POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

Progetto implementazione di un'architettura pervasiva per la raccolta di dati analitici relativi all'utilizzo dei mezzi pubblici

Relatore
prof. Giovanni Malnati

Candidato
Francesco Lombardo
matricola: 224287

Luglio 2018

Sommario

Negli ultimi anni, l'implementazione di sistemi in grado di tracciare gli spostamenti di flussi di persone attraverso i loro dispositivi wireless ha riscosso un enorme interesse sia dei ricercatori, che delle aziende. Informazioni del genere, infatti, possono rivelarsi particolarmente utili in diversi campi, tra cui, quello del trasporto pubblico.

Le peculiarità della tecnologia Wi-Fi possono essere sfruttate al fine di poter implementare un servizio che sia in grado di determinare il numero di passeggeri di un veicolo di trasporto pubblico, in maniera del tutto non-invasiva. Questo è possibile attraverso l'analisi di una specifica tipologia di messaggi, detti *probe request*, trasmessi all'interno di una rete Wi-Fi, da un qualsiasi dispositivo wireless, per implementare determinate funzionalità di management.

Sulla base di tale principio, in questo lavoro di tesi è stato progettato, sviluppato e messo in campo un sistema in grado collezionare e, successivamente, analizzare tale tipologia di messaggi.

In circa un mese di esercizio in due distinte località, il sistema ha accolto oltre un milione di record riferibili a circa 35.000 dispositivi differenti, cui si aggiungono 120.000 ulteriori dispositivi il cui indirizzo viene mascherato dal sistema operativo, come tecnica atta a garantire la privacy degli utenti. Tali dati, opportunamente filtrati, sia sulla dimensione spaziale che su quella temporale, permettono di identificare circa 10.000 corse diverse, evidenziandone il punto di salita e di discesa del passeggero. Anche se il sistema è ancora in uno stato prototipale e richiede ulteriori raffinamenti, è possibile affermare che i risultati attualmente riscontrabili, costituiscono un dato di estremo valore per l'azienda di trasporto, mettendo a disposizione informazioni analitiche su vasta scala a costi di acquisizione irrisori, senza richiedere il coinvolgimento attivo nè di utenti, nè di operatori.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Al giorno d'oggi viviamo in un mondo dominato sempre più dalla tecnologia tanto che ormai questa costituisce una parte integrante delle nostre vite. In aggiunta, il trend degli ultimi anni è stato quello di trasformare tutti quei dispositivi presenti nelle abitazioni, che normalmente vengono utilizzati nella vita quotidiana, rendendoli dei dispositivi sempre più *smart*. Tali oggetti sfruttano diversi tipi di tecnologie (RFID, Bluetooth, Wi-Fi, ZigBee, GPS, ecc...) per potersi connettere ad Internet al fine di comunicare i dati raccolti o accedere ad informazioni aggregate per fornire un dato servizio.

Le sveglie suonano prima in caso di traffico, le scarpe da ginnastica o gli orologi da polso raccolgono informazioni riguardanti i tempi, le velocità e le distanze percorse oppure le automobili, dialogando con l'ambiente circostante, possono facilitare la nostra esperienza di guida facendoci evitare le zone della città più trafficate. In generale, tutti gli oggetti possono acquisire un ruolo attivo grazie alla loro interconnessione in rete permettendo di rendere la nostra vita migliore, più semplice e più produttiva.

Questo concetto è stato poi esteso non solo alla singola abitazione ma più in generale ad un'intera città, che identifichiamo con il nome di *smart city*, all'interno della quale vengono utilizzati dispositivi intelligenti per fare in modo che l'uso di beni come, ad esempio, l'acqua o l'elettricità oppure la fruizione di servizi come il trasporto pubblico o la gestione del traffico, siano sempre più efficienti. Tuttavia, lo sviluppo di questi sistemi ha come grosso ostacolo quello di dover affrontare un investimento iniziale per poter realizzare l'infrastruttura generale e soprattutto per poterla integrare con i sistemi preesistenti che non tutti i Paesi, o le città, possono permettersi di sostenere.

In questo scenario, risulta necessario ricercare delle soluzioni alternative che facciano uso delle infrastrutture esistenti e che si vadano ad incorporare ad esse nel modo meno invasivo possibile, minimizzando al contempo i costi implementativi.

1.1 Possibili scenari

La possibilità di identificare automaticamente il numero totale di persone in uno specifico ambiente è un campo di particolare interesse per un serie di diverse applicazioni quali:

Congestione stradale Nelle grandi città questo costituisce un problema soprattutto nelle ore di punta. Molte persone, infatti, sono solite percorrere sempre lo stesso percorso. Conoscendo quali strade sono maggiormente congestionate, è possibile consigliare al guidatore una strada alternativa da percorrere per evitare di rimanere imbottigliati nel traffico.

Centri commerciali Il possessore di una particolare attività commerciale potrebbe essere interessato a conoscere quante persone e con quale frequenza visitano il suo negozio, il tempo medio che spendono al suo interno oppure chi lo visita regolarmente. Queste informazioni potrebbero essere utilizzate, ad esempio, per creare una distribuzione del numero di clienti durante il giorno in modo da ottimizzare la disponibilità degli impiegati durante le ore di punta oppure per proporre offerte personalizzate in base alle abitudini dei singoli clienti.

Attività sociali Ogni città è caratterizzata da diverse zone in cui è possibile trovare una maggiore concentrazione di persone soprattutto in determinate fasce orarie. In questo caso le informazioni riguardanti la densità delle persone in una determinata area geografica potrebbero essere usate, ad esempio, per costruire una mappa real-time delle zone maggiormente frequentate della città. In questo modo, i turisti o coloro che si sono trasferiti da poco in città potranno scoprire quali sono i posti più popolari e dove tipicamente le persone del luogo passano il loro tempo libero.

Trasporto pubblico I mezzi pubblici sono un'alternativa efficiente per muoversi in città rispetto a prendere la propria automobile poiché permettono di ridurre l'inquinamento atmosferico e di trasportare contemporaneamente più persone. Purtroppo però, il più delle volte i mezzi pubblici sono sotto-utilizzati. Stimare il numero di viaggiatori su di un autobus o un treno di una compagnia di trasporti può essere utile per determinare l'efficienza di un determinato veicolo in termini di utilizzo oppure quale siano le tratte più frequentemente utilizzate. Queste informazioni potrebbero essere usate dalla compagnia di trasporti al fine di redistribuire i loro mezzi per coprire più efficientemente tutte le tratte a seconda delle necessità. Un esempio potrebbe essere quello di diminuire il numero di bus per una rotta tipicamente poco utilizzata e aumentarli per una tratta che invece presenta una maggiore concentrazione di utenti.

Il grado di accuratezza che possiamo ottenere cercando di definire la densità della folla in un certo ambiente varierà a seconda delle tecnologie che è possibile adottare

tra cui, ad esempio, le più note sono: GPS, sensori a infrarossi/termici, analisi delle immagini, Wi-Fi o Bluetooth. Ciascuna di queste tecnologie comporta una serie di *benefici* e di *controindicazioni* che dovranno essere tenute in considerazione.

1.2 Lavori correlati

Nel contesto del trasporto pubblico, sono state implementate una serie di soluzioni con lo scopo di soddisfare la necessità delle compagnie di trasporto di poter analizzare e monitorare il tasso di utilizzo dei loro mezzi.

In particolare, in questa sezione verranno descritte e confrontate diverse tecnologie esistenti, sviluppate al fine di contare automaticamente il numero di passeggeri su mezzi quali autobus o treni.

1.2.1 Definizione del contesto

La necessità di definire il numero di viaggiatori presenti su un mezzo di trasporto pubblico deriva principalmente dall'interesse della compagnia di trasporti ad effettuare delle indagini finalizzate ad elaborare una serie di statistiche riguardanti l'uso dei propri mezzi. Questi dati rappresentano una base per definire e sviluppare le strategie di programmazione, gestione e pianificazione, oltre che ad essere fondamentali per un'equa ripartizione delle risorse finanziarie dell'azienda.

Tradizionalmente, tale necessità veniva soddisfatta tramite dei controlli di tipo *manuale* effettuati in maniera saltuaria, le quali, anche se in grado di assicurare un buon livello di affidabilità (tanto da essere utilizzate come riferimento nello studio dell'accuratezza dei sistemi di rilevamento automatico), sono inevitabilmente dipendenti dalle prestazioni dell'operatore che sta effettuando le misurazioni, che potrebbe essere influenzato dalla stanchezza, dalla distrazione causata dal tipo di operazione ripetitiva e da una serie di altri fattori, fornendo, in questo modo, dei dati limitati o di parziale utilità.

D'altro canto, negli ultimi anni sono stati sviluppati dei sistemi automatizzati, chiamati *APC (Automatic Passenger Counting)*, i quali hanno riscosso un grande interesse poiché sono in grado di svolgere il medesimo compito con maggiore efficienza a cui, però, corrisponde un minor grado di accuratezza. Nonostante, esistano nel mercato diversi tipi di tecnologie con tale scopo, queste, non essendo ancora pienamente mature, sono caratterizzate da problematiche comuni; infatti, non esiste attualmente una soluzione migliore delle altre. Ne segue che, l'adozione di una di queste metodologie dovrà essere analizzata nel dettaglio prima di essere applicata su di un reale sistema di trasporto pubblico.

1.2.2 Panoramica delle tecniche per i sistemi automatizzati

Le due principali metodologie per contare il numero di persone, che viaggiano su di un veicolo di una compagnia di trasporto, sono:

1. *Conteggio non legato al biglietto:* in questo caso, sono possibili due tecniche di monitoraggio:
   (a) in maniera *diretta*, tramite il monitoraggio del singolo passeggero sfruttando tecnologie installate a bordo del veicolo;
   (b) in maniera *indiretta*, monitorando l'intero carico del veicolo attraverso tecnologie applicate direttamente sulle sospensioni, sotto l'abitacolo o sull'infrastruttura della vettura.
2. *Conteggio legato al biglietto:* come nel caso di metropolitane o ferrovie, dove si fa affidamento all'utilizzo di tornelli, i quali permettono un controllo accurato del flusso in entrata e uscita, al fine di determinare il numero totale di persone presenti in un dato istante. Tale soluzione è ritenuta valida solo nel caso di impianti fissi con accessi riservati, mentre risulta essere di difficile applicazione per mezzi su gomma o ferroviari. La soluzione sopracitata rientra nella categoria delle soluzioni identificate con la sigla *ERF (Electronic Registering Fareboxes)*.

In questo contesto, in cui nessuna tecnologia è caratterizzata da specifici pregi che la rendono maggiormente preferibile rispetto alle altre, assume un ruolo fondamentale il *costo* necessario all'implementazione del sistema, che comprenderà la combinazione di hardware e software necessari affinché il sistema possa funzionare correttamente. Qualunque sia il metodo utilizzato, l'obiettivo è quello di ottenere un conteggio affidabile e preciso o comunque che presenti un margine d'errore irrisorio rispetto al numero esatto di persone presenti sul veicolo.

Di seguito, vengono analizzati nel dettaglio le tecniche maggiormente utilizzate in commercio [15] per risolvere il problema del conteggio dei passeggeri che utilizzano i mezzi di trasporto.

1.2.3 Sensori a infrarossi

La tecnologia ad infrarossi è una delle metodologie più collaudate per la tipologia di problema analizzata ed è, infatti, la soluzione più facilmente reperibile in commercio, dato che molti produttori di tecnologie APC si basano sull'impiego di sensori ad infrarossi.

I rilevatori ad infrarossi possono essere suddivisi in due categorie:

- sensori di *tipo attivo*, composti da un trasmettitore e da un ricevitore, che installati alla stessa altezza, uno di fronte all'altro, creano un fascio continuo di radiazioni.
- sensori di *tipo passivo*, detti anche *PIR (Passive InfraRed)*, che misurano i raggi infrarossi (IR) irradiati dagli oggetti che si trovano all'interno del loro campo visivo.

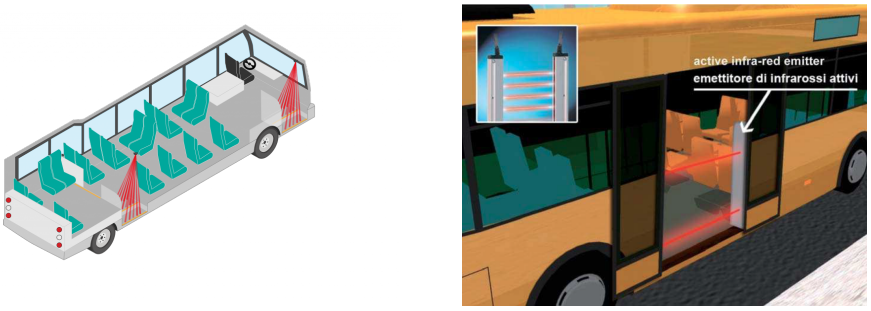
Per fare in modo che il conteggio dei passeggeri venga effettuato in maniera indipendente dal biglietto di viaggio, si fà, solitamente, uso di sensori a infrarossi in grado di rilevare il passaggio di persone attraverso dei varchi, come le porte degli autobus, dei tram o dei treni.

Ovviamente, tali sistemi presentano un certo margine d'errore per quanto riguarda l'accuratezza nell'effettuare il conteggio dei passeggeri, ma questo risulta essere contenuto entro il 5-10% rispetto al valore reale del numero totale di persone presenti.

Sensori a infrarossi attivi

I sensori di tipo attivo sono composti da un trasmettitore e da un ricevitore, indipendenti fra di loro. In questo contesto, vengono utilizzati dei LED che fungono da componenti attive, ovvero da *emettitori* ad infrarossi, posti in prossimità delle entrate del mezzo di trasporto; quando questi vengono attivati, emetteranno un raggio che attraverserà lateralmente o verticalmente il varco d'ingresso, fino a raggiungere un ricevitore. Compito di un *ricevitore* ad infrarossi è quello di verificare la continuità del fascio creato dall'emettitore.

Nel momento in cui si verifica l'interruzione del raggio ad infrarossi, sia esso dovuto a dei movimenti in entrata o in uscita dal veicolo, il ricevitore, non riscontrando più la presenza di questo, considererà che si è verificato il passaggio di una persona.



(a) Verticali

(b) Orizzontali

Figura 1.1: A sinistra vengono mostrati dei sensori IR verticali, mentre a destra vengono mostrati dei sensori IR orizzontali.

I sensori IR attivi, a seconda di come vengono installati sul mezzo di trasporto, possono appartenere a due categorie:

- Sensori orizzontali o "a barriera";
- Sensori verticali o "a tenda";

In entrambi i casi, il punto debole di questa soluzione è la necessità di installare più di un sensore per porta, poiché, altrimenti, non sarebbe possibile rilevare il numero esatto di passeggeri; questo comporta principalmente l'incremento dei costi di realizzazione.

Pertanto, i sensori a infrarossi attivi, seppur di facile applicazione, sono, quindi, raramente utilizzati per il conteggio dei passeggeri sui mezzi pubblici ma, tipicamente, vengono adottati per altre tipologie di applicazioni come, ad esempio, l'individuazione di intrusi in edifici privati e pubblici.

Sensori a infrarossi passivi

Figura 1.2: Esempio di sensore ad infrarossi passivo

L'impiego di sensori passivi comporta la generazione di un raggio meno puntuale rispetto a quello creato da un LED di un sensore IR attivo. In particolare, vengono utilizzati dei sensori che sono in grado di reagire alla radiazione infrarossa emessa da un corpo, ovvero permette di identificare le repentine variazioni di temperatura all'interno di un determinato raggio di azione.

L'intrusione all'interno dell'area (sensibile alle variazioni termiche) creata dal sensore, comporta un improvviso aumento della temperatura che permette di identificare tale evento.

Questo tipo di soluzioni sono spesso utilizzate al fine di realizzare dei rilevatori di movimento in ambienti interni o esterni di grandi dimensioni, che sono utili per il corretto funzionamento di un sistema di sorveglianza. È possibile, però, adattarli anche per applicazioni finalizzate al conteggio automatico, ma bisogna fare attenzione al corretto posizionamento e alla calibrazione del sensore utilizzato, poiché altrimenti, ciò comporterebbe un conteggio errato.

Inoltre, anche in questo caso, risulta necessario ricorrere a più di un sensore per porta per poter rilevare correttamente il flusso di persone in ingresso o in uscita dall'abitacolo, cosa che comporta una conseguente crescita dei costi implementativi.

Sensori a infrarossi attivi e passivi

Al fine di ottenere un maggior livello di precisione riguardo il conteggio, è possibile utilizzare le due tecniche, viste precedentemente, in maniera congiunta, beneficiando, così, dei vantaggi offerti dall'utilizzare sia la componente attiva che quella passiva di un sensore ad infrarossi.

La componente *passiva* del sensore avrà lo scopo di rilevare la variazione di calore all'interno di una certa area di interesse. Mentre, la componente *attiva* sarà caratterizzata da un emettitore ed un ricevitore e baserà il suo funzionamento sul principio per cui: il corpo di ogni persona riflette, se pur parzialmente, le radiazioni ad infrarossi che lo colpiscono.

Sulla base di tale proprietà, è possibile incorporare nello stesso componente, oltre al sensore passivo, sia l'emettitore che il ricevitore del sensore attivo, ottenendo notevoli vantaggi in termini di ingombro e complessità di installazione della soluzione.

In questo modo, l'emettitore irradierà luce infrarossa in direzione del pavimento e nel momento in cui una persona attraversa l'area monitorata dal sensore, tale fascio sarà parzialmente riflesso nella direzione opposta, verso il ricevitore.

Per poter riuscire ad effettuare un analisi di un flusso in movimento in un'area monitorata attraverso questa dei sensori del genere, è stato sviluppato il sistema *IRMA (InfraRed Motion Analyzer)*. All'ingresso di una persona all'interno dell'area monitorata dal sensore, attraverso il lavoro congiunto della componente attiva e passiva dello stesso, saranno generati una serie di segnali che, dopo essere stati digitalizzati, saranno poi trasmessi attraverso un'interfaccia seriale ad un *analyzer*, che si occuperà di aggiornare l'attuale valore del contatore.

1.2.4 Sensori a pedana

Una delle altre principali metodologie per contare il numero di passeggeri di un veicolo di trasporto si basa sull'utilizzo di *pedane in lega metallica*, dette anche *treadle switch based*, che sono in grado di deformarsi sotto l'effetto di un carico esterno, come, ad esempio, il peso di un passeggero che le attraversa.

Tali pedane sfruttano il principio degli interruttori a striscia, ovvero un insieme di lame che normalmente sono aperte e, sotto l'azione di una pressione esterna che deforma la piastra, vengono chiuse, fornendo un segnale interpretato dal sistema di controllo come il passaggio di una persona. Le *treadle mats*, come mostrato in Figura 1.3, vengono collocate in prossimità degli ingressi del veicolo, tipicamente

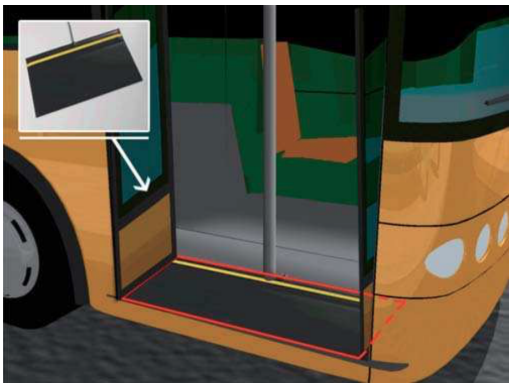


Figura 1.3: Esempio di installazione del sensore a pedana su di un veicolo di trasporto pubblico.

sui gradini d'accesso. Al fine di diminuire il numero di errori dovute a false rilevazioni, come quelle ottenute appoggiando una borsa o una stampella sul sensore, diverse case produttrici di questo sistema APC hanno implementato una serie di contromisure atte a risolvere il problema; ad esempio, tenendo scartando tutti gli input che vengono trasmessi mentre che il veicolo è in movimento.

L'accuratezza che possiamo ottenere attraverso questo particolare tipo di sensori è molto alta (superiore al 95%), ma sono caratterizzati da un forte *limite* che ne diminuisce l'estensione del campo applicativo; questi, infatti, necessitano di essere installati su dei veicoli che presentano degli ingressi rialzati con più di un gradino, poiché la presenza di uno solo di essi, non permetterebbe di distinguere tra ingresso o uscita di un passeggero, rendendo, di fatto, inadeguato il sistema di conteggio stesso.

Per questa ragione, i sensori a pedana risultano essere inadeguati per essere utilizzati su dei bus di linea cittadini, in quanto questi sono tipicamente caratterizzati da un singolo pianale ribassato, mentre sono molto più diffusi sui treni o sui tram.

Per quanto riguarda il costo di questi sensori, questo è generalmente più basso rispetto a quello dei sensori attivi/passivi, inoltre risulta essere irrisorio se paragonato al costo dell'interno veicolo di trasporto.

1.2.5 Altre tecniche

In letteratura, esistono molte altre tecniche che sono state successivamente adottate da alcune compagnie di trasporto pubblico al fine di riuscire a stimare il numero di passeggeri che fanno uso dei loro mezzi.

Alcuni, ad esempio, sfruttano le funzionalità delle videocamere che, attraverso una fase intermedia di elaborazione dello streaming video, sono in grado di fornire in output un risultato con una precisione superiore all'85% nel caso di un match

esatto [16]. Le attuali implementazioni sono capaci di rilevare le sagome dei passeggeri, permettendone sia una classificazione, in modo da distinguerle da quelle di borse o animali, nonché la loro direzione di spostamento, in modo da discriminare i passeggeri in ingresso sul veicolo da quelli in uscita dallo stesso. Tuttavia, gli svantaggi sono molteplici: il costo implementativo è alto, sono presenti dei problemi di occlusione visiva (non potendo ottenere dei buoni risultati in ambienti particolarmente affollati) e l'elaborazione è computazionalmente onerosa.

Altri, invece, non impiegano alcun tipo di sensore e basano la loro analisi su altri fattori. In particolare, alcuni fanno un forte affidamento sulla bontà dei viaggiatori che dovranno validare il loro titolo di viaggio, attraverso delle apposite apparecchiature, sia all'entrata che all'uscita dal mezzo. A differenza delle altre, questa tecnica è in grado di fornire soltanto una *stima grossolana* del flusso di viaggiatori, senza pretendere un particolare livello di precisione dei dati raccolti.

Un altro approccio, è quello di sfruttare i dispositivi dotati di tecnologia Bluetooth. Ogni dispositivo Bluetooth manda periodicamente dei messaggi, detti *discovery requests*, al fine di scoprire quali altri dispositivi si trovano nelle sue vicinanze. In [7], su degli autobus di un'agenzia di trasporto sono stati posizionati dei dispositivi in modalità *scanner* con lo scopo di raccogliere tutti i discovery request e costruire una mappa dei dispositivi attivi in un dato istante temporale. [20] presenta un approccio analogo, ma lo scopo è quello di fornire una stima del numero di partecipanti ad una partita di un campionato europeo di calcio.

Infine, la tecnologia che negli ultimi anni ha riscosso maggiore interesse è sicuramente il Wi-Fi che, essendo la principale soluzione per le comunicazioni a medio raggio, oggigiorno è incorporata in un numero sempre crescente di dispositivi *smart*, tra cui soprattutto smartphone e tablet. La loro diffusione è diventata così capillare, che sempre più persone posseggono almeno uno di questi dispositivi e li portano con loro ovunque vadano; ciò offre nuove possibilità per una serie di scopi quale la localizzazione e il tracciamento di individui oppure la stima della densità di una folla, senza la necessità di dover installare nessun costoso componente hardware aggiuntivo. Anche in questo caso, come abbiamo visto precedentemente per la tecnologia Bluetooth, i dispositivi Wi-Fi periodicamente mandano in rete dei frame che saranno catturati da dei nodi, opportunamente installati all'interno dell'aria di interesse, al fine di estrarre informazioni utili per una successiva analisi quali, ad esempio il MAC address[1] del dispositivo stesso [5, 17, 21].

[1]Indirizzo univoco assegnato dal produttore della scheda di rete

1.3 Definizione del problema

Scopo principale di questo lavoro di tesi è stato quello di implementare una soluzione in grado di stimare il numero di passeggeri a bordo di un mezzo di trasporto pubblico. La tecnologia Wi-Fi è stata reputata la più adeguata per realizzare questo obiettivo poiché permette di avere una serie di benefici di cui non potremmo usufruire utilizzando le altre tecnologie a disposizione. In particolare, siamo in grado di ottenere:

- *Basso costo:* L'hardware necessario è costituito principalmente da un microcontroller dotato di un interfaccia Wi-Fi che permette di mantenere il costo implementativo molto basso a differenza, per esempio, di soluzioni basate sull'analisi delle immagini.

- *Automazione:* Non viene richiesta nessuna collaborazione da parte degli utilizzatori del sistema perché questo possa funzionare correttamente. L'utente, infatti, in questo scenario assume un ruolo completamente *passivo* rispetto ad altre soluzioni, come, ad esempio, quelle basate sull'uso del GPS, RFID o NFC, che, invece, richiedono all'utente l'installazione di un applicativo ad-hoc (nel caso del GPS) o l'interazione con un particolare sistema (nel caso di RFID o NFC).

- *Scalabilità:* Il sistema non richiede particolari dipendenze per poter funzionare, se non la presenza di un AP[2] in mdo da poter accedere alla rete. Questo rende la soluzione in grado di scalare anche su un numero elevato di mezzi.

- *Ampia copertura e data rate[3]elevato:* A differenza del Bluetooth, il cui raggio di azione è di approssimativamente di un metro e caratterizzato da un basso data rate, il grosso vantaggio del Wi-Fi è quello di essere caratterizzato da una più ampia copertura (circa 100 metri all'aria aperta che scendono a 20 metri se sono presenti degli elementi che ne disturbano il segnale) e un elevato data rate (circa 11 Mbps); ciò ha delle ripercussioni soprattutto per quanto riguarda i costi implementativi, poiché con un tale raggio d'azione, a seconda dell'estensione del veicolo, saranno necessari al massimo un paio di microcontroller opportunamente posti in maniera equidistante tra di loro.

- *Analisi del comportamento degli utenti:* Dato che ogni scheda di rete Wi-Fi, installata in molti dispositivi smart - come gli smartphone, è identificata univocamente tramite un indirizzo MAC, è possibile tracciare e distinguere

[2] AP: Access Point

[3] Quantità di dati che possono essere trasferiti attraverso un canale di comunicazione in un dato intervallo di tempo

il comportamento dei possessori di tali dispositivi analizzando il flusso dei messaggi di controllo scambiati in rete.

- *Privacy degli utenti:* La riservatezza delle informazioni scambiate dagli utenti tramite dei dispositivi wireless viene garantita cifrando il contenuto del pacchetto scambiato in rete. Inoltre, sono state introdotte delle funzionalità che permettono ad un dispositivo di camuffarsi, non rivelando la propria identità in rete; in questo modo, il tracciamento degli utenti e delle loro abitudini potrà risultare più complesso.

Capitolo 2

La tecnologia Wi-Fi

In questo capitolo, vengono introdotti i concetti basilari inerenti il funzionamento di una rete Wi-Fi. Successivamente, viene analizzato nel dettaglio un particolare tipo di frame 802.11, specificando il motivo per cui è stato ritenuto utile ai fini della tesi, quali sono le possibili implicazioni riguardo alla privacy degli utenti e come questa viene preservata.

2.1 Concetti base

Il Wi-Fi è una tecnologia sviluppata al fine di permettere a dei dispositivi, basati sugli standard IEEE 802.11, di comunicare tra loro in una *rete locale senza fili* (WLAN).

2.1.1 Standardizzazione

Lo IEEE, acronimo di *Institute of Electrical and Electonic Engineers*, è un'organizzazione internazionale di scienziati professionisti, con l'obiettivo di promuovere la tecnologia per il bene dell'umanità. IEEE è nota per lavorare attivamente sulla realizzazione di standard tecnologici, organizzati in *progetti*, ciascuno del quale è caratterizzato da un numero per poter essere identificato. Un progetto rappresenta un problema generale da dover affrontare ed è tipicamente suddiviso in diverse unità operative, dette *working group*, che hanno il compito di occuparsi di un particolare aspetto del problema. Ogni working group viene anch'esso identificato attraverso un numero, il quale sarà associato al corrispondente progetto in base al seguente formato XX.X, dove XX rappresenta il numero del progetto, mentre X rappresenta il numero dello specifico working group.

Il progetto IEEE 802 definisce un'ampia famiglia di standard per reti locali (LAN) e per reti metropolitane (MAN), che si focalizzano sui primi due livelli della pila ISO/OSI, ovvero il livello fisico e data link: mentre il primo ha l'onere di

conoscere i dettagli tecnici attraverso cui poter trasmettere e ricevere informazioni, il secondo sarà caratterizzato da una serie di regole che specificano come accedere al mezzo di trasmissione, indipendentemente dal livello fisico sottostante.

IEEE 802.11 definisce un insieme di standard per le reti WLAN che si focalizzano principalmente sulla trasmissione delle informazioni e sugli aspetti legati alla sicurezza.

2.2 Architettura

2.2.1 Nomenclatura

Le reti 802.11 sono costituite da quattro principali componenti, che sono riassunte nella Figura 2.1.

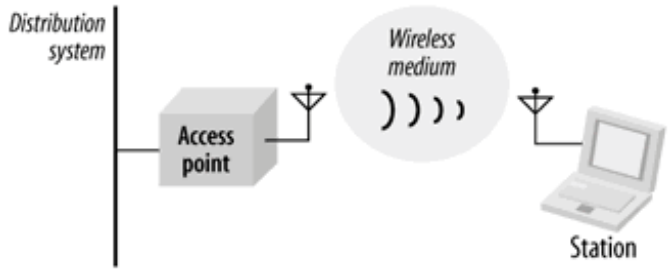


Figura 2.1: I principali componenti di una WLAN

Stazioni Esistono due tipi di stazioni wireless: gli access points (APs) e i client. Gli APs fungono da ponte tra i dispositivi wireless e la rete cablata, la quale permette l'accesso alla rete. I client sono costituiti da tutte quelle stazioni, mobili e non, caratterizzate da una scheda di rete wireless come pc, notebooks o smartphones.

Wireless medium Lo standard 802.11 prevede che le stazioni comunichino usando l'etere come mezzo di trasmissione.

Distribution System (DS) È un'infrastruttura che permette a diversi AP di essere interconnessi tra di loro al fine di poter fornire accesso a determinati servizi ad una stazione wireless in movimento. Lo standard 802.11 non specifica nessuna particolare tecnologia per realizzare un DS. Per questa ragione, è possibile realizzare il backbone di rete, che permette la comunicazione tra i diversi AP, sia utilizzando lo standard 802.3 per reti LAN, che tramite l'uso dello standard 802.11 per le reti wireless.

2.2.2 Tipi di reti

Lo standard IEEE 802.11 definisce un *Basic Service Set (BSS)* come l'insieme di una serie di stazioni. Ogni BSS è riconosciuto attraverso un identificativo univoco a 48-bit e da una stringa di caratteri, detta *Service Set IDentifier (SSID)*.

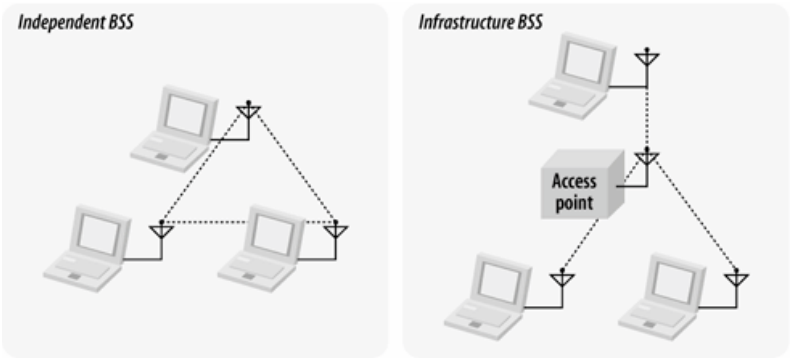


Figura 2.2: A sinistra, un insieme di stazioni che formano una rete ad-hoc, detta *Independent BSS*. A destra, tutte le stazioni, per poter comunicare tra di loro, dovranno prima essere associate ad un Access Point che gestirà tutto il traffico nella rete; questa architettura è detta *Infrastructure BSS*.

Lo standard definisce tre possibili architetture per un BSS:

- *Independent BSS o IBSS:* identifica una rete ad-hoc che rende possibile collegare in modo indipendente più stazioni wireless tra loro senza la necessità di un dispositivo centrale che funga da tramite. Questa soluzione è sicuramente la più economica, ma non è adatta a una rete composta da un numero elevato di dispositivi a causa della sovrapposizione dei diversi segnali e del conseguente calo di affidabilità. Tipicamente, infatti, un IBSS viene creata tra un piccolo numero di stazioni per soddisfare uno specifico bisogno (come quello di condividere dei documenti) e solo per un breve lasso di tempo; quando lo scopo sarà raggiunto l'IBSS verrà rimosso. Un esempio di questa architettura è mostrato a sinistra della Figura 2.2.
- *Infrastructure BSS o BSS:* a destra della Figura 2.2 viene mostrato un esempio di BSS ad infrastruttura; questo si distingue per la presenza di un AP che funge da terminatore di tutte le comunicazioni, anche quelle tra stazioni appartenenti allo stesso BSS. In questa particolare configurazione, le stazioni dovranno, innanzitutto, avviare una procedura di *associazione* con l'access point al fine di poter usufruire dei servizi offerti dalla rete. Sarà sempre il client che avrà l'onere di iniziare la procedura di associazione, mentre l'AP potrà decidere se garantire oppure negare l'accesso, basandosi sul contenuto della richiesta ricevuta. Lo standard 802.11 non pone alcun limite sul numero

di stazioni mobili che possono essere servite da un AP, ma, tipicamente, le implementazioni definiscono un numero massimo di stazioni al fine di mantenere basso il throughput[1] all'interno della rete.

- *Extended Service Set o ESS:* la copertura offerta da un BSS può andare bene per le abitazioni o i piccoli uffici, ma non è adatta per zone più ampie. Lo standard 802.11 permette di creare reti wireless di dimensioni arbitrarie, collegando insieme più BSS attraverso un backbone di rete (ovvero il DS) al fine di creare una rete logica estesa, detta appunto ESS; inoltre, non fornisce alcuna specifica riguardo alla tecnologia da dover utilizzare per la realizzazione del backbone di rete. La Figura 2.3 mostra che l'unione dei quattro BSS forma un ESS (posto che gli AP che creano ogni singolo BSS siano configurati per appartenere tutti allo stesso ESS). Le stazioni, all'interno di uno stesso ESS, potranno comunicare tra di loro, qualsiasi sia il loro BSS di appartenenza ed, inoltre, la comunicazione non verrà interrotta anche se queste si muovono da un BSS ad un altro. Infatti, per fare in modo di assicurare una costante connessione ai nodi mobili, che per definizione possono muoversi da un BSS ad un altro, è necessario che il BSS creato da un certo AP si vada ad intersecare con tutti gli altri nelle sue vicinanze. Nell'esempio mostrato in Figura 2.3, un nodo mobile che vuole spostarsi dal BSS2 al BSS1, dovrà necessariamente seguire un percorso che passi attraverso il BSS3 se non vorrà perdere la sua connettività alla rete. Un ESS rappresenta il più alto livello di astrazione supportato dalle reti 802.11; in questa architettura, il router rimarrà ignaro della specifica posizione di una stazione mobile e sarà compito dell'AP, con cui la stazione è associato, di smistare eventuali frame destinati ad essa.

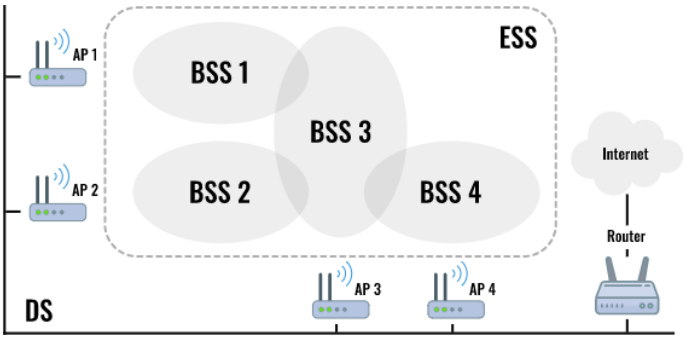


Figura 2.3: I diversi BSS mostrati in Figura sono interconnessi attraverso un backbone di rete, il DS, per formare una rete logica estesa detta *Extended Service Set*.

[1]Il throughput di un canale di comunicazione corrisponde alla capacità di trasmissione del canale effettivamente utilizzata.

2.3 MAC frames

Durante la comunicazione tra due stazioni vengono scambiati una serie di messaggi, detti *MAC frame*. Lo standard definisce diversi tipi di MAC frame, ma questi sono caratterizzati tutti da uno stesso generico formato.

2.3.1 Tipi di frame

Lo standard prevede tre diversi tipi di MAC frame: Data, Management e Control frames. I *Data frames* hanno il compito di trasportare dati applicativi. I *Control frames* vengono usati all'interno del protocollo CSMA/CA, fondamentale per poter regolare l'accesso al mezzo condiviso di due o più stazioni evitando il verificarsi di collisioni. In particolare, il protocollo prevede che una stazione, prima di inviare un frame, si metta in ascolto sul canale per verificare che questo sia effettivamente libero e, se non lo è, la trasmissione verrà tentata nuovamente in un istante successivo. Dopo ogni trasmissione, la stazione aspetta di ricevere un `acknowlegde (ACK) control frame` e se, dopo un certo lasso di tempo, questo non sarà stato ricevuto, allora si assumerà che si è verificata una collisione e si provvederà a ritrasmettere il frame. Infine, i *Management frames* sono indispensabili affinché una stazione possa associarsi ad un AP e autenticarsi presso la rete. Prima di poter accedere ai servizi offerti dalla rete, infatti, una stazione deve mandare un `association request management frame` che contiene informazioni come la velocità di trasmissione supportata dalla scheda di rete Wi-Fi o l'SSID della rete con cui la stazione si vuole associare. Nel caso in cui l'AP accetterà l'associazione, opzionalmente potrà anche richiedere le credenziali di accesso alla rete in modo che la stazione venga autenticata.

2.3.2 Formato di un frame

Il formato di un frame prevede una serie di campi che, a seconda del tipo di frame trasmesso, potranno o cambiare o non essere utilizzati. La figura 2.4 mostra il formato di un generico frame 802.11:

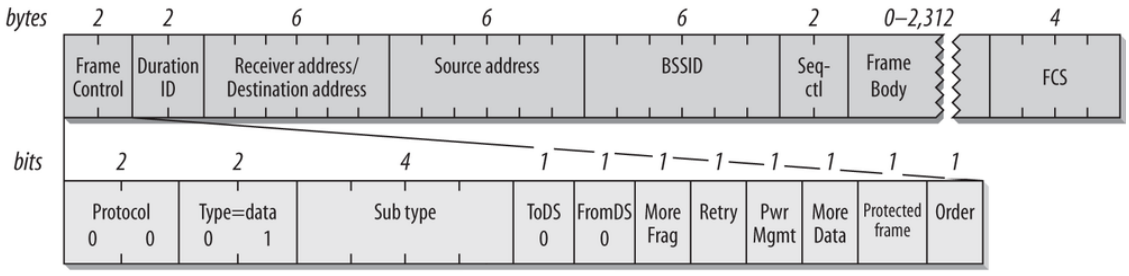


Figura 2.4: Formato di un MAC frame

Ogni frame è caratterizzato da:

- *Header:* contiene una serie di informazioni tra cui l'indirizzo della stazione sorgente (`Source Address`) e di quella di destinazione (`Destination Address`) e si estende dal campo `Frame Control` al campo `Sequence-Control`;
- *Frame Body:* campo di lunghezza variabile il cui contenuto differisce a seconda del tipo di frame;
- *Frame Check Sequence:* contiene un codice che permette di riconoscere eventuali errori di trasmissione che hanno corrotto il contenuto del frame.

Il primo campo dell'header è il `Frame Control` che è lungo 2 bytes ed è caratterizzato da una serie di sotto-campi, tra cui `Type` e `Sub-Type`, che permettono di identificare il particolare tipo di frame usato.

La Tabella 2.1 mostra quali sono i più importanti tipi di frame, specificando a quale macro-categoria questi appartengono: Management, Control o Data frame.

Type	**Sub-Type**
Management	Association request
	Association response
	Beacon
	Probe request
	Probe response
	Disassociation
	Authentication
	Deauthentication
Control	RTS
	CTS
	Acknowledgment (ACK)
Data	Data
	Null data (no data trasmitted)

Tabella 2.1: Sono riportati solo alcuni, dei molteplici sotto-tipi di frame esistenti, raggruppati per tipo.

A partire dal terzo campo dell'header, troviamo il `Destination Address` e il `Source Address`, ciascuno di lunghezza 6 bytes, che rappresentano, rispettivamente, l'indirizzo MAC della stazione di destinazione e di quello della sorgente.

MAC Address È un indirizzo di lunghezza 6 bytes, che viene assegnato in maniera univoca dal produttore ad ogni scheda di rete. In particolare, i primi tre ottetti, detti OUI (Organization Unique Identifier) sono assegnati direttamente

dall'IEEE al costruttore di dispositivi compatibili con lo standard Ethernet, mentre i successivi tre ottetti, detti NIC (Network Interface Controller), sono assegnati dal costruttore stesso, che deve impegnarsi a rispettare il solo vincolo dell'unicità. La Figura 2.5 mostra il dettaglio di come è composto un indirizzo MAC.

Se il bit meno significativo del primo ottetto dell'indirizzo è posto a 0, allora quell'indirizzo rappresenterà una singola stazione. Questo tipo di trasmissione è detta *unicast*. Viceversa, se lo stesso bit è posto ad 1, allora quell'indirizzo rappresenterà un gruppo di stazioni che potranno essere indirizzate tutte allo stesso tempo. In particolare, IEEE ha definito due principali indirizzi appartenenti a quest'ultimo tipo:

- *Broadcast:* è un indirizzo dove tutti i bits sono posti a 1; la rappresentazioni in notazione decimale è FF:FF:FF:FF:FF:FF. Un indirizzo broadcast verrà mandato a tutte le stazioni appartenenti ad una stessa LAN.
- *Multicast:* sarà ricevuto da tutte le stazioni all'interno della stessa LAN che sono state configurate per accettare frame mandati a quel particolare indirizzo.

Inoltre, un indirizzo può essere: *universally administered* o *locally administered*[2]. Questi sono distinti in base al valore del secondo bit meno significativo del primo ottetto dell'indirizzo che nel primo caso sarà posto a 0, mentre nel secondo sarà posto ad 1.

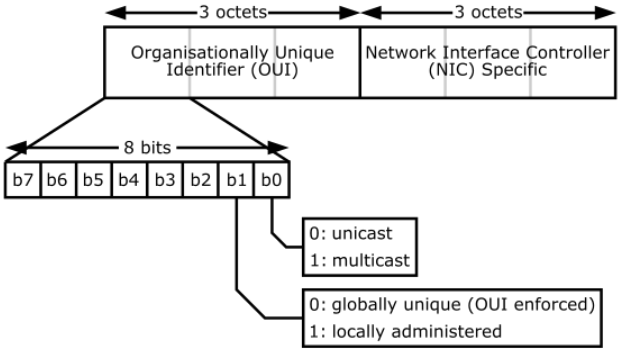


Figura 2.5: Un indirizzo MAC è costituito da 6 ottetti: i primi 3 ottetti identificano l'OUI, mentre i successivi 3 identificano il NIC.

Il campo `BSSID`, sempre lungo 6 bytes, rappresenta l'identificativo univoco associato a ciascun BSS, in modo da poterlo distinguere da un altro BSS appartenente

[2]Nel resto del documento si ci riferirà a queste due categorie di indirizzi anche con i termini "global address" e "local address" oppure "indirizzi globali" e "indirizzi locali".

alla stessa rete. In una infrastructure BSS, questo è costituito dal MAC address dell'interfaccia wireless dell'AP che ha creato quel dato BSS, mentre per una IBSS sarà generato in maniera casuale con lo Universal/Local bit settato ad 1. Un BSSID è detto *broadcast BSSID* se ogni suo byte è posto ad 1. Un BSSID, così costruito, viene usato esclusivamente nel caso in cui una stazione mobile stia cercando di connettersi ad una qualsiasi rete. I `Probe Request/Response management frames` sono i soli tipi di frames incaricati di poter fare uso di un broadcast BSSID.

L'ultimo campo dell'header è il `Sequence-Control`, lungo 2 bytes, che viene usato sia per operazioni di frammentazione che per identificare e scartare frame duplicati; questo consiste di due sotto-campi: `Sequence Number` e `Fragment Number`. Il Sequence Number, lungo 12 bits, contiene un valore progressivo, calcolato da un contatore modulo 4096, che indica il numero di frame attualmente trasmessi. Il contatore inizia da 0 e viene incrementato di 1 ogni volta che un pacchetto di alto livello viene gestito dallo strato data-link. Se un frame è parte di un pacchetto di alto livello frammentato, il valore del Sequence Number rimarrà invariato, ma verrà incrementato il campo Fragment Number. I numeri di sequenza non vengono utilizzati nei control frame; in questo caso, infatti, il campo Sequence-Control non sarà presente.

Il `body` del frame, anche detto payload, trasporta il contenuto informativo che dovrà essere trasferito da una stazione all'altra. In origine, lo standard 802.11 prevedeva una dimensione massima del payload di 2,304 bytes; scelto principalmente per permettere ad un'applicazione di mandare pezzi di informazioni grandi 2,048 bytes. Implementazioni successive, permettono di gestire body di dimensioni superiori in modo da poter inserire header addizionali, utili per specifiche applicazioni come per la sicurezza o per il QoS.

Il `Frame Check Sequence (FCS)` contiene un codice a 4 byte, calcolato attraverso un operazione matematica detta *Cyclic Redundancy Check o CRC* che prende in input sia tutti i campi MAC header che il body del frame. Il valore dell'FCS sarà utilizzato dalle stazioni per verificare l'integrità del frame ricevuto. Tuttavia, nel caso in cui si verificasse un errore durante la trasmissione, l'FCS non potrà fornire alcuna informazioni utile per poter correggere il frame corrotto; si dovrà provvedere, quindi, a richiedere la ritrasmissione del frame in questione.

È importante sottolineare che, anche nel caso in cui il body di un frame venga cifrato per via di meccanismi di sicurezza quali WEP, WPA2, ecc. . . , il corrispondente header sarà sempre trasmesso in chiaro, permettendo, così, di poter accedere alle informazioni contenute al suo interno.

2.4 Management Frames

I Management frame sono un componente essenziale in una rete wireless poiché vengono usati per fornire dei servizi di base, che in una rete cablata sono di più semplice implementazione per via della diversa natura di tale tecnologia.

Per esempio, l'autenticazione di un utente in una rete wired è un'operazione molto semplice: una volta collegato il cavo di rete ad una workstation, lo switch centrale, a cui l'altra estremità del cavo è associato, sarà incaricato di stabilire, tramite delle opportune regole definite dall'amministratore di rete, se l'indirizzo MAC di quella specifica macchina è abilitato o meno a poter accedere ai servizi offerti dalla rete.

Tuttavia, al fine di poter ottenere un comportamento simile in una rete wireless, le funzionalità di management sono una necessità. In particolare, 802.11 definisce tre principali step che una stazione deve seguire prima di poter operare all'interno della rete:

- Innanzitutto, deve essere identificata una rete wireless compatibile; questo, in una rete wired si traduce nel cercare un socket RJ45.
- Successivamente, una fase di *autenticazione* avrà luogo; in particolare, la stazione dovrà fornire una *proof of identity*[3] in modo che possa essere stabilito se questa sia abilitata ad avere accesso ai vari servizi offerti. Nel caso in cui l'autenticazione abbia esito negativo, alla stazione sarà negata la connettività; al contrario, in una rete wired, se l'autenticazione fallisce, la stazione continuerà ad avere connessione a livello fisico, ma non sarà in grado di accedere alle funzionalità della rete poichè la porta dello switch a cui è connessa sarà disabilitata.
- Infine, una volta che la stazione è stata autenticata, seguirà una fase di *associazione* con l'access point più vicino, in modo che la rete possa sapere in che modo instradare i pacchetti destinati a quel particolare nodo.

2.4.1 Struttura di un Management Frame

Un Management frame è uno particolare tipo di MAC frame e, per questa ragione, il suo formato è quello già visto in Figura 2.4.

Tuttavia, possono essere identificate due principali caratteristiche che lo contraddistinguono dalle altre tipologie di frame:

[3] *Una* Proof Of Identity *è un qualsiasi tipo di prova che viene richiesta al fine di verificare l'identità dichiarata da un utente.*

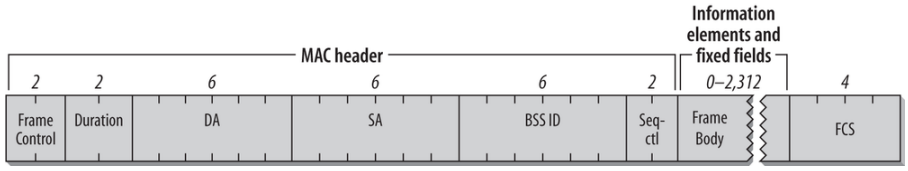


Figura 2.6: Un Management frame è uno specifico tipo di MAC frame, che trasporta una serie di specifiche informazioni, dette Information Elements.

- **BSSID** Il campo *BSSID*, in alcuni management frames, può essere usato al fine di mantenere determinate proprietà all'interno di un singolo BSS. In particolare, per limitare l'effetto dei management frame mandati in broadcast o multicast le stazioni hanno il compito di ispezionare il valore di tale campo e di filtrare tutti quei frame che indicano un BSSID differente rispetto a quello del BSS al quale la stazione è associata.
- **Body** All'interno del body di un management frame, come mostrato in Figura 2.6, sono trasportate una serie di specifiche informazioni, che possono essere di due tipi:
  - *Fixed-length fields:* sono dei campi a lunghezza fissa e che compaiono in un ben noto ordine; proprio per questa ragione, tali campi sono identificati senza la necessità di dover utilizzare un header che li precede.
  - *Variable-length fields:* sono un insieme di campi a lunghezza variabile, anche detti `Information Elements (IEs)`, che rappresentano un insieme di informazioni utilizzate nella comunicazione tra diversi sistemi.

Information Elements Un Information Element è caratterizzato da un formato del tipo Type-Length-Value (TLV), come quello mostrato in Figura 2.7. In par-

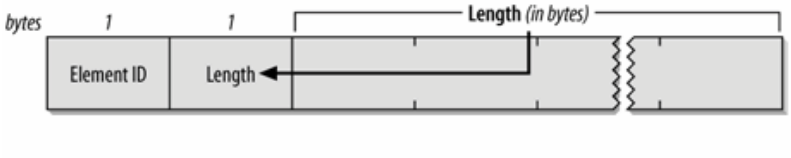


Figura 2.7: Formato generico di un Information Element.

ticolare, ogni blocco di dato sarà caratterizzato dal contenuto informativo vero è proprio, il *Value*, preceduto da due campi: il *Type*, che nel caso di un IE viene chiamato `Element ID`, che specifica il tipo di informazione trasportata e il *Length* che indica la lunghezza effettiva del campo *Value*. Questo tipo di formato conferisce

al Management frame una grande flessibilità, in quanto permette a nuove revisioni dello standard 802.11, di specificare nuovi tipi di Information Elements, senza pregiudicare il funzionamento delle precedenti implementazioni, le quali dovranno semplicemente ignorare tutti gli IE sconosciuti.

Tipi di Information Elements I diversi tipi di Information Elements, ad oggi ufficializzati nello standard, sono mostrati nella Tabella 2.2.

Element ID	**Nome**
0	Service Set Identifier (SSID)
1	Supported Rates
2	FH Parameter Set
3	DS Parameter Set
4	CF Parameter Set
5	Traffic Indication Map (TIM)
6	IBSS Parameter Set
7 (802.11d)	Country
8 (802.11d)	Hopping Pattern Parameters
9 (802.11d)	Hopping Pattern Table
10 (802.11d)	Request
16	Challenge text
32 (802.11h)	Power Constraint
33 (802.11h)	Power Capability
34 (802.11h)	Transmit Power Control (TPC) Request
35 (802.11h)	Transmit Power Control (TPC) Report
36 (802.11h)	Supported Channels
37 (802.11h)	Channel Switch Announcement
38 (802.11h)	Measurement Request
39 (802.11h)	Measurement Report
40 (802.11h)	Quiet
41 (802.11h)	IBSS DFS
42 (802.11g)	ERP information
45	HT Capabilities
48 (802.11i)	Robust Security Network
50 (802.11g)	Extended Supported Rates
107	Internetworking
127	Extended Capabilities
191	VHT Capabilities
221	Vendor Specific

Tabella 2.2: I diversi tipi di Information Elements.

2.4.2 Tipi di Management frames

Come mostrato nella Tabella 2.1, esistono diversi tipi di management frame; ciascuno di questi è definito con lo scopo di implementare una specifica funzionalità di gestione e manutenzione del livello data-link di una rete wireless. Di seguito, vengono analizzati i principali tipi di management frame:

Beacon Sono trasmessi ad intervalli regolari al fine di annunciare l'esistenza di una rete, in modo che le stazioni mobili, analizzando questi particolari frame, possano avere una panoramica delle reti disponibili in una data area. La frequenza con la quale vengono mandati questi frame può differire tra un AP ed un altro, ma lo standard prevede un valore di default pari a 100 ms. In un BSS, l'access point è responsabile di trasmettere dei Beacon, avendo cura di specificare i parametri necessari affinchè una stazione possa accoppiarsi alla quella rete. Mentre, in un IBSS la responsabilità è distribuita a tutte le stazioni che fanno parte della rete.

Authentication Specifica gli algoritmi supportati per la fase di *autenticazione*; questo processo, a secondo dell'algoritmo scelto, può prevedere un diverso numero di passaggi.

Association Request Una volta che una stazione si è autenticata con successo ad una rete, potrebbe richiedere di avviare la fase di *associazione* mandando un `Association Request frame`. Nel body di tale frame, troveremo una serie di IEs tra cui: l'`SSID`, al fine di specificare l'SSID della rete a cui ci si vuole associare e i `Supported Rates`, che specificano le velocità di trasmissione supportate dalla stazione.

Association Response Prima che un access point risponda ad una richiesta di associazione da parte di una stazione, dovrà verificare che i valori specificati all'interno di un *Association Request* corrispondano con i parametri della rete; in caso la verifica abbia esito positivo, l'AP provvederà a rispondere con un `Association Response frame` caratterizzato sia da dei campi fixed-length, tra cui un `Association ID` che identificherà l'associazione appena stabilita, che da una serie di IEs, che specificano i parametri che la stazione dovrà usare per quella particolare associazione.

Probe Request Sono utilizzati da una stazione per effettuare una scansione di un'area di interesse, al fine di effettuare una ricerca riguardo l'esistenza di reti wireless.

Probe Response Nel caso in cui un AP ha ricevuto un `Probe Request frame` con dei parametri compatibili con quelli della rete, provvederà a rispondere alla specifica stazione con un `Probe Response frame` che conterrà gli stessi parametri inseriti all'interno di un `Beacon`. In questo modo, la stazione potrà decidere di avviare la procedura autenticazione/associazione, necessaria per connettersi alla rete.

2.4.3 Probe Request management frames

Tra i diversi tipi di management frame precedentemente analizzati, sono state studiate nel dettaglio le funzionalità di un particolare tipo di frame: il `Probe Request`.

I Probe Request sono particolarmente utili al fine di implementare dei meccanismi di *discovery*, ovvero dei servizi che permettono di verificare la presenza di reti wireless in una determinata area. In particolare, una stazione che vuole conoscere, in maniera puntuale, quali reti sono presenti nelle sue vicinanze, manderà un Probe Request e imposterà, per mezzo di un *timer*, un tempo limite entro il quale ricevere delle eventuali risposte; successivamente, tutti gli APs che riceveranno tale frame, risponderanno alla stazione inviando un Probe Response. Quando il timer scadrà, il nodo analizzerà tutte le *response* ricevute, conoscendo, così, quante e quali tipi di reti sono presenti nelle sue vicinanze. Questo tipo di servizio è anche detto *active service discovery*, poiché compito della stazione sarà quello di interrogare, in maniera attiva, tutti gli APs in ascolto sulla rete.

Il grosso *vantaggio* di questo approccio è che, anche se viene consumata energia per trasmettere il frame in rete, la stazione non dovrà attendere un tempo indeterminato per una risposta da parte degli APs nei dintorni, ma potrà ottenere un risultato in un lasso di tempo determinato.

Nella Figura 2.8 viene mostrato, in dettaglio, il formato di un Probe Request e, nello specifico, il particolare del contenuto informativo trasportato all'interno del body. Infatti, la stazione provvederà ad inserire una serie di specifici IEs, tra cui, ad esempio, alcuni che ne descriveranno le caratteristiche supportate dalla scheda di rete della stazione (es. data rate, ecc...).

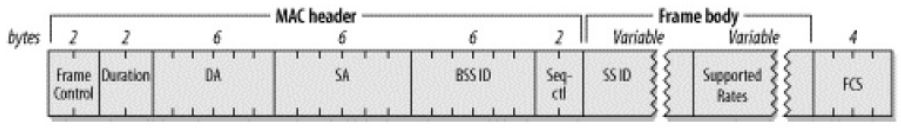


Figura 2.8: Specifico formato di un Probe Request Management frame.

Esistono due tipi di Probe Request che una stazione può mandare: direct o broadcast. Se un nodo vuole verificare la presenza di una determinata rete nei

suoi dintorni, manderà un *direct* probe request all'interno del quale sarà inserito l'Information Element che specifica l'SSID della rete che vuole ricercare. In tal caso, un AP risponderà a tale richiesta solo se l'SSID specificato nel probe request combaci con l'SSID gestito da quell'AP. Viceversa, un *broadcast* probe request sarà indirizzato a qualsiasi AP che si trovi nelle vicinanze; in questo caso, nel body del frame sarà specificato un IE con SSID nullo. L'uso di un direct probe request sarà necessario nel caso in cui una stazione si dovrà connettere ad un AP *hidden*, ovvero che non annuncia, tramite l'uso dei Beacon, la presenza della rete che ha provveduto a creare.

Lo *svantaggio* dell'approccio analizzato, è che talvolta, gli APs nei dintorni di una stazione potrebbero essere molti; processare tutte le risposte provenienti da ciascuno di questi potrebbe, quindi, richiedere un grosso carico computazionale e, conseguentemente, energetico, per la stazione. Soprattutto nel caso in cui un dispositivo è impostato in modalità "risparmio energetico", al fine di moderare il consumo energetico e allungare la durata della batteria, una soluzione potrebbe essere quella di fare uso di un *direct probe*, in modo da verificare la presenza di una rete a cui la stazione si era connessa in passato. Infatti, la maggior parte dei sistemi operativi, tipicamente, memorizzano una lista delle reti wireless a cui il dispositivo si è connesso; questo elenco è chiamato *Preferred Network List (PNL)* e contiene informazioni relative alle singole reti come l'SSID, i parametri di sicurezza o le caratteristiche fisiche supportate dalla rete.

Ad esempio, usando i direct probe request, in [18] è stato sviluppata una metodologia in grado di rivelare informazioni riguardo il proprietario di un dispositivo come la nazionalità, i luoghi visitati frequentemente oppure essere in grado di stabilire in che modo un certo individuo può essere messo in relazione con altri possessori di dispositivi wireless nelle sue vicinanze. Mentre, in [8] viene fatto uso degli SSID contenuti all'interno dei probe request al fine di determinare il luogo di origine dei partecipanti a degli eventi di massa, attraverso l'uso di un servizio noto come Wigle.net; un database che offre una corrispondenza tra gli SSID degli AP e le coordinate GPS a cui questi sono posizionati.

Mentre, intercettando un broadcast probe request all'interno di una certa area, ne conseguirà che il possessore di tale dispositivo si trovava all'interno di quel dato perimetro. Quindi, analizzando la distribuzione di tali frame è possibile determinare quanto spesso o quanto a lungo le persone hanno visitato un posto o le loro abitudini giornaliere.

Per quanto riguarda la frequenza di invio di un probe request in rete, lo standard non definisce alcuna restrizione, nè linea guida, lasciando tale onere al costruttore dell'apparato. Per tale ragione, ciò che otteniamo, è che la frequenza con cui vengono spediti i probe differisce, da dispositivo a dispositivo, in base ad una serie di fattori, come:

- Costruttore e/o tipologia della scheda di rete;

- Driver;
- Costruttore del dispositivo;
- Sistema Operativo;
- Applicazioni usate;

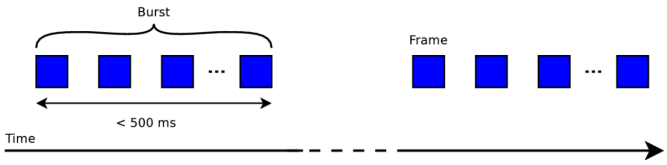


Figura 2.9: Trasmissione di Probe Request nel tempo da parte di una stazione. Un burst identifica un gruppo di frame trasmessi durante una finestra temporale inferiore a 500ms.

Tuttavia, [1] asserisce che è possibile identificare un certo pattern temporale, tramite cui i probe vengono spediti in rete. Infatti, come possiamo vedere dalla Figura 2.9, la stazione effettuerà la trasmissione all'interno di un *burst*, ovvero un breve intervallo temporale (tipicamente inferiore a 500ms), all'interno del quale vengono spediti un certo numero di probe.

Attraverso l'uso degli APs Cisco "Meraki" è stata condotta un'analisi [12] che ha permesso di ottenere informazioni riguardo ai dispositivi Wi-Fi e alla loro presenza in rete, esaminando i probe request spediti. In particolare, nella Tabella 2.3 sono riportate delle statistiche, ottenute empiricamente, inerenti la frequenza di invio dei probe da parte dei dispositivi monitorati attraverso tali apparati.

Stato	**Frequenza**
Dormiente (a schermo spento)	~1 volta/minuto
In attesa (a schermo acceso)	~10/15 volte/minuto
Associato	variabile, potrebbe essere necessario effettuare la ricerca manuale delle reti

Tabella 2.3: Frequenza di invio dei Probe Request, osservati empiricamente su diversi tipi di smartphones, con vari sistemi operativi (iOS, Android, ecc...) e in diverse condizioni (dormiente, in attesa, associato).

In aggiunta, in [2] vengono effettuati una serie di esperimenti su un certo numero di modelli di smartphone differenti, volti ad identificare i vari fattori che influenzano la frequenza di invio dei probe. In particolare, è stato appurato che questa

può variare anche a seconda della configurazione del dispositivo, oltre che alle altre caratteristiche elencate precedentemente; i principali fattori, che influiscono su questa, sono:

- Batteria in carica/non in carica;
- Modalità "risparmio energetico" abilitata/disabilitata;
- Schermo acceso/spento;
- Wi-Fi connesso/disconnesso;
- Impostazioni Wi-Fi;
- Bluetooth acceso/spento;
- Modalità "Aereo" abilitata/disabilitata.

I risultati ottenuti da tale documento, in termini di numero medio di probe spediti da una stazione, si sono rivelati essere congruenti con quelli individuati attraverso l'utilizzo degli apparati Cisco. Inoltre, è stato osservato che gli smartphone Android spediscono probe con una frequenza molto elevata, approssimativamente ogni minuto; questa varia a seconda della versione del sistema operativo presa in esame. Ad esempio, i dispositivi che montavano Android L 5.0.1, spedivano probe anche mentre erano associati ad un AP, mentre, il numero di probe inviati era direttamente legato al numero di reti salvate all'interno del PNL, per quei dispositivi che avevano Android 4.4.2.

Tra tutti i diversi tipi di management frame esistenti, si è ritenuto che il `Probe Request`, dato il contenuto informativo che trasporta, nonché il modo in cui esso è utilizzato in rete, costituisca un elemento di particolare interesse al fine del raggiungimento degli obiettivi prefissati da questo lavoro di tesi.

Proprio per queste sue caratteristiche, tuttavia, l'analisi di un probe request, potrebbe essere utile, non solo agli sviluppatori al fine di fornire dei servizi aggiuntivi, ma, soprattutto, a dei malintenzionati che voglio estorcere informazioni sensibili, quali identità e spostamenti degli utenti. Il paragrafo seguente sarà dedicato ad affrontare in maniera più dettagliata tale argomento.

2.4.4 Privacy degli utenti

Per via del fatto che, lo standard non prevede la trasmissione cifrata in rete nè dei frame di controllo, nè di quelli di management, chiunque potrebbe decidere di monitorare la rete e analizzarli. Negli ultimi anni, i ricercatori hanno constatato che il modo in cui sono stati progettati i protocolli, i quali fanno uso di questi particolari tipi di frame, può essere sfruttato da dei furfanti per rivelare dati personali riguardo ai possessori di dispositivi wireless.

Diversi tentativi sono stati effettuati per cercare di trovare una soluzione che fosse in grado di mitigare i problemi di sicurezza legati all'uso dei probe. Il concetto che sta alla base di queste soluzioni è quello di spezzare il legame esistente tra l'indirizzo MAC e l'utente possessore del dispositivo. Due studi, [6] e [4], suggeriscono di risolvere il problema alla radice introducendo, nel primo caso, degli identificativi usa e getta oppure, nel secondo caso, degli pseudonimi per gli indirizzi MAC dei dispositivi wireless. Anche se nello standard non è prevista alcuna soluzione riguardante tale problema, esistono due principali implementazioni commerciali da parte di Apple, a partire da iOS 8 [13], e di Android, a partire da Android Marshmallow [3]; in entrambi i casi, la soluzione adottata è quella di utilizzare degli indirizzi MAC randomizzati, ovvero degli indirizzi *locally administered*, che vengono utilizzati dai dispositivi per le funzioni di controllo e di management fino a quando questi non siano associati a qualche AP.

Risulta immediato riconoscere quali indirizzi MAC sono stati resi anonimi perché vengono generati degli indirizzi il cui prefisso non è stato assegnato a nessuna organizzazione e, tra le altre cose, sono individuabili analizzando il valore del bit U/L, che è rappresentato dal secondo bit meno significativo del primo ottetto del MAC address.

2.4.5 Tecniche per de-randomizzare un indirizzo

Vanhoef ed altri [19], hanno presentato una serie di tecniche che permettono di tracciare i dispositivi wireless indipendentemente dalle contromisure di sicurezza adottate per garantire la privacy dei possessori degli stessi.

Un possibile attacco è quello a cui sono soggetti tutti quegli apparati che supportano il protocollo *Wi-Fi Protected Setup (WPS)*, il quale permette a dei dispositivi non autenticati di negoziare una connessione sicura con un AP. Tuttavia, affinché tale connessione sicura possa essere instaurata, la stazione scambierà con l'AP dei probe request all'interno dei quali inserirà un IE di tipo Vendor Specific (vedi Tabella 2.2), che conterrà una serie di informazioni tra cui: il costruttore e il modello del dispositivo e un identificativo univoco chiamato *Universally Unique IDentifier-Enrollee (UUID-E)*, il quale sarà poi usato per creare la connessione WPS. Ma, queste informazioni risultano essere utili anche per poter tracciare il dispositivo stesso, infatti, è stato scoperto che l'UUID-E è derivato dall'indirizzo MAC reale del dispositivo e che un attaccante potrebbe usare una tabella di hash pre-calcolata ed effettuare un lookup di uno specifico UUID-E all'interno di tale tabella, al fine di ottenere l'indirizzo MAC reale di un dato dispositivo [9, 19].

Un'ulteriore tecnica, discussa da Vanhoef ed altri [19] e Matte ed altri [10], si basa sul calcolo di un'impronta, detta *fingerprint*, inerente ad un probe request, che ne costituisce, con un certo grado di approssimazione, un identificativo univoco della stazione che lo ha spedito. È possibile calcolare il fingerprint in base ad una serie di informazioni contenute nel body di un probe request:

SSID fingerprint: i dispositivi tipicamente mandano una serie di probe request, uno per ogni SSID contenuto nella propria PNL; per questa ragione, l'insieme di tutti gli SSID cercati da uno stesso dispositivo può costituirne un identificativo univoco. Nell'ultimo periodo, la tecnica dell'SSID fingerprint è stata progressivamente limitata a causa delle ragioni riguardanti la privacy degli utenti; tuttavia, può ancora essere presa in considerazione visto che non tutti i dispositivi sono aggiornati con l'ultima versione del sistema operativo, che adotta le modifiche necessarie per risolvere tali problemi.

IEs fingerprint: all'interno del body di un probe request, sono inseriti una serie di IEs che descrivono le capacità del dispositivo, come: la velocità di trasmissione supportata, corrispondente alle informazioni inserite all'interno dell'IE `Supported Rates`, oppure la capacità di connettersi ad altri nodi in rete, descritte nell'IE `Interworking Capabilities`. Dato che l'inserimento di un particolare tipo di IE è opzionale, ogni dispositivo specificherà, all'interno dei probe request da esso generati, una collezione di IEs che differirà da quelli spediti da un'altra stazione; per tale ragione, l'insieme degli IEs specificati da un dispositivo potranno essere sfruttati per calcolare un'impronta che identifichi univocamente la stazione.

In [11] viene analizzato un attacco basato sul tempo di arrivo di un probe; in particolare, dato che questi sono spediti con una frequenza che differisce per ogni dispositivo e che, tipicamente, vengono raggruppati all'interno di uno stesso burst (come mostrato nella figura 2.9), viene calcolata una firma, detta *signature*, per ogni gruppo di frame ricevuto e, successivamente, confrontata con tutte le alte firme conosciute attraverso una particolare metrica, al fine di identificare delle signature "simili". Tutti quei burst di probe, caratterizzati da indirizzi MAC differenti e che presentano signature simile tra di loro saranno considerati come uno stesso dispositivo.

Infine, in [19] viene presentato il funzionamento dell'attacco *Karma attack*, un attacco in cui il malintenzionato assume un ruolo *attivo* mediante la configurazione di un AP, in modo che questo annunci un insieme degli N più noti SSID; in questo scenario, un qualsiasi dispositivo, configurato automaticamente per connettersi ad una delle reti appartenenti a tale insieme, sarà costretto a comunicare con tale AP in uno stato non autenticato che lo porterà a rivelare il suo indirizzo MAC reale, bypassando così l'uso di un eventuale tecnica per la randomizzazione del suo indirizzo.

Capitolo 3

Architettura della soluzione

In questo capitolo, verrà discussa l'intera architettura del sistema implementato, ovvero sia l'hardware che il software necessario per poter perseguire i fini che questo lavoro di tesi si è preposto. In particolare, saranno analizzati il modo in cui i dati vengono raccolti su ogni mezzo di trasporto e come questi, successivamente, saranno trasmessi ad un server remoto, il quale avrà il compito di analizzarli e fornire, attraverso delle REST API, il risultato finale di tale elaborazione.

3.1 Panoramica generale

L'approccio che è stato adottato è incentrato sul lavoro descritto in [14] al fine di ottenere una stima del numero di passeggeri presenti su di un mezzo di trasporto. L'idea di base che viene sfruttata è quella per cui i dispositivi wireless mandano periodicamente in rete una serie di messaggi di management (argomento che è stato analizzato dettagliatamente nel Capitolo 2), tra questi una particolare tipologia, chiamata *probe request*, ha lo scopo di implementare determinati servizi di *discovery* che hanno l'obiettivo di individuare tutti gli APs presenti nelle vicinanze.

Monitorando continuamente il flusso dei probe request, che vengono trasmessi dalle diverse stazioni, risulta possibile effettuare una stima del numero di dispositivi e, indirettamente, del numero dei possessori di questi (assumendo che a ciascun device corrisponda una persona distinta), i quali si trovano nelle vicinanze dell'apparato di rete utilizzato per effettuare tale operazione di monitoraggio.

L'intera architettura del sistema è rappresentata nella Figura 3.1 e, come possiamo vedere, è caratterizzata da tre attori principali:

- Router
- Microcontrollore
- Server Remoto

Il *microcontrollore* sarà l'apparato che dovrà occuparsi di monitorare i probe request trasmessi in rete e di legare ciascuno di questi all'ultima posizione GPS del mezzo di trasporto nota. Infine, si occuperà di trasmettere i dati raccolti ad un server remoto.

Il *server remoto*, una volta ricevuti i dati, si occuperà di analizzarli e, successivamente, memorizzarli all'interno di una base dati.

Infine, il *router*, oltre ad implementare le funzionalità tipiche di questa tipologia di apparati, quali, ad esempio, lo smistamento dei pacchetti attraverso la rete, si occuperà anche di calcolare la posizione GPS relativa al veicolo di trasporto sul quale è installato.

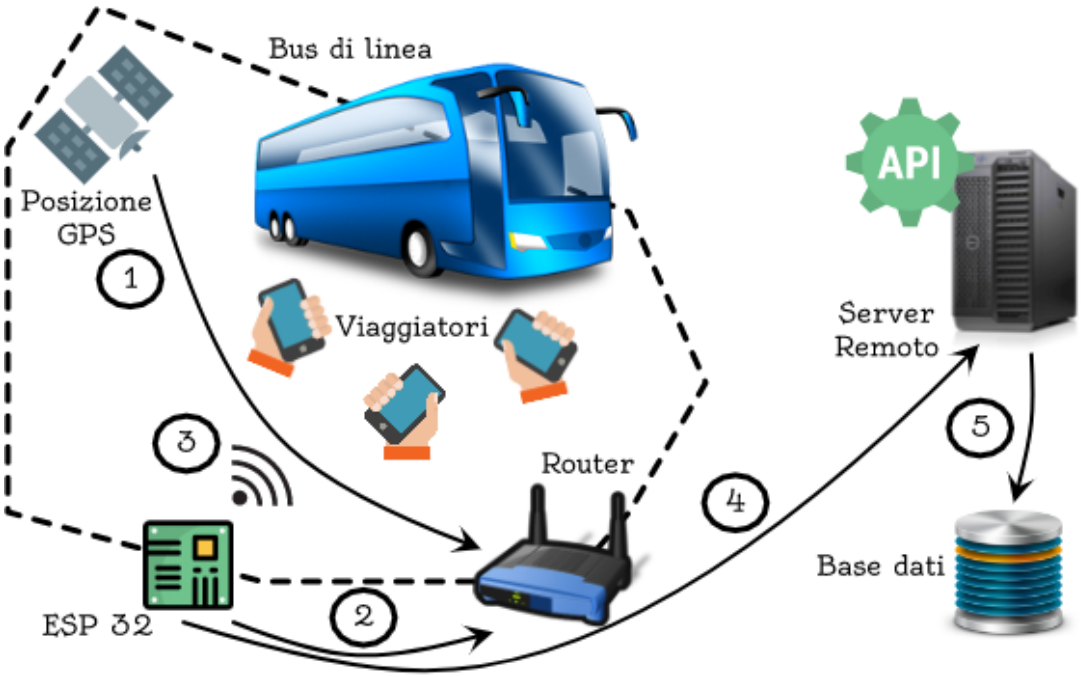


Figura 3.1: Architettura della soluzione implementata

All'interno di questa soluzione, risulta possibile distribuire, tra i vari componenti, la responsabilità di analizzare i dati ottenuti dal processo di monitoraggio e di determinare, sulla base di questi, una stima del numero totale di dispositivi presenti in un dato istante. Sono state valutate due principali possibilità: quella di fornire l'onere di tale elaborazione al microcontrollore oppure di attribuirla al server remoto.

Sebbene, in un primo momento, possa sembrare maggiormente conveniente la prima opzione, successivamente è stato ritenuto più opportuno optare per la seconda, affidando tutta la computazione al server.

Il vantaggio della prima soluzione è che, essendo la computazione effettuata tutta sul microcontrollore, l'implementazione del server diventa molto semplice, in quanto dovrà occuparsi esclusivamente della memorizzazione sulla base dati del risultato già elaborato, tuttavia, sono stati riscontrati una serie di svantaggi:

- *Mole di dati inviati:* dato che il microcontrollore dovrà occuparsi anche dell'elaborazione dei probe che sono stati ricevuti, non potrà fornire al server una risposta immediata, ma questa sarà ritardata in modo da poter raccogliere un numero sufficiente di probe necessari a poter generare un risultato aggregato utile per il server. La trasmissione in rete di questo risultato, tuttavia, può essere molto onerosa, in termini di banda richiesta, nel caso in cui il dispositivo sia posto in un ambiente particolarmente affollato.
- *Memoria insufficiente:* fornire al server una risposta ritardata, significa essere in grado di memorizzare tutti i probe ricevuti dall'ultima volta in cui il server è stato contattato. Questo comporta la necessità di dover salvare tali dati intermedi nella memoria del microcontrollore, che tipicamente è molto limitata.
- *Scalabilità:* nel caso in cui è necessario installare più di un microcontrollore sullo stesso mezzo di trasporto, questi potrebbero ricevere, in parte, gli stessi probe request ed elaborare le informazioni contenute in essi in maniera differente; in questo caso, il server dovrà poi implementare una qualche metodologia che gli permetta di accorgersi di questa duplicazione.

Viceversa, attraverso la seconda soluzione, è possibile ottenere diversi vantaggi:

- *Mole di dati inviati:* il microcontrollore, ogni qual volta verrà ricevuto un nuovo probe, provvederà ad estrarre e trasmettere al server soltanto le informazioni utili ai fini dell'elaborazione finale.
- *Scalabilità:* anche nel caso in cui più microcontrollori siano installati su di uno stesso veicolo, questi si occuperanno soltanto della trasmissione delle informazioni utili al server per la successiva elaborazione; la maggiore semplicità di questi dispositivi comporterà, quindi, una maggiore scalabilità della soluzione.
- *Firmware semplice:* l'implementazione del firmware di un microcontrollore sarà molto semplice, questo comporterà il grosso vantaggio di non doversi più preoccupare della limitata quantità di memoria a disposizione su questi dispositivi.

Nelle successive sezioni, verranno descritti nel dettaglio ognuno dei tre principali componenti del sistema in analisi.

3.2 Router Teltonika RUT955

Per quanto riguarda la scelta del router, è stato necessario cercare un apparato che, innanzitutto, fosse in grado di garantire connettività su di un mezzo di trasporto in movimento; in un contesto del genere, la rete 4G/3G si è rivelata la soluzione più veloce e versatile per poter ottenere una connessione a larga banda.

Un secondo requisito, fondamentale per poter effettuare successivamente un analisi, è la presenza di un modulo GPS attraverso il quale è possibile ottenere le coordinate del veicolo in movimento.

Figura 3.2: RUT955 è un router LTE progettato proprio per l'implementazione di applicazioni professionali.

Il router RUT955 è stato reputato la scelta più consona poiché è in grado di offrire una connessione WAN sia attraverso la tradizionale interfaccia WAN Ethernet ma anche attraverso il modulo 4G LTE integrato, il quale sfrutta la connettività di una scheda SIM di un qualsiasi operatore e, per questa ragione, si presta bene all'implementazione di soluzioni in mobilità, come su: auto, treni o autobus e in generale tutti quei luoghi dove è costoso ottenere una linea telefonica fissa. Per via del campo applicativo in cui questo dispositivo può essere utilizzato, esso è caratterizzato da un assemblaggio di tipo industriale, in modo da essere grado di supportare forti sollecitazioni sia termiche che meccaniche.

Il router RUT955 include anche una serie di altre funzionalità che si sono rivelate molto interessanti e che rendono tale apparato uno strumento molto completo, quali:

- La possibilità di funzionare secondo diverse *modalità*:

Wired: la connessione ad Internet è fornita tramite un modem ADSL;

4G/3G: la connessione ad Internet è fornita tramite la SIM inserita un'apposito slot del router;

Wi-Fi: la connessione ad Internet è fornita tramite un Access Point.

- *Supporto dual-SIM:* è possibile inserire due SIM differenti, specificandone una come primaria e l'altra come secondaria; se la connessione sulla SIM primaria non risulterà attiva, il router commuterà automaticamente sulla secondaria.
- *Sicurezza:*

  Firewall: dispone di un Firewall integrato configurabile, il quale protegge la rete LAN, che ha provveduto a creare, dal resto del mondo.

  Crittografia: è possibile applicare diversi sistemi di crittografia all'interfaccia wireless, al fine di proteggere la LAN dagli accessi non autorizzati.

- Altre funzionalità tipiche di un router come NAT, DHCP server, VLAN, ecc...
- Fornisce una serie di servizi avanzati, tra cui i principali sono:

  QoS permette di gestire diverse priorità per differenti tipi di traffico.

  MQTT protocollo di tipo *publisher-subscriber* che permette di mandare dei messaggi sfruttando il protocollo TCP/IP.

  NTP protocollo che permette al router di sincronizzare l'ora e data corrente con un server NTP.

  VPN metodo che permette di trasferire in maniera sicura i dati sfruttando una rete non sicura

  GPS permette di configurare diversi parametri, mostrati in Figura 3.3, che possono influire sulla frequenza con cui viene calcolata una nuova coordinata GPS.

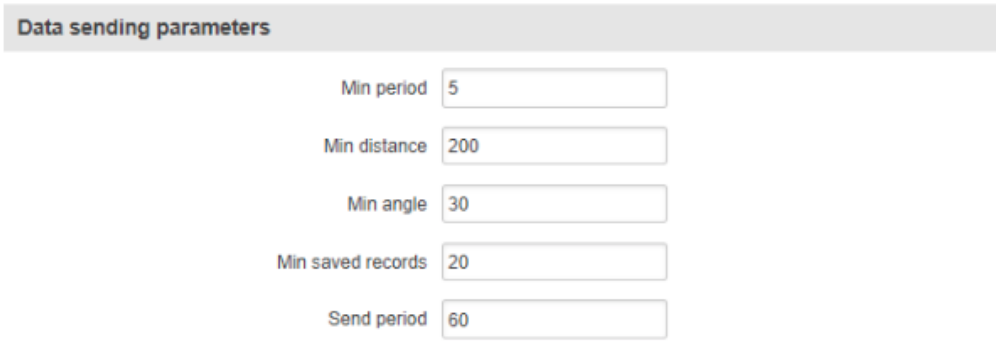


Figura 3.3: Parametri che influiscono sulla frequenza con cui viene calcolata ed inviata una coordinata GPS.

- Completamente configurabile e gestibile tramite un'interfaccia WEB

Analizzando la Figura 3.1, rappresentante l'architettura del sistema implementato, possiamo vedere come, il router RUT955 sarà coinvolto, principalmente, nello svolgimento di due operazioni (non tenendo in considerazione le operazioni di networking) utili per potersi coordinare con il microcontrollore.

Innanzitutto, al Punto 1, sfruttando l'antenna GPS di cui è dotato, il router sarà in grado di calcolare delle coordinate GPS con un certo grado di affidabilità (a seconda di quanto è forte il segnale GPS captato in quello specifico istante). In particolare, la frequenza con cui sarà calcolata una nuova posizione, dipenderà dai parametri mostrati in Figura 3.3, che specificano, rispettivamente:

Min period il tempo minimo (in secondi) per raccogliere i dati necessari all'elaborazione di una nuova posizione

Min distance la distanza minima (in metri) tra l'ultima posizione registrata e la corrente (anche se il *min period* non è stato ancora raggiunto)

Min angle l'angolo minimo della differenza tra le coordinate dell'ultima posizione registrata e le coordinate di quella corrente corrente (anche se il *min period* non è stato ancora raggiunto)

Min saved records numero minimo di coordinate registrate prima di poterle inviare (anche se il *Send period* non è stato ancora raggiunto)

Send period periodo entro il quale tutti i dati collezionati devono essere inviati

Infine, la seconda operazione svolta dal router, schematizzata al Punto 2 della Figura 3.1, sarà quella di fornire, attraverso un particolare servizio che viene abilitato al boot del sistema, tutte le posizione GPS aggiornate al microcontrollore, a mano a mano che queste verranno calcolate.

3.3 ESP-32

Tra i tanti microcontrollori disponibili nel mercato, per questo lavoro di tesi è stato scelto di utilizzare il system on chip (SoC) chiamato ESP32, principalmente per il suo rapporto qualità-prezzo. ESP32 è un SoC a basso costo che è stato creato e sviluppato dalla compagnia cinese Espressif Systems. Rapportato al suo prezzo di mercato, esso è caratterizzato da delle ottime specifiche tecniche, come:

Processore: microprocessore dual-core a 240MHz

Connettività Wireless: è dotato di un'antenna incorporata che viene sfruttata dai due moduli wireless presenti

Figura 3.4: ESP32 è un microcontrollore low-cost creato e sviluppato dalla compagnia cinese Espressif Systems.

- Wi-Fi (802.11 b/g/n)
- Bluetooth v.4 e Bluetooth Low Energy (BLE)

Memoria flash: 4MiB

I/O: tramite una gestione in DMA permette di raggiungere diversi tipi di periferiche, come: ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I²C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), SPI (Serial Peripheral Interface) e tante altre.

Sicurezza: sono supportate

- le funzionalità di sicurezza dello standard IEEE 802.11, come WPA/WPA2
- la generazione di password OTP a 1024-bit
- un motore crittografico implementato in hardware, che permette di sfruttare tutti i più noti meccanismi di sicurezza, come: AES, SHA-2, RSA, crittografia a curve ellittiche (ECC), generatore di numeri random (RNG)

All'interno dell'architettura descritta precedentemente, il SoC ESP32 assume un ruolo fondamentale poiché, come possiamo osservare dalla Figura 3.1, è coinvolto in tre principali tipi di operazioni.

Innanzitutto, all'avvio del sistema l'ESP32 provvederà a connettersi ad un servizio offerto dal router, come mostrato al Punto 2, attraverso cui sarà in grado di ottenere le posizioni GPS aggiornate del mezzo di trasporto su cui questi apparecchi sono installati.

Successivamente, come mostrato al Punto 3, l'ESP32 incomincerà a monitorare la rete, analizzando i frame trasmessi dai dispositivi dei passeggeri e filtrando esclusivamente quelli di un particolare tipo, ovvero i probe request.

Infine, dal probe, ottenuto al Punto 3, saranno estratte esclusivamente le informazioni di interesse per l'analisi da effettuare e unite all'informazione relativa alla posizione GPS corrente, ottenuta al Punto 2, e, infine, il tutto sarà poi trasmesso attraverso la rete verso il server remoto, come mostrato dal Punto 4.

3.4 Server remoto: Spring framework

Per lo sviluppo del server remoto è stato utilizzato un framework molto completo e versatile, chiamato *Spring*, usato spesso in ambiente *enterprise* e non, per via degli innumerevoli vantaggi che esso comporta.

Figura 3.5: Spring è un framework open-source utilizzato per implementare il server remoto.

Spring è uno dei framework open-source più popolari per lo sviluppo di applicazioni Java enterprise (dette anche *Java EE* o *Java Enterprise Edition*), ovvero tutte quelle applicazioni che hanno la necessità di soddisfare particolari requisiti di sicurezza, performance e affidabilità, infatti, esso permette di creare codice altamente performante, facile da testare e riutilizzabile.

Spring è dotato di una serie di benefici che lo rendono un framework particolarmente interessante:

1. *Uso di classi POJO:* Spring permette di sviluppare un'applicazione Java EE completa usando solo delle classi POJO (Plain Old Java Object), le quali vengono dotate di funzionalità enterprise in modo non invasivo.

2. *Modulare:* Spring ha un'architettura altamente modulare; ciò permette allo sviluppatore di includere, all'interno del suo progetto, esclusivamente i moduli di cui necessita.

3. *Integrabile:* La filosofia di Spring è quella di non dover reinventare la ruota ma, piuttosto, quella di rendere più facile l'utilizzo di funzionalità offerte da

altre tecnologie esistenti come i framework per la gestione di diversi tipi di ORM, quelli che forniscono funzionalità di log e tanti altri...

4. *Facile da testare:* Effettuare il testing di un'applicazione scritta con Spring è un operazione molto semplice ed intuitiva.

5. *Leggero:* Il componente principale che permette il funzionamento dell'intero framework, detto IoC Container, è molto snello e leggero; ciò risulta essere utile per lo sviluppo e la distribuzione di applicazioni su terminali dotati di limitate risorse (memoria o CPU).

3.4.1 REST Web Service

Con particolare riferimento all'architettura sviluppata, il server remoto è stato implementato sfruttando uno dei tanti moduli offerti dal framework Spring, chiamato *Spring Web MVC*, che è appositamente pensato per la realizzazione di applicazioni web. Tale framework è stato sfruttato al fine di realizzare un *REST web service* che consente di mettere a disposizione una serie di *API* utili per poter interagire con le informazioni memorizzate all'interno della nostra base dati.

REST

Un REST web service è basato sullo stile architetturale detto *Representational State Transfer (REST)*, ovvero un insieme di linee guida per la realizzazione di servizi web con specifiche proprietà, quali:

- **Affidabilità:** capacità di un sistema di garantire la qualità del servizio offerto, anche a fronte di malfunzionamenti locali delle sue parti.
- **Efficienza:** rapporto tra le risorse impegnate per offrire un servizio e il valore del risultato prodotto.
- **Prestazioni:** valutazione relativa al tempo e alle risorse necessarie per erogare un dato servizio.

In REST, i dati e le funzionalità sono considerati delle *risorse* a cui vi si può fare accesso tramite degli identificatori globali, detti *Uniform Resource Identifiers (URI)*, e ad un insieme specifico di *operazioni* che è possibile compiere su una data risorsa.

In un ambiente REST, le componenti di una rete (client e server) interagiscono attraverso un protocollo di comunicazione *stateless*, come tipicamente il protocollo HTTP, al fine di poter scambiare, attraverso interfacce e protocolli standardizzati (es. XML o JSON), delle *rappresentazioni* di una specifica risorsa.

3.4.2 MQTT

Al fine di realizzare la comunicazione efficiente tra l'ESP32 e il server remoto è stato adottato il protocollo MQTT.

MQTT (Message Queuing Telemetry Transport) è un protocollo di messaggistica molto leggero del tipo publish/subscribe che si basa sulle garanzie di affidabilità e riordinamento dei pacchetti offerte dal protocollo TCP/IP. È stato progettato per la connessione verso sedi remote in situazioni in cui è richiesto un basso impatto e/o la banda della rete è limitata; proprio per queste caratteristiche, MQTT assume un ruolo importante soprattutto nel contesto dell'Internet of things (IoT).

Publish/Subscribe pattern

Il pattern publish/subscribe è un'alternativa al tradizionale modello client-server in cui, piuttosto che avere un client che comunica direttamente con un certo endpoint, avremo un *publisher* ed uno o più *subscriber* che non parleranno mai direttamente, ma solo attraverso un terzo componente, detto *broker*. Il publisher manderà dei messaggi al broker che saranno distinti per argomento, detto *topic*, mentre, il subscriber effettuerà la sottoscrizione ad un particolare topic. Nel momento in cui il broker riceverà un messaggio appartenente ad uno specifico topic, esso si preoccuperà di inoltrarlo a tutti coloro che avranno, precedentemente, effettuato la sottoscrizione a quel determinato argomento.

QoS

MQTT fornisce la possibilità di definire dei meccanismi di Quality of Service (QoS) per i messaggi spediti da un publisher e, in particolare, offre 3 livelli che determinano un diverso grado di garanzia che il messaggio raggiunga la destinazione (il broker o un qualsiasi altro subscriber):

At most once (QoS=0): Il messaggio viene consegnato al massimo una sola volta o, in alternativa, non è completamente consegnato. Questo comportamento è dovuto al fatto che il messaggio non viene mai memorizzato durante la sua trasmissione attraverso la rete, per cui potrebbe andare perso nel caso in cui un client si disconnetta improvvisamente oppure si verifichi un guasto sul server.

At least once (QoS=1): Il messaggio viene sempre consegnato almeno una volta. Nel caso in cui il mittente non riceverà un ACK che ne confermi la corretta trasmissione, questo provvederà a spedirlo nuovamente; ciò potrebbe comportare che il destinatario processi più volte lo stesso messaggio. Inoltre, il messaggio dovrà essere memorizzato localmente sia sul mittente che sul destinatario fino a quando questo non sarà processato e un ACK di conferma sarà mandato al mittente. Nel caso in cui il destinatario è il broker, questo si occuperà di inoltrare tale messaggio a tutti i relativi subscriber.

Exactly once (QoS=2): Il messaggio è sempre mandato esclusivamente una volta; questo dovrà essere memorizzato localmente sia sul mittente che sul destinatario fino a quando il messaggio non sarà processato e un ACK di conferma sarà mandato al mittente. Questo è il metodo di trasferimento che offre il maggior grado di sicurezza ma, allo stesso tempo, è anche quello più lento perché comporterà lo scambio di un numero superiore di messaggi tra mittente e destinatario al fine di fornire tale garanzia.

RabbitMQ

Per fare in modo che l'ESP32 possa comunicare con il server remoto attraverso il protocollo MQTT è necessario utilizzare un broker, il quale costituisce il fulcro del funzionamento del pattern publish/subscribe. Il broker che è stato utilizzato è RabbitMQ, un broker open-source che implementa principalmente il protocollo AMQP, ma che offre anche la possibilità di installare dei moduli alternativi che permettono di far implementare a tale broker altri protocolli di tipo publish/subscribe; in particolare, è stato utilizzato un modulo specifico per il protocollo MQTT. In Figura 3.6, vengono schematizzate le interazioni che avvengono tra l'ESP32, il server e il broker RabbitMQ.

Figura 3.6: Schema dell'utilizzo di RabbitMQ per la comunicazione tra l'ESP32 e il server.

3.5 Base dati

Nella sezione 3.1 è stato discusso l'approccio adottato in questo lavoro di tesi per quanto riguarda l'analisi delle informazioni raccolte dall'ESP32. In particolare, considerando che ogni probe, intercettato dal microcontrollore, venga spedito al server per poter essere memorizzato su una base dati e che la frequenza di trasmissione dei probe da parte di un dispositivo, in uno qualsiasi degli stati schematizzati nella Tabella 2.3, possa essere superiore a 20 probe/minuto, comporta che una tradizionale base dati relazionale non sarebbe adatta a gestire una così grande mole di informazioni.

Di seguito verranno analizzati i vantaggi e gli svantaggi nell'utilizzo sia di basi dati relazionali che non-relazionali.

3.5.1 DBMS relazionali vs. non-relazionali

All'interno di una base dati relazionale, al fine di assicurare che tutte le transazioni siano processate in maniera affidabile, sono garantite quattro proprietà principali, note come proprietà ACID:

Atomicità: Ogni transazione è un'unità indivisibile, se una delle parti logiche di cui è composta fallisce, fallirà l'intera transazione.

Consistenza: L'esecuzione di una transazione deve preservare i vincoli di integrità definiti sulla base dati.

Isolamento: L'esecuzione di una transazione deve essere indipendente dall'esecuzione di altre transazioni.

Durabilità: I cambiamenti dovuti dal corretto completamento di una transazione devono essere memorizzati e disponibili all'interno della base dati in modo che non possano essere persi in caso di guasti.

Tuttavia i DBMS relazionali presentano alcuni *svantaggi*:

- È necessario definire uno schema ben preciso prima di poter effettuare l'inserimento dei dati e, inoltre, questi devono essere normalizzati, allo scopo di eliminare ridondanze e anomalie durante gli aggiornamenti.
- È difficile per un DBMS relazionale scalare orizzontalmente, ovvero, dati un insieme di nodi, partizionare le tabelle della base dati in modo che ogni nodo possa gestire solo una parte di esse, continuando, allo stesso tempo, a garantire tutte e quattro le proprietà ACID.

Viceversa, le basi dati non-relazionali, anche dette NoSQL, sono progettate in modo da poter rilassare i vincoli posti dalle proprietà ACID, infatti queste:

- Consentono l'inserimento e la manipolazione dei dati senza uno schema predefinito; in questo modo, è possibile apportare modifiche significative allo schema dei dati in tempo reale, senza preoccuparsi di possibili interruzioni del servizio.
- Supportano nativamente lo *sharding*, ovvero un metodo attraverso cui è possibile distribuire i dati tra un numero arbitrario di macchine, in maniera del tutto trasparente per le applicazioni.

Tuttavia, per poter ottenere un'architettura che sia in grado di scalare orizzontalmente, in contrapposizione alle proprietà ACID, un DBMS NoSQL dovrà supportare le cosiddette proprietà BASE:

Basic Available: Indica la capacità di un sistema di garantire la disponibilità dei dati per mezzo di una risposta (positiva o negativa) ad una data query.

Soft state: Indica che lo stato del sistema può cambiare nel tempo anche senza il verificarsi di alcun input.

Eventually consistent: A fronte dell'inserimento di nuovi dati, il sistema non sarà immediatamente consistente, ovvero tutti i nodi replica possiederanno la stessa copia dei dati, ma lo diventerà gradualmente, a mano a mano che l'informazione venga propagata.

Per le necessità dettate dal particolare contesto in cui il sistema sviluppato si colloca, è stato scelto di lavorare con dei DBMS non-relazionali principalmente per la loro capacità di memorizzare grandi volumi di dati che spesso non hanno una struttura definita a priori.

Esistono diverse tipologie di basi dati non-relazionali e queste possono essere classificate in quattro principali categorie: key-value, a documenti, a grafo e a colonna; ciascuno di questi differirà per il modo in cui i dati saranno gestiti e memorizzati. Nello specifico, è stato scelto di lavorare con un database appartenente alla categoria di quelli "a documenti", chiamato MongoDB.

3.5.2 MongoDB

MongoDB (da "hu**mongo**us", ovvero gigantesco in inglese) è un DBMS non-relazionale open-source, orientato ai documenti. In MongoDB vi sono tre principali compo-

Figura 3.7: MongoDB è il DBMS non relazionale che è stato utilizzato per memorizzare le informazioni raccolte.

nenti:

Database: contenitore di una o più strutture dati chiamate collezioni;

Collezioni: insieme di documenti correlati tra di loro che possono essere caratterizzati da uno schema differente;

Documenti: i dati sono memorizzati sotto forma di documenti che hanno un formato simile al JSON, chiamato *BSON*; documenti diversi di una stessa collezione possono avere uno schema differente.

Le principali caratteristiche di MongoDB sono:

- *Alte prestazioni:* MongoDB permette di ottenere alte prestazioni durante le operazioni di lettura/scrittura di un dato. In particolare, attraverso l'uso di *indici* è possibile interrogare in maniera efficiente i dati di una collezione.
- *Operazioni CRUD:* MongoDB supporta un ricco linguaggio per poter interrogare una determinata collezione e, in particolare, ogni operazione di lettura lavorerà su una singola collezione alla volta, mentre, ogni operazione di scrittura sarà atomica a livello di singolo documento.
- *Alta disponibilità:* La funzione in MongoDB che supporta la replicazione dei dati, chiamata *replica set*, permette di ottenere ridondanza senza il verificarsi di perdite.
- *Scalabilità orizzontale:* La funzionalità che permette a MongoDB di essere scalabile orizzontalmente è chiamata *sharding*, il quale permette di distribuire i dati attraverso un insieme di macchine.

Con riferimento al workflow mostrato in Figura 3.1, non appena il server remoto riceverà un nuovo dato, esso si occuperà di:

- salvarlo in una collezione del database MongoDB (schematizzato dal Punto 5)
- analizzarlo, mettendolo in relazione con tutti i dati precedentemente ricevuti, e, successivamente, salvando il risultato di tale analisi all'interno di una seconda collezione (schematizzato dal Punto 5)

Capitolo 4

Dettagli implementativi

In questo capitolo, verranno discussi i dettagli implementativi che riguardano ogni componente facente parte del sistema sviluppato. In particolare, sarà analizzato il servizio, implementato sul router, che si occupa di rendere disponibili le posizioni GPS al microcontrollore ESP32, la procedura attraverso cui quest'ultimo trasmette un nuovo dato attraverso la rete e, infine, il modo in cui i dati vengono analizzati sul server remoto.

4.1 Router

Sul router è stato svolto principalmente un lavoro di configurazione, attraverso il pannello offerto dall'interfaccia web, mirato a:

- creare una rete LAN wireless che l'ESP32 possa sfruttare in modo da ottenere connettività in rete;
- abilitare il servizio GPS.

4.1.1 Servizio GPS

Attraverso il pannello di controllo, nella sezione riguardante il servizio GPS, è possibile abilitare:

- il servizio GPS vero e proprio, ovvero abilitare il router ad utilizzare l'antenna GPS di cui è dotato;
- un servizio che ha lo scopo di mandare le informazioni riguardanti un insieme di posizioni GPS, raccolte in un dato lasso di tempo, ad un server e ad una certa porta, utilizzando il protocollo TCP o UDP. La frequenza con la quale queste informazioni saranno mandate muterà a seconda dei parametri, mostrati in Figura 3.3, che verranno specificati.

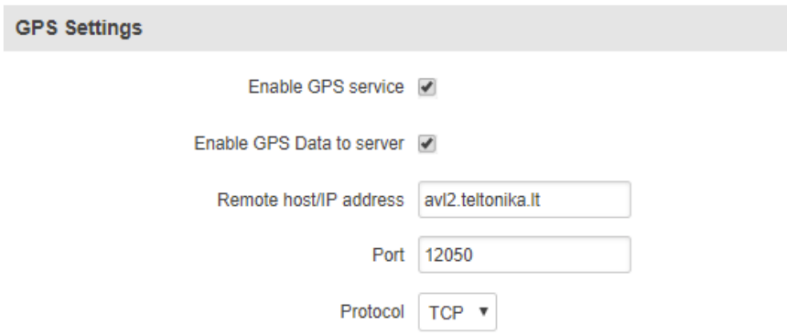


Figura 4.1: Sezione del pannello di controllo del router RUT955 che permette di configurare il servizio GPS.

Entrambe le opzioni sono state abilitate; conseguentemente, sul server è stato reso disponibile un servizio costantemente in attesa di dati su una certa porta TCP.

Protocollo per il trasferimento dei dati GPS

Nel momento in cui un nuovo insieme di posizioni GPS è disponibile, il router attuerà un protocollo per il loro trasferimento, il quale può essere riassunto nei seguenti passi:

1. il router manderà il proprio IMEI, in modo che sarà possibile distinguerlo da altri dispositivi configurati per contattare lo stesso servizio TCP;
2. il server dovrà rispondere con un ACK per accettare il trasferimento oppure con un NACK per rifiutarlo;
3. nel caso in cui venga ricevuto un ACK, il router provvederà ad inviare delle informazioni GPS, caratterizzate dal seguente formato:

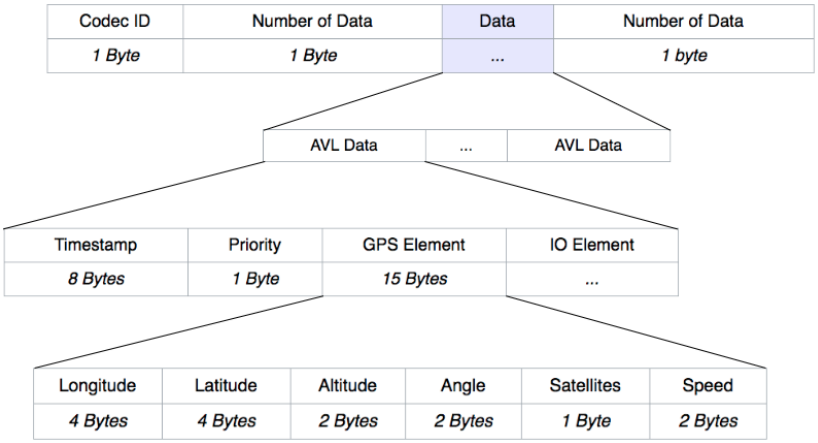


Figura 4.2: Formato delle informazioni che vengono trasmesse dal router al server, contenenti una serie di posizioni GPS.

Come è possibile osservare dalla Figura 4.2, il campo `Number of Data` specificherà il numero di `AVL Data` riportati all'interno del campo `Data`. Ogni AVL Data conterrà un `Timestamp` che specificherà la data e ora in cui è stato generato un certo `GPS Element`. Infine, quest'ultimo conterrà una serie di informazioni quali:

- la latitudine, longitudine e altitudine, che costituiscono la coordinata geografica;
- l'angolo di inclinazione rispetto all'emisfero Nord;
- il numero di satelliti visibili dall'antenna GPS;
- la velocità (in Km/h) del mezzo di trasporto in cui l'antenna GPS è installata.

Il server, una volta ricevute tali informazioni, provvederà a salvarle in una particolare collezione su MongoDB.

Script LUA

Il problema principale del servizio descritto precedentemente è che non permette di stabilire una correlazione tra le informazioni relative alle posizioni GPS calcolate dal router e le informazioni riguardanti i probe catturati dall'ESP32.

Fortunatamente, il router RUT955 fornisce la possibilità di specificare dei comandi, in linguaggio bash, che saranno eseguiti al boot del sistema. Sfruttando questa funzionalità, è stato possibile:

- Ricavare l'IMEI del router attraverso l'utilizzo dei comandi `gsmctl`, ovvero un insieme di comandi da shell che permettono di richiedere informazioni ad un modem;
- Realizzare uno script in linguaggio *LUA*, a cui sarà passato anche l'informazione relativa all'IMEI.

Lo script LUA avrà il compito di creare un servizio che resti in attesa di nuove connessioni su una specifica porta TCP. Nel momento in cui un nuovo client si connetterà a tale server, questo provvederà a:

1. Leggere le informazioni che provengono dall'antenna GPS, in maniera analoga al caso in cui tale dispositivo fosse un file e si dovesse leggere da esso. Tutti i produttori di dispositivi GPS supportano un formato dati standard chiamato NMEA, caratterizzato da molteplici tipi di messaggi, ciascuno dei quali specifica informazioni differenti.

2. Filtrare, tra tutti i diversi tipi di messaggi NMEA, quelli di tipo `$GPRMC`, estrapolando esclusivamente le informazioni necessarie, ovvero: latitudine, longitudine e timestamp dell'istante in cui quelle coordinate sono state calcolate.

3. Inviare le informazioni ricavate dai messaggi `$GPRMC`, arricchite dall'IMEI del router, al client.

Questo procedimento sarà ripetuto ogni qual volta il dispositivo GPS produrrà dei nuovi messaggi.

4.2 ESP32

Il microcontrollore ESP32 è stato programmato in linguaggio C sfruttando l'*ESP-IDF (Espessif IoT Development Framework)*, il quale fornisce una serie di API che permettono di usufruire a pieno delle funzionalità di cui tale microcontrollore è dotato. All'avvio, l'ESP32 effettuerà le seguenti operazioni:

1. Rimarrà in attesa fino a quando non sarà disponibile la rete Wi-Fi che il router dovrà provvedere a creare;

2. Si connetterà al server creato dallo script LUA sul router;

3. Inizializzerà la connessione MQTT con il broker remoto;

4. Abiliterà la modalità promiscua della scheda di rete Wi-Fi incorporata.

4.2.1 Servizio GPS

Ogni qual volta che una nuova posizione GPS verrà resa nota al microcontrollore, questo dovrà solamente occuparsi di sostituirla a quella precedente. Essendo in ambiente multi-thread è stato necessario proteggere l'accesso alla variabile che contiene l'ultima posizione GPS nota, la quale rappresenta una risorsa contesa tra più task, attraverso delle opportune primitive di acquisizione e rilascio di un *mutex*, messe a disposizione dal framework.

4.2.2 Modalità promiscua

Quando viene abilita questa modalità della scheda di rete Wi-Fi, della quale l'ESP32 è dotato, sarà possibile analizzare tutti i MAC frame di management e di dati, ma non quelli di controllo, poiché altrimenti sarebbe possibile sfruttare tali dispositivi per condurre degli attacchi che potrebbero compromettere la privacy degli utenti. Per tale ragione, nel momento in cui il microcontrollore è posto in modalità promiscua, il framework effettuerà a priori un primo filtraggio, scartando tutti i control frame.

Dato che in questo lavoro si è interessati esclusivamente al contenuto informativo dei management frame di tipo `Probe Request`, sono stati filtrati tutti quei frame che, riferendoci alla Figura 2.4, *non* sono caratterizzati da un campo `Frame Control` che contiene i seguenti valori per i sotto-campi `Type` e `Sub-Type`:

- `Type` = `0x00`
- `Sub-Type` = `0x04`

Da ogni probe request, saranno successivamente estratte le informazioni ritenute utili, ovvero: *Indirizzo MAC*, *Sequence Number*, *RSSI*, *Footprint* e *SSID* (il significato di ciascuno di questi campi verrà spiegato nel dettaglio nella sezione successiva).

4.2.3 MQTT

Una volta che dal probe sono state estratte le informazioni di interesse sarà possibile renderle note al server remoto effettuando un'operazione di *publish* di un messaggio attraverso la connessione MQTT, precedentemente inizializzata.

Il messaggio sarà una stringa JSON così definita:

```
{
"router": string,
"chip":  string,
"lat": string,
"lng": string,
"macAddress": string,
"unique": boolean,
"footprint": string,
"sequenceNum":   integer,
"ssid": string,
"rssi":   integer,
"timestamp": string
}
```

Figura 4.3: Formato della stringa JSON trasmessa dall'ESP32.

Il significato dei diversi campi viene riportato di seguito:

1. `router:` IMEI del router, ottenuto attraverso la connessione stabilita con il server scritto in LUA;
2. `chip:` indirizzo a 6 byte che identifica univocamente l'ESP32;

3. `lat:` latitudine dell'ultima posizione nota;
4. `lng:` longitudine dell'ultima posizione nota;
5. `macAddress:` indirizzo MAC della stazione che ha spedito il probe;
6. `unique:` vale `true` se il valore del bit U/L dell'indirizzo MAC è posto a 0, `false` altrimenti;
7. `footprint:` stringa esadecimale rappresentante un digest a 160bit, calcolato tramite SHA-1, su tutti gli IE non variabili (compreso l'SSID) contenuti all'interno del body del probe;
8. `sequenceNum:` numero di sequenza del probe ricevuto;
9. `ssid:` nel caso in cui il probe sia del tipo *direct*, questo campo conterrà il nome della rete ricercata dalla stazione, altrimenti vorrà dire che il probe sarà di tipo *broadcast*, e, in questo caso, tale campo sarà lasciato vuoto;
10. `rssi:` misura della potenza del segnale con cui è stato ricevuto il probe;
11. `timestamp:` data e ora in cui è stato il messaggio è stato spedito dall'ESP32, dato che il server potrebbe riceverlo in un secondo momento.

Infine, per evitare che alcuni messaggi vengano persi oppure che vengano ricevuti più di una volta dal destinatario, è stato scelto di pubblicare sempre messaggi che adottino una politica QoS del tipo "Exactly once", in modo da avere la garanzia che ogni singolo messaggio sia recapitato a destinazione esattamente una sola volta.

4.3 Server remoto

Sul server, implementato sfruttando le potenzialità del framework Spring, saranno concentrati l'insieme di tutti quei servizi che consentiranno di interagire con tutti gli altri componenti del sistema. In particolare, come viene schematizzato nella Figura 4.4, questo dovrà occuparsi di:

1. essere in ascolto su una specifica porta TCP, in modo da poter essere raggiungibile dal servizio GPS offerto dal router;
2. connettersi al broker RabbitMQ ed effettuare la *subscription* ad uno specifico topic, a cui l'ESP32 provvederà ad inviare dei messaggi;
3. esporre una serie di REST API che permettono ad un client di poter accedere ad un sotto-insieme delle informazioni memorizzate all'interno della base dati;

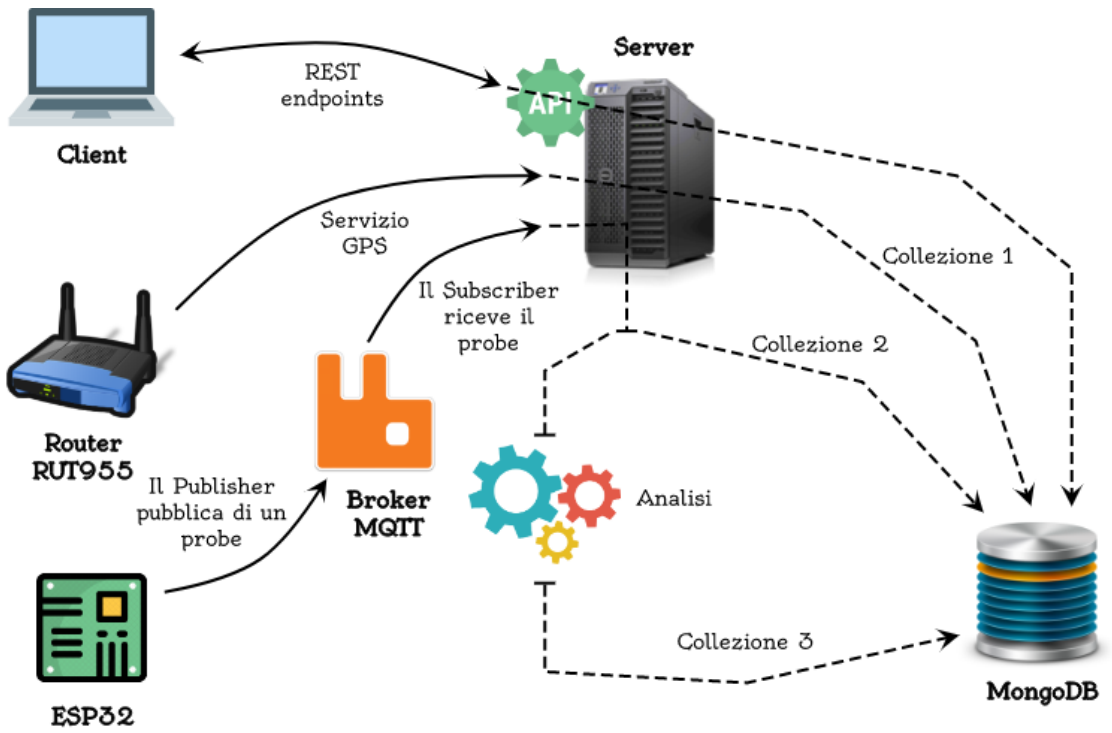


Figura 4.4: Tutte le possibili interazioni tra il server e gli altri componenti del sistema.

4. gestire diverse collezioni di documenti all'interno della base dati MongoDB.

Nelle successive sezioni sarà analizzato nel dettaglio il ruolo coperto dal router per ciascuno di questi servizi.

4.3.1 Servizio GPS

Il server al suo avvio, si metterà in ascolto su una determinata porta TCP, che corrisponderà a quella configurata all'interno della sezione "GPS settings" del router. Nel momento in cui un router si connetterà a tale porta, il server provvederà ad attuare il protocollo descritto nella Sezione 4.1.1. Infine, una volta estratte le informazioni di interesse, esso si occuperà di memorizzarle sotto forma di documento BSON all'interno della `Collezione 1`, come mostrato in Figura 4.4. Queste informazioni saranno utili esclusivamente per tracciare gli spostamenti del mezzo di trasporto su cui è stato installato uno specifico router, identificato tramite il suo IMEI.

4.3.2 Subscriber MQTT

Durante il boot del server verrà effettuata la connessione al broker RabbitMQ attraverso le opportune credenziali di accesso. Nel caso in cui tale procedimento vada a buon fine, verrà effettuata la sottoscrizione ad uno specifico topic, che coinciderà con quello usato dall'ESP32 per pubblicare i messaggi.

Successivamente, ad ogni nuovo messaggio che sarà ricevuto, il server effettuerà due operazioni distinte:

1. mapperà tutte le informazioni contenute al suo interno su di un documento BSON e memorizzerà quest'ultimo su una collezione in MongoDB, identificata con il nome `Collezione 2` nella Figura 4.4;
2. analizzerà le informazioni contenute all'interno del messaggio e cercherà di trovare delle correlazioni con i messaggi precedentemente ricevuti.

Analisi dei messaggi

I messaggi che sono ricevuti avranno il formato mostrato in Figura 4.3 e conterranno:

- delle informazioni relative ad un generico probe request;
- l'informazione relativa alla posizione GPS in cui tale probe è stato "sentito";
- la data e ora in cui il messaggio è stato inviato al server dall'ESP32.

L'insieme di tutte queste informazioni ci permette di essere in grado di stabilire una corrispondenza certa tra la posizione del veicolo di trasporto e il probe che è stato "sentito" dal microcontrollore a bordo di esso. Tuttavia, non è possibile determinare con certezza se il probe ricevuto dall'ESP32 sia effettivamente a bordo del mezzo di trasporto o meno; infatti, sono facilmente individuabili una serie di casi in cui uno o più probe ricevuti non dovranno essere presi in considerazione, ovvero:

- durante ogni sosta presso le fermate di linea di un autobus, saranno incrociati diversi utenti in attesa di qualche altro mezzo. In questo caso, i probe mandati dai dispositivi di tali persone sarebbero catturati dal microcontrollore;
- presso gli incroci, per via dei rallentamenti o delle soste dovute alla presenza di semafori che gestiscono il traffico. In questa situazione, l'ESP32 riceverà anche tutti quei probe provenienti dai dispositivi di persone che si trovano nelle vicinanze dell'autobus, a bordo dei veicoli in coda.
- durante il tragitto dell'autobus potrebbe verificarsi il caso in cui venga considerato nel conteggio dei passeggeri dell'autobus anche il dispositivo di una persona che, utilizzando un proprio veicolo, compia lo stesso percorso del mezzo di trasporto.

Tutti questi casi, sono identificati come *outliers* e devono essere tenuti in considerazione nel momento in cui viene effettuata un'analisi nei confronti dei dati raccolti.

Il formato dei dati contenuti all'interno della `Collezione 3` (Figura 4.4) avrà lo scopo di raggruppare le informazioni relative ai probe mandati dallo stesso dispositivo, anche nel caso in cui questo adotti delle tecniche di randomizzazione per camuffare il suo indirizzo MAC reale. Come possiamo vedere dalla Figura 4.5, le informazioni aggregate, memorizzate all'interno di un documento BSON su MongoDB, sono molto simili a quelle contenute all'interno della stringa JSON (Figura 4.3) trasmessa attraverso la connessione MQTT, ad eccezione del fatto che la maggioranza di tali campi sarà caratterizzata da un set di valori, piuttosto che da un singolo valore. Inoltre, sono stati introdotti alcuni campi aggiuntivi:

- *points:* indica la lista delle coordinate GPS associate a ciascun probe appartenente a tale aggregazione;
- *startTime:* specifica l'istante temporale in cui è stato ricevuto il primo probe da parte di un certo dispositivo;
- *endTime:* specifica l'istante temporale in cui è stato ricevuto il probe più recente da parte di un certo dispositivo;
- *counter:* indica il numero di probe appartenenti alla stesso aggregato;
- *duration:* ottenuto attraverso la differenza tra `startTime` e `endTime`, indica per quanto tempo il dispositivo è stato attivo.

```
{
"router": string,
"chips":  set<string>,
"macAddresses": set<string>,
"footprints": set<string>,
"points": set<GPSPoint>,
"ssids": set<string>,
"timestamps": set<string>,
"startTime": long,
"endTime": long,
"counter": integer,
"duration": integer
}
```

Figura 4.5: Formato del documento BSON contenente le informazioni aggregate riguardanti i probe ricevuti.

Algorithm 1: Pseudo-codice relativo all'analisi di un generico *probe*.

input: L'oggetto *probe*, il quale rappresenta un generico messaggio ricevuto dal server attraverso la connessione MQTT.

```
begin
    /* Viene ricercato all'interno della Collezione3, se è già
       presente un'informazione aggregata che soddisfi i criteri
       specificati                                              */
    stored = findByRouterAndMacAddrOrFootprint(probe.router,
     probe.macAddress, probe.footprint);
    if stored != NULL then
        /* Esiste un riscontro nella collezione. Verifico se il
           timestamp del probe rientri, entro una certa soglia,
           nell'intervallo di tempo in cui è stato ricevuto
           l'ultimo probe appartenente all'aggregato             */
        if probe.timestamp » stored.endTime then
            /* Il probe è stato ricevuto molto dopo
               stored.endTime. Dovrò, quindi, considerarlo come
               un nuovo aggregato contenente: probe.ssid,
               probe.footprint, probe.macAddress, probe.router.  */
            db.store(newEntry);
        else
            /* Il probe è stato ricevuto in un lasso temporale
               compatibile con stored.endTime. Aggiorno le
               informazioni dell'aggregato con quelle contenute
               nell'oggetto probe.                               */
            db.update(stored, probe);
    else
        /* Non esiste un riscontro nella collezione. Inserisco
           una nuova entry aggregata contenente: probe.ssid,
           probe.footprint, probe.macAddress, probe.router       */
        db.store(newEntry);
```

Ogni qual volta verrà ricevuto un nuovo messaggio attraverso la connessione MQTT, il server procederà ad effettuare un'analisi mirata all'individuazione di una qualche correlazione con le informazioni aggregate che sono state precedentemente memorizzate nella base dati. In particolare, il server cercherà di aggregare tali dati in base all'informazione relativa al `MAC address` e alla `footprint` del probe, che sono contenuti all'interno di ciascun messaggio ricevuto. Lo scopo sarà quello di raggruppare tutte le informazioni relative a quei probe che:

- o hanno lo stesso indirizzo MAC globale (ovvero il campo `unique` all'interno del messaggio è settato a `true`);
- o hanno un indirizzo MAC locale (o randomizzato - ovvero, il campo `unique` all'interno del messaggio è settato a `false`) differente, ma saranno caratterizzati dalla stessa `footprint`;

Condizione aggiuntiva a quelle precedenti sarà che, soltanto le informazioni relative ai probe che provengono dallo stesso mezzo di trasporto, e che quindi saranno caratterizzate dallo stesso valore del campo `router`, potranno essere confrontate ed, eventualmente, aggregate tra di loro. Nell'algoritmo 1, la funzione `findByRouterAndMacAddrOrFootprint` si occuperà di verificare che le precedenti tre condizioni siano rispettate.

Successivamente, in base al risultato ottenuto tramite l'esecuzione di tale metodo, sarà stabilito in che maniera effettuare l'operazione di aggregazione. In particolare, l'algoritmo schematizza i seguenti passaggi:

- Nel caso in cui esista un riscontro con un aggregato presente nella collezione (`stored != NULL`), verrà verificata la distanza temporale dell'istante in cui è stato ricevuto il probe (`probe.timestamp`) rispetto all'ultimo aggiornamento dell'aggregato (`stored.endTime`).
  - Se il probe è stato ricevuto in un lasso di tempo superiore ad una certa *soglia*, allora verrà creato un nuovo aggregato con le informazioni contenute all'interno dell'oggetto *probe*;
  - Altrimenti, verrà aggiornato l'aggregato *stored*, avendo cura di:
    - Aggiungere il `macAddress`, la `footprint` e l'`ssid` del probe rispettivamente alle collezioni `stored.macAddresses`, `stored.footprints` e `stored.ssids`;
    - Incrementare il valore del campo `stored.counter`;
    - Aggiornare il valore del campo `stored.endTime` con il `timestamp` del probe.
- Altrimenti, verrà creato un nuovo aggregato con le infomazioni contenute all'interno dell'oggetto *probe*.

4.3.3 REST API

Per quanto riguarda il servizio REST, sono stati esposti quattro *endpoint* che permettono ad un client di ottenere una serie di informazioni, memorizzate all'interno delle diverse collezioni definite su MongoDB.

1. **"/devices":** restituirà tutti quei probe aggregati che sono associati ad uno stesso `router`, escludendo quelli in cui il valore del campo `timestamp` non è compreso nell'intervallo definito dai parametri `from` e `to`, tale che:

$$from < timestamp < to \tag{4.1}$$

   Inoltre, per ragioni legate alla privacy degli utenti, non saranno restituiti i *macAddresses* dei loro dispositivi, ma un digest calcolato tramite la funzione di hash MD5;

2. **"/devices/gps":** restituirà le posizioni gps dei probe, memorizzati all'interno della `Collezione 2`, che sono associati ad uno stesso `router`, escludendo tutti quei probe il cui `timestamp` non sarà compreso nell'intervallo definito dai parametri `from` e `to`, tale che:

$$from < timestamp < to \tag{4.2}$$

3. **"/devices/probes":** restituirà tutti i probe memorizzati all'interno della `Collezione 2`, basandosi sulle stesse regole già descritte al punto 1;

4. **"/positions":** restituirà tutte le informazioni ottenute attraverso il servizio GPS del router (memorizzate all'interno della `Collezione 1`); anche in questo caso, sarà verificata la validità della seguente condizione:

$$from < timestamp < to \tag{4.3}$$

Data la grande mole di dati che potrebbe essere restituita a fronte di una singola richiesta, per ognuno degli endpoint descritti sarà possibile effettuare delle *richieste paginate*, specificando, come ulteriori parametri della richiesta, i campi `nPage` e `nItem`. Questi, infatti, permetteranno di definire il numero massimo di oggetti restituiti per richiesta (*nItem*) ed un numero incrementale (*nPage*), corrispondente al numero della pagina a cui la richiesta si riferisce.

Capitolo 5

Realizzazione del sistema e risultati ottenuti

In quest'ultimo capitolo, sarà discusso di come il sistema implementato è stato installato su dei mezzi di trasporto reali e dei risultati che è possibile apprezzare, dopo circa un mese di utilizzo.

5.1 Messa in campo

Dopo una fase di "rodaggio" iniziale del sistema, durante la quale è stato verificato il suo corretto funzionamento di base e di tutte le sue componenti, è stato successivamente deciso di avviare una fase prototipale, mettendo il sistema in funzione su due autobus di linea di due diverse cittadine del Nord della Francia: Creil e Compiègne.

La procedura di installazione a bordo dei due mezzi si è rivelata molto semplice per via del limitato numero di dipendenze con cui il sistema è stato pensato. Infatti, il numero di componenti, appartenenti all'architettura delineata in Figura 3.1, che necessitano si essere fisicamente presenti sopra il mezzo di trasporto è molto ristretto ed è costituito da:

- uno o più microcontrollori ESP32
- un router RUT955

Ciascuno di questi dispositivi, mostrati in Figura 5.1, al fine di poter svolgere il loro dovere, necessiteranno, principalmente, di una sorgente elettrica attraverso cui poter essere alimentati.

Questa caratteristica assicura all'intero sistema un elevato grado di *flessibilità* che gli permetterà di poter essere installato, senza particolari difficoltà, in disparati ambienti, anche quelli che offrono degli spazi piuttosto limitati.

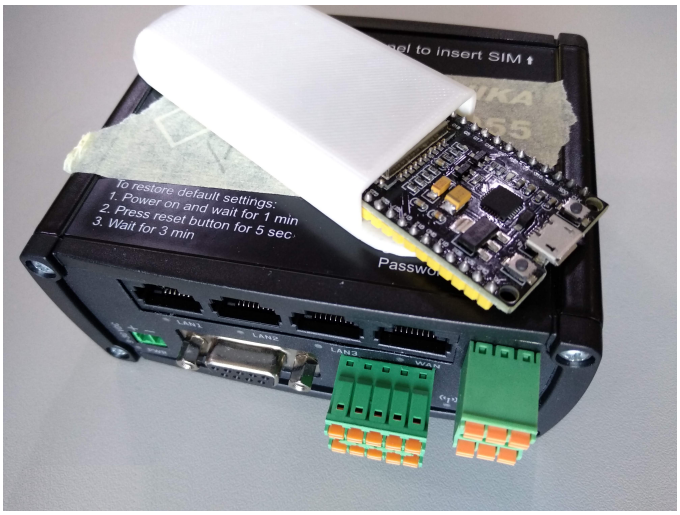


Figura 5.1: Una delle coppie, costituita dal microcontrollore ESP32 e router RUT955, che sono state installate sui veicoli.

Di seguito, vengono mostrate le collocazioni attuali delle coppie di dispositivi installate a bordo dei due mezzi di trasporto.

In particolare, le Figure 5.2, 5.3, 5.5 e 5.4 fanno vedere i dettagli dell'ubicazione del router e del microcontrollore poste a bordo, nel primo caso, dell'autobus in servizio presso la città di Compiègne, mentre, nel secondo caso, sull'autobus attivo presso la città di Creil.

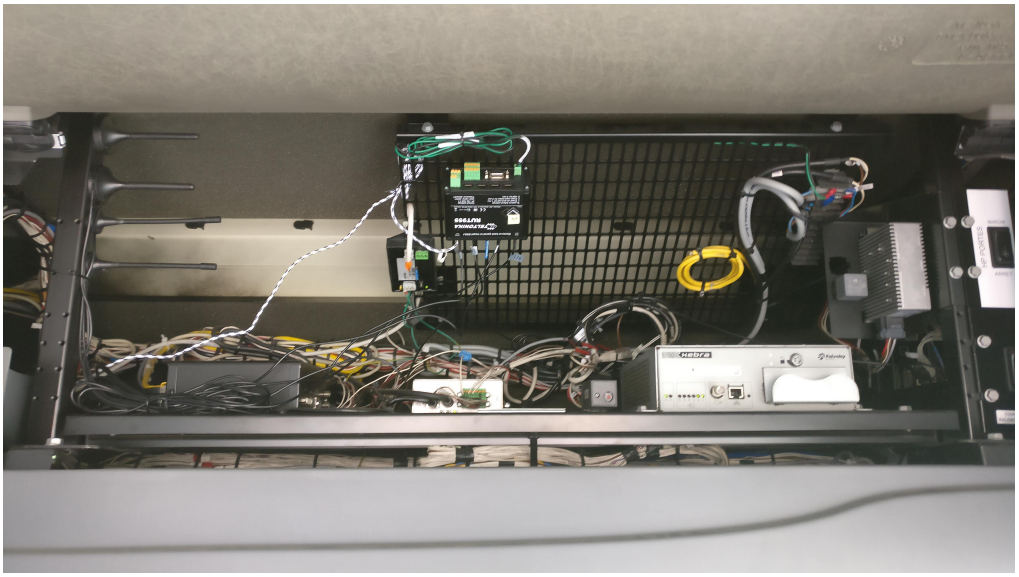


Figura 5.2: Dettaglio relativo all'ubicazione del microcontrollore ESP32 e del router RUT955 sul bus di linea attivo presso la città di Compiègne.

Figura 5.3: Ubicazione del microcontrollore ESP32 e del router RUT955 sul bus di linea attivo presso la città di Compiègne.

Come possiamo vedere, il router, essendo pensato appositamente al fine di essere utilizzato per applicazioni in mobilità, è dotato di un supporto “ad aggancio”, il quale permette al dispositivo di essere facilmente sostituibile ma, al contempo, lo assicura da qualsiasi tipo di urto o movimento improvviso del veicolo.

Un ulteriore considerazione da dover fare riguarda il posizionamento ottimale delle antenne del router. Infatti, come è possibile osservare dalla Figura 3.2, questo dispositivo è caratterizzato da una serie di antenne, di cui:

- due per la connessione LTE

- due per la rete Wi-Fi
- una per il GPS

Ciascuna di queste, deve essere necessariamente collocata in modo che il segnale ricevuto non venga schermato o indebolito e, per fare ciò, è stata molto utile una caratteristica di cui sono dotate tali tipologie di antenne, ovvero un magnete, che ne permette una facile collocazione in qualsiasi superficie metallica dell'autobus.

Figura 5.4: Ubicazione del microcontrollore ESP32 e del router RUT955 sul bus di linea attivo presso la città di Creil.

Il posizionamento che ha richiesto una maggiore attenzione è stato sicuramente quello dell'antenna GPS, in quanto non deve essere possibile che la sua collocazione comporti un'attenuazione del segnale, poiché, in questo caso, la precisione delle informazioni riguardanti la posizione del mezzo sarebbe compromessa.

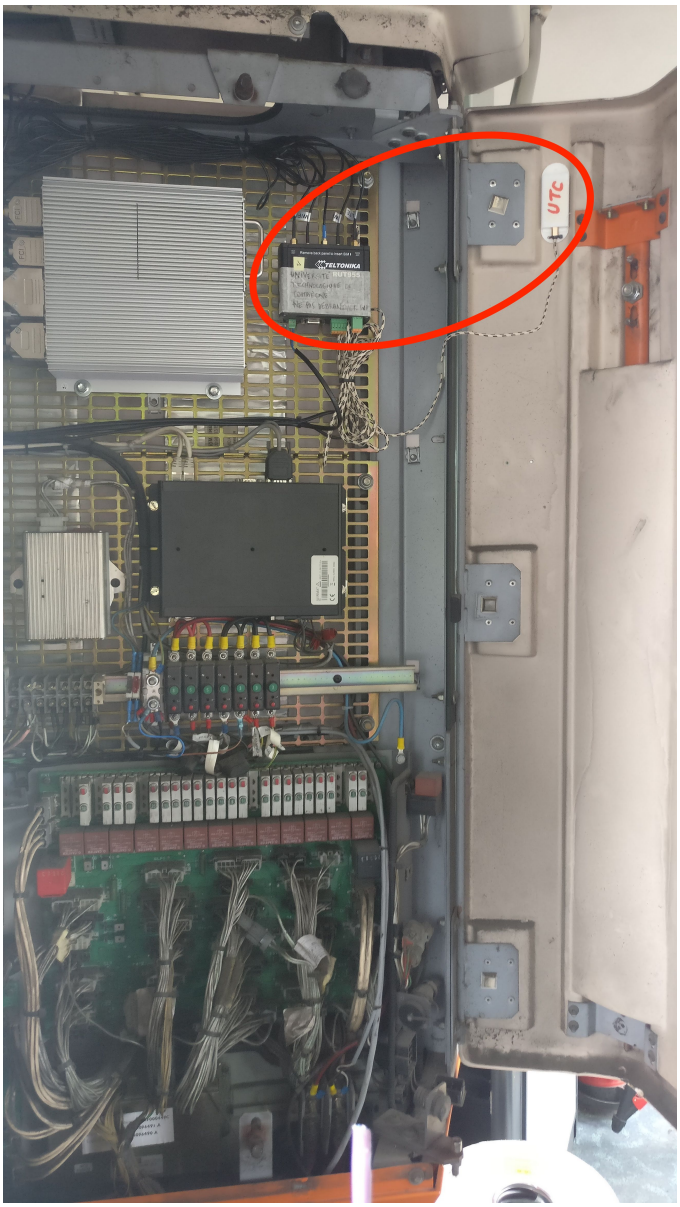


Figura 5.5: Dettaglio relativo all'ubicazione del microcontrollore ESP32 e del router RUT955 sul bus di linea attivo presso la città di Creil.

5.2 Risultati ottenuti

Dopo circa un mese dal rilascio della sua versione prototipale nelle due distinte località francesi di Creil e Compìegne, il sistema ha potuto collezionare oltre un milione di record inerenti i probe spediti dai dispositivi dei passeggeri dei due autobus di linea.

Effettuando un'analisi dei dati raccolti, è stato possibile osservare che:

- circa 35.000 indirizzi MAC, utilizzati per trasmettere i probe in rete, sono reali (ovvero, degli *universally administered addresses*) e quindi, ciascuno di essi, permette di identificare in maniera univoca un dispositivo;
- mentre, sono stati individuati circa 120.000 indirizzi MAC *locally administered*, ovvero i quali sono stati randomizzati dal sistema operativo del dispositivo reale prima di essere inseriti all'interno del probe da trasmettere in rete, come metodo preventivo mirato a salvaguardare la privacy degli utenti. In questo caso, dovranno essere messe in atto delle tecniche in grado di invertire tale processo di mascheramento, come discusso nella sezione 2.4.5.

È necessario sottolineare che, al fine di poter essere in grado di stimare la quantità dei viaggiatori a bordo di un mezzo di trasporto, viene fatta un'assunzione statica per la quale ogni singolo individuo risulta essere in possesso di un solo dispositivo wireless.

Tale considerazione, ci porta a reputare come accettabili alcuni casi particolari che comportano l'alterazione del conteggio effettivo del numero di passeggeri totali, come ad esempio, il caso in cui un individuo abbia con se più di un dispositivo oppure, il caso totalmente opposto in cui non ne possegga nemmeno uno. Infatti, tali *casi limite* sono da considerarsi rari e, in ogni caso, del tutto ininfluenti sul risultato finale, dato che l'obiettivo non è quello di ottenere un informazioni puntuale, ma piuttosto una stima.

Filtrando opportunamente i dati raccolti, sia sulla dimensione spaziale che su quella temporale, è stato possibile ricavare il numero di corse differenti effettuate dai due mezzi, che è pari a circa 10.000. Attualmente, per ognuna di queste corse è possibile individuare con precisione il punto di salita e quello di discesa di un passeggero lungo una certa tratta percorsa da un autobus di linea.

La Figura 5.6 rappresenta un esempio che dimostra il modo in cui il sistema è in grado di individuare un passeggero a bordo di un dato mezzo di trasporto. In questo caso specifico, viene preso in considerazione l'autobus attivo nella città di Compìegne in un qualsiasi giorno di servizio. Tra i diversi risultati che vengono individuati in tale giorno, è stato preso in considerazione un qualsiasi dispositivo

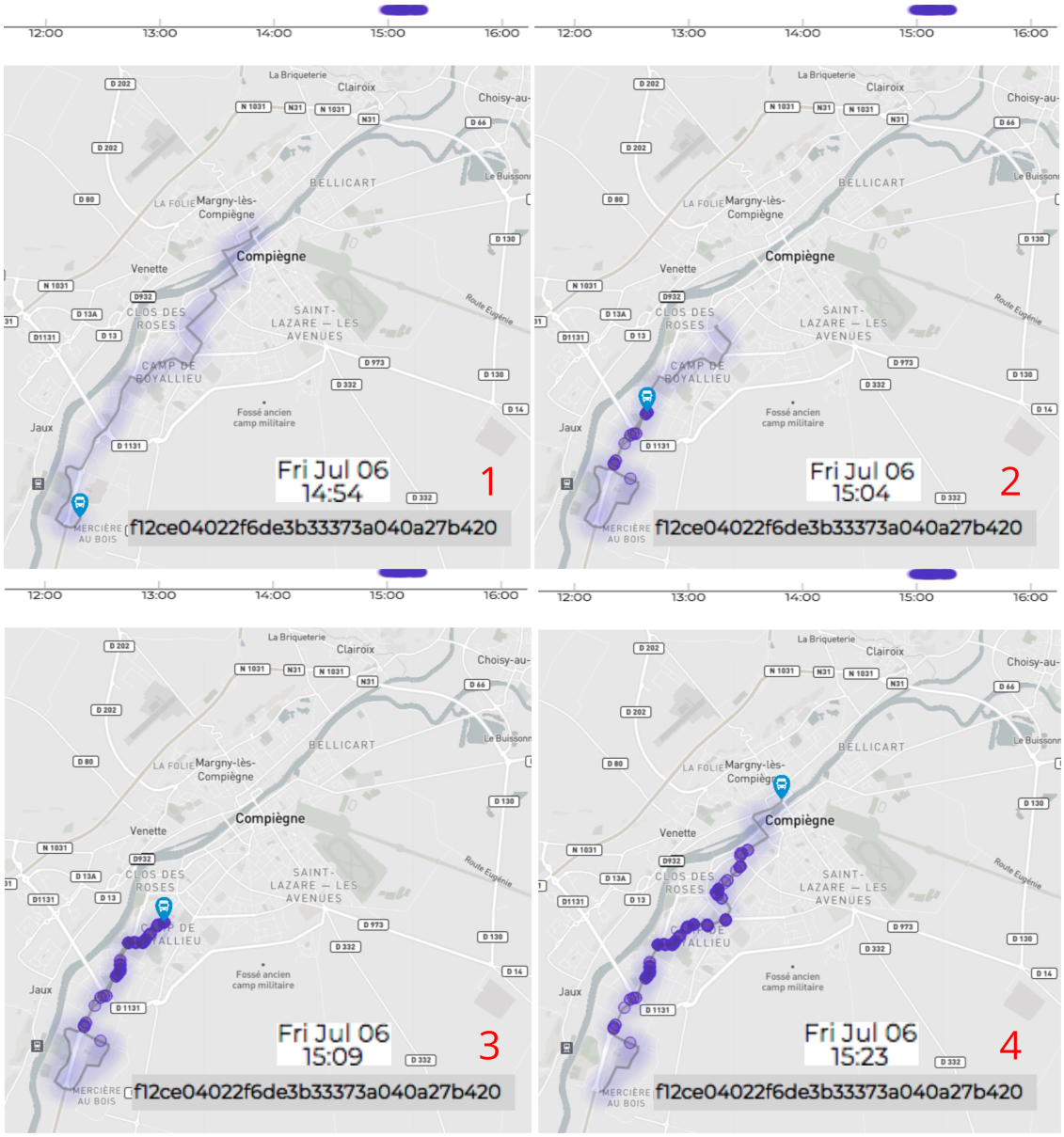


Figura 5.6: Esempio di tracciamento di un passeggero, identificato da un hash MD5 del suo indirizzo MAC, a bordo dell'autobus attivo a Compìegne. Dalla Figura è anche possibile individuare il punto in cui il passeggero è salito a bordo del veicolo, e quello in cui è, successivamente, sceso.

che permetteva di evidenziare in che modo potesse essere possibile tracciare gli spostamenti del suo possessore.

In particolare, la Figura è suddivisa in quattro quadranti, ciascuno dei quali è caratterizzato da:

- una timeline (in alto), che specifica durante quali ore del giorno è stato sentito uno specifico dispositivo;
- una mappa, che mostra in che punto tale dispositivo è stato "sentito";
- un digest MD5 ottenuto dall'indirizzo MAC del dispositivo, il quale costituisce un modo per poterlo identificare senza dover compromettere la privacy degli utenti;
- una data e ora per poter caratterizzare le precedenti informazioni.

Il primo quadrante mostra il tragitto che l'autobus ha percorso partendo dal centro di Compìegne fino alla sua posizione attuale, durante il quale il dispositivo preso in esame non è stato ancora sentito.

Il secondo, il terzo e il quarto quadrante mostrano l'individuazione di una serie di frame riconducibili allo stesso device, siano essi mandati con un indirizzo MAC reale o randomizzato, ciascuno dei quali viene identificato da un disco opaco di colore viola. Come possiamo notare, la frequenza con la quale il dispositivo annuncia la sua presenza in rete, attraverso l'uso dei probe request, è particolarmente alta, permettendoci di poter stabilire con un elevato livello di precisione la fermata in cui il passeggero è salito a bordo dell'autobus e, successivamente, la fermata presso la quale ha deciso di scendere.

Capitolo 6

Conclusioni

Oggigiorno, si cerca di sfruttare, quanto più possibile, le conoscenze acquisite in ambito tecnologico principalmente per la realizzazione di servizi mirati al miglioramento della qualità della vita delle persone. Recentemente, la tendenza è stata quella di fare uso della tecnologia per definire un nuovo concetto di città, detta *smart city*, la quale si pone l'obiettivo di realizzare delle nuove tipologie di servizi, o di migliorare quelli già esistenti, in modo da ridefinire la *user experience* degli utenti.

In particolare, nell'ambito del trasporto pubblico, un servizio che ultimamente ha riscosso grande attenzione è sicuramente quello del conteggio dei passeggeri, che risulta essere fondamentale per la gestione e la pianificazione efficiente del numero necessario di veicoli e della loro distribuzione attraverso il territorio. Nel Capitolo 1, sono state discusse le principali tecniche APC attualmente esistenti sul mercato, nonché il modo in cui l'applicazione di queste possa permette di conoscere, in maniera approfondita, l'utilizzo di una flotta di veicoli in servizio, in base a diverse situazioni: considerando i giorni festivi o quelli feriali, in base al mese dell'anno, a seconda delle condizioni climatiche, e così via...

In questo lavoro, è stata sviluppata una tecnica per il conteggio dei passeggeri basata sulla tecnologia Wi-Fi; per tale ragione, nel Capitolo 2 è stato analizzato il funzionamento di base delle reti che sfruttano tale tecnologia. In particolare, si è visto come i dispositivi wireless, al fine di implementare determinati servizi di gestione, mandano in rete una serie di frame e che, tra tutti questi, i *probe request* possono esseri sfruttati per monitorare il flusso di persone su di un veicolo di trasporto pubblico.

L'architettura che è stata implementata, di cui è stato discusso approfonditamente nei Capitoli 3 e 4, ha permesso di ottenere una serie di vantaggi:

- Il numero totale di componenti, coinvolti all'interno del sistema sviluppato, è molto ridotto, rendendo più facile:

  - L'installazione, poiché, come discusso nel Capitolo 5, il sistema dovrà essere integrato in un ambiente in cui coesistono un diverso numero di tecnologie e che, a seconda delle diverse situazioni, dovrà anche adattarsi alla specifica conformazione del mezzo di trasporto;
  - La gestione e/o la manutenzione, dato che ogni componente è fisicamente indipendente dall'altro.

- Il *costo* della soluzione implementata risulta essere decisamente irrisorio, se confrontato con il costo dell'intero sistema di trasporto;
- La *bontà dei dati raccolti* dal sistema non è basata sul contributo dei viaggiatori. Infatti, il sistema, sfruttando le proprietà intrinseche della tecnologia Wi-Fi, è in grado di implementare il servizio in maniera del tutto *non-invasiva* per gli utenti;
- L'alta *scalabilità* della soluzione. Infatti, è possibile definire il numero di ESP32, facenti parte dello stesso sistema, in maniera proporzionale alle dimensioni del mezzo di trasporto su cui la soluzione sarà installata, senza la necessità di applicare nessuna particolare configurazione aggiuntiva.

Dopo aver implementato il sistema e verificato il corretto funzionamento di base, è stato possibile testarne le effettive potenzialità grazie all'opportunità di installarlo su dei mezzi di trasporto reali, come descritto nel Capitolo 5. I risultati ottenuti permettono di concludere che il sistema, anche se ancora in una fase prototipale e non ancora in grado di fornire un'analisi completa riguardo ai passeggeri presenti sui mezzi, costituisce l'infrastruttura di base che permetterà, attraverso lavori futuri, di poter ricavare una serie di statistiche particolarmente interessanti per la compagnia di trasporto.

Bibliografia

[1] Mathieu Cunche e Célestin Matte. «On Wi-Fi Tracking and the Pitfalls of MAC Address Randomization». In: ().

[2] Julien Freudiger. «How Talkative is Your Mobile Device?: An Experimental Study of Wi-Fi Probe Requests». In: *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. WiSec '15. New York, New York: ACM, 2015, 8:1–8:6. ISBN: 978-1-4503-3623-9. DOI: 10.1145/2766498.2766517. URL: http://doi.acm.org/10.1145/2766498.2766517.

[3] Google. *Android 6.0 Changes*. URL: https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html.

[4] Marco Gruteser e Dirk Grunwald. «Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis». In: *Mobile Networks and Applications* 10.3 (giu. 2005), pp. 315–325. ISSN: 1572-8153. DOI: 10.1007/s11036-005-6425-1. URL: https://doi.org/10.1007/s11036-005-6425-1.

[5] M Handte et al. «Crowd density estimation for public transport vehicles». In: *CEUR Workshop Proceedings* (2014).

[6] Tao Jiang, Helen J. Wang e Yih-Chun Hu. «Preserving Location Privacy in Wireless Lans». In: *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*. MobiSys '07. San Juan, Puerto Rico: ACM, 2007, pp. 246–257. ISBN: 978-1-59593-614-1. DOI: 10.1145/1247660.1247689. URL: http://doi.acm.org/10.1145/1247660.1247689.

[7] Vassilis Kostakos. «Using Bluetooth to capture passenger trips on public transport buses». In: *arXiv* (2008).

[8] A. Di Luzio, A. Mei e J. Stefa. «Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests». In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. Apr. 2016, pp. 1–9. DOI: 10.1109/INFOCOM.2016.7524459.

[9] Jeremy Martin, Erik Rye e Robert Beverly. «Decomposition of MAC Address Structure for Granular Device Inference». In: *Proceedings of the 32Nd Annual Conference on Computer Security Applications*. ACSAC '16. Los Angeles, California, USA: ACM, 2016, pp. 78–88. ISBN: 978-1-4503-4771-6. DOI: 10.1145/2991079.2991098. URL: http://doi.acm.org/10.1145/2991079.2991098.

[10] Jeremy Martin et al. «A Study of MAC Address Randomization in Mobile Devices and When it Fails». In: *Proceedings on Privacy Enhancing Technologies* (2017). URL: https://content.sciendo.com/view/journals/popets/2017/4/article-p365.xml.

[11] Célestin Matte et al. «Defeating MAC Address Randomization Through Timing Attacks». In: *ACM WiSec 2016*. Darmstadt, Germany, lug. 2016. DOI: 10.1145/2939918.2939930. URL: https://hal.inria.fr/hal-01330476.

[12] Cisco Meraki. *White Paper CMX Analytics*. URL: https://documentation.meraki.com/MR/Monitoring_and_Reporting/Location_Analytics.

[13] Bhupinder Misra. *iOS8 MAC randomization - Analyzed!* 2018. URL: https://blog.mojonetworks.com/ios8-mac-randomization-analyzed/.

[14] A. B. M. Musa e Jakob Eriksson. «Tracking Unmodified Smartphones Using Wi-fi Monitors». In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. SenSys '12. Toronto, Ontario, Canada: ACM, 2012, pp. 281–294. ISBN: 978-1-4503-1169-4. DOI: 10.1145/2426656.2426685. URL: http://doi.acm.org/10.1145/2426656.2426685.

[15] I Pinna e Bruno Dalla Chiara. «Automatic passenger counting and vehicle load monitoring». In: 65 (feb. 2010), pp. 101–138.

[16] Damian Roqueiro e Valery A. Petrushin. «Counting People using Video Cameras». In: *International Journal of Parallel Emergent and Distributed Systems* (2007).

[17] Lorenz Schauer, Martin Werner e Philipp Marcus. «Estimating Crowd Densities and Pedestrian Flows Using Wi-fi and Bluetooth». In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. MOBIQUITOUS '14. London, United Kingdom: ICST (Institute for Computer Sciences, Social-Informatics e Telecommunications Engineering), 2014, pp. 171–177. ISBN: 978-1-63190-039-6. DOI: 10.4108/icst.mobiquitous.2014.257870. URL: http://dx.doi.org/10.4108/icst.mobiquitous.2014.257870.

[18] Marco V. Barbera et al. «Signals from the crowd: Uncovering social relationships through smartphone probes». In: (ott. 2013), pp. 265–276.

[19] Mathy Vanhoef et al. «Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms». In: *ACM AsiaCCS*. Xi'an, China, mag. 2016. DOI: 10.1145/2897845.2897883. URL: https://hal.inria.fr/hal-01282900.

[20] Jens Weppner e Paul Lukowicz. «Bluetooth Based Collaborative Crowd Density Estimation with Mobile Phones». In: *IEEE International Conference on Pervasive Computing and Communications* (2013).

[21] Yaoxuan Yuan et al. «Crowd Density Estimation Using Wireless Sensor Networks». In: *Seventh International Conference on Mobile Ad-hoc and Sensor Networks* (2011).