



UNIVERSITÀ DEGLI STUDI DI GENOVA  
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

*Monitoraggio di flussi e densità pedonale in zone  
congestionabili mediante sensori innovativi*

TESI DI LAUREA MAGISTRALE

Gianluca Ceccoli

Relatore:

Prof. Roberto Sacile

Correlatore:

Dott. Luca Zero

ANNO ACCADEMICO 2018/2019

# Sommario

Sommario .....	2
1. Introduzione .....	4
2. Le Cinque Terre.....	5
3. Stato dell'arte .....	6
3.1. Dispositivi .....	7
3.1.1. Videocamere.....	7
3.1.2. Infrarossi e laser .....	8
3.1.3. Sniffer .....	9
3.2. Progetti di monitoraggio .....	10
3.2.1. SAIL Amsterdam (2015).....	10
3.2.2. Pellegrinaggio Hajj (2016).....	11
3.2.3. Il caso Venezia (2018).....	12
3.3. Tecnologia preesistente in loco.....	13
4. Caso di studio: Wi-Fi .....	14
4.1. MAC Address .....	14
4.2. Randomizzazione.....	16
4.2.1. Android .....	16
4.2.2. Apple.....	18
4.2.3. Windows.....	20
4.2.4. Linux.....	20
5. Prima fase del progetto .....	21
5.1. Fingerprinting .....	21
5.2. Aggregazione .....	22
5.3. Indipendenza .....	23
6. Seconda fase del progetto.....	25
6.1. Gestione RAM.....	25
6.2. Riferimenti spazio-temporali.....	25
6.3. Applicazione Web GIS.....	26
6.4. Collocamento degli sniffer.....	29
6.4.1. Monterosso .....	30
6.4.2. Vernazza .....	31
6.4.3. Corniglia.....	32
6.4.4. Manarola .....	33
6.4.5. Riomaggiore.....	34
6.5. Autonomia degli sniffer .....	36

7.	Dettagli tecnici.....	38
7.1.	Specifiche tecniche .....	38
7.2.	Monitor mode .....	39
7.3.	Script Python .....	42
7.4.	Database.....	44
7.4.1.	Login .....	44
7.4.2.	Sensors_colocation.....	45
7.4.3.	log .....	45
7.4.4.	logv3 .....	46
7.5.	Applicazione Web.....	47
8.	Terza fase del progetto.....	50
9.	Considerazioni finali .....	52
9.1.	Precisione .....	52
9.2.	Accuratezza.....	52
9.3.	Portata massima.....	52
9.4.	Perdita informazioni .....	52
10.	Conclusione .....	52
11.	Bibliografia.....	53

# 1. Introduzione

La crescita sempre più rapida della popolazione nel mondo e la conseguente urbanizzazione hanno portato a un incremento nel numero di attività come eventi sportivi, raduni politici, dimostrazioni pubbliche ecc. che risultano in più frequenti raduni di folle. In questi scenari, la Crowd Analysis sta ricevendo crescente attenzione per poter assicurare miglior gestione della folla o riconoscimento automatico di situazioni potenzialmente pericolose. Il rilevamento di situazioni anormali può migliorare significativamente l'efficienza della sorveglianza, risparmiando l'attenzione umana per i casi più importanti, previa segnalazione di un sistema.

Il costo sempre più accessibile dei sistemi di videosorveglianza ha come conseguenza che i metodi tradizionali per il monitoraggio del traffico pedonale facciano affidamento su tecnologie basate su acquisizione video o contapersone (fotocellule o barriere laser) che tuttavia sollevano questioni estetiche o di sicurezza per i dispositivi stessi, oltre a quelle legate alla privacy delle immagini registrate. Come ogni altro problema di Computer Vision, la Crowd Analysis comporta molte sfide da affrontare come occlusioni, distribuzione non uniforme delle persone, illuminazione variabile o prospettive sfavorevoli che rendono l'analisi particolarmente difficile.

D'altra parte, l'ammodernamento dei dispositivi mobili e la propensione dell'uomo a possederne uno, ha permesso che il monitoraggio dei cellulari possa competere nel campo della stima dei movimenti del traffico pedonale. Se il sistema di rilevazione è dotato di ricevitori Wi-Fi, è possibile catturare l'origine-destinazione (O-D) di un pedone, il suo travel time, wait time ricavando informazioni sul flusso di un sottoinsieme di pedoni, possessori di devices con modulo Wi-Fi acceso.

Lo scopo principale di questo elaborato è quello di studiare la fattibilità di implementazione di un sistema di monitoraggio di flussi impiegando degli sniffer Wi-Fi i cui dati saranno integrati con quelli disponibili da altri mezzi di monitoraggio preesistenti, che consenta, in tempo reale o semi-reale, di stimare l'afflusso pedonale in aree particolarmente predisposte a congestiamenti come i borghi delle Cinque Terre. Lo scopo secondario è quello costruire un dataset in modo da essere in grado in futuro di effettuare predizioni circa l'andamento dell'affluenza. Per realizzare il progetto è necessario studiare lo stato dell'arte attuale nel campo della randomizzazione del MAC address per garantire quanto più possibile la veridicità dei dati raccolti dai sensori e al contempo mettere alla prova la sicurezza riguardo la privacy che i produttori di hardware e software vorrebbero garantire impiegando la randomizzazione sopracitata.

L'idea è quella di installare più stazioni Wi-Fi, inizialmente in posizioni strategiche presso i cinque borghi per poi valutare una possibile specializzazione presso uno dei paesi. Una prima fase consiste nella definizione del network di sensori atti al monitoraggio dei flussi turistici. Una seconda fase prevede la progettazione e realizzazione di una base di dati per gestire in modo centralizzato i dati provenienti dalla rete di sensori. Come ultimo passo, sarà necessario rendere i dati disponibili in forma grafica agli utenti accreditati mediante un ambiente WebGIS.

Nel secondo capitolo verrà fornita una breve panoramica delle Cinque Terre e della loro topografia. Nel terzo capitolo verrà discusso lo stato dell'arte riguardo i dispositivi di monitoraggio, i passati esperimenti di stima del flusso pedonale in grandi eventi e le risorse già esistenti in loco, disponibili ad essere integrate nel sistema di monitoraggio alle Cinque Terre. Nei capitoli 4 5 e 6 si approfondirà il caso di studio, con maggiori informazioni riguardo il MAC address e la sua randomizzazione e con la spiegazione di come sia stato sviluppato il sistema. Nel capitolo 7 verrà fornita una spiegazione tecnica di come sia possibile eseguire lo sniffing dei dispositivi mobili. Nell'ottavo ...

## 2. Le Cinque Terre

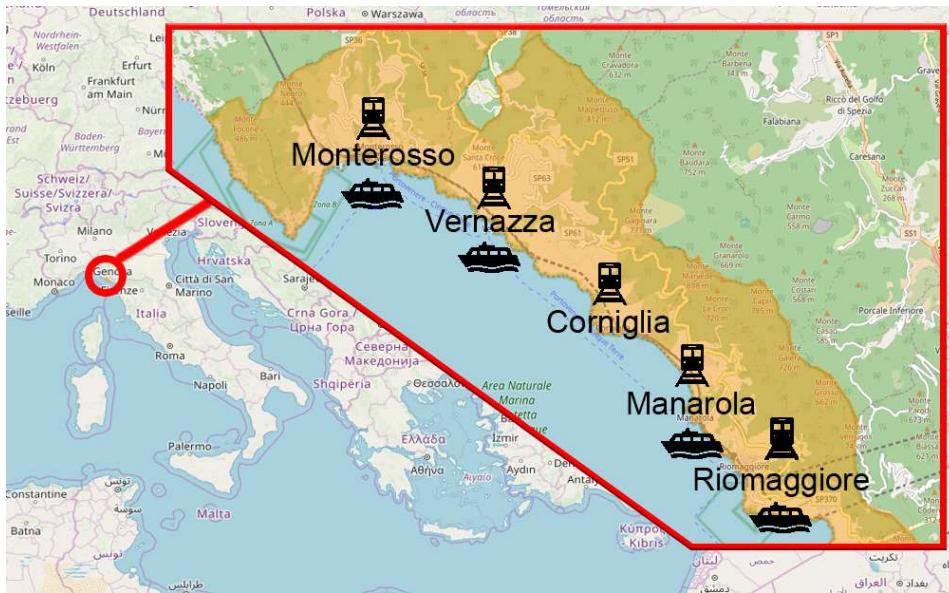


Figura 1 Overview del Parco Nazionale delle Cinque Terre

Il Parco Nazionale delle Cinque Terre è un frastagliato tratto di costa della Riviera ligure di levante situato nel territorio della provincia di La Spezia tra Punta Mesco e Punta di Montenero, nel quale si trovano cinque borghi, da ovest verso est: Monterosso al Mare, Vernazza, Corniglia, Manarola e Riomaggiore. Nel 1997 l'UNESCO ha inserito le Cinque Terre nella lista del Patrimonio Mondiale dell'Umanità come "paesaggio culturale".

Gran parte dei parcheggi sono riservati ai residenti e in ragione della mancanza di spazi adeguati, la discesa ai borghi è possibile solo a piedi. Esiste tuttavia un servizio di bus elettrici gestito dal Parco Nazionale che nei periodi di maggior afflusso turistico garantisce i collegamenti. Nella stagione turistica un servizio di battelli di linea collega tutti i borghi ad eccezione di Corniglia (che non è situata sulla costa) con Porto Venere, a sua volta collegata con La Spezia, Lerici e con gli altri paesi della riviera di Levante. Data la situazione geografica piuttosto impervia dei borghi, il mezzo di trasporto di gran lunga più utilizzato per raggiungerli è la ferrovia che fa tappa in tutte le terre e vede una notevole frequenza di transiti. Una volta nei pressi dei paesi, esistono due sentieri principali che i turisti percorrono per visitare il Parco. Il primo (sentiero numero 1) è conosciuto come Alta via delle Cinque Terre e collega Levanto con Portovenere. Permette di percorrere un tragitto più elevato e nell'entroterra rispetto al secondo ma è attraversato da molti sentieri trasversali che consentono di raggiungere la costa. Il secondo sentiero (sentiero numero 2), meglio conosciuto come Sentiero Azzurro, è quello più apprezzato dai turisti. Collega le Cinque Terre con un percorso sul litorale e regala parecchi scorci panoramici. Visto il suo successo, l'accesso è regolato tramite l'acquisto della Cinque Terre Card.

A causa della storicità dei sentieri e per motivi di sicurezza, alcuni tratti vengono spesso chiusi/manutenuti secondo le necessità più o meno impreviste. [1]

### 3. Stato dell'arte

La Crowd Analysis può essere effettuata su due livelli: quello microscopico e quello macroscopico. Il primo si focalizza sul singolo individuo all'interno di una folla e come per esempio questo interagisca con gli altri pedoni o con l'ambiente in cui è immerso con modelli che sono computazionalmente più onerosi rispetto a quelli macroscopici. Per contro, i modelli macroscopici tendono a considerare la folla come se fosse un fluido [2] e pertanto forniscono misure più approssimate. Ai diversi punti di vista corrispondono due facce della stessa medaglia per ogni problema che si considera. Quelli a cui siamo interessati sono il conteggio di persone e il tracking dei loro movimenti, che secondo l'interpretazione microscopica assumono il significato rispettivamente di LOI (line-of-interest counting), che conta il numero di persone che passano attraverso una data linea virtuale, e del tracking individuale di ogni pedone registrando la sua velocità istantanea e la sua direzione. Se invece consideriamo la prospettiva macroscopica questi due problemi si identificano con ROI (region-of-interest counting), che conta il numero di persone in una data regione, e con il tracciamento generico della folla, registrando la sua velocità media e direzione del flusso.

Un importante aspetto delle folle è la loro densità ed è naturale pensare che folle di diverse densità dovrebbero ricevere diversi livelli di attenzione. Gli studi effettuati prima da Fruin [3] e poi ripresi da Polus et al. [4] ci forniscono una chiara idea del problema del *livello del servizio* per il traffico pedonale definito come: flusso libero, flusso limitato, flusso denso e flusso interrotto in accordo con una misura della densità definita come il numero di pedoni per unità di area. Weidmann inoltre dimostrò empiricamente che lo scopo dello spostamento influisce sulla velocità di moto [5] che per il traffico turistico o ricreativo stimò essere 0.99 m/s. Le stime sulla densità e sul movimento della folla ci possono aiutare a stabilire se siano in atto o meno comportamenti anormali o allarmanti [6] e a gestire situazioni potenzialmente pericolose.

Le soluzioni a LOI condividono tre step fondamentali: rilevazione del foreground per discriminare i pixel in movimento rispetto a quelli statici dello sfondo, riconoscimento di ogni singola persona nella scena e tracciamento della sua traiettoria per aumentare il contatore nel momento in cui questa interseca la linea virtuale d'interesse. Le differenze tra i vari metodi ricadono nell'approccio utilizzato per ogni step e nel tipo di dispositivo utilizzato. I primi tentativi nel conteggio delle persone sono progettati per processare i video catturati da telecamere RGB. Molti di questi metodi utilizzano combinazioni di rilevazione del foreground come frame differencing o background subtraction.



Figura 2 Frame originale



Figura 3 Estrazione foreground

Il problema principale di queste soluzioni è che presentano diverse difficoltà in scenari molto affollati e soggetti a cambiamenti di luce. Per limitare l'impatto di questi problemi, hanno cominciato a sorgere studi su come svolgere un'analisi 3D per sfruttare l'informazione riguardo la profondità. I primi metodi che seguono questa linea di pensiero erano basati sull'utilizzo di telecamere stereoscopiche anche se con l'uscita di Microsoft Kinect, che utilizza la luce strutturata, nuove opportunità si sono aperte proprio con l'utilizzo di questo device [7], [8].

In questo capitolo verrà presentata una panoramica dei dispositivi esistenti da impiegare nel monitoraggio del traffico pedonale e delle esperienze di significativa importanza in cui sono stati utilizzati.

### 3.1. Dispositivi

Esistono diversi tipi di dispositivi che vengono quotidianamente utilizzati in questo campo: dispositivi che acquisiscono immagini, laser, fotocellule e sniffer. Le prime tre categorie riescono a gestire più o meno bene l'informazione riguardo il numero di ingressi/uscite da un varco, quindi problemi di tipo LOI. Le telecamere, inoltre, possono essere impiegate anche per la stima della densità pedonale nelle immagini catturate (ROI). Gli sniffer, in particolare WiFi, si distinguono per la loro versatilità, specialmente se utilizzati in sinergia con molteplici esemplari.

#### 3.1.1. Videocamere

I metodi convenzionali si approcciano al problema della Crowd Analysis utilizzando generiche telecamere a colori o in scala di grigio e si possono dividere in tre categorie: rilevamento e tracking dei pedoni, regressione sulla base di feature e clustering delle traiettorie. I fotogrammi possono anche essere utilizzati per rilevare comportamenti anomali nella folla [6] o, grazie a metodi recenti basati su deep learning, per fornire una stima della densità di questa [9], [10], [11].

Nonostante siano stati fatti grandi progressi, i problemi menzionati sono ben lontani dall'essere risolti a causa di svariate sfide che includono cambiamenti nell'illuminazione, ombre, occlusioni e diversità nell'aspetto dei pedoni. Inoltre, molte delle soluzioni si basano su immagini ad alta risoluzione e modelli complessi che rendono difficile l'applicabilità real-time. Per migliorare le performance del conteggio e per ridurre la complessità computazionale, sono stati proposti diversi metodi che sfruttano sensori di profondità differenti (Kinect, telecamere a tempo di volo, telecamere stereoscopiche ecc) in sinergia o in sostituzione alle telecamere RGB [7], [12], [13].

Sul mercato vi sono molteplici soluzioni, relativamente a basso costo, molte delle quali sfruttano appunto telecamere stereoscopiche [14], [15].



Figura 4 Telecamera stereoscopica Hikvision IDS-2CD6810F/C

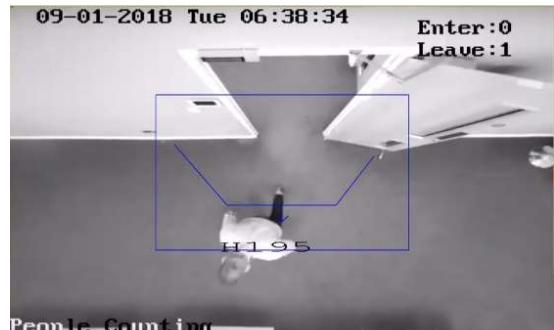


Figura 5 Frame tratto dalla telecamera a sinistra [16]

Da tenere in considerazione l'impatto estetico non indifferente che l'installazione comporta e il fatto che in generale, per grandi masse, l'accuratezza dell'analisi cala. Trattando immagini, se il software non considera il problema, possono sorgere controversie riguardo la privacy e per conteggiare il numero di ingressi/uscite da un varco vi è il vincolo del posizionamento presso la verticale dello stesso.

### 3.1.2. Infrarossi e laser

Infrarossi e laser forniscono un'alternativa efficiente ai problemi sopra menzionati. In studi passati, telecamere ad infrarossi, scanner laser o una combinazione dei due sono stati impiegati in questo campo. Grazie all'abilità di acquisire immagini tramite radiazioni di calore, le camere IR forniscono caratteristiche uniche mentre grazie agli scanner laser si possono ottenere informazioni sulla profondità. In aggiunta, con device laser, immagini ad alta risoluzione non sono necessari, riducendo di molto la complessità computazionale e rendendo le operazioni idonee ad un ambiente real-time. Wu et al. [17] mostrano un approccio innovativo basato su un laser ad infrarossi multipunto per ricavare l'informazione sulla profondità e costruire un modello spaziale dello sfondo per l'individuazione delle persone senza il bisogno di immagini ad alta risoluzione.

STMicroelectronics [18] ci dimostra come la ricerca in questo campo stia dando vita a soluzioni sempre più convenienti e minimali [19]. LZR-SIGMA, per esempio, è un dispositivo laser ad infrarossi con misura a tempo di volo [20] facilmente reperibile sul mercato europeo.



Figura 6 LZR®-SIGMA. Soluzione contapersone laser con campo di rilevazione ad alta risoluzione (250 punti per battente)



Figura 7 SIGMA possiede un sistema di conteggio affidabile che offre una misurazione accurata del flusso di adulti, bambini e gruppi di persone.

### 3.1.3. Sniffer

Considerata la notevole importanza che gli smartphone stanno a mano a mano acquisendo per via della loro capacità di accedere al web utilizzando dati mobili e reti Wi-Fi, rispecchiando talvolta in scenari distopici la figura di estensione del nostro corpo, l'analisi del traffico wireless si è rivelata un'ottima alternativa ai metodi convenzionali sopraccitati. Si definisce sniffing (eavesdropping) l'attività di intercettazione passiva dei dati che transitano in una rete. I software che sono in grado di captare i singoli pacchetti del traffico Internet prendono il nome di sniffer e quelli Wi-Fi necessitano di un modulo wireless che stia in ascolto del traffico. Laddove i metodi di videosorveglianza o quelli basati su laser e IR si applicano molto bene ad analisi di tipo LOI, facendo leva sull'attaccamento che l'uomo ha per il proprio dispositivo mobile, gli sniffer Wi-Fi trovano efficace impiego nella stima delle densità, flussi e tempo di viaggio. Recenti studi vedono impiegati molteplici sniffer che in sinergia rendono possibile la stima della posizione di un dispositivo mobile all'interno di un'area schematizzabile in celle grazie a tecniche di machine learning [21], la stima della densità di persone all'interno di un centro commerciale [22], la stima del flusso e dei percorsi pedonali [23], [24], [25] o il monitoraggio di grandi eventi come il pellegrinaggio Hajj verso la Mecca [26].

Esistono alcuni prodotti specializzati [27] ma la maggior parte degli scienziati, che in questi anni hanno cercato di sfruttare questo metodo di monitoraggio emergente, si affida ancora a soluzioni "casalinghe", spesso costituite da una o più stazioni composte da un single-board computer (come Raspberry Pi [28]) con un modulo Wi-Fi usb esterno. Il motivo di questa scelta risiede nell'installazione di un sistema che, a differenza di quelli di videosorveglianza, risulta economico, semplice e senza la necessità di hardware aggiuntivo.

L'ottenimento di informazioni tramite lo sniffing Wi-Fi di per sé non è un processo complicato anche perché una delle caratteristiche delle reti wireless è proprio quella di essere broadcast. I dispositivi con il modulo Wi-Fi acceso, infatti, sono regolarmente in cerca di connettività. Mandano continuamente richieste per cercare reti a cui hanno già effettuato un accesso in passato. Queste richieste contengono tra le altre cose, informazioni che consentono l'identificazione del dispositivo inviante. Lo scopo però che in questo caso si vuole raggiungere tramite lo sniffing, ovvero la Crowd Analysis, comporta uno studio non banale, dovuto a varie ragioni. In primo luogo, i dati ottenuti sono quelli di un sottoinsieme del traffico reale, in cui il penetration rate dei dispositivi con Wi-Fi attivato gioca un ruolo fondamentale. In secondo luogo, la rilevazione di un dispositivo dipende in primis da esso stesso poiché, nonostante il corretto funzionamento del Wi-Fi, può risultare comunque momentaneamente invisibile. Infine, a seconda dei casi, i dispositivi potrebbero non lasciare un'impronta univoca e quindi il conteggio andrebbe a considerare dei doppiioni o peggio triple, quadruple e via dicendo copie come entità separate.

Nei capitoli 4 5 e 6 si approfondirà il caso di studio, con maggiori informazioni riguardo il MAC address e la sua randomizzazione e con la spiegazione di come sia stato sviluppato il sistema. Nel capitolo 7 verrà fornita una spiegazione tecnica di come sia possibile eseguire lo sniffing dei dispositivi mobili.

## 3.2. Progetti di monitoraggio

Oggiorno, molti eventi di larga scala sono ospitati in aree urbane che non sono progettate per ricevere questo grande afflusso di spettatori. Oltre al numero di visitatori fuori misura, il comportamento di queste folle è spesso imprevedibile e non contemplato. Mai come in questo periodo sono necessari sistemi di monitoraggio delle folle che forniscano informazioni real-time riguardo lo stato del traffico pedonale sull'area dell'evento per garantire la sicurezza dei visitatori prevenendo situazioni di pericolo e colli di bottiglia o identificandole tempestivamente.

Di seguito sono riportati alcuni tentativi di monitoraggio effettuati in passato durante eventi di larga scala i quali hanno utilizzato più tecnologie contemporaneamente per avere una visione il più completa possibile della situazione.

### 3.2.1. SAIL Amsterdam (2015)

SAIL è un evento marittimo quinquennale che si tiene ad Amsterdam, in Olanda [29]. Ha il vanto di essere una delle più grandi (se non la più grande) manifestazione nautica al mondo e senz'altro il più grande evento pubblico Olandese. Ogni cinque anni più di 600 navi tra velieri, fregate, natanti, navi moderne, militari e repliche di imbarcazioni storiche navigano lungo il canale del mare del Nord per poi ormeggiare al porto di IJhaven ad Amsterdam. Il legame tra Amsterdam e il mare è profondo e SAIL non è solo un'esibizione di grandi navi. In poche parole, comprende "qualsiasi cosa che galleggi" e durante l'edizione



Figura 8 La Amerigo Vespucci italiana

del 2000 si raggiunse l'impressionante numero di 8000 imbarcazioni. Accanto alle attività in acqua ci sono molte attrazioni anche a terra come musica, arte, cultura e tante attività per bambini.

SAIL fu organizzato per la prima volta nel 1975 e visto il grande successo, si è deciso di ripetere l'evento in forma quinquennale, riunendo nelle recenti edizioni circa 2 milioni di persone sul territorio.

Durante SAIL 2015, una combinazione di sistemi di conteggio video, tracciatori GPS e sensori Wi-Fi fu impiegata per determinare lo stato del traffico pedonale [30], [31]. I dati vennero raccolti utilizzando 8 telecamere per conteggio, 20 sensori Wi-Fi e 324 tracciatori GPS furono distribuiti ai visitatori per registrare le loro traiettorie. Il problema fondamentale dello studio però fu che non c'era una vera e propria conoscenza assoluta (in termini tecnici ground-truth) del traffico reale a cui fare riferimento, ma i calcoli vennero comparati alle immagini registrate da altre telecamere aeree. Le 8 telecamere ebbero il compito di calcolare il numero di pedoni che attraversano una sezione trasversale. In teoria, i dispositivi impiegati avevano un'accuratezza del 98% decrescente a 92% in situazioni di alta densità ma nessuna considerazione su velocità o percorsi può essere fatta con questi dati.

I sensori Wi-Fi invece ebbero lo scopo di derivare i flussi e il tempo di percorrenza, acquisendo informazioni riguardo i pedoni dotati di un qualsiasi dispositivo mobile (es. smartphone) all'interno del loro raggio d'azione. I dati contenenti i gli indirizzi MAC dei dispositivi, l'ID del sensore e il primo e ultimo timestamp di rilevamento di ogni dispositivo rendono possibile il calcolo cumulativo dei singoli dispositivi e dei tempi di attraversamento da sensore a sensore (supponendo che ad un dispositivo corrispondesse un pedone).

Confrontando i dati ottenuti dai sensori Wi-Fi e dalle telecamere, si nota che i sensori percepirono circa un terzo dei pedoni conteggiati dalle telecamere e la metà di questi (quindi 1/6) corrispondono a dispositivi unici. Nonostante un penetration rate così basso, i calcoli eseguiti su approssimativamente 2.3 milioni di visitatori sono sufficienti per avere un'idea dei percorsi effettuati e sui tempi di tragitto per alcune tratte. Per impiegare i sensori Wi-Fi nella stima della densità tuttavia, sono necessarie stime accurate sul penetration rate (rapporto tra pedoni totali e numero di dispositivi mobili rilevati) in modo da poter stimare in maniera più accurata il traffico reale.

Nello studio citato vengono illustrati nel dettaglio 3 algoritmi di stima dello stato per stimare travel time, densità e flussi [31].

### 3.2.2. Pellegrinaggio Hajj (2016)

Hajj è il tradizionale pellegrinaggio annuale verso la Mecca. Essendo il quinto pilastro dell'Islam e momento di purificazione per i fedeli, l'Hajj è considerato il pellegrinaggio più grande al mondo in cui si riuniscono 2-3 milioni di musulmani provenienti da tutto il globo. L'Hajj è una sequenza di riti praticati in tempi e luoghi prestabiliti e questi vincoli spazio-temporali rendono la gestione dell'Hajj un processo complesso. È con lo studio di come si comportano i pellegrini, dei loro pattern, delle interazioni, delle necessità e domande che si può fornire un livello di servizio soddisfacente. Con questo pensiero, A. Basalamah [26] installò un sistema composto da 8 sniffer Wi-Fi autoalimentati a energia solare in una delle aree destinate ad ospitare i pellegrini per un giorno, Arafat, 12 km a sud est della Sacra Città della Mecca. Arafat è un'area deserta per tutto il resto dell'anno in cui durante il pellegrinaggio vengono allestite delle tende per i fedeli e nel 2016 furono stimate 185.000 persone. La collezione dei dati cominciò due giorni prima del picco stimato e continuò per i successivi tre giorni. I risultati ottenuti mostrarono che il sistema fu in grado di rilevare circa il 37.5% della folla, corrispondente a 69.467 dispositivi unici. Dalla rete di sensori, Basalamah fu in grado di identificare le aree più affollate, stabilire gli orari dei picchi di

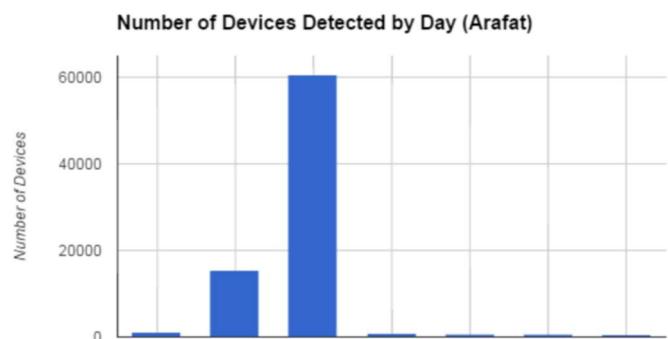


Figura 9 Dispositivi unici per giorno

mobilità, estrapolare informazione riguardo lo stato economico dei pellegrini avendo a disposizione le marche delle case produttrici dei dispositivi e diverse altre statistiche.

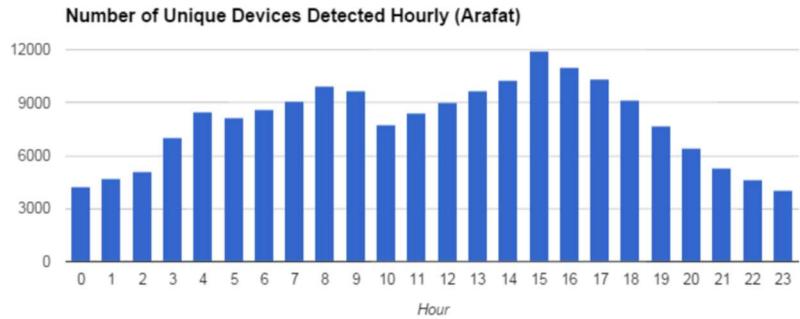


Figura 10 Dispositivi unici per ora (giorno di picco)

### 3.2.3. Il caso Venezia (2018)

VENIS - Venezia Informatica e Sistemi S.p.A. è l'azienda di servizi ICT e l'operatore locale di comunicazioni elettroniche del Comune di Venezia. Nei primi mesi del 2018, VENIS ha aperto un bando rivolto a soggetti privati ad offrirsi per una sperimentazione gratuita di impianti atti al conteggio dei flussi pedonali nelle aree limitrofe alla Stazione Santa Lucia ed in Piazza San Marco. Le sperimentazioni hanno visto come oggetto l'installazione gratuita e il test di impianti tecnologici e/o sensoristica e/o videorilevazione nelle aree del Centro Storico di Venezia. In sei mesi, i prodotti di sei ditte sono stati messi a confronto e per ognuno è stato stilato un resoconto riguardo tipologia del sensore, attendibilità, flessibilità, installazione, estetica e disponibilità dei dati. I dispositivi presi in analisi si riferiscono a laser, telecamere stereoscopiche, sniffer Wi-Fi, telecamere ad infrarossi e telecamere con annesso software per il riconoscimento di transiti. Di seguito viene riportata una tabella riassuntiva, ma è disponibile alla lettura il report completo [32].

Ditta	Sensore	Attendibilità	Flessibilità	Installazione	Estetica	Disponibilità dati
Hitachi	3D ToF	Molto buona	Portata limitata	Semplice	Ridotte dimensioni ma richiede mensola di protezione e box apparati	Real-time
Nicoli	Barriera LASER	non valutata	Portata elevata	Va installata in quota, installazione semplice	Dimensioni generose ma installata ad altezza elevata non si nota, richiede box apparati	Real-time
Nicoli	Telecamera stereoscopica	Molto buona	Buona	Semplice	Poco impattante, richiede box apparati	Real-time
Nicoli	Sniffer Wi-Fi	non valutata	Elevata	Semplice	Poco impattante (può essere anche nascosto)	Anche real time, in funzione della informazione richiesta
ISCOM	Telecamera	Molto buona	Elevata	Semplice	Poco impattante	Anche real-time (con precisione ridotta)
AXIANS	Telecamera stereoscopica	Eccellente	Elevata	Semplice	Poco impattante, richiede box apparati	Real-time
ZIBONI	Sensore PIR a batteria	Buona, non adatto a situazioni di congestione	Molto Elevata	Semplice	Poco impattante	Differita di un giorno
ZIBONI	Telecamera IR	Buona	Portata limitata	Semplice	Piccolo ma evidente, richiede box apparati	Real-time
Telecom Italia	Telecamera (transiti)	Molto buona	Elevata	Semplice	Poco impattante, richiede box apparati	Real-time
Telecom Italia	Telecamera (presenze)	non valutata	Elevata	Semplice	Poco impattante, Richiede box apparati	Real-time

Figura 11 Sintesi dei risultati

### 3.3. Tecnologia preesistente in loco

## 4. Caso di studio: Wi-Fi

Lo standard IEEE 802.11 definisce tre tipi di frame che vengono scambiati dai dispositivi Wi-Fi: Management frames, Control frames e Data frames. I sistemi sniffer in esame fanno leva sulle informazioni contenute nei Management frames per cui gli altri due tipi di non verranno menzionati oltre. Per quanto riguarda i Management frames, esistono quattro principali sottotipi: Beacons, Probes, Association e Authentication a loro volta divisi in ulteriori classi.

I Beacon frames (sottotipo 0x08) sono messaggi che gli AP (access point) inviano periodicamente in modalità broadcast per segnalare ai dispositivi nei dintorni la propria esistenza e il proprio SSID. Le Probe requests (sottotipo 0x04) sono dei frame speciali inviati da una stazione client che richiede informazioni da un particolare AP specificato da un SSID o da tutti quelli nei dintorni con un SSID broadcast. L'invio di una Probe request avviene quando la scheda wireless del dispositivo sta eseguendo il cosiddetto active scanning, cioè la ricerca attiva di AP, mentre il semplice ascolto dei frame Beacon è denominato passive scanning. In seguito ad una Probe request, gli AP emettono una Probe response (sottotipo 0x05) contenente le informazioni necessarie a stabilire una connessione.

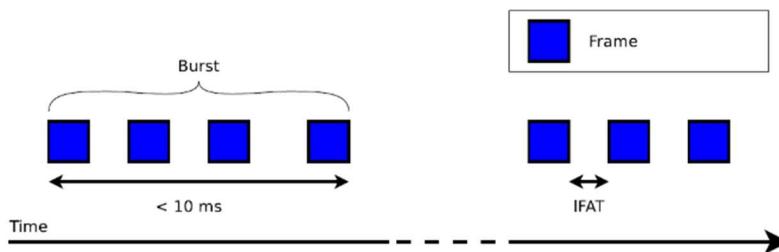


Figura 12 Schema di una sequenza di trasmissione. Un burst è un gruppo di frame di tipo probe request inviati da un device nell'arco di 10ms. L'inter-frame arrival time (IFAT) indica il lasso di tempo tra due frame. [33]

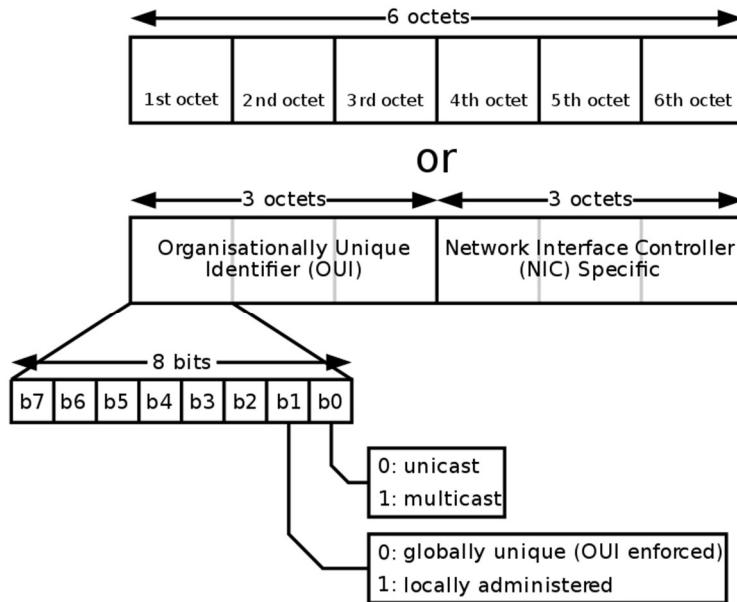
Ogni dispositivo elettronico porta con sé un'informazione univoca, un indirizzo MAC (media access control) che può essere intercettato quando esso tenta di collegarsi ad una rete wireless o il collegamento è già stato effettuato perché è contenuto in chiaro nelle Probe requests. È quindi mediante lo sniffing di queste richieste, durante la fase di active scanning, che si possono collezionare informazioni riguardo i client (nel nostro caso gli smartphone dei turisti) come indirizzo MAC, potenza del segnale (RSSI), timestamp, sequence number della richiesta e svariati altri dati. Come intuibile, le Probe requests sono più frequenti quando il dispositivo non è connesso a nessuna rete e la loro frequenza diminuisce drasticamente quando esso si connette.

In risposta alla possibilità di ottenere facilmente informazioni dalle Probe requests, molti dei sistemi operativi per smartphone stanno via via implementando sistemi di randomizzazione dell'indirizzo MAC in modo da proteggere la privacy degli utenti e impedire il tracking dei loro spostamenti. In particolare, come constatato empiricamente in un precedente progetto del sottoscritto [34] che conferma quanto trovato in [35], i dispositivi iOS sono quelli che effettuano la randomizzazione in modo più ostico per quello che riguarda l'attribuzione di indirizzi randomici allo stesso dispositivo.

### 4.1. MAC Address

L'indirizzo MAC (Media Access Control) è un codice di 48 bit (6 byte) assegnato in modo univoco dal produttore ad ogni scheda di rete ethernet o wireless prodotta al mondo, tuttavia

modificabile a livello software. Al fine di garantire l'univocità dell'indirizzo MAC tra i dispositivi, l'IEEE assegna blocchi di indirizzi alle organizzazioni in cambio di una tassa. Questi blocchi, conosciuti più comunemente come OUI (Organizationally Unique Identifier), possono essere comprati e registrati tramite l'IEEE che da all'organizzazione il controllo e la responsabilità per tutti gli indirizzi con un dato prefisso a tre byte. I produttori a questo punto sono liberi di assegnare i rimanenti tre byte a qualsiasi valore essi vogliano con l'unica accortezza di non utilizzare lo stesso indirizzo due volte.



*Figura 13 Diagramma che mostra la struttura di un indirizzo MAC-48, con indicate esplicitamente le posizioni del multicast/unicast bit e OUI/local bit*

Oltre all'indirizzo MAC globale del produttore e di dominio pubblico, i dispositivi moderni utilizzano frequentemente un indirizzo localmente assegnato che è contraddistinto da un local bit nel byte più significativo dell'indirizzo. Gli indirizzi localmente assegnati non danno la garanzia di univocità e generalmente non sono utilizzati in una maniera persistente. Questi infatti, possono essere utilizzati per generare indirizzi MAC randomici come misura aggiuntiva per la privacy. Poiché in fase di active scanning i dispositivi comunicano la loro presenza agli AP nelle vicinanze e le Probe request richiedono l'indirizzo MAC della sorgente, se un device utilizza il suo indirizzo globale allora sta effettivamente facendo broadcast della sua identità. Analogamente agli OUI, i produttori possono acquistare da IEEE dei prefissi CID (Company Identifier) con la promessa di non utilizzarli in forma globale per i dispositivi. Di conseguenza, i CID hanno sempre il local bit asserito. Un esempio di questo tipo è dato dal prefisso DA:A1:19 posseduto da Google. Con l'impiego di MAC address randomizzati che cambiano nel corso del tempo, il tracking dei dispositivi risulta non più banale. Nel momento in cui un device però si connette ad un AP, smette di randomizzare ed utilizza il suo MAC reale. Per contro, come già accennato, l'evento che ci interessa sniffare, ovvero la fase di active scanning, si verifica con frequenza nettamente inferiore.

## 4.2. Randomizzazione

La randomizzazione dell'indirizzo MAC, a cui da questo momento in poi ci riferiremo semplicemente con "randomizzazione", è una comune misura di protezione per la privacy applicata da buona parte dei sistemi operativi di uso quotidiano. Viene impiegata per prevenire il tracking degli utenti tramite le probe requests che sono trasmesse durante le scansioni network IEEE 802.11.

Questo significa che, per esempio, può essere utilizzato un nuovo MAC address per ogni scansione, dove una scansione consiste nell'invio di probe requests su tutti i canali utilizzabili. Tuttavia, siccome non esiste ancora uno standard sulla randomizzazione, iOS, Android, Windows e Linux implementano le proprie varianti. Questo fa porre la domanda circa la sicurezza che la randomizzazione dovrebbe garantire.

Come precedentemente accennato, la randomizzazione si ferma una volta che un dispositivo viene associato con un AP, per questo nella letteratura si trovano diversi attacchi il cui scopo è risalire al MAC univoco a partire da quello randomico. Un esempio è il cosiddetto Karma attack in cui si costruisce un fake AP il cui SSID corrisponde a uno di quelli ricercati dal dispositivo di cui si vuole risalire all'indirizzo globale, costringendo così il dispositivo a comunicare il suo vero indirizzo [35], [36].

Sia perché con le recenti implementazioni di randomizzazione i dispositivi trasmettono più raramente i propri SSID conosciuti, riducendo l'efficacia del Karma attack, sia perché questo lavoro vuole essere del tutto passivo, senza violazioni di privacy o collezione di dati sensibili, non verranno usati attacchi sui singoli dispositivi ma verrà studiato un modo per aggregare quanto più possibile i MAC address randomici sotto uno stesso dispositivo.

In nessun momento si è tentato di decifrare qualsivoglia dato o di intraprendere azioni attive per modificare il comportamento della rete e soprattutto non abbiamo mai avuto modo di collegare indirizzi MAC alle identità degli individui possessori dei dispositivi.

### 4.2.1. Android

I sistemi Android cominciarono ad impiegare la randomizzazione con la versione 6.0 veniva effettuata durante le scansioni in background se l'hardware la supportava. Fino all'uscita di Android 8.0 però questa era utilizzata da una piccolissima percentuale di dispositivi. Google acquistò poi il proprio CID DA:A1:19 utilizzato tutt'ora dalla stragrande maggioranza dei dispositivi Android che randomizzano, lasciando ai software la randomizzazione dei restanti 3 byte dell'indirizzo MAC ( $2^{24}$ ). Con la versione 9.0 del robottino verde l'opzione aggiuntiva si fece ancora più accessibile comparendo direttamente tra le impostazioni sviluppatore dei dispositivi ma di default non attiva. Nonostante Google abbia gettato tutte le basi per la randomizzazione sui dispositivi, e che praticamente ogni altro cellulare all'infuori degli iPhone monta una versione di Android, questo non è il sistema operativo in cui si nota la maggior quantità di randomizzazione. Per problemi di compatibilità tra software e hardware o per decisione dei produttori stessi che customizzano le versioni del sistema operativo di Google, la maggior parte della randomizzazione arriva da Apple che produce sia hardware che software. Secondo tutti gli studi svolti fino ad oggi, per esempio,

non si è mai ancora registrato un dispositivo Samsung che effettuasse la randomizzazione, nonostante questo sia il maggior produttore di dispositivi Android sul mercato.

Un esempio di randomizzazione da parte di Android è mostrato dalla seguente tabella, ottenuta da un esperimento in un ambiente quanto più controllato possibile. Con il tool Wireshark mi sono messo in ascolto dei pacchetti wireless in transito con l'intenzione di filtrare le probe request inviate da uno Xiaomi Redmi Note 7 in mio possesso, non associato ad alcuna rete. Il dispositivo monta la versione 9.1 di Android e come si può notare, in accordo con quanto descritto precedentemente, Wireshark automaticamente sostituisce il CID “DA:A1:A19” con il tag “Google\_” mostrando i 3 byte finali dell’indirizzo su cui il sistema operativo effettua la randomizzazione.

Time (s)	Source	Destination	Protocol	Lenght	Info
0.000000000	Google_0e:fe:86	Broadcast	802.11	153	Probe Request, SN=1246
0.020056565	Google_0e:fe:86	Broadcast	802.11	153	Probe Request, SN=1247
20.478772870	Google_16:71:b9	Broadcast	802.11	153	Probe Request, SN=1272
20.519636137	Google_16:71:b9	Broadcast	802.11	153	Probe Request, SN=1275
40.607593376	Google_fc:02:ca	Broadcast	802.11	153	Probe Request, SN=1305
40.627627762	Google_fc:02:ca	Broadcast	802.11	153	Probe Request, SN=1306
60.529810066	Google_00:b4:8a	Broadcast	802.11	153	Probe Request, SN=1333
60.550520580	Google_00:b4:8a	Broadcast	802.11	153	Probe Request, SN=1334
80.558958993	Google_a9:08:4f	Broadcast	802.11	153	Probe Request, SN=1362
80.578992854	Google_a9:08:4f	Broadcast	802.11	153	Probe Request, SN=1363
100.575174121	Google_c8:d7:94	Broadcast	802.11	153	Probe Request, SN=1393
100.597433718	Google_c8:d7:94	Broadcast	802.11	153	Probe Request, SN=1394
126.177714102	Google_62:54:e4	Broadcast	802.11	153	Probe Request, SN=1416
126.197759739	Google_62:54:e4	Broadcast	802.11	153	Probe Request, SN=1417
153.825444973	Google_e7:d1:31	Broadcast	802.11	153	Probe Request, SN=1441
153.996834953	Google_e7:d1:31	Broadcast	802.11	153	Probe Request, SN=1442
157.493772620	Google_e2:48:82	Broadcast	802.11	153	Probe Request, SN=1466
157.513863436	Google_e2:48:82	Broadcast	802.11	153	Probe Request, SN=1467
177.401985590	Google_fb:a2:0c	Broadcast	802.11	153	Probe Request, SN=1489
177.449180018	Google_fb:a2:0c	Broadcast	802.11	153	Probe Request, SN=1492
201.581892344	Google_27:46:70	Broadcast	802.11	153	Probe Request, SN=1520
201.603682772	Google_27:46:70	Broadcast	802.11	153	Probe Request, SN=1521
221.572785571	Google_c8:88:a9	Broadcast	802.11	153	Probe Request, SN=1548
221.592971051	Google_c8:88:a9	Broadcast	802.11	153	Probe Request, SN=1549
243.207523855	Google_e5:43:1c	Broadcast	802.11	153	Probe Request, SN=1576
243.227686911	Google_e5:43:1c	Broadcast	802.11	153	Probe Request, SN=1577
266.572511158	Google_0e:dc:d6	Broadcast	802.11	153	Probe Request, SN=1603
266.592636835	Google_0e:dc:d6	Broadcast	802.11	153	Probe Request, SN=1604
286.593584674	Google_71:26:8e	Broadcast	802.11	153	Probe Request, SN=1631
286.613684433	Google_71:26:8e	Broadcast	802.11	153	Probe Request, SN=1632

Tabella 1 Randomizzazione Xiaomi Redmi Note 7

La tabella riporta cinque minuti (trecento secondi) di ascolto durante i quali il dispositivo emette 30 probe request sul canale di ascolto, corrispondenti a 15 indirizzi MAC differenti. Si nota subito che i burst intercettabili su un canale sono composti da due richieste e che l’indirizzo cambia ad ogni burst.

Intervallo medio di randomizzazione (inter-burst)	20.625 s
Minimo intervallo inter-burst	3.7 s
Massimo intervallo inter-burst	33.9 s
Minimo Sequence number intra-burst	1
Minimo Sequence number inter-burst	22
Massimo Sequence number intra-burst	3
Massimo Sequence number inter-burst	30

La lunghezza dei frame in ogni caso è sempre la stessa, 153 bytes, caratteristica su cui si basa il meccanismo di fingerprinting descritto nel capitolo 5, mentre gli SSID conosciuti non compaiono mai. Il destinatario di ogni richiesta è sempre ff:ff:ff:ff:ff:ff (broadcast) ovvero qualsiasi AP in ascolto.

#### 4.2.2. Apple

Apple introdusse la randomizzazione dell'indirizzo MAC a partire da iOS 8 ma non fu un grande successo. Essa si manifestava a patto che fossero soddisfcesse tre condizioni: il wifi fosse attivo ma non connesso ad alcuna rete, il dispositivo fosse in "sleep mode" (schermo spento per inattività) e i servizi di localizzazione fossero disattivati. Neanche a dirlo quasi nessuno avrebbe mantenuto il dispositivo in questo stato. La maggior parte delle app non funziona nemmeno se i servizi di localizzazione non sono attivati ed era del tutto inutile se la randomizzazione sarebbe terminata con l'accensione del display. Da iOS 9, Apple estese questa funzionalità a quello che definisce come locations and auto-join scans [x] che garantisce che la randomizzazione venga effettuata anche a schermo acceso. Al contrario di Google che ha acquistato il proprio CID, gli indirizzi Randomizzati di Apple sembrano non avere un prefisso predefinito e la randomizzazione avviene su tutti i bit ad eccezione del local bit settato a 1 e del multicast bit a 0. Ne risulta una randomizzazione su  $2^{46}$  bit.

Come per Android è disponibile un esempio di randomizzazione iOS registrato da un iPhone 6 con iOS 9. in mio possesso, nelle medesime condizioni del precedente esperimento.

Time (s)	Source	Destination	Protocol	Length	Info
0.000000000	1e:69:da:39:e9:7a	Broadcast	802.11	176	Request, SN=912
0.003260863	1e:69:da:39:e9:7a	Broadcast	802.11	176	Request, SN=913
3.229036747	ae:ef:ff:f5:f4:b6	Broadcast	802.11	176	Request, SN=949
3.249793790	ae:ef:ff:f5:f4:b6	Broadcast	802.11	176	Request, SN=950
17.124327398	d6:5c:53:c5:ce:c8	Broadcast	802.11	176	Request, SN=986
17.144741222	d6:5c:53:c5:ce:c8	Broadcast	802.11	176	Request, SN=987
20.062601984	06:2b:1d:2d:d8:f9	Broadcast	802.11	176	Request, SN=1023
20.082064693	06:2b:1d:2d:d8:f9	Broadcast	802.11	176	Request, SN=1024
26.086563110	b6:45:d3:ef:c3:7a	Broadcast	802.11	176	Request, SN=1062
26.107303803	b6:45:d3:ef:c3:7a	Broadcast	802.11	176	Request, SN=1063
30.028257945	8a:29:9b:41:79:f2	Broadcast	802.11	176	Request, SN=1097
30.048418686	8a:29:9b:41:79:f2	Broadcast	802.11	176	Request, SN=1098
38.093549750	92:c9:45:16:67:a1	Broadcast	802.11	176	Request, SN=1134
38.114193199	92:c9:45:16:67:a1	Broadcast	802.11	176	Request, SN=1135
51.215853754	be:20:90:e8:0f:6b	Broadcast	802.11	176	Request, SN=1170
51.217149709	be:20:90:e8:0f:6b	Broadcast	802.11	176	Request, SN=1171

61.820964072	9e:e9:d2:62:c4:b5	Broadcast	802.11	176	Request, SN=1207
61.841800507	9e:e9:d2:62:c4:b5	Broadcast	802.11	176	Request, SN=1208
66.156033562	d2:32:e7:09:15:0d	Broadcast	802.11	176	Request, SN=1246
66.176003065	d2:32:e7:09:15:0d	Broadcast	802.11	176	Request, SN=1247
69.123112913	e2:16:2c:ec:3e:77	Broadcast	802.11	176	Request, SN=1281
69.149998832	e2:16:2c:ec:3e:77	Broadcast	802.11	176	Request, SN=1282
71.975087397	8a:62:b6:21:9b:c7	Broadcast	802.11	176	Request, SN=1318
72.003422778	8a:62:b6:21:9b:c7	Broadcast	802.11	176	Request, SN=1319
80.086840657	4a:5e:15:5a:7a:6e	Broadcast	802.11	176	Request, SN=1357
80.107596215	4a:5e:15:5a:7a:6e	Broadcast	802.11	176	Request, SN=1358
84.106216095	a2:4d:3a:85:67:2c	Broadcast	802.11	176	Request, SN=1392
84.126922600	a2:4d:3a:85:67:2c	Broadcast	802.11	176	Request, SN=1393
92.158398074	42:b8:63:4c:33:bb	Broadcast	802.11	176	Request, SN=1429
92.176315718	42:b8:63:4c:33:bb	Broadcast	802.11	176	Request, SN=1430
105.246452275	b6:67:97:0a:d6:bf	Broadcast	802.11	176	Request, SN=1466
105.266143503	b6:67:97:0a:d6:bf	Broadcast	802.11	176	Request, SN=1467
123.258235152	4a:9a:8d:19:95:29	Broadcast	802.11	176	Request, SN=1503
123.278961637	4a:9a:8d:19:95:29	Broadcast	802.11	176	Request, SN=1504
141.275726553	b6:70:f1:80:7a:44	Broadcast	802.11	176	Request, SN=1540
141.296125659	b6:70:f1:80:7a:44	Broadcast	802.11	176	Request, SN=1541
174.300246061	9a:00:33:27:84:5e	Broadcast	802.11	176	Request, SN=1583
174.320940406	9a:00:33:27:84:5e	Broadcast	802.11	176	Request, SN=1584
178.774824196	46:60:42:eb:93:ed	Broadcast	802.11	176	Request, SN=1622
178.794451431	46:60:42:eb:93:ed	Broadcast	802.11	176	Request, SN=1623
182.868590586	2e:68:86:13:f4:af	Broadcast	802.11	176	Request, SN=1657
182.888304434	2e:68:86:13:f4:af	Broadcast	802.11	176	Request, SN=1658
187.339999041	4a:62:68:7d:cb:b4	Broadcast	802.11	176	Request, SN=1693
187.361313909	4a:62:68:7d:cb:b4	Broadcast	802.11	176	Request, SN=1694
195.365936991	de:d6:a1:d6:cf:e7	Broadcast	802.11	176	Request, SN=1730
195.386577098	de:d6:a1:d6:cf:e7	Broadcast	802.11	176	Request, SN=1731
208.365194904	42:6c:41:97:08:6e	Broadcast	802.11	176	Request, SN=1767
208.385796120	42:6c:41:97:08:6e	Broadcast	802.11	176	Request, SN=1768
220.761622918	7e:18:0d:89:00:98	Broadcast	802.11	176	Request, SN=1804
220.782187109	7e:18:0d:89:00:98	Broadcast	802.11	176	Request, SN=1805
226.370413892	3a:09:8f:26:a0:de	Broadcast	802.11	176	Request, SN=1841
226.391156374	3a:09:8f:26:a0:de	Broadcast	802.11	176	Request, SN=1842
240.587248682	4a:0e:f2:f6:d3:10	Broadcast	802.11	176	Request, SN=1880
240.606907671	4a:0e:f2:f6:d3:10	Broadcast	802.11	176	Request, SN=1881
244.577034944	7e:6c:c6:72:25:bf	Broadcast	802.11	176	Request, SN=1915
244.597226528	7e:6c:c6:72:25:bf	Broadcast	802.11	176	Request, SN=1916
252.612058797	72:25:53:5c:7a:36	Broadcast	802.11	176	Request, SN=1952
252.631753251	72:25:53:5c:7a:36	Broadcast	802.11	176	Request, SN=1953
265.605307175	6a:bf:00:70:18:f1	Broadcast	802.11	176	Request, SN=1989
265.625013224	6a:bf:00:70:18:f1	Broadcast	802.11	176	Request, SN=1990
283.639501897	9e:6c:30:79:09:a2	Broadcast	802.11	176	Request, SN=2032
283.659220793	9e:6c:30:79:09:a2	Broadcast	802.11	176	Request, SN=2033
288.725051371	1e:04:51:b8:4f:9d	Broadcast	802.11	176	Request, SN=2071
288.745717831	1e:04:51:b8:4f:9d	Broadcast	802.11	176	Request, SN=2072

La prima cosa che salta all'occhio è sicuramente la maggior quantità di record registrati nello stesso arco di tempo, 32 per la precisione. La seconda sostanziale differenza riguarda gli indirizzi randomici stessi. A differenza di Google, Apple non ha un CID proprio, di conseguenza l'indirizzo viene generato casualmente in tutta la sua interezza (ad eccezione del local bit e del multicast bit).

Analogamente al caso precedente, iOS non rivela i propri SSID conosciuti, le lunghezze dei frame rimangono invariate e l'indirizzo viene randomizzato ad ogni burst.

Intervallo medio di randomizzazione (inter-burst)	9.4 s
Minimo intervallo inter-burst	3.22 s
Massimo intervallo inter-burst	33 s
Minimo Sequence number intra-burst	1
Minimo Sequence number inter-burst	34
Massimo Sequence number intra-burst	1
Massimo Sequence number inter-burst	40

#### 4.2.3. Windows

Microsoft supporta la randomizzazione a partire da Windows 10. È possibile attivarla direttamente dalle impostazioni a patto che l'hardware e i driver lo consentano. Quello che è interessante della randomizzazione di Microsoft è che utilizza MAC address randomici anche quando il dispositivo si sta connettendo ad una rete wireless, al contrario di Android ed Apple che randomizzano solamente quando effettuano la scansione. Windows è in grado di raggiungere questo obiettivo perchè utilizza sempre lo stesso MAC randomico ogni volta che si connette ad una rete specifica. Per assicurare che il client venga sempre riconosciuto, l'indirizzo per rete viene calcolato nel seguente modo:

```
address = SHA-256(SSID, macaddr, connectionId, secret) [x]
```

*Equazione 1 Creazione indirizzo randomico per rete su Windows*

dove *SSID* è ovviamente il nome della rete, *macaddr* indica il vero indirizzo MAC del dispositivo, *connectionId* è un parametro che cambia ogni qualvolta l'utente dimentica la rete e *secret* è un parametro crittografico a 256 bit, unico per interfaccia che viene generato durante l'inizializzazione del sistema e mantenuto uguale ad ogni reboot.

#### 4.2.4. Linux

Anche Linux supporta la randomizzazione che è stata introdotta nella versione del kernel 3.18. A seconda della versione del firmware il MAC cambia più o meno rapidamente durante le scansioni. La durata di default per un MAC randomico in *wpa\_supplicant* è di 60 secondi.

## 5. Prima fase del progetto

Un primo prototipo di sniffer Wi-Fi è stato realizzato durante l'anno accademico 2017/2018 dal sottoscritto con la collaborazione dei due colleghi Chiara Leoni e Ulisse Quartucci. Il progetto prevedeva l'ammodernamento di uno script Python preesistente che permettesse lo sniffing del traffico Wi-Fi. Il software era pensato per girare su una stazione indipendente composta da un Raspberry Pi dotato di modulo Wi-Fi esterno per lo sniffing, che rendesse disponibili i risultati su un database.

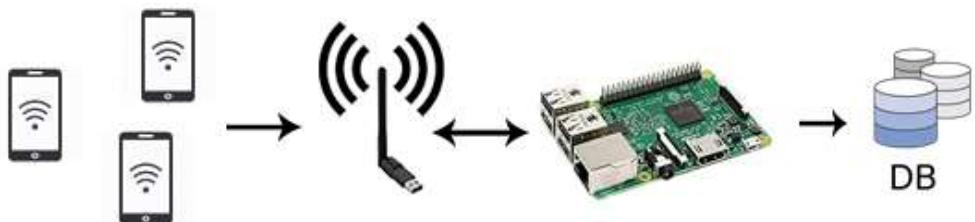


Figura 14 Schema operativo dello sniffer Wi-Fi

Basandoci su test sul campo e su [35], [36] e [37] abbiamo stabilito un nostro criterio il cui scopo è quello di attribuire ad un unico dispositivo i MAC address randomici che questo può aver generato. La prima fase ha visto la sua conclusione con l'ottenimento ad intervalli regolari del numero di MAC address unici relativi a quegli istanti di tempo e quindi il numero di dispositivi con il modulo Wi-Fi acceso e tracciabili nel raggio d'azione dell'antenna wireless.

### 5.1. Fingerprinting

L'idea di base dello script è quella di dividere le probe request in gruppi che stiano a identificare le diverse marche e modelli di dispositivi. Per quanto riguarda i MAC globali sniffati non vi è alcun problema, ma i MAC randomici devono essere aggregati. Tramite la libreria scapy, il sistema si mette in ascolto dei pacchetti in transito e recupera le informazioni in essi contenute al fine di costruire una *signature* che rappresenta la chiave che distingue i gruppi. A questa viene poi associato il MAC address contenuto nel pacchetto da cui era stata generata. La signature può essere quindi considerata come una chiave derivata dalle caratteristiche fisiche di ciascun dispositivo [36], [37], le quali vengono trasmesse all'interno del pacchetto. Questo comporta che eguali dispositivi abbiano signature coincidenti.

Come precedentemente studiato in [36], le probe request includono nel loro corpo dati sotto forma di *Information Elements* (IE) anche chiamati *parametri taggati* o più semplicemente *tags*. Questi IE non sono obbligatori ma vengono inviati dai dispositivi per segnalare agli AP le proprie caratteristiche fisiche. Ad ogni IE corrisponde un ID e un campo per i valori. Siccome sono opzionali, non tutti gli IE sono inclusi da tutti i dispositivi e di conseguenza il set di IE trasmessi varia da device a device. Sulla base della bibliografia, che ci ha aiutato a capire quali fossero gli elementi più significativi e discriminanti, sono stati oggetto di nostre attenzioni i seguenti tag:

Information Element	ID
Service Set Identity (SSID)	0
Supported Rates	1
Power Capabilities	33
HT Capabilities	45
Extended Supported Rates	50
Extended Capabilities	127
VHT Capabilities	191
Vendor Id	221

Nonostante lo standard 802.11, che definisce che i tag devono essere dichiarati in ordine crescente, molti dispositivi ignorano quanto stabilito e li trasmettono con un ordine scelto a loro discrezione. Per questo motivo anche l'ordine in cui compaiono gli IE può essere utilizzato come potenziale fonte di informazione.

## 5.2. Aggregazione

All'interno delle Probe requests, ad accompagnare ogni MAC, vi sono informazioni utili alla derandomizzazione come timestamp, RSSI e sequence number della richiesta che vengono collezionate dallo script in un log da associare all'indirizzo MAC. L'esecuzione del processo porta ad avere una struttura dati in cui ad una signature possono essere associati più indirizzi MAC e ad un indirizzo MAC uno storico di log. Se una signature contiene molteplici MAC randomici, allora tramite uno studio sui log si cerca di capire se questi siano stati generati dallo stesso dispositivo o meno.

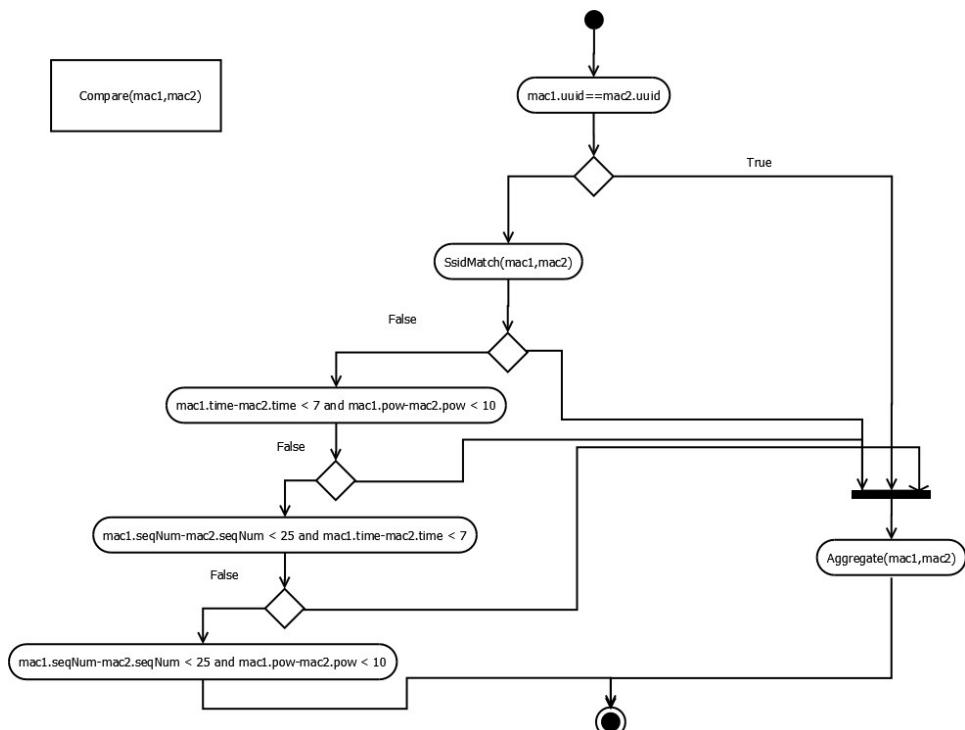


Figura 15 Activity diagram della funzione Compare(). Questa funzione viene usata per stabilire se due indirizzi MAC randomici siano stati generati dallo stesso dispositivo.

I criteri per l’aggregazione sono basati sulle differenze tra i log di due MAC differenti per i valori sopracitati:

1. piccole variazioni di sequence number;
2. medie variazioni di sequence number e potenza;
3. piccole variazioni di timestamp e potenza(?);

Seppur rari, vi sono casi in cui un dispositivo può trasmettere le informazioni WPS (Wi-Fi Protected Setup), protocollo supportato da alcuni dispositivi che permette una connessione sicura con un AP anche a dispositivi non autenticati, da cui si ricava l’UUID (Universally Unique Identifier) che, come scoperto in [36], viene generato direttamente dal MAC globale. In questi casi è sufficiente confrontare gli UUID per stabilire se i due MAC randomici provengano dallo stesso dispositivo. Un altro caso particolare si ha quando le richieste inviate dai dispositivi non sono di tipo broadcast ma indirizzate a specifici AP, quindi contenenti gli SSID degli hotspot con cui vorrebbero stabilire una connessione, conosciuti in precedenza. In una situazione del genere, liste di AP uguali portano all’aggregazione dei due MAC random. Una volta trovate le condizioni per cui è possibile l’aggregazione, al MAC con il log più recente vengono aggiunti i log dell’altro, che viene etichettato come “da cancellare”.

Un thread separato, era dedicato all’invio periodico dei dati raccolti tramite richiesta http ad uno script php che li scriveva su database ospitato su un sito di hosting online gratuito. Nella seconda fase del progetto questo sito di hosting si è trasformato in un calcolatore interno all’università, adibito a web server tramite uno stack LAMP.

### 5.3. Indipendenza

Il sottoscritto e il resto del team desideravamo una stazione che fosse in grado di autogestirsi. L’unica operazione che avrebbe dovuto eseguire l’uomo sarebbe stata quella di alimentare il dispositivo e assicurarsi che fosse connesso ad internet per l’invio dei dati. In caso di cali di alimentazione, crash inaspettati o interruzioni della rete, la stazione avrebbe dovuto intraprendere azioni correttive. In questo scenario di indipendenza ci siamo serviti del demone **Cron**, un processo di Unix che esegue istruzioni o comandi in determinate date e orari, predefinite dall’utente. **Cron** può gestire i comandi, “**jobs**”, in modo da eseguirli una sola volta alla prima occorrenza della data/ora prescelta, ma è anche possibile fargli eseguire ripetutamente i comandi, per esempio nel caso in cui si voglia eseguire una determinata operazione ogni giorno in una data ora, e in questo caso viene usato il ‘**Crontab**’. **Crontab** usa un demone chiamato **Crond**, un processo costantemente in esecuzione background, che una volta al minuto, legge i contenuti del registro comandi pianificati ed esegue quelli per cui si è esaurito il periodo di attesa. I comandi mandati in esecuzione da **Crontab** vengono chiamati **Cronjob**. Da crontab si possono specificare anche comandi da eseguire in determinate situazioni, ad esempio ad ogni reboot del dispositivo.

Fatte queste dovute premesse per poter procedere nella piena comprensione del documento, segue quello che era il crontab alla fine della prima fase del progetto.

```
#scheda di rete in monitor mode  
@reboot sudo ifconfig wlan1 up  
@reboot sudo airmon-ng start wlan1  
#script di sniffing al reboot  
@reboot sudo python snifferWiFi.py
```

*Equazione 2 crontab alla fine prima parte del progetto*

Ad ogni avvio del Raspberry, il demone Crond eseguiva i comandi per mettere in monitor mode la scheda wireless esterna e faceva partire lo script in Python.

Il grande inconveniente di questa prima parte del progetto era quello riguardante la saturazione della memoria RAM del Raspberry (926Mb) nel giro di qualche ora di esecuzione del processo. Momentaneamente, in attesa di una rivisitazione del codice, la soluzione, poco elegante, per cui optammo fu quella di aggiungere un comando di reboot ogni ora tra i job del crontab.

È bene precisare che il Raspberry è in possesso di una scheda wireless integrata che però non può essere messa in monitor mode. Da qui la necessità di una scheda esterna, notevolmente migliore anche per quanto riguarda la portata.

## 6. Seconda fase del progetto

Utilizzando come punto di partenza i risultati ottenuti nella prima fase, ho proseguito da solo, senza più l'ausilio dei due colleghi citati in precedenza, nella realizzazione di quella che definirei una seconda fase del progetto: il collocamento dei sensori presso i borghi delle Cinque Terre, la realizzazione dell'applicazione web GIS, e lo studio di una soluzione più elegante rispetto a quella della prima fase per l'autonomia dei dispositivi sniffer.

### 6.1. Gestione RAM

Per la questione RAM mi sono rivolto ai due studenti Contu Cristiano e Marchetti Mattia che ringrazio per l'interessamento al progetto. Il compito a loro affidato era inizialmente quello di alleggerire il carico computazionale degli sniffer demandando a loro esclusivamente le operazioni di sniffing e invio dei pacchetti al database. Tutto l'onere della computazione lo avrebbe dovuto avere un'applicazione lato server in modo da non doverci più preoccupare della potenza limitata dei Raspberry. Poco dopo aver effettuato le modifiche, i ragazzi hanno riscontrato errori nel ricostruire il pacchetto una volta inviato al server. Per poter scrivere su database il pacchetto grezzo, questo veniva convertito dal tipo `scapy.packet` al tipo stringa. L'operazione inversa non è stata possibile a causa della perdita di alcune informazioni durante i passaggi di tipo che, data la scarsa documentazione della libreria `scapy`, non siamo riusciti a riottenere.

A questo punto, i requisiti del progetto sono cambiati in aumentare l'autonomia dei dispositivi sviluppando un algoritmo che impiegasse efficacemente l'utilizzo della memoria principale.

Le modifiche inizialmente vedevano l'implementazione di un altro thread indipendente di cancellazione che periodicamente eliminasse la struttura dati contenente tutte le informazioni dello script e facesse l'uso manuale del garbage collector di Python. Dopo un'analisi del processo non si notavano significativi svuotamenti di memoria nel momento in cui il thread di cancellazione veniva eseguito, quindi si è deciso di non utilizzare il thread di cancellazione e di migliorare il modo in cui le entries della struttura dati venivano eliminate.

Alla fine di questa fase di riorganizzazione del codice ho ottenuto piccoli miglioramenti nella durata dell'esecuzione dello script che raggiungeva circa le quattro/sei ore ma che comunque necessitava in qualche modo di essere riavviato per ottenere la completa indipendenza, soprattutto alle Cinque Terre dove la mole di dati sarebbe stata sicuramente più elevata.

### 6.2. Riferimenti spazio-temporali

Il secondo punto che ho preso in considerazione è stato quello di aggiungere al database una tabella riportante MAC address per MAC address le rilevazioni di ingresso e uscita, in modo da avere traccia dell'ora in cui ogni dispositivo viene rilevato in una determinata stazione, in vista di un successivo possibile calcolo di itinerario per ogni MAC.

Nei capitoli successivi verrà fornita nel dettaglio la struttura del database completo.

### 6.3. Applicazione Web GIS

Se da un lato c'è la necessità di assemblare i dispositivi, preoccuparsi del loro funzionamento e della raccolta dati, dall'altro c'è anche il problema della visualizzazione. Per riprendere il discorso riguardo l'intasamento dei luoghi congestionabili e il monitoraggio in tempo reale, mi sono dedicato allo sviluppo di un'applicazione web che permettesse la visualizzazione dei dati in maniera grafica e intuitiva, direttamente su una mappa in corrispondenza delle coordinate geografiche in cui i dispositivi sarebbero stati posizionati.

Dopo una schermata di login, viene presentata all'utente la seguente schermata:

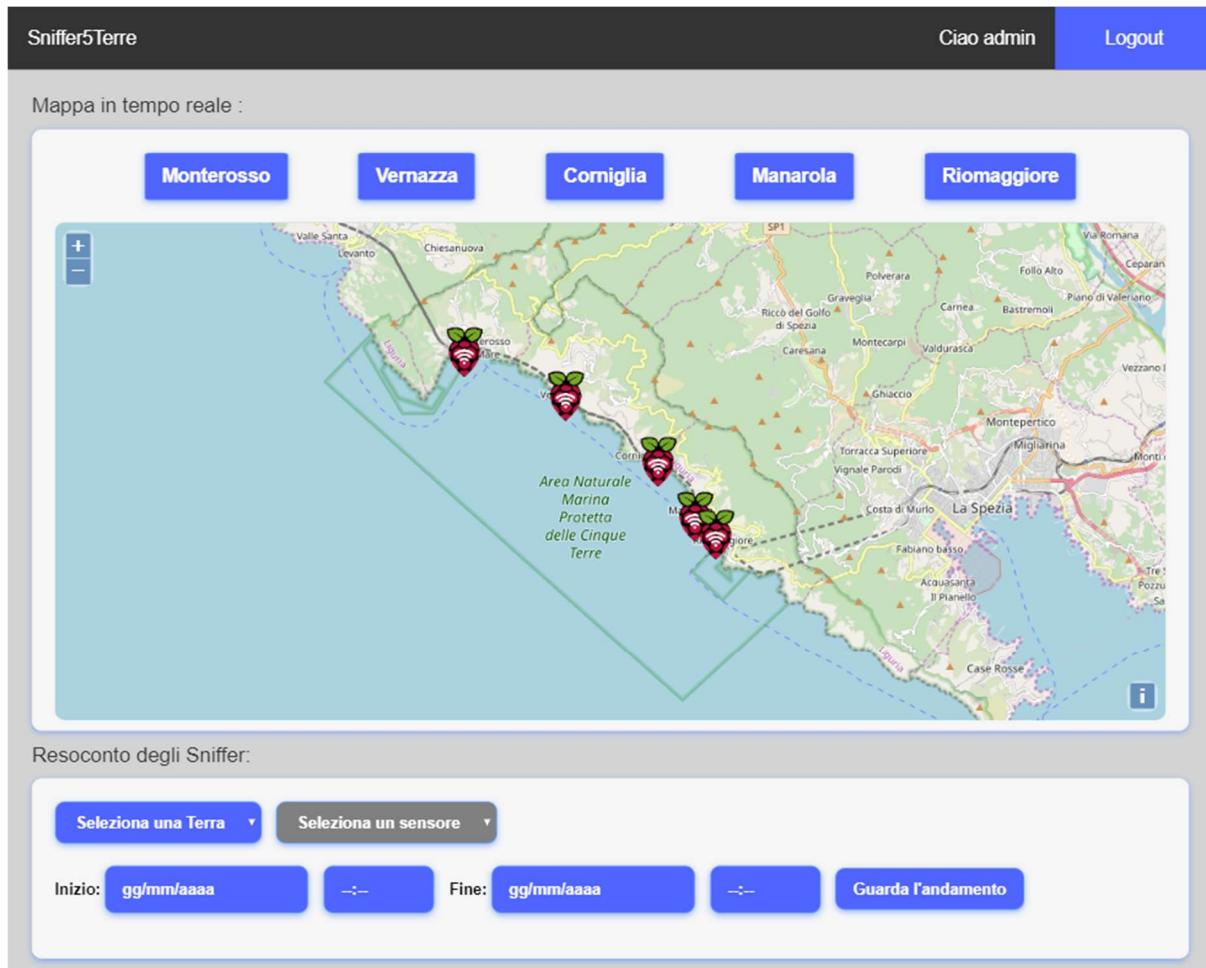
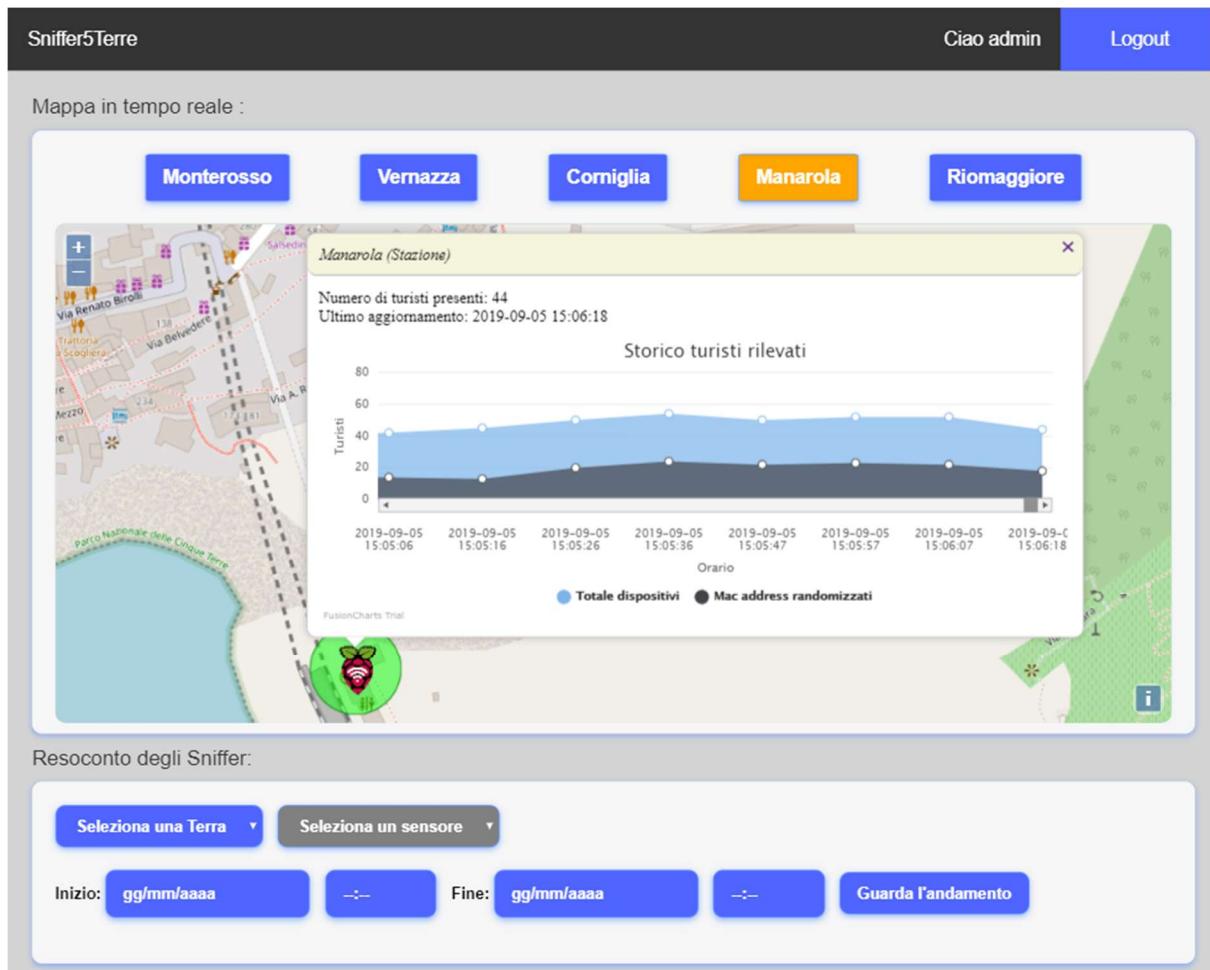


Figura 16 Console sniffer5terre

La mappa mostra una panoramica del Parco delle Cinque Terre in cui sono visibili i marker indicanti le posizioni degli sniffer che sono attivi. Cliccando su uno dei bottoni si viene catapultati all'interno del paese scelto e il marker ingrandito mostra intorno a sé quello che dovrebbe essere il suo raggio d'azione, colorato a seconda della densità turistica percepita in quel momento (verde poco affollato, giallo mediamente affollato, rosso molto affollato). La pagina web controlla periodicamente se ci sono cambiamenti di stato, dove lo "stato" è la condizione tale per cui l'area si riempie di un colore piuttosto che un altro, e se questo cambia, l'utente vede cambiare davanti a sé il colore automaticamente.

Il marker e la sua area sono cliccabili e permettono la visualizzazione di un popup che mostra maggiori informazioni quali il nome associato allo sniffer, il numero di turisti stimato e l'orario di ultimo aggiornamento a cui il dato si riferisce. Inoltre, all'interno della nuova scheda, viene rappresentato sottoforma di grafico lo storico di rilevamenti riferiti all'ultima ora trascorsa rispetto a quella attuale. Nel caso in cui per qualche malfunzionamento non fossero disponibili dati nell'ultima ora, il popup conterà la dicitura "Nessun dato disponibile nell'ultima ora" e visualizzerà l'ultimo rilevamento disponibile con la relativa marca temporale.



*Figura 17 Vista del popup*

Sotto alla scheda contenente la mappa, vi è una seconda scheda utile a visualizzare i dati raccolti da uno sniffer a scelta tra quelli disponibili e in un intervallo di tempo personalizzato.

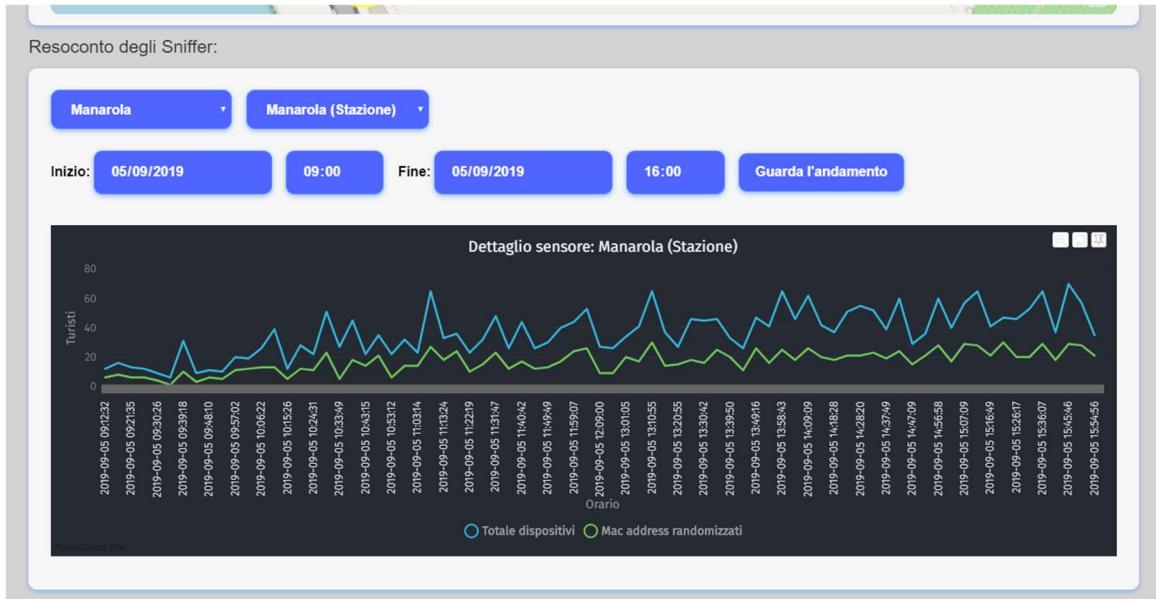


Figura 18 Selezione arbitraria di un intervallo di tempo

Siccome, a differenza del popup, l'intervallo di tempo può essere più esteso di un'ora, questo grafico permette lo zoom di un'area a scelta per ottenere una vista più dettagliata all'interno dell'intervallo selezionato.



Figura 19 Zoom sul grafico



Figura 20 Dettaglio a risoluzione maggiore

## 6.4. Collocamento degli sniffer

Il passo fondamentale per il corretto funzionamento dell'applicazione web è stato quello dell'individuazione di spot idonei al posizionamento dei dispositivi. I luoghi di interesse possono essere piazze, stazioni, o passaggi che si comportino come collo di bottiglia per il flusso pedonale e che quindi sarebbe bene tenere sotto controllo.

Siccome i dati devono essere disponibili su un database online per poter essere visualizzati dall'applicazione, è opportuno che gli sniffer abbiano l'accesso ad Internet, tramite LAN o tramite GPRS. Siccome il progetto ha visto in questo periodo la sua prima vera realizzazione, in accordo con l'Ente Parco, i luoghi che inizialmente sono stati individuati per la messa in atto corrispondono alle cinque stazioni ferroviarie dei cinque paesi, o meglio ai 5 Terre Point in corrispondenza delle stazioni. Oltre a consentire una visione globale delle cinque frazioni, questa strategia ci ha permesso di connettere i dispositivi tramite LAN agevolando il processo di connessione ad Internet.

Visto che il progetto in sé sarebbe stato, ed è tutt'ora, un esperimento per determinare la fattibilità di monitoraggio e di predizione, una volta piazzati i dispositivi questi avrebbero dovuto essere accessibili tramite reti esterne in modo tale da poter effettuare aggiornamenti software senza dover accederci da una rete locale, quindi senza effettivamente recarsi in loco.

Per sopperire a questo problema, ogni dispositivo è stato registrato sul sito remot3.it [38] ed equipaggiato con l'installazione dell'applicazione dedicata che permette di eseguire il reverse ssh tunneling.

Nei prossimi paragrafi seguiranno le posizioni nel dettaglio di ogni singolo dispositivo installato.

### 6.4.1. Monterosso

Monterosso è la prima stazione raggiungibile se si viaggia da nord verso sud ed è anche una delle località più trafficate assieme a Riomaggiore in quanto, a differenza dei borghi interni, il primo e l'ultimo sono soggetti ad un numero maggiore di soste da parte dei treni regionali.



Figura 21 Stazione di Monterosso



Figura 22 5 Terre Point di Monterosso - esterno



Figura 23 5 Terre Point di Monterosso - interno

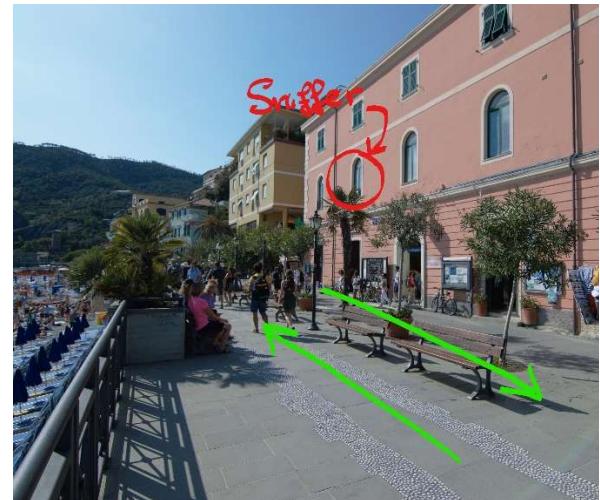


Figura 24 Passeggiata di Monterosso

Come si può notare dalle immagini, il dispositivo è posizionato in modo tale da poter ascoltare il traffico sia all'interno della stazione che lungo la passeggiata sottostante.

### 6.4.2. Vernazza

Vernazza è il secondo borgo che si incontra da nord a sud e, nonostante sia uno di quelli interni, anch'esso è solito ricevere un grande afflusso di turisti. L'ufficio del 5 Terre Point di Vernazza è proprio tra i due binari della stazione, il che permette un monitoraggio completo di questa. Siccome l'accesso alla ferrovia è possibile solo tramite una scala che scende sotto il ponte della stazione, anche l'area sottostante è raggiungibile dallo sniffer.



Figura 25 Stazione di Vernazza



Figura 26 Stazione di Vernazza

### 6.4.3. Corniglia

Proseguendo il tragitto verso sud si arriva al terzo paese, Corniglia. Anche in questo caso, come a Vernazza, la stazione ha un'unica via d'uscita dove i turisti si incamminano verso il paese e i sentieri. Il 5 Terre Point si trova proprio accanto a questo vicolo di cui fa angolo, costringendo i passanti a transitare all'interno del raggio d'azione dello sniffer.

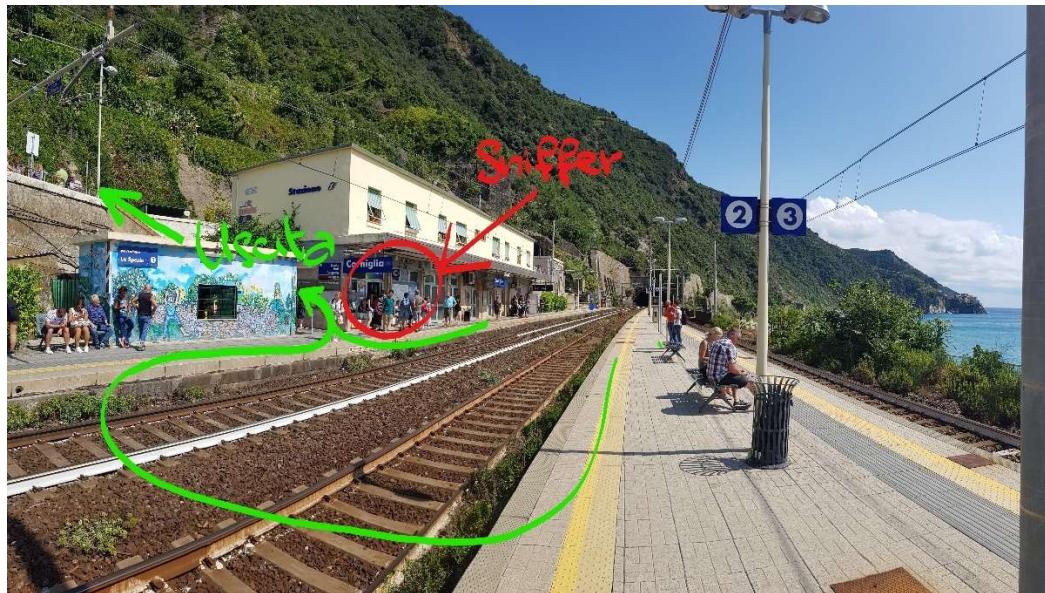


Figura 27 Stazione di Corniglia



Figura 28 Via d'uscita della stazione di Corniglia

#### 6.4.4. Manarola

Il penultimo paese della lista è Manarola. Come le due frazioni precedenti, per uscire dalla stazione i turisti sono costretti a transitare presso una piazzetta nei pressi dell'edificio adibito a 5 Terre Point. Nel caso di Manarola, un tunnel piuttosto stretto, il cui ingresso è sito di fronte all'edificio in questione, s'interpone tra la stazione e il paese. Motivo in più per monitorare la suddetta area.



Figura 29 Piazzetta di Manarola antistante lo sniffer



Figura 30 5 Terre Point di Manarola

#### 6.4.5. Riomaggiore

L'ultimo borgo, che assieme a Monterosso delimita i confini delle Cinque Terre, è Riomaggiore in cui, come già accennato, il traffico si intensifica. Ci sono diversi punti d'ingresso o uscita dalla stazione, alcuni non proprio vicini al 5 Terre Point, ma questo ha la fortuna di trovarsi nella piazza in cui i turisti transitano per raggiungere il paese, anche questa volta, tramite un tunnel.



Figura 31 5 Terre Point di Riomaggiore - esterno

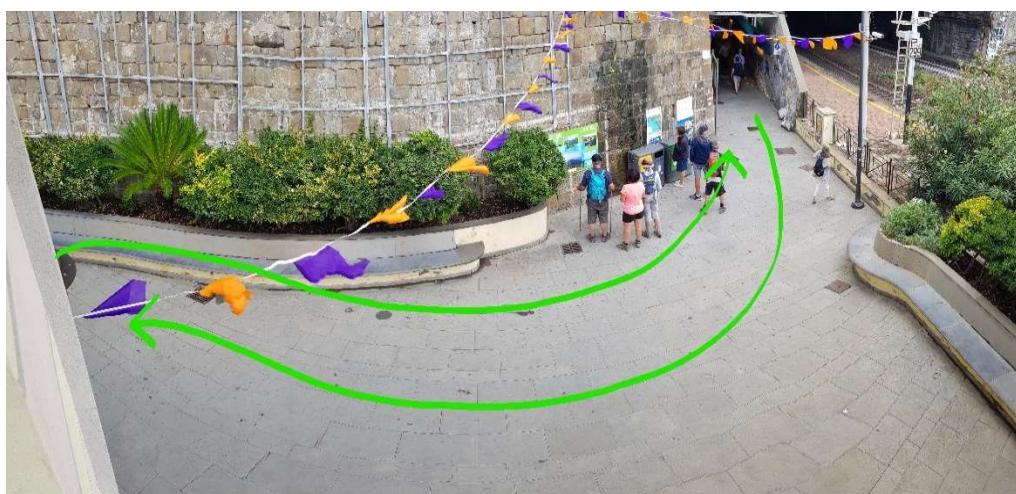


Figura 32 Accesso al tunnel di Riomaggiore raggiungibile dallo sniffer



Figura 33 5 Terre Point di Riomaggiore – interno

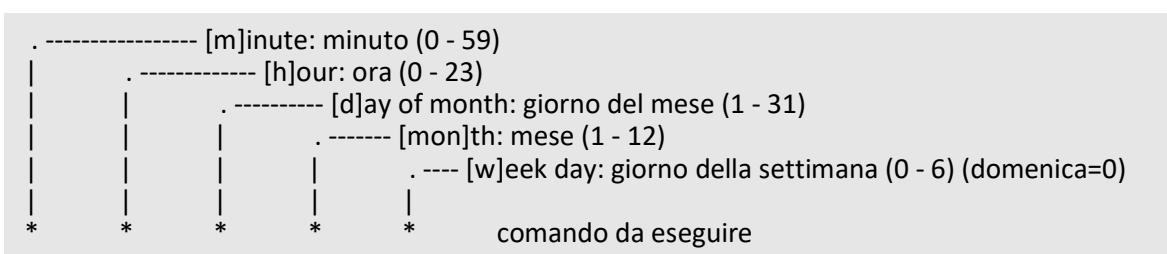
## 6.5. Autonomia degli sniffer

Per quello che riguarda la soluzione artigianale del reboot periodico che avevamo momentaneamente pensato alla fine della prima fase del progetto è necessaria una piccola premessa.

- Un sistema Linux-based mostra le periferiche wireless come wlanX e le periferiche ethernet come ethX dove X corrisponde ad un indice intero crescente  $\geq 0$  utilizzato per distinguere le interfacce.
  - Lo script di sniffing ha bisogno che gli venga passato come parametro il nome dell'interfaccia di rete con la quale sniffare il traffico.
  - L'interfaccia da passare deve essere dapprima identificata mediante il suo nome e successivamente essere messa in monitor mode, che tra le altre cose comporta l'aggiunta del suffisso "mon" al suo nome.

Di conseguenza, convinti che la scheda wireless integrata ottenessse sempre l'identificativo wlan0 e quella esterna wlan1, all'interno dello script python il nome dell'interfaccia con cui sniffare era hardcoded e sempre lo stesso: wlan1mon. Facendo girare il Raspberry per diversi giorni, mi sono accorto che non venivano assegnati sempre gli stessi identificatori alle interfacce di rete ma capitava che venissero invertiti. Dal momento il parametro era hardcoded, poteva non esserci corrispondenza tra questo e l'identificativo assegnato dal sistema operativo alla scheda esterna. Il cronjob in **[Errore. L'origine riferimento non è stata trovata.]**, tendando di mettere in monitor mode wlan1, avrebbe potuto eseguire l'operazione sulla scheda integrata che, non potendo essere messa in monitor mode, non avrebbe permesso la generazione di wlan1mon e lo script in python sarebbe andato incontro ad un errore. Ho trovato due soluzioni possibili che saranno descritte nel dettaglio tecnico nel prossimo capitolo.

Documentandomi sul crontab, sono venuto a conoscenza riguardo le sue vere potenzialità. Per schedulare un job possiamo ricorrere a cinque unità di misura secondo il seguente schema



### *Equazione 3 Struttura di un cronjob*

in cui:

- L'operatore virgola `,` specifica una lista di valori, ad esempio: `1,3,4,7,8`.
  - L'operatore trattino `-` specifica un intervallo di valori, ad esempio: `1-6`, che equivale a `1,2,3,4,5,6`.
  - L'operatore asterisco `*` specifica tutti i possibili valori di un campo. Ad esempio, un asterisco nel campo dell'ora è equivalente a «ogni ora»

Per esempio, per eseguire myScript.py ogni ora tra le 10:00 e le 22:00 ogni giorno si utilizzerà:

```
0 10-21 * * * python myScript.py
```

*Equazione 4*

che si traduce in: ogni minuto 0 di ogni ora compresa tra le 10 e le 21(compresa) di ogni giorno del mese, di ogni mese, in ogni giorno della settimana, esegui myScript.py. Come si può intuire l'unità di misura più piccola che si può controllare è il minuto e per eseguire un cronjob in un intervallo di n minuti, diciamo 30, si può utilizzare l'operatore / posizionando nello slot dei minuti: \*/30.

Per maggiori spiegazioni riguardo il crontab consultare [39] o [40].

## 7. Dettagli tecnici

In questo capitolo illustrerò il sistema per come è a progetto concluso, la logica degli sniffer, la struttura del database e la realizzazione dell'applicazione web.

### 7.1. Specifiche tecniche

Ogni stazione indipendente è caratterizzata dalle seguenti componenti:

- Scheda di rete ALFA Network AWUS036NH



Chipset	Ralink RT3070
WiFi Standards	IEEE 802.11/b/g/n
WiFi Frequency	2.4GHz
Antenna Type	2.4GHz 5dBi dipole antenna
Wireless Performance	802.11b: up to 11Mbps 802.11g: up to 54Mbps 802.11n: up to 150Mbps
Wireless Security Interface	WEP, WPA, WPA2, WPA Mixed, WPS USB 2.0, Mini USB, type B female connector
OS Requirement	Windows XP, Vista, 7, 8/8.1 and Windows 10 32/64bit, macOS 10.5 to 10.14 or later Linux

- Raspberry Pi 3 Model B+ & OS



SoC	Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
Networking	Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
RAM	1GB LPDDR2 SDRAM
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)
GPIO	40-pin GPIO header, populated
Storage	microSD
Ports	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
Dimensions	82mm x 56mm x 19.5mm, 50g
Operating System	Raspbian GNU/Linux 10 (buster)
Kernel	Linux 4.19.66-v7+

## 7.2. Monitor mode

Come già anticipato, alla fine della prima fase del progetto vi era un errore che non permetteva lo sniffing con la corretta scheda wireless. Per capire le soluzioni, deve essere fatta un'altra premessa su come gestire le interfacce di rete in Unix.

Con il comando shell ifconfig, un sistema Linux-based mostra le interfacce di rete. Il comando iwconfig è simile a ifconfig ma è dedicato alle interfacce wireless.

```
pi@raspberrypi:~ $ iwconfig
eth0      no wireless extensions.

lo      no wireless extensions.

wlan0    IEEE 802.11 ESSID:off/any
        Mode:Managed Access Point: Not-Associated Tx-Power=31 dBm
        Retry short limit:7  RTS thr:off  Fragment thr:off
        Power Management:on

wlan1    IEEE 802.11 ESSID:off/any
        Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
        Retry short long limit:2  RTS thr:off  Fragment thr:off
        Power Management:off
```

*Equazione 5 output di iwconfig*

Come si può notare dal precedente output, entrambe le schede di rete wireless collegate al Raspberry (quella esterna usb e quella integrata sulla scheda madre) hanno come dicitura Mode:Managed, che è la modalità classica per potersi collegare ad una rete e navigare in internet.

Ai fini del progetto, è necessario cambiare modalità alla scheda wireless esterna e metterla in monitor mode. Per farlo abbiamo utilizzato la suite aircrack-ng. Supponendo che wlan0 si riferisca alla scheda integrata sulla scheda madre e wlan1 alla scheda esterna segue:

```

pi@raspberrypi:~ $ sudo ifconfig wlan1 up
pi@raspberrypi:~ $ sudo airmon-ng start wlan1

PHY Interface Driver Chipset
phy0 wlan0 brcmfmac Broadcom 43430
phy1 wlan1 rt2800usb Ralink Technology, Corp. RT5370

(mac80211 monitor mode vif enabled for [phy1]wlan1 on [phy1]wlan1mon)
(mac80211 station mode vif disabled for [phy1]wlan1)

pi@raspberrypi:~ $ iwconfig
eth0 no wireless extensions.

wlan1mon IEEE 802.11 Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm
    Retry short long limit:2 RTS thr:off Fragment thr:off
    Power Management:off

lo no wireless extensions.

wlan0 IEEE 802.11 ESSID:off/any
    Mode:Managed Access Point: Not-Associated Tx-Power=31 dBm
    Retry short limit:7 RTS thr:off Fragment thr:off
    Power Management:on

```

*Equazione 6 iwconfig dopo aver messo la scheda in monitor mode*

Si noti che wlan1 è passata in Mode:Monitor e ora si chiama wlan1mon. Siccome l'indice si manifesta in ordine crescente, durante la fase 1, pensavamo che alla scheda esterna venisse sempre assegnato l'identificatore wlan1, per cui lo script di sniffing riceveva wlan1mon come scheda con cui sniffare. Come anticipato, l'assegnazione non è sempre così e tentare di mettere in monitor mode la scheda integrata non generava nessuna wlan1mon, mandando in errore lo script.

La prima soluzione ha come fine quello di disabilitare la scheda wireless integrata sulla scheda madre del Raspberry, in modo che l'unica abilitata e visibile rimanga quella esterna e di conseguenza prenda sempre il nome wlan0. Non essendoci più altre schede wireless in competizione per l'assegnazione dei nomi, il parametro da passare alla funzione di sniff sarà sempre hardcoded ma questa volta "wlan0mon".

È servita qualche ricerca su internet per scoprire che per disabilitare il WiFi integrato è necessario modificare un file di configurazione del sistema operativo raggiungibile in */etc/modprobe.d/raspi-blacklist.conf* aggiungendo le seguenti righe:

```

blacklist brcmfmac
blacklist brcmutil

```

*Equazione 7 /etc/modprobe.d/raspi-blacklist.conf*

La seconda soluzione ha come fine quello di provare a mettere in monitor mode una delle due interfacce tramite uno script shell che si esegue ad ogni reboot del dispositivo. Se ifconfig mostra una scheda in monitor mode allora si è eseguita l'operazione sulla scheda corretta e si passa quell'identificativo alla funzione. Se ifconfig non mostra nessuna scheda in monitor mode allora si ripete la stessa operazione sulla seconda interfaccia che sicuramente sarà quella corretta. Lo script in python deve essere esteso con il modulo netifaces che permette di vedere le interfacce di rete e con un semplice ciclo che salva il nome più lungo che riesce a trovare tra queste.

```
@reboot sudo sh monitor-routine.sh
```

*Equazione 8 Script shell da eseguire al riavvio all'interno del crontab*

```
sudo ifconfig wlan0 up
sudo airmon-ng start wlan0
T2=$(ifconfig | grep "wlan0mon" | cut -d ':' -f1)
#se la scheda era wlan0
if [${T2}="wlan0mon"]
then
    echo "wlan0mon activated"
else
    sudo ifconfig wlan1 up
    sudo airmon-ng start wlan1
    echo "wlan1mon activated"
fi
```

*Equazione 9 monitor-routine.sh*

```
import netifaces
x=netifaces.interfaces()
iface=""

for interface in x:
    if(len(interface)>len(iface)):
        iface=interface
```

*Equazione 10 Codice aggiuntivo nello script di sniffing in python*

Siccome nel nostro ambiente esistono solo wlan0 e wlan1 e soltanto una delle due verrà rinominata in wlanXmon, allora iface conterrà l'identificativo corretto da passare alla funzione di sniff.

### 7.3. Script Python

Dalla prima fase del progetto alla sua conclusione, il codice ha subito diverse modifiche, alcune per migliorare la complessità computazionale in termini di operazioni necessarie a confrontare i MAC randomici, altre per limitare lo spreco della preziosa e limitata RAM dei Raspberry.

Ad oggi lo script di sniffing è composto da due thread separati, uno per lo sniffing e la computazione del pacchetto sniffato (ThreadSniffing) e uno per l'invio periodico dei dati al database (ThreadInvio).

Segue l'Activity diagram del primo thread.

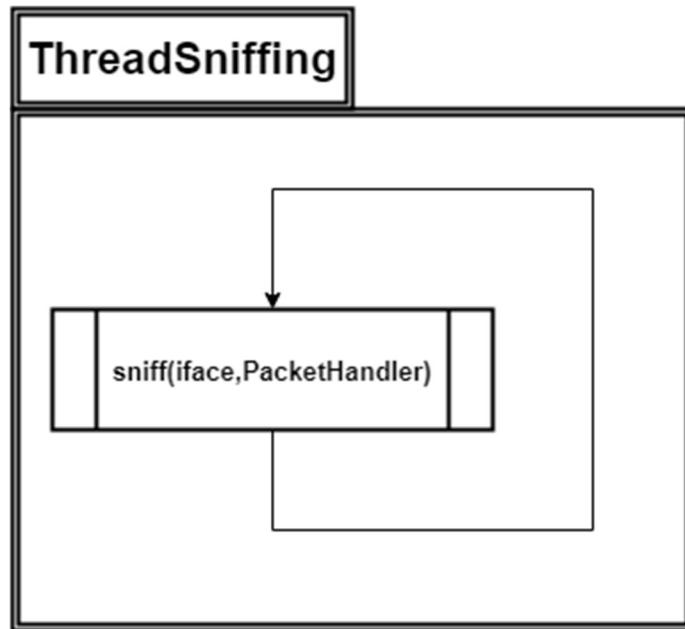
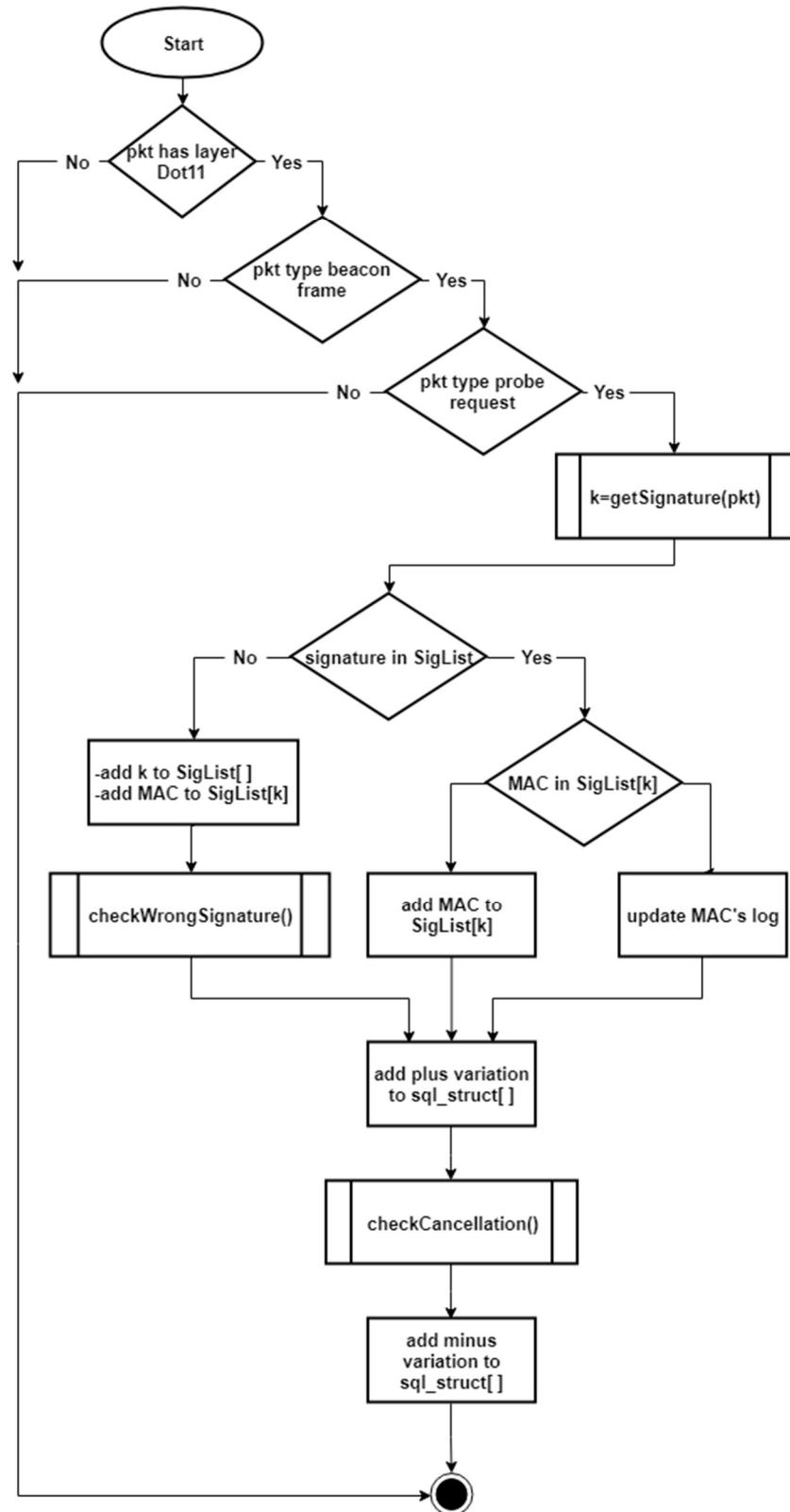


Figura 34 Activity diagram di ThreadSniffing

Il thread in questione è un processo che esegue in loop la funzione sniff fornita dalla libreria scapy. Come menzionato nei capitoli 6.1 e 7.1, questa ha bisogno di un parametro che indica l'interfaccia di rete con cui ascoltare il traffico e di un secondo parametro, una funzione da eseguire dopo aver ricevuto il pacchetto. Di seguito l'Activity diagram di PacketHandler:

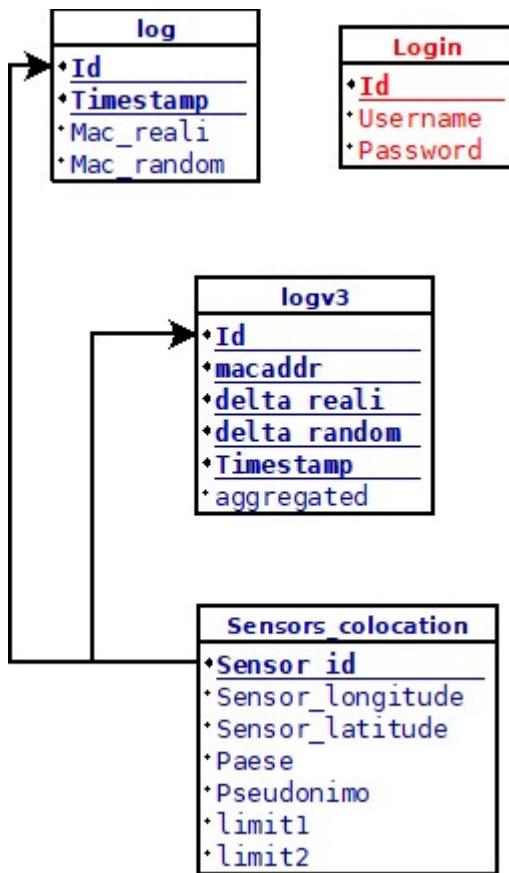
## PacketHandler(pkt)



## 7.4. Database

Il progetto si appoggia ad un sistema Debian GNU/Linux 9 (stretch) su cui è stato opportunamente installato uno stack LAMP. Il server ospita il database e l'applicazione web descritta nel capitolo 6.3.

Il database è composto da quattro tabelle due delle quali sono riempite dagli sniffer piazzati nelle aree di monitoraggio.



### 7.4.1. Login

Da specifiche iniziali, l'applicazione avrebbe dovuto consentire la visualizzazione dei dati solamente agli utenti autenticati del parco. Come questi otterranno le credenziali non è ancora stato specificato. In futuro potrebbero essere presenti funzionalità variabili a seconda del livello di accesso o le credenziali potrebbero essere le stesse per tutti. In ogni caso la tabella è già predisposta e pronta all'utilizzo.

- **Id**: chiave primaria. Indice intero che si autoincrementa;
- **Username**: username;
- **Password**: password.

Field	Type	Null	Key	Default	Extra
Id	int(11)	NO	PRI	NULL	auto_increment
Username	varchar(24)	NO		NULL	
Password	varchar(52)	NO		NULL	

#### 7.4.2. Sensors\_colocation

In Sensors\_colocation sono riportati i dati distintivi degli sniffer. Ogni riga è aggiunta manualmente ogni volta che il sistema viene esteso con un nuovo sensore sniffer.

- **Sensor\_id:** chiave primaria. Identificatore univoco intero assegnato ad ogni sniffer;
- **Sensor\_longitude:** coordinata longitudinale della posizione dello sniffer;
- **Sensor\_latitude:** coordinata latitudinale della posizione dello sniffer;
- **Paese:** nome del paese delle Cinque Terre a cui le coordinate si riferiscono;
- **Pseudonimo:** pseudonimo dello sniffer per facilitarne la lettura sulla mappa e la distinzione tra possibili sniffer localizzati all'interno dello stesso paese;
- **limit1:** limite di persone secondo il quale l'area sulla mappa passa da verde a gialla;
- **limit2:** limite di persone secondo il quale l'area sulla mappa passa da gialla a rossa.

Field	Type	Null	Key	Default	Extra
Sensor_id	int(11)	NO	PRI	NULL	
Sensor_longitude	decimal(7,6)	NO		NULL	
Sensor_latitude	decimal(8,6)	NO		NULL	
Paese	varchar(32)	NO		NULL	
Pseudonimo	varchar(32)	NO		NULL	
limit1	int(11)	NO		NULL	
limit2	int(11)	NO		NULL	

#### 7.4.3. log

Log è la prima tabella che i sensori popolano durante l'esecuzione dello script di sniffing. Ciclicamente vengono inviati i dati relativi a quanti dispositivi sono presenti nel raggio d'azione del sensore riferiti a quell'istante di tempo. I dati che vengono inviati sono:

- **Id:** parte di chiave primaria composta. Indica l'identificativo del sensore da cui provengono i dati della tupla;
- **Timestamp:** seconda parte di chiave primaria composta. Indica l'istante di tempo a cui si riferiscono i dati inviati;
- **Mac\_reali:** numero di dispositivi con MAC address globale in memoria in quell'istante;
- **Mac\_random:** numero di dispositivi con MAC address randomizzato in memoria in quell'istante. Il dato si riferisce già al numero fornito in output dal processo di aggregazione.

Field	Type	Null	Key	Default	Extra
Id	int(11)	NO	PRI		NULL
Timestamp	timestamp	NO	PRI	CURRENT_TIMESTAMP	
Mac_reali	int(11)	NO			NULL
Mac_random	int(11)	NO			NULL

#### 7.4.4. logv3

L'ultima tabella del database è nuovamente riempita dai sensori durante l'esecuzione dello script python e contiene l'informazione relativa all'aggiunta e rimozione di ogni singolo MAC address rilevato dal set degli indirizzi in memoria in un dato istante. In caso di sviluppi futuri, se si volessero costruire gli itinerari percorsi dai turisti rilevati di stazione in stazione, grazie a questa tabella si potrebbero identificare le prime e ultime rilevazioni di uno stesso indirizzo in ogni punto di interesse.

- **Id:** parte di chiave primaria composta. Indica l'identificativo del sensore da cui provengono i dati della tupla;
- **macaddr:** parte di chiave primaria composta. Indica l'indirizzo MAC in ingresso o uscita dal set attualmente in memoria;
- **delta\_reali:** parte di chiave primaria composta. Indica la variazione del numero di MAC globali conseguente all'ingresso/uscita di macaddr. Valori ammessi -1, 0, +1.
- **delta\_random:** parte di chiave primaria composta. Indica la variazione del numero di MAC randomizzati conseguente all'ingresso/uscita di macaddr. Valori ammessi -1, 0, +1.
- **Timestamp:** parte di chiave primaria composta. Indica l'istante di tempo in cui l'ingresso/uscita è stata rilevata;
- **aggregated:** riferito ad un indirizzo randomizzato. Indica se macaddr sia uscito dal set di indirizzi in memoria perché aggregato ad un altro (valore 1) o se perché trascorso il timeout per cui l'indirizzo viene considerato vecchio (valore 0).

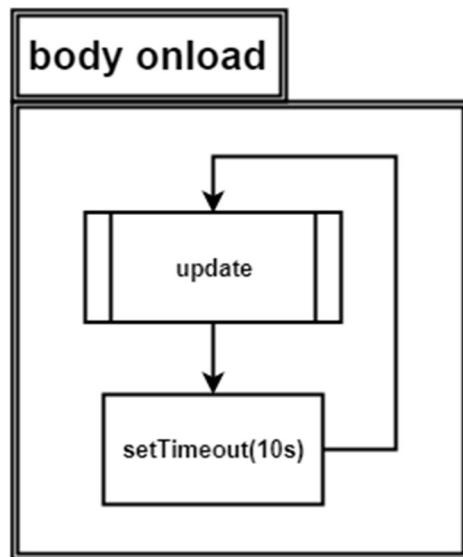
Field	Type	Null	Key	Default	Extra
Id	int(11)	NO	PRI		NULL
macaddr	varchar(32)	NO	PRI		NULL
delta_reali	int(11)	NO	PRI		NULL
delta_random	int(11)	NO	PRI		NULL
Timestamp	datetime	NO	PRI		NULL
aggregated	int(11)	NO			NULL

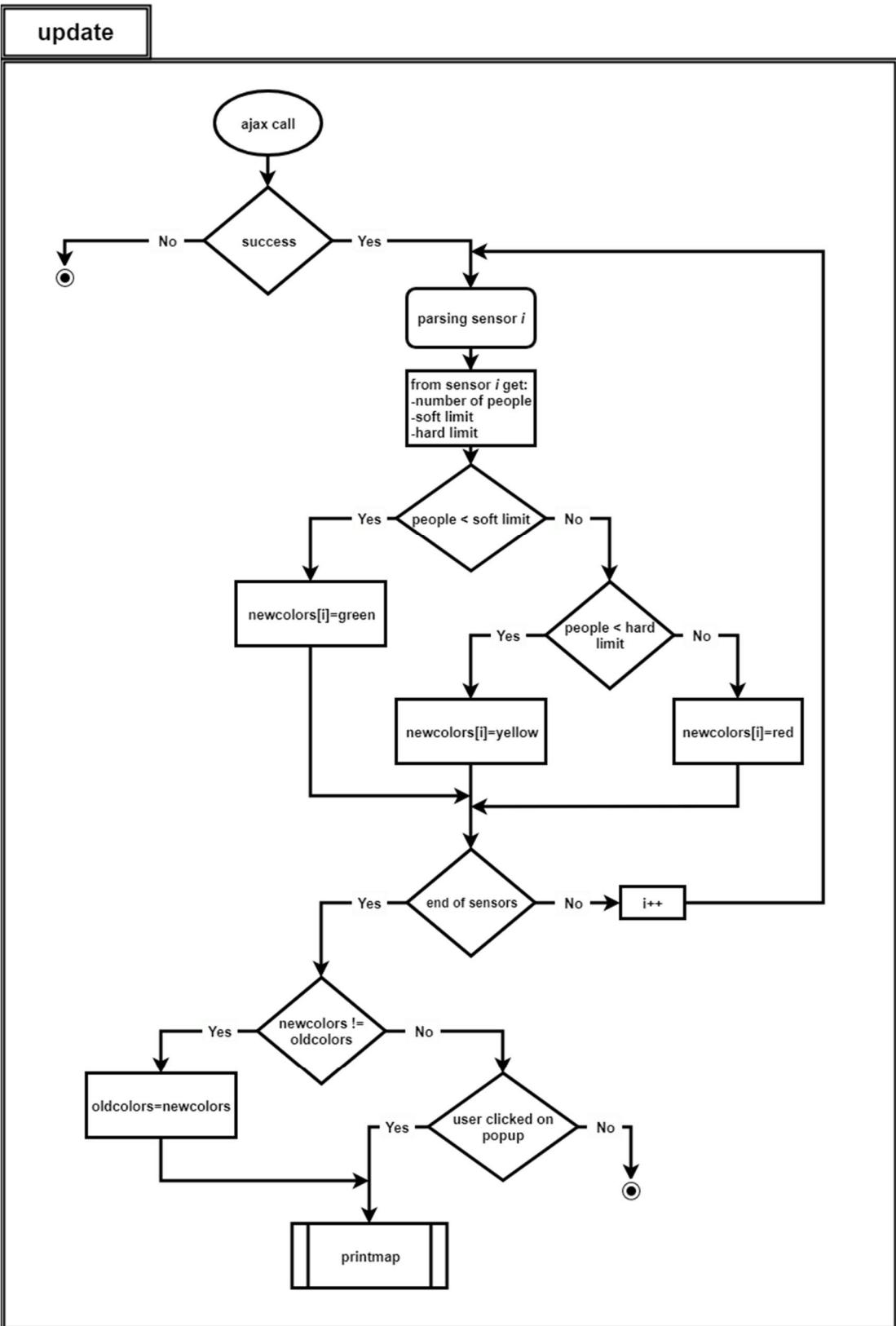
Siccome ci sono molteplici sensori che popolano questa tabella, l'unico modo per garantire una chiave primaria univoca è la combinazione dei primi cinque campi.

## 7.5. Applicazione Web

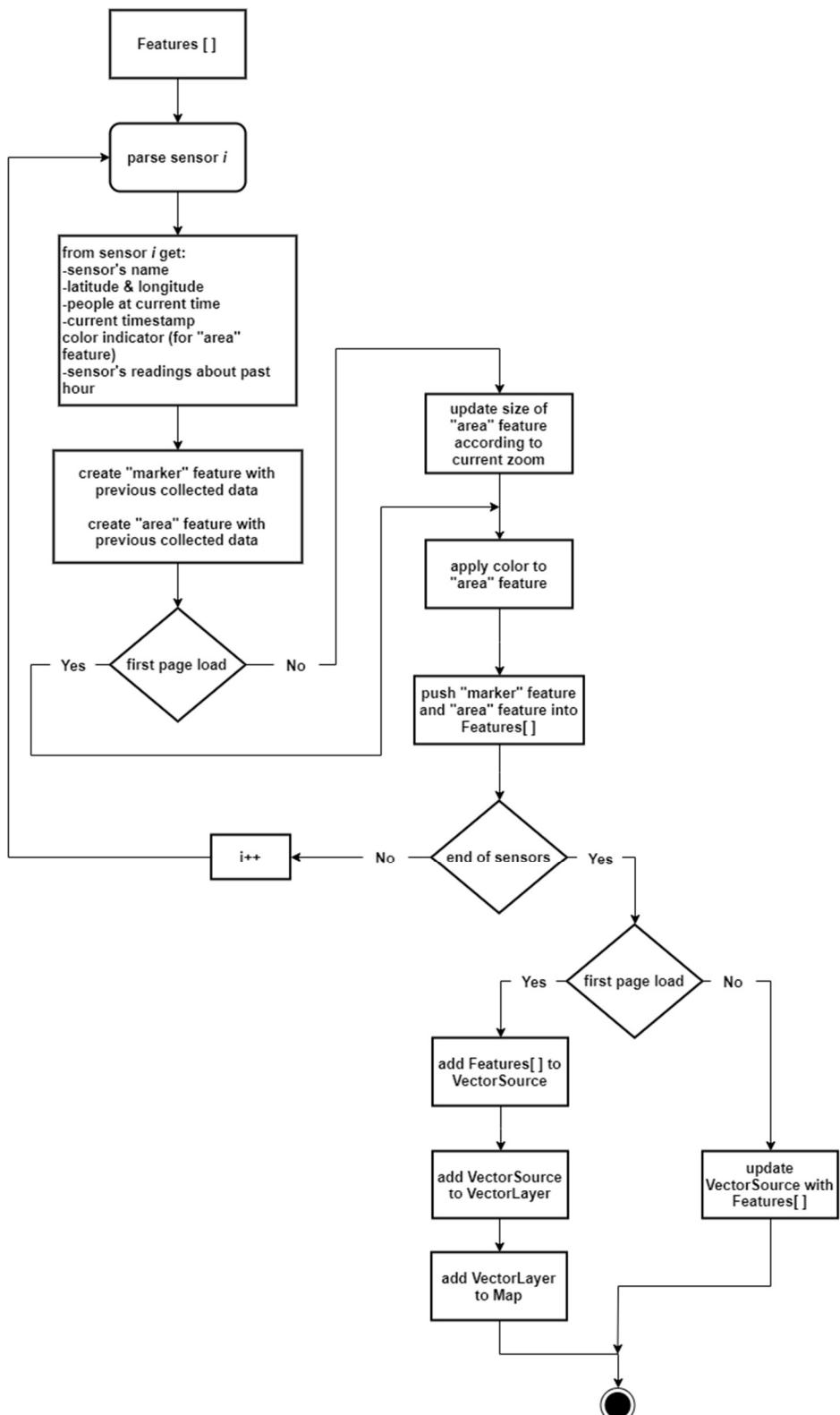
La console di visualizzazione descritta nel capitolo 6.3 utilizza OpenLayers 3 e OpenStreetMap. Il codice javascript contenuto nella pagina richiede periodicamente i dati al database precedentemente descritto tramite chiamate ajax e modifica la mappa in caso vi siano cambiamenti di stato. Lo stato è quell'insieme di condizioni che determinano i colori di ogni sensore presente sulla mappa. Di conseguenza, nel caso in cui presso uno dei sensori si verifichi una congestione improvvisa, l'utente vedrà cambiare il colore dell'area interessata automaticamente.

Oltre l'aggiornamento periodico, la funzione di update può anche essere invocata manualmente quando uno dei marker indicanti i sensori vengono cliccati per mostrare i dati aggiornati del grafico contenuto nel popup.





## printmap



## 8. Terza fase del progetto

La terza fase del progetto ha visto come attività quelle di adattamento del codice al luogo di posizionamento, di taratura e calibrazione dei dispositivi. Fino a prima del collocamento dei dispositivi presso le Cinque Terre, questi erano stati testati in ambienti non congestionabili quanto quelli definiti dal progetto, per cui è stata necessaria un'analisi del loro funzionamento in loco.

CPU 100%

CONTENUTO RICHIESTA ELEVATO

MEMORIA IN FASCE ORARIE

Siccome lo script di sniffing non è in grado di girare ventiquattr'ore su ventiquattro per un ammontare di tempo indefinito, è stato opportuno raccogliere i dati necessari ad individuare le fasce di tempo in cui eseguirlo. Come configurazione di partenza, è stato compilato un crontab che eseguisse lo script ogni ora per 3580 secondi. Un altro cronjob, consiste nella scrittura periodica (ogni 14 minuti) della percentuale di RAM utilizzata con la relativa data e ora di riferimento. Ad ogni esecuzione dello script quindi, corrispondono 5 dati riguardanti la RAM (per ogni ora: minuto 0, minuto 14, minuto 28, minuto 42 e minuto 56).

```
#al reboot scheda in monitor mode  
@reboot sudo sh monitor-routine.sh  
  
#ogni 14 minuti di ogni ora scrittura % ram utilizzata (minuti 0,14,28,42,56)  
*/14 * * * * sh memory-script.sh  
  
#script eseguito ogni 60 minuti (20 secondi di intervallo)  
0 * * * * sudo timeout 3580 python snifferWiFi_v3-5.py
```

Equazione 11 Configurazione iniziale del file crontab

Memory-script.sh si serve del comando shell **free**, di cui un esempio sotto, per calcolare la percentuale di memoria impiegata rispetto a quella disponibile e la riporta in un file di log chiamato memory-log.txt assieme al timestamp in cui la percentuale è stata calcolata.

```
pi@raspberrypi:~ $ free  
  
              total        used        free      shared  
Mem:       948304      395376      450632       6336      102296      494232  
Swap:      102396          0      102396
```

Equazione 12 Esempio di output del comando free

```
echo $(date '+%a %d %b %Y %T') $(free | grep Mem | awk '{print ($3+$5+$6)/$2 * 100.0}')% >>  
memory-log.txt
```

Equazione 13 memory-script.sh

Per una settimana, dal 03/09/2019 al 10/09/2019, il Raspberry posizionato presso il welcome center di Manarola ha girato secondo la precedente impostazione. Per riassumere il contenuto del log verranno riportate le percentuali riguardanti l'ultima registrazione delle ore più trafficate (10-20) (quella del minuto 56) poiché il file per intero conta 840 entries. In aggiunta è stato affiancato il numero medio di dispositivi rilevati nell'arco di quell'ora, per sottolineare la correlazione tra numero di dispositivi e consumo della memoria.

## 9. Considerazioni finali

9.1. Precisione

9.2. Accuratezza

9.3. Portata massima

9.4. Perdita informazioni

## 10. Conclusione

## 11. Bibliografia

- [1] Consorzio Turistico Cinque Terre, «Benvenuto su 5 Terre | 5 Terre,» [Online]. Available: <https://www.cinquerterre.it/>.
- [2] D. Helbing, «Models for Pedestrian Behavior,» 1998.
- [3] J. J. Fruin, «Designing for pedestrians: a level of service concept,» *Highway Research Board*, pp. 1-15, 1971.
- [4] A. Polus, J. L. Schofer e A. Ushpiz, «Pedestrian Flow and Level of Service,» *Journal of Transportation Engineering*, vol. 109, n. 1, pp. 46-56, 30 12 2008.
- [5] U. Weidmann, «Transporttechnik der Fussgänger Transporttechnische Eigenschaften des Fussgängerverkehrs, Literaturauswertung,» 1992.
- [6] G. Xiong, J. Cheng, X. Wu, Y. L. Chen, Y. Ou e Y. Xu, «An energy model approach to people counting for abnormal crowd behavior detection,» *Neurocomputing*, vol. 83, pp. 121-135, 15 4 2012.
- [7] P. Vera, S. Monjaraz e J. Salas, «Counting pedestrians with a zenithal arrangement of depth cameras,» *Machine Vision and Applications*, vol. 27, n. 2, pp. 303-315, 1 2 2016.
- [8] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi e S. Z. Li, «Water filling: Unsupervised people counting via vertical kinect sensor,» in *Proceedings - 2012 IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2012*, 2012.
- [9] V. A. Sindagi e V. M. Patel, «A survey of recent advances in CNN-based single image crowd counting and density estimation,» *Pattern Recognition Letters*, vol. 107, pp. 3-16, 1 5 2018.
- [10] Q. Chang, Y. Qi, W. Zhou e J. Liu, «An Auto-adaptive CNN for Crowd Counting in Monitor Image,» in *Proceedings of 2018 6th IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2018*, 2018.
- [11] Y. Zhang, D. Zhou, S. Chen, S. Gao e Y. Ma, «Single-image crowd counting via multi-column convolutional neural network,» in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [12] L. Del Pizzo, P. Foggia, A. Greco, G. Percannella e M. Vento, «Counting people by RGB or depth overhead cameras,» *Pattern Recognition Letters*, vol. 81, pp. 41-50, 1 10 2016.
- [13] Terabee, «Time-of-Flight principle: Technologies and advantages - Terabee,» [Online]. Available: <https://www.terabee.com/time-of-flight-principle/>.
- [14] L. Hikvision Digital Technology Co, «IP Camera | Network Camera - Hikvision,» [Online]. Available: <https://www.hikvision.com/en/Products/Network-Camera>.
- [15] VIVOTEK Inc., «People Counting :: VIVOTEK ::,» [Online]. Available: <https://www.vivotek.com/solutions/people-counting>.

- [16] DVSLTD, «Hikvision People Counting Camera Set Up - YouTube,» [Online]. Available: <https://www.youtube.com/watch?v=zAK90GxVn5s>.
- [17] H. Wu, C. Gao, Y. Cui e R. Wang, «Multipoint infrared laser-based detection and tracking for people counting,» *Neural Computing and Applications*, vol. 29, n. 5, pp. 1405-1416, 1 3 2018.
- [18] STMicroelectronics, «Time of Flight (ToF) Sensors - STMicroelectronics,» [Online]. Available: <https://www.st.com/en/imaging-and-photonics-solutions/proximity-sensors.html?querycriteria=productId=SC1934#>.
- [19] STMicroelectronics, «People Counting Using a Single ST Time-of-Flight Sensor - YouTube,» [Online]. Available: <https://www.youtube.com/watch?v=c91Ve-g0J2U>.
- [20] BEA, «LZR®-SIGMA | BEA Europe,» [Online]. Available: <https://www.bea-sensors.com/it/prodotto/lzr-sigma/>.
- [21] A. E. Redondi e M. Cesana, «Building up knowledge through passive WiFi probes,» *Computer Communications*, vol. 117, pp. 1-12, 1 2 2018.
- [22] X. Tang, B. Xiao e K. Li, «Indoor Crowd Density Estimation Through Mobile Smartphone Wi-Fi Probes,» *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 25 4 2018.
- [23] Y. Fukuzaki, N. Nishio, M. Mochizuki e K. Murao, «A pedestrian flow analysis system using Wi-Fi packet sensors to a real environment,» 2014.
- [24] J. Weppner, B. Bischke e P. Lukowicz, «Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface,» 2016.
- [25] A. Kurkcu e K. Ozbay, «Estimating Pedestrian Densities, Wait Times, and Flows with Wi-Fi and Bluetooth Sensors,» *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2644, n. 1, pp. 72-82, 7 7 2017.
- [26] A. Basalamah, «Crowd Mobility Analysis using WiFi Sniffers,» *International Journal of Advanced Computer Science and Applications*, vol. 7, n. 12, 4 1 2017.
- [27] Hak5, «WiFi Pineapple - Hak5,» [Online]. Available: <https://shop.hak5.org/products/wifi-pineapple>.
- [28] Raspberry Pi Foundation, «Buy a Pi – Raspberry Pi,» [Online]. Available: <https://www.raspberrypi.org/products/>.
- [29] «Home | Sail Amsterdam,» [Online]. Available: <https://www.sail.nl/?lang=en>.
- [30] W. Daamen, Y. Yuan, D. Duives e S. Hoogendoorn, «Comparing three types of real-time data collection techniques: Counting cameras, Wi-Fi sensors and GPS trackers,» *Proceedings of Pedestrian and Evacuation Dynamics*, pp. 568-574, 2016.
- [31] Y. Yuan, W. Daamen, D. Duives e S. Hoogendoorn, «Comparison of three algorithms for real-Time pedestrian state estimation - Supporting a monitoring dashboard for large-scale events,» in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2016.

- [32] E. Boni e E. Piccin, «Risultati della sperimentazione dei sistemi di monitoraggio dei flussi pedonali,» 14 Settembre 2018. [Online]. Available: [https://www.venis.it/sites/www.venis.it/files/redazione/Trasparenza/762\\_AT\\_180925\\_all1.pdf](https://www.venis.it/sites/www.venis.it/files/redazione/Trasparenza/762_AT_180925_all1.pdf).
- [33] C. Matte, M. Cunche, F. Rousseau e M. Vanhoef, «Defeating MAC Address Randomization Through Timing Attacks,» in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks - WiSec '16*, New York, New York, USA, 2016.
- [34] G. Ceccoli, C. Leoni e U. Quartucci, «Relazione progetto Wi-Fi sniffing Cinque Terre,» Luglio 2018. [Online]. Available: [https://drive.google.com/file/d/1uJ-T7iVB9G1NUSr3NJQTa27\\_o4kqakIH/view](https://drive.google.com/file/d/1uJ-T7iVB9G1NUSr3NJQTa27_o4kqakIH/view).
- [35] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye e D. Brown, «A Study of MAC Address Randomization in Mobile Devices and When it Fails,» 2017.
- [36] M. Vanhoef, C. Matte, M. Cunche, L. Cardoso, F. Piessens, L. S. Cardoso, †. Iminds-Distrinet e K. U. Leuven, «Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms An Analysis of Wi-Fi Network Discovery Mechanisms,» 2016.
- [37] D. Gentry e A. Pennarun, «Passive Taxonomy of Wifi Clients using MLME Frame Contents».
- [38] remot3.it Inc, «remote.it - Virtual Private Internet,» [Online]. Available: <https://remote.it/>.
- [39] Wikipedia.org, «crontab - Wikipedia,» [Online]. Available: <https://it.wikipedia.org/wiki/Crontab>.
- [40] Envato Tuts+, «Scheduling Tasks with Cron Jobs,» [Online]. Available: <https://code.tutsplus.com/tutorials/scheduling-tasks-with-cron-jobs--net-8800>.