi-fi.org/knowledge-center/faq/how-does-a-client-roam. Last accessed 9 november 2018. 2

[11] Paul Chandra. Handoff / handover mechanism for mobility improvement in wireless communication. In *Global Journal of Research In Engineering*, 2014. 5, 9

[12] M Ylianttila, R Pichna, J Vallstrom, J Makela, A Zahedi, P Krishnamurthy, and K Pahlavan. Handoff procedure for Heterogeneous Wireless Networks. In *Global Telecommunications Conference, 1999. GLOBECOM'99*, volume 5, pages 2783–2787. IEEE, 1999. 5

[13] Konstantinos Baltzis. *Hexagonal vs Circular Cell Shape: A Comparative Analysis and Evaluation of the Two Popular Modeling Approximations*. 04 2011. 6

[14] Nasif Ekiz, Tara Salih, Sibel Kucukoner, and Kemal Fidanboylu. An overview of handoff techniques in cellular networks. In *International journal of information technology*, volume 2, pages 132–136, 2005. 9

[15] Neha Gupta. Handoff in cellular system. In *International Journal of Technical Research and Applications*, volume 4, pages 111–113, 2016. 9, 11

[16] Ishwar Ramani and Stefan Savage. Syncscan: practical fast handoff for 802.11 infrastructure networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 675–684. IEEE, 2005. 16

[17] Minho Shin, Arunesh Mishra, and William A Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 70–83. ACM, 2004. 17

[18] Arunesh Mishra, Minho Shin, and WA Arbaush. Context caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1. IEEE, 2004. 18

[19] Pralhad Deshpande, Anand Kashyap, Chul Sung, and Samir R Das. Predictive methods for improved vehicular wifi access. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 263–276. ACM, 2009. 22

[20] Jon Froehlich and John Krumm. Route prediction from trip observations. Technical report, SAE Technical Paper, 2008. 22

[21] John Krumm. A Markov Model for Driver Turn Prediction. In *SAE 2008 World Congress*, 2016. 22

[22] GetVoIP. The History of Wifi. https://getvoip.com/history-of-wifi/, 2016. Last accessed 11 december 2018. 23

[23] CableFree. The History of WiFi: 1971 to Today. https://www.cablefree.net/wireless-technology/history-of-wifi-technology/, 2017. Last accessed 9 november 2018. 24

[24] Economist. A brief history of Wi-Fi. https://www.economist.com/technology-quarterly/2004/06/10/a-brief-history-of-wi-fi, 2004. Last accessed 11 december 2018. 24

[25] Jahanzeb Farooq and Bilal Rauf. An Overview of Wireless LAN Standards IEEE 802.11 and IEEE 802.11e. In *Department of Computing Science*, 2006. 25

[26] Moustafa A Youssef, Arunchandar Vasan, and Raymond E Miller. Specification and analysis of the dcf and pcf protocols in the 802.11 standard using systems of communicating machines. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 132–141. IEEE, 2002. 25

[27] Jiunn Deng and Ruay-Shiung Chang. A priority scheme for IEEE 802. 11 DCF access method. *IEICE transactions on communications*, 82(1):96–102, 1999. 25

[28] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000. 25

[29] Vincent Vermeer. Wireless LANs; Why IEEE 802.11 DSSS? In *WESCON/97. Conference Proceedings*, pages 172–178. IEEE, 1997. 26

[30] Robert C Dixon. *Spread Spectrum systems: with commercial applications*, volume 994. Wiley New York, 1994. 26

[31] Robert Scholtz. The origins of Spread-Spectrum Communications. *IEEE transactions on Communications*, 30(5):822–854, 1982. 26

[32] William Stallings. Data and Computer Communications. *Prentice Hall*, 2005. 29

[33] Vishal Gupta and Mukesh Kumar Rohil. Information embedding in IEEE 802.11 beacon frame. In *National Conference on Communication Technologies & its impact on Next Generation Computing CTNGC*. sn, 2012. 30

[34] J.Philip Craiger. 802.11, 802.1X, and Wireless Security. 2002. 30

[35] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and Ed Levkowetz. Extensible Authentication Protocol (EAP). Rfc, 2004. 30, 34

[36] Fanzheng Kong and Weili Huang. IEEE802.1X of protocol analysis and improvement. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 3, pages V3–282. IEEE, 2010. 31

[37] FCT. Descrição da Eduroam. https://www.eduroam.pt/pt/sobre/descricao, 2015. Last accessed 9 november 2018. 32, 45

[38] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. Rfc, 2018. 33

[39] P. Funk and S. Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). Rfc, year =2008. 33

[40] G. Zorn. Microsoft PPP CHAP Extensions, Version 2. Rfc, 2000. 34

[41] A. Ashwin, D. Simon, G. Zorn, and S. Josefsson. Protected EAP Protocol (PEAP). Rfc, 2003. 34

[42] N. Cam-Winget, D. McGrew, J. Salowey, and H. Zhou. The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). Rfc, 2007. 34

[43] John Callaham. The history of Android OS: its name, origin and more. https://www.androidauthority.com/history-android-os-name-789433/, 2018. Last accessed 11 december 2018. 35

[44] Android Developers. android.content. https://developer.android.com/reference/android/content/package-summary, 2018. Last accessed 11 december 2018. 36

[45] Android Developers. android.net.wifi. https://developer.android.com/reference/android/net/wifi/package-summary, 2018. Last accessed 11 december 2018. 36

[46] Android Developers. AsynTask. https://developer.android.com/reference/android/os/AsyncTask, 2018. Last accessed 11 december 2018. 38

[47] David L. Mills. Internet Time Synchronization: The Network Time Protocol. In *IEEE Transactions on Communications*, 1991. 39