

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Aplicação Móvel para Melhorar a Qualidade do Acesso WiFi em Localizações Saturadas

Pedro Medeiros Sousa

PARA APRECIACÃO POR JÚRI

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Dr. Ricardo Morla

Co-orientador: Prof. Dra. Ana Aguiar

29 de Junho de 2015

Resumo

Na última década, as redes WiFi expandiram-se rapidamente e foram alvo de grande desenvolvimento tecnológico, pelo que substituíram grande parte das infraestruturas de rede com fios. Desta forma, face ao elevado crescimento de dispositivos dotados da tecnologia WiFi, as redes WiFi tornaram-se cada vez mais saturadas, evidenciando-se anomalias na ligação e degradação da experiência do utilizador.

As redes têm, muitas vezes, elevada densidade espacial de pontos de acesso instalados sem planeamento prévio. Além disso, há locais onde há aglomeração, pontual ou regular, de utilizadores. Todos estes fatores causam problemas de diversa natureza na qualidade de serviço, pelo que tem havido cada vez mais interesse em estudar e desenvolver técnicas para diagnosticar e melhorar o desempenho das redes WiFi. Esta dissertação enquadra-se no âmbito do diagnóstico e, consiste no desenvolvimento de um sistema capaz de recolher estatísticas de vários dispositivos móveis com a finalidade de informar o utilizador sobre a qualidade de acesso à rede num determinado local comparativamente com outros utilizadores e com outros locais.

O sistema desenvolvido seguiu a abordagem Cliente-Servidor. A aplicação cliente foi desenvolvida para a plataforma Android e consistiu em dois módulos, um deles é responsável pela recolha de informação, mais especificamente a velocidade real da ligação e a latência da mesma, enquanto que o outro é responsável pela aquisição de *feedback* a partir do servidor e exibição do mesmo ao utilizador. O servidor é responsável pelo armazenamento de todas as estatísticas enviadas pelos utilizadores e por gerar as mensagens de *feedback* a retornar aos utilizadores.

Por fim, a validação do sistema foi efetuada através da colocação de 4 telemóveis no mesmo local e em locais diferentes. A análise dos resultados permitiu detetar padrões de desempenho distintos em determinados dispositivos, quer no mesmo local, quer em locais diferentes. Para o mesmo local, o sistema foi capaz de distinguir o desempenho de todos os telemóveis, tendo-se evidenciado, pela negativa, o telemóvel colocado junto a uma janela, o que permite a um utilizador perceber que o seu dispositivo está na origem do problema. Quanto aos locais distintos, a experiência efetuada permitiu caracterizar o desempenho das várias localizações e, desta forma, detetar locais com melhor qualidade de serviço, o que permite ao utilizador mudar de lugar caso ache necessário.

Abstract

Over the last decade, WiFi networks have been rapidly expanding and have been subject of a great technological development, and therefore have replaced great part of the wired network infrastructures. So, due to the fast growth of the amount of devices capable of using the WiFi technology, WiFi networks became increasingly saturated, which leads to problems in the connection and the user's experience degradation.

Networks often have a high density of access points installed without previous planning. Besides, there are places where the agglomeration of users occurs, punctual or regular. All these factors directly affect the quality of the service. To counter it, there are more and more studies and techniques being developed to diagnose and improve the performance of the WiFi networks. This dissertation aims at the diagnosis part and consists in the development of a system capable of receiving statistics of several mobile devices, with the purpose of informing the user about the quality of the network's access in a specific spot compared to several different users in different locations.

The system developed followed the Client-Server approach. The client application was developed to the Android Platform and consisted of two modules. One responsible for the information gathering, specifically the network connection's throughput and its latency, while the other one is responsible for receiving the server's feedback and for the information display to the user.

Finally, the system validation consisted of 4 cellphones placed in the same location and then in different places. The analysis of the results showed different performance patterns in some devices, in the same location or in different ones. For the same place, the system was capable of distinguishing the performance of all phones, being negatively highlighted the phone placed near a window, which allows the user to understand that his device is the source of the problem. As to the different locations, the experience conducted allowed the performance characterization of the several locations and, consequently, the detection of places with better quality of service, which allows the user to change his place, if he considers it necessary.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais, José e Maria Sousa, e irmão, José Sousa, por me terem apoiado incondicionalmente e oferecido todas as condições necessárias para a conclusão do meu percurso académico.

Agradeço à minha namorada, Rita Cerqueira, por ter estado presente em todos os momentos cruciais, por toda a paciência, compreensão e por ter sido um exemplo constante de ambição e determinação.

Agradeço também a todos os meus amigos que direta ou indiretamente me ajudaram ao longo desta etapa.

Por fim, agradeço ao Prof. Dr. Ricardo Morla e à Prof. Dra. Ana Aguiar por todo o empenho, disponibilidade, orientação e apoio oferecido durante a realização desta dissertação.

Pedro Medeiros Sousa

*“If the Internet teaches us anything,
it is that great value comes from leaving core resources in a commons,
where they’re free for people to build upon as they see fit.”*

Lawrence Lessig

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Estrutura do Documento	2
2	IEEE 802.11	5
2.1	Introdução	5
2.2	Arquitetura Lógica	6
2.3	Modos de Operação	7
2.4	Acesso ao Meio	8
2.5	Estabelecimento de Ligação	8
2.6	Problemas de Conetividade	9
3	Revisão da Literatura	11
3.1	Mecanismos Físicos	11
3.2	Mecanismos Protocolares	11
3.3	Mecanismos Aplicacionais	12
3.4	Métricas	13
3.5	Indicadores em Android	13
3.6	Discussão	15
4	Solução Proposta	17
4.1	Aplicação Prática	17
4.2	Arquitetura	18
5	Implementação	23
5.1	Comunicação	23
5.2	Cliente	23
5.3	Servidor	27
6	Análise de Resultados	33
6.1	Metodologia	33
6.2	Casos de Uso	35
6.3	Testes	38
7	Conclusões	45
7.1	Sumário	45
7.2	Funcionalidades e Resultados	45
7.3	Trabalho Futuro	46

Referências

47

Lista de Figuras

2.1	Esquemático de uma IBSS (modo Ad-Hoc)	7
2.2	Esquemático de uma BSS e de uma ESS	7
4.1	Padrões distintos de desempenho na mesma rede	18
4.2	Diagrama do funcionamento do sistema	18
5.1	Exemplo de trama enviada para o servidor em formato JSON	26
5.2	Exemplo de resposta do servidor em formato JSON	27
6.1	Interface para recolha de dados	33
6.2	Interface para aquisição de <i>feedback</i>	34
6.3	Exemplo de <i>feedback</i> recolhido da experiência 1	35
6.4	Exemplo de <i>feedback</i> recolhido da experiência 2	37
6.5	Resultado da experiência 1 para o telemóvel F1	39
6.6	Resultado da experiência 1 para o telemóvel 2E	39
6.7	Resultado da experiência 1 para o telemóvel 3B	39
6.8	Resultado da experiência 1 para o telemóvel 0A	40
6.9	Resultado da experiência 2 para o local "I103"	41
6.10	Resultado da experiência 2 para o local "Biblioteca piso 0"	42
6.11	Resultado da experiência 2 para o local "Biblioteca piso 6"	42
6.12	Resultado da experiência 2 para o local "Laboratório de redes"	42

Lista de Tabelas

2.1	Comparação das normas IEEE 802.11	6
2.2	Resumo das causas de problemas de conectividade nas redes WiFi	10
3.1	Exemplo de informação acessível pelas APIs de rede do Android	15
4.1	Cenários de obtenção do <i>feedback</i>	19
4.2	Resultado do <i>feedback</i> para os cenários 1 a 9	20
4.3	Mensagens de <i>feedback</i> enviadas para o utilizador	20
6.1	Dados da experiência 1	38
6.2	Dados da experiência 2	41

Abreviaturas e Símbolos

AP	<i>Access Point</i>
API	<i>Application Programming Interface</i>
BSS	<i>Basic Service Set</i>
BSSID	<i>Basic Service Set Identification</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
DCF	<i>Distributed Coordination Function</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Domain Name System</i>
DS	<i>Distribution Service</i>
ESS	<i>Extended Service Set</i>
HCF	<i>Hybrid Coordination Function</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBSS	<i>Independent Basic Service Set</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
JSON	<i>JavaScript Object Notation</i>
LLC	<i>Logic Link Layer</i>
LQI	<i>Link Quality Information</i>
MAC	<i>Media Access Control</i>
MIMO	<i>Multiple-Input and Multiple-Output</i>
NFC	<i>Near Field Communication</i>
OS	<i>Operating System</i>
PCF	<i>Point Coordination Function</i>
QoS	<i>Quality of Service</i>
REST	<i>Representational State Transfer</i>
RSSI	<i>Received Signal Strength Indicator</i>
RTT	<i>Round-Trip Time</i>
SINR	<i>Signal to Interference plus Noise Ratio</i>
SSID	<i>Service Set Identifier</i>
STA	<i>Station</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>Datagram Protocol</i>

Capítulo 1

Introdução

Na última década, o desenvolvimento tecnológico das comunicações sem fios fez com que o desempenho das mesmas fosse muito semelhante ao das comunicações cabladas em termos de largura de banda, latência, segurança, e etc. Como consequência, as tecnologias sem fios como o WiFi, redes celulares, Bluetooth, ZigBee e NFC (*Near Field Communication*) foram amplamente adotadas para substituir e complementar as infraestruturas de comunicações com fios existentes. Desta forma, com o elevado crescimento das redes sem fios, o WiFi tornou-se na tecnologia sem fios predominante [1] não só para uso doméstico mas também em ambiente empresarial, universitário, hospitalar e nos mais variados locais públicos. Os principais motivos para a utilização desta última tecnologia em grande escala, estão relacionados com o aumento da mobilidade, largura de banda elevada, baixo custo e redução do tempo de instalação. Por exemplo, é esperado que o número de *hotspots* WiFi aumente de 1.3 em 2011 para 5.8 milhões em 2015 [2].

1.1 Motivação

Dados da indústria revelam que a procura por dispositivos WiFi está a aumentar mais rapidamente do que a capacidade suportada pelas redes sem fios [3] e essa situação continuará a verificar-se, pelo menos, por algum tempo. A razão reside no facto de, atualmente, existir uma grande utilização de dispositivos moveis tal como computadores portáteis, *smartphones*, *tablets*, consolas de jogos, e etc. O poder de processamento, capacidade de memória, interface com o utilizador e interfaces de comunicação (WiFi, Bluetooth, redes celulares e NFC) destes dispositivos desenvolveram-se muito, por conseguinte os utilizadores desejam melhor conectividade e alta qualidade de serviço em qualquer lugar e em qualquer momento [4].

Apesar do conjunto de benefícios descritos anteriormente, o constante aumento dos dispositivos móveis equipados com WiFi acarreta também novos desafios. As aplicações usadas pelos utilizadores dos mesmos estão na expectativa e algumas têm mesmo o requisito de utilizar alta largura de banda e elevada qualidade de serviço. Isso inclui o download de ficheiros e dados em massa, tal como conteúdos multimédia, áudio e vídeo de alta definição, acesso a *clouds*, etc [4]

[3]. Desta forma, o principal desafio consiste em desenvolver técnicas que contribuam e melhorem a capacidade e a conectividade das redes WiFi de forma ubíqua.

O facto das redes sem fios WiFi serem utilizadas em grande escala, aliado à falta de planeamento na instalação e na frequência de operação, fez com que surgissem ambientes de alta densidade, com uma grande quantidade de dispositivos a aceder ao meio. Dado que o meio de comunicação utilizado é partilhado com todos os dispositivos que estejam a operar na mesma área de cobertura geográfica e canal de comunicação, surgem problemas de conectividade e diversas interferências que reduzem a qualidade da experiência do utilizador [1] [5].

Por conseguinte, existe uma enorme preocupação em reduzir a interferência e melhorar a conectividade, de forma a aumentar a qualidade de serviço prestada e tornar a experiência do utilizador mais positiva de acordo com as suas expectativas. Nesse sentido, surgiram estudos e desenvolveram-se métodos na área do controlo da topologia, na atribuição de recursos rádio, na camada de controlo do acesso ao meio, nos protocolos de encaminhamento de dados e mesmo ao nível da aplicação.

1.2 Objetivos

Devido à realidade descrita anteriormente, muitas soluções foram propostas [4] [6] [7] [8] [9] [10] [11] [12] [13], e algumas exigem uma mudança no controlo do acesso ao meio 802.11. No entanto, as soluções implementadas ao nível do *firmware*, nem sempre são compatíveis com o standard, pelo que não são comercialmente viáveis.

Esta tese pretende explorar a perspetiva da aplicação que, apesar de ser mais arriscada por ter um nível de controlo inferior, tem mais potencial para ser desenvolvida visto que não exige mudanças no *firmware*, apenas tem o requisito de instalar uma aplicação num dispositivo móvel. Assim sendo, nesta dissertação é pretendido desenvolver um sistema de diagnóstico, através de um servidor e de uma aplicação móvel em Android, com o objetivo de melhorar a qualidade do acesso à rede WiFi e a experiência do utilizador. Nesse sentido, a aplicação é responsável por avaliar o estado da ligação de rede com base num conjunto de métricas adequadas, de forma a recolher informação relevante capaz de caracterizar o desempenho da rede, reportar o resultado desse diagnóstico para um servidor e receber do mesmo estatísticas de outros utilizadores.

Desta forma, os utilizadores conseguirão obter um *feedback* e perceber se o seu equipamento está na origem do problema ou se já ocorreram outras anomalias reportadas para o servidor por parte de outros utilizadores. Assim sendo, esta abordagem traz benefícios tanto para os utilizadores finais como para os administradores de rede que, por este meio, conseguem ter noção das dificuldades técnicas em determinados locais.

1.3 Estrutura do Documento

O presente capítulo teve como finalidade apresentar o contexto no qual se insere a dissertação, descrever os aspetos motivadores em relação à escolha do tema, através de uma breve descrição dos

problemas associados ao mesmo, e abordar os objetivos propostos para a dissertação. No próximo capítulo é descrito o funcionamento básico das redes sem fios WiFi e os restantes 5 capítulos desde documento visam a apresentação e discussão dos estudos já efetuados e relacionados com o tema em questão, a apresentação da solução proposta e respetiva arquitetura, a descrição da implementação da solução, a análise dos resultados obtidos com a implementação efetuada e, por último, são apresentadas as conclusões do trabalho.

Capítulo 2

IEEE 802.11

Este capítulo tem como objetivo apresentar uma breve introdução à norma IEEE 802.11, descrever a sua arquitetura, o mecanismo de acesso ao meio, os modos de funcionamento, as etapas para estabelecer uma conexão e abordar as possíveis anomalias associadas a uma rede sem fios WiFi.

2.1 Introdução

Em 1997, o IEEE (*Institute of Electrical and Electronics Engineers*) definiu e aprovou a norma IEEE 802.11, genericamente conhecida por WiFi. Este método de comunicação sem fios, permitia taxas de transmissão de 1 e 2 Mbps na gama de frequência de 2.4 GHz. No entanto, não surgiu grande entusiasmo em utilizar esta nova tecnologia dado que, na altura, já existiam sistemas de comunicação sem fios a operar a 10 Mbps.

Em 1999, foram aprovadas duas revisões da norma em questão, a IEEE 802.11a e a IEEE 802.11b, com frequência de operação de 5 e 2.4 GHz e taxa de transmissão de 54 e 11 Mbps, respetivamente. Contudo, devido ao baixo custo e disponibilização para venda antecipada, foi o standard 802.11b que conquistou uma grande parte da quota do mercado. Já em 2003, surgiu a norma 802.11g que veio permitir uma taxa de transmissão máxima de 54 Mbps, na gama de frequência de 2.4 GHz, já anteriormente suportada pela versão 802.11a.

Com o crescente aparecimento de dispositivos WiFi e respetiva substituição pelos equipamentos com fios, surgiram vários requisitos de qualidade de serviço (QoS - *Quality of Service*), pelo que em 2005 foi aprovada a emenda IEEE 802.11e, a qual inclui, além de outras inovações, diferentes níveis de QoS, suporte para a transmissão de voz e de vídeo.

Já em 2009, de forma a corresponder com as expectativas dos utilizadores em usufruir do mesmo desempenho das redes com fios, foi aprovada a norma IEEE 802.11n, a qual trouxe grandes melhorias na taxa de transmissão, permitindo velocidades até 600 Mbps com recurso a várias antenas e *streams* de dados (MIMO - *Multiple-Input and Multiple-Output*). Os dispositivos que implementam esta norma apresentam também uma característica híbrida, isto porque estão aptos a operar em ambas as gamas de frequência (2.4 e 5 GHz) [14].

O progresso tecnológico nas redes sem fios WiFi, representado na Tabela 2.1, não abranda e enquanto é comercializada a última versão disponibilizada para o mercado, já outras se encontram em desenvolvimento. Cada vez que é lançada uma revisão da norma IEEE 802.11, os melhoramentos são visíveis e a compatibilidade com dispositivos mais antigos é sempre tida em consideração. Atualmente, são comercializados equipamentos compatíveis com a norma IEEE 802.11ac que operam a 5 GHz (não permitida em todos os países [15]), cujas inovações centram-se no menor consumo de energia, na maior cobertura de rede, em 3 vezes mais capacidade de utilizadores em simultâneo e 6 vezes maior taxa de transmissão do que a versão 802.11n [16].

Tabela 2.1: Comparação das normas IEEE 802.11

Versão da Norma	Data de Aprovação	Velocidade Máxima	Frequência de Operação
802.11	1997	1 - 2 Mbps	2.4 GHz
802.11b	1999	11 Mbps	2.4 GHz
802.11a	1999	54 Mbps	5 GHz
802.11g	2003	54 Mbps	2.4 GHz
802.11n	2009	72 - 600 Mbps	2.4 e 5 GHz
802.11ac	2014	433 - 1300 Mbps	5 GHz

2.2 Arquitetura Lógica

Numa rede WiFi, qualquer dispositivo dotado da tecnologia 802.11, tal como computadores e dispositivos moveis, é referenciado como uma estação. Quando duas ou mais estações (STA - *Station*) tentam comunicar entre si, formam uma BSS (*Basic Service Set*), que é nada mais nada menos do que a arquitetura básica de uma rede IEEE 802.11. As BSSs podem ser classificadas como independentes ou infraestruturadas.

As BSSs independentes (IBSS - *Independent Basic Service Set*), também conhecidas como redes Ad-Hoc, são normalmente de carácter temporário e de tamanho reduzido. Implicam que todas as estações comuniquem diretamente entre si, como é explicitado na Figura 2.1, daí a razão de serem compostas por um pequeno grupo de estações. Não existe um ponto de acesso (AP - *Access Point*) e nenhuma estação dá permissão para iniciar uma comunicação.

Quando as BSS estão interligadas, a rede passa a ser considerada infraestruturada. A interligação entre duas ou mais BSS é efetuada por um sistema de distribuição (DS - *Distribution System*) que permite aumentar a cobertura de rede e pode ser implementado com o recurso a cabos ou ligação WiFi entre os APs. Neste sentido, o AP é considerado uma estação, com um endereço próprio, e serve de elo de ligação para o DS, pelo que sempre que qualquer estação pretenda comunicar, todos os dados são transmitidos para o AP e posteriormente encaminhados pelo DS até ao seu destino [14] [17].

Ao utilizar BSSs e DSs para criar redes de maior dimensão e mais complexas, aumenta-se o nível da hierarquia e surgem as ESSs (*Extended Service Set*). Toda a rede formada pela ESS, Figura 2.2, é vista como uma IDSS para a camada de controlo lógico da ligação (LLC - *Logical*

Link Control). Isso implica que todas as estações pertencentes à mesma ESS podem comunicar e mover-se entre diferentes BSSs de forma transparente para a LLC [17].

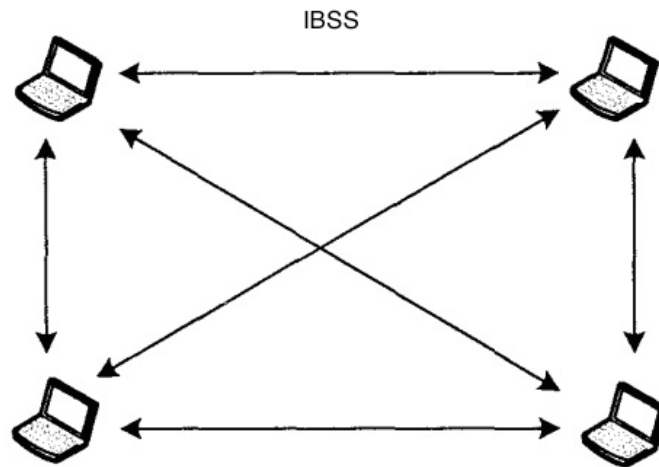


Figura 2.1: Esquemático de uma IBSS (modo Ad-Hoc)
Fonte: [18]

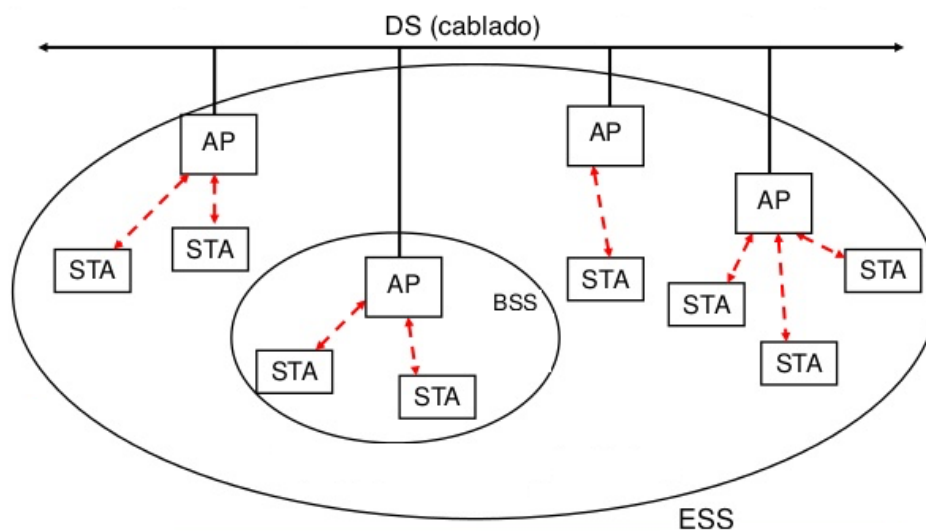


Figura 2.2: Esquemático de uma BSS e de uma ESS
Fonte: [19]

2.3 Modos de Operação

Os diversos dispositivos dotados da tecnologia WiFi possuem diferentes modos de funcionamento tendo em conta o papel que desempenham na rede e a tipologia da mesma, são eles: *Ad-Hoc*, *Managed*, *Master*, *Repeater*, *Secondary* e *Monitor*.

Os modos *master*, *repeater* e *secondary* são utilizados por dispositivos intermédios e pontos de acesso à rede. O modo *master* implica que o equipamento funcionará como um AP, o *repeater* reencaminha os pacotes de dados entre outros nós da rede e o modo *secondary* tem como objetivo configurar um nó da rede para servir como um backup de um AP ou de um repetidor.

Quanto aos restantes modos de funcionamento, o ad-hoc permite um equipamento criar uma rede sem AP e com um único BSS, o *managed* é o modo pré-definido de muitos equipamentos comerciais tal como computadores e *smartphones* e permite que esse mesmo equipamento funcione como cliente de uma determinada rede composta por um ou mais APs.

Por último, com especial destaque é descrito o modo *monitor*, visto que este modo de funcionamento não está disponível em todas as placas de rede clientes, depende do fabricante e do modelo, no entanto está presente numa grande quantidade de computadores pessoais e permite que um dispositivo não associado a qualquer rede WiFi possa monitorizar de forma passiva, numa determinada frequência, todos os pacotes de dados ao seu alcance [20].

2.4 Acesso ao Meio

O controlo de acesso ao meio é efetuado ao nível da camada MAC (*Medium Access Control*) e existem três funções de coordenação diferentes: a DCF (*Distributed Coordination Function*), PCF (*Point Coordination Function*) e HCF (*Hybrid Coordination Function*).

A DCF é de uso obrigatório, usada em redes ad-hoc e infraestruturadas, e utiliza o mecanismo de CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), que basicamente significa que uma estação tem de escutar e esperar que o meio de comunicação esteja livre para enviar dados e não existe suporte para deteção de colisões.

A PCF é de uso opcional, usa serviços DCF, é direcionada para redes infraestruturadas, e utiliza um mecanismo de *polling* centralizado no AP em que o acesso à rede é sem contenção.

O mais recente mecanismo de coordenação é o HCF, também de uso opcional, que acrescenta funcionalidades ao DCF e PCF por forma a melhorar o QoS, é um método híbrido dado que é baseado em *polling* e em contenção. A principal diferença é a utilização de filas com múltiplas prioridades por estação com a finalidade de priorizar o tráfego com maior importância [21] [22].

2.5 Estabelecimento de Ligação

Numa rede infraestruturada, antes de qualquer estação efetuar a ligação à rede, o estado da mesma é considerado como não autenticada e não associada. O processo de ligação à rede WiFi por parte das estações clientes até estarem aptas a transmitir dados é realizado em várias etapas: aquisição de *beacons*, *probing*, autenticação 802.11 de baixo nível, associação, autenticação WPA/WPA2/802.1X e aquisição de endereço IP (*Internet Protocol*).

Os pontos de acesso usam *frames*, denominados *beacons*, enviados em *broadcast* periodicamente para permitirem que os clientes tenham conhecimentos das redes disponíveis numa determinada área. Por este meio os clientes podem escolher que rede e ponto de acesso usar.

Quando um utilizador pretende estabelecer uma ligação a uma determinada rede, o cliente envia *probe requests* especificando o SSID (*Service Set Identifier*) em questão com a finalidade de negociar as taxas de transmissão, o tipo de encriptação requerida pelo AP e as normas 802.11 suportadas por ambos. No caso de nenhum SSID for especificado, o *probe request* poderá ser enviado em *broadcast* e utilizado para obter uma lista das redes em redor e as suas características (o que implica que o AP esteja a fazer *broadcast* do seu SSID).

Após a escolha da rede e verificada a compatibilidade da mesma (método de encriptação por exemplo), a estação cliente envia um pedido de autenticação 802.11 de baixo nível para um ou mais APs da rede pretendida. Este tipo de autenticação em nada tem a ver com os mecanismos WPA2 ou 802.1X usados atualmente, originalmente foi direcionado para encriptação WEP, no entanto caiu em desuso devido às suas vulnerabilidades.

Neste ponto, a estação encontra-se autenticada em baixo nível mas não associada. Uma estação pode estar autenticada em vários APs, contudo só pode estar associada a um AP de cada vez. Após a escolha do AP a estação envia um pedido de associação que contém os tipos de encriptação escolhidos e outras características para estabelecer a ligação de rede.

Após a confirmação do AP, a estação fica associada. No entanto, se for necessário autenticação WPA/WPA2 ou 802.1X, é necessário comunicar com o AP ou com um servidor de autenticação (através do AP) de forma a validar as credenciais de acesso à rede.

Uma vez concluído o processo de autenticação, caso a configuração de endereço IP não seja manual, é realizado um pedido de DHCP (*Dynamic Host Configuration Protocol*) em *broadcast* para adquirir um endereço IP e, após resposta positiva, a estação está apta a iniciar a transferência de dados [23] [24] [25].

2.6 Problemas de Conetividade

Com o crescente aumento das redes sem fios IEEE 802.11, também aumentaram os problemas de conetividade. Estes podem estar diretamente relacionados com os clientes, com a infraestrutura e administração da rede ou com o meio envolvente à mesma. Na Tabela 2.2 estão resumidas as causas desses problemas, que são descritos nos parágrafos seguintes.

Quanto ao equipamento do cliente, o próprio hardware e os controladores podem conter anomalias que levam a uma desconexão sem razão aparente, podem existir barreiras na comunicação tal como uma *firewall* e, no caso de configuração estática do endereçamento IP e DNS (*Domain Name Service*), podem surgir endereços duplicados, anomalias no encaminhamento de dados por rotas mal definidas e falha na obtenção de pedidos DNS. Muitas vezes o cliente também poderá ter as credenciais de acesso à rede ou mesmo os mecanismos de segurança e encriptação da mesma configurados incorretamente.

No que diz respeito à infraestrutura e administração da rede, a colocação dos APs pode contribuir para a obtenção de um sinal de rede fraco devido à presença de uma obstrução física entre o AP e o cliente. Isto acontece quando os administradores de rede não têm em atenção todos os locais em que os clientes pretendem usufruir de acesso à rede. A proximidade entre os APs também

é uma razão válida, assim como a configuração automática ou aleatória da frequência de operação dos APs em canais sobrepostos ou adjacentes entre si. O descuido no dimensionamento da rede associado à grande quantidade de utilizadores pode não só trazer problemas de conectividade, dado que o *slot* temporal disponível para cada utilizador tende a diminuir com o aumento dos pedidos de transmissão (o que provoca a diminuição do *throughput*), mas também ser a causa de muitas falhas de obtenção de endereço IP nos dispositivos clientes por DHCP.

Por outro lado, ao redor de muitas redes WiFi surgem problemas de conectividade relacionados com interferências provocadas por APs de outras redes cujos canais de operação estão sobrepostos ou em frequências próximas [26]. Não só esta última razão é válida como também se verifica o uso da frequência de 2.4 GHz em muitos outros equipamentos eletrônicos e domésticos, como é o caso dos micro-ondas e dos telefones sem fios da rede fixa [1]. A grande afluência de diferentes redes WiFi, APs e utilizadores num mesmo espaço físico cria ambientes de elevada densidade, provocando desta forma interferências na conectividade e verificam-se sistematicamente falhas nos processos de associação e autenticação dos clientes na rede [9].

Tabela 2.2: Resumo das causas de problemas de conectividade nas redes WiFi

Dispositivos Clientes	Hardware e controladores Firewall IP e DNS manuais Credenciais de acesso Mecanismos de segurança e encriptação
Infraestrutura e Administração de rede	Colocação dos APs e proximidade entre eles Frequência de operação dos APS Dimensionamento da rede/DHCP
Interferências	APs e clientes de outras redes Equipamento <i>non-WiFi</i> @ 2.4 GHz na proximidade Ambientes de alta densidade

Após esta breve abordagem à norma IEEE 802.11, são apresentados, no próximo capítulo, alguns estudos com a intenção de melhorar e diagnosticar a conectividade das redes sem fios WiFi.

Capítulo 3

Revisão da Literatura

No capítulo anterior foram apresentados os modos de funcionamento assim como os diversos problemas de conectividade associados às redes IEEE 802.11. Por conseguinte, evidenciou-se a necessidade da existência de investigações nesta área, com a finalidade de melhorar a qualidade do serviço de rede prestado aos utilizadores. Nesse sentido, este capítulo tem como objetivo apresentar e discutir as propostas apresentadas pela comunidade científica para solucionar o problema. As propostas que foram surgindo focam sobretudo aspetos relacionados com o ponto de vista da camada física, protocolar e aplicacional, pelo que optou-se por distinguir, neste capítulo, estes 3 tipos de mecanismos. Também são abordadas as várias métricas utilizadas na análise do desempenho da rede, são apresentados indicadores que podem ser recolhidos no sistema Android e a sua aplicabilidade para o sistema. A última secção deste capítulo consiste numa breve discussão/resumo dos temas abordados nas secções anteriores.

3.1 Mecanismos Físicos

Na tentativa de melhorar a qualidade de ligação da rede e reduzir as interferências a nível da camada física, surgiram antenas com transmissão direcional [4] e utilizaram-se antenas distribuídas [12] para melhorar a cobertura. Com a intenção de descobrir formas de interferências externas à rede, que de certa forma possam estar a condicionar o desempenho da mesma, surgiu a comercialização de sistemas de *hardware* com a capacidade de detetar interferências provocadas por dispositivos non-WiFi, tal como o *Spectrum XT*, o *AirMaestro* e o *CleanAir* [27]. Também foram realizadas investigações relacionadas com a deteção de dispositivos non-WiFi a trabalhar na gama de frequências de 2.4 GHz através da utilização de placas de rede WiFi [11] [27].

3.2 Mecanismos Protocolares

No sentido de melhorar a conectividade e o desempenho das redes sem fios IEEE 802.11, surgiu muito interesse em estudar, alterar e testar diversas alterações protocolares [28] no encaminhamento de dados [8] [11], na atribuição dinâmica de canais de funcionamento aos APs [7] [8]

[11] [12], no uso de múltiplos canais de transmissão [4] [12], no controlo dinâmico da potência de transmissão [8] [12] [13], na utilização de mecanismos de distribuição da carga/tráfego pelos APs [12], na modificação da *stack* de rede mac80211 do *kernel* do Linux [11], na modificação do modo de acesso ao meio e na alteração de parâmetros da camada MAC [4] [10] [11] [12] [13]. Por exemplo, a atribuição dinâmica de canais de funcionamento aos APs permitiu aumentar a qualidade de serviço por diminuir a interferência mutua entre APS em redes sem planeamento, e a modificação da *stack* de rede do Linux permitiu diminuir o *overhead* e aumentar o *throughput*.

3.3 Mecanismos Aplicacionais

De forma a resolver problemas de conectividade nas estações que pretendem aceder à rede e a dar *feedback* aos utilizadores, desde cedo foram implementadas ferramentas de diagnóstico da ligação a nível do sistema operativo (OS - *Operating System*). Qualquer computador ou dispositivo móvel com tecnologia WiFi está equipado com ferramentas capazes de efetuar *debug* durante a ligação à rede e informar o utilizador de qualquer problema encontrado, seja ele problema de associação ao AP, autenticação ou de obtenção de endereço IP.

No entanto, por vezes a qualidade do sinal da rede é elevado e existem problemas de conectividade alheios ao cliente, tal como baixo *throughput*, latência elevada e falhas ou quedas na ligação. Para analisar essas situações existem aplicações, como por exemplo o *Wireshark*, que a partir do modo *monitor* das placas WiFi têm a capacidade de analisar as redes e os respetivos APs na vizinhança através do *sniffing* do tráfego. Deste modo, é possível obter uma estimativa da quantidade de utilizadores ligados a uma rede através da análise do endereço de origem das tramas e da quantidade de tramas de dados capturadas. A única barreira é o facto deste modo de funcionamento não ser compatível com todos os dispositivos WiFi e exigir permissões de administrador, fatores que condicionam a sua utilização.

Na plataforma móvel Android já foram efetuados estudos direcionados para o diagnóstico e deteção de erros de ligação, para a análise da qualidade de acesso à rede e de identificação da localização dos dispositivos. O estudo efetuado para a deteção de problemas na ligação WiFi do *smartphone* [9] foi baseado na análise de *logs* gerados internamente pelo Android e por mensagens de erro despoletadas pela *API (Application Programming Interface) de wireless do Android*, no entanto carece da comparação com os resultados de outros utilizadores, uma vez que se foca apenas no diagnóstico de um determinado dispositivo em particular. Os estudos com o objetivo de caracterizar a qualidade de acesso à rede não se focaram na deteção de erros no dispositivo, mas na medição de estatísticas (tais como o *throughput*, latência e qualidade do sinal) para traçar um perfil da qualidade da ligação [29] [30] [31]. Um desses estudos foi um pouco diferente no sentido em que a aplicação desenvolvida utilizou uma placa WiFi externa ligada ao telemóvel por USB com a capacidade de detetar dispositivos non-WiFi [32]. Também foram realizados estudos para a determinação da localização de dispositivos WiFi, os quais se basearam essencialmente nos identificadores dos pontos de acesso e nos valores de RSSI [33] [34] [35] [36] [37].

3.4 Métricas

Esta secção tem como finalidade mencionar de forma geral as métricas utilizadas pelos diversos métodos que contribuíram para ter noção da qualidade de ligação da rede e do desempenho da mesma.

O RSSI (*Radio Signal Strength Indicator*) e o SINR (*Signal to Interference plus Noise Ratio*) são constantemente utilizados como métricas para quantificar a qualidade da ligação WiFi nos dispositivos clientes [38]. O RSSI traduz-se na força do sinal transmitido pelo AP e o SINR está relacionado com o nível de ruído em background (quando nenhum pacote é transmitido) que, quanto mais alto for, significa que o meio apresenta menos ruído[20].

Um elevado número de APs nas proximidades (da mesma ou de outras redes), assim como uma grande quantidade de APs na mesma frequência ou em frequências adjacentes contribui negativamente para as interferências na rede. Deste modo, são métricas importantes a utilizar nos clientes para classificar o ambiente em redor dos mesmos [7] [12].

O número de retransmissões está relacionado com o número de pacotes que o canal da rede não conseguiu entregar [20], portanto é um fator que permite avaliar o desempenho da rede, assim como a velocidade da ligação [10] [13] [39] e o tempo de resposta da mesma [29] [30], que são fatores muito determinantes da qualidade de serviço de rede. No que toca à velocidade da ligação, a taxa de transmissão entre o AP e o cliente é negociada tendo em conta a qualidade do sinal de receção no cliente. O valor real obtido (*throughput*) nunca é semelhante ao valor da taxa de transmissão, devido ao *overhead* dos pacotes [4] [6] e também por sofrer alterações negativas em ambientes de elevada densidade, visto que o tempo atribuído a cada estação para transmitir é reduzido e podem ocorrer erros. Por conseguinte, ambos os fatores devem ser tidos em consideração na qualidade do acesso à rede pelo dispositivo cliente, pelo que são métricas importantes e bastante usadas para analisar o desempenho da ligação.

3.5 Indicadores em Android

No sistema operativo Android é possível recolher várias estatísticas relacionadas com o estado e com a configuração da rede WiFi. Esta subsecção aborda esses indicadores e indica a sua aplicabilidade neste projeto.

Através da análise do *log* do sistema, com a ferramenta *logcat*, é possível detetar anomalias na ligação de rede, a nível do *kernel*, tais como:

- "Association timed out"
- "Authentication timed out"
- "Connection to AP lost"
- "DHCPv4 request timed out"
- "Non-local response from host"

Contudo, a visualização dos *logs* do sistema está muito restrita ao utilizador comum e a visualização de mensagens relacionadas com o estado da rede apenas está disponível com permissões de administrador.

Através do utilitário *wpa_cli* é possível diagnosticar em tempo real a ligação de rede WiFi e obter todas as mensagens de baixo nível relativas à associação ao ponto de acesso e quaisquer erros que possam ocorrer. Essas mensagens especificam um determinado *código*, o qual se traduz na razão da falha de autenticação ou na perda de conectividade com o ponto de acesso. No entanto este utilitário também necessita de permissões de administrador.

Os utilitários *ifconfig* e *iwconfig* permitem obter muita informação, sem permissões de administrador, relativa ao estado da rede nomeadamente:

- *Standard* da placa WiFi
- Modo de operação
- Frequência
- Ponto de acesso associado
- Taxa de transmissão
- Potência de transmissão
- Retransmissões a nível do MAC
- Qualidade do sinal
- Nível de ruído em *background*
- Endereço IP
- Endereço físico
- Quantidade de pacotes enviados/recebidos
- Quantidade de bytes enviados/recebidos

No entanto, nem todos os dispositivos suportam a totalidade dos campos descritos anteriormente, tal como a quantidade de retransmissões a nível do MAC ou o nível de ruído, pelo que a sua aplicabilidade neste sistema fica desde logo descartada.

Existe uma grande quantidade de dispositivos WiFi que suportam o modo *monitor*, o qual necessita de privilégios de administrador. Apesar desse facto, o modo *monitor* em conjunto com a ferramenta *aircrack* iria permitir escutar o meio de transmissão e estimar o grau de utilização dos pontos de acesso através da análise da quantidade de dados transmitidos associados a um determinado AP.

Muita da informação enunciada anteriormente necessita de privilégios administrativos, denominado acesso *root* na comunidade Android. Esse modo de acesso não existe na maioria dos

telemóveis uma vez que não é uma funcionalidade pré-instalada e a sua instalação normalmente invalida a garantia dos dispositivos. Assim sendo, todas as funcionalidades que envolvessem o acesso a informação privilegiada não foram consideradas para a implementação deste projeto.

A plataforma Android possui uma grande quantidade de *frameworks*, que permitem o acesso a muita informação da ligação de rede sem privilégios administrativos e sem ferramentas externas, tal como representado na Tabela 3.1.

Tabela 3.1: Exemplo de informação acessível pelas APIs de rede do Android

WifiManager	Qualidade do sinal (%)	
	Informação DHCP	
	Lista dos APs (ScanResult)	BSSID
		SSID
		Frequência
		RSSI
Informação da ligação (Wifiinfo)	Estado da ligação L2/L3 Endereço IP Endereço físico Taxa de transmissão RSSI SSID	

3.6 Discussão

Nas plataformas móveis, mais concretamente em *smartphones*, já existiram investigações relacionadas com o WiFi, no entanto muitas delas foram direcionadas para o impacto do consumo de energia na vida útil da bateria [40], para a utilização das várias interfaces de rede disponíveis em simultâneo de forma a acelerar o acesso à rede [3] e para a utilização de uma determinada interface de comunicação tendo em conta o tipo de aplicação [41] [42]. O sistema Android, tal como qualquer outro OS atual, tem incorporado um sistema de diagnóstico de rede. No entanto, não existe um mecanismo simples que indique a razão de possíveis quedas ou degradação da qualidade da ligação de rede. Posto isto, surgiram diversas investigações e criação de aplicações direcionadas para a melhoria da qualidade de serviço da rede e diagnóstico da performance da mesma [9] [29] [30] [31] [32], que consistiram apenas na recolha de *logs* do sistema e deteção de anomalias; na medição da latência da rede através de um servidor na rede local; na aquisição do RSSI, latência, *throughput*, taxa de transmissão, localização e posterior envio para uma base de dados com a intenção de construir um mapa de desempenho da rede; na utilização de uma placa WiFi externa com a capacidade de detetar dispositivos non-WiFi.

Contudo, as soluções existente carecem no *feedback* para o utilizador. Seria útil para o mesmo saber se o problema está relacionado com o seu equipamento ou se a zona em questão é problemática, isto é, se já outros utilizadores tiveram problemas naquele local. É essencialmente este ponto que distingue este projeto de todos os outros, visto que se pretende utilizar as estatísticas de todos

os utilizadores da rede para devolver um *feedback* ao utilizador. O próximo capítulo descreve a solução proposta e a arquitetura adotada na implementação deste projeto.

Capítulo 4

Solução Proposta

Neste capítulo, é apresentada a arquitetura do sistema, no qual se pretende enviar estatísticas do estado da rede para um servidor e obter o *feedback* relativo ao desempenho da mesma, isto a partir de um dispositivo cliente com tecnologia WiFi. Inicialmente, é apresentado um exemplo de aplicação prática e os requisitos associados a este tipo de sistema. Por último, é abordada e explicitada a arquitetura funcional do sistema.

4.1 Aplicação Prática

Cada vez existem mais dispositivos móveis com tecnologia WiFi, que por sua vez está presente em muitos locais públicos, empresas, universidades, etc. No caso das universidades, existem milhares de utilizadores e, por vezes, cada utilizador possui diversos dispositivos WiFi, tal como *smartphones*, *tablets* e computadores portáteis.

De forma a satisfazer a necessidade dos utilizadores, normalmente existem dezenas de pontos de acesso à rede distribuídos nos mais variados locais de forma a abranger a maior área de cobertura possível. No entanto, existem sempre locais com melhor e pior qualidade de acesso à rede. A razão pode ser temporária ou específica de um determinado local e pode ser provocada por interferência, pela distância ao ponto de acesso ou por uma elevada afluência de utilizadores.

Posto isto, o sistema a desenvolver tem a finalidade de resolver essa situação na medida em que os utilizadores, perante uma situação de fraca qualidade de serviço, podem diagnosticar a sua ligação através da recolha de dados que caracterizem a qualidade da mesma. Os dados recolhidos são comparados com outros utilizadores no mesmo local e noutros locais, o que permite diagnosticar a origem do problema. Assim, o utilizador tem a noção se apenas o seu dispositivo é considerado problemático ou se o mesmo acontece com outros utilizadores. Tal como representado na Figura 4.1, podem existir vários padrões de desempenho para vários locais e diferentes dispositivos na mesma rede. Deste modo, podem existir locais em que a maioria dos dispositivos não apresentam anomalias, locais em que o acesso à rede apresenta anomalias constantemente e locais sem quaisquer problemas de desempenho no acesso à rede.

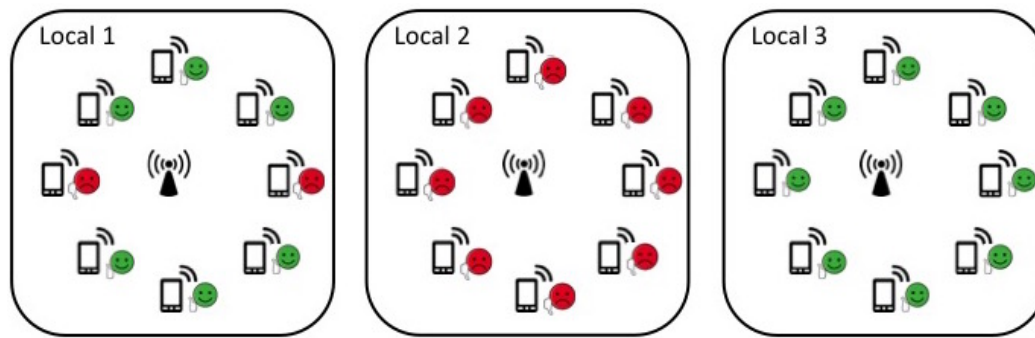


Figura 4.1: Padrões distintos de desempenho na mesma rede

Para a implementação de um sistema com a capacidade de diagnosticar as várias situações apresentadas na Figura 4.1, é necessário capturar dados, efetuar o seu processamento e construir um *feedback* para o utilizador, o que envolve vários requisitos funcionais. Desde logo existe a necessidade de ter um servidor com a capacidade de armazenar toda a informação proveniente dos utilizadores, é ainda necessário definir a informação a recolher nos dispositivos clientes, o método de comunicação entre o servidor e o cliente, o método de processamento dos dados pelo servidor e as mensagens de *feedback* a devolver ao utilizador.

4.2 Arquitetura

A arquitetura geral do sistema, de forma abstrata, consiste na interação entre um servidor e um ou mais dispositivos clientes, tal como exemplificado na Figura 4.2.

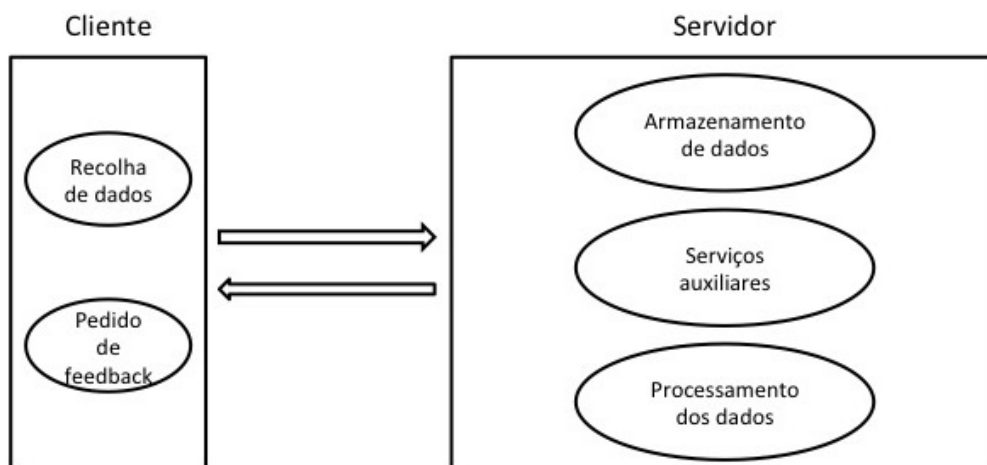


Figura 4.2: Diagrama do funcionamento do sistema

O cliente executa uma aplicação que é responsável pela recolha de dados, pelo envio dos mesmos para o servidor e pela exibição do *feedback* gerado pelo servidor ao utilizador. O servidor tem a capacidade de armazenar a informação enviada pelos utilizadores e, quando solicitado por um

determinado utilizador, processar os dados armazenados de forma a gerar um *feedback* a retornar para o utilizador em questão.

4.2.1 Recolha de Dados

A obtenção de *feedback* por parte do utilizador está dependente da recolha de informação relevante e do processamento da mesma. Para o utilizador comum, uma boa experiência de acesso à rede consiste essencialmente num tempo de resposta curto e numa velocidade de ligação razoável, isto é, adequada quanto baste às suas intenções. Por conseguinte, para analisar a realidade descrita anteriormente, utilizaram-se, respetivamente, as seguintes métricas:

- RTT
- *Throughput*

A análise das métricas permite classificar o desempenho do dispositivo cliente face a outros dispositivos quer no mesmo local, quer noutros locais, assim como em diversos instantes temporais. Assim sendo, é necessário identificar o instante de captura da informação e recolher dados no dispositivo cliente que permitam a sua localização e identificação.

A localização do dispositivo pode ser feita por *fingerprinting*, o que envolve a recolha do BSSID do ponto de acesso associado assim como todos os BSSIDs dos pontos de acesso ao alcance, ou de forma manual através da introdução de um identificador da localização, bem conhecido e definido pelos utilizadores da rede. Optou-se por utilizar a localização manual, no entanto manteve-se a recolha de informação dos vários BSSIDs para efeitos de *debugging* ou futura aplicação. Quanto à identificação do dispositivo, utilizou-se o endereço físico da placa WiFi do dispositivo cliente.

4.2.2 Processamento dos Dados

O processamento dos dados armazenados no servidor, tem como objetivo a obtenção de *feedback* para o utilizador da aplicação cliente. Nesse sentido, as estatísticas de *throughput* e de RTT (*Round-Trip Time*) recolhidas no dispositivo em análise têm de ser comparadas com outros dispositivos no mesmo local e em locais diferentes, assim como para vários instantes temporais. A identificação numérica dos vários cenários de comparação encontram-se representados na Tabela 4.1

Tabela 4.1: Cenários de obtenção do *feedback*

	Dispositivo em análise	Outros dispositivos no mesmo local	Outros dispositivos noutros locais
30 min atrás	1	4	7
30 min atrás nos últimos 3 dias	2	5	8
desde sempre	3	6	9

Nos cenários 1, 4 e 7, as estatísticas recolhidas no dispositivo em análise são comparadas com todas as estatísticas recebidas nos últimos 30 minutos pelo próprio dispositivo em análise, por outros dispositivos no mesmo local e por outros dispositivos noutros locais, respetivamente. O mesmo se sucede para os restantes 6 cenários, no entanto a comparação é efetuada com estatísticas recebidas nos últimos 30 minutos para os 3 dias anteriores e para a totalidade de amostras.

Quer para o próprio dispositivo em análise, cenários 1 a 3, quer para outros dispositivos no mesmo local, cenários 4 a 6, e ainda para outros dispositivos noutros locais, cenários 7 a 9, as estatísticas armazenadas no servidor correspondentes a cada cenário são recolhidas e é efetuada uma análise independente tanto para o *throughput* como para o RTT. O resultado da análise, especificado na Tabela 4.2, consiste no cálculo da média e do desvio padrão do conjunto de entradas associadas a um determinado cenário, e posterior relacionamento com os valores de entrada do dispositivo cliente em análise.

Tabela 4.2: Resultado do *feedback* para os cenários 1 a 9

	$entrada < \bar{x} - \sigma$	$\bar{x} - \sigma \leq entrada \leq \bar{x} + \sigma$	$entrada > \bar{x} + \sigma$	s/ info
RTT	+	ok	-	!
<i>Throughput</i>	-	ok	+	!

Caso exista informação no servidor para um determinado pedido, a análise da Tabela 4.2 permite observar 3 desfechos possíveis do *feedback* para cada cenário:

- ”-“: desempenho inferior ao esperado
- ”ok“: desempenho dentro da normalidade
- ”+“: desempenho superior ao esperado

4.2.3 Mensagens de *Feedback*

As mensagens de *feedback* apresentadas ao utilizador, representadas na Tabela 4.3, são sempre relativas à informação recolhida no dispositivo cliente e são semelhantes para cada um dos cenários, exceto nos cenários 7 a 9 com a introdução das localizações antes das mensagens de *feedback*. Desta forma, a partir do desfecho obtido para cada cenário, é apresentada apenas uma mensagem por métrica, ou apenas uma mensagem no caso de não haver informação suficiente no servidor para a análise de um determinado cenário.

Tabela 4.3: Mensagens de *feedback* enviadas para o utilizador

RTT	(-) Tempo de resposta superior
	(ok) Tempo de resposta esperado
	(+) Tempo de resposta inferior
<i>Throughput</i>	(-) Velocidade de ligação inferior
	(ok) Velocidade de ligação esperada
	(+) Velocidade de ligação superior
Sem informação	(!) Sem informação na BD

A implementação da arquitetura descrita ao longo deste capítulo é detalhada, de forma pormenorizada, no próximo capítulo.

Capítulo 5

Implementação

Neste capítulo é descrito o processo de implementação do sistema, o qual aborda o modelo de comunicação entre o servidor e o cliente, assim como o modo de funcionamento e as características de cada um.

5.1 Comunicação

A comunicação entre o cliente e o servidor foi implementada através de serviços web REST (*Representational State Transfer*), o que implica que o cliente utilize o protocolo HTTP (*Hypertext Transfer Protocol*) para efetuar pedidos POST e GET para enviar e receber informação do servidor, respetivamente. No corpo do pedido POST são enviadas as estatísticas para o servidor e no corpo da resposta do servidor a um pedido GET é enviado o resultado do *feedback* [43].

No entanto, para que tanto o cliente como o servidor consigam comunicar é necessário interpretar a informação trocada entre ambos. Por conseguinte, optou-se por usar o formato JSON (*JavaScript Object Notation*), o qual suporta os mais variados tipos de variáveis e consiste numa trama delimitada por chavetas, cujas variáveis são separadas por vírgulas e os valores são associados às variáveis através de 2 pontos, conforme o exemplo abaixo:

```
{"String ":" valor ", "Inteiro ":20}
```

5.2 Cliente

A aplicação para utilização em dispositivos clientes foi desenvolvida na plataforma Android com recurso à plataforma de desenvolvimento [Android Studio](#) e focou-se essencialmente em 2 aspetos. O primeiro consistiu na aquisição de estatísticas e respetivo envio para o servidor, enquanto que o segundo consistiu num pedido de *feedback* ao servidor e apresentação do mesmo ao utilizador.

5.2.1 Recolha de Dados

Para a obtenção de *feedback* funcionar é necessário que existam dados no servidor, e para tal é necessário efetuar a sua recolha pelos utilizadores da aplicação cliente. A aplicação tem de recolher dados capazes de identificar o dispositivo, o local, o instante de captura e tem de ser capaz de adquirir as métricas definidas na arquitetura.

Para a identificação dos dispositivos, fez-se uso da classe `WifiInfo` que disponibiliza a função `getMacAddress()` que retorna uma string cujo conteúdo é o endereço físico da placa WiFi do dispositivo.

Apesar da identificação da localização ser introduzida manualmente pelos utilizadores a partir da aplicação, foi recolhida informação relacionada com os pontos de acesso ao alcance para aplicações futuras e/ou *debugging*. Assim sendo, recorreu-se à função `getScanResults()`, que devolve uma lista dos pontos de acesso detetados no último *scan* efetuado. Essa lista, entre outros campos, inclui o BSSID e o RSSI dos pontos de acesso. O valor do RSSI, em dBm, foi convertido para uma escala de 0 a 100 com a função `calculateSignalLevel(int rssi, int numLevels)`, evidenciando a qualidade do sinal em percentagem. Foi ainda recolhido o BSSID relativo ao ponto de acesso ao qual o cliente se encontra associado através da função `getBSSID()`.

O instante de captura das estatísticas a enviar para o servidor foi recolhido com a função `Date()` e `SimpleDateFormat("yyyy-MM-dd HH:mm:ss")`. A primeira função retorna o tempo, com uma precisão em milissegundos, desde 1 de Janeiro de 1970 em UTC. A segunda função converte o valor numérico para uma formato interpretável pelos humanos.

Quanto às métricas, *throughput* e RTT, a sua recolha necessitou de duas ferramentas auxiliares, o *iperf* e o *hpcbench*, respetivamente. Ambas as ferramentas são gratuitas, o código fonte está disponível à comunidade e funcionam com base no paradigma cliente-servidor. Deste modo, foi necessário compilar o código fonte da respetiva aplicação cliente para a plataforma Android, mais concretamente para a arquitetura *armv7*. O processo de compilação foi executado num sistema computacional de arquitetura *armv7* e com sistema operativo Debian, nesse sentido utilizou-se o compilador *gcc* com o parâmetro *-static* para forçar o uso de bibliotecas estáticas, visto que não estão disponíveis no sistema operativo Android. Os ficheiros binários gerados necessitam de ser executados a partir de um terminal, e para tal, fez-se uso do seguinte excerto de código para executar um determinado comando:

(...)

```
ProcessBuilder ps = new ProcessBuilder(command);
ps.redirectErrorStream(true);
Process pr = ps.start();
```

(...)

Quer para o cálculo do *throughput*, quer para o cálculo do RTT é necessário executar comandos no terminal:

- *Throughput*
 - `iperf -c <servidor> -p <porta>`
- RTT
 - `udptest -ah <servidor> -p <porta> -r 10`

O valor do RTT poderia ser conseguido através do comando *ping*, no entanto as versões Android mais recentes necessitam da permissão de administrador. Para contornar essa situação fez-se uso do *hpcbench* para calcular o valor do RTT.

Além das métricas RTT e *throughput*, definidas na arquitetura do sistema, foram implementadas mais métricas para futuras aplicações:

- APsameCH - número de APs na mesma frequência [7] [12]
- ConnectRetry - número de tentativas de associação ao AP
- ConnectTime - segundos de estabelecimento da ligação
- RSSIquality - qualidade do sinal em percentagem [4] [5] [7] [9]
- LinkSpeed - velocidade nominal da ligação em Mbps [9] [11] [13]
- LatencyJitter - variação da latência da rede em ms
- LossRate - percentagem de perda de pacotes [11] [12]
- Txretry - número de retransmissões ao nível do MAC

Também foi implementada outra funcionalidade que consiste na introdução do nome da localização do dispositivo pelo utilizador para questões de debugging.

Após toda a informação ter sido recolhida, é organizada de acordo com os seguintes campos de dados para ser posteriormente enviada para o servidor:

- TimeCaptured (timestamp)
 - Instante temporal em que o cliente capturou as estatísticas a enviar para o servidor
- DeviceID (string)
 - Identificador do dispositivo cliente (endereço físico da placa WiFi)
- APs (string)

- Lista do BSSID e da respetiva qualidade de sinal (em percentagem) dos pontos de acesso visíveis pelo cliente
- AP (string)
 - Identificador do BSSID do ponto de acesso ao qual o cliente estava ligado
- LocalDebug (string)
 - Identificador da localização do dispositivo cliente
- Stats (string)
 - Lista com todas as métricas recolhidas pelo dispositivo cliente

O envio da trama em JSON para o servidor, exemplificado na Figura 5.1, é efetuado com um POST via HTTP para o servidor.

```
{
  "TimeCaptured": "2015-05-26 14:22:29",
  "DeviceID": "10:68:3F:7B:8A:F1",
  "APs": [{"AP": "F0:29:29:79:8E:62", "RSSIquality": 82}, {"AP": "F0:29:29:79:8E:60", "RSSIquality": 82}],
  "AP": "F0:29:29:79:8E:60",
  "LocalDebug": "bib piso 0",
  "Stats":
  "{
    "APsameCH": 0,
    "ConnectRetry": 0,
    "ConnectTime": 2,
    "RSSIquality": 82,
    "LinkSpeed": 24,
    "Throughput": 13527,
    "RTT": 6,
    "LatencyJitter": 3,
    "LossRate": 4,
    "Txretry": 0
  }"
}
```

Figura 5.1: Exemplo de trama enviada para o servidor em formato JSON

5.2.2 Aquisição de Feedback

A aquisição de *feedback* consiste apenas no envio de um pedido GET via HTTP para o servidor. O servidor interpreta o pedido e constrói a respetiva resposta em JSON, a aplicação apenas a tem de decodificar e apresentar ao utilizador utilizador.

A resposta do servidor, em formato JSON, encontra-se exemplificada na Figura 5.2 e o conteúdo da mesma apresenta o *feedback* a ser decodificado pela aplicação cliente.

```

{
  "QueryID_30_same_Ans": "(!) Sem informacao na BD",
  "QueryID_30_mult_same_Ans": "(+) Velocidade de ligacao superior\n(-) Tempo de resposta superior",
  "QueryID_all_same_Ans": "(ok) Velocidade de ligacao esperada\n(ok) Tempo de resposta esperado",

  "Others_30_same_Ans": "(-) Velocidade de ligacao inferior\n(ok) Tempo de resposta esperado",
  "Others_30_mult_same_Ans": "(-) Velocidade de ligacao inferior\n(-) Tempo de resposta superior",
  "Others_all_same_Ans": "(-) Velocidade de ligacao inferior\n(ok) Tempo de resposta esperado",

  "Others_30_other_Ans": "(!) Sem informacao na BD\n",
  "Others_30_mult_other_Ans": "1103:\n(+) Velocidade de ligacao superior\n(+) Tempo de resposta inferior\n",
  "Others_all_other_Ans": "bib piso 0:\n(+) Velocidade de ligacao superior\n(+) Tempo de resposta inferior\n
    bib piso 6:\n(+) Velocidade de ligacao superior\n(+) Tempo de resposta inferior\n
    1103:\n(+) Velocidade de ligacao superior\n(+) Tempo de resposta inferior\n"
}

```

Figura 5.2: Exemplo de resposta do servidor em formato JSON

A interpretação da resposta do servidor consiste descodificação da trama JSON e na obtenção das várias mensagens de *feedback*, do tipo strings, para cada um dos cenários 1 a 9 da Tabela 4.1, tal como exemplificado no excerto de código abaixo representado:

(...)

```
json.getString("QueryID_30_same_Ans")
```

(...)

A opção de criar as mensagens de *feedback* no servidor e não no cliente apenas teve o objetivo de reduzir a necessidade de processamento do lado do cliente.

5.3 Servidor

Neste sistema, o conceito de servidor consiste numa plataforma composta por vários serviços interligados que, juntos, satisfazem todas as necessidades da aplicação cliente. Os serviços disponibilizados pelo servidor consistem numa interface de comunicação com o dispositivo cliente, numa base de dados para armazenamento de estatísticas e em serviços que auxiliam a aquisição das métricas pelos utilizadores. As respetivas abordagens para cada serviço encontram-se descritas nas sub-seções seguintes.

5.3.1 Interface

A interface de comunicação com o cliente foi baseada em PHP através de um servidor web Apache. O serviço em PHP, após receber uma ligação de um cliente, analisa se o método do pedido corresponde a um POST ou a um GET e, perante o resultado, executa o processo adequado. Se o pedido corresponder a um POST, a trama recebida em JSON é processada e os valores são armazenados na base de dados. Por outro lado, na ocorrência de um pedido GET, é efetuado um processo mais longo que envolve a recolha de estatísticas da base de dados, o processamento

das mesmas e o envio do *feedback* resultante para o cliente, em JSON, no corpo da mensagem "HTTP/1.1 200 OK".

5.3.2 Armazenamento

O servidor, ao receber um POST, tem de ter um suporte de armazenamento para guardar a informação dos vários dispositivos clientes. Por conseguinte, optou-se por criar uma tabela numa base de dados MySQL com os seguintes campos e respetivo tipo de variável:

- ID (inteiro)
 - Identificador da entrada na base de dados, é incrementado automaticamente sempre que um cliente envia estatísticas para o servidor
- TimeReceived (timestamp)
 - Instante temporal em que o servidor recebeu os dados do cliente
- TimeCaptured (timestamp)
- DeviceID (string)
- APs (string)
- AP (string)
- LocalDebug (string)
- Stats (string)

Quando é efetuado um POST para o servidor, é necessário extrair a informação para a armazenar na tabela da base de dados. A decodificação da trama em JSON enviada pelo cliente foi realizada através do código abaixo apresentado:

```
$json = file_get_contents( 'php://input' );  
$json_data = json_decode( $json );
```

O próximo passo consiste na verificação da integridade da trama decodificada, que é realizada através da análise dos campos de dados enviados pelo cliente. Na eventualidade dos campos de dados enviados pelo cliente não corresponderem ao esperado pelo servidor, é enviada para o cliente a seguinte mensagem de erro:

```
(...)  
  
$result_data = array(  
    'SVack' => "invalid JSON parameters",  
);  
  
(...)
```

Após a verificação dos parâmetros, é realizada a ligação à base de dados e os parâmetros são enviados. O procedimento descrito é realizado através do seguinte código:

(...)

```
$con = mysqli_connect("db.fe.up.pt","ee12246",<password>,"ee12246");
```

(...)

```
$query_insert = mysqli_query($con, "INSERT INTO wifiserver
(TimeCaptured, DeviceID, APs, AP, LocalDebug, Stats)
VALUES(' $json_data ->TimeCaptured ',
' $json_data ->DeviceID ',
' $json_data ->APs ',
' $json_data ->AP',
' $json_data ->LocalDebug ',
' $json_data ->Stats ')");
```

(...)

5.3.3 Serviços Auxiliares

Ao cliente interessa caracterizar o desempenho da sua ligação da forma mais eficiente possível, o que implica que o tráfego gerado para aquisição das métricas, ao chegar ao AP, seja encaminhado pela rede com fios até ao servidor. Por conseguinte, criaram-se 3 serviços no servidor, utilizados pelo cliente para obtenção de estatísticas, 2 deles através do iperf e o outro usando o hpcbench. Ambos são poderosas ferramentas com múltiplas funcionalidades para testar o desempenho e a qualidade das redes, tanto em modo TCP (*Transmission Control Protocol*) como em UDP (*User Datagram Protocol*).

Foram configurados 2 serviços para o iperf, um deles em modo TCP na porta 5001 e outro UDP na porta 5002. O modo TCP foi utilizado para medir o *throughput* do cliente, enquanto que o modo UDP foi implementado para suportar as funcionalidades extra e teve a finalidade de medir a percentagem de perda de pacotes e a variação da latência (*jitter*). Quanto ao hpcbench, apenas foi utilizado o modo UDP na porta 5003 com o objetivo de efetuar medições do RTT. Os serviços foram colocados em execução numa máquina virtual, dentro da rede do campus onde foram realizadas as experiências, através da evocação dos seguintes parâmetros:

- iperf em TCP
 - iperf -s -p 5001
- iperf em UDP
 - iperf -u -s -p 5002

- hpcbench em UDP
 - `udpserver -p 5003`

5.3.4 Processamento

O processamento consiste na seleção de informação da base de dados para posterior tratamento com a finalidade de gerar mensagens de *feedback* para o utilizador. Para o processo de tratamento de dados recorreu-se a todos os campos da tabela da base de dados com exceção do *TimeReceived*, *APs* e *AP*, os quais poderão ter uma aplicação futura.

O campo *TimeReceived*, apesar de ter sido planeado, não chegou a ter qualquer utilidade na implementação efetuada. A ideia deste campo consistia em obter a diferença horária entre a aquisição de estatísticas pelo cliente e a sua receção no servidor. Devido a eventuais problemas de ligação, poderia não ser possível enviar as estatísticas para o servidor e, como tal, as mesmas seriam armazenadas *offline* no dispositivo para posterior envio na presença de uma ligação com qualidade.

Durante a fase de planeamento e definição da arquitetura, a lista dos pontos de acesso, campo *APs*, mostrou vir a ser importante para possível deteção da localização do telemóvel através da identificação dos pontos de acesso e da qualidade do sinal do mesmo. No entanto, um telemóvel nem sempre vê os mesmos pontos de acesso, e esse método de deteção mostrou-se impraticável. No sentido de contornar o problema utilizou-se o campo *LocalDebug* em detrimento do campo *APs*.

No caso do campo *AP*, o objetivo consistiria em detetar, na mesma zona, pontos de acesso com melhores estatísticas de ligação. Tal hipótese foi descartada uma vez que a criação de *feedback* não teve essa informação em consideração.

Quanto ao campo *Stats* da base de dados, que contém uma listagem de todas as métricas recolhidas no cliente, apenas foi processado o RTT e o *throughput*, tal como o que foi definido na arquitetura.

Na ocorrência de um pedido GET, é recolhida a informação da base de dados associada a esse pedido. Após a leitura dessa informação, é necessário recolher os dados correspondentes a cada um dos cenários representados na Tabela 4.1 do Capítulo 4 e efetuar a comparação dos resultados obtidos para cada cenário com os valores de referência do pedido GET.

A determinação dos resultados, tal como descrito na arquitetura, necessita do cálculo da média e do desvio padrão das amostras relativas a cada cenário. Após calculada a média e o desvio padrão, são determinadas as mensagens de *feedback*, de acordo com o especificado na Tabela 4.3 do Capítulo 4.

Uma vez concluído o processamento de cada cenário, caso o número de entradas na base de dados seja inferior a 4, o utilizador é informado de que não existe informação suficiente para a determinação do *feedback*. O *threshold* de 4 foi um valor arbitrário, que apenas serviu para obrigar a que o cálculo da média e do desvio padrão tivesse um valor mínimo de amostras. No final do processamento de todos os cenários, o *feedback* gerado pelo servidor é enviado para o cliente no

formato JSON, tal como exemplificado anteriormente na Figura 5.2 na secção correspondente à implementação do cliente.

No próximo capítulo são apresentados e discutidos os resultados obtidos através da implementação efetuada.

Capítulo 6

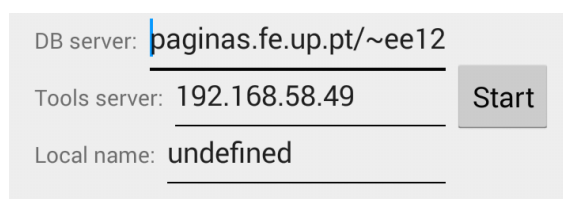
Análise de Resultados

Este capítulo descreve a metodologia utilizada para validar todo o sistema, aborda os casos de uso associados a cada tipo de *feedback*, e termina com a descrição dos ensaios experimentais efetuados na obtenção de resultados.

6.1 Metodologia

Devido à dificuldade inerente à distribuição de uma aplicação cliente a vários utilizadores, optou-se por utilizar 4 telemóveis para efetuar testes e, desta forma, validar o sistema. Para isso, foi necessário adaptar o código da implementação tanto para o envio de dados para o servidor como para a obtenção de *feedback*.

Quanto ao envio de dados para o servidor, apenas foi implementado um ciclo com período de 5 minutos, iniciado e/ou parado através de um botão, tal como exemplificado na Figura 6.1. A cada 5 minutos o WiFi do dispositivo é ligado, as estatísticas são recolhidas, de seguida são enviadas para o servidor e o WiFi é desligado.



The image shows a user interface for data collection. It consists of three input fields and a button. The first field is labeled 'DB server:' and contains the text 'paginas.fe.up.pt/~ee12'. The second field is labeled 'Tools server:' and contains the IP address '192.168.58.49'. The third field is labeled 'Local name:' and contains the text 'undefined'. To the right of the second and third fields is a grey button labeled 'Start'.

Figura 6.1: Interface para recolha de dados

O campo *DB server* corresponde à configuração do endereço do serviço WEB do servidor para execução dos pedidos HTTP POST e GET. O campo *Tools server* está associado ao endereço da máquina virtual que contém os serviços de *iperf* e *hpcbench* para medição das estatísticas como o RTT e *throughput*. Por último, o campo *Local name* serve para configurar a localização do dispositivo.

A aquisição de *feedback* para uma determinada entrada enviada para a base de dados, exemplificado na Figura 6.2, consistiu no envio de um pedido GET com um parâmetro na URL correspondente ao campo *ID* da tabela da base de dados. Esse parâmetro identifica uma entrada na base de dados e devolve o *feedback* relativo à mesma, simulando a aquisição de *feedback* por parte do utilizador.

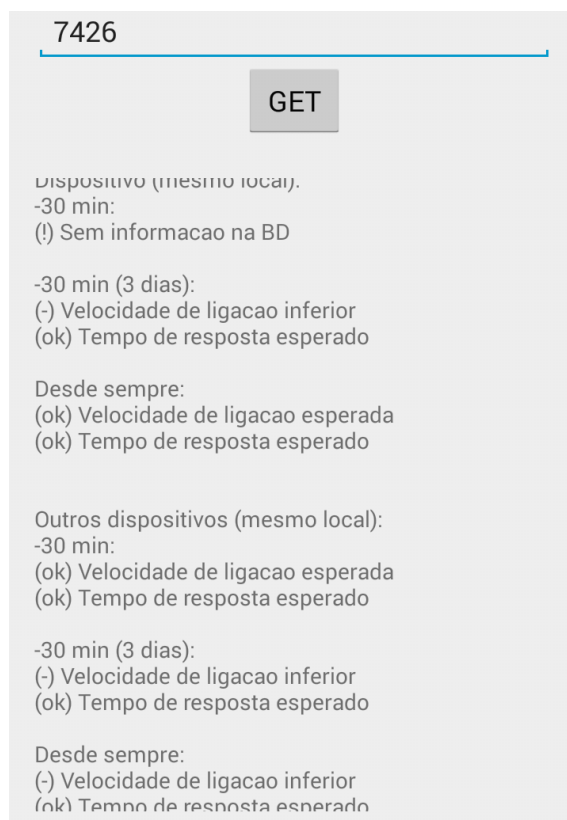


Figura 6.2: Interface para aquisição de *feedback*

A imagem acima exemplifica o pedido de *feedback* para a entrada na base de dados, do servidor, correspondente ao identificador 7426, e permite a visualização do *feedback* obtido para os cenários 1 a 6. A validação do funcionamento do sistema foi efetuada de forma semelhante, mas com o processamento de todas as entradas da base de dados. Para isso, criaram-se *scripts* em linguagem *python*, que processam todas as entradas da base de dados associadas a um determinado teste e analisam o *feedback* associado a cada uma dessas entradas. No final, com a ferramenta *gnuplot*, evocada no *script*, são efetuados histogramas que expressam, em percentagem, os vários desfechos do *feedback* em função do número total de entradas associadas a um teste. Assim, é possível verificar as diferenças de *feedback* entre os vários dispositivos e locais associados.

Os *scripts* em *python*, um para cada teste efetuado, simulam todo o processo efetuado pelo serviço PHP do servidor para a obtenção de *feedback* a devolver ao utilizador. Fazem a ligação direta à base de dados, obtêm as estatísticas, calculam a média e desvio padrão, guardam a soma cumulativa de cada desfecho do *feedback*, assim como o valor total de entradas associadas ao teste.

Essa soma é guardada num ficheiro e, posteriormente, o ficheiro é processado com o *gnuplot* para a criação dos histogramas.

Para testar o sistema, foram efetuados 2 testes com 4 telemóveis colocados de forma estratégica de acordo com os requisitos experimentais:

- Experiência 1
 - Avaliação dos cenários 1 a 6
 - Dispositivos colocados no mesmo local
 - 97 horas consecutivas, de Segunda-feira a Sexta-feira
- Experiência 2
 - Avaliação dos cenários 7 a 9
 - Dispositivos colocados em locais diferentes
 - 73 horas consecutivas, de Terça-feira a Sexta-feira

6.2 Casos de Uso

Para cada cenário, o utilizador recebe um *feedback* que, dependendo do conteúdo do mesmo, pode traduzir-se em várias ações por parte dos utilizadores. Nas Figuras 6.3 e 6.4 estão exemplificados 2 *feedbacks* recebidos correspondentes à experiência 1 e 2, respetivamente. As imagens encontram-se modificadas com indicação dos vários cenários e com indicação de um código de cores para melhor perceção da informação.

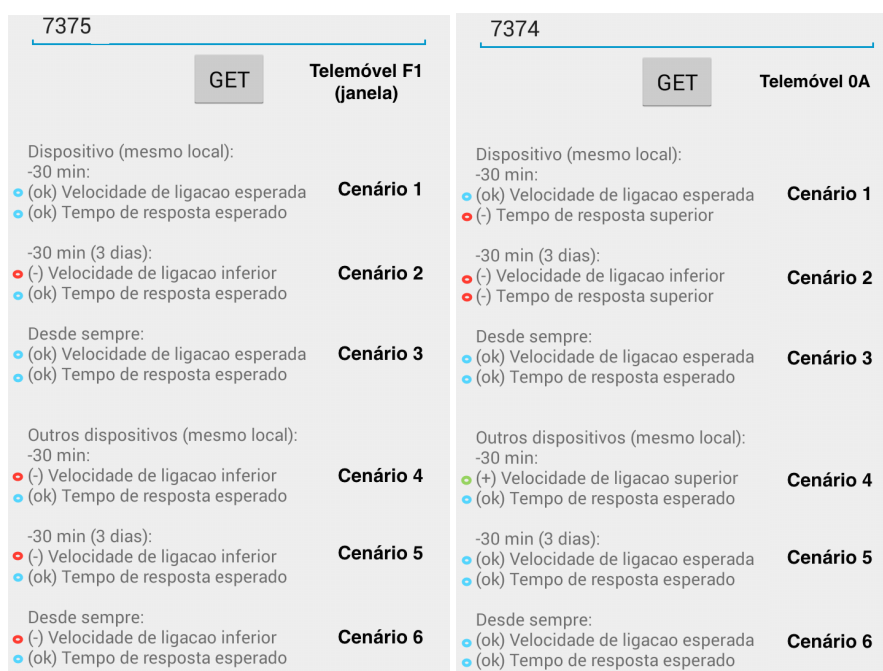


Figura 6.3: Exemplo de *feedback* recolhido da experiência 1

A Figura 6.3 exibe o *feedback* obtido a partir de duas entradas da experiência 1: telemóvel F1 e telemóvel 0A, respetivamente o pior e o melhor dispositivo do comparativo. Após observação da mesma, o utilizador pode retirar as seguintes conclusões:

- Os 3 primeiros cenários são apenas relativos ao dispositivo em análise, daí o utilizador pode verificar o comportamento do seu dispositivo, para um determinado local e instante temporal, e determinar se o mesmo é coerente ou se existem oscilações de desempenho. Para isso a amostra enviada para o servidor é comparada com:
 - Cenário 1 - todas as amostras enviadas pelo utilizador para o servidor nos últimos 30 minutos
 - Cenário 2 - todas as amostras enviadas pelo utilizador para o servidor nos últimos 30 minutos para os 3 dias anteriores
 - Cenário 3 - todas as amostras associadas ao dispositivo para o local em análise
- Em suma, em relação aos cenários 1 a 3, o utilizador do telemóvel F1 pode constatar que o desempenho do mesmo está dentro do padrão, contudo, para o mesmo instante, está com pior desempenho que o observado nos últimos 3 dias. O utilizador do dispositivo 0A pode concluir que o dispositivo apresenta um comportamento padronizado, no entanto está com um desempenho inferior ao observado há meia hora atrás, quer em relação ao instante em análise, quer em relação à mesma hora para os 3 dias anteriores.
- Os cenários 4, 5 e 6 correspondem a todos os dispositivos no mesmo local exceto o dispositivo em análise, o que permite que utilizador compare o desempenho do seu dispositivo com os restantes utilizadores. Nesse sentido, as estatísticas enviadas para o servidor relativas ao dispositivo em análise são comparadas com:
 - Cenário 4 - todas as amostras enviadas pelos outros utilizadores para o servidor nos últimos 30 minutos
 - Cenário 5 - todas as amostras enviadas pelos outros utilizadores para o servidor nos últimos 30 minutos para os 3 dias anteriores
 - Cenário 6 - todas as amostras associadas a outros dispositivos para o local em análise
- De forma sucinta, no que diz respeito aos cenários 4 a 6, o utilizador do telemóvel F1 pode constatar que, no mesmo local, existem dispositivos com melhor desempenho em todos os cenários, o que sugere que dispositivo esteja com algum problema ou que esteja a ser vítima de algum tipo de interferência. Quanto ao utilizador do dispositivo 0A, o desempenho do mesmo está de acordo com o observado pelos diversos utilizadores, no entanto ainda consegue ser melhor do que o observado há meia hora atrás pelos outros utilizadores.

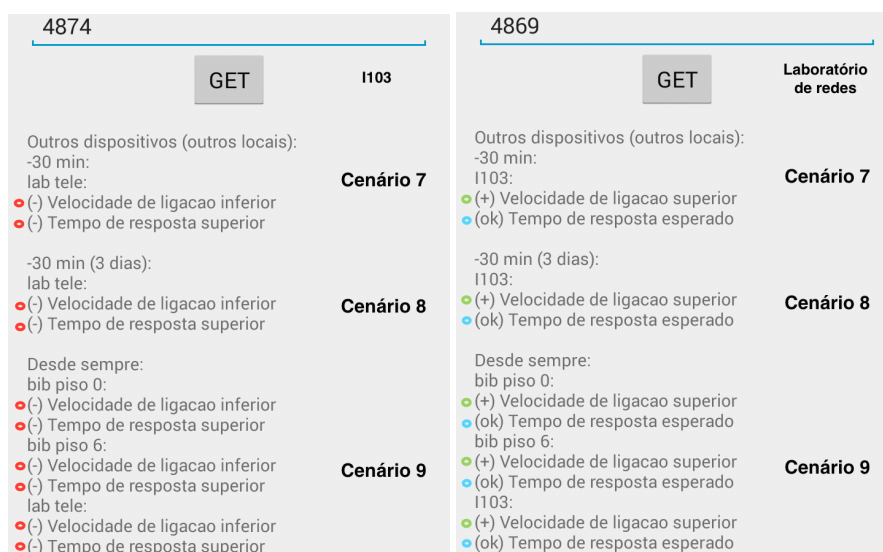


Figura 6.4: Exemplo de *feedback* recolhido da experiência 2

A Figura 6.4 exhibe o *feedback* obtido a partir de duas entradas da experiência 2: I103 e laboratório de redes, respetivamente o pior e o melhor local do comparativo. Os cenários 7, 8 e 9, apresentados na imagem, são relativos à comparação do dispositivo em análise com outros dispositivos noutros locais, o que permite ao utilizador verificar a qualidade de acesso à rede noutros locais e determinar se existem locais com melhor desempenho. Por conseguinte, a amostra enviada para o servidor é comparada com:

- Cenário 7 - todas as amostras enviadas por outros utilizadores, de locais diferentes, para o servidor nos últimos 30 minutos
- Cenário 8 - todas as amostras enviadas por outros utilizadores, de locais diferentes, para o servidor nos últimos 30 minutos para os 3 dias anteriores
- Cenário 9 - todas as amostras associadas a outros dispositivos para locais diferentes

Em suma, o utilizador no gabinete I103 consegue constatar que a sua qualidade de acesso é pior em todos os cenários. Assim sendo, e visto que o cenário 8 indica que àquele instante temporal a conectividade tem sido constantemente mais fraca do que no laboratório de redes, o utilizador percebe que o melhor a fazer é mudar de localização. Quanto ao utilizador no laboratório de redes, todos os cenários apontam para um desempenho dentro na normalidade em termos de latência de acesso e um desempenho superior em relação à velocidade efetiva da rede. Deste modo não existe qualquer necessidade do utilizador em questão mudar de localização.

6.3 Testes

Os telemóveis utilizados foram todos [LG Nexus 4](#) com sistemas operativo Android 4.4 e 5.0, configurados para utilizar somente o espectro de 2.4 GHz em comunicação WiFi, visto que hoje em dia a maioria dos dispositivos WiFi ainda só suportam o espectro de 2.4 GHz, e o desempenho nos pontos de acesso a 5 GHz poderia ser muito superior em locais saturados mas com alguns pontos de acesso a 5 GHz.

Em ambas as experiências apenas foram consideradas entradas válidas, isto é, estatísticas recolhidas com sucesso (valor associado diferente de -1). As métricas RTT e *throughput* estão dependentes de serviços auxiliares prestados pelo servidor, *hpcbench* e *iperf*. No eventual existência de problemas de conectividade, a aquisição de qualquer uma das duas métricas pode falhar, situação que atribui o valor de -1 às métricas. Contudo, existe também a possibilidade dos telemóveis não conseguirem estabelecer ligação ao servidor, sendo que neste caso não há indicação e apenas se pode observar um número inferior de entradas. Após a recolha de todos os dados experimentais, a base de dados ficou com 4604 registos e 3600 KiB, o que corresponde a uma média de 800 bytes por entrada na base de dados.

6.3.1 Experiência 1 - Cenários 1 a 6

O objetivo desta experiência, que aborda os cenários 1 a 6, consiste em analisar o desempenho de um dispositivo, num determinado local, ao longo do tempo e compara-lo com outros dispositivos no mesmo local. Assim sendo, foram colocados 4 dispositivos no mesmo local durante 4 dias consecutivos. Um dos dispositivos foi colocado junto a uma janela, distanciado aproximadamente 2 metros dos outros 3 que foram colocados juntos. As estatísticas relacionadas com percentagem de dados recolhidos e utilizados na experiência encontram-se organizadas na Tabela 6.1.

Tabela 6.1: Dados da experiência 1

Telemóvel	Total de entradas	Entradas válidas	Utilização (%)
F1 (janela)	841	213	25
2E	222	138	62
3B	800	522	65
0A	759	488	64

O posicionamento de um dos telemóveis perto da janela, implicou uma maior distância ao ponto de acesso, sujeitou o telemóvel a mais interferências e, apesar da curta distância de 2 metros em relação aos outros 3 dispositivos, fez com que fosse o dispositivo com a pior relação entre entradas válidas e o número total de entradas, atingindo somente 25 % de entradas utilizadas na experiência. Apesar de todos os telemóveis serem do mesmo modelo, o 2E contém um controlador WiFi diferente, o que poderá ter levado a um número substancialmente inferior de entradas reportadas ao servidor. No entanto, apesar do reduzido número de entradas na base de dados, a taxa de utilização foi semelhante à obtida com os outros 2 dispositivos. Os histogramas que se

seguem, Figuras 6.5, 6.6, 6.7 e 6.8, apresentam os resultados obtidos para cada telemóvel durante a experiência, pela sequência da Tabela 6.1.

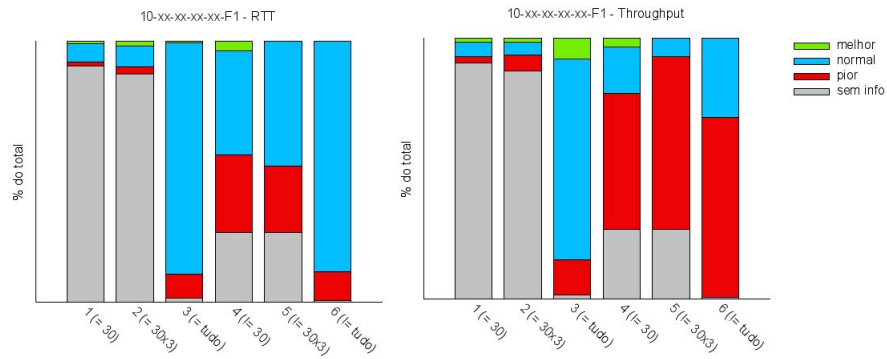


Figura 6.5: Resultado da experiência 1 para o telemóvel F1

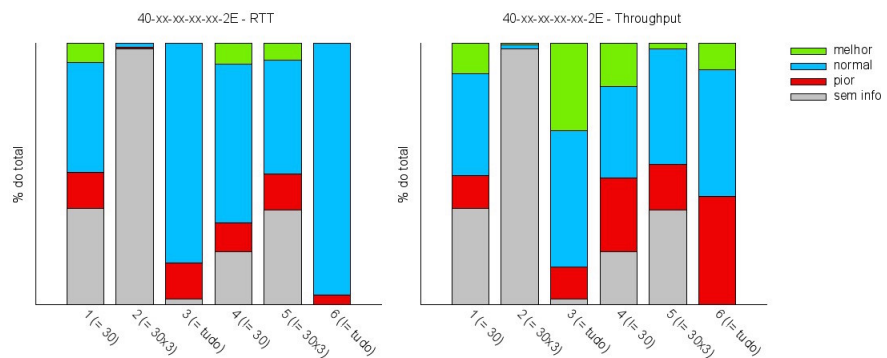


Figura 6.6: Resultado da experiência 1 para o telemóvel 2E

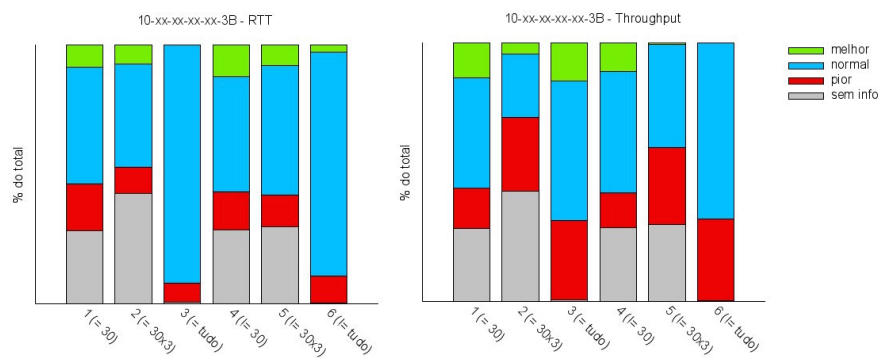


Figura 6.7: Resultado da experiência 1 para o telemóvel 3B

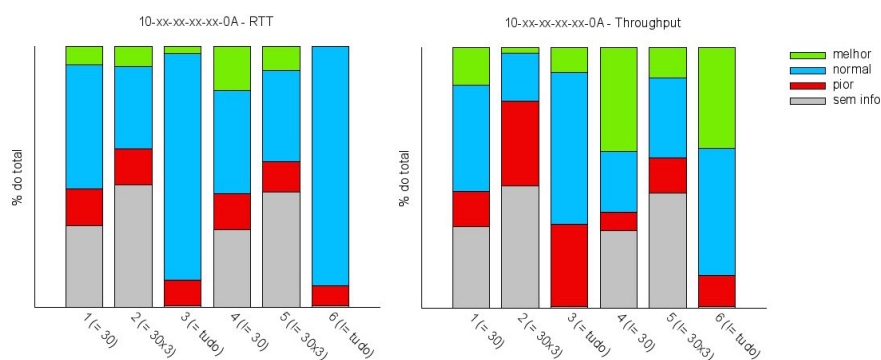


Figura 6.8: Resultado da experiência 1 para o telemóvel 0A

Após análise dos histogramas anteriores, em relação aos cenários 1 a 3, que visam a comparação das estatísticas do próprio dispositivo em análise, verifica-se essencialmente que:

- Quanto ao RTT, todos os dispositivos apresentaram um funcionamento padronizado, visto que apresentam todos uma maior percentagem de valores "normais"
- Quanto ao *throughput*, os dispositivos apresentaram um funcionamento padronizado, com exceção do seguintes casos:
 - No cenário 2, obteve-se uma taxa elevada de valores "piores" para os dispositivos F1, 3B e 0A, o que implica uma degradação do desempenho em relação a dias anteriores
 - No cenário 3, o dispositivo 2E obteve uma taxa elevada de valores "melhores" em comparação com os restantes dispositivos no mesmo cenário, o que se refletiu num desempenho mais inconstante

Em relação aos cenários 4 a 6, que refletem a comparação do dispositivo em análise com outros dispositivos, verifica-se essencialmente que:

- Quanto ao RTT, todos os dispositivos apresentaram um funcionamento padronizado, exceto a seguinte situação:
 - No cenário 4 e 5, o dispositivo F1 apresenta uma taxa de valores "piores" maior em relação aos outros dispositivos, o que corresponde a uma latência superior em relação aos outros dispositivos no mesmo instante
- Quanto ao *throughput*, verificaram-se 3 casos distintos:
 - O dispositivo F1 apresentou, nos 3 cenários em questão, os piores resultados e a maior taxa de resultados "piores" obtidos
 - O dispositivo 0A obteve os melhores resultados gerais, e a maior taxa de resultados "melhores" para cada um dos 3 cenários

- Os dispositivos 2E e 3B obtiveram um desempenho semelhante, muito melhor que o F1, no entanto pior que o 0A

Em suma, os resultados desta experiência permitiram validar o funcionamento do sistema, na medida em que o mesmo mostrou-se apto a detectar as anomalias de desempenho no dispositivo colocado junto à janela. Essa detecção consiste, essencialmente na comparação dos cenários 4 a 6 e, os piores resultados são evidentes.

6.3.2 Experiência 2 - Cenários 7 a 9

A finalidade desta experiência, que aborda os cenários 7 a 9, consiste em analisar o desempenho de um dispositivo, num determinado local, através da comparação com outros dispositivos noutros locais, e em diferentes instantes de tempo. Desta forma é possível determinar a qualidade do acesso à rede em vários locais. Por conseguinte, foram colocados 4 dispositivos em locais distintos durante 3 dias consecutivos. As estatísticas relacionadas com percentagem de dados recolhidos e utilizados na experiência encontram-se organizadas na Tabela 6.2.

Tabela 6.2: Dados da experiência 2

Local	Total de entradas	Entradas válidas	Utilização (%)
I103	584	232	40
Biblioteca piso 0	554	60	11
Biblioteca piso 6	518	37	7
Laboratório de redes	326	235	72

A sala I103, apresenta um sinal razoável, mas está sujeita a várias interferências e são reportados diversos problemas de acesso pelos utilizadores. A biblioteca é um meio saturado, com grande quantidade de utilizadores, o que justifica a baixa taxa de entradas válidas para a experiência. Apesar do laboratório de redes ser alvo de algumas interferências reportadas pelos utilizadores e de ter o menor número de entradas na base de dados, o telemóvel lá colocado ficou a escassos metros do ponto de acesso, o que se refletiu na maior taxa de utilização da experiência. Os resultados obtidos para cada localização estão apresentados, pela sequência da Tabela 6.2, nas Figuras 6.9, 6.10, 6.11 e 6.12.

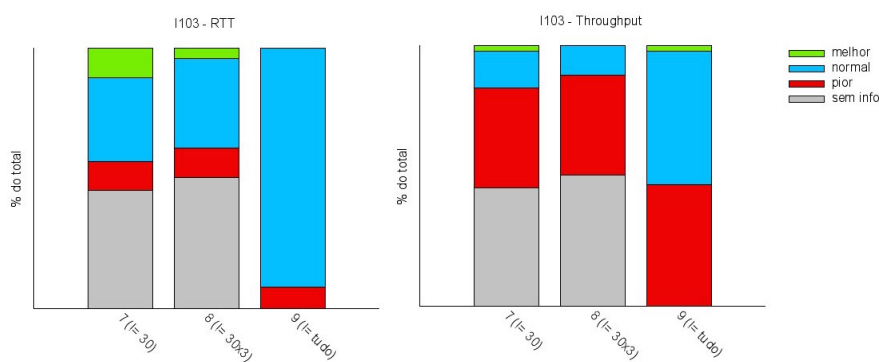


Figura 6.9: Resultado da experiência 2 para o local "I103"

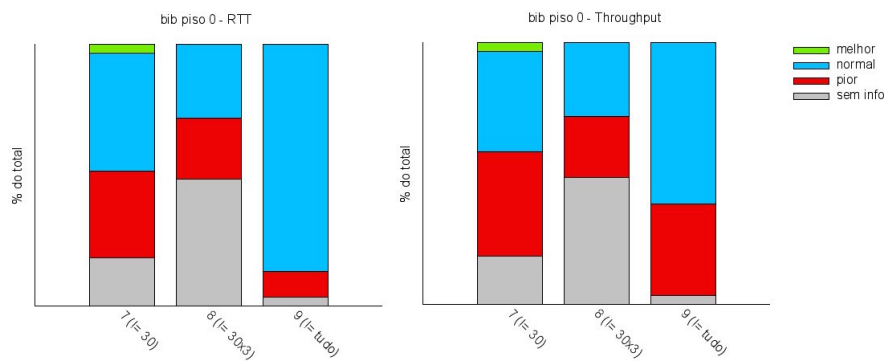


Figura 6.10: Resultado da experiência 2 para o local "Biblioteca piso 0"

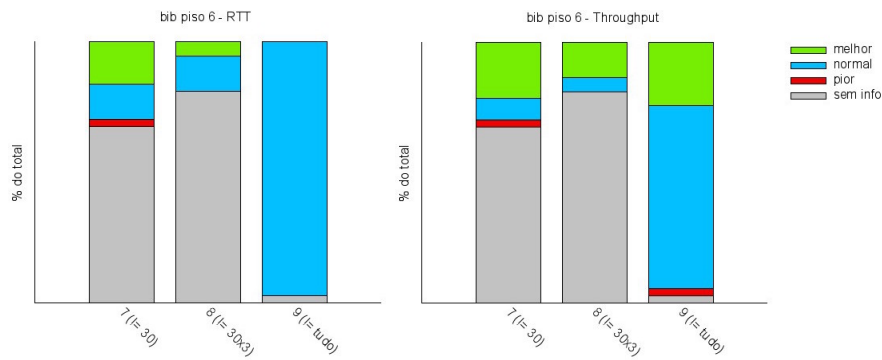


Figura 6.11: Resultado da experiência 2 para o local "Biblioteca piso 6"

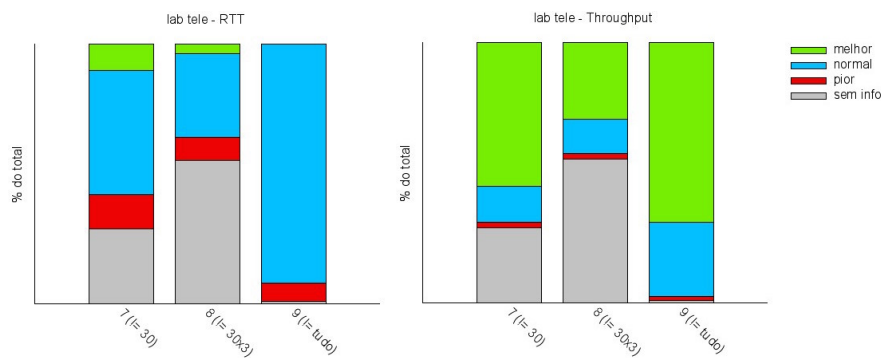


Figura 6.12: Resultado da experiência 2 para o local "Laboratório de redes"

Após análise dos histogramas anteriores, relativos aos cenários 7 a 9, que correspondem à comparação do dispositivo em análise com outros dispositivos noutros locais, verifica-se essencialmente que:

- Quanto ao RTT, verificaram-se 3 casos distintos:
 - As localizações I103 e o laboratório de redes apresentam um funcionamento padronizado, observando-se uma maior quantidade de valores "normais"
 - Para os cenários 7 e 8, o piso 0 da biblioteca apresenta uma taxa de valores "piores" semelhante aos valores "normais", correspondendo ao pior caso do comparativo
 - O piso 6 da biblioteca apresenta a menor taxa de valores piores para todos os cenários, o que se traduz nos melhores valores de RTT obtidos na experiência

- Quanto ao *throughput*, verificaram-se 4 casos distintos:
 - O local I103 obteve a maior quantidade de valores "piores" do comparativo, pelo que é a localização menos aconselhada para acesso à rede
 - O piso 0 da biblioteca apresenta uma taxa de valores "piores" semelhante aos valores "normais"
 - O piso 6 da biblioteca apresenta um desempenho bastante satisfatório, com um desempenho bastante padronizado no cenário 9 e com elevadas taxas de valores "melhores" nos cenários 7 e 8, no entanto é o que apresenta a maior taxa de falta de informação
 - Para todos os cenários, 7 a 9, o laboratório de redes é o local que apresenta as maiores taxas de valores "melhores", o que corresponde ao melhor caso do comparativo

Em síntese, os resultados desta experiência contribuíram para a validação do sistema, uma vez que o mesmo mostrou-se capaz de detetar diferenças de desempenho nos diferentes locais. Os locais I103 e o piso 0 da biblioteca mostraram ser os piores locais tanto em RTT como em *throughput* tal como o esperado, devido à informação pré-existente conhecida. Por outro lado, o laboratório de redes mostrou ser a localização com o melhor desempenho.

Para finalizar, após a apresentação e análise dos resultados obtidos neste projeto, no próximo capítulo são apresentadas as conclusões e abordado o trabalho futuro.

Capítulo 7

Conclusões

Neste capítulo é apresentado um sumário de todo o trabalho efetuado ao longo desta dissertação, em comparação com os objetivos inicialmente definidos e os resultados obtidos. No final, são perspetivados melhoramentos futuros.

7.1 Sumário

O objetivo principal deste trabalho consistiu em desenvolver um sistema para aplicação em dispositivos móveis Android com a finalidade de comparar o desempenho de acesso à rede entre vários dispositivos. O sistema a desenvolver teve de ser capaz de processar informação de vários locais e em diferentes instantes temporais.

Para cumprir todos esses objetivos, o projeto passou por várias fases:

- A primeira fase consistiu em estudar as APIs de rede do Android e ver que informação relevante poderia ser extraída
- A segunda fase foi destinada à definição da metodologia de controlo do sistema e das métricas a utilizar
- A última fase consistiu na implementação e avaliação do sistema

Este documento demonstrou a necessidade da criação de sistemas de diagnóstico para as redes WiFi. Um sistema de diagnóstico, bem implementado e disseminado por diferentes localizações e dispositivos, ajuda a detetar problemas de desempenho num dispositivo específico, num determinado local ou instante temporal.

7.2 Funcionalidades e Resultados

Tal como é possível concluir a partir dos capítulos 5 e 6, o sistema de diagnóstico desenvolvido cumpriu os resultados esperados. Teve a capacidade de processar com sucesso as estatísticas de equipamentos móveis no mesmo local e em locais diferentes.

No final, o sistema de diagnóstico foi capaz de:

- Recolher estatísticas periodicamente nos dispositivos clientes
- Apresentar ao utilizador o resultado da comparação do RTT e do *throughput* com outros dispositivos, locais e diferentes instantes temporais, o que permitiu:
 - Para um determinado local e instante temporal, detetar dispositivos problemáticos, isto é, com pior desempenho
 - Para um determinado local, detetar padrões constantes de mau desempenho
 - No mesmo intervalo temporal, detetar locais com melhor desempenho
 - No mesmo instante temporal, ao longo de vários dias, detetar padrões constantes de mau desempenho

7.3 Trabalho Futuro

Apesar dos objetivos do projeto terem sido cumpridos, o projeto em si ainda tem muitas portas abertas. O sistema implementado utilizou apenas o RTT e o *throughput* como métricas para comparação com os outros utilizadores, no entanto foram recolhidos mais dados tal como enunciado no ponto 5.2.1 do capítulo de implementação.

No capítulo 6, foram realizadas experiências para validar o sistema. Para obter os resultados dessas experiências, utilizaram-se *scripts* em *python*. No entanto, apesar do sistema ter dispositivos móveis Android como aplicação prática, seria bastante interessante desenvolver uma plataforma para os administradores de rede com a finalidade de monitorizar o desempenho da rede em diversas localizações.

A aplicação apenas exhibe ao utilizador o *feedback* devolvido pelo servidor, mas poderia também fazer sugestões ao utilizador com base nas estatísticas armazenadas no servidor. Por vezes, num determinado local, existem vários pontos de acesso e, dado que o identificador do ponto de acesso é recolhido, essa informação poderia ser utilizada para comparar o desempenho entre vários pontos de acesso, o que por sua vez poderia ser utilizado para sugerir ao utilizador que, num determinado local, se associasse a um ponto de acesso com melhor desempenho.

Referências

- [1] S. Rayanchu, “Models and systems for understanding wireless interference,” Ph.D. dissertation, Ann Arbor, 2012. [Online]. Disponível em: <http://search.proquest.com/docview/1043755991?accountid=43623>
- [2] R. Zhou, “Exploiting cross-technology interference for efficient network services in wireless systems,” Ph.D. dissertation, Ann Arbor, 2014. [Online]. Disponível em: <http://search.proquest.com/docview/1614473023?accountid=43623>
- [3] K.-K. Yap, T.-Y. Huang, M. Kobayashi, Y. Yiakoumis, N. McKeown, S. Katti, and G. Parulkar, “Making use of all the networks around us,” *ACM SIGCOMM Computer Communication Review*, vol. 42, p. 455, 2012.
- [4] A. P. Subramanian and C. Science, “Improving Capacity and Connectivity in Wireless Access Networks,” *ProQuest Dissertations and Theses*, no. May, p. 197, 2009. [Online]. Disponível em: <http://search.proquest.com/docview/305104152?accountid=43623>
- [5] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, “Measurement-based models of delivery and interference in static wireless networks,” p. 51, 2006.
- [6] E. Magistretti, “Coordination and Interference in 802.11 Networks: Inference, Analysis and Mitigation,” Ph.D. dissertation, Ann Arbor, 2013. [Online]. Disponível em: <http://search.proquest.com/docview/1469386241?accountid=43623>
- [7] H. Balbi, N. Fernandes, F. Souza, R. Carrano, C. Albuquerque, D. Muchaluat-saade, and L. Magalhaes, “Centralized channel allocation algorithm for IEEE 802.11 networks,” ser. 2012 Global Information Infrastructure and Networking Symposium (GIIS 2012). Univ. Fed. Fluminense (UFF), Niteroi, Brazil BT - 2012 Global Information Infrastructure and Networking Symposium (GIIS 2012), 17-19 Dec. 2012: IEEE, 2012, p. 7 pp. [Online]. Disponível em: <http://dx.doi.org/10.1109/GIIS.2012.6466657>
- [8] V. Navda and C. Science, “Cross-layer design for interference mitigation and mobility support in wireless access networks,” Ph.D. dissertation, Ann Arbor, 2007. [Online]. Disponível em: <http://search.proquest.com/docview/304750369?accountid=43623>

- [9] S. Subbarao, “Diagnosing Wi-Fi association issues and optimizing network performance,” Ph.D. dissertation, Ann Arbor, 2013. [Online]. Disponível em: <http://search.proquest.com/docview/1418268758?accountid=43623>
- [10] R. Madan, A. Sampath, and N. Khude, “Enhancing 802.11 carrier sense for high throughput and QoS in dense user settings,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, pp. 253–259, 2012.
- [11] Q. Javed, “Improving the performance of wireless mesh networks with better medium access and routing strategies,” Ph.D. dissertation, Ann Arbor, 2012. [Online]. Disponível em: <http://search.proquest.com/docview/1267825853?accountid=43623>
- [12] R. Sombrutzki, P. Vidales, A. Zubow, P. Vidales, and J.-P. Redlich, “On channel assignment, distributed antennas and network load distribution in dense IEEE 802.11 infrastructure networks,” in *6th International Wireless Communications and Mobile Computing Conference, IWCMC 2010*, Humboldt University, Rudower Chaussee 25, Berlin, Germany, 2010, pp. 37–41. [Online]. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-77955137061&partnerID=40&md5=00be9f44302ee5f23eca69577f507e0c>
- [13] S. Hakim, “Radio Resource Management in Dense WiFi Networks,” Ph.D. dissertation, Ann Arbor, 2010. [Online]. Disponível em: <http://search.proquest.com/docview/847554572?accountid=43623>
- [14] R. Costa, “RT-WiFi: Uma Arquitetura para Comunicação de Tempo-Real em Redes IEEE 802.11 Infraestruturadas,” 2013. [Online]. Disponível em: <http://repositorio-aberto.up.pt/handle/10216/70523>
- [15] A. Johnson, *LAN Switching and Wireless CCNA Exploration Labs and Study Guide Instructor’s Edition*.
- [16] Intel, *Next Gen 802.11ac Wifi for Dummies*.
- [17] Microsoft, “How 802.11 Wireless Works,” 2003. [Online]. Disponível em: [https://technet.microsoft.com/en-us/library/cc757419\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc757419(v=ws.10).aspx) [Acedido em: 2014-02-08]
- [18] “Network Structure,” 2011. [Online]. Disponível em: <http://mobilus.org.ua/struktura-seti.html> [Acedido em: 2014-02-11]
- [19] J. C. Z. W. Steven Conner, Jan Kruys, Kyeongsoo (Joseph) Kim, “IEEE 802.11s Tutorial Overview of the Amendment for Wireless Local Area Mesh Networking Key Contributors,” *IEEE 802 Plenary*, p. 93, 2006.
- [20] J. Tourrilhes, “iwconfig(8) - Linux man page.” [Online]. Disponível em: <http://linux.die.net/man/8/iwconfig> [Acedido em: 2014-02-08]

- [21] J. Ruela, “Wireless LANs - IEEE 802.11 e 802.11e,” 2009. [Online]. Disponível em: http://paginas.fe.up.pt/~jruela/Apontamentos/802_11e_v0910_RBL_2slides.pdf
- [22] M. Burton, “802.11 Arbitration,” *Ophthalmology*, vol. 119, no. September, pp. 208–209, 2012.
- [23] “Client and Access Point Association.” [Online]. Disponível em: <https://nasserbuzid.wordpress.com/client-and-access-point-association/> [Acedido em: 2014-02-08]
- [24] “IEEE 802.11 Architecture,” 2013. [Online]. Disponível em: http://www.tutorial-reports.com/wireless/wlanwifi/wifi_architecture.php [Acedido em: 2014-02-08]
- [25] Cisco, “802.11 Association process explained,” 2015. [Online]. Disponível em: https://kb.meraki.com/knowledge_base/80211-association-process-explained [Acedido em: 2014-02-08]
- [26] —, “Unified Wireless Network: Troubleshoot Client Issues,” 2009. [Online]. Disponível em: <http://www.cisco.com/c/en/us/support/docs/wireless/4400-series-wireless-lan-controllers/107585-client-con.html#topic4> [Acedido em: 2014-02-08]
- [27] S. Rayanchu, A. Patro, and S. Banerjee, “Airshark: Detecting non-WiFi RF Devices Using Commodity WiFi Hardware,” *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 137–154, 2011. [Online]. Disponível em: <http://doi.acm.org/10.1145/2068816.2068830>
- [28] A. Kashyap, “Measurement-based modeling of interference in wi-fi networks: techniques and applications,” *ProQuest Dissertations and Theses*, no. December, p. 134, 2010. [Online]. Disponível em: <http://gradworks.umi.com/33/86/3386251.htmlhttp://search.proquest.com/docview/304376890?accountid=43623>
- [29] S. Rosen, S.-j. Lee, J. Lee, P. Congdon, Z. Mao, and K. Burden, “MCNet: Crowdsourcing wireless performance measurements through the eyes of mobile devices,” *IEEE Communications Magazine*, vol. 52, no. 10, pp. 86–91, 2014. [Online]. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6917407>
- [30] W. Li, R. K. P. Mok, D. Wu, and R. K. C. Chang, “On the Accuracy of Smartphone-based Mobile Network Measurement.”
- [31] V. Radu, L. Kriara, and M. K. Marina, “Pazl: A mobile crowdsensing based indoor WiFi monitoring system,” *2013 9th International Conference on Network and Service Management, CNSM 2013 and its three collocated Workshops - ICQT 2013, SVM 2013 and SETM 2013*, pp. 75–83, 2013. [Online]. Disponível em: <http://www.cnsm-conf.org/2013/documents/papers/CNSM/p75-radu.pdf>

- [32] A. Patro, P. Panwaria, and S. Banerjee, “WiSense : A client based framework for wireless diagnosis,” 2014. [Online]. Disponível em: http://pages.cs.wisc.edu/~patro/papers/wisense_nsd2014.pdf
- [33] S. A. Yuvaraj, L. C. Siddanna Gowd, and A. I. Technology, “High performance implementation of RSSI based Wi-Fi location tracker for android applications,” *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, pp. 97–106, 2015. [Online]. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84920996903&partnerID=tZOtx3y1>
- [34] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, “Achieving privacy preservation in WiFi fingerprint-based localization,” in *Proceedings - IEEE INFOCOM*, ser. IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. State Key Lab. of Inf. Security, Inst. of Inf. Eng., Beijing, China BT - IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 27 April-2 May 2014: IEEE, 2014, pp. 2337–2345. [Online]. Disponível em: <http://dx.doi.org/10.1109/INFOCOM.2014.6848178>
- [35] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, “Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization,” *2011 International Conference on Localization and GNSS, ICL-GNSS 2011*, pp. 53–57, 2011.
- [36] N. Hernandez, M. Ocana, S. Humanes, P. Revenga, D. P. Pancho, L. Magdalena, and N. Hern, “A WiFi-based software for indoor localization,” in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, Jul. 2014, pp. 2345–2351. [Online]. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84912553599&partnerID=tZOtx3y1>
- [37] M. A. Mahamud and M. Chowdhury, “Indoor location system with Wi-Fi and alternative cellular network signal,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 3, pp. 59–70, 2015. [Online]. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84926343727&partnerID=tZOtx3y1>
- [38] Torben Grøn Helligsø, “RSSI and LQI vs . Distance Measurement,” no. May, p. 36, 2011.
- [39] J. a. C. Cichaczewski, “Análise da Interferência Mútua Entre Redes IEEE 802.11 e IEEE 802.15.4,” 2013.
- [40] A. J. Pyles, X. Qi, G. Zhou, M. Keally, X. Liu, and X. Liuy, “SAPSM: Smart adaptive 802.11 PSM for smartphones BT - 14th International Conference on Ubiquitous Computing, UbiComp 2012, September 5, 2012 - September 8, 2012,” ser. UbiComp’12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing. College of William and Mary, United States McGill University, United States: Association for Computing Machinery, 2012, pp. 11–20. [Online]. Disponível em: <http://dx.doi.org/10.1145/2370216.2370219>
- [41] F. Patriarca, S. Salsano, M. Bonola, and P. Cerqua, “UPMT Per-application mobility management solution: A demo for linux and android terminals BT - 10th ACM International

- Symposium on Mobility Management and Wireless Access, MobiWac 2012, October 24, 2012 - October 25, 2012,” ser. MobiWac’12 - Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access. Dip. Ingegneria Elettronica, Universita di Roma Tor Vergata, Roma, ItalyCNIT - Research Unit, Universita di Roma Tor Vergata, Roma, Italy: Association for Computing Machinery, 2012, pp. 163–166. [Online]. Disponível em: <http://dx.doi.org/10.1145/2386995.2387026>
- [42] O. B. Yetim and M. Martonosi, “Adaptive delay-tolerant scheduling for efficient cellular and WiFi usage BT - 15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2014, June 19, 2014,” ser. Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, WoWMoM 2014. Princeton University, United States: Institute of Electrical and Electronics Engineers Inc., 2014. [Online]. Disponível em: <http://dx.doi.org/10.1109/WoWMoM.2014.6918945>
- [43] a. Rodriquez, “RESTful Web services: The basics,” no. February, pp. 1–11, 2008. [Online]. Disponível em: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>