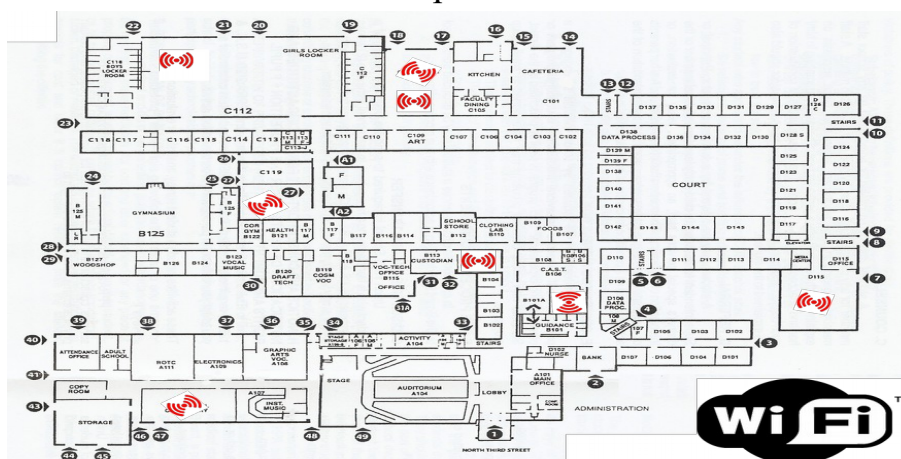




INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores



Sistema de localização interior de baixo custo

LUÍS CARLOS PAULA FERREIRA

Licenciado

Projeto para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Orientadores:

Professor Doutor Nuno Miguel Machado Cruz

Professor Doutor João Carlos Amaro Ferreira

Júri:

Presidente: Professor Doutor Nuno Miguel Soares Datia

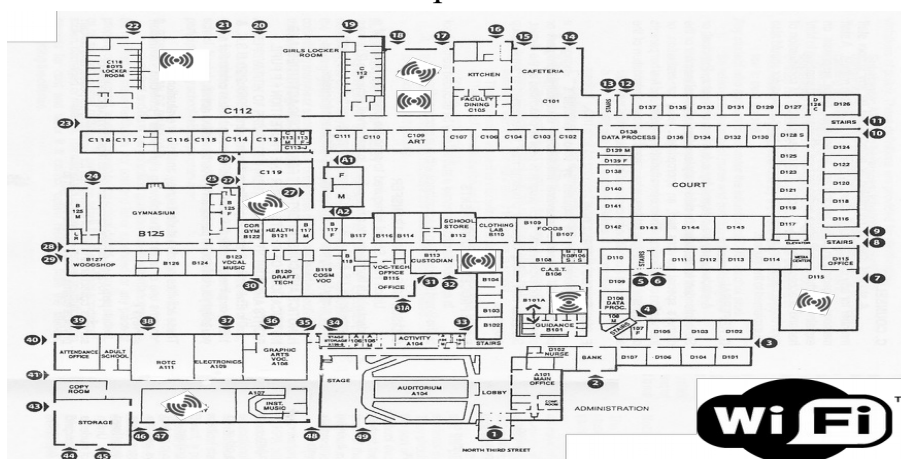
Vogais: Professor Especialista Pedro António Marques Ribeiro

Fevereiro de 2016



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores



Sistema de localização interior de baixo custo

LUÍS CARLOS PAULA FERREIRA

Licenciado

Projeto para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Orientadores:

Professor Doutor Nuno Miguel Machado Cruz

Professor Doutor João Carlos Amaro Ferreira

Júri:

Presidente: Professor Doutor Nuno Miguel Soares Datia

Vogais:

Professor Especialista Pedro António Marques Ribeiro

Fevereiro de 2016

Agradecimentos

Este projeto não seria possível sem a ajuda de outras pessoas individuais e coletivas que aqui faço questão de mencionar:

O doutor Nuno Cruz, orientador, sobretudo pela motivação e empenho;

O ISEL, pela oportunidade de aprender com os melhores;

O meu colega de trabalho, Nuno Martins, pelos ensinamentos Linux;

A comunidade open-source Linux, à qual fico em dívida que tenciono pagar.

A todas estas eu agradeço, com a certeza que passarei a outros o conhecimento comigo partilhado.

Resumo

A geolocalização de pessoas é uma importante fonte de informação no âmbito de processos de marketing. Esta informação é útil, por exemplo, no cenário de decisão de colocação de publicidade num centro comercial, como se propõe neste projeto.

Com este projeto pretende-se desenvolver uma solução de geolocalização de pessoas. A solução baseia-se numa rede Wi-Fi na perspetiva da infraestrutura, em ambiente conhecido à partida. A solução proposta considera tanto o hardware como o software necessários para o objetivo.

Para tal, foram exploradas soluções comerciais da Cisco e soluções desenvolvidas à medida baseadas em software gratuito, bem como serviços web de geolocalização. Propõe-se uma implementação de referência baseada em raspberry pi.

Palavras-chave: geolocalização interior, Wi-Fi, RSS, hardware, Linux, kismet, tshark, wireshark, reconhecimento de padrões, raspberry pi, modo monitor.

Abstract

The geolocation of people is an important source of information within marketing processes. This information is useful, for example, in advertising placement decision in a shopping center setting, as is proposed in this project.

This project aims to develop a people geolocation solution. The solution is based on an Wi-Fi network in the infrastructure perspective, on a previously known environment. The proposed solution considers both the hardware and the software required for the purpose.

To this end, Cisco commercial solutions and custom developed solutions based on free software were explored, as well as geolocation web-services. A raspberry pi reference implementation is proposed.

Keywords: indoor geolocation, RSS, Wi-Fi, RSS, hardware, Linux, kismet, tshark, wireshark, pattern matching, raspberry pi, monitor mode.

Nomenclatura

USB	Barramento Série Universal
MAC	Endereço <i>Media Access Control</i>
Wi-Fi	implementação de rede local sem fios do standard 802.11 do IEEE
IEEE	Instituto de Engenheiros Eletricistas e Eletrónicos
AP	Ponto de Acesso Wi-Fi
SGBD	Sistema de Gestão de Bases de Dados
API	Interface de Programação de Aplicações
HTTP	Protocolo de Transferência de Hipertexto
CLI	Interface de Linha de Comandos
RF	Radiofrequência
RP-SMA	Conector radiofrequência Sub-Miniatura versão A de Polaridade Invertida
IO	Entrada/Saída
WGS84	World Geodetic System 1984
UTM	Universal Transverse Mercator

Índice

Índice

1 Introdução.....	14
2 Revisão bibliográfica.....	16
2.1 Algoritmos.....	16
2.1.1 Trilateração.....	16
2.1.2 Triangulação.....	18
2.1.3 Tipo mapas de rádio.....	18
Algoritmos Determinísticos.....	18
Algoritmos Probabilísticos.....	20
Solução Cisco.....	20
3 Enquadramento do Problema.....	21
3.1 Considerações Iniciais.....	21
3.2 Wi-Fi.....	22
4 Trabalho Desenvolvido.....	24
4.1 Recolha.....	24
4.1.1 Infraestrutura existente.....	24
4.1.2 Infraestrutura dedicada.....	25
Protótipo (Fase 1).....	26
Protótipo (Fase 2).....	27
Placas Wi-Fi USB.....	28
Instalação Física.....	29
Software de captura.....	33
Protótipo (Fase 3).....	35
Considerações de Operacionalização.....	35
4.1.3 Comparação.....	38

4.2 Processamento.....	39
4.2.1 Serviço de localização.....	39
4.2.2 Solução de localização do tipo reconhecimento de padrões.....	41
Implementação (Fase 2).....	42
4.2.3 Solução de localização do tipo n-lateração.....	44
Implementação (Fase 3).....	45
Simplificações.....	46
Padrão <i>Observable</i> e Reactive Extensions.....	46
Net Topology Suite e ProjNet.....	46
OData.....	47
Server Sent Events.....	49
Notas.....	49
4.3 Solução Global (Fase 2).....	49
4.4 Solução Global (Fase 3).....	51
5 Resultados Obtidos.....	54
6 Conclusões e Trabalho Futuro.....	59
6.1 Conclusões.....	59
6.2 Trabalho futuro.....	59
6.2.1 Recolha.....	60
6.2.2 Processamento.....	61
6.2.3 Solução Global.....	62
6.2.4 Considerações Gerais.....	62
6.3 Privacidade.....	64
7 Bibliografia.....	66
8 Anexos.....	68
8.1 Análise de custo de placas Wi-Fi.....	68
8.2 Lista de Software Utilizado.....	69
8.3 Recolha.....	70

8.3.1 Script de preparação do ambiente.....	70
8.3.2 Script para colocação das interfaces em modo monitor.....	70
8.3.3 Script de captura.....	71
8.3.4 Exemplo de ficheiro de configuração.....	74
8.3.5 Deamon System V (init script).....	74
8.4 Processamento.....	76
8.4.1 Script Windows CLI para ingestão de tramas.....	76
8.4.2 Exemplo de comando para recolha da ground truth.....	76
8.4.3 Exemplo de script para construção da ground truth.....	76
8.4.4 Script SQL para criação do modelo.....	77
8.4.5 Código do serviço web.....	79

Índice de Imagens

Índice de imagens

Ilustração 1: Planta da instalação, com pontos de recolha numerados.....	31
Ilustração 2 Antena omnidirecional rubber duck alto ganho, com extensão RP-SMA.....	32
Ilustração 3: Placa com chipset MT7601, ligada a conector USB-A fêmea e a conector RP-SMA macho.....	32
Desenho 1: Exemplo conceptual de implementação física.....	37
Ilustração 4: Processo de geolocalização (mapa de rádio).....	50
Ilustração 5: Diagrama de sequência para processos de ingestão e subscrição.....	53
Ilustração 6: Gráfico de correção da estimativa por espaço.....	55

Índice de Textos

Índice de textos

4.1.1 captura em modo monitor (Cisco Aironet 1200).....	25
preparação da captura.....	26
versão simplificada do script de captura.....	27
Script bash para instalação do firmware MT7601U.....	29
Mensagens do kernel - timeout usando cabo de 5m.....	30
Mensagens kernel - instabilidade com 4 placas Wi-Fi.....	33
4.2.1 interação HTTP de localização (MLS).....	40
4.4 Interação HTTP de subscrição de estimativas de localização.....	52
5 Interrogação - número de estimativas correctas e incorrectas.....	55

1 Introdução

A geolocalização é uma importante ferramenta diariamente ao serviço das pessoas, sendo nesse âmbito suportada por processos informáticos. O modo de geolocalização mais visível é aquele onde o pedido de localização parte da própria pessoa, respondendo à questão "Onde estou eu?"; Outra forma de localização, é a localização da pessoa partindo da rede.

Esta última vertente de localização manifesta-se no dia-a-dia, por exemplo, no histórico de localização de provedores de geolocalização como o Google, Bing e Apple, entre outros. Isto mostra que também o sistema de localização sabe onde os clientes se encontram. Esta informação é considerada valiosa em vários cenários, incluindo o de marketing direcionado.

Com este projeto pretende-se desenvolver uma solução de localização, com base em Wi-Fi, em ambiente de interior, que possa suportar processos de decisão de colocação de publicidade num centro comercial. A solução a desenvolver deve operar de forma não intrusiva e não deve prejudicar as transmissões sem fios na área de operação. Considera-se como solução não intrusiva uma solução que não imponha um comportamento em particular à pessoa a localizar. O problema a resolver é, portanto, o da localização em espaço conhecido.

Existindo diferentes formas de categorizar uma solução de geolocalização interior, um dos primeiros desafios que se coloca é a tipificação da solução quanto ao espectro de sinal, quanto à extremidade da estimativa de localização, quanto à métrica do sinal e

quanto ao método de processamento desse sinal. Destes, dois foram já definidos pelo tema do trabalho, conforme a sistematização que se segue no capítulo Wi-Fi.

Seguem-se, numa parte inicial deste documento, um capítulo dedicados à exposição de trabalho relevante na área, objeto de estudo que ajudou a guiar o desenvolvimento da solução. Posteriormente, é incluído um capítulo que descreve o âmbito do problema e, conseqüentemente, ajuda a delimitar os moldes da solução. O capítulo seguinte descreve a solução, desde a fase de concepção até às fases finais de prototipagem, incluindo considerações de instalação. Dedicase um capítulo a enquadrar a solução desenvolvida em realidades futuras, apontando trabalho futuro pertinente.

2

Revisão bibliográfica

Atualmente existem poucas soluções integradas de geolocalização confiáveis que permitam localização partindo de uma rede Wi-Fi [FB01]. Todas as soluções baseadas numa rede Wi-Fi referidas utilizam hardware dedicado. Dentro destas, a abordagem integrada “Cisco Wireless Control System” apresenta-se promissora, dado que efetua a recolha junto dos pontos de acesso Wi-Fi [TS01] [CS01] (possivelmente já existentes no local de recolha para o cenário escolhido) e disponibiliza os dados do lado da infraestrutura [CS02]. Considera-se equivalente a solução Ekahau, dado que utiliza a mesma tecnologia Cisco [Eka01]. Em alternativa, poder-se-ão avaliar outras soluções baseadas na observação das *frames* Wi-Fi [IEEE 802.11].

Dado que é necessário desenvolver o subsistema de geolocalização, será necessário escolher um algoritmo de localização. Existem vários algoritmos disponíveis [SGT01], seguindo-se uma avaliação dos mesmos.

2.1 Algoritmos

2.1.1 Trilateração

O método de trilateração faz corresponder um determinado valor da métrica a uma distância do ponto de referência. A localização dos sensores é, portanto, necessária. Correlacionando as métricas observadas por vários pontos de referência, obtêm-se a estimativa da localização. Este algoritmo pode ser aplicado sobre várias métricas [LTA08]:

- TOA – Time Of Arrival: Utiliza o tempo entre envio e a recepção do sinal para determinar a distância percorrida, tendo em conta a velocidade de propagação.
- TDOA – Time Difference Of Arrival: Aplica-se sobre a diferença de tempos na recepção entre vários sensores.
- RSS – Received Signal Strength

A utilização da métrica Time Of Arrival é impossível de implementar nas premissas deste trabalho, pois requerem conhecimento preciso do momento de emissão da trama, momento esse que não está disponível.

A utilização da métrica Time Difference Of Arrival não foi considerada exequível por não haver suficiente resolução temporal para permitir a sua aplicação: Admitindo uma propagação de sinal a $300\text{m}/\mu\text{s}$, seria necessária uma resolução na ordem dos 10 nano-segundos para erros de aproximadamente 3m. Para um maior erro contribui ainda outro fator: Numa situação onde não exista linha-de-visão desobstruída, a reflexão de sinal pode fazer variar o tempo de propagação do mesmo, e, portanto, a diferença do tempo de chegada.

A aplicação deste algoritmos sobre a métrica potência do sinal recebido (RSS – Received Signal Strength) sofre de problemas semelhantes: falta de resolução da métrica e evolução descontínua da estimativa em função da métrica (aumentando o valor da métrica de um valor fixo, pode fazer aumentar a distância para alguns casos e fazer descer noutros). Outro fator que introduz erro é a variância, para um determinado ponto fixo, da potência de sinal (seguindo distribuição uniforme [PLM08]). Esta variância existe porque a força de sinal recebido depende de fatores ambientais. Este algoritmo deverá ser preterido em favor de outro que trate destes problemas. Este é um dos algoritmos utilizados pelo projeto Mozilla Ichnaea no seu serviço de geolocalização [Calc15].

O algoritmo de trilateração aplicado à métrica RSS requer a utilização de um modelo de atenuação (*path-loss model*), que transforme o domínio da força de sinal no domínio da distância em espaço físico, como o apresentado em [LTA08].

Uma generalização deste tipo de algoritmos é a n-lateração, que utiliza um qualquer número n ($n > 2$) de sensores com vista a melhorar a precisão em caso de erros das métricas.

2.1.2 Triangulação

Este algoritmo calcula a distância do emissor a partir da distância entre dois pontos de captura e os ângulos formados entre ambas as retas que unem o emissor a cada um dos pontos de captura com a reta que une os pontos de captura. Este algoritmo só é aplicável sobre a métrica ângulo de chegada (AOA – Angle Of Arrival). Numa instalação com hardware equipado com antenas omnidirecionais numa configuração típica, esta métrica não é possível de obter. É de notar que este algoritmo necessita da informação de localização dos pontos de captura.

2.1.3 Tipo mapas de rádio

Este tipo de algoritmos tem como objetivo modelar o espaço em função da força de sinal recebido, para encontrar a melhor correspondência.

Este tipo de algoritmos requer um mapeamento prévio do espaço sob análise, registando a métrica força de sinal. A este mapeamento prévio corresponde a primeira de duas fases, a fase *offline*, também conhecida por *Site Survey*. A segunda fase estima a localização com base na métrica e nos dados de treino recolhidos na primeira fase. Alguns algoritmos deste grupo tratam explicitamente do problema da oscilação da força de sinal para um ponto fixo, havendo outros ainda que suportam o seu tratamento. Os algoritmos deste tipo podem apresentar melhores resultados do que o de trilateração [Calc15]. Estes lidam bem com cenários do mundo real como a atenuação de obstáculos, e são utilizados em soluções proprietárias de localização pela Cisco [CLT06], como a “Wireless Location Appliance”.

Algoritmos deste tipo podem classificar-se enquanto determinísticos e probabilísticos, segundo [SGT12]:

Algoritmos Determinísticos

Os algoritmos determinísticos usam o valor da força de sinal recebido em cada ponto de captura para uma determinada localização. Seguem-se exemplos de algoritmos deste tipo:

1. *Nearest Neighbour in Signal Space*: A estimativa de localização é a localização do ponto mapeado que minimiza a distância euclidiana no espaço da força de sinal (relativamente à amostra a estimar). Dispensa informação sobre a localização relativa dos pontos de captura.
2. *Nearest Neighbour in Signal Space – Average*: A estimativa de localização é a média das localizações de vários pontos que minimizam a distância euclidiana no espaço da força de sinal. A quantidade de pontos a considerar e o tipo de média são variações ao algoritmo. Tal como o anterior algoritmo, dispensa informação sobre a localização relativa dos pontos de captura.
3. *Smallest Polygon*: A estimativa da localização é o centroide do polígono de menor área formado por pontos no espaço geográfico que minimizam a distância euclidiana à amostra no espaço da força de sinal. Este algoritmo assemelha-se ao anterior, e tem como variações ao algoritmo o número de pontos a considerar. Tem como requisito a existência de informação de posicionamento absoluto (coordenadas) das amostras de treino (recolhidas na fase *offline*).
4. *Approximate Point-in-Triangulation*: A estimativa da localização é o centroide da área formada pela intersecção de todos os triângulos formados pelos pontos de captura que observam o dispositivo a localizar. Este algoritmo utiliza a métrica força de sinal no domínio booleano (presente ou ausente), requerendo assim uma menor complexidade do ponto de captura. No entanto, para uma maior precisão na amostra, será expectável a necessidade de um conjunto de pontos de captura maior do que para os algoritmos que consideram um maior domínio na métrica. O algoritmo utiliza a localização geográfica dos pontos de captura, dispensando a primeira fase de recolha. Utilizando pontos de recolha com capacidade de reportar a força de sinal com precisão maior que dois níveis (presente e ausente), ter-se-á a versatilidade de escolher um limiar de força de sinal (cut-off) para a presença. Este algoritmo requer posicionamento absoluto (coordenadas) dos pontos de captura.

Algoritmos Probabilísticos

Os algoritmos probabilísticos usam a distribuição de valores da força de sinal recebido, sendo que esta depende de fatores ambientais. Assume-se uma distribuição Gaussiana dos valores de *RSS* e calculam-se indicadores estatísticos (como a média e a variância) com base nas recolhas efetuadas na fase *offline*. A estimativa da localização corresponde ao local com maior probabilidade de ser igual à amostra.

Solução Cisco

Apesar de não ser público o algoritmo utilizado pela Cisco apresentado em [CLT06], são mencionadas a métrica (força de sinal) e técnicas empregues (captura efetuada pelos APs) na solução Wireless Location Appliance. Sabe-se que é criado um modelo de predição de força de sinal com base na posição dos APs e planta do espaço (fornecida pelo utilizador). A descrição do algoritmo assume à partida a possibilidade de um dispositivo a localizar ser observado por diferentes pontos de captura (APs) com diferentes intensidades de sinal. Não fica claro na análise da documentação se é feito uso das tramas emitidas pelos próprios APs como parte da construção do modelo.

3

Enquadramento do Problema

Sendo necessário definir o âmbito deste projeto, e, antes disso, definir com mais detalhe o problema que se propõe resolver, seguem-se as devidas definições.

3.1 Considerações Iniciais

Existem atualmente dois tipos de soluções de geolocalização quanto à origem do pedido: Uma com base na visão do dispositivo (o objeto a localizar), outra com base na visão da infraestrutura (*network based*).

No primeiro tipo, o dispositivo envia um pedido de geolocalização para o serviço, comunicando-lhe a sua visão (visão de dispositivo), construída com base em elementos por ele recolhidos, dos quais se destacam:

- Força de sinal dos *Access Points* Wi-Fi na proximidade, incluindo tempo desde a última observação;
- Força de sinal de *beacons* Bluetooth na proximidade;
- Velocidade, aceleração e orientação, com base nos sensores do dispositivo (acelerómetro, giroscópio, magnetómetro);

Com base na visão da infraestrutura, e tendo em conta o requisito de definir uma solução não-intrusiva, é possível observar (passivamente) vários espectros rádio analisando as emissões espúrias dos dispositivos. Particularmente úteis são a

observação do Wi-Fi e Bluetooth, dada a sua quase onipresença e disponibilidade de técnicas e tecnologias para recolha e tratamento de informação.

As soluções de localização com base na visão de dispositivo não cumprem os requisitos enunciados, no entanto, são o tipo mais abundante. Dado que se parte de uma visão de infraestrutura, uma hipótese será combinar a recolha não-intrusiva de uma visão de infraestrutura com o processo de geolocalização de uma solução com base na visão de dispositivo. Esta opção apenas é viável caso se cumpram os seguintes pressupostos no sistema a desenvolver:

1. É possível interagir com o serviço da mesma forma que o dispositivo original;
2. É possível transformar a visão de infraestrutura na visão de dispositivo entregue ao serviço;

Esta hipótese de solução híbrida evidencia a primeira divisão lógica dos desafios a abordar:

1. Recolha de dados de posicionamento;
2. Processamento dos dados para obtenção de informação de localização;

3.2 Wi-Fi

Partindo do problema inicial da localização, e tendo em conta o contexto de um centro comercial, parte-se do princípio que existirá provavelmente uma infraestrutura Wi-Fi instalada. Isto deverá ser tido em consideração na abordagem ao problema, dado que se quer minimizar o impacto no serviço Wi-Fi providenciado aos utilizadores do espaço sob observação (visitantes do centro comercial).

No caso do Wi-Fi afigura-se como possibilidade a recolha passiva de *frames* do grupo *management*, tipo *probe request* [IEEE 802.11], dado que são espúrias, originam no dispositivo a localizar e contêm informação relevante (força de sinal, endereço MAC do dispositivo). Os canais Wi-Fi utilizados pelos clientes poderão ser quaisquer uns, pelo que a solução a desenvolver deverá conseguir captar sobre qualquer canal Wi-Fi (de entre os canais em uso na Europa).

Da bibliografia estudada retira-se uma tipificação possível das diferentes técnicas disponíveis para resolver o problema da localização. Categoriza-se da seguinte forma este problema:

1. quanto à categoria do sinal: **RF** (rádio frequência), mais concretamente, o espectro usado pelo Wi-Fi, está definido à partida;
2. quanto à categoria da extremidade da estimativa de localização: **baseado na rede**, pois é a partir dos elementos fornecidos pela rede de captura que irá ser feita a localização. O dispositivo a localizar considera-se passivo;
3. quanto à métrica do sinal: **indefinida** (a avaliar entre tempo de chegada e força de sinal);
4. quanto ao método de processamento do sinal: **indefinido** (a avaliar entre trilateração e reconhecimento de padrões).

Está assim definido o objeto do trabalho a realizar.

4

Trabalho Desenvolvido

Segue-se o detalhe da análise das soluções enunciadas no capítulo anterior, separando dois sistemas: O da recolha de informação dos dispositivos, e o do processamento dessa mesma informação com vista à geolocalização.

4.1 Recolha

O sistema de recolha de informação tem como objetivo a recolha das tramas Wi-Fi emitidas pelos dispositivos a localizar e disponibilizar informação sobre essas tramas por forma a integrar com o sistema de processamento.

4.1.1 Infraestrutura existente

Uma possível solução passa por utilizar infraestrutura previamente existente. A solução requer Access Points e um Wireless LAN Controller. Após experimentação com *Access Points* da linha Aironet, constatou-se que o modelo 1100 não tem a funcionalidade de recolha necessária (captura de *probe requests* e envio para consola). O modelo 1200 apresenta essa funcionalidade, no entanto, as captações efetuadas não apresentam elementos essenciais à localização (força de sinal e relação sinal-ruído) na interface de *debug*, conforme se constata no *output* recolhido a partir de ligação telnet:

Texto 1: captura em modo monitor (Cisco Aironet 1200)

User Access Verification

```
Username: Cisco
Password:
AP10-ID011C>enable
Password:
AP10-ID011C#term mon
AP10-ID011C#
*Mar  1 03:09:38.180: ToLwapp:      000581, 001, -11b, ffff.ffff.ffff,
0024.2c22.6812 SSID: NULL
*Mar  1 03:09:38.181: ToLwapp:      000582, 001, -11b, ffff.ffff.ffff,
0024.2c22.6812 SSID: eduroam
*Mar  1 03:09:38.182: ToLwapp:      000583, 001, -11b, ffff.ffff.ffff,
0024.2c22.6812 SSID: Vodafone-42F541
*Mar  1 03:09:38.332: ToLwapp:      000619, 001, -11b, ffff.ffff.ffff,
0024.2c22.6812 SSID: NULL
*Mar  1 03:09:40.448: ToLwapp:      00065D, 001, -11b, ffff.ffff.ffff,
74f0.6d77.e8c5 SSID: eduroam
```

A diferença nas capacidades dos modelos Aironet levou a reconsiderar a operacionalização de uma solução baseada em infraestrutura Cisco: O requisito de existir uma rede wireless Cisco não basta, sendo necessário detalhar quais os equipamentos suportados, o que reduz ainda mais a probabilidade da pré-existência de uma infraestrutura compatível no ambiente de destino. Para isto contribui também o facto de diferentes versões do sistema operativo apresentarem diferentes subconjuntos de funcionalidades. Outro aspeto a considerar na operacionalização é a heterogeneidade na interface de vários modelos, mesmo sendo estes compatíveis: a mesma funcionalidade pode ser acedida de diferentes formas e apresenta diferente saída consoante o modelo e/ou versão do IOS. Esta heterogeneidade dificulta a automatização e portabilidade da solução.

4.1.2 Infraestrutura dedicada

Uma abordagem para a recolha dos dados relativos ao Wi-Fi relevantes para a localização é usar ferramentas de análise de redes Wi-Fi para PC para a recolha desses elementos. Uma proposta de solução é usar mini-PC equipados com placas de rede Wi-Fi

que exponham a capacidade de modo monitor. Esta solução, apesar de requerer infraestrutura dedicada (possivelmente duplicada em relação à infraestrutura de serviço existente), minimiza o investimento na mesma, alavancando em hardware *off-the-shelf* de baixo custo. Outra das estratégias utilizadas para minimizar esse investimento passa por colocar vários pontos de captura (placas Wi-Fi USB) em cada ponto de recolha (mini-PC).

Por forma a melhor determinar as capacidades de uma solução deste tipo, tornou-se importante prototipar os módulos de recolha e processamento de dados. Assume-se, assim, a separação de responsabilidades entre recolha e tratamento de tramas, sendo necessário definir uma interface entre estes dois módulos.

Descrevem-se de seguida os passos da prototipagem e opções tomadas durante a implementação.

Protótipo (Fase 1)

O primeiro protótipo desta solução de recolha consiste num mini-PC Raspberry Pi 2 equipado com uma placa de *chipset* compatível (no teste efetuado, *chipset* Ralink), no que toca ao hardware. A existência de drivers Linux com capacidade de modo monitor, para uma determinada placa Wi-Fi, pode ser consultada na página da comunidade do subsistema wireless [LW01]. A solução proposta faz uso de utilitários de configuração de rede (ifconfig, iwconfig) e de captura de rede (tshark), como se explica a seguir.

Partindo de uma instalação limpa da distribuição Linux Raspbian¹ (distribuição de Maio de 2015), é necessário instalar e configurar o pacote tshark (versão 1.8.2) e configurar a interface de wireless para modo Monitor, como se exemplifica nos comandos que seguem.

Texto 2: preparação da captura

```
sudo apt-get install tshark
dpkg-reconfigure wireshark-common
#(opção yes)

sudo ifconfig wlan0 down
sudo iwconfig wlan0 mode Monitor
sudo ifconfig wlan0 up
```

¹ Assume-se que o Raspbian inclui driver para a placa Wi-Fi usada.

A captura em si consiste em mudar periodicamente o canal Wi-Fi ao qual a placa está afeta enquanto se registam os *probe requests*, conforme o *script bash* que se segue.

Texto 3: versão simplificada do script de captura

```
stdbuf -oL tshark -i wlan0 -I -f 'broadcast' -R 'wlan.fc.type == 0 &&
wlan.fc.subtype == 4' -T fields -e frame.time_epoch -e wlan.sa -e
radiotap.dbm_antsignal
```

```
while true;
do
    #echo "Scanning channels..."
    for channel in {1..14}
    do
        #echo "Switching to channel $channel"
        sudo iwconfig wlan0 channel $channel
        sleep 1
    done
done
```

Este procedimento serve de base para a captura, registando apenas *probe requests*, e apenas o endereço MAC do dispositivo, força de sinal e marca temporal da captura. Juntando a estes dados a identificação do ponto de captura, que pode advir, por exemplo, de configuração, tem-se a informação de captura.

Protótipo (Fase 2)

Os objetivos da segunda fase do protótipo são:

1. Implementar um ponto de recolha com mais do que um ponto de captura;
2. Conseguir captura com placa Wi-Fi diferente (modelo e chipset);
3. Ter em conta a automatização do processo;
4. Implementar um processo de importação e armazenamento de dados.

Para suportar a instalação em ambientes desligados (sem conectividade externa), foi decidido utilizar persistência local das tramas captadas. No protótipo foi utilizado para persistência o sistema de ficheiros local, suportado no cartão de memória. Esta solução

deverá ser reequacionada em ambiente de produtivo, conforme descrito em Considerações de Operacionalização. Esta solução permite também a utilização de um sistema de ficheiros remoto (e.g. CIFS, SMB, NFS) no caso de uma instalação em ambientes ligados. Uma outra alternativa, útil em ambientes com ligação condicionada (em tempo parcial), é o envio esporádico dos ficheiros de captura para um ponto remoto, (usando, por exemplo, *rsync* ou *logrotate*).

Placas Wi-Fi USB

Por forma a implementar um ponto de recolha com mais do que um ponto de captura, foi necessário adicionar mais placas WiFi. As placas WiFi adicionais inicialmente disponíveis para teste têm chipset Realtek RTL8192cu. A abundância de placas Wi-Fi USB com este chipset a baixo custo no mercado nacional, levou a considerar-se este dispositivo como parte da solução. Inclui-se como anexo a Tabela 5: Preço de placas Wi-Fi (Julho 2015, MediaMarkt), que evidencia esta situação.

Existe indicação [LW01] que o chipset RTL8192cu disponibiliza modo monitor. Em experimentação, verificou-se a falta deste suporte: O suporte existe na árvore oficial do kernel Linux, suportado pelo [módulo rtlwifi](#); No entanto, devido a [problemas reportados](#) pelos utilizadores do Raspbian, a distribuição usa um driver alternativo entregue pelo fabricante do chipset ([módulo rtl8192cu](#)) sem suporte a modo monitor. Tentou-se compilar o driver rtlwifi, sem conseguir produzir um módulo instalável no ambiente de testes.

A instalação de uma outra placa com chipset da família RT2800U foi “plug & play”. O módulo correspondente foi encontrado, e carregado, bem como o firmware necessário. Para as placas com chipset MT7601U testadas (Ilustração 3), foi necessário apenas instalar o firmware, dado que o driver foi portado da versão 4.2 do kernel oficial para a versão 4.1 do kernel usado no Raspbian, estando já presente. Segue-se o processo de instalação do firmware.

Texto 4: Script bash para instalação do firmware MT7601U

```
# get the firmware sources from the vendor
cd /usr/src

# or wget http://groenholdt.net/Computers/RaspberryPi/MediaTek-MT7601-USB-WIFI-on-
the-Raspberry-Pi/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
wget
http://cdn-cw.mediatek.com/Downloads/linux/mt7610u\_wifi\_sta\_v3002\_dpo\_20130916.tar
.bz2

# or tar xf DPO*
tar xf mt*

# or cd DPO*
cd mt*

# copy the vendor's firmware to our default load-path
cp mcu/bin/MT7601.bin /lib/firmware/mt7601u.bin
```

Verificou-se, para diferentes placas Wi-Fi, diferentes níveis de suporte ao modo monitor:

- RTL8192cu: suporte Linux, mas não na distribuição em uso;
- MT7601U: driver incluído, firmware não instalado;
- RT2800U: driver e firmware incluídos – “plug & play”.

Instalação Física

Para implementar os pontos de captura, foram colocadas inicialmente 2 placas Wi-Fi USB, uma junto do RaspberryPi, outra afastada de aproximadamente 3m, com linha-de-visão desobstruída, recorrendo a uma extensão USB. Verificou-se que os *probe requests* recolhidos apresentavam, conforme esperado, diferentes valores de força de sinal.

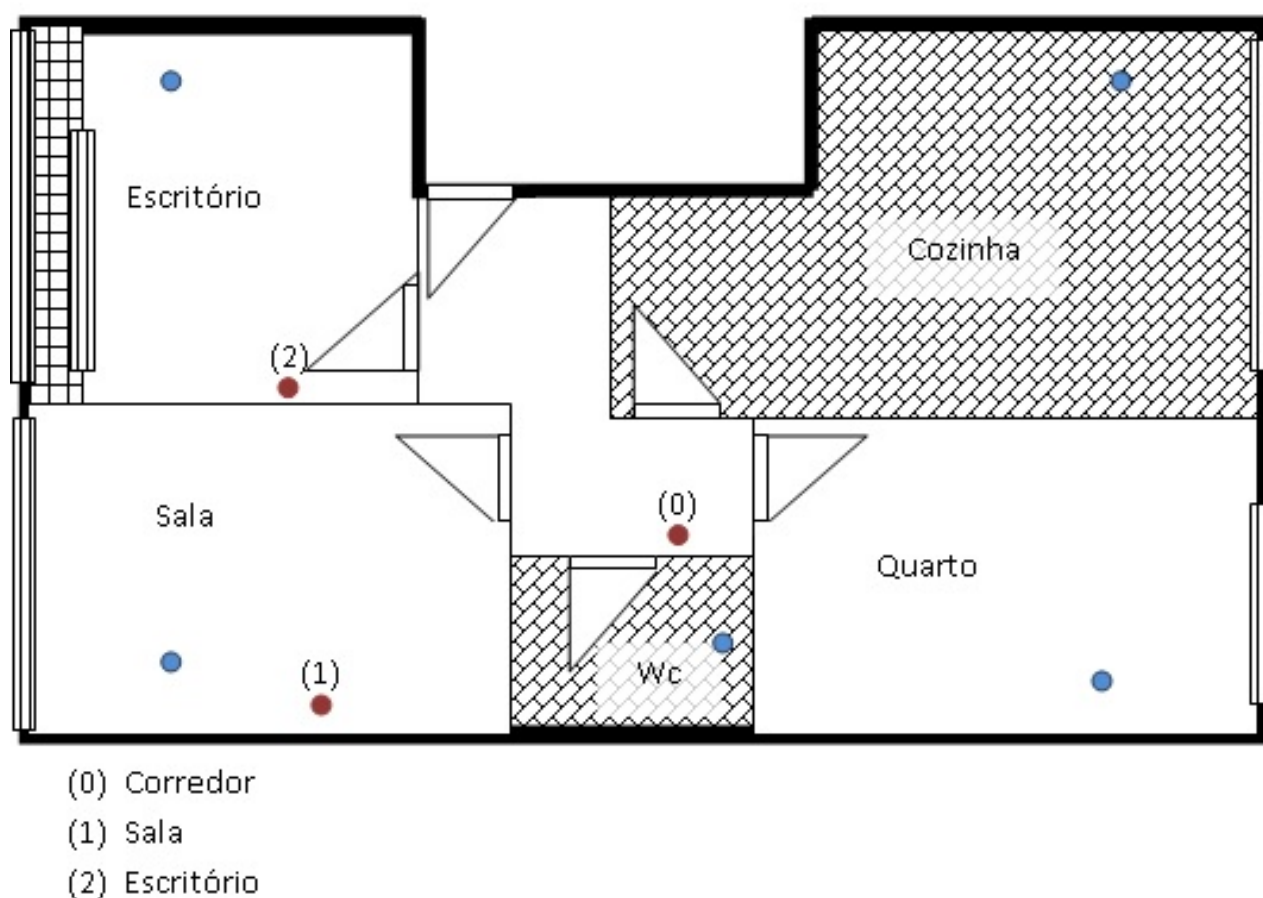
Seguidamente, experimentou-se usar um cabo de 5m (máximo teórico permitido pela especificação USB), sem sucesso. Seguem-se as mensagens emitidas pelo kernel:

Texto 5: Mensagens do kernel - timeout usando cabo de 5m

```
[ 148.965711] usb 1-1.3.4.2: new high-speed USB device number 39 using dwc_otg
[ 149.432019] usb 1-1.3.4.2: New USB device found, idVendor=148f, idProduct=7601
[ 149.432051] usb 1-1.3.4.2: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 149.505718] usb 1-1.3.4.2: reset high-speed USB device number 39 using dwc_otg
[ 149.606955] mt7601u 1-1.3.4.2:1.0: ASIC revision: 76010001 MAC revision:
76010500
[ 149.608575] mt7601u 1-1.3.4.2:1.0: Firmware Version: 0.1.00 Build: 7640 Build
time: 201302052146____
[ 150.006267] mt7601u 1-1.3.4.2:1.0: EEPROM ver:0b fae:00
[ 150.172990] mt7601u 1-1.3.4.2:1.0: Error: RX urb failed:-71
      (suprimidas mensagens semelhantes)
[ 150.175019] mt7601u 1-1.3.4.2:1.0: Error: MCU resp evt:0 seq:5-4!
[ 150.175035] mt7601u 1-1.3.4.2:1.0: Error: mt7601u_mcu_wait_resp timed out
```

A topologia de instalação final desta fase contempla 3 pontos de captura, compostos por 2 diferentes modelos de placas Wi-Fi, estando afastados entre si de pelo menos 2,5m, e em diferentes divisões da casa. Isto significa que têm entre si pelo menos uma parede, que funciona como elemento atenuador de sinal. Como guia visual, é apresentada a Planta da instalação, com pontos de recolha numerados e a vermelho escuro, com pontos de calibração a azul (úteis na determinação de valores máximos e mínimos para o interior do espaço).

Ilustração 1: Planta da instalação, com pontos de recolha numerados



Os pontos de captura estão equipados com antenas omnidirecionais, com antenas alinhadas sob diferentes eixos (perpendiculares entre si). Duas das placas Wi-Fi foram equipadas com antenas omnidirecionais convencionais do tipo “rubber duck” e outra foi equipada com uma extensão RP-SMA e uma antena omnidirecional de alto ganho (8 dBi), cuja ilustração se mostra abaixo. Estes 3 pontos de captura ligam-se a um hub USB alimentado. A instalação, contempla ainda cablagem USB e RF, ocupando pouco espaço.

Ilustração 2 Antena omnidirecional rubber duck alto ganho, com extensão RP-SMA



Ilustração 3: Placa com chipset MT7601, ligada a conector USB-A fêmea e a conector RP-SMA macho



A gama de valores encontrada foi entre -33 dBm captados pelo ponto 2 para dispositivos na mesma divisão, e -70 dBm captados pelo ponto 1 para dispositivos presentes no quarto. Numa instalação com 4 pontos de captura, verificou-se que a carga do sistema subiu continuamente até provocar instabilidade. Ao executar o processo de captura, não foi possível interagir com as placas Wi-Fi por forma a colocá-las em modo monitor respondendo com a mensagem “SIOCSIFFLAGS: Device or resource busy”. Apresentam-se abaixo as mensagens do kernel evidenciando o problema a manifestar-se após conexão da 4ª interface USB.

Texto 6: Mensagens kernel - instabilidade com 4 placas Wi-Fi

```
[ 280.485653] usb 1-1.3.3: new high-speed USB device number 21 using dwc_otg
[ 280.603833] usb 1-1.3.3: New USB device found, idVendor=148f, idProduct=5370
[ 280.603863] usb 1-1.3.3: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 280.603881] usb 1-1.3.3: Product: 802.11 n WLAN
[ 280.603897] usb 1-1.3.3: Manufacturer: Ralink
[ 280.603913] usb 1-1.3.3: SerialNumber: 1.0
[ 280.735674] usb 1-1.3.3: reset high-speed USB device number 21 using dwc_otg
[ 280.846780] ieee80211 phy4: rt2x00_set_rt: Info - RT chipset 5390, rev 0502
detected
[ 280.889199] ieee80211 phy4: rt2x00_set_rf: Info - RF chipset 5370 detected
[ 280.890398] ieee80211 phy4: Selected rate control algorithm 'minstrel_ht'
[ 280.892551] usbcore: registered new interface driver rt2800usb
[ 281.037221] ieee80211 phy4: rt2x00lib_request_firmware: Info - Loading firmware
file 'rt2870.bin'
[ 281.038377] ieee80211 phy4: rt2x00lib_request_firmware: Info - Firmware
detected - version: 0.29
[ 281.225631] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x06 failed for offset 0x7010 with error -110
[ 281.325635] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x06 failed for offset 0x0404 with error -110
[ 281.535648] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x07 failed for offset 0x02a0 with error -110
[ 281.735671] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x06 failed for offset 0x0208 with error -110
[ 281.835634] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x07 failed for offset 0x1000 with error -110
(suprimidas mensagens semelhantes)
[ 293.716024] ieee80211 phy4: rt2x00usb_vendor_request: Error - Vendor Request
0x07 failed for offset 0x1000 with error -110
[ 293.735705] ieee80211 phy4: rt2800_wait_csr_ready: Error - Unstable hardware
[ 293.735754] ieee80211 phy4: rt2800usb_set_device_state: Error - Device failed
to enter state 4 (-5)
```

Software de captura

Para efetuar a recolha das tramas (*probe requests*), foi escolhida a ferramenta tshark, da suíte wireshark, tendo também sido avaliadas as ferramentas aircrack-ng (da suíte homónima) e kismet server (da suíte kismet). A escolha da ferramenta de registo recaiu sobre o tshark, pois permite preservar a informação da interface que deu origem a cada trama captura. Esta informação é particularmente útil no cenário considerado, onde se usa cada interface como um ponto de captura distinto. Adicionalmente, o tshark apresenta

uma sintaxe de especificação de filtros de captura (e de exibição) mais poderosa, que permite minimizar o processamento e persistência de captações de tramas inúteis para o problema em questão (e.g. tramas do tipo *beacon*). Esta aplicação implementa os filtros de captura (opção -f) ao nível do driver, que corre em modo *kernel* – uma vantagem face ao uso dos filtros de visualização (opção -R) e à aplicação posterior de filtros, pois apresenta melhor performance. Assim, a boa prática da filtragem em tshark é preferir os filtros de captura tanto quanto possível. Outra das vantagens da utilização do tshark é a possibilidade de dividir ou agregar as funcionalidades de recolha (captura), filtragem e persistência em mais ou menos passos, conforme necessário: a modularidade advém do tshark aceitar o formato “pcap” (e seu sucessor, “pcapng”) tanto na entrada como na saída, podendo compor um *pipeline*. Assim, o tshark cumpre os requisitos pretendidos:

1. Execução durante longos períodos de tempo (rotação do ficheiro de registo);
2. Recolha seletiva das tramas relevantes (poderosa API de filtro);
3. Inclusão da indicação da interface de captura da trama;

Uma consequência da poderosa API de filtro, que permite composições, é a de poder recolher todas as tramas apenas mediante alteração do script utilizado. Esta alteração tem consequências no volume de dados a processar, mas também potenciais impactos na privacidade dos utilizadores sobre monitorização, conforme discutido no capítulo 6.3.

O formato de persistência escolhido foi o “pcapng”, que persiste as tramas captadas e meta-informação relevante, como a marca temporal de captura e a interface de origem. Este formato persiste alguma informação supérflua, o que é uma desvantagem numa solução com armazenamento limitado. Na prática, para tramas do tipo *probe request*, o espaço ocupado é, em média, de 160 bytes por trama. Apesar desta desvantagem, o formato é o standard *de facto* e é amplamente suportado.

Por forma a captar os probe requests em diferentes canais wireless, é necessário proceder ao *channel hopping*, colocando as interfaces wireless a varrer (“saltitar” entre) canais, passando determinado tempo em cada um. Assim, rentabilizam-se as interfaces, dado que cada uma consegue captar tramas em vários canais wireless. Para este efeito utiliza-se o kismet server, da suíte kismet, que disponibiliza esta funcionalidade com configuração avançada (em tempo e canal) com granularidade da interface. Esta ferramenta está configurada, por defeito, a “desfasar” o varrimento das diferentes

interfaces, por forma a evitar a situação de várias interfaces se encontrarem a captar o mesmo canal simultaneamente. O kismet server está também configurado, por defeito, com uma distribuição que favorece os canais Wi-Fi tipicamente mais utilizados com tempo adicional de permanência. Não sendo garantida a captura de todas as tramas, a solução de *channel hopping* usada foi desenhada de forma a minimizar as tramas não capturadas. A ferramenta poderia ser substituída pela ferramenta kismet drone, da mesma suíte, e com funcionalidade semelhante à usada. A ferramenta escolhida para efetuar *channel hopping* é superior à solução apresentada no Texto 3: versão simplificada do script de captura, pelas funcionalidades mencionadas, apesar de conceptualmente equivalente.

Protótipo (Fase 3)

Para explorar soluções alternativas, que suportem diferentes cenários de operação, foi adaptado o protótipo da Fase 2. Este protótipo tem como objetivo dar resposta aos seguintes requisitos adicionais:

1. Operação em ambiente ligado (com conectividade permanente ao subsistema de processamento);
 1. Implementar localização reativa (*real time* / *near real time*);
2. Implementar e permitir avaliação de algoritmo alternativo;

Deve continuar a responder a todos os requisitos das fases anteriores, acumulando assim os requisitos das anteriores fases com os novos requisitos acima apresentados.

A implementação desta fase é em tudo semelhante à anterior, modificando os parâmetros de invocação do tshark, de forma a emitir as tramas em formato de texto sendo estas enviadas para o *webservice* que implementa o sistema de localização.

Considerações de Operacionalização

Esta metodologia de recolha, ao propor uma solução ao nível físico, está sujeita a limitações das tecnologias utilizadas. Destas limitações, destacam-se:

1. Comprimento máximo teórico de 5m [US01] do cabo USB;

2. Alimentação das placas Wi-Fi no barramento USB;
3. Número de portas USB disponíveis;
4. Distância máxima útil de captura Wi-Fi (da placa escolhida);
5. Processamento das captações

Como solução para os pontos 1, 2 e 3, sugere-se o uso de *Hubs* USB com alimentação própria. A colocação de um *Hub* alimentado com um cabo de 3m, seguido de uma extensão USB de 3m até à placa permite duplicar a distância entre o mini-PC (ponto de recolha) e a antena (ponto de captura). Podem-se acrescentar *Hubs* alimentados adicionais até perfazer 30m, segundo a especificação. Para cobrir distâncias superiores, podem-se usar adaptadores de USB para protocolos de transporte adequados a maiores distâncias ou extensões USB próprias para o efeito.

Para solucionar o ponto 4, é necessário ter em conta que a distância máxima de captura depende de vários fatores, entre os quais o padrão de radiação da antena e o seu posicionamento face aos obstáculos. Usar placas Wi-Fi com antena amovível acrescenta uma série de possibilidades:

- usar uma antena de maior sensibilidade;
- usar uma antena direcional;
- acrescentar uma extensão (RP-SMA) entre a placa e a antena.

Aumentar a distância entre a placa e a antena pode permitir alcançar um melhor posicionamento, evitando possíveis obstáculos. Usar uma antena de maior sensibilidade permite tipicamente cobrir uma área maior. As antenas direcionais apresentam um padrão de radiação de diferente geometria. Combinando estas opções, conseguem-se idealizar soluções para mais desafios concretos de instalação física. Segue-se uma sugestão de padrão conceptual para instalação física de um ponto de recolha com um ou mais pontos de captura.

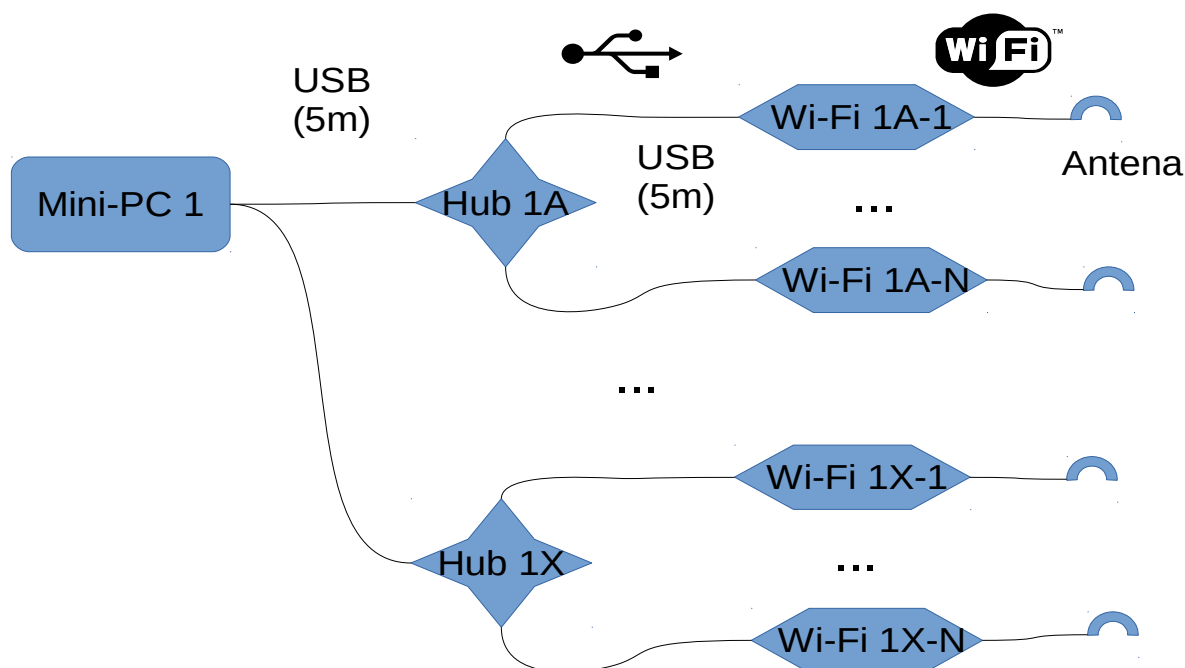
A limitação do processamento necessário para efetuar as captações enunciado no ponto 5, resulta numa combinação de fatores, para os quais contribuem:

1. a capacidade de processamento do mini PC;

2. o número de placas Wi-Fi (e outros dispositivos) presentes no barramento USB;
3. a combinação de driver e chipset do controlador USB;
4. a combinação de drivers e chipset das placas Wi-Fi;
5. outros fatores (carga do sistema, configurações do *kernel*, etc...);

Apresenta-se abaixo uma referência para implementação física genérica, sem considerar limites dependentes do hardware concreto.

Desenho 1: Exemplo conceptual de implementação física



- Raspberry Pi B+ (mini PC) :29,90€
- cartão microSD 4GB: 1€
- Fonte alimentação 1A c/ cabo: 1,66€
- hub USB alimentado: 9,52€

1 Preços unitários incluindo entrega em Portugal continental, consultados em eBay.com, em de Julho de 2015

- extensão USB 5m: 3,36€
- placa Wi-Fi USB chipset Ralink c/ antena externa (RP-SMA): 3,63€
- extensão RP-SMA 5m: 1,7€

A utilização de um cartão microSD (ou de outro suporte de armazenamento de memória flash) introduz constrangimentos de longevidade na operação do sistema. Desta forma, deve ser tido em conta que as memórias de tipo flash apresentam tipicamente uma degradação de tempo de vida e performance quando sujeitas a operações de escrita. Para minimizar este efeito, e com o objetivo de minimizar o número de escritas efetuadas, devem-se tomar medidas que se enquadram na competência de administrador de sistemas, entre as quais:

- Escolher, impactando os dispositivos, um agendador (*scheduler*) de IO apropriado;
- Afinar os parâmetros de agendamento de IO, ao nível do dispositivo;
- Utilizar um sistema de ficheiros adequado, como o F2FS [F2FS15];
- Utilizar parâmetros do sistema de ficheiros que minimizem as escritas;
- Afinar os parâmetros (globais) da cache do sistema operativo, garantindo tamanho suficiente e evitando/atrasando ao máximo as escritas;

4.1.3 Comparação

Para melhor diferenciar as diferentes abordagens (Infraestrutura existente Cisco vs Linux), segue-se uma comparação entre as duas soluções estudadas:

Desvantagens da solução Linux:

- Não aproveitamento de infraestrutura existente (caso ideal);
- Sem ponto central de configuração;

Vantagens da solução Linux:

- Otimização da instalação para recolha;
- Não interfere com o serviço Wi-Fi;
- Não é limitada pelo licenciamento;
- Pontos de recolha autónomos.

4.2 Processamento

Partindo dos dados recolhidos por uma das abordagens anteriores, e para chegar à localização dos dispositivos, e, por conseguinte, das pessoas, é necessário definir o sub-sistema de localização. Este sub-sistema tem como entrada os dados de captura e tem como saída informação sobre a localização de cada dispositivo.

4.2.1 Serviço de localização

A solução aqui descrita consiste no uso de um serviço de geolocalização com base na vista de dispositivo. A adaptação necessária para construir uma “imitação” da perspetiva de um dispositivo com base na perspetiva da infraestrutura seria desenvolvida por medida.

Para esse propósito, foi testado o serviço Mozilla Location Service (MLS), gratuito, construindo uma visão de dispositivo apenas com informação Wi-Fi. A API do MLS é compatível com a API do Google Maps, facilitando a inter-operabilidade e migração entre serviços. O caso de teste partiu da observação da força de sinal de APs no campus do ISEL, [zona com cobertura pelo MLS](#).

Texto 7: Interação HTTP de localização (MLS)

```
POST https://location.services.mozilla.com/v1/geolocate?
key=A_MINHA_CHAVE HTTP/1.1
User-Agent: Fiddler
Content-Type: application/json
Host: location.services.mozilla.com
Content-Length: 529

{"items": [ {
  "wifiAccessPoints": [ {
    "macAddress": "24:01:c7:76:08:40",
    "age": 0,
    "channel": 6,
    "signalStrength": -38
  }, {
    "macAddress": "24:01:c7:76:08:42",
    "age": 0,
    "channel": 6,
    "signalStrength": -39
  }, {
    "macAddress": "24:01:c7:76:08:62",
    "age": 0,
    "channel": 11,
    "signalStrength": -60
  } ] }
]}

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Date: Tue, 24 Mar 2015 20:07:46 GMT
Server: nginx
Content-Length: 67
Connection: keep-alive

{"location": {"lat": 38.7938, "lng": -9.1046}, "accuracy": 50000.0}
```

Esta opção revelou uma precisão muito baixa, na ordem das dezenas de Quilómetros, longe da precisão na ordem dos metros, necessária para a localização em interior. Uma opção não muito diferente desta é a de utilizar uma instalação local do [Mozilla Ichnaea](#), projeto *open source* que serve de base tecnológica ao MLS.

O Ichnaea utiliza trilateração e comparação entre observações de utilizadores e assinaturas de sinal geradas pelo sistema [Calc15].

O mesmo teste (3 APs na proximidade do dispositivo) contra a API de geolocalização da Google sugere que é usada a informação da localização do IP do cliente em vez da visão do cliente enviada. Forçar o uso da visão de cliente, usando o parâmetro correspondente não tem consequência junto do MLS, e reduz a precisão, no caso da Google. Face à falta de alternativas (preferencialmente gratuitas), e sendo necessário dar resposta ao desafio da recolha, a abordagem híbrida foi posta de parte.

4.2.2 Solução de localização do tipo reconhecimento de padrões

Uma possível solução para o subsistema de processamento dos dados recolhidos é desenvolver à medida uma solução de localização que permita, face às amostras recolhidas, atribuir-lhes uma localização (variável dependente). Uma solução desta natureza, classifica-se, quanto ao processamento das métricas, como sendo reconhecimento de padrões (diferente, por exemplo, da triangulação).

Para o problema apresentado, o de localização num espaço previamente conhecido, e considerando os requisitos, considera-se adequada a identificação do espaço ocupado por cada dispositivo num determinado instante no tempo. A classe de algoritmos que se ajusta a este cenário é a classe de algoritmos de classificação.

Para este propósito, será conveniente ter uma base de dados que sirva de repositório para as captações e para os dados de treino do modelo. Os dados de treino para uma solução desenvolvida por medida podem ser recolhidos no terreno (*ground truth*) por forma a maximizar a sua fiabilidade. Será também necessário automatizar a forma como as novas captações são colocadas na base de dados, desenhando um ETL(T).

Esperando-se um grande volume de dados e sem prever requisitos de atomicidade em operações entre várias captações ou integridade referencial, são, à partida, elegíveis para repositório os tradicionais SGBD relacionais, bem como sistemas não-relacionais, que tipicamente se adequam a cenários de Big Data como este. No entanto, por forma a implementar o cruzamento das amostras com os dados do modelo de treino, a linguagem SQL e os SGBD relacionais afiguram-se adequados.

Para comprovar a possibilidade de implementação bem como ajudar a validar a qualidade dos dados recolhidos, foi iniciada a implementação de um protótipo deste subsistema, composto por:

- uma base de dados mongoDB;
- um modelo de treino construído com base em recolha de campo;
- um *script* em R.

Dado que a manipulação dos dados é feita do lado do cliente, esta solução foi preterida a favor de uma solução que permita delegar no servidor a manipulação dos dados.

Implementação (Fase 2)

Para tornar mais úteis as captações efetuadas, foi implementado um processo de importação de dados, recorrendo a scripts CLI Windows (.cmd), como o Script Windows CLI para ingestão de tramas (página 76). Os dados relativos às tramas foram importados para uma base de dados relacional Postgres, estando disponíveis para interrogações. Desta forma, e tendo por base recolhas prévias, será possível, por comparação, determinar a localização de cada dispositivo no instante de cada trama registada. Evidencia-se assim a necessidade de efetuar correlações entre os dados do modelo de treino (*ground truth*) e os dados recolhidos com vista à localização de dispositivos, bem como correlações entre várias tramas dos dados a processar. Perante esta necessidade, escolheu-se o SQL como linguagem de implementação dos algoritmos, pois permite simultaneamente interrogar diferentes conjuntos de dados, relacionando-os. O Postgres surgiu como solução SQL multi-plataforma (o que permite uma maior flexibilidade na instalação) e extensível, podendo assim responder a necessidades específicas de tratamento de dados.

A ingestão dos dados sobre as tramas no sistema de localização faz-se inserindo em tabela própria (“Input”) tuplos correspondentes a tramas, com os seguintes atributos:

1. “MAC” – Identificação do dispositivo emissor;
2. “dBm” – Força de sinal da captura;
3. “Timestamp” – Marca temporal da ocorrência de captura da trama;
4. “Interface” – Identificador do ponto de captura (parte da identificação do local);

5. “Source” – Identificador do ponto de recolha (parte da identificação do local).

A *ground truth* segue um formato em tudo semelhante ao formato da ingestão, acrescentando um atributo “Place”, que identifica o espaço à qual está associada a trama recolhida, parte do modelo construído recorrendo ao Script SQL para criação do modelo (pág. 77).

A recolha da *ground truth* foi efetuada com o sistema de captura descrito no tópico Protótipo (Fase 2) num prédio de 4 andares. Verificou-se que as tramas não são captadas exatamente no mesmo instante no tempo em todos os pontos de captura. Isto deve-se, entre outros fatores, ao breve instante de tempo em que cada interface despende a efetuar a comutação de canal no processo de *channel hopping*. Esta situação impõe um desafio adicional, dado que a generalidade dos procedimentos de localização com base na força de sinal estudados aplicam o reconhecimento de padrões apenas a uma distância no espaço da força de sinal, considerando que cada amostra corresponde a um só momento no tempo. O problema aqui apresentado seria também relevante na aplicação de soluções que usem como métrica o tempo de chegada (TOA – Time Of Arrival), dada a baixa resolução temporal das amostras. Tendo em conta que cada trama pode não ser observada em todos os pontos de captura ao alcance, torna-se necessário solucionar este problema. A solução adotada foi a de considerar como pertencendo à mesma amostra as tramas recolhidas num curto período de tempo: 3,6 segundos, que equivalem sensivelmente a 5m percorridos à velocidade típica de deslocação a pé. Esta decisão é válida para a construção de amostras a localizar, e suporta-se na prática, considerando que tipicamente os equipamentos a localizar emitem várias tramas no intervalo de tempo considerado.

A solução de filtragem anteriormente descrita é implementada através de uma função que, dado um endereço MAC, um ponto no tempo e um limite temporal (especificado em segundos), devolve, dentro das tramas ingeridas, a média da força de sinal para cada um dos diferentes pontos de captura (de diferentes pontos de recolha) das tramas correspondentes ao endereço MAC e pertencentes ao intervalo de tempo centrado no tempo especificado em argumento.

Um processo semelhante é realizado sobre os dados de treino, agregando por um ponto no tempo um grupo de tuplos centrados nesse mesmo ponto. Para disponibilizar

este pré-agregado recorreu-se a uma vista materializada, o que se traduz em melhorias na velocidade de execução das interrogações perceptíveis à escala humana.

A solução adotada utiliza o SGBD Postgres 9.4, e permite delegar neste o processo de geolocalização. A solução alavanca em funcionalidades chave do Postgres 9.4, como as funções definidas pelo utilizador e vistas materializadas. Apesar disto, uma funcionalidade equivalente poderia ser conseguida em ambientes mais limitados como o SQLite.

O algoritmo de reconhecimento de padrões (do tipo classificação) implementado em SQL foi o K-Nearest Neighbours, considerando a distância euclidiana no espaço de força de sinal (dBi), com $K = 13$.

Nesta fase de protótipo, a interface de geolocalização é exposta como procedimento armazenado no SGBD, facilitando a sua utilização por componentes externos, como camadas de acesso a dados (no contexto de uma aplicação), ou *scripts*.

Para facilitar o processo de recolha da *ground truth* (fase offline) para a instalação efetuada, foram também desenvolvidos *scripts* auxiliares que recolhem dados ingeridos para formar o conjunto de dados de treino e conjuntos de dados de teste (pág. 76).

4.2.3 Solução de localização do tipo n-lateração

Afim de avaliar de um algoritmo alternativo ao de tipo reconhecimento de padrões, e tendo sido escolhida a base (n-lateração) foi necessário implementar um sistema que o implementasse. A avaliação incide sobre a operacionalização de um sub-sistema deste tipo, ou seja, como ele se integra com os demais sub-sistemas e que requisitos impõe para a sua operação.

Dado que a n-lateração é aplicada à métrica força de sinal, é necessário usar um modelo de distância para transformar o domínio da força de sinal do domínio da distância. Um dos modelos propostos na literatura consultada é a adaptação do modelo Hata-Okumura [PLM08], originalmente usados para grandes distâncias em áreas urbanas. A adaptação tem como variáveis independentes, para além da distância (em metros, na adaptação usada), a frequência do sinal (em MHz). Esta adaptação do modelo contempla

dois parâmetros de ajuste: S (relativo ao ruído do sinal) e N (expoente de atenuação), ambos medidos em dB. A literatura avança também uma gama de valores típicos para os parâmetros, onde S varia entre 3 e 20 dB, e N varia entre 2 (para espaço desobstruído) e 5 (para ambientes com muitos obstáculos).

Diferentes adaptações incluem os valores das atenuações e ganhos de transmissor e recetor, bem como da potência de emissão como parâmetros adicionais, ou não, contribuindo apenas para influenciar N.

Implementação (Fase 3)

A terceira fase do protótipo apresenta um sistema de localização reativo baseado em n-lateração. Este expõe interface HTTP para ingestão das tramas capturadas, bem como interface adicional para consumo do resultado da geolocalização.

A interface de ingestão de tramas aceita pedidos HTTP de verbo POST e *content-type* “text/tab-separated-values”. A ingestão de cada uma das tramas espoleta um processo de associação da informação sobre o ponto de captura que a originou (posição e parâmetros do modelo de distância), aplicação do modelo de distância e aplicação da n-lateração.

O resultado final é a localização estimada do emissor da trama ingerida, que é exposto por HTTP. Esta composição de operações que resultam num fluxo de ocorrências de localização é expressa, na implementação proposta, no padrão *observable* e baseia-se na biblioteca Reactive Extensions, conforme detalhado abaixo. As coordenadas expostas nas estimativas usam o sistema referencial WGS84, escolhido pela sua ubiquidade.

O referencial de trabalho dos modelos de distância e da n-lateração são geometrias (com unidade em metros), pelo que será necessário posteriormente projetar para WGS84 as coordenadas do referencial utilizado. Foi escolhido trabalhar com UTM – Universal Transverse Mercator, um sistema cujas coordenadas são expressas em metros, simplificando a aplicação dos modelos de distancia.

Simplificações

Para simplificação, a frequência usada no modelo de distância é fixa (centrada na gama do Wi-Fi) – este é um aspeto a melhorar numa implementação mais refinada, assumindo que está disponível a frequência de emissão da trama capturada (ou, como alternativa, a frequência sintonizada pelo ponto de captura).

Outra simplificação relevante é a utilização de uma zona fixa do sistema de coordenadas UTM, correspondente a Portugal (zona 29). Este aspeto poderá ser melhorado colocando em configuração a zona UTM em uso, balizando a área de operação a uma só zona UTM.

Padrão *Observable* e Reactive Extensions

Para tratar o fluxo de tramas capturadas, transformando-o num fluxo de estimativas de geolocalização para cada emissor, foi usado o padrão *observable*. O conceito nuclear deste padrão é o observável (*observable*), sendo este envolvido na maior parte das operações. Este conceito resume-se um fluxo de elementos. A biblioteca usada para suportar as manipulações de observáveis é a Reactive Extensions, uma biblioteca muito completa que permite as principais operações:

- a transformação de vários tipos de fontes num observável (evento, invocação de método, alarme);
- composições de diversos observáveis (N-1 e 1-N);
- transformações de observáveis (1-1);
- reação sobre elementos de um observáveis;

Net Topology Suite e ProjNet

A aplicação do algoritmo de n-lateração é parcialmente suportada pelo uso de tipos geométricos através da biblioteca NET Topology Suite. Esta biblioteca é uma transposição

de Java Topology Suite para tecnologia “.NET”, implementando a especificação “Simple Features Specification for SQL” do *Open GIS Consortium*. O conceito fundamental da biblioteca é a geometria, com suporte no tipo Geometry. Esta biblioteca disponibiliza operações geométricas como:

- obtenção do centro;
- obtenção da intersecção;
- obtenção dos pontos que minimizam a distância entre duas geometrias;

Adicionalmente, foi usada a biblioteca ProjNET4GeoAPI, que interage com a NET Topology Suite, por forma a projetar os pontos geométricos de um referencial em metros (para aplicação do modelo de distância) em pontos geográficos num referencial espacial.

A Net Topology Suite foi utilizada para implementar a n-lateração, segundo a descrição que segue, nos pontos 3 e seguintes:

1. Receber a trama capturada;
2. Obter o sensor correspondente;
3. Aplicar o modelo de distância;
4. Atualizar a memória de tramas do mesmo MAC;
5. Obtenção das geometrias (circunferências) das tramas;
6. Intersectar as geometrias;
7. Obter o ponto melhor candidato (minimização da distância);

OData

O padrão OData vem adicionar uma sintaxe de expressão de filtros com suporte a geometrias numa interface HTTP [ODATA14]. Isto permite declarar interesse na subscrição de eventos com uma especificidade tão rica quanto desejável. Utiliza-se nesta implementação apenas um subconjunto das convenções de URI da quarta versão da especificação OData, não se dando suporte total à especificação, o que é particularmente

evidente no formato utilizado na resposta, que é do tipo *text/event-stream* ao invés de *application/json*, *application/atom+xml* ou equivalentes.

Por exemplo, é possível, nesta implementação, subscrever eventos de localização para quando um determinado MAC se encontrar a uma distância superior a um valor X^1 de um determinado ponto, num intervalo de tempo. A declaração de interesse específico concretiza-se especificando o parâmetro “\$filter” da query string.

Seguem-se exemplos para alguns casos de uso:

Caso de uso	Filtro
Detetar aproximação ao ponto GPS N 38.79, W 9.11.	\$filter=MAC ne null&geo.distance(Coordinates, geography'Point(38.79, -9.11)') lt 1
Seguir o percurso da pessoa que tem os dispositivos com MACs 00:73:8d:9e:55:6a e 30:0d:43:03:41:58.	\$filter=MAC eq '00:73:8d:9e:55:6a' or MAC eq '30:0d:43:03:41:58'
Localizar um equipamento perdido (assumindo que tem o Wi-Fi ligado).	\$filter=MAC eq '00:73:8d:9e:55:6a'
Verificar se um equipamento está ligado / tem o WiFi ligado.	idem
Verificar acesso não autorizado (MACs autorizados no filtro) à área definida pelo polígono descrito.	\$filter=MAC ne '00:73:8d:9e:55:6a' and MAC ne '30:0d:43:03:41:58'&geo.intersects(Coordinates, geography'POLYGON ((38.78 -9.11, 38.79 -9.11, 38.79 -9.12, 38.78 -9.12, 38.78 -9.11)))'

Tabela 1: Exemplos de casos de uso e respetivos filtros OData v4

No sistema implementado foi utilizada a biblioteca OdataLib, biblioteca disponibilizada pela Microsoft que suporta a implementação de um serviço completo OData v4 na *framework* ASP.NET WebAPI 2.

1 A unidade de distancia é a do sistema referencial

Server Sent Events

A utilização do padrão Server Sent Events [SSE15] permite expor o fluxo de localizações seguindo o padrão *observable*, em HTTP, na forma de notificações via *push*. O consumo destes dados é facilitado pelo suporte existente nos browsers atuais (API javascript), e pelas vantagens decorrentes da utilização de um padrão. A implementação deste padrão, em ASP.NET WebAPI2 é conseguida através do tipo PushStreamContent, que utiliza o fluxo da resposta HTTP.

Notas

A implementação tende a segregar diferentes responsabilidades (ingestão, enriquecimento, modelo de distância, n-lateração e publicação de estimativas) em diferentes módulos para aumentar o desacoplamento e desta forma simplificar a adaptação deste sistema a diferentes realidades.

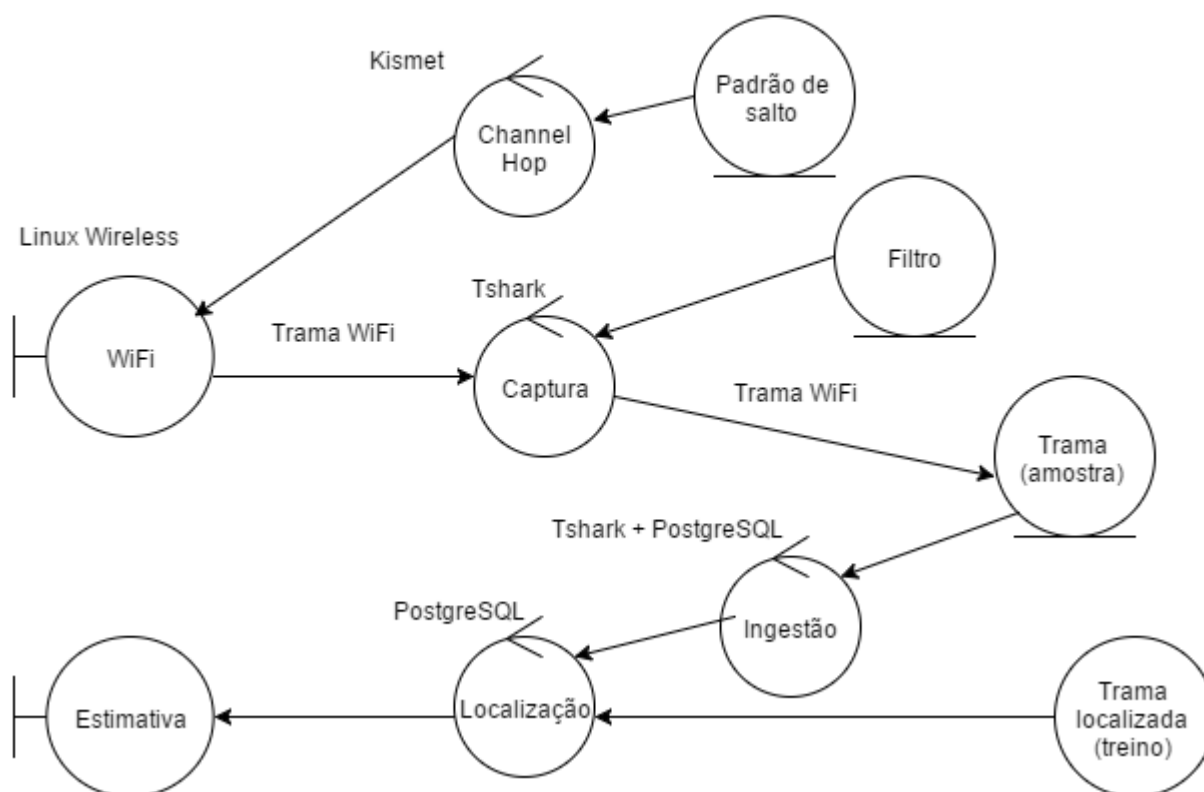
4.3 Solução Global (Fase 2)

Considera-se útil enquadrar novamente as duas componentes da solução, por forma a ter o contexto geral do funcionamento integrado. Revendo a arquitetura da solução, considera-se responsabilidade do sistema de recolha disponibilizar informação sobre as tramas relevantes, efetuando a captura das mesmas; A ingestão desses dados e localização dos dispositivos neles mencionados e estimativa de localização é responsabilidade do sistema de processamento. A interface entre os dois sistemas é feita no formato pcapng, mantendo até esse ponto toda a informação sobre as tramas recolhidas.

Segue-se abaixo uma imagem que mostra os diferentes componentes de software que constituem a solução e relação entre eles.

Ilustração 4: Processo de geolocalização (mapa de rádio)

Processo de geolocalização



A forma de obtenção do conjunto de dados de treino (Trama localizada na ilustração) pode ser, por exemplo, uma de duas expressas acima: através de importação direta recorrendo a *script* Windows CLI ou catalogando tramas ingeridas com a sua localização, recorrendo a comandos SQL como os exemplificados em Exemplo de script para construção da ground truth (pág. 76).

4.4 Solução Global (Fase 3)

A solução global da terceira fase deste trabalho, apresenta pela primeira vez uma interface “amigável” para o consumo da geolocalização, ao expor para o efeito uma API web (ou *web service*). Esta interface HTTP segue padrões relevantes da área, como descrito nos pontos OData e Server Sent Events. Segue-se um exemplo de interação com esta interface:

Texto 8: Interação HTTP de subscrição de estimativas de localização

GET

[http://localhost:63119/api/Location?\\$filter=MAC%20eq%20%2700:73:8d:9e:55:6a%27&geo.distance\(Coordinates,%20geography%27Point\(38.794381715947182,%20-9.1108818794661452\)%27\)%20lt%2010](http://localhost:63119/api/Location?$filter=MAC%20eq%20%2700:73:8d:9e:55:6a%27&geo.distance(Coordinates,%20geography%27Point(38.794381715947182,%20-9.1108818794661452)%27)%20lt%2010)
HTTP/1.1

Host: localhost:63119

Connection: keep-alive

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.103 Safari/537.36

DNT: 1

Accept-Encoding: gzip, deflate, sdch

Accept-Language: pt-PT,pt;q=0.8,en-US;q=0.6,en;q=0.4

HTTP/1.1 200 OK

Cache-Control: no-cache

Pragma: no-cache

Content-Type: text/event-stream

Expires: -1

Vary: Accept-Encoding

Server: Microsoft-IIS/10.0

X-AspNet-Version: 4.0.30319

X-SourceFiles: =?UTF-8?B?

QzpcVXNlcnNcYWZhZ2FcZG9jdW1lbnRzXHZpc3VhbCBzdHVkaW8gMjAxNVxQcm9qZWNOc1xMb2NhdGlvblxMb2NhdGlvbjNcYXBpXExvY2F0aW9u?=
hdGlvblxMb2NhdGlvbjNcYXBpXExvY2F0aW9u?=
X-Powered-By: ASP.NET

X-Powered-By: ASP.NET

Date: Wed, 10 Feb 2016 20:04:10 GMT

data: {"Coordinates":{"Latitude":-

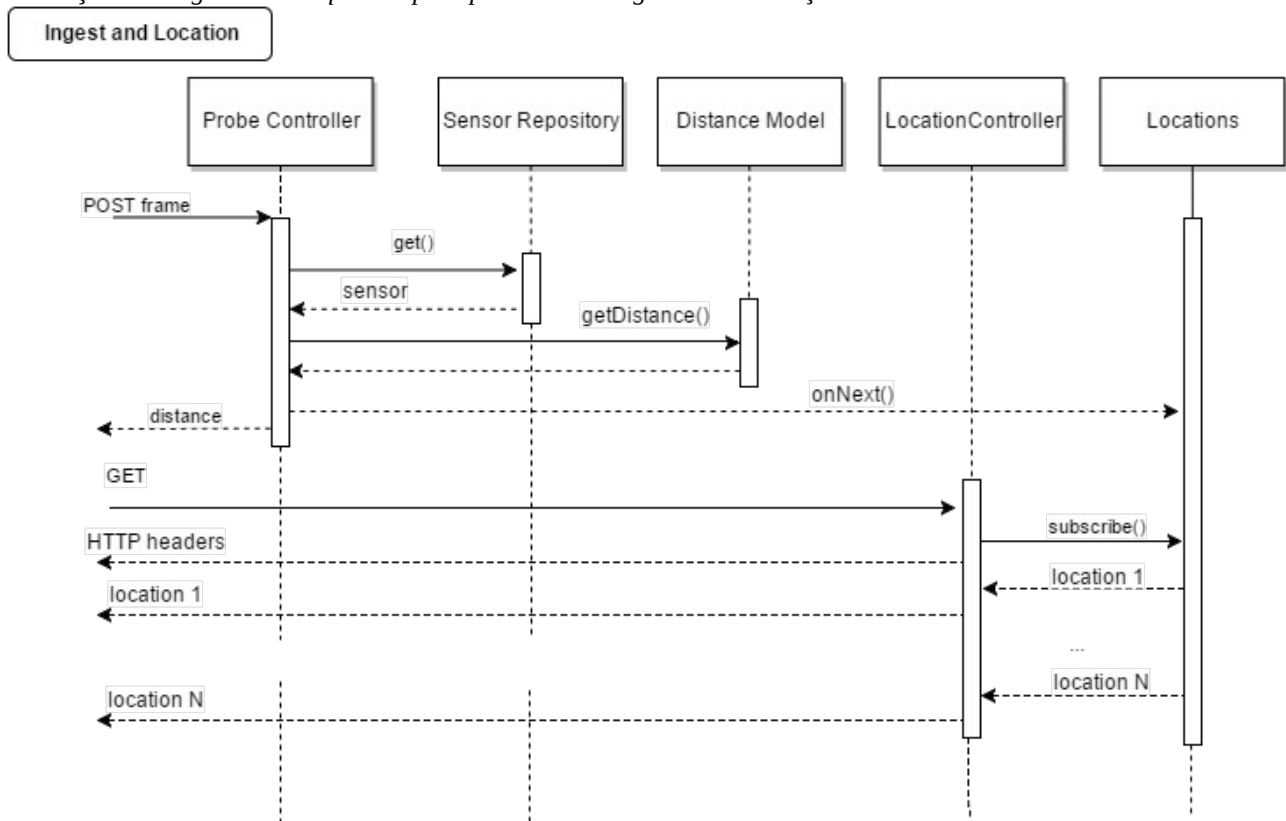
9.1108912894344183,"Longitude":38.794398451963104,"IsEmpty":false,"Z":"NaN",
"M":4.7313177256334766,"CoordinateSystem":
{"EpsgId":4326,"Id":"4326","Name":"WGS84"}}, {"MAC":"00:73:8d:9e:55:6a", "Time
stamp":"2016-02-10T20:04:16.804Z", "Tag":null}

data: {"Coordinates":{"Latitude":-

9.1108912894344183,"Longitude":38.7943984519631,"IsEmpty":false,"Z":"NaN", "
M":5.4636404733927808,"CoordinateSystem":
{"EpsgId":4326,"Id":"4326","Name":"WGS84"}}, {"MAC":"00:73:8d:9e:55:6a", "Time
stamp":"2016-02-10T20:04:16.806Z", "Tag":null}

Apresenta-se o diagrama de sequência dos processos de ingestão e de consumo das estimativas de localização através da subscrição (declaração de interesse) dos eventos, evidenciando como um processo desencadeia o outro.

Ilustração 5: Diagrama de sequência para processos de ingestão e subscrição



5

Resultados Obtidos

O primeiro resultado obtido relevante prende-se à avaliação da viabilidade de operar um sistema de localização de baixo custo com as arquiteturas propostas, que se avalia como positiva, dado que este se manteve em operação regular. A operação deu-se durante vários períodos superiores a 24 horas; O sistema foi monitorizado e apresentou valores de utilização de CPU (18%), user load (0.5), IO do armazenamento e ocupação de memória (38%) estáveis e baixos ou perto do residual, com exceção do user load que apresentou um valor médio; O sistema testado foi um RaspberryPi 2 sem dissipador, numa caixa, e manteve-se a cerca de 43°C durante o dia. Estes valores devem ser enquadrados com o número de tramas processadas, cuja média é de aproximadamente 1 trama por segundo.

Faltam, no entanto, elementos de avaliação prática que permitam aferir a real escalabilidade do sistema, pelo que apenas pode ser classificada como indeterminada.

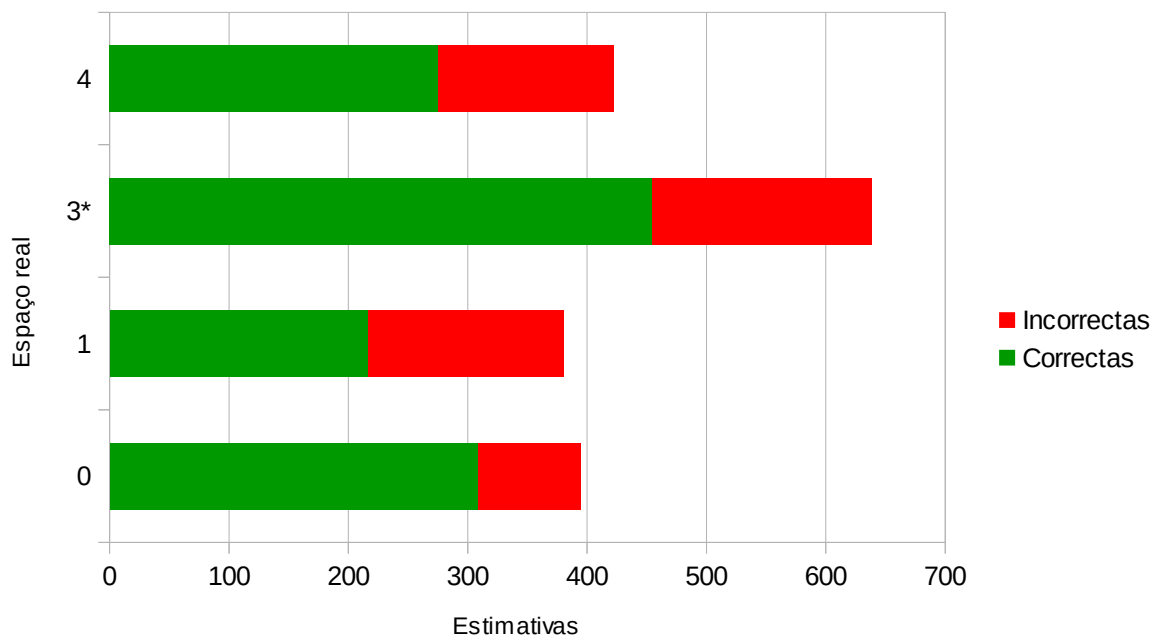
A configuração do protótipo da fase 2 permitiu identificar corretamente, de entre 4 espaços distintos separados de pelo menos 3,2m (correspondendo a diferentes pisos do prédio), 68% das vezes. Para obter o indicador da qualidade da estimativa apresentado, utilizou-se como conjunto de dados de teste o mesmo conjunto de dados de treino, efetuando-se a interrogação abaixo:

Texto 9: Interrogação - número de estimativas correctas e incorrectas

```
select count(*), "Actual" = "Estimation"
from(
    select "Place" as "Actual", "Location"("MAC", "Timestamp") "Estimation"
    from "FilteredTraining"
) as "Base"
group by "Actual" = "Estimation";
```

Detalhe dos resultados obtidos pode ser consultado no Gráfico de correção da estimativa por espaço, construído com interrogação semelhante à anterior. Os espaços alinham-se formando uma reta transversal ao plano formado pelos 3 pontos de captura, pelo que se pode considerar o pior caso para um algoritmo de triangulação. A escolha dos espaços levou, propositadamente, a esta situação, para testar o protótipo perante situações mais difíceis.

Ilustração 6: Gráfico de correção da estimativa por espaço



Observando o gráfico supra-mencionado, notam-se diferenças entre as taxas de erro para diferentes espaços físicos. Esta constatação sugere proximidade entre os diferentes espaços físicos, no espaço vetorial de força de sinal. Tal acontece quando os diferentes pontos de captura registam tramas oriundas de diferentes espaços físicos com iguais valores de força de sinal. Na tabela seguinte, Espaço estimado vs Espaço real – Fase 2,

contam-se as ocorrências para cada espaço estimado, agrupadas por espaço real de origem da captura.

Espaço Real Estimado	0	1	3	4
0	309	116	23	26
1	75	217	5	53
3	0	12	455	68
4	11	36	156	275

Tabela 2: Espaço estimado vs Espaço real – Fase 2

Das várias formas tentadas de obter uma estimativa de localização num espaço previamente conhecido, a combinação de um ponto de recolha físico baseado em mini-PC Linux, juntamente com um processo de ingestão para um sistema de reconhecimento de padrões implementado em SQL permitiu resultados satisfatórios. Esta abordagem, alavancando numa arquitetura extensível, permite perspetivar melhorias a aplicar ao protótipo do projeto.

Para a mesma configuração de pontos de recolha, mas para determinar a divisão da casa onde o dispositivo se encontra, foram efetuadas, em separado novas recolhas de treino e de teste. O objetivo desta segunda recolha foi testar o comportamento de um modelo de treino com captura de um *smart-phone* contra um teste efetuado com um *smart-watch*. Este teste foi efetuado no sentido de perceber se o modelo de treino poderia ser utilizado para localizar dispositivos com diferentes características de emissão, especificamente no caso de dispositivos *wearable*. As diferentes características de emissão resultam de maiores constrangimentos de capacidade da bateria, de aquecimento e absorção de radiação pelo utilizador, que se traduzem em menores consumos, e, no que toca à interface Wi-Fi, a menores potências de emissão. A uma menor potência de emissão para estes dispositivos, juntam-se maiores constrangimentos no posicionamento de antenas, resultando em menores valores de RSS nos pontos de captura. Os resultados obtidos mostraram que para determinar a localização (de entre as 5 divisões da casa), apenas 40% das (958) vezes um *smart-watch* é corretamente

localizado recorrendo a um conjunto de dados de treino obtido a partir de um smart-phone. Para localizar um *smart-phone* nas mesmas condições, o sucesso é de aproximadamente 62% (para um conjunto de 924 amostras).

Uma das melhorias possíveis a implementar passa por otimizar a colocação dos pontos de captura, para além das boas práticas indicadas na bibliografia Cisco [LTA08], capítulo 5.

Na avaliação do protótipo da fase 3, analisaram-se as estimativas obtidas quanto ao espaço físico (divisão da casa) em que se encontram. Considera-se que uma estimativa se encontra num espaço físico quando esta está contida no polígono que o representa. Para possibilitar a comparação, foram descartadas as estimativas que não estão contidas em nenhum dos espaços físicos, até porque esse caso é impossível no algoritmo usado na fase 2.

O protótipo da fase 3 apresentou resultados globalmente piores no que diz respeito à localização, conforme a tabela abaixo:

Espaço Real Estimado	A	B	C	D
A	6	0	0	0
B	82	85	66	63
C	8	15	33	29
D	4	0	0	8

Tabela 3: Espaço estimado vs Espaço real – Fase 3

Este resultado não invalida de todo a arquitetura do sistema, mas põe em causa a combinação do modelo de distância e algoritmo de localização, que deve ser revista, dado que a mesma distribuição da rede de sensores (pontos de captura) apresentou, para um algoritmo alternativo, melhor desempenho. Retira-se também desta observação que a

validação cruzada entre os resultados de diferentes algoritmos ajuda a enquadrar algoritmos alternativos.

Foi também possível observar o comportamento de envio de probe-requests por parte de smartphones: Um smartphone android 4.4 envia probe-requests, com os seguintes períodos:

Situação	Período
Ecrã de bloqueio	30s
Ecrã de configuração WiFi (definições)	5s
Ecrã desligado	30m

Tabela 4: Frequencia Probe-Requests

Esta análise ajuda a suportar uma decisão sobre o uso exclusivo de probe-requests, mediante a frequência de amostragem desejada e o comportamento esperado dos dispositivos a localizar. No entanto, é relevante ter em conta que vários fatores ponderam o envio de cada tipo de tramas Wi-Fi, entre os quais:

- o sistema operativo;
- o estado de utilização do mesmo;
- a configuração do equipamento.

Quanto a outras classes de dispositivos (wearables, tablets, laptops), é legítima a expectativa que os valores de período entre emissão de tramas aumente com o aumento das restrições energéticas do dispositivo, ainda que de forma distinta para os diversos tipos de tramas.

6

Conclusões e Trabalho Futuro

6.1 Conclusões

O projeto afigura-se viável, ainda que restem desafios relevantes para uma solução mais evoluída. A correta identificação de espaços separados de um valor inferior a 5m no caso desfavorável de espaços alinhados perpendicularmente aos pontos de captura foi possível para 68% das captações.

Muitas das suposições iniciais relativas à exequibilidade do projeto sobre plataformas concretas, como a instalação Cisco e a mini-PC Linux revelaram-se falsas: A execução sobre Cisco revelou constrangimentos de compatibilidade com o hardware de rede. A execução sobre mini-PC Linux, apesar de possível com custos razoáveis, tem também constrangimentos ao nível do hardware compatível.

6.2 Trabalho futuro

Espera-se que os desafios e problemas descritos neste relatório sejam relevantes para implementações similares. Seguem-se alguns desafios sem uma proposta de solução no âmbito deste projeto:

6.2.1 Recolha

No âmbito da recolha, um passo natural a tomar seria testar uma instalação com mais do que um ponto de recolha. Uma instalação deste tipo permitiria avaliar o comportamento da solução sobre uma área maior, e tomar contacto com decisões de maior cobertura versus maior precisão na área coberta. Essa avaliação ajudaria a estimar infraestrutura para uma instalação concreta.

Para melhor suportar a decisão sobre o tipo de tramas a utilizar, será relevante um maior estudo da frequência de tramas por tipo, para as diferentes classes de dispositivos a localizar, tendo em conta o sistema operativo e as definições do dispositivo. Surgem como relevantes as definições do dispositivo relacionadas com a rede Wi-Fi e gestão de energia.

A fim de garantir o baixo custo da solução, será útil avaliar a possibilidade de usar hardware alternativo, em particular no caso do mini-PC. Algumas arquiteturas que se afiguram como candidatas naturais em dispositivos embarcados de baixo custo são as ARMv6 e MIPS, ainda que correndo o risco de se tornarem obsoletas rapidamente. Avaliar outras placas Wi-Fi quanto à sua compatibilidade com o sistema, potencialmente contribuirá para um aumento das combinações de hardware possíveis.

Outro melhoramento relevante é a otimização para instalações desligadas, desenvolvendo uma solução de cópia automática para pen drive USB, simplificando o processo desligado de recolha.

Num sentido oposto, possibilitando aplicações alternativas, seria interessante otimizar a solução para uma instalação ligada, fazendo chegar imediatamente as captações até ao sub sistema de processamento. Isto possibilitaria a implementação de localização em tempo real, útil também em âmbitos para além do suporte de decisão de colocação de publicidade. Seria, portanto, vantajoso, implementar um processo de registo histórico (total ou parcial) para a solução de recolha em tempo real.

Por forma a fornecer mais informação no momento da decisão da colocação da publicidade, será também relevante associar à informação de cada trama o fabricante da

placa Wi-Fi que a produziu. Esta informação permite melhor conhecer o dispositivo que a originou, e, por conseguinte, afigura-se útil na decisão do tipo de publicidade a mostrar. Desta forma, associar a cada trama o fabricante da placa Wi-Fi permitiria conhecer e segmentar melhor os utilizadores do espaço sob análise.

Solucionar o problema de performance tornaria a solução ainda mais versátil e rentável, aumentando o número de pontos de captura por ponto de recolha. O primeiro passo a seguir será diagnosticar o problema atual, percebendo a causa da elevada carga do sistema.

Aumentar a quantidade de pontos de captura em cada ponto de recolha, usando uma combinação de hardware que permita exceder as 3 placas Wi-Fi conectadas, poderá permitir melhorar a estimativa para casos em que dois espaços (classes) são definidos pelo mesmo vetor no espaço vetorial da força de sinal.

Para garantir o baixo custo da solução, com preocupações para o custo total da posse, é relevante fazer uma medição do consumo elétrico da operação do subsistema de recolha, percebendo como este se altera em função da composição utilizada e da escala do mesmo.

6.2.2 Processamento

Os algoritmos de reconhecimento de padrões escolhidos para implementar são bastante simples, mas poderão ser substituídos. Será útil avaliar algoritmos alternativos, por forma a encontrar melhores soluções. Concretamente, dever-se-á encontrar forma de localizar com precisão dispositivos com diferentes características de emissão (desde *wearables* de muito baixa potência até computadores portáteis).

Por forma a tornar mais útil a utilização deste sistema com o objetivo de obter informações estatísticas sobre a distribuição de pessoas num espaço físico previamente conhecido (ao longo do tempo), será necessário desenvolver interrogações que tenham em conta a localização de vários dispositivos. Esse desenvolvimento poderá suportar relatórios de localização, por espaço e num dado período temporal, para determinado grupo de pessoas. Um exemplo de uma interrogação útil será qual o tempo de

permanência de todas as pessoas e número de pessoas, para cada espaço, na semana anterior ao início do ano letivo.

Evoluir a solução por forma a mostrar dados projetados na planta do edifício permitiria uma interpretação mais intuitiva dos mesmos, substituindo uma legenda textual para a estimativa de localização por uma representação no mesmo domínio da estimativa – o espaço físico. Esta opção será interessante tanto para a localização de um indivíduo, como para relatórios estatísticos que considerem grupos de pessoas.

6.2.3 Solução Global

Partindo de uma solução de processamento em tempo real, que se assemelha a uma solução de vigilância, serão possíveis outros casos de utilização, ainda no âmbito da publicidade:

1. Ligar / desligar dinamicamente painéis e ecrãs publicitários, consoante a presença de pessoas no espaço circundante;
2. Alternar, em tempo real, a publicidade exibida, adaptando-a ao grupo alvo presente em cada momento, por exemplo, da seguinte forma: Sabendo que perto de um ecrã publicitário se encontra(m) dispositivo(s) com endereço MAC do fabricante Apple Inc., poderá ser relevante exibir publicidade relevante a esse mesmo grupo alvo (e.g. promoção de capas para iPhone na loja mais perto).

6.2.4 Considerações Gerais

A proposta de solução, apesar de tecnicamente viável, apresenta ainda relevantes desafios a superar. Destacam-se, entre estes, os que contribuem para uma melhor qualidade da estimativa de localização, em particular o problema da quantidade de interfaces USB conectáveis ao Raspberry Pi.

A solução foi desenvolvida mantendo em aberto a hipótese de ser migrada para outras arquiteturas, nomeadamente a predecessora (ARMv6) e a sucessora. A solução não tem limitações de obtenção de peças especialmente relevantes, salvo o requisito de suporte de modo monitor por parte do conjunto placa Wi-Fi e respetivo driver Linux. Desta forma, espera-se que seja possível manter a solução por um período alargado de tempo, tal

como os elementos que a compõem. São relevantes as seguintes considerações sobre os mesmos, numa perspetiva de manutenção evolutiva e corretiva:

- Drivers: O kernel 4.1, usado nesta solução, foi lançado com suporte oficial até 2018. Isto garante que os drivers usados funcionarão até lá; A versão 4.2 do kernel suporta os drivers wireless da versão 4.1, aumentando a confiança sobre a usabilidade de drivers para além de 2018 (fim do suporte do kernel 4.2) numa solução atualizada.
- Hardware: Os dois chipsets utilizados na solução juntam-se a outros para compor a oferta de mercado no que diz respeito às placas Wi-Fi com suporte a modo monitor em Linux. Espera-se que a novos chipsets introduzidos no mercado correspondam ofertas de drivers com suporte ao modo monitor (disponibilizados pelo fabricante ou pela comunidade Linux), à semelhança do que tem vindo a acontecer. Destaca-se nesta oferta o hardware produzido pela Mediatek e sua subsidiária Ralink, pelo suporte a modo monitor e pela disponibilização do código fonte dos drivers Linux sob licença GPL: esta combinação permite o desenvolvimento pela comunidade do suporte a modo monitor em Linux, caso este não exista.
- Scripts captura: O desenvolvimento do script que inicia a captura no arranque do sistema foi desenvolvido para a plataforma System V, que foi substituída pela System D nas mais recentes distribuições Linux. Para evoluir a solução para uma distribuição mais recente, perspetiva-se a necessidade de desenvolver scripts de arranque para a plataforma System D. Esta necessidade deve-se a uma retro-compatibilidade entre as plataformas seguir uma abordagem melhor esforço. A distribuição usada no protótipo está, no que toca à plataforma de scripts de arranque, desatualizada no contexto das distribuições Linux modernas. A distribuição Raspbian utilizada baseia-se na última versão Debian que usa System V (versão 7), estando já disponível uma nova versão, baseada em Debian 8, que utiliza System D.
- Pacotes / suítes usados(as):
 - O pacote Debian tshark (bem como o wireshark) disponibilizado pela Debian, na sua versão compatível com o Raspbian utilizado num RaspberryPi 2 foi lançado em Abril de 2015. O pacote disponível para a versão mais recente do Raspbian

em Raspberry Pi 2 é ainda mais recente (Setembro de 2015). Não se antevêem, portanto, constrangimentos de suporte atual nem futuro.

- O pacote kismet não tem novas versões desde 2013, continuando a ser disponibilizado para a versão Raspbian utilizada e sua sucessora.

Manter a solução, do ponto de vista corretivo, é, pelo anteriormente exposto, um trabalho exequível. Já a migração para um sistema de outra versão ARM, mantendo uma distribuição Linux baseada em Debian, pode implicar a utilização de diferentes versões dos pacotes Debian (por exemplo, compiladas segundo a convenção *armel*). Utilizar uma distribuição não baseada em Debian implica substituir os pacotes utilizados por pacotes compatíveis com a distribuição escolhida, ou, em alternativa, pela compilação das fontes.

6.3 Privacidade

Sendo a privacidade um tema atualmente relevante, torna-se um aspeto a considerar no desenvolvimento de soluções que recolhem dados e/ou meta-dados com origem em dispositivos pessoais, como é o caso da solução apresentada.

A solução e os métodos aqui descritos podem ser utilizados para fins de vigilância e supervisão, pelo que foi objetivo da implementação minimizar a informação sensível recolhida. Desta forma, foram aplicados filtros de captura diretamente no processo de recolha, evitando a recolha de algumas tramas com dados sensíveis: Não são recolhidas tramas de dados, onde se incluem tráfego potencialmente desprotegido (em claro) efetuado sobre redes Wi-Fi abertas (sem cifra). Esta característica é conseguida utilizando filtros de captura (opção -f) na aplicação tshark, técnica que apresenta uma performance superior às alternativas. Também na ingestão são aplicados filtros que reduzem a informação utilizada para efeitos de geolocalização, sendo apenas utilizado o endereço MAC como identificador do dispositivo.

A escolha das tramas probe-request minimiza a observação de dados que possam ser usados para inferir a identidade do utilizador do dispositivo localizado (MAC e SSID). Para esta realidade contribui a forma como o tshark implementa o filtro de captura: ao fazê-lo em modo kernel, não existem outros tipos de tramas expostos em memória do espaço de endereçamento do utilizador, contribuindo assim para respeitar o princípio da privacidade. Espera-se que a escolha do tipo de tramas a observar, bem como os dados processados

pelo sistema sejam ponderados pelo respeito do direito à privacidade de utilizadores (utilizadores esses que potencialmente não estarão conscientes da operação do sistema); Desta forma, deverá ser feita uma escolha ética.

Sobre a utilização do endereço MAC como identificador do dispositivo, é pertinente assinalar, por um lado, que este pode, em certos casos, identificar o dispositivo, mas por outro, a sua utilização não é estritamente necessária no âmbito do problema. Numa implementação alternativa, poderá alterar-se cada endereço MAC sem perder a unicidade (para endereços MAC diferentes nas tramas capturadas, persistir endereços MAC diferentes), por exemplo, através de uma função de dispersão (*hash*); Será, no entanto, importante manter a informação do fabricante (indicado pelos primeiros 3 octetos) do endereço MAC.

Para relativizar a importância da recolha do endereço MAC do é importante mencionar que existem aparelhos que mascaram o seu endereço MAC, em linha com preocupações de privacidade. Existem atualmente soluções de proteção que desligam o Wi-Fi em certos cenários [SWM01] e que alteram esporadicamente o endereço MAC, como alguns aparelhos Apple com sistema operativo iOS 8 ou mais recente [iOSsec15].

A primeira das técnicas terá maior impacto na solução, dado que consegue em certas circunstâncias impedir a recolha de tramas, impedindo a localização do dispositivo. A técnica de alteração do endereço MAC pode falsear dados estatísticos, pois altera a relação entre endereço MAC e dispositivo, originalmente assumida como sendo de 1 para 1.



Bibliografia

Bibliografia

FB01: Ferenc Brachmann, A comparative analysis of standardized technologies for providing indoor geolocation functionality, 2012

TS01: Tim Smith, Collecting a Wireless sniffer trace using the Cisco Lightweight AP in Sniffer mode, 2014 - <https://supportforums.cisco.com/document/75236/collecting-wireless-sniffer-trace-using-cisco-lightweight-ap-sniffer-mode>

CS01: Cisco Systems, Inc., Cisco IOS Command References, 2014

CS02: Cisco Systems, Inc., Cisco Wireless LAN Controller Command Reference, Release 8.0, 2014

Eka01: Ekahau, Inc, Palmetto Health Uses Ekahau RTLS and CiscoMSE v7.5 to Improve Operational Efficiency, 2015 - http://www.ekahau.com/userData/ekahau/documents/case-studies/Ekahau_RTLS_Palmetto_CS.pdf

IEEE 802.11: IEEE, Telecommunications and information exchange between systems: Local and metropolitan area networks—Specific requirements, 2012 - <http://standards.ieee.org/getieee802/download/802.11-2012.pdf>

SGT01: Ana Roxin, Jaafar Gaber, Maxime Wack, Ahmed Nait Sidi Moh, Survey of Wireless Geolocation Techniques, 2007

LTA08: Cisco Systems, Inc, Wi-Fi Location-Based Services 4.1 Design Guide, 2008

PLM08: Atreyi Bose; Chuan Heng Foh , A Practical Path Loss Model For Indoor WiFi Positioning Enhancement , 2008

Calc15: Mozilla, Ichnaea Documentation, 2015 - <https://mozilla.github.io/ichnaea/calculation.html>

CLT06: Cisco Systems Inc., Wi-Fi Based Real-Time Location Tracking: Solutions and Technology , 2006 - http://www.intermec.com/public-files/white-papers/en/cisco_locationtracking_wp_web.pdf

SGT12: Ana Roxin, Jaafar Gaber, Maxime Wack, Ahmed Nait Sidi Moh, Survey of Wireless Geolocation Techniques, 2007

LW01: Arend van Spriel, Johannes Berg, Ruoyao Xi, N. Izumi, Eitan Bar, Bartosz Markowski, Existing Linux Wireless drivers, 2015 - <https://wireless.wiki.kernel.org/en/users/Drivers>

US01: USB Implementers Forum, Inc., USB Frequently Asked Questions, - <http://www.usb.org/developers/usbfaq#cab1>

F2FS15: Changman Lee; Dongho Sim; Joo-Young Hwang; Sangyeun Cho, F2FS: A New File System for Flash Storage, 2015

ODATA14: Michael Pizzo; Ralf Handl; Martin Zurmuehl, OData Version 4.0 Part 2: URL Conventions Plus Errata 02, 2014

SSE15: Ian Hickson, Server-Sent Events, 2015 - <https://www.w3.org/TR/2015/REC-eventsourcing-20150203/>

SWM01: Kismet Wireless, Smarter Wi-Fi Manager, 2015 - <http://www.kismetwireless.net/android-swm/>

iOSsec15: Apple Inc., iOS Security, 2015 - http://www.apple.com/business/docs/iOS_Security_Guide.pdf

8.1 Análise de custo de placas Wi-Fi

Marca e Modelo	Preço	Chipset
TP-LINK WN725	9,90€	RTL8192cu
TP-LINK WN821N	14,90€	RTL8192cu
TP-LINK WN823N	16,90€	RTL8192cu
TP-LINK WN822N	21,90€	RTL8192cu ¹
NETGEAR N300	21,90€	RTL8192cu

Tabela 5: Preço de placas Wi-Fi (Julho 2015, MediaMarkt)

¹ A partir da versão 3 do hardware.

8.2 Lista de Software Utilizado

Software	Versão
Distribuição Linux Raspbian	Maio 2015
Pacote debian tshark	1.8.2
Pacote debian kismet	2013.03.R1
Pacote debian gawk	1:4.1.3
Pacote debian sed	4.2.1
Drivers Linux MediaTek MT7601	3.0.0.4
PostgreSQL	9.4 (Windows, 64-bit)
Visual Studio	2015 Ultimate
Internet Information Services	10.0 Express
.NET Framework	4.6.1
Nuget Microsoft.AspNet.WebApi	5.2.3
Nuget NetTopologySuite	1.14
Nuget ProjNET4GeoAPI	1.3.0.3
Nuget Rx-Linq	2.2.4

8.3 Recolha

8.3.1 Script de preparação do ambiente

```
#!/bin/bash

sudo apt-get install -y tshark
sudo dpkg-reconfigure wireshark-common
# (choose yes)

# enables capturing with current user
sudo adduser `id -n -u` wireshark

# limits ifplugd scope of action to eth0 interface
sudo sed -i 's/^.*HOTPLUG_INTERFACES=.*$/HOTPLUG_INTERFACES=""/'
/etc/default/ifplugd
# sudo sed -i 's/^.*HOTPLUG_INTERFACES=.*$/HOTPLUG_INTERFACES="all"/' /etc/default/ifplugd
sudo sed -i 's/^.*INTERFACES=.*$/INTERFACES="eth0"/' /etc/default/ifplugd
# sudo sed -i 's/^.*INTERFACES=.*$/INTERFACES="auto"/' /etc/default/ifplugd

# may be reverted with
# sudo dpkg-reconfigure ifplugd
```

8.3.2 Script para colocação das interfaces em modo monitor

```
#!/bin/bash
# requires ifconfig and iwconfig
# tries to stop ifplugd for the interface

set -e
function usage
{
    echo "usage: -m <mode> -i <interface> [-i <interface>] | -h"
    echo ""
    echo "  -i interface:      wireless interface"
    echo "  -m <mode>:        optional, defaults to monitor; see iwconfig for more info"
    echo "  -h:               prints this help"
}

mode="monitor"

while [ "$1" != "" ]; do
    case $1 in
        -m | --mode )
```

```

                                shift
                                mode=($1)
                                ;;
                                -i | --interface )      shift
                                interfaces+=($1)
                                ;;
                                -h | --help )           usage
                                exit
                                ;;
                                * )                     usage
                                exit 1
        esac
        shift
done

if [ ${#interfaces[@]} -eq 0 ]; then
    usage
    exit 1
fi

for i in "${interfaces[@]}"
do
    sudo ifplugd -c -i $i &> /dev/null && \
    printf "stopping ifplugd for $i\n" && \
    sudo ifplugd -i $i -k -W

    ifconfig $i down; \
    iwconfig $i mode $mode;

    ifconfig $i up;
done

```

8.3.3 Script de captura

```

#!/bin/bash
set -m

function usage()
{
    echo "usage: $0 [--auto | -a] [--init | -n] <APs output file> [-o |
--outputFile] <clients output file> [-i | --interfaces] <interface1
interface2 ... interfaceN>\n"
    echo "for auto option, AP output file must be supplied. if file exists,
APs will not be logged"
    echo "\-o and \-i are mandatory"
}

while [[ $# > 1 ]]
do
    key="$1"

```

```

case $key in
    -a|--auto)
        AUTO="yes"
        ;;
    -n|--init)
        APOUTFILE="$2"
        shift
        ;;
    -o|--outputFile)
        OUTFILE="$2"
        shift # past argument
        ;;
    -i|--interfaces)
        INTERFACES="$2"
        shift # past argument
        ;;
    *)
        # unknown option
        usage && exit 1
        ;;
esac
shift # past argument or value
done

# echo "$OUTFILE"
# echo "$APOUTFILE"
# echo "$INTERFACES"
# exit

if [ "$OUTFILE" == "" -o "$INTERFACES" == "" ] ;then
    usage && exit 1
fi

if [ "$AUTO" != "" -a "$APOUTFILE" ] ;then
    usage && exit 1
fi

# builds -i arg0 -i arg1 -i arg2 ...
TINTERFACES=`echo $INTERFACES | sed -r 's/^| / -i /g'`
MINTERFACES=`echo $INTERFACES | sed -r 's/^| / -i /g'`
KINTERFACES=`echo $INTERFACES | sed -r 's/^| / -c /g'`

function process_EXIT()
{
    jobs -p | xargs kill
    printf "\n[STOP]\n"
}

trap "process_EXIT" EXIT

function enable_monitor()

```



```

{
    printf "\n[START] setting monitor mode"
    if (wireless_mode.sh $MINTERFACES) ;
    then
        printf "\n[OK] set monitor mode!\n"
        return 0
    else
        printf "\n[NOT OK] set monitor mode!\n"
        return 1
    fi
}

# see https://www.wireshark.org/docs/dfref/
# see https://www.wireshark.org/docs/man-pages/pcap-filter.html
#stdbuf -oL \
# -f 'not (subtype beacon or subtype probe-resp or subtype assoc-resp or subtype
reassoc-resp)'

function background_clients_log()
{
printf "\n[START] client logging\n"
tshark -q \
-f 'subtype probe-req' \
-F pcapng -b filesize:65536 -w $OUTFILE \
-I $INTERFACES &
}

function background_APs_log()
{
printf "\n[START] AP logging\n"
tshark -q \
-f 'subtype beacon' \
-F pcapng -a duration:360 -w $APOUTFILE \
-I $INTERFACES &
}

enable_monitor || exit 2

printf "\n[START] channel-hopping\n"
# uses kismet-server for channel-hopping
# n - no logging; s - silent; x - force channel-hopping
kismet_server -n -s -x $KINTERFACES &

sleep 3

# runs tshark for logging
if [ "$APOUTFILE" != "" ] ; then
    background_APs_log
fi
background_clients_log

```

```
while pgrep -P "$BASHPID" > /dev/null; do
    wait
done
```

8.3.4 Exemplo de ficheiro de configuração

```
#!/bin/bash

CLIENTOUTFILE="/etc/wlanradar/client.cap"
APOUTFILE="/etc/wlanradar/ap.cap"
INTERFACES="wlan0 wlan1 wlan2"
```

8.3.5 Deamon System V (init script)

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          wlanradar
# Required-Start:    $all
# Required-Stop:     $local_fs $network $named $time $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Description:       Logs nearby wlan clients
### END INIT INFO

NAME=wlanradar
PATH=/sbin:/usr/sbin:/bin:/usr/bin
SCRIPT=/usr/bin/$NAME.sh
RUNAS=root

PIDFILE=/var/run/$NAME.pid
LOGFILE=/var/log/$NAME.log

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh

start()
{
    if [ -f "$PIDFILE" ] && kill -0 $(cat "$PIDFILE"); then
        echo 'Service already running' >&2
        return 1
    fi
    if [ ! -x $script ] ;then
```

```

    echo 'unable to run' $script>&2
    return 2
fi
echo 'Starting service' >&2
local CMD="$SCRIPT --auto -n \"\$APFILE\" -o \"\$OUTFILE\"
-i \"\$INTERFACES\"&>> \"\$LOGFILE\" & echo \$!"
su -c "$CMD" $RUNAS > "$PIDFILE"
echo 'Service started' >&2
}

stop()
{
    if [ ! -f "$PIDFILE" ] || ! kill -0 $(cat "$PIDFILE"); then
        echo 'Service not running' >&2
        return 1
    fi
    echo 'Stopping service' >&2
    kill -15 $(cat "$PIDFILE") && rm -f "$PIDFILE"
    echo 'Service stopped' >&2
}

uninstall()
{
    echo -n "Are you really sure you want to uninstall this service? That cannot be
undone. [yes|No] "
    local SURE
    read SURE
    if [ "$SURE" = "yes" ]; then
        stop
        rm -f "$PIDFILE"
        echo "Notice: log file will not be removed: '$LOGFILE'" >&2
        update-rc.d -f wlanradar remove
        rm -fv "$0"
    fi
}

restart()
{
    stop
    start
}

case "$1" in
    start|stop|uninstall|restart)
        $1
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|uninstall}"
esac

```

8.4 Processamento

8.4.1 Script Windows CLI para ingestão de tramas

```
@echo off
("tshark.exe" -r "%1" -T fields -E occurrence=f -e radiotap.dbm_antsignal -e
frame.time_epoch -e wlan.sa -e frame.interface_id) ^
| awk "{ print $0 \"\\traspberrypi2\" }" | "psql.exe" -U postgres -w -c
"copy \\Input\\ (\\dBm\\, \\Timestamp\\, \\MAC\\, \\Interface\\, \\Source\\)
FROM STDIN;" postgres
```

8.4.2 Exemplo de comando para recolha da ground truth

```
tshark -r "\\LACIE-CLOUDBOX\\afaganga\\capturas\\client_00001_20150910194259.cap"
-T fields -E occurrence=f -e radiotap.dbm_antsignal -e frame.time_epoch -e
wlan.sa -e frame.interface_id -Y "(wlan.sa == 80:6a:b0:10:91:71 or wlan.sa ==
c0:bd:d1:1c:3a:11) and frame.time >= \\\"Sep 10, 2015 19:46:47\\\" and frame.time <=
\\\"Sep 10, 2015 19:51:00\\\"" | awk "{ print $0 \"\\traspberrypi2\\t1\" }" |
"psql.exe" -U postgres -w -c "copy \\\"Training\\\"
(\\dBm\\, \\Timestamp\\, \\MAC\\, \\Interface\\, \\Source\\, \\Place\\) FROM ST-
DIN;" postgres
```

8.4.3 Exemplo de script para construção da ground truth

```
--delete from "Training";
with ref as
(
select timestamp '1999-10-27 14:04' as "Begin", timestamp '1999-10-27 14:04' as
"End", 'dummy' as "Place"
union all
select timestamp '2015-10-26 19:15', timestamp '2015-10-26 19:26', 'escritorio'
union all
select timestamp '2015-10-26 19:27', timestamp '2015-10-26 19:38', 'cozinha'
union all
select timestamp '2015-10-26 19:40', timestamp '2015-10-26 19:50', 'quarto'
union all
select timestamp '2015-10-26 19:51', timestamp '2015-10-26 20:01', 'wc'
union all
select timestamp '2015-10-26 20:03', timestamp '2015-10-26 20:15', 'sala'
)
insert into "Training"
select "Input".*, "Place"
from ref
inner join "Input"
```

```

on to_timestamp("Timestamp") between "Begin" and "End"
--and "MAC" = '00:73:8d:9e:55:6a'
and "MAC" = 'c0:bd:d1:1c:3a:11'
;

```

8.4.4 Script SQL para criação do modelo

```

CREATE TABLE "Input"
(
    "MAC" character(17),
    "dBm" integer,
    "Timestamp" numeric(20,9) NOT NULL,
    "Interface" character varying(20) NOT NULL,
    "Source" character varying(20) NOT NULL,
    CONSTRAINT "Input_pkey" PRIMARY KEY ("Timestamp", "Interface", "Source")
)

CREATE TABLE "Training"
(
    "MAC" character(17),
    "dBm" integer,
    "Timestamp" numeric(20,9) NOT NULL,
    "Interface" character varying(20) NOT NULL,
    "Source" character varying(20) NOT NULL,
    "Place" character varying(20) NOT NULL,
    CONSTRAINT "Training_pkey" PRIMARY KEY ("Source", "Interface", "Timestamp")
)

CREATE OR REPLACE VIEW "FilteredTraining" AS
SELECT "Training"."MAC",
       "Training"."dBm",
       "Training"."Timestamp",
       "Training"."Interface",
       "Training"."Source",
       CASE
           WHEN "Training"."Place"::text = 'sala 1'::text THEN 'sala'::character varying
           WHEN "Training"."Place"::text = 'sala 2'::text THEN 'sala'::character varying
           ELSE "Training"."Place"
       END AS "Place"
FROM "Training"
WHERE "Training"."Place"::text <> ALL (ARRAY['2o'::character varying::text,
'cozinha'::character varying::text, 'escritorio'::character varying::text,
'wc'::character varying::text, 'quarto'::text]);

CREATE OR REPLACE VIEW "GroupedTraining" AS
WITH context AS (
    SELECT original."Timestamp" AS "Mark",
           other."MAC",
           other."dBm",

```

```

        other."Timestamp",
        other."Interface",
        other."Source",
        other."Place",
        abs(other."Timestamp" - original."Timestamp") AS delta
    FROM "FilteredTraining" original
        JOIN "FilteredTraining" other ON other."MAC" = original."MAC" AND
other."Place"::text = original."Place"::text
    )
SELECT "timeDeltas"."Mark",
    "timeDeltas"."MAC",
    "timeDeltas"."dBm",
    "timeDeltas"."Timestamp",
    "timeDeltas"."Interface",
    "timeDeltas"."Source",
    "timeDeltas"."Place",
    "timeDeltas".delta
FROM ( SELECT context."Mark",
    context."MAC",
    context."dBm",
    context."Timestamp",
    context."Interface",
    context."Source",
    context."Place",
    context.delta
    FROM context) "timeDeltas"
LEFT JOIN ( SELECT context."Mark",
    context."MAC",
    context."dBm",
    context."Timestamp",
    context."Interface",
    context."Source",
    context."Place",
    context.delta
    FROM context) correlation ON "timeDeltas"."Mark" = correlation."Mark"
AND "timeDeltas"."MAC" = correlation."MAC" AND "timeDeltas"."Source"::text =
correlation."Source"::text AND "timeDeltas"."Interface"::text = correlation."In-
terface"::text AND "timeDeltas"."Place"::text = correlation."Place"::text AND
"timeDeltas".delta > correlation.delta
WHERE correlation.delta IS NULL AND "timeDeltas".delta < 6::numeric;

CREATE OR REPLACE FUNCTION "Filter"(
    IN mac character varying,
    IN "Time" numeric,
    IN filterseconds numeric DEFAULT 3.6)
RETURNS TABLE("dBm" numeric, "Interface" character varying, "Source" character
varying) AS
$BODY$
SELECT AVG("dBm") as "dBm", "Interface", "Source"
FROM "Input"
WHERE "MAC" = MAC
and "Timestamp" between "Time" - FilterSeconds/2 and "Time" + FilterSeconds/2
GROUP BY "Interface", "Source"

```

```

$BODY$
LANGUAGE sql VOLATILE STRICT;

CREATE OR REPLACE FUNCTION "Location"(
    mac character varying,
    "Time" numeric,
    filterseconds numeric DEFAULT 3.6)
RETURNS character varying AS
$BODY$

select "Place"
from(
    select "Place", count(*)
    from(
        select "Place", 1/ SUM(("Sample"."dBm" - "Reference"."dBm")^2) as
"distance"
        from
        (
            select * from "Filter"(mac, "Time", filterseconds)
        ) as "Sample"
        INNER JOIN
        (
            SELECT "Mark", "dBm", "Source", "Interface", "Place"
            FROM "GroupedTraining"
        ) as "Reference"
        ON "Reference"."Source" = "Sample"."Source"
        and "Reference"."Interface" = "Sample"."Interface"
        group by "Reference"."Mark", "Place"
        order by "distance" asc
        limit 13
    ) as "votes"
    group by "Place"
    order by "count" desc
    limit 1
) as "winner"

$BODY$
LANGUAGE sql VOLATILE STRICT;

```

8.4.5 Código do serviço web

Dada a extensão deste componente, o mesmo não é aqui reproduzido; Este anexo encontra-se apenas entre os que acompanham (em formato digital) este relatório.