



SAPIENZA  
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

Tesi di Laurea in Ingegneria Elettronica

Utilizzo di modelli di propagazione indoor per  
l'ottimizzazione della copertura WiFi in ambienti  
eterogenei

Relatore:

Maria-Gabriella Di Benedetto

Candidato:

Francesca Ronci

Anno Accademico 2016/2017

## Sommario

### 1. Introduzione

### 2. Modelli di canale per la localizzazione indoor

2.1. Modelli di propagazione dei segnali in ambiente indoor.....	2
2.1.1. Modello One Slope.....	3
2.1.2. Modello Multi-Wall (MW).....	3
2.1.3. Modello Multi-Wall Multi-Floor (MWMF).....	5
2.2. Riepilogo.....	6

### 3. Struttura e risultati dell’algoritmo per la disposizione degli Access Point in ambienti indoor

3.1. Introduzione.....	7
3.2. Fase offline, generazione di fingerprint virtuali e calcolo dei parametri per il modello MW .....	8
3.2.1. Strategie per la creazione di fingerprint virtuali.....	8
3.2.2. Risultati ottenuti.....	10
3.2.3. Osservazioni sulla CDF dell’errore di stima dei parametri.....	11
3.3. Struttura dell’algoritmo.....	15
3.3.1. Generazione della matrice ambiente.....	16
3.3.2. Maglia delle possibili posizioni degli AP.....	18
3.3.3. Creazione delle matrici di attraversamento di muri e porte.....	19
3.3.4. Matrice delle potenze ricevute in ogni cella da ciascun AP.....	20
3.3.5. Criteri per la scelta della posizione del primo AP.....	20
3.4. Risultati dell’algoritmo per i primi due piani del Diet.....	21
3.4.1. Risultati per il primo piano.....	21
3.4.2. Risultati per il secondo piano.....	23
3.5. Riepilogo e Conclusioni.....	25

### 4. Codice MATLAB

4.1. Funzioni.....	26
4.1.1. Generazione matrice ambiente e della griglia di Access Point.....	26
4.1.2. Valuta muri e porte attraversate.....	30
4.1.2.1. Conta numero muri e numero porte attraversati dal segnale di ogni Access Point.....	30
4.1.2.2. Calcolo distanza fra gli AP e tutte le celle della matrice.....	33
4.1.3. Calcolo della potenza ricevuta in ogni cella della matrice.....	34
4.2. Main script.....	35

### 5. Conclusioni e sviluppi futuri.....

### Bibliografia.....

# **1. Introduzione**

Al giorno d'oggi l'aumento consistente degli smartphone, tablet, laptop, e le nuove avanguardie tecnologiche hanno reso indispensabile il nostro bisogno di "essere connessi". Ciò, infatti, è diventato essenziale sia che ci troviamo in grandi spazi aperti come parchi, piazze o addirittura in strada, sia che ci troviamo in spazi chiusi come in uffici, centri commerciali o più semplicemente a casa. Abbiamo bisogno della connessione per comunicare con le altre persone, che possono trovarsi anche all'altro capo del mondo, con gli amici o, addirittura, con i dispositivi che ci circondano, per ottenere qualsiasi tipo di informazione o "essere social", cioè per stare collegati con qualsiasi cosa.

Per questi motivi studi recenti si impegnano a garantire la possibilità della connessione nei vari tipi di ambiente, poiché ogni spazio ha caratteristiche di propagazione differenti. Nonostante ciò possiamo distinguere due macro categorie:

- Ambienti **indoor**,
- Ambienti **free space**.

In particolare, lo scopo di questa tesi è quello di pianificare una distribuzione di sensori che permettano la formazione di una rete WiFi , garantendone l'accessibilità in ogni zona, centrale o marginale, dell'ambiente chiuso, cioè indoor.

Saranno esposti i vari modelli che descrivono la propagazione delle onde nel capitolo 2, verranno specificati i parametri che caratterizzano il modello scelto, l'algoritmo usato per comporre il programma e i risultati della sua applicazione nell'ambiente scelto nel capitolo 3, poi verranno illustrati gli script che compongono il programma compilati con Matlab nel capitolo 4, e infine esporremo le nostre conclusioni e daremo uno sguardo agli sviluppi futuri legati al nostro studio.

## 2 Modelli di canale per la localizzazione indoor

### 2.1 Modelli di propagazione dei segnali in ambienti indoor

Lo scopo di questa tesi è di pianificare al meglio la rete WLAN IEEE 802.11 in modo da assicurarsi localizzazione indoor con la tecnologia Wi-Fi.

La pianificazione della rete deve ottenere la massimizzazione delle prestazioni trovando un compromesso tra i fattori che influiscono sulla qualità della rete, come ad esempio il costo, la copertura, il roaming, il throughput, le interferenze e la sicurezza.

La tecnologia Wi-Fi opera nella banda non licenziata ISM (Industrial Scientific Medical), utilizzata per applicazioni industriali, scientifiche e per apparecchiature mediche, e si trova ad una frequenza di 2.4 GHz, è perciò molto importante modellare in maniera adeguata il canale per non avere importanti riduzioni della prestazione, dovute per esempio alla presenza di muri, porte o altri ostacoli, nel percorso diretto fra trasmettitore e ricevitore.

L'espressione analitica che caratterizza la potenza dell'onda ricevuta è:

$$RSSI = P_{Tx} - P_L \quad (dBm)$$

In cui RSSI = Received Signal Strength Indicator è espresso in dBm ed è la potenza del segnale ricevuto;  $P_{Tx}$  in dBm è la potenza in trasmissione del segnale e  $P_L$  è l'attenuazione di path loss dovuta ai vari elementi ostacolanti.

Sono stati sviluppati diversi modelli per lo studio della propagazione indoor e possono essere divisi in due tipologie:

- Modelli statistici;
- Modelli deterministici.

I **modelli statistici** sono anche detti empirici perché per studiare la propagazione di un segnale è preso in considerazione un ambiente di sviluppo generico. Questi modelli si basano sulla caratterizzazione statistica del segnale ricevuto, effettuata con misure e introdotta in semplici modelli matematici.

Possiamo applicare i modelli statistici sia in sistemi indoor, sia outdoor, ma visto l'argomento di tesi ci interessiamo di quelli indoor i quali consistono nella valutazione del path loss del segnale radio e su un insieme di misurazioni fatte su edifici differenti.

Nei prossimi paragrafi studieremo il metodo **One Slope**, il **Multi-Wall Classico** e il **Multi-Wall Multi-Floor**.

I **modelli deterministici** sono detti "*site specific propagation model*" e si basano sulla teoria della propagazione delle onde, applicandole ad edifici e avendo solamente dettagli sull'ambiente di sviluppo e non misurazioni, come nel caso dei modelli statistici. Quindi inserendo i dettagli concernenti l'ambiente nelle condizioni al contorno delle equazioni di Maxwell otteniamo un modello complesso formato da operazioni matematiche considerevolmente articolate anche per i processori attuali.

Per quanto spiegato sopra approfondiremo i modelli statistici.

### **2.1.1 Modello One Slope**

Il modello One Slope considera una dipendenza lineare tra il path loss (dB) e il logaritmo della distanza  $d$  tra il trasmettitore e il ricevitore considerato. La formula analitica è:

$$L_{os}(d) = l_0 + 10\gamma \log(d) \text{ (dB)}$$

In cui  $l_0$  è il path loss a 1 metro di distanza e  $\gamma$  è detto indice di decadenza della potenza o esponente di path loss. Nel caso dello spazio libero  $\gamma = 2$  e  $l_0 \approx 40.22 \text{ dB}$  alla frequenza di 2.45 GHz

Questo è un modello molto semplice da usare ma non tiene conto delle caratteristiche dell'ambiente di sviluppo, perciò non è molto accurato.

### **2.1.2 Modello Multi-Wall (MW)**

Il modello Multi-Wall è una generalizzazione del One Slope. Questo introduce dei fattori di attenuazione dovuti alle perdite causate dai muri e dalle porte attraversate nel percorso diretto fra trasmettitore e ricevitore, il che mi permette di ottenere una maggiore accuratezza rispetto al modello One Slope. La formula analitica diventa:

$$P_L(d) = L_{os}(d) + M_w \text{ (dB)}$$

Da questa posso vedere che il path loss è lo stesso calcolato prima con l'aggiunta di  $M_w$  che è detta componente di Multi-Wall e si calcola con la seguente formula.

$$M_w = l_c + \sum_{i=1}^I k_{wi} l_i + \sum_{n=1}^N \chi_n l_d + \sum_{n=1}^{N_{fd}} \lambda_n l_{fd} \quad (dB)$$

In cui  $l_c$  è una costante,  $k_{wi}$  è il numero di muri di tipo  $i$  attraversati dal segnale nel percorso diretto,  $l_i$  è la costante di attenuazione dovuta a muri di tipo  $i$  con  $i=1,2... I$ ;  $N_d$  e  $N_{fd}$  sono rispettivamente il numero di porte normali e di porte antincendio attraversate dal segnale nel percorso diretto e  $\chi_n$  e  $\lambda_n$  sono variabili binarie che considerano se la  $n$ -esima porta normale o la  $n$ -esima porta antincendio è aperta, valore 0, o chiusa, valore 1.

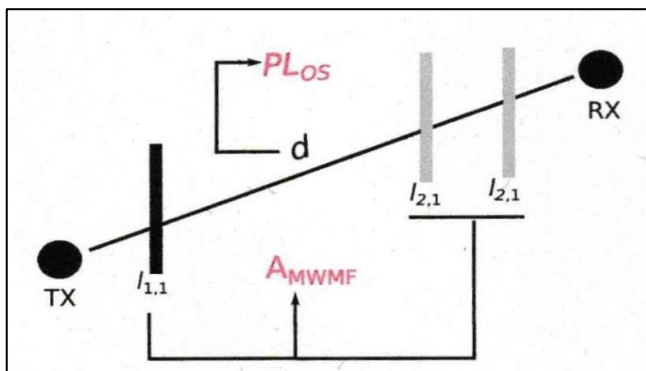
Parametri del modello $M_w$	Significato
$l_c$	Fattore costante(dB)
$l_d$	Attenuazione dovute a porte normali
$l_{fd}$	Attenuazione dovute a porte antincendio
$l_1$	Attenuazione dovuta a muri sottili (dB) (0,20]cm
$l_2$	Attenuazione dovuta a muri medi (dB) (20,40]cm
$l_3$	Attenuazione dovuta a muri larghi (dB) (40,60]cm
$l_4$	Attenuazione dovuta a muri esterni (dB) (60,80]cm

**Tabella 1: parametri del modello Multi-Wall e il loro significato.**

Una forma più generica della componente di MW che tiene conto di più oggetti che contribuiscono all'attenuazione del segnale nel cammino diretto è:

$$M_w = l_c + \sum_{n=1}^{N_{obj}} \sum_{i=1}^{I_n} N_{n,i} l_{n,i} \quad (dB)$$

In cui  $l_c$  è la costante di path loss,  $N_{obj}$  è il numero delle diverse famiglie di oggetti,  $I_n$  è il numero dei tipi di oggetti considerati della famiglia  $n$ ,  $N_{n,i}$  è il numero di oggetti della famiglia  $n$  e del tipo  $i$ ,  $l_{n,i}$  è l'attenuazione dovuta dall'oggetto della famiglia  $n$  e tipo  $i$ .



**Figura 1: Rappresentazione del modello MW ed esempio di connessione TX-RX.**

$$N_{obj} = 2, l_1 = 1, l_2 = 1, N_{1,1} = 1, N_{2,1} = 2.$$

Esiste un'estensione del Multi-Wall, detta **Multi-Wall Linear** che considera una dipendenza lineare dell'attenuazione con la distanza  $d$  fra trasmettitore e ricevitore.

$$L(d) = \alpha d + l_0 + M_w \text{ (dB)}$$

In cui  $\alpha$  e  $l_0$  sono due parametri e  $M_w$  è lo stesso calcolato per il Multi-Wall sopra scritto.

I parametri per il modello Multi-Wall sono stati esposti in [3] e calcolati concentrandosi su un tipo di ambiente indoor, gli uffici. Questi sono caratterizzati, per esempio, da stanze piccole allineate cui si accede da corridoi con porte normali e/o antincendio e spesso da muri sottili di separazione fra le varie stanze.

Si è anche studiata la deviazione standard dell'errore di misura dei parametri illustrati in [3] per i modelli sopra descritti e si può notare come il Multi-Wall Classico e il Multi-Wall Linear abbiano valori quasi uguali, ma cosa più importante, sono molto migliori del modello One Slope.

Modelli di canale	Ambiente uffici
One Slope	7.4
Multi Wall Classico	4.3
Multi Wall Linear	4.4

Tabella 2: deviazione standard dell'errore di misura dei parametri in [3].

Per questo motivo in seguito adotteremo il modello del canale Multi-Wall Classico.

### **2.1.3 Modello Multi-Wall Multi-Floor (MWMF)**

Come propone il nome, questo modello è un'estensione del Multi-Wall e introduce il fattore piano per il calcolo del path loss, quindi lo valutiamo in 3D e non solo in 2D, perciò cerchiamo la potenza dei segnali ricevuti attraverso i piani limitrofi a quello in cui si trova il ricevitore.

La formula analitica che descrive il path loss concernente il MWMF è:

$$PL_{MWMF} = PL_{OS} + A_{MWMF} \text{ (dB)}$$

In cui  $PL_{OS}$  è lo stesso espresso in precedenza:

$$PL_{OS}(d, \gamma) = L_{OS}(d) = l_0 + 10\gamma \log(d) \text{ (dB)}$$

Il termine  $A_{MWMF}$  fornisce l'attenuazione dovuta all'attraversamento da parte del segnale nel percorso diretto fra trasmettitore e ricevitore ed è formulato così:

$$A_{MWMF} = l_c + \sum_{n=1}^{N_{obj}} \sum_{i=1}^{I_n} N_{n,i} l_{n,i} + N_f^{\left[\frac{N_f+2}{N_f+1}b\right]} l_f \quad (dB)$$

In cui  $l_c$  è la costante di path loss,  $N_{obj}$  è il numero delle diverse famiglie di oggetti,  $I_n$  è il numero dei tipi di oggetti considerati della famiglia  $n$ ,  $N_{n,i}$  è il numero di oggetti della famiglia  $n$  e del tipo  $i$ ,  $l_{n,i}$  è l'attenuazione dovuta dall'oggetto della famiglia  $n$  e tipo  $i$ ,  $N_f$  è il numero dei piani attraversati,  $l_f$  è il valore di attenuazione dovuto all'attraversamento dei piani e  $b$  è un parametro empirico della propagazione 3D.

In seguito vedremo come i valori così calcolati e applicati per stimare la potenza ricevuta in un punto in ambiente indoor genereranno un errore accettabile rispetto alla potenza di segnale effettivamente misurata in quel punto.

## 2.2 Riepilogo

Sono stati descritti dei modelli di propagazione che ci permettono di calcolare la forza del segnale e la dispersione dovuta all'ambiente indoor in analisi. Questi dati sono fondamentali per la pianificazione di reti WiFi in ambienti indoor che siano caratterizzate da buoni valori di copertura, e sicurezza.

I modelli statistici hanno bisogno di un gran numero di misure e sono relativi ad ambienti generici, al contrario dei modelli site-specific che si basano sul sito di sviluppo e non necessitano di molte misurazioni ma di informazioni specifiche dell'ambiente in considerazione.

La prima categoria di modelli è meno accurata, ma di più semplice applicazione, al contrario della seconda categoria, la quale ci da una predizione dei modelli di propagazione nell'ambiente di sviluppo molto più precisa a discapito del costo più elevato del progetto e della facilità di applicazione, in quanto si considerano le equazioni di Maxwell. Proprio per questo motivo il secondo modello è a volte impraticabile e quindi si è scelto, per il lavoro di tesi, di sfruttare nell'algoritmo un modello di propagazione statistico, e in particolare il Multi-Wall.

Così partendo dalle misurazioni iniziali e dalla piantina dell'ambiente indoor di sviluppo si può studiare e prevedere la propagazione delle onde elettromagnetiche nel suddetto ambiente.



## 3 Struttura e risultati dell'algoritmo per la disposizione degli Access Point in ambienti indoor

### 3.1 Introduzione

Lo scopo di questa tesi è di pianificare una rete wireless WiFi 802.11, che per grande diffusione della sua strumentazione e anche per il costo accessibile, è particolarmente adatta a questo progetto.

L'obiettivo dell'algoritmo studiato nei prossimi paragrafi è di formare una rete ottimale di sensori che ci permette di ottenere servizi di localizzazione in ambiente indoor. Per **disposizione ottimale** si intende la combinazione delle posizioni dei sensori in modo da ottenere una copertura completa dell'ambiente indoor di sviluppo utilizzandone il minor numero possibile. La **copertura completa** consiste nell'avere, in qualsiasi area dell'edificio oggetto di studi, una potenza di segnale ricevuta maggiore o uguale di un valore di soglia. Questo valore di soglia è stabilito in base alle caratteristiche necessarie per il collegamento che si vuole impostare.

Giacché consideriamo la rete Wi-Fi 802.11, il termine sensori può essere associato agli **Access Point** (AS), inoltre saranno considerati i segnali trasmessi in banda ISM (Industrial, Scientific e Medical) la quale lavora alla frequenza di 2.4 GHz.

Cominceremo prendendo come input la piantina dell'edificio del Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni dell'università La Sapienza di Roma, e l'algoritmo produrrà come output una disposizione ottimale degli Access Point nella struttura sopra considerata. Perciò sarà necessario estrapolare dall'unico input tutte le informazioni necessarie per l'analisi della propagazione del segnale, come ad esempio la presenza di muri più o meno spessi, porte aperte o chiuse e altri oggetti che provocano l'attenuazione del segnale trasmesso.

Quindi partendo dal lavoro di tesi della collega Valentina Piccolo [1], cercheremo di migliorarne l'accuratezza adottando i parametri ottenuti dal modello Multi-Wall; vedremo come vengono calcolati questi valori e come variano in base al cambiamento di alcuni dettagli nelle misure effettuate in una fase precedente all'algoritmo detta fase offline, come per esempio la differente orientazione del dispositivo o dell'utente durante la misura.

**MATLAB** è un linguaggio ad alto livello del calcolo tecnico che fornisce un'interfaccia intuitiva ed efficace per sviluppare algoritmi, per questo motivo è stato scelto per creare uno script utilizzato nel nostro progetto.

## **3.2 Fase offline, generazione di fingerprint virtuali e calcolo dei parametri per il modello MW**

La fase offline è la base di partenza del nostro algoritmo e consiste nel fare il *fingerprinting*, cioè si crea una mappa radio dell'area di sviluppo considerando i valori di RSS, in alcuni punti detti *Reference Point* (RP), provenienti da AP situati nelle varie posizioni in cui potrebbero trovarsi, come ad esempio sui muri dei corridoi, oppure solamente nei muri delle stanze o in entrambi.

Grazie a questa fase noi siamo in grado di stimare, con buona approssimazione, la posizione di un dispositivo che riceve un segnale dal l-esimo AP. Infatti, osservando la RSS cerco il valore al n-esimo RF che più si avvicina a questa.

L'accuratezza e la complessità dell'algoritmo studiato in questa tesi dipendono principalmente dalla pianificazione della fase offline, cioè dal numero di misure prese e dalla quantità di RP, ma anche dal modello di stima usato per la fase online. Si è cercato di fare misure multiple in modo da contrastare alcuni effetti variabili di propagazione come la dissolvenza, lo *shadowing*, i cammini multipli, l'impatto umano dovuto alla presenza di persone interposte tra il cammino diretto del segnale e il ricevitore e inoltre anche l'orientazione del dispositivo e o dell'utente.

Una soluzione per fare meno misure è di generare *fingerprint* virtuali. Ciò è possibile interpolando una serie di rilevamenti iniziali con un modello di propagazione indoor con parametri *site-specific*.

Quindi lo scopo di questa fase è trovare un compromesso tra l'accuratezza del modello Multi-Wall e la quantità di Reference Point, e di conseguenza del numero di misure effettuate nell'ambiente di sviluppo indoor.

### **3.2.1 Strategie per la creazione di fingerprint virtuali**

Il modello MW e può essere utilizzato per generare fingerprint virtuali nell'ambiente indoor di sviluppo.

Partendo dall'area d'interesse  $A$ , dal numero  $N$  di RP considerati e dal numero  $L$  di Access Point posti nei punti in cui possiamo metterli, all'interno di  $A$ , il nostro obiettivo è di stabilire per ogni  $RP_n$ ,  $K * L * 1$  fingerprint  $\hat{s}_{n,k}$ , dove  $\hat{s}_{l,n,k}$  è la potenza stimata RSS per un determinato collegamento TX-RX, dal l-esimo AP al n-esimo RP nella k-esima

direzione considerata, dove  $k=1,2,3,4$  indica una delle quattro sensi possibili. Per questi motivi il modello MW deve essere calibrato su di una serie di misurazioni reali.

L'accuratezza del modello dipende principalmente dalla strategia adottata per acquisire le misure e dal numero di volte che le ripeto.

Si possono distinguere quattro strategie:

1. **Strategia I, No Fitting:** Si basa sull'uso dei parametri di propagazione dei segnali in ambienti indoor generici presi dalla letteratura, cioè senza sfruttare i risultati delle misure fatte sul luogo di sviluppo. Ci si aspetta una scarsa accuratezza e semplicità di applicazione in quanto non comporta alcuna rilevazione in  $A$ .
2. **Strategia II, Mockup Fitting:** Si basa sui parametri ottenuti in un ambiente modello  $A'$  per le particolari orientazioni  $k$ , che sono usati per generare fingerprint virtuali nell'ambiente di sviluppo  $A$ . Il modello  $A'$  viene scelto con caratteristiche topologiche molto simili all'ambiente reale  $A$ , cioè affini nella struttura complessiva e nel numero e tipo di oggetti presenti nell'ambiente. Questa strategia offre il vantaggio di poter generare fingerprint virtuali in tutti gli ambienti reali, simili per caratteristiche topologiche e di propagazione all'ambiente modello  $A'$ , in cui si vuole studiare la diffusione dei segnali.
3. **Strategia III, Environment Fitting:** Si basa sull'acquisizione di una serie di misure, svolte per ogni orientazione  $k$ ,  $N_{real,k}$  e per ogni AP dell'ambiente di sviluppo  $A$ . Tutte queste misure sono ottimizzate globalmente per ottenere un insieme di parametri di modello, validi per la  $k$ -esima direzione e per tutti gli AP.
4. **Strategia IV, Specific AP Fitting:** Si basa sull'acquisizione di una serie di misure, svolte per ogni orientazione  $k$ ,  $N_{real,k}$  e per ogni AP dell'ambiente di sviluppo  $A$ , proprio come nella Strategia III. La differenza con la tecnica precedente sta nell'ottimizzazione poiché, in questo caso, i parametri non sono calcolati in modo globale bensì in maniera individuale. Cioè ci saranno specifici parametri per ciascun AP nella  $k$ -esima direzione considerata. Si può immaginare che questa strategia risulterà la più accurata.

Avendo elencato le caratteristiche delle quattro tipologie di strategie possiamo riconoscere che nel lavoro di Valentina [1] è stata applicata la I, mentre, come vedremo dopo, in questa tesi si è scelta la strategia III. In seguito potremo osservare l'effettiva differenza dei risultati dovuta all'utilizzo di una strategia rispetto ad un'altra.

Per la strategia III, considerato il modello Multi-Wall, tutte le misure reali  $M$ , e per ogni  $m$ -esima misurazione (con  $m = 1,2... M$ ), un set di parametri topologici  $T_m$ , il metodo dei minimi quadrati ci consente di ottenere un insieme di parametri  $S$  che ottimizza i valori calcolati.

Il problema dell'ottimizzazione dei parametri è descritto così:

$$\{S\}_{opt} = \underset{\{S\}}{argmin} \left\{ \sum_{m=1}^M |P_m - (EIRP - PL_{MW}(d_m, T_m, S))|^2 \right\}$$

In cui, per la m-esima misura,  $P_m$  è la potenza misurata al ricevitore considerando il trasmettitore ad una distanza  $d_m$  che irradia in maniera isotropa una potenza EIRP (Effective Isotropic Radiated Power), e  $PL_{MW}(d_m, T_m, S)$  è il path loss calcolato con le formule in precedenza descritte.

### 3.2.2 Risultati ottenuti

Adottando la strategia III, le rilevazioni sono state fatte nell'edificio del Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni (DIET), in particolare nel I e nel II piano del palazzo.

È stata presa in considerazione l'area di grandezza  $12 * 42 m^2 A$ , per ciascun piano. Inoltre sono stati individuati  $N_1 = 65$  Reference Point al primo piano ed  $N_2 = 69$  Reference Point al secondo. Qui di seguito vediamo la piantina del primo piano con le posizioni degli RP, punti verdi, le posizioni degli AP considerati, punti blu, e le quattro direzioni considerate rappresentate dalle frecce numerate per comodità.

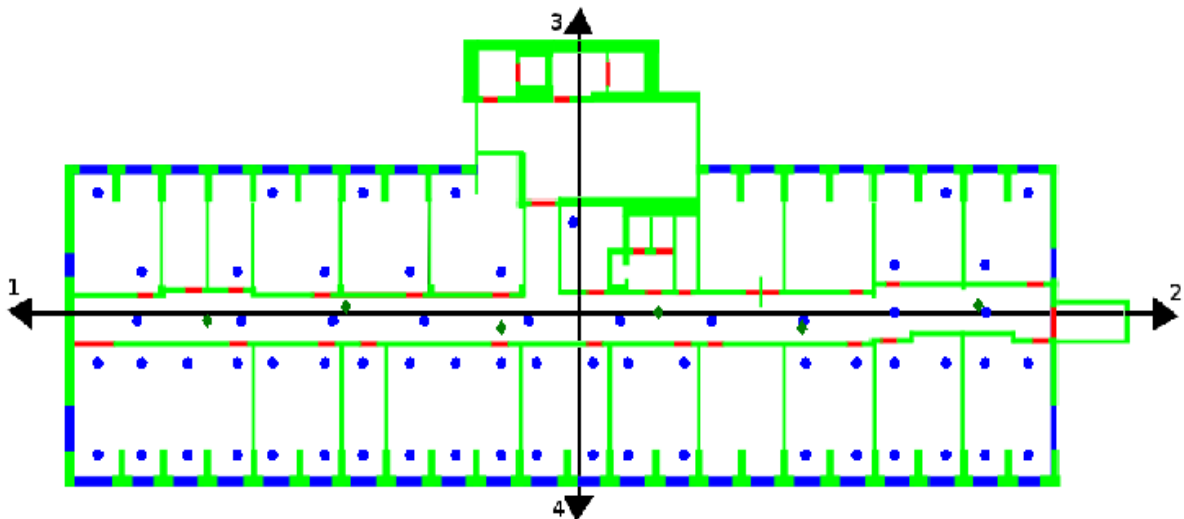


Figura 2: Esempio per il primo piano. I puntini blu rappresentano i RP, i puntini verdi gli AP considerati e le frecce la k-esima orientazione dell'utente o del dispositivo al momento della misura.

Negli RP sopra mostrati sono state fatte le misure e ripetute cinque volte per ogni orientazione e quindi, nel caso Multi-Wall, per tutti i RP ci sono state  $q = 5 * 4 * L_1 * 1$  rilevazioni per il primo piano e  $q = 5 * 4 * L_2 * 1$  nel secondo piano. Abbiamo ottenuto i parametri mostrati di seguito svolgendo l'ottimizzazione dei valori di tutti gli AP, e non considerando un set di parametri per ogni AP perché sarebbe stato dispendioso e non necessario per il nostro studio.

Per il metodo con cui sono state svolte le misure otteniamo la conferma che stiamo lavorando con la strategia III, prima descritta.

I parametri usati nella tesi [1] li troviamo esposti nell'articolo [3].

Piano considerato	Parametri del modello	Significato
Piano I	$l_c = 62.73 \text{ dB}$ $l_2 = 2.84 \text{ dB}$ $l_d = 4.37 \text{ dB}$ $10\gamma = 24.89 \text{ dB}$	Fattore costante Attenuazione dovuta a muri medi (20,40] cm Attenuazione dovuta alle porte antincendio Esponente di propagazione
Piano II	$l_c = 65 \text{ dB}$ $l_2 = 2.7 \text{ dB}$ $l_d = 3.21 \text{ dB}$ $10\gamma = 22.07 \text{ dB}$	Fattore costante Attenuazione dovuta a muri medi (20,40] cm Attenuazione dovuta alle porte antincendio Esponente di propagazione

**Tabella 3: Valori dei parametri calcolati nel DIET per il modello MW.**

### **3.2.3 Osservazioni sulla CDF dell'errore di stima dei parametri**

È molto interessante studiare la CDF (*cumulative distribution function*) dell'errore del modello poiché ci mostra l'effettivo distacco del valore stimato da quello realmente misurato.

$$\epsilon_k = |\hat{S}_{l,n,k} - S_{l,n,k}|$$

Questa espressione indica con  $\epsilon_k$  il valore dell'errore di stima in modulo trovato comparando il valore RSSI  $\hat{S}_{l,n,k}$  stimato per  $l$ -esimo AP e  $n$ -esimo RP e la  $k$ -esima direzione dell'utente o del dispositivo e quello della RSSI  $S_{l,n,k}$  realmente misurato.

L'immagine seguente ci mostra gli effetti degli errori di stima dei parametri del modello MW su un valore medio del path loss  $\overline{PL}_k$ . Si evince l'effettivo miglioramento dell'errore nel caso in cui nella stima è presa in considerazione l'orientazione della misura effettuata e ciò si riflette in una maggiore accuratezza per il nostro algoritmo.

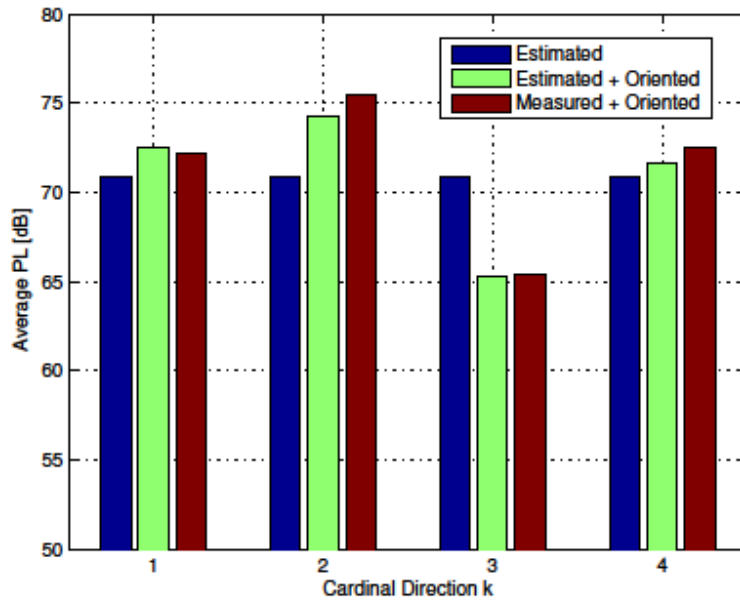


Figura 3: Impatto dell'orientazione sulle misure e sulle stime di path loss.

Possiamo studiare la CDF in funzione dell'errore di stima  $\epsilon$  (dB) al variare di  $\rho=[0.01,1]$  per le varie strategie esposte del paragrafo precedente;  $\rho$  è un parametro che determina quante misurazioni sono state fatte per la procedura di ottimizzazione per ogni l'orientazione.

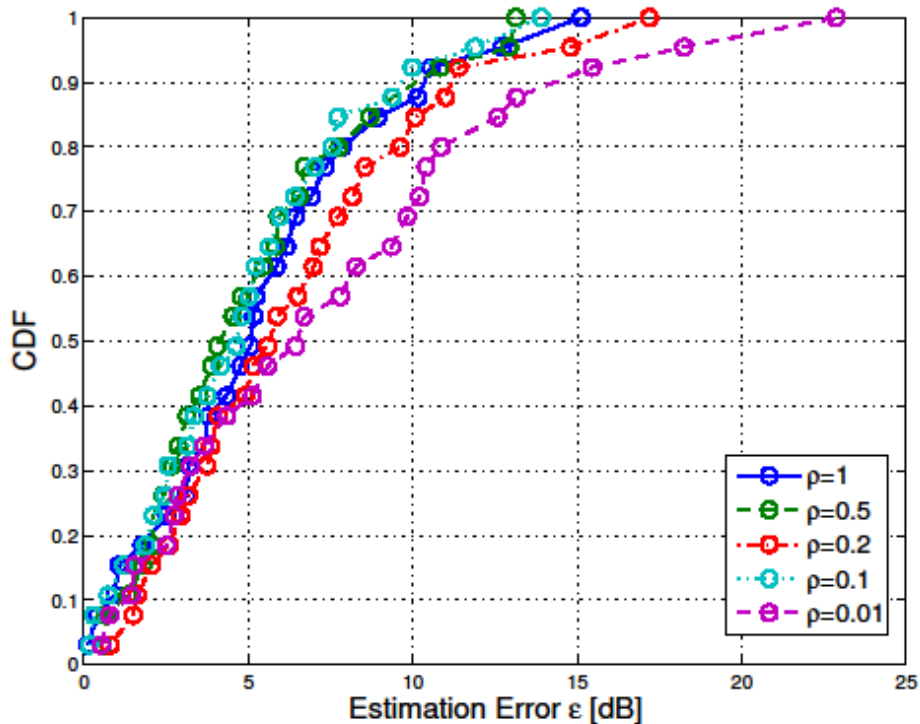


Figura 4: CDF dell'errore AP considerato usando la strategia II.

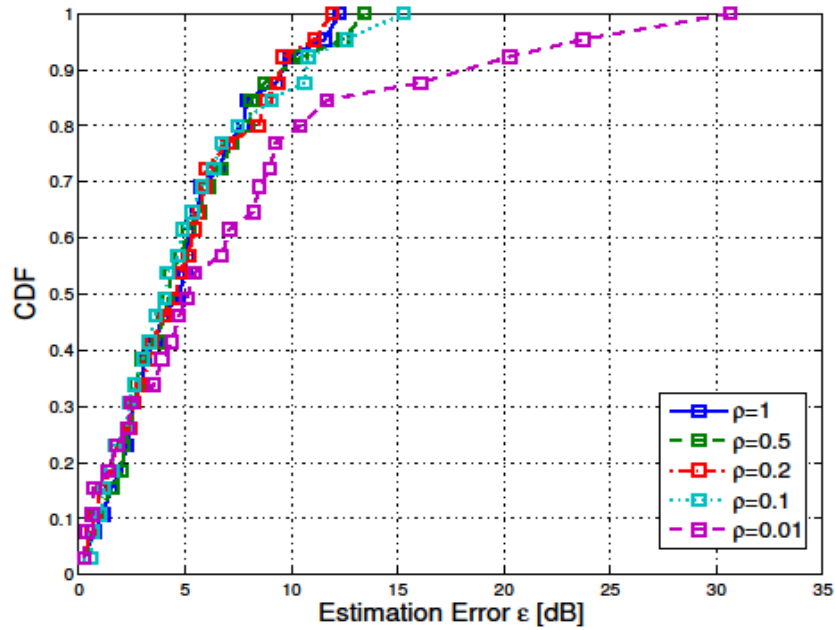


Figura 5: CDF dell'errore AP considerato usando la strategia III.

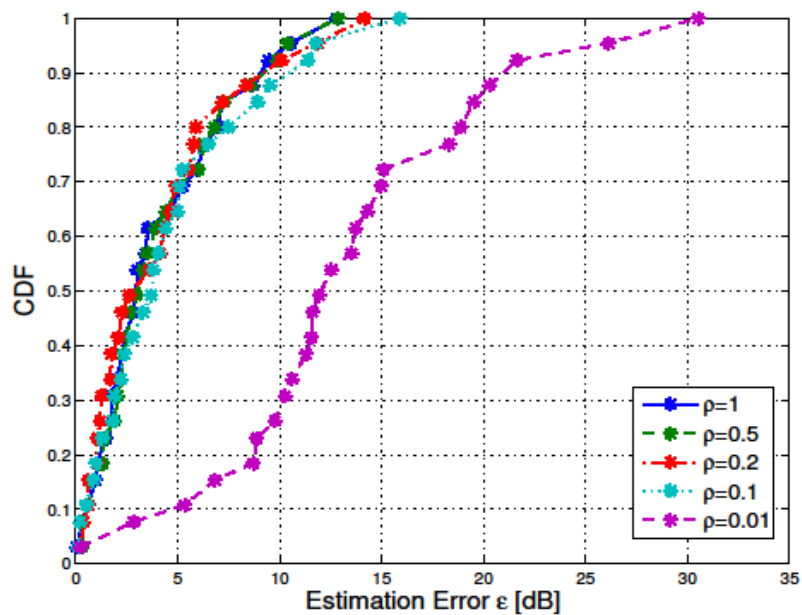


Figura 6: CDF dell'errore per AP considerato usando la strategia IV.

Deduciamo dalle immagini come le prestazioni delle strategie, migliorino all'aumentare del parametro  $\rho$ , anche se, per esempio, dal valore  $\rho = 0.1$  al  $\rho = 1$  non ho una sostanziale differenza, quindi è importante ripetere più volte le misure, ma non è necessario farlo troppe volte, perché tanto otterremmo risultati paragonabili fra loro.

La complessità delle strategie III e IV sono paragonabili ed anche i loro risultati, lo stesso non si può dire per la strategia II che risulta sicuramente più semplice delle altre due, anche se mantiene un buon livello di accuratezza.

Possiamo fare un confronto fra le quattro strategie mettendoci nel caso di mezzo, cioè considerando  $\rho = 0.5$ , e riportando in un unico grafico i risultati.

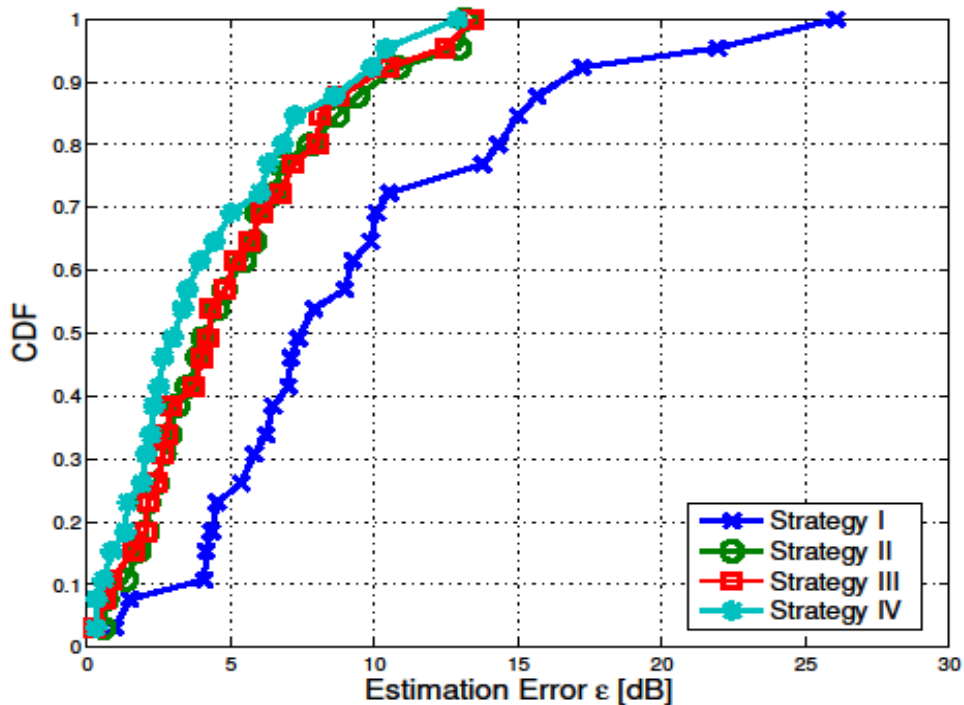


Figura 7: CDF dell'errore per un dato AP con  $\rho=0.5$  per le 4 strategie.

A colpo d'occhio vediamo subito lo scostamento della strategia I, dalle altre strategie. Possiamo inoltre vedere che la Strategia IV è la migliore in termini di accuratezza e le Strategie II e III hanno risultati molto simili.

Appurato che la strategia I, cioè quella usata per giungere ai risultati riportati in [1], è affetta da un errore consistente rispetto alle altre strategie, questo lavoro apporta un netto miglioramento considerando per la stima dei valori dei parametri la strategia III.

Per questi motivi ci aspetteremo una maggiore accuratezza dei risultati rispetto a quelli del lavoro di partenza.



### **3.3 Struttura dell'algoritmo**

Dopo aver completato la fase offline e trovato i valori dei parametri andiamo a studiare l'algoritmo che ci permetterà di ottenere una distribuzione ottimale di Access Point per pianificare una rete Wi-Fi ed ottenere una copertura completa dell'ambiente considerato.

Specifichiamo la struttura dell'algoritmo nei seguenti passi:

- 1) Si stabilisce se l'ambiente di sviluppo è controsoffittato o no, in base a questa caratteristica si calcolano i dati utilizzati dall'algoritmo che si evincono dalla piantina, cioè :
  - a. Generazione della matrice ambiente;
  - b. Determinazione delle possibili posizioni degli AP;
  - c. Formazione della matrice contenente l'attraversamento dei muri da parte del segnale proveniente da ciascun AP e diretto verso tutte le celle della matrice ambiente;
  - d. Formazione della matrice contenente l'attraversamento di porte da parte del segnale proveniente da ciascun AP e diretto verso tutte le celle della matrice ambiente;
  - e. Generazione della matrice che contiene la distanza di ogni cella dalla possibile posizione dell'AP.
- 2) Ora si sceglie in modalità random la posizione del primo AP tra tutte quelle possibili.
- 3) Messo il primo AP, si trova la matrice di dimensione  $M \times N$ , in cui sono inseriti i valori della potenza percepita in ogni cella. In sostanza questa matrice mostrerà la stima del raggio di azione del primo AP.
- 4) La posizione del secondo AP viene stabilita scegliendo, tra tutte le posizioni possibili che sono rimaste, quella che mi permette di ottenere il maggior numero di celle aventi una potenza ricevuta sopra il valore di soglia.
- 5) Per il posizionamento degli altri AP si ripete il punto 4) fino a che il numero di celle, in cui il valore della potenza ricevuta è maggiore di quello soglia, è pari al 100%, cioè fino a quando non otteniamo la copertura completa dell'ambiente di sviluppo.

Alla fine del processo otterremo in output le posizioni degli AP necessari per la copertura completa e anche la matrice contenente le potenze ricevute dagli AP in ogni cella della matrice ambiente.

### 3.3.1 Generazione della matrice ambiente

Iniziamo a descrivere i passi per la creazione della matrice partendo dal nostro input, cioè le piantine del primo e del secondo piano del Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni dell'università La Sapienza di Roma in cui sono state raccolte le misure.

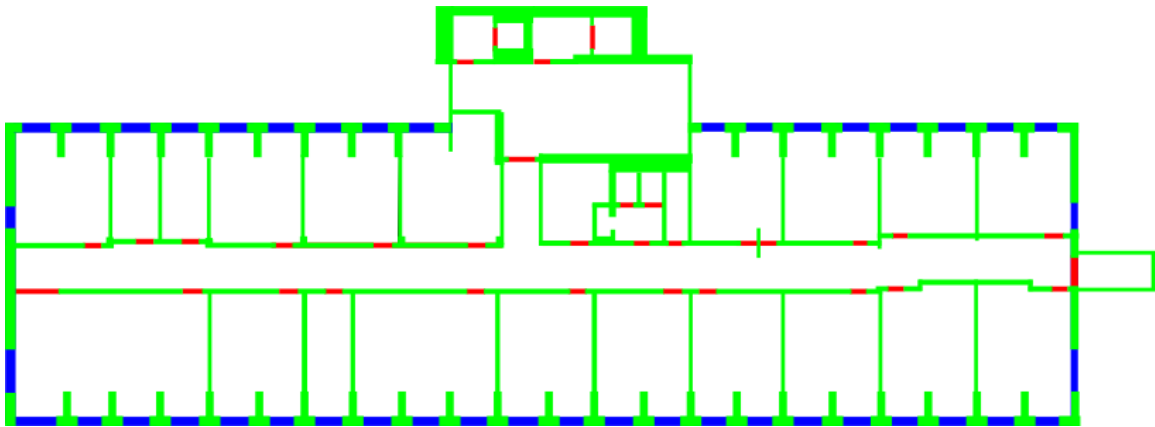


Figura 8: Piantina del primo piano del DIET.

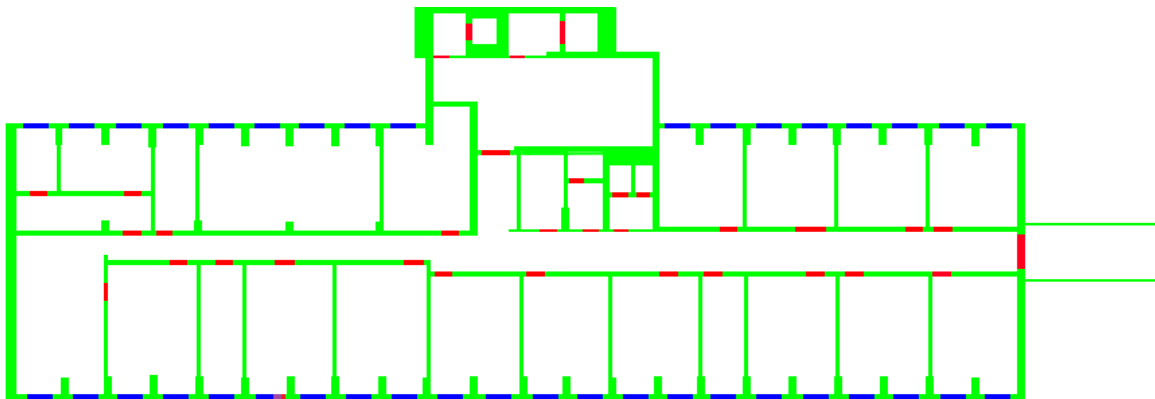
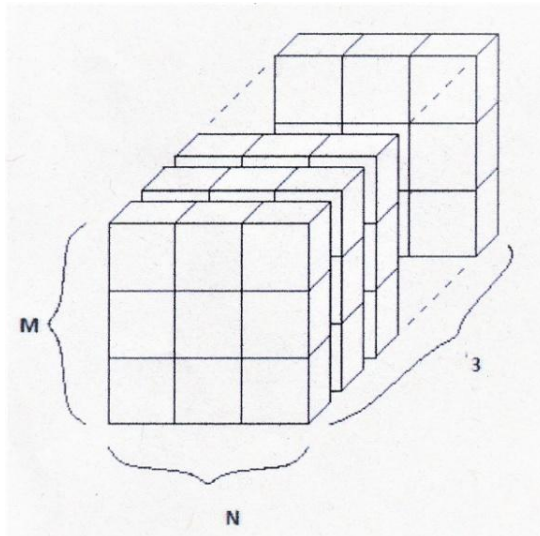


Figura 9: Piantina del secondo piano del DIET.

Questo passo si fa utilizzando la funzione Matlab "imread" che prende in ingresso un'immagine e rende una matrice  $(M \times N \times 3)$  in cui ad ogni pixel è associata una terna di valori formata dai colori primari Red-Green-Blue (RGB). A titolo di esempio il pixel alle coordinate  $(m \times n)$  sarà di colore rosso se la terna di valori associata sarà  $(255, 0, 0)$ , verde se  $(0, 255, 0)$  e blu se  $(0, 0, 255)$ .



**Figura 10: matrice tridimensionale ( $M \times N \times 3$ ) per la rappresentazione dei pixel con la terna di colori RGB.**

Dunque ogni cella della matrice rappresenterà un pixel che a sua volta corrisponderà ad una porzione di spazio, la quale sarà più o meno grande in base alla risoluzione desiderata, ovviamente più sarà scelta piccola e maggiore sarà l'accuratezza.

Nel nostro caso la risoluzione ottenuta è  $0,057 \times 0,057 \text{ m}^2$  per pixel, cioè ogni cella di matrice corrisponde ad un pixel che rappresenta una porzione di area dell'ambiente di sviluppo pari a  $0,057 \times 0,057 \text{ m}^2$ .

A questo punto si imposta una seconda matrice, partendo dalla precedente e di uguale dimensione ( $M \times N \times 1$ ) in cui ad ogni cella si assegna un valore:

- 0 se il pixel è una superficie calpestabile;
- 1 se il pixel rappresenta una porzione di muro;
- 2 se il pixel rappresenta una porzione di porta;
- 3 se il pixel rappresenta una porzione di finestra.

Grazie a questo passaggio associamo ad ogni cella della matrice un valore che caratterizza l'ambiente di sviluppo e quindi, attribuiamo ad ogni colore della terna RGB l'elemento che effettivamente si trova in quel pixel. Come si evince dalle figure 8 e 9 abbiamo scelto di rappresentare in verde i muri, in blu le finestre e in rosso le porte.

### **3.3.2 Maglia delle possibili posizioni degli Access Point**

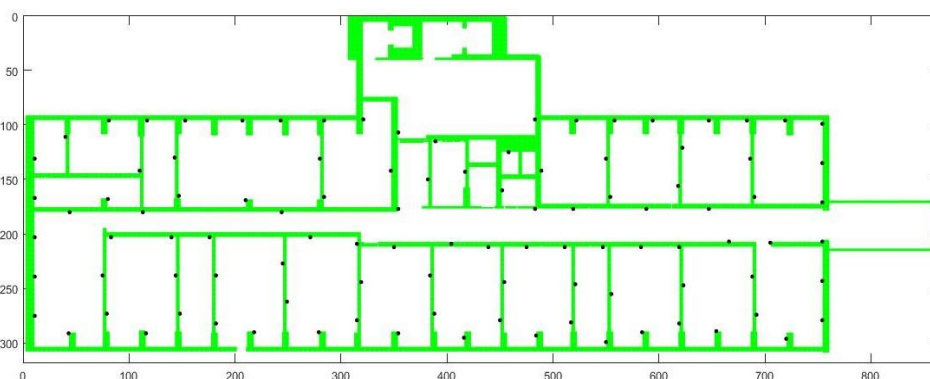
Vogliamo considerare solo alcuni possibili punti in cui posizionare l'Access Point, per esempio sulle pareti delle stanze o dei corridoi, in modo da avere una serie di posizioni utilizzabili e non l'intera piantina come punto di partenza.

Il criterio secondo cui sono scelte le possibili posizioni degli Access Point è la distanza, maggiore o uguale ad un valore soglia, che deve trovarsi fra i vari sensori. Questo ci restituirà una rete più o meno fitta, in base al valore utilizzato, di possibili posizioni degli AP.

L'area di ciascun piano del DIET non è molto estesa perciò abbiamo preso un valore di 2m.



**Figura 11: disposizione degli AP a 2 metri gli uni dagli altri piano 1.**



**Figura 12: disposizione degli AP a 2 metri gli uni dagli altri piano 2.**

### 3.3.3 Creazione delle matrici di attraversamento di muri e porte

A questo punto, considerando un AP alla volta, si studia il raggio emesso dal sensore scelto e, con uno script, si creano due matrici  $M \times N$  in cui ogni cella contiene il conteggio dei muri e delle porte attraversati dal raggio per giungere nella cella considerata. Questo calcolo è effettuato per tutte le celle della matrice ambiente e per ogni posizione possibile dell'AP.

Per chiarezza si è indicato un esempio in cui sono tracciate le rette congiungenti un AP e due celle della matrice che indicano il raggio emesso dall'AP.

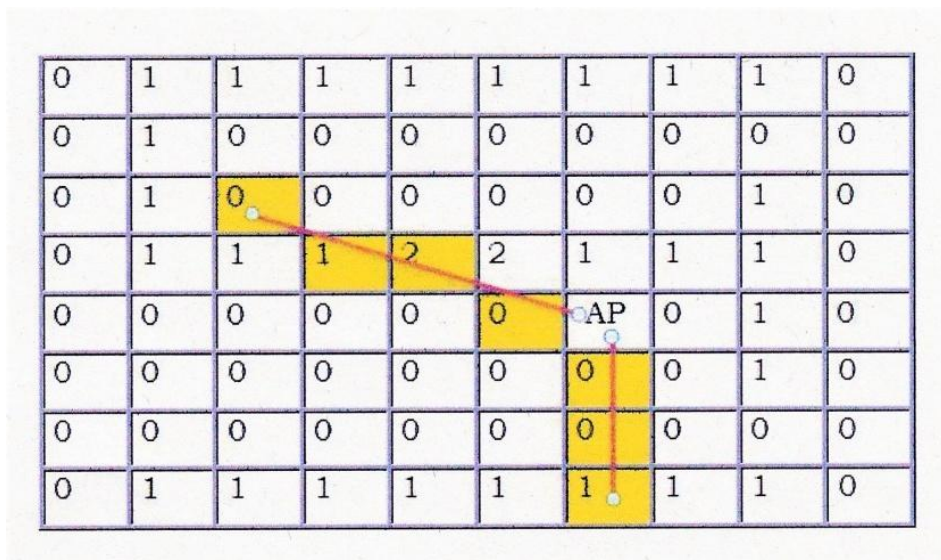


Figura 13: esempio di calcolo dei muri e delle porte attraversate.

Possiamo vedere dalla Figura 10 che il raggio in alto attraversa una cella che corrisponde ad una porta e una che rappresenta un muro, perciò nelle matrici generate, per quella particolare posizione dell'AP e celle a cui facciamo arrivare il raggio, troveremo il valore 1 sia per la matrice dei muri che per quella riguardante le porte. Il raggio in basso, al contrario, non attraversa porte perciò, nelle stesse matrici, troveremo, nella cella in cui arriva il raggio, valore nullo.

### **3.3.4 Matrici delle potenze ricevute in ogni cella da ciascun AP**

A questo punto calcoliamo la potenza ricevuta in ogni punto dell'ambiente considerando ciascun possibile Access Point. Per comporre queste matrici è necessario elencare tutte le possibili posizioni assumibili dagli AP sottoforma di coordinate e poi queste si inseriscono nella matrice ambiente.

Per il conteggio applichiamo il modello Multi-Wall, inserendo i parametri specificati nel paragrafo precedente. Riportiamo brevemente le espressioni analitiche:

$$P_r(d) = EIRP - PL_{MW}(d) \quad (dB)$$

$$PL_{MW} = PL_{OS} + A_{MW} \quad (dB)$$

$$PL_{OS}(d) = l_0 + 10\gamma \log(d) \quad (dB)$$

$$A_{MW} = l_c + \sum_{n=1}^{N_{obj}} \sum_{i=1}^{I_n} N_{n,i} l_{n,i} \quad (dB)$$

Da queste formule si nota l'importanza della generazione delle matrici descritte precedentemente.

### **3.3.5 Criteri per la scelta della posizione del primo AP**

Inizialmente si è pensato di mettere il primo AP nella posizione in cui era in grado di coprire il maggior numero di sotto aree, ma valutando i risultati ottenuti dalle varie prove in [1], è stato notato che con questo metodo non si raggiunge il risultato migliore e quindi l'ottimizzazione della copertura.

Cioè si è visto che utilizzando come prima posizione quella con il maggior numero di celle con potenza ricevuta sopra la soglia, non si raggiungeva l'obiettivo del minor numero di AP usati per la copertura totale. Decidiamo di applicare questo concetto anche al nostro programma.

Perciò sceglieremo la posizione del primo AP in modalità random, grazie ad una funzione di Matlab, e non posizionandolo in base al maggior numero di celle coperte.

### **3.4 Risultati dell' algoritmo per i primi due piani del DIET**

In questo capitolo sono raccolti i risultati ottenuti applicando l' algoritmo, sopra descritto, sui primi due piani dell' edificio in cui si trova il Dipartimento di Ingegneria dell' Informazione, Elettronica e Telecomunicazioni.

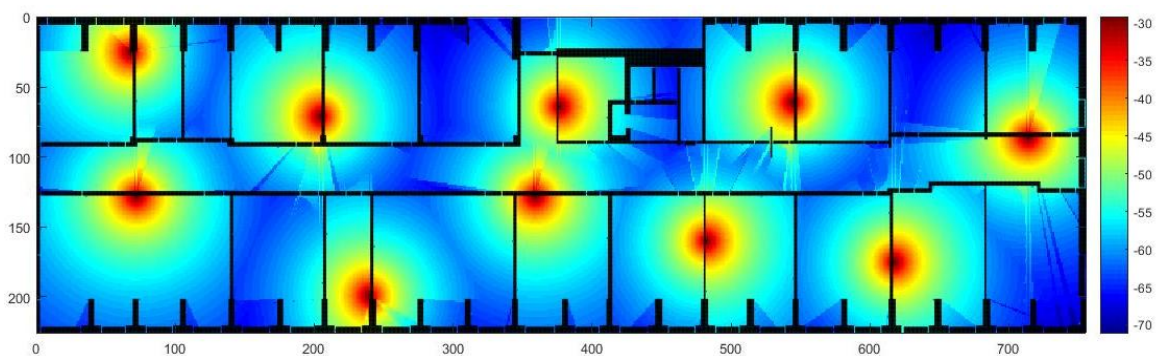
Si è scelto di impostare un valore di potenza ricevuta uguale a quello adottato nella tesi [1], cioè  $-65 \text{ dBm}$ . Questo ci garantirà un Data Rate teorico di  $54 \text{ Mbit/s}$  necessario per avere una buona connessione ad internet. Per un confronto considereremo anche un valore di soglia pari a  $-75 \text{ dBm}$ .

Mostreremo nelle seguenti immagini la disposizione degli AP, puntini neri, nei primi due piani del DIET, tralasciando il contributo fornito da AP posti nei piani limitrofi, e la mappa radio da loro generata. Questa mostrerà la distribuzione delle potenze ricevute in ogni punto della pianta tramite l' associazione della scala dei colori visibili alle potenze ricevute in base all' intensità della potenza ricevuta in ogni punto.

L' area dell' ambiente di sviluppo è di  $12 * 42 \text{ m}^2$ .

Mostriamo i risultati ottenuti i due piani, confrontandoli in base alla soglia utilizzata e ai parametri utilizzati nel modello.

#### **3.4.1 Risultati ottenuti per il primo piano.**



**Figura 14: disposizione ottimale degli AP per il primo piano del DIET con la soglia di potenza ricevuta a  $-65 \text{ dBm}$ .**



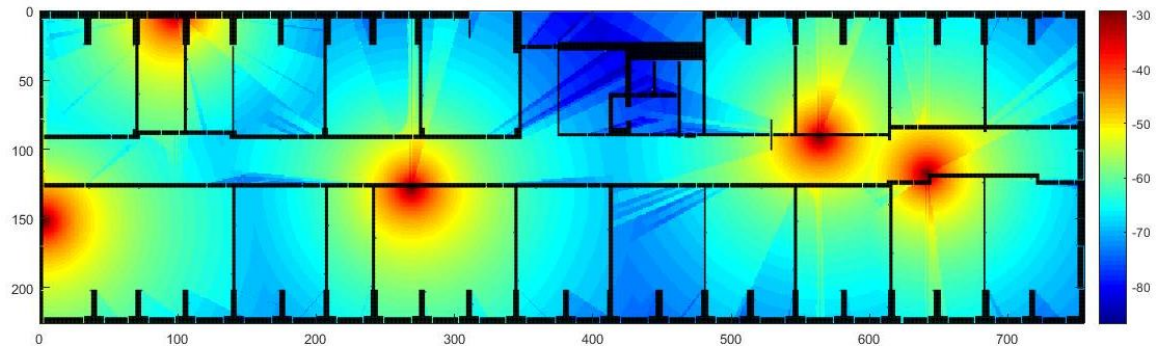


Figura 15: disposizione ottimale degli AP per il primo piano del DIET con la soglia di potenza ricevuta a  $-75 \text{ dBm}$ .

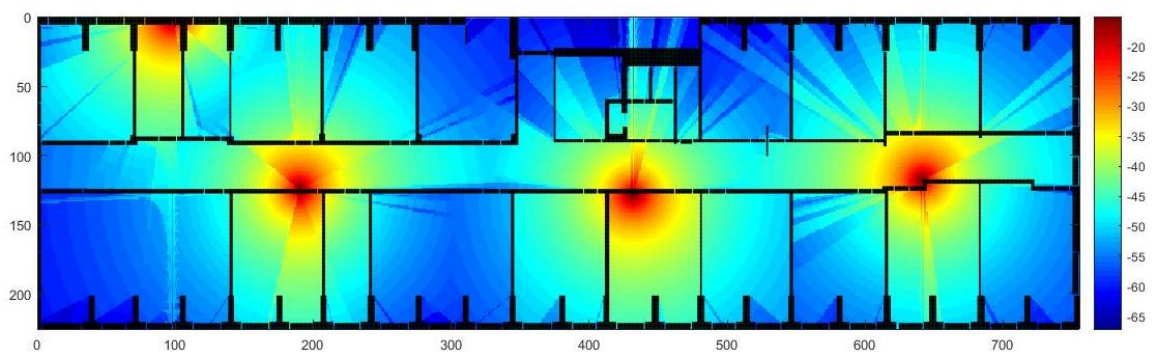


Figura 16: disposizione ottimale degli AP per il primo piano del DIET con i parametri del progetto [1] e  $-65 \text{ dBm}$  di soglia.

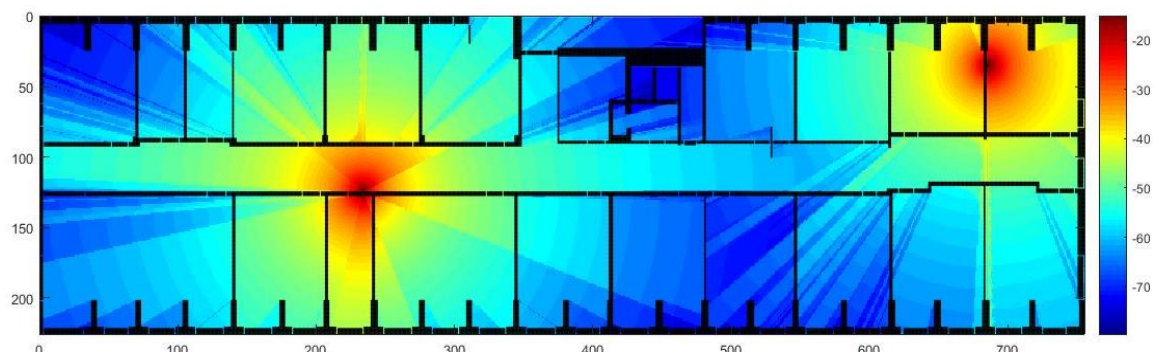


Figura 17: disposizione ottimale degli AP per il primo piano del DIET con i parametri del progetto [1] e  $-75 \text{ dBm}$  di soglia.



### 3.4.2 Risultati ottenuti per il secondo piano

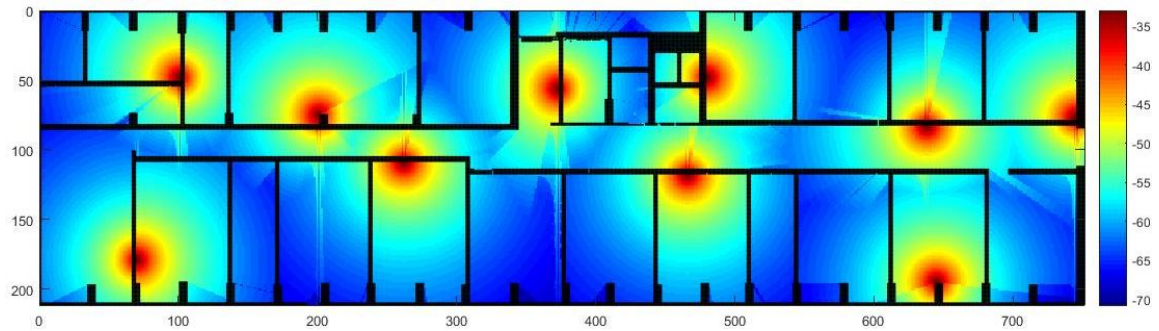


Figura 18: disposizione ottimale degli AP per il secondo piano del DIET con la soglia di potenza ricevuta a  $-65\text{ dBm}$ .

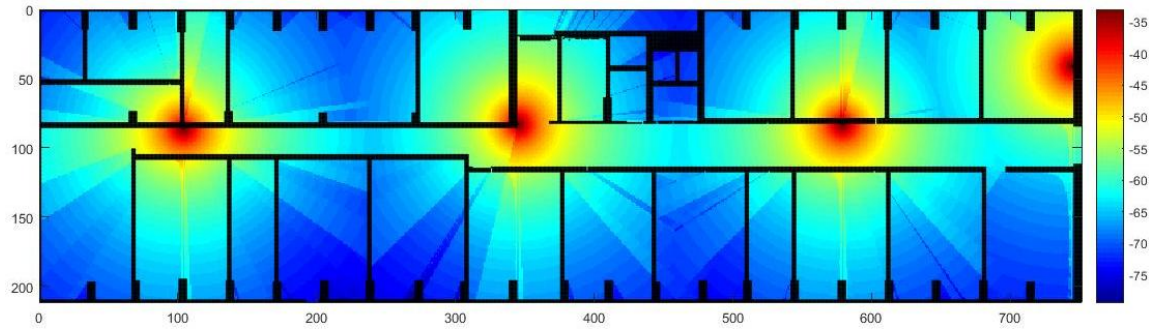


Figura 19: disposizione ottimale degli AP per il secondo piano del DIET con la soglia di potenza ricevuta a  $-75\text{ dBm}$ .

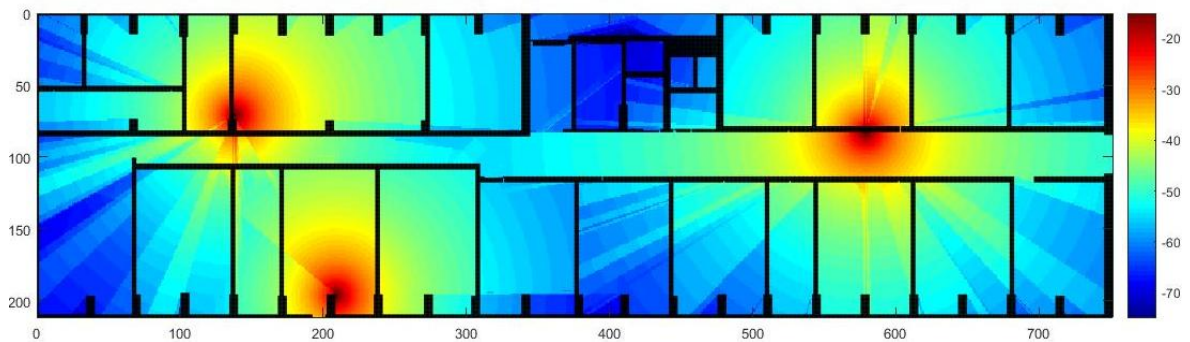
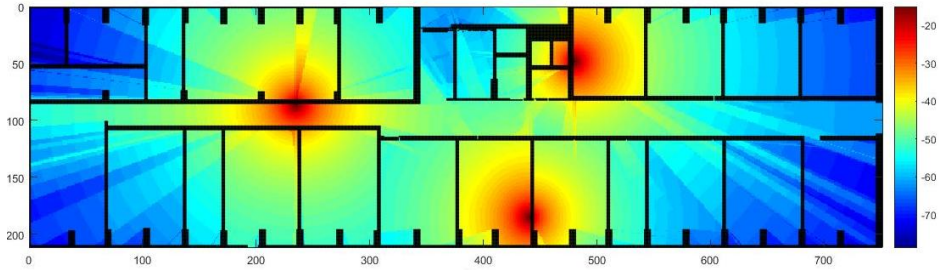


Figura 20: disposizione ottimale degli AP per il secondo piano del DIET con i parametri del progetto [1] e  $-65\text{ dBm}$  di soglia.



**Figura 21: disposizione ottimale degli AP per il secondo piano del DIET con i parametri del progetto [1] e  $-75\text{ dBm}$  di soglia.**

Abbiamo mostrato gli esiti dell'uso, sia di parametri presi dalla letteratura, sia di parametri specifici dell'ambiente di sviluppo, in modo da vedere come effettivamente cambiano i risultati usando diversi valori. Questo ci permette di dire che l'algoritmo genera effettivamente una dislocazione ottimale di Access Point per la copertura completa nell'ambiente di studio.

### **3.5 Riepilogo e conclusioni**

In questo capitolo sono stati specificati i parametri relativi al nostro ambiente di sviluppo, ottenuti con l'applicazione della strategia III, con particolare attenzione all'errore di stima commesso.

In seguito abbiamo descritto passo-passo le vari fasi dell'algoritmo partendo dall'edificio in analisi, il quale è stato diviso in sottoaree, ciascuna di dimensione  $0.057 * 0.057 m^2$ , e rappresentato in Matlab da una matrice  $M \times N$ . Una volta discretizzato l'ambiente di lavoro si è generata una lista di coordinate in cui poter posizionare gli AP poi, si è stata creata la matrice delle potenze ricevute in ogni sottoarea e quelle in cui si contano gli oggetti attraversati dal cammino diretto del segnale. Infine, impostato il valore di minima potenza ricevuta a  $-65 dBm$ , a  $-75 dBm$ , abbiamo ottenuto i risultati dell'ottimizzazione degli AP per la copertura completa dell'ambiente in analisi nei vari casi di piani, soglie e parametri utilizzati.

Riassumendo sono stati necessari 10 AP al primo piano e 10 AP al secondo per ottenere la copertura totale con una potenza minima di  $-65 dBm$  in questo progetto, mentre con i parametri nel lavoro [1], in cui ho lo stesso valore di soglia, sono serviti 4 AP al primo piano e 3 AP al secondo.

Nel caso del valore di soglia impostato a  $-75 dBm$ , abbiamo 5 AP al primo piano e 4 AP nel secondo, per la copertura completa, mentre per i parametri nel lavoro [1], in cui ho lo stesso valore di soglia, sono serviti 2 AP al primo piano e 3 AP al secondo.

L'algoritmo risulta robusto, grazie alle scelte che abbiamo motivato, e assegna effettivamente il numero minore di AP necessari per ottenere la copertura completa con un'accuratezza maggiore nel caso di utilizzo di parametri calcolati ad hoc.

## **4 Codice MATLAB**

Matlab è un linguaggio ad alto livello del calcolo tecnico che fornisce un'interfaccia intuitiva ed efficace per sviluppare algoritmi, per questo motivo è stato scelto per creare uno script utilizzato nel nostro progetto.

### **4.1 Funzioni**

#### **4.1.1 Generazione della matrice ambiente e creazione della griglia di Access Point**

```
fattoreScala = 0.057; %distanza in metri tra due punti
vicini della griglia

distanzaMinima = 2; %distanza minima tra AP espressa in
metri

NomeFile = 'TopologiaPiano1.mat';

%NomeFile = 'TopologiaPiano2.mat';

Inp=load(NomeFile);

    Inp2 = [];

d_scalata = distanzaMinima/fattoreScala; %distanza
orizzontale minima tra AP normalizzata rispetto alla
distanza minima dei punti nella griglia (RP)

%Mridotta è la matrice contenuta nel file caricato che ha
solo i punti che ci interessano

matrice = Inp.Mridotta;

% in matrice è salvato 0, 1, 2 o 3 per ogni punto della
griglia (RP)
% 0 se è spazio libero
% 1 se muro
% 2 se porta
% 3 se finestra
```

```

% AP è una matrice che contiene in ciascuna riga la coppia
di indici che
% identificano una possibile posizione dell'AP nella griglia
(matrice)

if isfield(Inp2, 'AP')
    AP = Inp2.AP;
else
    AP(1,:)=[0 0];
end

for r=1:size(matrice,1)-1
    for c=1:size(matrice,2)-1
        % se il punto considerato è in un muro salva nelle
variabili d ed s che tipo di materiale c'è nei punti a
destra e a sinistra e nelle variabili a e b che tipo di
materiale c'è nei punti sopra e sotto per il perimetro della
griglia, mancando uno dei punti da considerare (o sopra, o
sotto, o a destra, o a sinistra), lo imposta come se fosse
un punto di muro

        if matrice(r,c)==1
            if r==1 && c~=1
                d=matrice(r,c+1);
                s=matrice(r,c-1);
                a=1;
                b=matrice(r+1,c);
            else
                if r~=1 && c==1
                    d=matrice(r,c+1);
                    s=1;
                    a=matrice(r-1,c);
                    b=matrice(r+1,c);
                else
                    if r==1 && c==1
                        d=1;
                        s=1;
                        a=1;
                        b=1;
                    else
                        d=matrice(r,c+1);
                        s=matrice(r,c-1);
                        a=matrice(r-1,c);
                        b=matrice(r+1,c);
                    end
                end
            end
        end
    end

    if d==0

```

```

        nuovoAP=[c+1 r];

        %risposta vale 1 se il nuovo AP è già stato
considerato
        risposta=ismember(nuovoAP,AP,'rows');

        % distanza è un array (riga) di distanze tra
        il possibile nuovo punto e tutti i possibili punti
        già determinati
        distanza=dist(nuovoAP,AP') ;

        if (risposta==0 &&
iseempty(find(distanza<=d_scalata,1)))
            AP(size(AP,1)+1,:)=[c+1 r];

        end
    end

    if s==0
        nuovoAP=[c-1 r];

        %risposta vale 1 se il nuovo AP è già stato
considerato
        risposta=ismember(nuovoAP,AP,'rows');

        % distanza è un array (riga) di distanze tra
        il possibile nuovo punto e tutti i possibili punti
        già determinati
        distanza=dist(nuovoAP,AP') ;

        if (risposta==0 &&
iseempty(find(distanza<=d_scalata,1)))
            AP(size(AP,1)+1,:)=[c-1 r];

        end
    end

    if a==0
        nuovoAP=[c r-1];

        %risposta vale 1 se il nuovo AP è già stato
considerato
        risposta=ismember(nuovoAP,AP,'rows');

```

```

        % distanza è un array (riga) di distanze tra
        il possibile nuovo punto e tutti i possibili punti
        già determinati
        distanza=dist(nuovoAP,AP') ;

        if (risposta==0 &&
isempty(find(distanza<=d_scalata,1)))
            AP(size(AP,1+1),:)= [c r-1];

        end
    end

    if b==0
        nuovoAP=[c r+1];

        %risposta vale 1 se il nuovo AP è già stato
considerato
        risposta=ismember(nuovoAP,AP,'rows');

        % distanza è un array (riga) di distanze tra
        il possibile nuovo punto e tutti i possibili punti già
        determinati
        distanza=dist(nuovoAP,AP') ;

        if (risposta==0 &&
isempty(find(distanza<=d_scalata,1)))
            AP(size(AP,1)+1,:)= [c r+1];

        end
    end
end
end
end

% ordino posizione AP
Y=zeros(size(AP,1),2);
[Y(:,1),I]=sort(AP(:,1),1); %ordino gli AP per valore della
coordinata x

for t=1:size(AP,1)
    Y(t,2)=AP(I(t,1),2);
end

clear AP
AP=Y;
s=AP;
clear AP

```

```

AP=s(2:size(s,1),:);

Mridotta = matrice;
M = Inp.M;
Xoffset = Inp.Xoffset;
Yoffset = Inp.Yoffset;

% visualizzazione
matriceAP=zeros(size(Inp.M,1),size(Inp.M,2));
for h=1:size(AP,1)
    matriceAP(Xoffset+AP(h,2),Yoffset+AP(h,1))=1;
end

spy(Inp.M, 'g')
hold on
% spy(matrice, 'g')
% hold on
spy(matriceAP,'black',10)

```

#### **4.1.2 Valuta muri e porte attraversate**

```

% load Mridotta
% load matriceAP
coordinate=AP;
dim_righe=size(Mridotta,1);
dim_colonne=size(Mridotta,2);
x_a=coordinate(:,1)';
y_a=coordinate(:,2)';
[num_muri,num_porte]=contaMuri_2a(Mridotta,x_a,y_a);
h=5; % 3metri=5celle
[distanza]=calcolaDistanza_2b(coordinate,dim_righe,dim_colonne,h);

```

##### **4.1.2.1 Conta numero muri e numero porte attraversati dal segnale di ogni Access Point**

```

% Mridotta la matrice che descrive la planimetria
%x_a=vettore riga delle coordinate x degli AP
%y_a=vettore riga delle coordinate y degli AP
%OUTPUT
%restituisce una matrice dim_righe x dim_colonne x nAP che
contiene cella per cella il numero di muri attraversati da
AP a-esimo

```



%restituisce una matrice dim\_righe x dim\_colonne x nAP che contiene cella per cella il numero di porte attraversata da AP a-esimo

```
function [num_muri,numporte]=contaMuri_2a(Mridotta,x_a,y_a)
dim_righe=size(Mridotta,1);
dim_colonne=size(Mridotta,2);
nAP=length(x_a);
num_muri=zeros(dim_righe,dim_colonne,nAP);
numporte=zeros(dim_righe,dim_colonne,nAP);
for a=1:nAP % per ogni Access Point
    a
    for r=1:dim_righe

        for c=1:dim_colonne

            % fisso la cella in analisi M(r,c) e calcolo la
            retta che unisce cella ed AP questa esprime i punti
            attraversati

            x_m=c;

            y_m=r;

            if (x_a(1,a)==x_m) && (y_a(1,a)==y_m)

                x_attraversate=x_m;
                y_attraversate=y_m;
            end

            if (x_a(1,a)==x_m) && (y_a(1,a)~=y_m)
%retta verticale

                val_min=min([y_a(1,a) y_m]);
                val_max=max([y_a(1,a) y_m]);
                y_attraversate=[val_min:1:val_max;];
%vettore riga che ha un passo in piu per analizzare il
numero di muri attraversati

            x_attraversate=ones(1,length(y_attraversate))*x_m;

            else
                if (x_a(1,a)~=x_m) && (y_a(1,a)==y_m)
%retta orizzonate

                    x_attraversate=[min([x_a(1,a)
x_m]):1:max([x_a(1,a) x_m])];
%vettore riga
```

```

y_attraversate=ones(1,length(x_attraversate)).*y_m;

end
end

if (x_a(1,a)~=x_m) && (y_a(1,a)~=y_m)

    m=(y_a(1,a)-y_m)/(x_a(1,a)-x_m);
    [val_min,I]=min([x_a(1,a) x_m]);
    val_max=max([x_a(1,a) x_m]);
    if I==1
        q=y_a(1,a)-m*x_a(1,a);
    else
        q=y_m-m*x_m;
    end
    x_attraversate=[val_min+1:0.1:val_max];
    y_attraversate=m.*x_attraversate+q;

end

% si sono tracciate le rette ora bisogna
individuare quanti muri attraversano

x=round(x_attraversate);
y=round(y_attraversate);
d_x=length(x);
d_y=length(y);
l=1;
while l<=d_x

    if x(1,l)<=0 || y(1,l)<=0 ||
x(1,l)>dim_colonne || y(1,l)>dim_righe

        if l==d_x
            x=[x(1,1:l-1)];
            y=[y(1,1:l-1)];

        end
        if l==1
            x=[x(1,2:length(x))];
            y=[y(1,2:length(y))];

        else
            x=[x(1,1:l-1) x(1,l+1:length(x))];
            y=[y(1,1:l-1) y(1,l+1:length(y))];
        end
        l=l-1;
        d_x=length(x);
    end
    l=l+1;
end

```

```

        d_x=length(x);
    end

    for k=1:d_x-1 % escludo dal conteggio le celle
in analisi

        if Mridotta(y(1,k),x(1,k))==1 &&
Mridotta(y(1,k+1),x(1,k+1))==0

            num_muri(r,c,a)=num_muri(r,c,a)+1;
        end
        if Mridotta(y(1,k),x(1,k))==2 &&
Mridotta(y(1,k+1),x(1,k+1))==0
            num porte(r,c,a)=num porte(r,c,a)+1;
        end
    end
end
end
end
end
end
end

```

#### 4.1.2.2 Calcolo distanza fra gli Access Point e ogni cella della matrice

%restituisce la distanza in metri tra gli AP ed i punti centrali in cui ho diviso la piantina

```

function
[distanza]=calcolaDistanza_2b(coordinate,dim_righe,dim_colonne,h)

AP=size(coordinate,1);

distanza=zeros(dim_righe,dim_colonne,AP);
fattoreScala=0.057;
for a=1:AP

    for r=1:dim_righe

        for c=1:dim_colonne

            distanza(r,c,a)=((((c-coordinate(a,1))^2)+(r-coordinate(a,2))^2+h^2)^0.5)*fattoreScala; %distanza in
metri ricevendo in ingresso le coordinate della matrice
        end
    end
end
end
end
end

```

### 4.1.3 Calcolo della potenza ricevuta in ogni cella della matrice ambiente

```
%restituisce una matrice in cui ogni cella contiene ala
potenza ricevuta secondo il modello multi wall

function
[PotenzaLineare,Pricevuta_dB]=calcoloPotenza_3a(num_muri,distanza,numporte)

lc=62.73; % contiene anche il valore di 10[dB] 1piano
% lc=65; %2piano
% lc=47.4; %LETTERATURA

l2=2.84; %in [dB] 1 piano
% l2=2.7; %2piano
% l2=3.9; %LETTERATURA

ld=4.37; % in [dB] 1 piano
% ld=3.21; %2piano
% ld=1.4; %LETTERATURA

EIRP=20; % [dB]

Los=24.89*log10(distanza); %1piano
% Los=22.07*log10(distanza); %2piano
% Los=23.2*log10(distanza); %LETTERATURA

multi_wall=num_muri.*l2+numporte*ld; % considero i muri
tutti uguali e tutte porte semplici

L_d=Los+lc+multi_wall;

Pricevuta_dB=EIRP-L_d; %dB
%L_dLineare=10.^(L_d/10);
%Pricevuta=Pt-L_dLineare;
%PotenzaLineare=Pricevuta;
%Pricevuta_dB=10*log(PotenzaLineare);

PotenzaLineare=10.^(Pricevuta_dB/10);

end
```

## 4.2 Main script

```
dim_righe=size(Mridotta,1);
dim_colonne=size(Mridotta,2);
fattoreScala=0.057;
APdef=AP;

%% il primo AP posizionato casualmente

puntiAP=APdef;
r= round(1 + (size(puntiAP,1)-1).*rand(1,1));

Id=[1:size(APdef,1)]';
puntiAP=[puntiAP Id];

AP=puntiAP(r,1:2);
APusati=puntiAP(r,:);

if r==1
    puntiAPrimasti=puntiAP(2:size((puntiAP),1),:);
else if r==size(AP,1)
    puntiAPrimasti=puntiAP(1:size(AP)-1,:);
    else
        puntiAPrimasti=[puntiAP(1:r-1,:); puntiAP(r+1:size(puntiAP,1),:)];
    end
end
Idutilizzati=APusati(:,3);

% calcolo la potenza ricevuta nell'intera struttura con solo questo AP
funzione che restituisce in metri la distanza tra l'Ap inserito ed tutti
i punti centrali delle celle Base in cui e stata divisa la struttura
dis(:, :, 1)=distanza(:, :, r);

% si calcola la potenza considerando il num di muri attraversati
muri(:, :, 1)=num_muri(:, :, r);
porte(:, :, 1)=num_porte(:, :, r);

[PotenzaLineare, Pricevuta_dB]=calcoloPotenza_3a(muri,dis,porte);
Pot(1,1)={Pricevuta_dB};

% a questo punto per capire se e necessario inserire un altro AP
dobbiamo valutare quante celleBase sono al di sotto del valore di
Sensibilita

% Sensibilita=-65;          % [dBm]
Sensibilita=-75;          % [dBm]

Y=size((find(Pricevuta_dB>Sensibilita)),1);
h=1;
Massimi=Pricevuta_dB;

while (Y<0.95*(dim_righe*dim_colonne))
```

```

    % bisogna valutare quale possibile posizione degli AP riduce il
    numero di celleScoperte al massimo cosi da minimizzare il numero di AP
    presenti

    for d=1:size(puntiAPrimasti,1)
        AProve=[AP; puntiAPrimasti(d,1:2)];

    % [distanza]=calcolaDistanza(AProve,dim_righe,dim_colonne);

        dis=distanza(:, :,puntiAPrimasti(d,3));
        muri=num_muri(:, :,puntiAPrimasti(d,3));
        porte=num_porte(:, :,puntiAPrimasti(d,3));
        [PotenzaLineare,Pricevuta_dB]=calcoloPotenza_3a(muri,dis,porte);

    %[PotenzaLineare,Pricevuta_dB]=calcolo_pot(muri,dis);
        % immagazzino di volta in volta la potenza ottenuta per
    verificare al variare dei diversi AP
        Pot(1,h+1)={Pricevuta_dB};

    % la cella e coperta se uno degli AP in quel punto supera la soglia
    max che stabilisco

        MatriceMassimi(1,d)={max(Massimi,Pot{1,h+1})};

    celleCoperte(1,d)=size((find(MatriceMassimi{1,d}>=Sensibilita)),1);
        celleCoperteId(1,d)=puntiAPrimasti(d,3);

    end

        [Y,I]=max(celleCoperte(1,:));
        Massimi=MatriceMassimi{1,I};
        [r,c]=find(puntiAPrimasti(:,3)==celleCoperteId(1,I));
        APusati=[APusati; puntiAPrimasti(r,:)];

    AP=APusati(:,1:2) ;           %sono le coordinate degli AP inseriti
    Idutilizzati=APusati(:,3) '   %sono gli ID degli AP utilizzati
    n=1;

    % elimino L'AP gia inserito dalla serie

        for l=1:size(puntiAPrimasti,1)
            if puntiAPrimasti(l,3)~=Idutilizzati(1,size(Idutilizzati,2))
                rimasti(n,:)=puntiAPrimasti(l,:);
                n=n+1;
            end
        end

    clear k l

    %clear puntiAPrimasti celleCoperte
    puntiAPrimasti=rimasti;
    %clear rimasti
    h=h+1;
    Y

end

```

```

combinazioniPossibili={APusati};
celle={Y};
%clear APusati Y Idutilizzati h Y

clear h

matriceAP=zeros(size(matrice,1),size(matrice,2));
for h=1:size(APdef,1)
    matriceAP(APdef(h,2),APdef(h,1))=1;
end
clear h

matriceAPusati=zeros(size(matrice,1),size(matrice,2));
for h=1:size(AP,1)
    matriceAPusati(AP(h,2),AP(h,1))=1;

end

% spy(matrice)
% hold on
% spy(matriceAP,'r',10)
% hold on
% spy(matriceAPusati,'black',20)

imagesc(Massimi)
colormap jet

hold on
spy(matrice,'black')

hold on
% spy(matriceAP)

hold on
spy(matriceAPusati,'k')

```

## **5 Conclusioni e sviluppi futuri**

In questo lavoro abbiamo rivisitato il progetto [1] apportando migliorie che ci permettono di ottenere risultati più precisi e accurati.

La ristrutturazione più importante è, senza dubbio, quella dell'introduzione di parametri empirici per il modello Multi-Wall. Questi valori infatti sono stati ricavati nell'ambiente indoor in cui vogliamo studiare la mappa radio, tenendo conto di tutti i fattori di interesse, ossia le caratteristiche topologiche dell'edificio.

L'utilizzo di parametri ad hoc porta indubbiamente ad una visione più realistica della situazione, perciò siamo giunti, utilizzando l'algoritmo in [1], ad una nuova pianificazione degli AP nei primi due piani dell'edificio del Dipartimento di Ingegneria dell'Informazione, Informatica e Statistica.

Considerando le necessità tecnologiche del momento, le quali, con molta probabilità, continueranno ad aumentare e ad evolversi, possiamo immaginare l'importanza di questo studio.

Detto ciò, non ci resta che considerare le possibili applicazioni future di questo lavoro.

Un possibile sviluppo futuro è la considerazione della propagazione dei segnali generati da Access Point dislocati in piani adiacenti nell'edificio ambiente di sviluppo, cioè avere un approccio tridimensionale, e quindi trasversale.

Un altro aspetto da valutare è l'utilizzo della maglia delle possibili posizioni degli AP.

Nel progetto [1], si è visto che, applicando l'algoritmo e ripetendolo finché ogni posizione non è stata scelta come primo AP otteniamo risultati diversi in base alla distribuzione iniziale scelta. Infatti diminuendo la dimensione della maglia usata per la distribuzione della posizione degli AP troviamo che il numero di sensori necessari per la copertura completa diminuisce.

Per determinare in modo opportuno il valore di maglia si potranno mettere a confronto gli esiti dell'algoritmo con l'utilizzo di varie maglie, in modo da vedere se effettivamente una maglia più stretta e capillare darà un risultato migliore di ottimizzazione di copertura.



## Bibliografia

- [1] Valentina Piccolo. "Dislocazione ottima dei sensori per localizzazione indoor".
- [2] Giuseppe Caso, Luca De Nardis, " Virtual and Oriented WiFi Fingerprinting Indoor Positioning based on Multi-Wall Multi-Floor Propagation Models". Springer Science+Business Media New York 2016.
- [3] Andrea Borelli, Cristiano Monti, Marco Vari, Franco Mazzenga, "Channel Models for IEEE 802.11b Indoor System Design", IEEE Communication Society 2004.
- [4] Giuseppe Caso, Luca De Nardis, "On the Applicability of Multi-Wall Multi-Floor Propagation Model to WiFi Fingerprinting Indoor Positioning", International Conference on Future Access Enablers for Ubiquitous and Intelligent Infrastructures, LNICST Vol.159, Springer, Berlin(2015).