

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Informatica per il Management

**Progettazione di un Servizio Web  
per la Geolocalizzazione di:  
Access Point in contesti di Alta Mobilità**

Tesi di Laurea in Laboratorio di Programmazione di Internet

**Relatore:**

**Dott.**

**Stefano Ferretti**

**Presentata da:**

**Gianluca Orlando**

**Sessione prima**

**Anno Accademico 2011-2012**

'A Paolo e Maria Teresa, i miei genitori, che mi hanno spronato con  
passione e dedizione fino ad oggi, e che sempre lo faranno'

'A Gaia, che é stata la mia forza e il mio sostegno da quando mi é accanto'

'Un grazie speciale a Marco, che mi ha aiutato nel lato "tecnico", e a tutti  
gli amici e parenti che credono in me'

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Il Sistema ABPS</b>	<b>9</b>
2.1	Struttura di ABPS . . . . .	10
<b>3</b>	<b>Il Localizzatore di Hotspot</b>	<b>13</b>
3.1	Lo Stato Dell'Arte . . . . .	14
3.2	La Mappatura delle Reti Wireless . . . . .	14
3.2.1	Wigle . . . . .	14
3.2.2	Wefi . . . . .	17
3.2.3	Google Maps (Street View) . . . . .	18
3.2.4	Altri Servizi . . . . .	19
3.2.5	L'Analisi Dei Dati . . . . .	20
3.3	Il Mio Sistema . . . . .	21
3.3.1	Perché Wigle . . . . .	21
3.3.2	L'interazione con Wigle.net . . . . .	21
<b>4</b>	<b>Implementazione</b>	<b>23</b>
4.1	Python . . . . .	23
4.2	Librerie . . . . .	24

<b>5</b>	<b>Sperimentazione</b>	<b>27</b>
5.1	Contesto Urbano . . . . .	28
5.2	Contesto a Bassa Densità di Hotspot . . . . .	42
<b>6</b>	<b>Conclusioni e Sviluppi Futuri</b>	<b>49</b>

# Capitolo 1

## Introduzione

In un mondo dove l'efficienza, la velocità e l'economicità dei sistemi informatici diventa sempre più un dogma, la necessità di disporre di una connessione ad Internet sempre disponibile è oramai un bisogno primario.

Tuttavia, esistono ancora problematiche o sfide che devono essere affrontate, specialmente per quanto riguarda il lato "mobile" della connettività. Su device mobili è possibile disporre di connessioni di dati 3G (ed in futuro arriverà in Italia anche il 4G), HSDPA, EDGE piuttosto che wireless, ma il passaggio dalle une all'altra si dimostra spesso lento e poco performante.

Ciò diventa un ostacolo quando si utilizzano applicazioni che richiedono l'utilizzo di una connessione internet costante ma non si ha la possibilità di rimanere allacciati ad una rete wireless (magari perché si è in movimento, come quando si utilizza un'applicazione VoiP).

Inoltre, per utilizzare una rete a pacchetto o una rete dati, è necessario disporre di un sistema dotato di scheda SIM.

Trattando proprio le applicazioni VoiP, se volessimo basarci solo su reti wireless, ci ritroveremmo a perdere in continuazione la connessione con relativa perdita della comunicazione. Avere la possibilità di passare rapidamente da

una rete all'altra senza compromettere la chiamata (o qualsiasi altra operazione), sarebbe un notevole successo sia in termini di efficienza che in termini di costi.

Non solo i dispositivi portatili e mobili sarebbero interessati dalla possibilità di utilizzare reti wireless diverse in contesti di alta mobilità, ma anche gli stessi dispositivi di telefonia mobile, i quali per svariate ragioni come tariffe poco convenienti o limitazione dei minuti a disposizione, potrebbero sfruttare solamente applicazioni VoIP per le comunicazioni.

Il sistema ABPS (Always Best Packet Switching) sviluppato dal Professor Ghini dell'università di Bologna con i suoi studenti, ha permesso di ovviare al problema di passaggio fra reti eterogenee (ad esempio, da WiFi a rete di dati) in modo automatico senza far perdere la comunicazione. Questo procedimento è detto "seamless communication", ossia comunicazione priva di "punti di giuntura", in quanto l'utente non si accorge del passaggio da un tipo di rete ad un'altra. In questa tesi ho voluto analizzare un possibile ampliamento di tale sistema, sviluppando un programma capace di segnalare le reti wireless vicine, in modo tale da permettere ad un dispositivo mobile di connettersi automaticamente da una rete wireless all'altra senza dover passare attraverso la rete di dati fornita dalle compagnie telefoniche.

Nel capitolo 2 descriverò brevemente il sistema ABPS, spiegando in modo rapido e conciso il suo funzionamento e le sue caratteristiche principali.

Nel capitolo 3 esplicherò il problema analizzato, lo stato dell'arte al momento della realizzazione del mio sistema nonché il sistema stesso.

Nel capitolo 4 procederò con la spiegazione delle scelte fatte per l'implementazione del sistema, nonché una breve descrizione dell'implementazione stessa.

Nel capitolo 5 forniró un semplice esempio di ciò che il mio sistema é in grado di effettuare.

Infine, nel capitolo 6, tratteró le conclusioni raggiunte e i possibili sviluppi futuri di tale sistema.



## Capitolo 2

# Il Sistema ABPS

Il sistema ABPS é un'architettura per la fornitura di seamless mobile services. Con questo sistema, gli utenti possono utilizzare tutti i tipi di connessioni wireless di cui un terminale mobile é dotato.

Come un'applicazione multimediale sfrutta il protocollo UDP, ABPS consente la trasmissione di ciascun pacchetto UDP attraverso la miglior interfaccia di rete (NIC) ( in termini di reattività e disponibilità di banda ad esempio) tra quelli a disposizione. Esso consente l'utilizzo di politiche di bilanciamento del carico e di recupero dati. ABPS é un'architettura che si rivolge primariamente alla comunicazione VoIP di un dispositivo mobile, sia esso un computer portatile piuttosto che uno smartphone, dotato di diverse interfacce di rete. ABPS supporta le applicazioni multimediali basate sui protocolli SIP e RTC/RTCP, evitando i ritardi generati dai messaggi di handshake dei due metodi.

ABPS-SIP/RTP é progettato per soddisfare questi 3 requisiti:

- per garantire la comunicazione e la continuità del servizio;
- per abilitare l'uso simultaneo di tutte le schede NIC;

- per ridurre al minimo il ritardo del segnale cercando il tunnel migliore.

La possibilità di avere a disposizione diverse interfacce di rete sullo stesso dispositivo, permette di dare continuità alla connettività nel caso venga a mancare una rete WiFi, dato che il terminale può appoggiarsi alla rete GPRS/UMTS in mancanza o "debolezza" di un access point, come può succedere in contesti urbani.

Un'altra possibilità è quella di poter scegliere il NIC preferito per l'accesso alla rete. Questa funzione è detta Mobility Management delle reti wireless del modello "Always Best Connected". Una "classifica" delle interfacce preferite, obbliga il dispositivo ad utilizzare determinate interfacce di rete in base alla preferenza dell'utente, passando alle successive solo quando il segnale della precedente diventa troppo debole. ABPS è dunque un'applicazione installata su un terminale mobile equipaggiato con diversi NIC, in grado di rilevare i diversi access point presenti nelle vicinanze, e capace quindi di collegarsi al migliore scegliendo in base ad un'analisi del traffico e della potenza del segnale. Il sistema entra in funzione da un client quando esso decide di connettersi ad un access point. Da quel momento in poi, il client ABPS controlla la presenza di tutti gli access point presenti nell'area per un certo intervallo di tempo, al termine del quale vengono valutati potenza del segnale, quantità di traffico e interferenze di essi, e registra una classifica dal migliore al peggiore secondo tali parametri. Il client cercherà dunque di associarsi a tali access point in base alla loro posizione nella classifica.

## 2.1 Struttura di ABPS

Il sistema è formato da 3 processi chiave:

- analizzatore di rete;

- gestore di associazioni con gli access point;
- coordinatore.

L'analizzatore di rete rimane in ascolto di tutti i canali wireless, "sniffando" i pacchetti inviati dagli hotspot. Analizzandoli, crea una lista in ordine di qualità degli access point tramite determinate caratteristiche, che sono calcolate stimando per ciascuno di essi, il volume di traffico che stanno sostenendo (in percentuale), le interferenze con i canali vicini, le tipologie dei pacchetti sniffati e la potenza del segnale.

Il gestore di associazioni deve appunto associare un terminale wireless ad uno degli hotspot classificati dall'analizzatore di rete. Prima che avvenga l'associazione, esso verifica che il coordinatore non abbia escluso quel particolare access point, inserendolo nella lista delle esclusioni. Quindi, se l'hotspot è stato escluso o non è raggiungibile per qualsiasi motivo, il gestore passa al successivo.

Finché non si stabilisce un collegamento con un access point, il gestore continua a ripetere queste operazioni. Se al termine della lista non è avvenuta alcuna associazione, allora il gestore rilancia l'analizzatore di rete per ottenere una nuova lista dei migliori access point e ripetere l'intero procedimento.

Il coordinatore si comporta da mediatore fra il gestore e l'analizzatore, insieme al WPA Supplicant (che contiene le varie operazioni a livello Data Link). Avviato il client, il coordinatore avvia gli altri due processi e resta in attesa di un'associazione. Non appena l'interfaccia di rete è associata, ottiene l'indirizzo IP per completare la fase di "collegamento" (Data Link). Se il client si disconnette, attende una nuova associazione e ri-ottiene l'indirizzo IP.

Un'altra funzionalità importante e sicuramente utile in termini di perfor-

mance, è la lista delle esclusioni. In essa, il coordinatore salva gli estremi di tutti gli access point a cui il client (l'analizzatore di rete per la precisione) non è riuscito ad associarsi per mancanza di indirizzo IP. Ciò permette di evitare che il sistema tenti più volte di collegarsi allo stesso hotspot che non dispone di una connessione Internet.

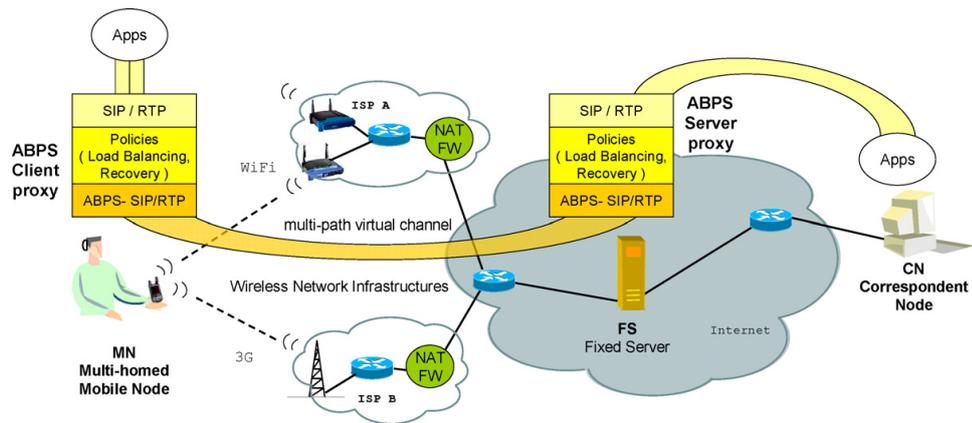


Figura 2.1: Una rappresentazione del sistema ABPS

## Capitolo 3

# Il Localizzatore di Hotspot

Il mio progetto consisteva nell'integrare ABPS con un modulo software capace di rilevare in anticipo le reti wireless, in modo tale da permettere al sistema (e quindi al terminale) di saltare da un wireless all'altro senza passare dalla rete dati delle compagnie telefoniche. Lo scopo é duplice: in primis, evitare i costi e il consumo di banda mensile delle compagnie telefoniche; in secundis, di affidarsi a reti wireless anche quando non é disponibile la copertura GPRS/UMTS (ove possibile).

In primo luogo é stato necessario trovare un modo per avere a disposizione una mappatura delle reti wireless. Ho consultato diversi progetti e portali web fra cui una sezione di Google Maps, Wigle.net, WeFi e Wired Italia (Progetto "Sveglia Italia"). Dunque é stato necessario trovare un modo per interagire con i portali, e infine, rendere i dati cosí ottenuti disponibili in un formato semplice, portabile e durevole, in modo tale che sia facilmente integrabile nel sistema ABPS.

### 3.1 Lo Stato Dell'Arte

Al momento della realizzazione della tesi, non ho trovato alcun progetto che avesse tutti i requisiti necessari per essere implementato direttamente nel sistema ABPS. È stato dunque necessario effettuare un "collage" di diverse tecnologie.

### 3.2 La Mappatura delle Reti Wireless

Considerando i servizi capaci di fornire un database aggiornato di tutte le possibile reti wireless, solo uno si è dimostrato utile poiché opensource sia nella consultazione, sia nell'acquisizione di dati, sia nella fornitura del codice: il progetto Wigle.net.

#### 3.2.1 Wigle

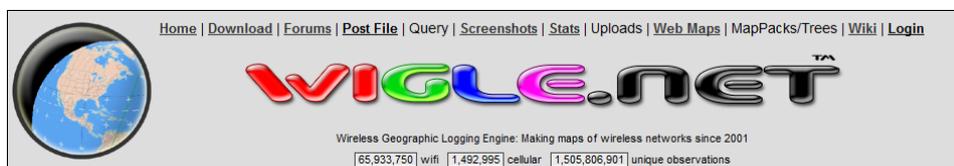


Figura 3.1: Il banner di Wigle

Wigle è un progetto nato nel 2001 e da allora ha collezionato più di 64 milioni di reti wifi uniche (ossia uniche sia come Net-ID che come SSID).

È un'applicazione web submission-based che tramite un database centrale, mappa tutte le reti wireless rilevate da clienti wigle.

Wigle dispone di diversi tipi di client:

- DiGLE (nativo per Windows);
- JiGLE (Java cross-platform client);
- TinGLE (nativo per MacOSX);

- WiGLEWiFi (applicazione Android).

Tutti questi client possono effettuare query, update e mappare il database attraverso le API fornite dagli autori. Il database é aperto previa registrazione sul loro sito.

Attualmente, Wigle accetta file in formato NetStumbler, file esportati MiniStumbler, test in formato DStumbler, Kismet, XML, CSV, formato GPS, Pocket Warrior, MacStumbler, KisMAC e anche attraverso il forum online.

WIGLE non gira su reti 802.11a ma su 802.11b o successive, poiché le reti di tipo "a" hanno un raggio troppo ristretto e quindi sono difficilmente raggiungibili.

Per usare WIGLE (con qualsiasi client) é sufficiente registrarsi (nickname e password). Fatto ciò, si possono mandare liberamente le rilevazioni delle reti wireless via client automaticamente o manualmente.

Sempre sotto richiesta (direttamente via mail), é possibile far rimuovere una rete wireless inviata tramite client. L'invio di una nuova rete fornisce un "feedback" positivo al proprio username. Più reti si inviano, più si sale nella classifica degli utenti "benefattori". Non si ha nessun beneficio salendo in classifica, se non un proprio piacere personale (e una menzione sulla homepage di Wigle). Per accedere alla visualizzazione delle reti, si può utilizzare direttamente il sito internet oppure utilizzare uno dei client (anche se necessitano di java 1.3 o superiore, una macchina windows per il client windows o una macchina Apple per il client MacOS). Le mappe così ottenute possono essere scaricate sul client in pacchetti derivati dalla mappa principale del database e vengono riconosciute "per stato e contea" (o stato-città per le città grosse, es: New-York).

La versione online delle query al database é differente e a mio parere, più efficiente. Resituisce una serie di indicatori posizionati su una mappa Google,

esattamente come quando cerchiamo ristoranti, negozi, eccetera. Inoltre tali indicatori sono visualizzabili anche come tabelle in puro HTML. Il database online é interrogabile via browser.

**Query the DB**

**Query for networks**  
Addresses are for the U.S. only (2002 Census data)

Street Address (1600 Pennsylvania Ave):

State (DC):

Zip (20502):

Variance (+/- degrees): 0.010 ▾

Latitude (47.252643):  to:

Longitude (-87.256243):  to:

Last Update (20010925174546):

BSSID/MAC (0A:2C:EF:3D:25:1B):  (Or 1st 3 Octets: 0A:2C:EF)

SSID or Network Name (foobar):

Must Be a FreeNet

Must Be a Commercial Pay Net

Must Have DHCP Enabled

Only Networks I Was the First to Discover

---

**Query for location data of a single network**

BSSID/MAC (0A:2C:EF:3D:25:1B):

Figura 3.2: Form per le query online del database di Wigle

Qui é doverosa una precisazione: l'interrogazione del database tramite form puó avvenire in tre modalitá:

- inserendo Address, Zip Code e Contea (quindi funziona solo per gli Stati Uniti);

- inserendo latitudine e longitudine (e funziona per tutto il mondo);
- inserendo direttamente Net-ID o SSID o indirizzo MAC (e anche così funziona per tutto il mondo).

Anche città non statunitensi sono ricche di reti Wireless rilevate. Bologna ad esempio ne possiede diverse centinaia.

L'unica pecca, forse, è la presenza multipla di indicatori per una rete estesa sia nella versione tabellare HTML che nella versione Google Maps (come Iperbole o Almagi), tuttavia ritengo che non vi sia una soluzione migliore per mostrare la copertura di una rete del genere, quindi Wigle segna tutti i punti sulla mappa da cui è possibile associarsi ad essa. Un altro fattore interessante, è la differenza di colore negli indicatori che indicano se la rete è aperta/libera o privata.

### 3.2.2 Wefi

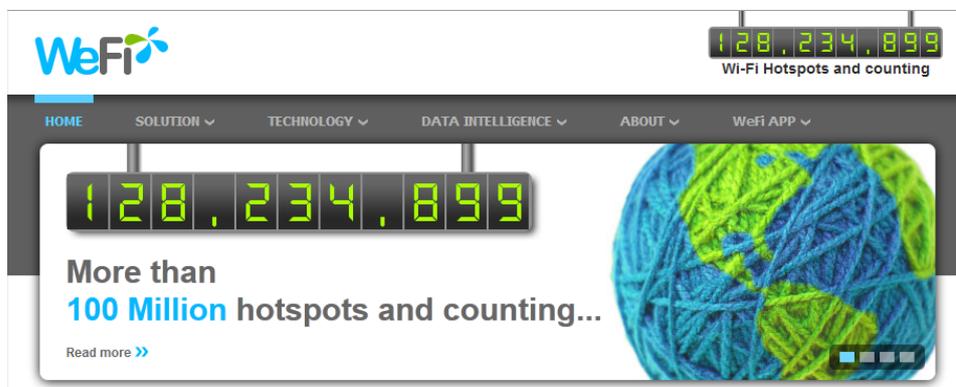


Figura 3.3: Il banner di WeFi

WeFi è un'applicazione web simile a WIGLE, che tiene traccia di tutte le reti wireless segnalate su un database centrale. È un "connection-manager". Lo scopo di WeFi è permettere a chi possiede un client WeFi (disponibile per Android, Windows e Symbian) di connettersi automaticamente alle reti WeFi rilevate, scegliendo la migliore (sia dal punto di vista di potenza che di

sicurezza) senza necessità di cliccare manualmente (o toccare lo schermo se parliamo di touchscreen). Le reti WeFi vengono segnalate come per Wigle, in base al tipo di accesso (libero, previa registrazione, a pagamento). Lo scopo di WeFi è dare la possibilità a chi possiede un client WeFi di far parte di un gruppo di utenti "privilegiati", che possono connettersi automaticamente alle reti Wireless libere e possedere un database mappato di tutte le reti WeFi nel mondo. Le reti dotate di password sono comunque protette. Se qualcuno iscrive la propria rete al database di WeFi, la password rimarrà solamente sul terminale, e WeFi ignorerà la composizione della password. Quindi il client WeFi non salva le password e non le invia a nessuno.

Dal client è possibile anche accedere alla sezione "mappe" impostando una via o uno stato (esattamente come su google maps, poiché si appoggia ad esso) e WeFi restituisce la mappa relativa con evidenziate le reti WeFi, distinguendole fra gratuite (azzurre) e a pagamento (gialle). Tuttavia non mi è stato possibile capire se c'è la possibilità di interagire con il loro database come invece si può fare tramite WIGLE. WeFi è orientato alla commercializzazione pertanto non fornisce i dati del suo database in un formato aperto.

### **3.2.3 Google Maps (Street View)**

Google aveva avviato un progetto di localizzazione di reti WiFi libere per la sezione Google Maps, in modo da permettere ai suoi utenti di rilevare hotspot a libero accesso per collegarsi a internet. Usando le Google Cars infatti, oltre a raccogliere dati per la Street View, i dipendenti dell'azienda di Mountain View avevano registrato anche tutti questi access point.

Niente di illegale all'effettivo (si parla infatti di reti Wireless prive di protezioni e rilevabili senza bisogno di hacking), tuttavia il progetto è stato annullato per motivi di privacy, come ha spiegato il vicepresidente Alan Eustace:

”The engineering team at Google works hard to earn your trust – and we are acutely aware that we failed badly here. We are profoundly sorry for this error and are determined to learn all the lessons we can from our mistake.”

Ossia:

”Il team ingegneristico di Google lavora duramente per guadagnarsi la vostra fiducia – e attualmente siamo coscenti di aver malamente fallito con questo progetto. Siamo profondamente dispiaciuti per questo errore e siamo determinati ad imparare tutte le lezioni possibili dai nostri errori.”

Quindi purtroppo (o per fortuna) Google non corre in aiuto dell’utenza per quanto riguarda la segnalazione delle reti WiFi.

### 3.2.4 Altri Servizi

Esistono altri servizi che a livello locale segnalano la presenza di reti Wireless, ma poter interagire con tutti diventerebbe complicato per via dei diversi formati utilizzati per la rappresentazione dei dati, ed inoltre il fatto che vengano raccolti in modo così differente li renderebbe anche poco attendibili.

Progetti del genere sono ”RomaWireless”, un sito internet molto scarno che segnala su una mappa in stile Google Maps tutti gli access point della rete wireless ”gratuita” (accesso consentito per massimo 2 ore al giorno, tramite account e password previa registrazione sul loro sito) a Roma.

Un altro progetto che sulla carta sarebbe stato utilissimo ma che non é mai sbocciato, é ”Sveglia Italia”, lanciato da Wired, ma mai approdato.



Figura 3.4: La copertina del progetto Sveglia Italia!

Sveglia Italia avrebbe dovuto portare il wireless gratuito in 150 piazze dei comuni d'Italia, in concomitanza con l'anniversario del 150o anno dell'unità nazionale. Tuttavia il progetto non sembra essere stato portato avanti, e tuttora non si trovano notizie concrete su cosa stia succedendo o sia successo a tal proposito.

### 3.2.5 L'Analisi Dei Dati

Per la lettura e la trascrittura dei dati, ho analizzato alcuni servizi già disponibili sul web, capaci di ricavare il contenuto delle pagine web usando semplicemente i click del mouse (click and drag, drag and drop...).

Yahoo fornisce ben due servizi di questo tipo: Dapper e Yahoo Pipes. Sfortunatamente, Dapper è rimasto offline da Gennaio 2012 fino agli inizi di Giugno quindi mi è stato possibile analizzarlo solo nella fase finale della tesi. Dapper permette in 5 passaggi di ottenere dati in vari formati (XML, RSS feed, Google Gadget, Google Maps, ...) anche da più pagine web, e farne un mashup come più ci aggrada. Tuttavia, ha un limite. Sebbene sia possibile effettuare un login, non è possibile ricavare i dati dalle pagine generate (o

accessibili) post-login. Questo rende impossibile usare le query online del portale Wigle.net ad esempio.

Yahoo Pipes invece, presenta un problema ancora piú a monte. Non é possibile inviare dati tramite metodo "POST" di HTTP, quindi non si riesce neanche ad effettuare il login, il che ci riporta al problema sopra descritto.

### **3.3 Il Mio Sistema**

Valutando le varie opzioni disponibili, la mia scelta é ricaduta su Wigle, appoggiandomi ad un client che interagisce con esso.

#### **3.3.1 Perché Wigle**

Wigle é un open project e i sorgenti dei suoi client sono tutti disponibili e non criptati. I dati salvati sui suoi database sono aperti e liberamente consultabili, disponibili in diversi formati ed é necessaria solo l'iscrizione gratuita al sito per potervi accedere.

Durante lo "studio" del sito, ho anche contattato l'amministratore che mi ha risposto poche ore dopo, trasmettendomi subito un segnale di convinzione e funzionalità del progetto.

Inoltre, é un progetto attivo dal 2001 e quindi ha collezionato milioni di reti WiFi, ed é l'unico che ne "fornisce" numerose per le città non americane.

#### **3.3.2 L'interazione con Wigle.net**

L'alternativa che si era posta era fra modificare il client Java di Wigle, oppure creare un analizzatore di pagine Web usando un linguaggio di scripting (PHP, Python, Ruby, ...). La disponibilità di codice già pronto ha reso molto allettante la possibilità di sfruttare il già presente client di Wigle, tuttavia il

gran numero di classi presenti, i commenti al codice forniti dai programmatori orientati piú ai creatori stessi che ad eventuali programmatori esterni, hanno fatto vertire la mia scelta sul creare un piccolo client in un linguaggio web-oriented.

Il mio sistema dunque interagisce con il portale Wigle.net, in tre passi:

- effettua il login sul portale;
- lancia una query basandosi su coordinate geografiche (latitudine/longitudine);
- salva il risultato della query (una tabella HTML), in formato testuale (.txt);

Nei vari linguaggi web-oriented, esistono diverse librerie dedicate alla "lettura" delle pagine HTML (ma anche XML o altri formati), ma quella che mi é sembrata piú immediata é stata Beautiful Soup di Python.

Nel prossimo capitolo vedremo in dettaglio il funzionamento degli script.

## Capitolo 4

# Implementazione

Lo script si basa su due file .txt: uno di input e uno di output.

Nel file di input sono inseriti i set di coordinate necessarie ad effettuare una query su Wigle.net tramite URL. Un esempio di URL é questo:

```
http://wisle.net/gps/gps/main/confirmquery/?latrange1=49.391  
&latrange2=49.114&longrange1=-123.300034&longrange2=-  
122.7495
```

ove la parte in grassetto indica le 4 coordinate (due per la latitudine e due per la longitudine) separate da una '&'.

### 4.1 Python

Ho scelto Python poiché é dotato di un'ampia comunitá attivissima che ha creato svariate librerie per numerosi problemi differenti.

In tutto questo, é un linguaggio semplice nella sintassi una volta che ci si é ambientati, nonché molto permissivo. Ad esempio, i punti e virgola non sono obbligatori alla fine di ogni comando, sebbene sia una buona norma utilizzarli; i cicli invece, sono gestiti in base all'indentazione: tutte le istruzioni

non indentate dentro al comando di ciclo non ne fanno parte.

## 4.2 Librerie

Beautiful Soup (BF), in particolare la versione 4 (quella utilizzata nel progetto), permette in pochissime righe di codice di leggere interi documenti HTML (ma anche XML o altri linguaggi, basta implementare il giusto motore), carpirne i campi e i valori con un solo comando, rendendo chiaramente la lettura di una tabella HTML come quella di Wigle un'operazione velocissima.

BF4 permette non solo di ottenere l'intera pagina HTML, ma anche i singoli campi, singoli elementi o gruppi di elementi dello stesso tipo (ad esempio, tutti gli `<a>` o tutti gli `<h1>`). Per il login invece, ho implementato una libreria `HttpRequest` che, dato il nome del campo della pagina HTML da compilare, è in grado di inserire una stringa ed effettuare l'invio dei dati:

```
"print "Login to Wigle.net...";  
login = functions.httpRequest("https://wagle.net//gps/gps/main/login",  
POSTData = "credential_0": "waglebot", "credential_1": "pro-  
va", "destination": "/gps/gps/main", "noexpire": True);"
```

I campi `credential_0` e `credential_1` sono i campi username e password di Wigle. Dopo aver normalmente creato nome utente e password sul portale, ho poi usato gli stessi campi per lo script della libreria `HttpRequest`.

Il campo `destination` è un elemento di tipo "hidden" di HTML, sempre presente però nel form di login (vedi figura):



Figura 4.1: Codice Sorgente del login di Wigle

Lo script si divide dunque in quattro blocchi:

- lettura del file delle coordinate;
- login su wigle;
- esecuzione della query tramite URL composto dallo script usando le coordinate ottenute dal file;
- lettura della tabella restituita e salvataggio su file di output.

Le librerie utilizzate invece sono tre:

- BeautifulSoup 4 (BS4), la libreria che legge le pagine web basandosi su un motore HTML, XML, etc;
- html5lib, il motore per BS4 per processare l'html;

- `functions`, ossia una libreria di supporto contenente script "ready-to-go" per processare gli URL.

La prima parte del programma dunque legge il file di input, contenente delle semplici coordinate separate da uno spazio. Il programma é in grado di leggere piú set di coordinate, quindi é possibile inviare piú aree da analizzare, cosicché sia in grado di rilevare non solo le Wi-Fi presenti nella direzione in cui si sta muovendo, ma anche in altre aree limitrofe.

La lettura da file di latitudini e longitudini, avviene tramite un semplice ciclo che crea un array con i vari set di coordinate.

Alla fine di questo processo, invocando la funzione `HttpRequest` della libreria "functions", effettuo il login usando il metodo POST di HTTP. I campi "username" e "password" sono prelevati direttamente dal codice sorgente della pagina di Wigle.

A questo punto, si apre il file di output in scrittura, e si avvia un ciclo in cui viene inserita una breve intestazione per ogni set di coordinate per segnalare a quale area corrisponderanno gli access point rilevati, ed utilizzando la stessa funzione "HttpRequest" usata precedentemente, compongo l'URL necessario per la query al database, con i campi correttamente compilati usando le coordinate presenti nel file di input (l'esempio di URL si può vedere all'inizio del capitolo 4 "Implementazione").

All'interno di questo ciclo, per ogni query eseguita, invoco finalmente la potente libreria BS4, grazie alla quale lo script cattura tutti gli elementi "td" dell'html. Di questi, mantengo solo quelli importanti, ossia quelli contenenti SSID, NETID, trilat e trilong, scartando i campi (spesso vuoti) "comment", "type" (che distingue una rete infrarossi da una ad-hoc, che ai fini del nostro utilizzo non importa) o "channel" (ossia quanti indirizzi ip può supportare). Al termine del ciclo, lo script si limita a chiudere il file.

## Capitolo 5

# Sperimentazione

Per questo progetto ho voluto simulare due situazioni "tipo": una situazione urbana, in pieno centro città, dove gli access point sono una "giungla", e Wigle ne mostra tantissimi, e una a bassa copertura di reti WiFi, così si può notare quando il sistema ABPS entra in funzione passando dalla rete Wireless alla rete di dati.

Nelle pagine seguenti affiancherò la visualizzazione delle reti Wireless come su Wigle.net, e il risultato ottenuto dal mio script. Non le elencherò tutte poiché poche saranno sufficienti per capire il funzionamento dello script.

## 5.1 Contesto Urbano

In questo primo esempio ho voluto simulare il percorso (a piedi) da Porta San Donato, a Piazza della Ravennana.

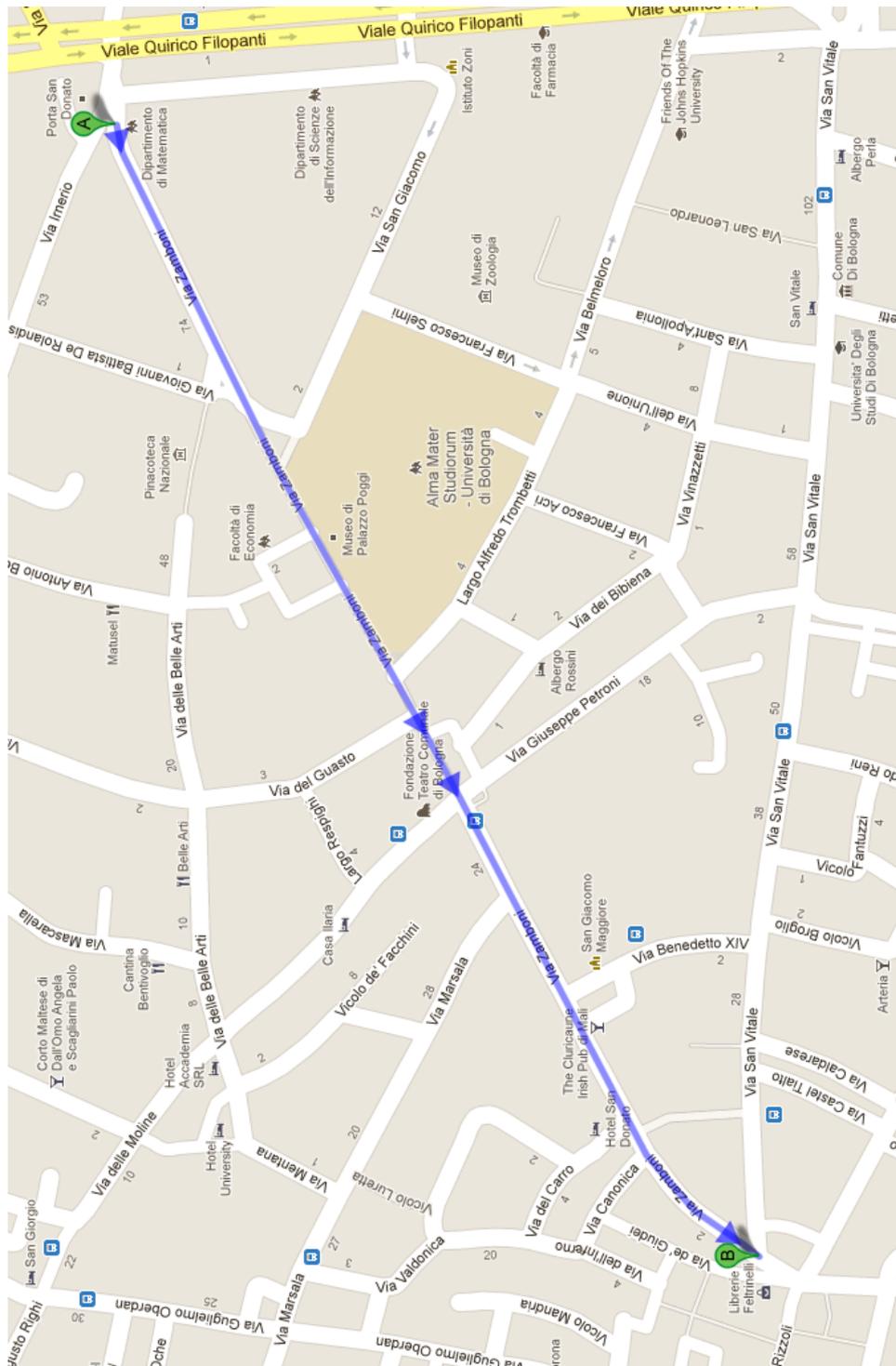


Figura 5.1: Percorso da Porta San Donato a Piazza Della Ravennana



Porta San Donato

# MinLat: 44.4978; MaxLat: 44.4987; MinLong: 11.3553; MaxLong: 11.3569

#

00:1d:6a:b2:f7:6e,Alice-85058814,44.49832153,11.35644436

00:1c:a2:68:e4:bc,Alice-71353422,44.49862289,11.35664368

00:0f:90:8b:7d:a0,ALMAWIFI,44.49792862,11.35639572

00:26:0a:ef:8b:60,ALMAWIFI,44.49848557,11.35545063

00:1c:58:10:1d:a0,ALMAWIFI,44.49868774,11.35660362

00:1c:58:10:1d:30,ALMAWIFI,44.49833298,11.35614967

38:22:9d:fe:2c:84,FASTWEB-1-38229DFE2C84,44.49849319,11.35669899

02:24:01:4e:e1:4b,InfostradaWiFi,44.49866867,11.35635567

00:1c:58:10:17:00,ALMAWIFI,44.49867249,11.35670185

00:1d:6a:85:65:49,Alice-76514377,44.49824905,11.35687923

00:1d:8b:c8:28:5a,Alice-15646926,44.49831009,11.35658932

00:1d:8b:4e:c8:84,FASTWEB-1-001D8B4EC884,44.49862671,11.35679817

00:1c:10:90:ae:ea,linksys,44.49865341,11.35660172

00:1d:8b:67:5c:4c,Alice-56427118,44.49827576,11.35680676

00:0f:3d:34:bb:3c,bruno,44.49846268,11.35664463

00:15:e9:0b:26:62,default,44.49820328,11.35592651

00:1c:58:10:1d:70,ALMAWIFI,44.49827957,11.35678291

00:1d:6a:d7:c3:58,Alice-60921966,44.49831390,11.35680103

00:1c:58:10:1c:20,ALMAWIFI,44.49842072,11.35589314

00:1c:58:10:1c:60,ALMAWIFI,44.49846649,11.35645580

00:1c:58:10:13:f0,ALMAWIFI,44.49859619,11.35594463

00:18:84:88:e3:e8,ORANGE,44.49863434,11.35684681

00:1c:58:10:1d:80,ALMAWIFI,44.49847031,11.35545444

02:24:01:4e:a8:f7,qui quo qua,44.49859238,11.35671806

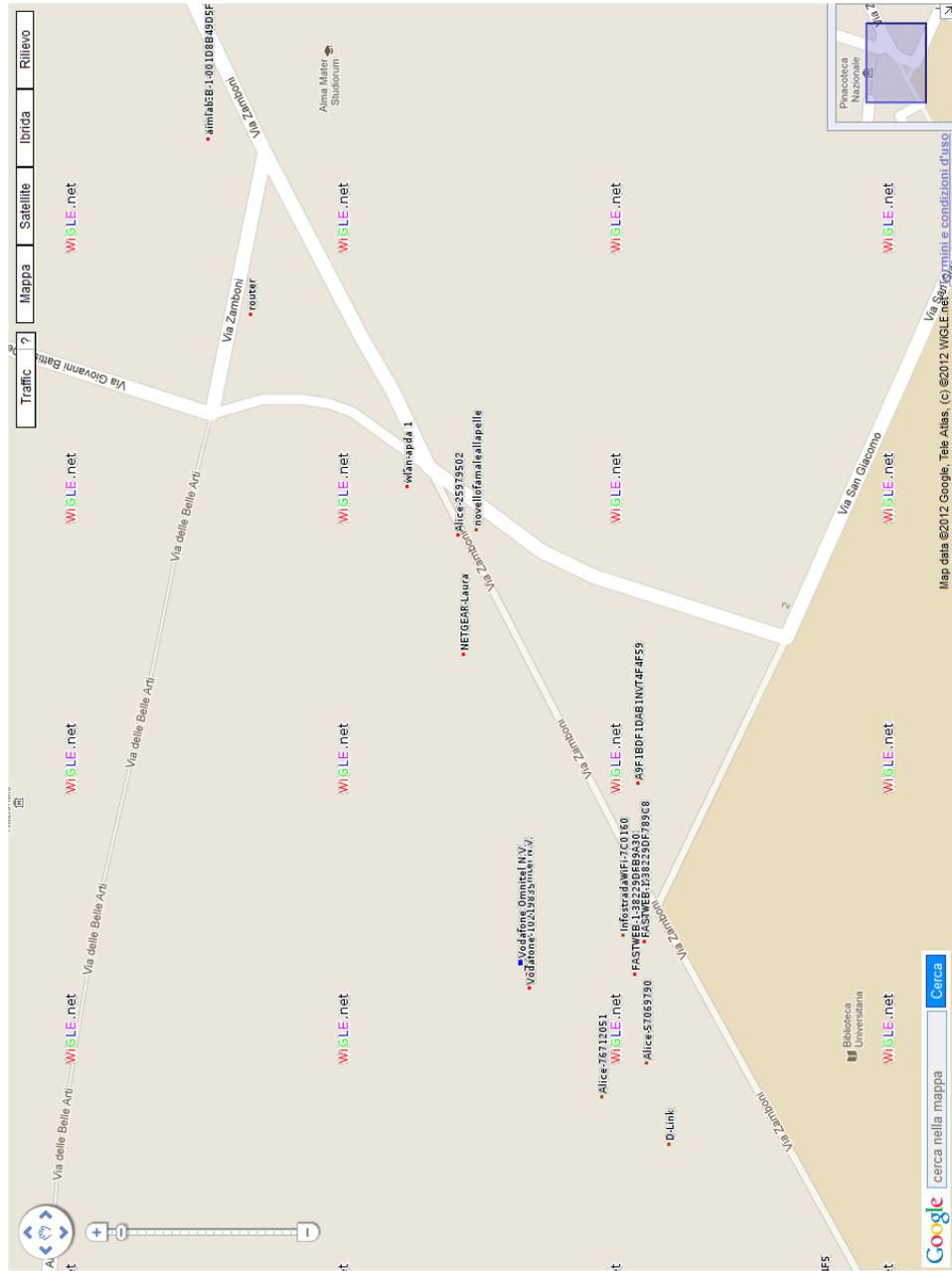


Figura 5.3: Via San Giacomo

Via Zamboni Incrocio San Giacomo

# MinLat: 44.4969; MaxLat: 44.4978; MinLong: 11.3526; MaxLong: 11.3542

#

00:1d:6a:88:89:d5,Alice-76720341,44.49722672,11.35305023

00:26:44:7a:c0:fc,Infostrada 1,44.49744034,11.35362816

00:16:b6:52:69:05,aimlab,44.49761963,11.35406971

f0:7d:68:11:30:4a,D-Link,44.49720383,11.35279274

00:1c:a2:58:74:09,Alice-57069790,44.49722290,11.35289574

00:1c:a2:bc:74:40,novellofamaleallapelle,44.49737930,11.35357380

00:04:ed:d2:41:0d,wlan-ap,44.49744034,11.35362816

00:04:ed:ba:00:7f,router,44.49758148,11.35384655

00:1d:6a:88:69:73,Alice-76712051,44.49726486,11.35285378

00:21:63:D3:6A:C3,Vodafone-10219835,44.49732971,11.35299110

00:1d:8b:4b:6d:3c,FASTWEB-1-001D8B4B6D3C,44.49723434,11.35300922

50:67:F0:45:85:CC,ZyXEL,44.49733353,11.35300922

38:22:9d:fb:9a:30,FASTWEB-1-38229DFB9A30,44.49723434,11.35300922

00:26:44:7c:01:60,InfostradaWiFi-7C0160,44.49724579,11.35305786

00:1f:33:42:e4:30,NETGEAR-Laura,44.49739075,11.35341549

00:1c:a2:69:61:9b,Alice-25979502,44.49739456,11.35356712

00:1d:8b:49:d5:f4,FASTWEB-1-001D8B49D5F4,44.49761963,11.35406971

38:22:9d:fb:7f:1c,FASTWEB-1-38229DFB7F1C,44.49723434,11.35300922

38:22:9d:fa:40:ac,FASTWEB-1-38229DFA40AC,44.49722672,11.35305023

8e:c4:25:f3:3f:00,A9F1BDF1DAB1NVT4F4F59,44.49723053,11.35325146

38:22:9d:f7:89:c8,FASTWEB-1-38229DF789C8,44.49722672,11.35305023



## Piazza Scaravilli

# MinLat: 44.4965; MaxLat: 44.4974; MinLong: 11.3512; MaxLong: 11.3528

#

00:1e:4a:07:c8:00,CICU,44.49669266,11.35152149

1C:AA:07:B1:D4:D0,ALMAWIFI,44.49696350,11.35225868

00:1E:4A:07:C8:10,ALMAWIFI,44.49665070,11.35152531

00:1c:58:10:1a:80,ALMAWIFI,44.49673080,11.35159302

00:1d:6a:85:c7:2f,Alice-76539439,44.49650955,11.35167313

00:1c:58:10:14:a0,ALMAWIFI,44.49695587,11.35190201

00:19:06:e0:15:60,ALMAWIFI,44.49702835,11.35224342

00:1e:58:97:8d:48,iZoni,44.49706650,11.35222054

00:1c:58:10:17:80,ALMAWIFI,44.49677658,11.35165787

00:25:45:b5:3d:30,ALMAWIFI,44.49681091,11.35178757

00:19:5b:8c:a8:82,iZoni,44.49656296,11.35169506

00:14:7c:bb:e5:89,AP,44.49654007,11.35171604

00:1e:4a:07:f8:f0,ALMAWIFI,44.49698257,11.35223770

a2:72:c8:20:c4:c1,HP43F367,44.49706650,11.35247993

f0:7d:68:11:30:4a,D-Link,44.49720383,11.35279274

38:22:9d:2f:9e:64,Alice-64527809,44.49662399,11.35204220

00:1c:58:10:1d:e0,ALMAWIFI,44.49706650,11.35247993

00:1c:58:10:18:20,ALMAWIFI,44.49665451,11.35146999

20:2B:C1:59:B3:5C,InfostradaWiFi-009805,44.49709320,11.35227680

00:3A:9A:0E:8A:C0,ALMAWIFI,44.49668884,11.35152149

00:1C:58:10:16:70,ALMAWIFI,44.49665070,11.35152531

00:1c:58:10:19:30,ALMAWIFI,44.49690628,11.35203266

00:1c:58:10:14:70,ALMAWIFI,44.49699020,11.35227871

00:1c:58:10:16:60,ALMAWIFI,44.49665451,11.35146999



## Piazza Verdi

# MinLat: 44.4957; MaxLat: 44.4965; MinLong: 11.3496; MaxLong: 11.3512

#

02:1b:e4:e4:e2:03,Iperbole Wireless,44.49628067,11.35038280

02:1b:e4:e4:e2:04,GooMobile.it,44.49628067,11.35038376

00:19:cb:f4:fd:ef,ZyXEL G-570S,44.49584961,11.35073185

00:04:ED:66:9C:3D,wlan-ap,44.49608231,11.35014820

c0:3f:0e:a4:86:94,giorgioefrancesca,44.49623489,11.35031700

00:1c:58:10:1a:a0,ALMAWIFI,44.49642181,11.35095024

00:24:36:ae:14:91,wtcbo-apple,44.49609756,11.35003471

00:21:63:A8:96:B2,Vodafone-10175266,44.49620438,11.34996986

00:12:0E:BB:C9:3C,Untitled,44.49616623,11.34987164

00:90:d0:f9:08:bf,SpeedTouch8350DB,44.49619675,11.35023689

00:c0:02:4b:77:ec,Admin,44.49627686,11.35021973

64:87:D7:E0:DB:D0,Alice-32031491,44.49590302,11.34978867

1c:af:f7:9b:77:c8,magnifico,44.49613190,11.35081100

00:16:47:0c:d8:60,ALMAWIFI,44.49623489,11.35031700

00:1c:58:10:18:40,ALMAWIFI,44.49649429,11.35114288

00:18:4d:bf:d4:08,Angelini,44.49584961,11.35087681

00:24:FE:43:F4:E4,Fritz! Ehiweb.it - 000012,44.49637222,11.35010719

00:13:33:A4:54:26,S. Giacomo Maggiore,44.49611664,11.35004520

00:1C:F0:B1:BE:D6,dlink,44.49593735,11.34977531

00:21:96:22:95:e0,FASTWEB-1-0021962295D8,44.49627686,11.35040379

00:1c:58:10:18:c0,ALMAWIFI,44.49649048,11.35111809

00:15:e9:27:b7:4a,SANSYS29,44.49587631,11.35071659

00:01:38:88:63:6a,LiberoWifi,44.49607849,11.35000896

00:13:33:8e:cb:f1,RTL867x-ADSL,44.49612808,11.35091019



Piazza Rossini

# MinLat: 44.4952; MaxLat: 44.496; MinLong: 11.3481; MaxLong: 11.3497

#

F4:EC:38:9D:6C:4E,GooMobile.it,44.49550629,11.34845448

02:0C:42:F9:01:CB,Iperbole Wireless,44.49534988,11.34834671

00:0C:42:F9:01:CB,GooMobile.it,44.49535751,11.34836006

00:1a:30:31:50:a0,provbo,44.49553299,11.34875774

00:1d:6a:85:26:cf,Alice-76498383,44.49595642,11.34944916

00:1d:6a:71:31:47,Alice-75190343,44.49557495,11.34858990

00:1d:8b:4a:7b:f4,FASTWEB-1-001D8B4A7BF4,44.49562454,11.34844494

00:1d:6a:86:a8:58,Alice-76597080,44.49575043,11.34923458

C0:C1:C0:1A:29:62,cerere,44.49562454,11.34892464

00:3A:9A:0E:BD:70,ALMAWIFI,44.49578476,11.34914684

00:1f:9e:d5:ff:20,ALMAWIFI,44.49543762,11.34823036

30:39:F2:D0:64:3F,Telecom-47211577,44.49537277,11.34838581

00:3a:98:88:39:00,ALMAWIFI,44.49574661,11.34920883

00:3a:99:33:cb:80,SanDonato Guest,44.49540710,11.34822845

00:90:d0:dd:c9:4f,ciaomacca,44.49547958,11.34869957

c2:ce:46:57:63:2b,hpsetup,44.49551392,11.34825897

c4:71:fe:b3:8a:d0,SanDonato Guest,44.49551392,11.34825897

00:18:f8:a4:1b:cb,linksys,44.49596405,11.34964943

00:16:0a:09:7d:9c,Sweex LW050v2,44.49557114,11.34879494

00:23:8e:87:c8:ac,Alice-52492657,44.49592209,11.34960365

5c:33:8e:21:d9:df,Alice-37099487,44.49562454,11.34844494



Piazza Della Ravegnana (Due Torri)

# MinLat: 44.4941; MaxLat: 44.4949; MinLong: 11.3459; MaxLong: 11.3475

#

00:1D:7E:AC:91:CE,LinkWi-Fi,44.49444580,11.34673309

00:25:53:91:3c:b5,InfostradaWiFi,44.49456406,11.34657669

00:1d:8b:73:b6:6c,Alice-37339897,44.49456787,11.34668350

00:1d:8b:4e:bf:8c,FASTWEB-1-001D8B4EBF8C,44.49468613,11.34674263

30:39:F2:B7:72:2F,Telecom-45576745,44.49469757,11.34603024

38:22:9d:39:cd:a0,BT00215389,44.49441147,11.34625530

00:01:36:A6:23:2B,FASTWEB-1-8AyLuUxaxYFs,44.49478912,11.34702969

00:21:29:67:e9:d0,CHLOCAL2K9,44.49486542,11.34701824

4A:7F:87:F1:3F:8D,portthru,44.49431992,11.34658718

00:19:cb:ec:46:64,ZyXEL,44.49446869,11.34625721

30:39:F2:B5:7B:97,Telecom-45448081,44.49478912,11.34702969

00:1d:6a:6f:3b:9b,Alice-75061915,44.49467850,11.34681320

00:24:01:38:4a:f8,dlink,44.49489975,11.34717655

00:1f:9f:47:85:b1,MiSE DE DGERM DIIUNMIGBO,44.49438858,11.34648037

00:23:8e:87:c8:d8,Alice-52491225,44.49467850,11.34686470

00:1d:8b:61:ed:7c,Alice-96432479,44.49434280,11.34655857

1c:bd:b9:b7:ce:3c,Emicon,44.49440002,11.34654140

00:1d:8b:6e:c7:10,Alice-58137905,44.49468613,11.34680176

00:1B:2F:AA:91:78,luna,44.49478912,11.34702969

00:21:29:67:d5:c0,CHLOCAL2K9,44.49487686,11.34707546

00:21:91:34:50:f3,dlinkSV1,44.49430847,11.34652424

00:1e:8c:7e:ea:be,coldbo1,44.49458313,11.34665298

00:13:49:a3:04:95,ZyXEL,44.49466324,11.34743404

02:24:01:4d:b2:af,InfostradaWiFi,44.49418640,11.34650040

Come si può notare, un ipotetico utente che si muove lungo questo percorso, sarebbe sempre coperto da un access point. Il sistema ABPS sarebbe in grado di rilevarli in anticipo e connettere automaticamente l'utente da un hotspot al successivo senza che questo si renda conto del passaggio da una rete WiFi ad un'altra.

Se immaginiamo che il soggetto in questione sia uno studente dell'Alma Mater, con credenziali quindi adatte all'accesso di ALMAWIFI, possiamo notare come in quattro sezioni su sei sia disponibile tale rete wireless. Dunque, immaginando che inizi il suo percorso connesso a tale rete all'altezza della facoltà di matematica (all'incrocio fra via Zamboni e mura Anteo Zamboni, nell'immagine 5.2), sarà costretto a cambiare rete quando passa nella zona di via San Giacomo (5.3), per poi tornare ad Almajifi all'altezza di Piazza Scaravilli (5.4), mantenerla in piazza Verdi (5.5) e ancora in piazza Rossini (5.6) e cambiare nuovamente nei pressi di piazza della Ravennana (5.7).

Adesso andremo ad analizzare un percorso molto meno coperto da reti wireless.

## 5.2 Contesto a Bassa Densità di Hotspot

Nel secondo esempio, osserveremo come il sistema ABPS sia fondamentale per mantenere persistente una connessione internet in un contesto dove il wireless lascia spazio alla rete di dati. Il percorso scelto inizia all'ingresso dei Giardini Margherita (Porta Santo Stefano) e termina all'incrocio fra via Murri e via Dello Sterlino.



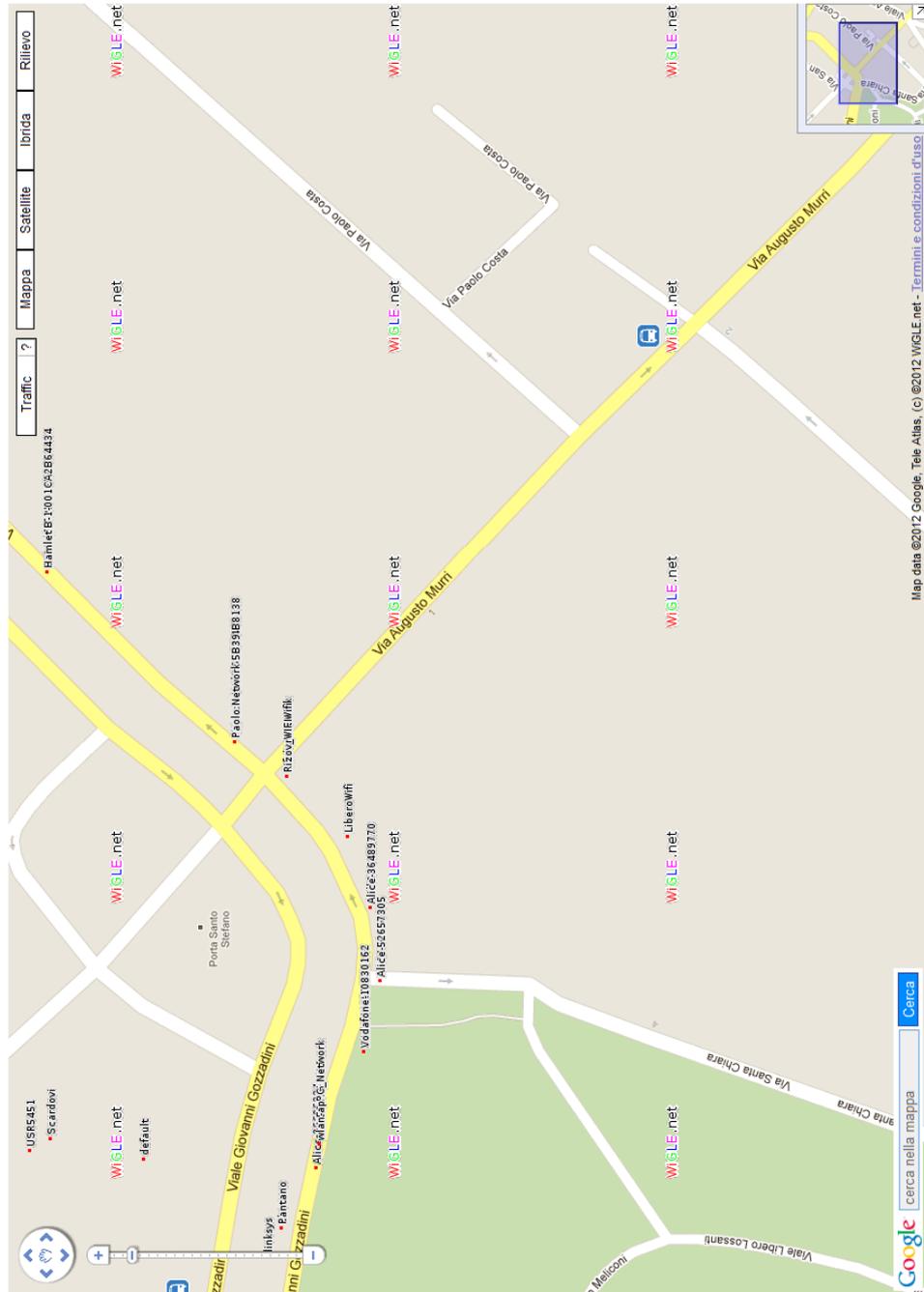


Figura 5.9: Porta Santo Stefano - Via Murri

```
# MinLat: 44.4833; MaxLat: 44.485; MinLong: 11.355; MaxLong: 11.3582
#
00:90:d0:db:ef:b2,SpeedTouch805B39,44.48456955,11.35662842
00:04:ed:62:d0:35,wlan-ap,44.48490143,11.35705280
00:c0:49:f1:ce:9e,USR5451,44.48493195,11.35560894
00:26:5a:87:5d:72,D-Link,44.48434067,11.35585594
00:1d:6a:70:bf:17,Alice-75161111,44.48442078,11.35562515
00:03:6f:3d:eb:00,-1-00036F3DEAF8,44.48490143,11.35705280
00:25:53:04:57:f8,Alice-52657305,44.48431396,11.35603333
00:1e:52:7b:e6:92,Apple Network,44.48447800,11.35654259
00:0f:3d:42:99:10,default,44.48472977,11.35558701
5c:33:8e:18:8c:2a,Alice-36489770,44.48432922,11.35621548
00:24:89:25:03:c9,Vodafone-10830162,44.48434067,11.35585594
00:24:b2:49:26:fe,Pantano,44.48448944,11.35541630
00:1f:1f:10:62:c9,Default,44.48490143,11.35705280
38:22:9d:33:a8:b0,Alice-64887889,44.48434067,11.35585594
00:1f:f3:f4:48:c3,Apple Network,44.48456955,11.35662842
00:1d:8b:4b:81:38,FASTWEB-1-001D8B4B8138,44.48456955,11.35662842
00:1c:a2:dc:dd:15,InfostradaWifi,44.48447800,11.35654259
00:04:ed:90:b6:0f,wlan-ap,44.48442078,11.35562515
00:18:39:28:01:b6,Elettra,44.48448944,11.35541630
00:18:39:2f:78:36,linksys,44.48451233,11.35534668
00:40:05:55:4d:e7,PC34-NETWORK,44.48431396,11.35603333
00:1d:0f:f4:b3:27,Hamlet,44.48490143,11.35705280
00:19:e0:67:dd:19,Hamlet,44.48432922,11.35621548
90:84:0d:d7:19:21,BullCar,44.48432922,11.35621548
00:25:53:f8:5b:6c,Alice-63280633,44.48434067,11.35585594
```



# MinLat: 44.4813; MaxLat: 44.4829; MinLong: 11.3584; MaxLong:  
11.3616

#

C0:C1:C0:F9:BB:47,Bologna Wi-Fi,44.48188400,11.36038113

38:22:9D:FE:F9:88,FASTWEB-1-38229DFEF988,44.48188400,11.36038113

00:1D:6A:AD:A1:B9,Alice-79151289,44.48268509,11.35913467

98:FC:11:82:D7:C9,StudioVecchietti,44.48268509,11.35913467

00:1D:8B:51:CB:A4,SLOWWEBGmB,44.48277283,11.35902882

00:25:53:03:04:7C,Alice-52575577,44.48170853,11.36050034

00:19:5B:77:8A:ED,D-Link,44.48278046,11.35910988

00:14:7C:BC:C2:6B,GSSLAW,44.48188400,11.36038113

00:1A:70:95:B0:66,crislian,44.48277283,11.35902882

00:24:89:0F:3B:5F,Vodafone-10523323,44.48278046,11.35910988

F0:7D:68:FB:C8:A3,D-Link,44.48188400,11.36038113

In questo secondo e piú breve esempio, si nota come il sistema rilevi diverse reti wireless, ma essendo esse concentrate all'inizio e alla fine del percorso, il dispositivo mobile equipaggiato con il sistema ABPS si ritroverá a passare da una rete WiFi, alla rete di dati, ad una rete WiFi nuovamente quando raggiungiamo la destinazione.

## Capitolo 6

# Conclusioni e Sviluppi Futuri

Il sistema ABPS é indubbiamente un potente strumento di connettività, che permette ad un utente di mantenersi sempre connesso sfruttando la soluzione migliore.

La possibilità di passare da una rete ad un'altra senza accorgersene e mantenere sempre attiva una connessione Internet, rende ABPS un sistema all'avanguardia e utilissimo ai fini di svariate applicazioni, soprattutto per il lato "mobile" della connettività.

Il mio script si limita ad aggiungere un'appendice ad ABPS, permettendogli di "prevedere" le locazioni delle reti wireless, risparmiando tempo sulla parte di rilevazione. Ciò é reso possibile dai parametri di latitudine e longitudine, che consentono di cercare gli hotspot in un'area precisa, evitando che il terminale diffonda segnali attorno in cerca di qualsiasi access point in quanto già rilevato e salvato sul database di Wigle.net.

Un altro miglioramento potrebbe essere la sostituzione del file in input con il sistema GPS di cui sono dotati i terminali mobili. Il GPS sarebbe in grado di localizzare il dispositivo e con un semplice algoritmo, si potrebbero cercare gli hotspot in un'area circoscritta a quella attuale (sempre sfruttando Wi-

gle), in modo da essere pronti al passaggio di connessione nel momento in cui l'utente si sposta. Chiaramente il refresh delle reti rilevate raggiungerebbe la massima efficienza se venisse effettuato con un intervallo di tempo ragionevolmente breve quando l'utente si sposta a velocità sostenuta (ad esempio, in automobile), e più lentamente quando lo spostamento dell'utente avviene ad una velocità più contenuta.

Un'alternativa ancora migliore sarebbe l'esecuzione del processo sopradescritto non appena il sistema GPS rileva un cambiamento della posizione del terminale, invece che seguendo un intervallo di tempo.

Durante lo sviluppo dello script, la scelta dell'uso della libreria Beautiful Soup 4 si è dimostrata vincente, semplificando di molto il lavoro di programmazione e concedendomi più tempo per test e verifiche di interoperabilità.

Inoltre, Python dispone di svariati emulatori che lo rendono compatibile con i vari sistemi operativi mobili, come Android, iOS e Symbian, e la conversione per il mobile dovrebbe essere quindi più semplice che la "traduzione" in un altro linguaggio di programmazione.

# Bibliografia

[Antonio Baglivo e Giuseppe De Marco, 2010] Politiche di selezione di Access Point

[Leonard Richardson, 2012] Beautiful Soup 4 Documentation:  
<http://www.crummy.com/software/BeautifulSoup/bs4/doc/>

[Python Software Foundation, 2012] Python Documentation:  
<http://docs.python.org/library>

[Vittorio Ghini, Stefano Ferretti, Fabio Panzieri, 2011] The Journal of Systems and Software: The "Always Best Packet Switching" architecture for SIP-based mobile multimedia services

[Stefano Rizzo, 2010] Sicurezza del sistema ABPS in Symbian

[Wikipedia, 2012] r1f1 Seamless protocol: [http://en.wikipedia.org/wiki/ITU\\_G.992.3/4](http://en.wikipedia.org/wiki/ITU_G.992.3/4)

[ProgrammableWeb, 2012] Google Maps Mashups:  
<http://www.programmableweb.com/api/google-maps/mashups>

[ProgrammableWeb, 2012] Ennovation HotSpot:  
<http://www.programmableweb.com/mashup/innovation-hotspot>

- [ProgrammableWeb, 2012] WhereAndWifi.com:  
<http://www.programmableweb.com/mashup/whereandwifi.com>
- [gWiFi, 2012] Using Google Maps to find free WiFi:  
<http://www.gwifi.net/>
- [RomaWireless, 2012] RomaWireless: <http://www.romawireless.com/hotspot.htm>
- [Dynamick, 2012] Wefi, la mappa degli hotspot WiFi liberi:  
<http://www.dynamick.it/hotspot-wifi-liberi-wefi-842.html>
- [Wigle, 2012] Wireless Geographic Logging Engine: Making maps of wireless networks since 2001: <http://www.wigle.net>
- [WeFi, 2012] Wi-Fi Hotspots and counting:  
<http://http://www.wefi.com/>
- [HTML.it, 2012] Guida Python: <http://programmazione.html.it/guide/leggi/39/guida-python/>