

# UNIVERSITÀ DI PISA

FACOLTÀ DI INGEGNERIA  
CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA DELLE  
TELECOMUNICAZIONI  
CURRICULUM:  
"Elaborazione del segnale e del Telerilevamento"



## **Sviluppo di un sistema di localizzazione ZigBee e valutazione sperimentale delle prestazioni in ambienti chiusi**

*Relatore:*

Prof. Ing. GIUSEPPE IANNACCONE

Prof. MARIA SABRINA GRECO

Dott. Ing. ELISA SPANÒ

*Tesi di Laurea di:*  
LEANDRO DAL MASO

February 17, 2012

*Per quanto le leggi della matematica si riferiscano alla realtà,  
esse nn sono certe, e per quanto siano certe,  
esse non si riferiscono alla realtà.*

*Albert Einstein*

# Indice

<b>I LOCALIZZAZIONE</b>	<b>1</b>
<b>1 Wireless Sensor Network : Aspetti Generali</b>	<b>1</b>
1.1 Applicazioni . . . . .	2
<b>2 Sistemi di localizzazione</b>	<b>6</b>
2.1 Tecniche Range-Based: Stima della distanza tra nodi . . . . .	7
2.1.1 Tempo di Arrivo . . . . .	8
2.1.2 Tempo di Volo . . . . .	9
2.1.3 Angolo di Arrivo . . . . .	12
2.1.4 Receive Signal Strenght Indicator . . . . .	13
2.2 Tecniche Range-Based : Algoritmi di stima utilizzando come parametro l’RSSI	17
2.2.1 Min-Max . . . . .	17
2.2.2 Trilaterazione . . . . .	19
2.2.3 Multilaterazione . . . . .	23
2.3 Tecniche Range-Free: Algoritmi di stima . . . . .	25
2.3.1 $K$ -Nearest Neighbor . . . . .	26
2.3.2 Metodi Bayesiani . . . . .	27
2.3.3 Support Vector Machine . . . . .	30
2.3.4 Multidimensional Vector Regression . . . . .	32
2.4 Parametri per valutare le prestazioni di un sistema di localizzazione . . . . .	33
<b>3 Scelta dell’Algoritmo di localizzazione</b>	<b>36</b>
<b>II ZIGBEE</b>	<b>41</b>
<b>4 Il Protocollo</b>	<b>41</b>
4.1 Introduzione . . . . .	41
4.2 Cartteristiche ZigBee . . . . .	42
4.2.1 Dati Tecnici . . . . .	42
4.3 Stack del protocollo . . . . .	44
4.4 Livello Applicazione . . . . .	46
4.4.1 Application Profile . . . . .	46
4.4.2 Cluster . . . . .	47
4.4.3 Attributi . . . . .	48
4.4.4 Device ID . . . . .	48
4.4.5 EndPoint . . . . .	48
4.5 La Rete ZigBee . . . . .	49

<b>5</b>	<b>Sistema di Localizzazione ZigBee</b>	<b>50</b>
5.1	Introduzione . . . . .	50
5.2	Specifiche : Profilo , Cluster , Dispositivo . . . . .	51
5.3	Specifiche : Payload Messaggi . . . . .	51
5.3.1	Server . . . . .	52
5.3.2	Client . . . . .	59
<b>6</b>	<b>BeeStack : Trasmissione e Ricezione dei pacchetti ZigBee</b>	<b>64</b>
6.1	Software Freescale . . . . .	64
6.2	Trasmissione e ricezione dei pacchetti ZigBee . . . . .	65
<b>7</b>	<b>Schede Utilizzate nel Progetto</b>	<b>69</b>
7.1	Tipi di Schede . . . . .	69
7.2	MC13224: Misura dell'RSSI . . . . .	70
<b>III</b>	<b>SISTEMA DI LOCALIZZAZIONE IMPLEMENTATO</b>	<b>72</b>
<b>8</b>	<b>Implementazione del Sistema di Comunicazione e del Cluster RSSI Location</b>	<b>72</b>
8.1	Implementazione del Cluster RSSI Location del T.A. . . . .	73
8.2	Sistema di Comunicazione . . . . .	75
8.2.1	Gateway . . . . .	76
8.2.2	Ancora . . . . .	77
8.2.3	Location Node . . . . .	78
8.3	Calibration tag . . . . .	80
<b>9</b>	<b>Implementazione del motore di localizzazione</b>	<b>83</b>
9.1	Calibrazione . . . . .	83
9.2	Localizzazione . . . . .	87
9.2.1	Inizializzazione del motore di localizzazione . . . . .	88
9.2.2	Motore per la Localizzazione . . . . .	90
9.3	Esempio di Applicazione . . . . .	93
<b>IV</b>	<b>SISTEMA DI LOCALIZZAZIONE EKAHAU</b>	<b>96</b>
<b>10</b>	<b>Overview del Sistema RTLS</b>	<b>96</b>
10.1	Ekahau Positioning Engine . . . . .	97
10.2	Ekahau Site Survey . . . . .	99
<b>V</b>	<b>VERIFICHE SPERIMENTALI : TEST E CONFRONTI PRE-STAZIONALI</b>	<b>101</b>

<b>11 Ambiente dei test</b>	<b>101</b>
<b>12 Test Ekahau</b>	<b>103</b>
<b>13 Test del Sistema di localizzazione ZigBee</b>	<b>106</b>
13.1 Test 1 . . . . .	106
13.2 Test 2 . . . . .	109
13.3 Test 3 . . . . .	110
<b>VI CONCLUSIONI</b>	<b>115</b>
<b>14 Confronti Prestazionali e Sviluppi Futuri</b>	<b>115</b>
14.1 Confronti Prestazionali . . . . .	115
14.2 Sviluppi Futuri . . . . .	116

## Parte I

# LOCALIZZAZIONE

## 1 Wireless Sensor Network : Aspetti Generali

Recenti sviluppi nell'ambito delle telecomunicazioni e dell'elettronica hanno enormemente esteso l'utilizzo di sistemi elettronici a banda stretta nelle reti *Wireless*. Le reti *Wireless* sono costituite da dispositivi in grado di trasmettere informazione utile senza fili, ovvero in assenza di un collegamento fisico diretto tra gli elementi che compongono la rete.

Una particolare tipologia di rete *Wireless* è quella denominata *Wireless Sensor Network* (WSN) che, caratterizzata da una architettura distribuita, è realizzata da un insieme di sistemi elettronici autonomi detti "nodi sensori" o più semplicemente "nodi" della rete. Tali sistemi sono costituiti principalmente da un'unità di rilevamento dati (temperatura, potenza, etc...), denominata sensore, capace di prelevare dati dall'ambiente circostante, da un microcontrollore e da una parte radio. L'informazione trasmessa o elaborata da tali nodi può essere quindi di varia natura e dipende dal tipo di unità di rilevamento presente sul nodo della rete.

La crescente importanza di tale tipologia di reti è dovuta al continuo incremento delle informazioni richieste in tempo reale e in qualsiasi posizione nell'ambiente in cui viviamo e lavoriamo. I principali vantaggi che hanno motivato l'uso delle reti senza fili in alcune applicazioni rispetto alle reti cablate sono il costo e il tempo di installazione. Creare una rete di nodi senza fili richiede infatti un tempo ed un costo di gran lunga inferiore rispetto ad una rete cablata. Inoltre la possibilità di alimentare i nodi a batteria permette di creare reti in zone dove non è possibile stendere cavi o ottenere l'energia elettrica. Le WSN hanno però alcuni svantaggi rispetto le reti cablate, sono infatti meno affidabili, permettono velocità in trasmissione inferiori e soprattutto sono più vulnerabili ad attacchi esterni.

La possibilità di trasmettere informazione riguardo grandezze fisiche, come temperatura o umidità, relative all'ambiente in cui è installata la rete, rende tale tecnologia particolarmente adatta ad implementare sistemi di controllo e monitoraggio in applicazioni in cui non sono richieste elevate prestazioni in termini di affidabilità e velocità in trasmissione.

I sistemi di controllo che si possono implementare con tali tecnologie sono molteplici e dipendono dal tipo di sensore installato sui nodi della rete; si pensi ad esempio ad una rete WSN installata in un'abitazione costituita da nodi con sensori di temperatura. La possibilità di trasmettere l'informazione sulla temperatura delle stanze dell'abitazione ad

un controllore, permette di implementare un sistema di controllo del condizionamento e del riscaldamento dell'abitazione.

Nel seguito del presente elaborato , dove non specificato, sarà fatto riferimento ad un particolare tipo di WSN: le WPAN (*Wireless Personal Area Network*). Il motivo di questa scelta sarà chiarito in seguito.

Nella sezione seguente sono riportate alcune applicazioni più comuni delle reti WSN.

## 1.1 Applicazioni

Una rete wireless utilizzata per implementare sistemi di controllo e monitoraggio deve avere le seguenti caratteristiche :

- non è necessario che i nodi trasferiscano grandi quantità di dati. Infatti nelle applicazioni più comuni non è necessario trasferire file audio o video che richiedono un elevato data rate in trasmissione;
- non tutti i nodi della rete possono essere alimentati dalla rete elettrica, quindi la durata delle batterie deve essere elevata;

e i seguenti requisiti:

- il numero di nodi presenti all'interno di una rete deve poter essere elevato, si pensi ad esempio a sistemi di controllo dedicati ad industrie di vaste dimensioni;
- il costo di ogni singolo nodo deve essere basso;
- è necessario disporre di un livello adeguato di sicurezza nel trasferimento dati.

Sia in letteratura che nella pratica i tentativi di implementare sistemi di controllo e monitoraggio impiegando le reti WSN sono molteplici. Le applicazioni più comuni per cui tali sistemi vengono realizzati sono le seguenti :

- *Applicazioni militari*: In questo contesto risulta necessario creare dei sistemi per il controllo e la sicurezza delle persone e oggetti che operano in tale ambito. In alcuni casi potrebbe rivelarsi utile la possibilità di installare un sistema di questo tipo in modo rapido. Le possibili applicazioni di tale tecnologia vanno dal monitoraggio di forze alleate, degli equipaggiamenti e delle munizioni alla sorveglianza del campo di

battaglia [3]. Ad esempio la *Nevada Nanotech Systems*, un'azienda fondata nel 2004 in Nevada <http://www.nevadanano.com/company/> che si occupa dello sviluppo di sensori per applicazioni commerciali e governative in collaborazione con l'Università del Nevada, sta sviluppando un sistema di sensori in grado di rilevare tracce di esplosivi, di elementi chimici tossici e biologici analizzando le proprietà elettriche e termiche esalate dalle sostanze in questione; la *SI2 Technologies* <http://www.si2technologies.com/about-si2-technologies/>, un'azienda fondata nel 2003 nei pressi di Boston che si occupa dello sviluppo di sistemi elettronici per applicazioni militari, ha sviluppato sensori elettro-acustici per rivelare la presenza di sottomarini.

- *Applicazioni ambientali*: Nelle applicazioni ambientali le zone da monitorare sono spesso vaste e lontane dall'alimentazione elettrica (es. foreste). In questi casi si rivela perciò utile la possibilità di creare un sistema di monitoraggio costituito da nodi senza fili e alimentati a batterie. Recentemente è stato implementato un sistema per la prevenzione degli incendi nelle foreste attraverso il controllo della temperatura dell'area monitorata [4]. La prevenzione delle alluvioni, il monitoraggio delle attività vulcaniche, etc... sono attività per le quali sono stati sviluppati o si stanno sviluppando sistemi costituiti da sensori *Wireless*.
- *Applicazioni civili*: In questo contesto sono particolarmente utili i sistemi dedicati al controllo automatico dell'ambiente in cui vivono le persone. La tecnologia WSN trova infatti vasto impiego nella domotica, cioè l'insieme delle tecnologie atte a migliorare la qualità della vita nella casa. Una rete di nodi senza fili può essere utilizzata per il controllo del sistema di riscaldamento dell'abitazione, di condizionamento, di illuminazione e per molti altri aspetti. Un sistema di monitoraggio particolarmente utile è quello che permette di monitorare i consumi energetici degli elettrodomestici casalinghi o delle macchine industriali.
- *Applicazioni Industriali*: I tradizionali sistemi automatici industriali per il controllo e il monitoraggio sono cablati e richiedono un alto costo di installazione e manutenzione. Per cercare di ridurre i costi nel settore industriale vengono studiati sistemi di controllo *Wireless*, prestando particolare attenzione alla loro affidabilità [5] caratteristica fondamentale di questo tipo di ambienti in cui un errore può compromettere la sicurezza di molte persone.

Un esempio di tecnologia sviluppata per implementare le reti WSN è quella *ZigBee* [17]. I nodi di una *ZigBee* per comunicare si basano sul protocollo di comunicazione al livello fisico e MAC IEEE 802.15.4 [54]. Lo standard citato è per reti *WPAN*.

In Fig.1 sono illustrate le tipiche applicazioni per cui tale tecnologia è stata implementata.

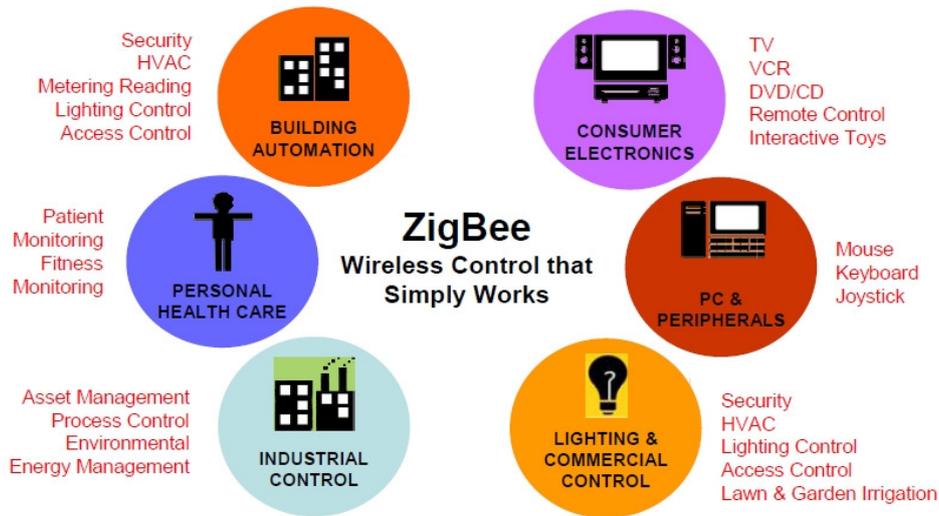


Figura 1: Panoramica delle possibili applicazioni della tecnologia ZigBee.. [17]

In figura i nomi indicati nei cerchi rappresentano il contesto in cui è indicato utilizzare la tecnologia ZigBee, mentre in rosso sono indicate le applicazioni che è possibile implementare in tali contesti. Le applicazioni indicate mostrano come la tecnologia ZigBee sia stata implementata principalmente per lo sviluppo di sistemi di controllo e monitoraggio.

Ai giorni nostri, una tematica di particolare importanza è la sicurezza nei luoghi di lavoro. Dato il successo delle reti WSN nell'implementazione di sistemi di controllo in ambienti di lavoro di dimensioni notevoli come industrie o ospedali, nell'ultimo decennio, parte della ricerca è stata dedicata allo sviluppo di un sistema in grado di localizzare i nodi della rete WSN, tale sistema è indicato come "sistema di localizzazione".

*Localizzare un oggetto significa conoscere le sue coordinate in un sistema di riferimento, ad esempio le coordinate geografiche di longitudine, latitudine e altitudine.*

Avere un sistema di localizzazione efficiente all'interno della rete permetterebbe di monitorare e salvaguardare persone e beni in possesso di un nodo. Inoltre, per alcune applica-

zioni dei sistemi di controllo o monitoraggio, potrebbe essere utile conoscere le coordinate di un nodo non note a priori. Se per esempio ci fosse un incendio all'interno di una struttura in cui è presente una rete WSN sarebbe possibile conoscere la posizione delle persone che indossano un nodo della rete.

Occorre poi notare come l'utilizzo delle reti WSN sia particolarmente utile per l'implementazione di sistemi di localizzazione all'interno di un ambiente chiuso (*indoor*), in cui non è possibile utilizzare il sistema di localizzazione satellitare *Global Positioning System* (GPS).

Inoltre, avere un sistema di localizzazione costituito da nodi a radio frequenza è particolarmente più efficiente rispetto ad un sistema di controllo costituito da telecamere. La localizzazione di una persona o un oggetto in possesso di un nodo all'interno di un sistema costituito avviene infatti solo se l'oggetto è nella zona di vista monitorata da ciascuna telecamera mentre, all'interno di una rete WSN, la localizzazione avviene in maniera automatica all'interno del raggio di copertura delle rete stessa.

L'obiettivo del lavoro eseguito è stato dunque quello di implementare un sistema di localizzazione da poter applicare ad una qualsiasi rete WSN preesistente costituita da nodi ZigBee. La scelta d'implementare un sistema di localizzazione con tale tecnologia è dovuta alle seguenti motivazioni:

- La tecnologia ZigBee è particolarmente adatta all'implementazione di sistemi di controllo e monitoraggio rispetto alle altre tecnologie Wireless. Tale aspetto sarà ampiamente illustrato nel capitolo II;
- Dato che l'uso di tale tecnologia non è ancora del tutto affermato nell'implementazione di questi sistemi, risulta utile verificare la possibilità di implementare un sistema di localizzazione efficiente per possibili applicazioni future.

Per valutare le prestazioni del sistema implementato nella parte V sarà eseguito un confronto prestazionale con un sistema di localizzazione già presente in commercio che utilizza la tecnologia Wireless Wi-Fi. Tale sistema è stato implementato dall'azienda Ekahau e sarà analizzato nella parte IV.

Nel seguito del presente capitolo verranno illustrati i possibili algoritmi di localizzazione e i parametri da poter impiegare per sviluppare un sistema di localizzazione con la tecnologia WSN.

## 2 Sistemi di localizzazione

Per poter procedere con l'analisi dei sistemi di localizzazione presenti in letteratura è utile descrivere gli elementi che costituiscono tali sistemi. In particolare nelle reti dei sistemi di localizzazione sono presenti due tipologie di nodi :

- *Nodo da localizzare*: Tale oggetto viene denominato *Unknown* e rappresenta il nodo di cui si vogliono rivelare le coordinate.
- *Nodo/i di riferimento*: I nodi di riferimento, detti ancore, sono gli elementi della rete WSN che permettono di ottenere le informazioni necessarie al sistema di localizzazione per stimare la posizione del nodo Unknown. In particolare sono definite due tipologie di configurazione della rete WSN in funzione della disposizione di tali elementi. Se i nodi di riferimento hanno una posizione fissa il sistema viene definito *Anchor-Based*, se i nodi non hanno una posizione statica nel tempo il sistema viene definito *Anchor-Free* [6] si pensi ad esempio a dei sensori lancianti in mare per monitorare l'attività delle maree.

In un sistema di localizzazione per ambienti industriali “o generalmente *indoor*” si suppone che i nodi di riferimento siano statici . In particolare la presente ipotesi è ragionevole se si pensa ad un sistema di localizzazione a lungo termine dove si rende necessario alimentare i nodi di riferimento con l'alimentazione elettrica. In un sistema di localizzazione *real-time* infatti i ricevitori devono essere in continuo “ascolto” quindi con un consumo energetico elevato. Nel presente elaborato verranno dunque analizzati unicamente sistemi di localizzazione in configurazione *Anchor-Based*.

Ai giorni nostri sono stati realizzati diversi prototipi di sistemi di localizzazione ognuno dei quali utilizza tecniche e tecnologie diverse atte a fornire la posizione del nodo incognito. Una prima distinzione netta tra i sistemi di localizzazione può essere fatta in base al tipo dato utilizzato per stimare la posizione del nodo. In particolare si hanno :

- *Sistemi Range-Based*: In tali sistemi, elaborando il valore di alcuni parametri descritti in seguito, è possibile ottenere una stima della distanza tra l'ancora e il nodo incognito. Avendo a disposizione tre o più valori di distanza con Ancore diverse e le coordinate delle stesse ancore è possibile applicare uno specifico algoritmo per stimare la posizione del nodo incognito.

- *Sistemi Range-Free*: In questi sistemi viene trascurata la possibilità di ottenere informazioni sulla distanza tra due nodi. La localizzazione viene effettuata utilizzando tecniche di stima alternative che saranno discusse nella sezione 2.3.

La scelta di sistema di localizzazione da implementare è dettata dal tipo di tecnologia impiegata per implementare tale sistema.

La messa in opera di un sistema di un localizzazione in un ambiente è composta dalla fase della “Calibrazione” e dalla fase di “Localizzazione”:

- *Calibrazione*: Le modalità con cui tale fase viene eseguita dipendono dal tipo di sistema di localizzazione implementato.
- *Localizzazione*: Questa operazione corrisponde alla fase di lavoro del sistema. La localizzazione di un nodo viene eseguita elaborando i dati ottenuti dall’interazione *Unknown-Ancore*, secondo un opportuno algoritmo.

Nel seguito saranno analizzate le tecniche di localizzazione finora sviluppate andando a valutare la possibilità di utilizzare tali algoritmi con la tecnologia da noi utilizzata.

## 2.1 Tecniche Range-Based: Stima della distanza tra nodi

I parametri principali utilizzati nelle tecniche *Range-Based* per stimare la distanza tra due nodi della rete sono i seguenti:

- *Tempo di Arrivo* (TOA)
- *Tempo di Volo* (TOF)
- *Angolo di Arrivo* (AOA)
- *Receive Signal Strength Indicator* (RSSI)

### 2.1.1 Tempo di Arrivo

Tale parametro viene utilizzato nel sistema GPS per stimare la distanza tra il satellite e il ricevitore GPS [53]. Come illustrato in Fig.2, è possibile stimare la distanza tra due nodi conoscendo il tempo  $t_t$  che impiega un “pacchetto trasmesso” (*beacon*) per raggiungere il nodo ricevente. Noto  $t_t$  e ipotizzando la velocità di propagazione del segnale a RF costante, è possibile ricavare la distanza tra i due nodi attraverso la ben nota relazione:

$$d = c \cdot t_t \quad (1)$$

dove:

- $c$  è la velocità di propagazione nel vuoto ( $c \simeq 3 \cdot 10^8$ ) [m/s];
- $d$  è la distanza tra i due nodi [m];
- $t_t$  tempo di arrivo [s].

Per poter misurare correttamente il tempo di Arrivo devono essere soddisfatte le condizioni di:

- *Sincronizzazione*: I due nodi devono essere sincronizzati allo stesso orario. Il nodo trasmittente invia infatti un “pacchetto” il cui contenuto informativo si identifica nell’orario in cui è stato trasmesso. In ricezione, per ottenere il tempo necessario a raggiungere il nodo ricevente, viene fatta la differenza tra l’orario in cui è stato ricevuto il pacchetto e l’informazione oraria contenuta in esso. Considerando la relazione (1) occorre notare che se gli orologi sono sfasati di 10 ns la misura di  $d$  è errata di 3 m. Risulta dunque evidente che gli “orologi” dei nodi della rete dovrebbero essere perfettamente sincronizzati per non incorrere in errori. Per tale motivo nel sistema GPS, per stimare la posizione di un nodo, viene utilizzato un quarto satellite necessario a sincronizzare (calibrare) i nodi della rete utilizzando il clock di un orologio atomico;
- *Distanza e Clock*: Oltre al problema della sincronizzazione esiste un problema legato alla distanza e al clock. Nelle reti *WPAN*, essendo la distanza massima tra due nodi dichiarata a  $\simeq 100$  m, si avrebbe un tempo  $t_{tMAX} \simeq 333$  ns. I chip ZigBee come l’MC13224 della Freescale hanno un clock interno massimo pari 26 MHz [24] ovvero

un periodo  $T = 39$  ns. Il periodo di clock dovrebbe essere dell'ordine dei ps per non introdurre errori rilevanti nella stima della distanza dalla misura del tempo  $t_t$ .

Il parametro detto TOA non può essere dunque utilizzato per stimare la distanza tra due nodi in una rete WPAN.

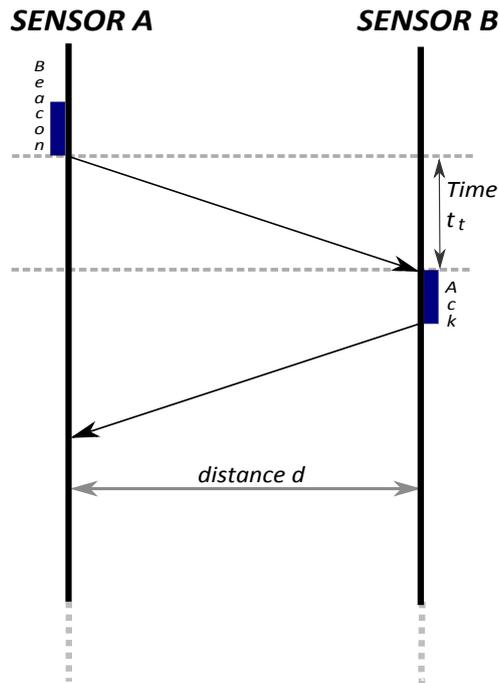


Figura 2: Tempo di Arrivo.

### 2.1.2 Tempo di Volo

Per evitare le problematiche legate alla sincronizzazione, in alcune applicazioni, si è pensato di utilizzare il parametro detto *tempo di volo*. Il TOF è una parte del *round trip time* ovvero il tempo necessario ad un nodo ad inviare un messaggio e ricevere risposta. Il principio su cui si fonda viene illustrato in Fig.(3).

In figure il nodo A trasmette un pacchetto (*beacon*) che impiega un tempo  $t_p$  (*tempo di viaggio*) per giungere al nodo B. Il nodo B, ricevuto il messaggio, risponde con un pacchetto di conferma (*Ack*) dopo un tempo  $t_{REPLYB}$  (*computing time*) necessario ad elaborare i dati. Il pacchetto inviato verso il nodo sorgente impiega nuovamente un tempo  $t_p$  per giungere a destinazione.

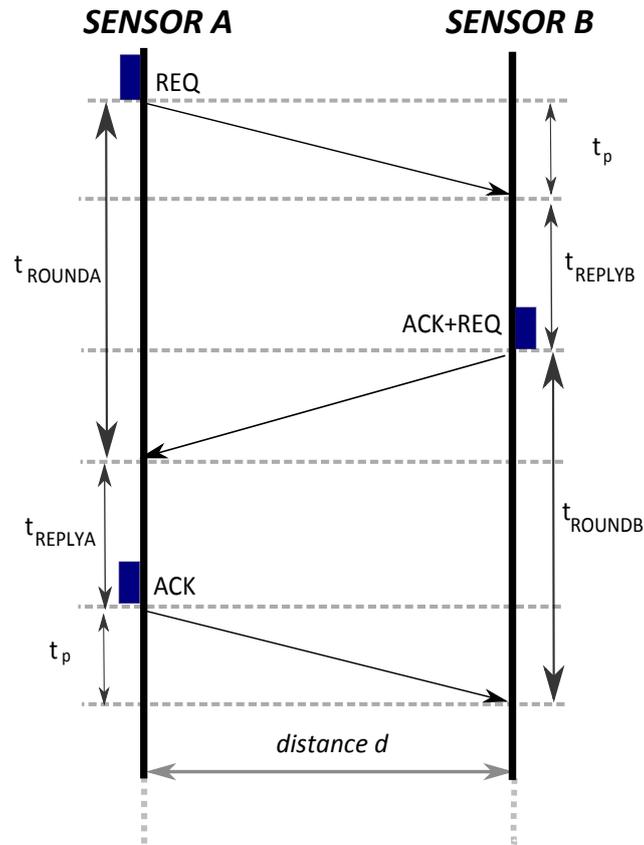


Figura 3: Scambio di pacchetti per la misura del TOF.

Il nodo A, grazie a un contatore, registra il tempo necessario per l'esecuzione di queste operazioni ovvero  $t_{ROUND A}$ . Noto  $t_{REPLY B}$  è possibile ricavare:

$$t_{of} = t_{ROUND A} - t_{REPLY B} \quad (2)$$

dove:

- $t_{of}$  è il tempo di volo [s]

Da notare che:

$$t_{of} = 2t_p \quad (3)$$

Dunque, dalla relazione(1) si ottiene la distanza  $d$  tra nodo A e nodo B come:

$$d = c \left( \frac{t_{of}}{2} \right) \quad (4)$$

Il TOF è usato in tecnologie in cui non è possibile avere una sincronizzazione precisa come nel GPS.

Tale parametro non può però essere usato in un sistema di localizzazione costituito da una rete di nodi ZigBee. Sulle schede ZigBee viene spesso caricato un software che permette il multitasking. Questo fattore rende fortemente variabile il tempo  $t_{REPLYB}$ , che può assumere valori anche di alcuni ms. I due termini  $t_{ROUNDA}$  e  $t_{ROUNDB}$  sono in pratica molto maggiori della loro differenza (dell'ordine dei ns in ambienti *indoor*). Inoltre, essendo i clock dei chip ZigBee utilizzati non abbastanza elevati, la misura dei singoli termini  $t_{ROUNDA}$  e  $t_{ROUNDB}$  sarebbe affetta da un errore considerevole provocando un errore relativo troppo elevato sulla stima di  $t_{of}$ . Tali problematiche possono essere risolte se la tecnologia impiegata ha un clock elevato (GHz) e estremamente preciso.

In letteratura si trovano infatti numerosi studi volti ad utilizzare tale parametro nelle reti costituite da dispositivi UWB (*Ultra WideBand*), come accade per esempio nel sistema descritto in [7]. La UWB è una tecnica di trasmissione sviluppata per trasmettere e ricevere segnali mediante l'utilizzo di impulsi di energia a radiofrequenza di durata temporale estremamente ridotta (da poche decine di picosecondi a qualche nanosecondo), quindi con occupazione spettrale molto ampia. Tale tecnologia, utilizzando un sistema di temporizzazione estremamente preciso (potenzialità *hardware* superiori rispetto i nodi ZigBee), potrebbe permettere di stimare il TOF con sufficiente precisione. Tuttavia l'utilizzo di questa tecnologia non rientra nell'obiettivo di creare un sistema di localizzazione a basso costo. Tali dispositivi hanno infatti un costo di gran lunga superiore rispetto ai nodi utilizzati nelle reti WPAN.

Occorre aggiungere che l'azienda *Jennic* ha costruito una scheda ZigBee (JN5148) con hardware dedicato alla misura di questo parametro [http://www.jennic.com/files/product\\_briefs/JN-DS-JN5148M0-1v4.pdf](http://www.jennic.com/files/product_briefs/JN-DS-JN5148M0-1v4.pdf). Fatta eccezione per questo caso specifico, le schede ZigBee non hanno questo tipo di potenzialità. Utilizzando le schede Jennic non sarebbe dunque possibile implementare un sistema di localizzazione da applicare ad una qualsiasi rete ZigBee.

### 2.1.3 Angolo di Arrivo

La tecnica descritta in questa sezione impiega un array di antenne o un'antenna rotante molto direttiva in ricezione per ogni ancora della rete al fine di determinare, in un sistema di riferimento comune ad ogni nodo fisso, l'angolo di provenienza del segnale trasmesso dall'Unknown alle Ancore della rete, per questo detta *Angolo di Arrivo*. Nel caso di un array di antenne in ricezione si riesce a stimare l'angolo di provenienza di un segnale in base alle differenze di fase del segnale ricevuto da ogni antenna che costituisce l'array. Noto l'angolo di provenienza di almeno due Ancore distinte è possibile ottenere la stima della posizione dell'Unknown come mostrato in Fig.4.

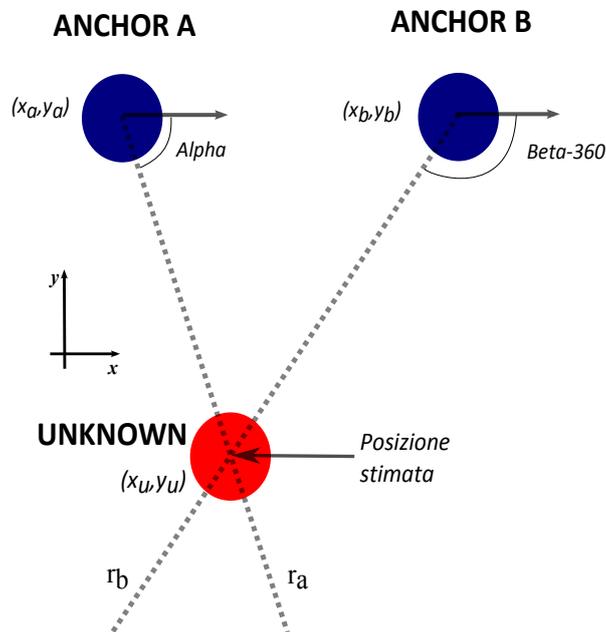


Figura 4: Esempio di localizzazione con il metodo AoA.

Infatti noti  $\alpha$  e  $\beta$  è possibile ricavare le equazioni delle rette  $r_a$  e  $r_b$ .

Risolvendo il sistema (5):

$$\begin{cases} r_1 \rightarrow y_1 = m_1x + q_1 \\ r_2 \rightarrow y_2 = m_2x + q_2 \end{cases} \quad (5)$$

se tale sistema è risolubile si ottengono le coordinate  $(x_u, y_u)$  del nodo di posizione incognita.

Tuttavia per implementare questa tecnica la parte radio del sensore deve essere sofisticata e questo si traduce in costi elevati (la parte radio di un sistema di trasmissione è onerosa), nella difficoltà della realizzazione dell'hardware e della gestione di esso e nell'incremento del volume/spazio occupato dal sensore. In letteratura infatti sono pochi gli studi orientati verso l'utilizzo di questa tecnica. Un prototipo di sistema sviluppato è illustrato in [48] anche se la prova sperimentale effettuata è limitata e insufficiente a provare le potenzialità di questa tecnica. Dato che i nodi utilizzati devono essere a basso costo e di piccole dimensioni (un array o un'antenna rotante direttiva occupa uno spazio considerevole relativamente all'ambiente di impiego delle reti in questione) la tecnica illustrata non può essere impiegata per raggiungere gli obiettivi prefissi dal presente elaborato.

#### 2.1.4 Receive Signal Strength Indicator

L'RSSI è un parametro correlato al valore della potenza del segnale ricevuto da un nodo. Dalla conoscenza del suo valore è possibile stimare la distanza tra nodo trasmittente e nodo ricevente. Generalmente i nodi ZigBee hanno la possibilità di misurare il parametro RSSI, dunque sarebbe possibile implementare un sistema di localizzazione utilizzando tale parametro.

Per comprendere come sia possibile stimare la distanza tra due nodi dal valore di RSSI si consideri la formula ottenuta dal Teorema del Collegamento:

$$P_R = P_T \frac{G_T G_R \lambda}{(4\pi)^2 d^n} \quad (6)$$

dove:

- $P_R$  potenza del segnale ricevuto [W]
- $P_T$  potenza del segnale trasmesso [W]
- $G_T, G_R$  guadagno di antenna in trasmissione e in ricezione;
- $\lambda$  lunghezza d'onda del segnale [m<sup>-1</sup>]
- $d$  distanza tra i due nodi [m]
- $n$  *path-loss* è numero puro. Dipende dalle condizioni di propagazione del segnale ( $n = 2$  in spazio libero)

Considerando  $G_T, G_R$  di valore unitario (antenne omnidirezionali) l'RSSI [dBm] è espresso nel modo seguente:

$$RSSI = P_{Tx} - P_{LOS} \quad (7)$$

dove:

- $P_{Tx}$  è la potenza trasmessa [dBm]
- $P_{LOS}$  è il *Propagation Loss*

$$P_{LOS} = -P(d_0) + 10n \log_{10} \left( \frac{d}{d_0} \right) \quad (8)$$

in cui  $d_0$  è la distanza di riferimento dove è fissato il trasmettitore e  $d$  è la distanza effettiva tra il trasmettitore e il ricevitore.

Considerando l'equazione (7) è possibile esprimere il valore dell'RSSI ottenuto a distanza  $d$  come :

$$RSSI(d) = RSSI(d_0) - 10n \log_{10} \left( \frac{d}{d_0} \right) \quad (9)$$

Sotto le ipotesi di  $P_{Tx} = 0$  dBm e  $d_0 = 1$  m la (9) si semplifica nel modo seguente :

$$RSSI(d) = - (10n \log_{10} d - P(d_0)) \quad (10)$$

Il valore dell'RSSI viene espresso in dBm essendo le potenze trasmesse dai nodi dell'ordine del mW.

Nei sistemi di localizzazione che utilizzano la relazione (10), per ottenere una stima della distanza tra i due nodi, la fase di Calibrazione è necessaria per stimare il valore del *path-loss*. Considerando il modello descritto dall'equazione (10) è possibile stimare il valore di  $n$  se si trasmette un segnale tra due schede poste a distanza fissata, conoscendo  $P(d_0)$  e misurando il valore dell'RSSI come:

$$n = - \left( \frac{RSSI - P(d_0)}{10 \log_{10} d} \right) \quad (11)$$

Noto  $n$ , è possibile stimare la distanza tra un nodo ricevente e un nodo trasmittente conoscendo la potenza trasmessa e misurando il valore dell'RSSI in ricezione secondo la seguente relazione ottenuta invertendo la (10) :

$$d = 10^{-\left(\frac{RSSI - P(d_0)}{10n}\right)} \quad (12)$$

In Fig. 5 in rosso viene mostrato l'andamento teorico della curva espressa dalla relazione (12) considerando  $n = 2$  ovvero comunicazione in spazio libero. Inoltre nella presente figura viene mostrato in blu l'andamento della relazione RSSI-Distanza ottenuta sperimentalmente durante il lavoro di tesi. Le misure sono state effettuate utilizzando in trasmissione e ricezione nodi ZigBee con installato il chip MC13224 della Freescale aumentando la distanza tra tali nodi. Le misure sono state svolte in spazio libero e facendo comunicare i nodi sempre in direzione di massimo trasferimento di potenza.

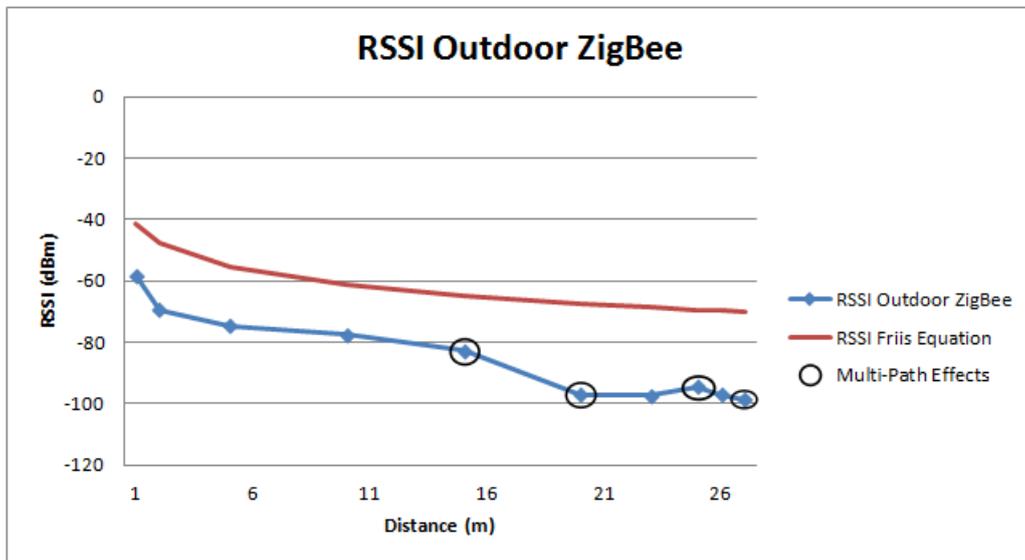


Figura 5: Andamento teorico e reale del valore di RSSI in funzione della distanza.

Dai risultati sperimentali ottenuti si evidenzia che nella realtà dei fatti la relazione RSSI-Distanza è più complessa rispetto quella descritta nell'equazione (10), infatti l'RSSI ricevuto ad una distanza fissata è spesso differente dal valore aspettato anche considerando un valore di *path-loss*  $n \neq 2$ .

Il problema principale del modello descritto dall'equazione (6) risiede nel fatto che in questa relazione non sono stati considerati i fenomeni tipici a cui sono soggetti i segnali a RF. L'aleatorietà del segnale ricevuto dipende principalmente dai seguenti fenomeni:

- *Multi-path*: Il *Multi-path* è un fenomeno legato alle riflessioni multiple. Un'antenna "omnidirezionale" irradia potenza in tutte le direzioni. Studiando il fenomeno con

un'analisi a raggi risulta evidente che una parte di essi, non essendo nella direzione della linea di vista tra i due nodi (*Line of Sight* LOS), incontrerà degli ostacoli. Una parte di potenza incidente verrà riflessa giungendo al ricevitore. Il raggio riflesso è caratterizzato da uno sfasamento rispetto il raggio diretto dovuto ad un percorso maggiore e da un'attenuazione aleatoria dovuta alla riflessione. In ricezione la combinazione del tutto aleatoria di tutti i raggi crea un fenomeno di distorsione del segnale ricevuto nominato *Fading* che ne altera il valore. Infatti a 2.4 GHz essendo la lunghezza d'onda del segnale trasmesso 12.5 cm un percorso di 7.25 cm maggiore provoca uno sfasamento di  $\pi$ . Gli effetti di questo fenomeno sono evidenziati in Fig.5.

- *Assorbimento*: Il fenomeno dell'assorbimento incide particolarmente sul valore del segnale ricevuto nei casi in cui la LOS viene oscurata (*Shadowing*). I nodi utilizzati nelle reti WPAN (*Wi-fi, ZigBee, Bluetooth, etc..*) trasmettono alla frequenza libera di 2.4 GHz. Tale valore è vicino alla frequenza di risonanza dell'acqua (2.45 GHz). Dunque qualsiasi oggetto con una percentuale di questo elemento nella composizione (porte, muri, etc..) e persino le persone, poste tra trasmettitore e ricevitore, attenuano fortemente la potenza del segnale che li attraversa.
- *Guadagno e Polarizzazione dell'onda*: Nel caso particolare di un sistema di localizzazione in cui il segnale trasmesso dal nodo UnKnown è ricevuto da più antenne per ognuna di esse si dovrebbe considerare nell'equazione (6) un Guadagno in Trasmissione diverso da quello unitario, dato che per il trasmettitore le antenne riceventi sono poste in direzioni diverse. Inoltre la relazione (6) è stata ottenuta facendo le ipotesi di adattamento di polarizzazione dell'onda. Nelle applicazioni reali tale ipotesi non è mai verificata.

Le problematiche descritte rendono in pratica il valore dell'RSSI non predicibile a priori. Attualmente vengono ancora svolte ricerche volte a migliorare il modello semplificato mostrato in questa sezione. Ad esempio in [6] si è tentato di introdurre nel modello le attenuazioni dovute alla polarizzazione ed al guadagno d'antenna.

Sebbene sia nota l'aleatorietà del valore della potenza in ricezione e l'imprecisione del modello descritto, l'RSSI è il parametro più usato in letteratura nei sistemi di localizzazione implementati in reti WPAN data la sua facilità di misura e la sua relazione "diretta" con la distanza.

## 2.2 Tecniche Range-Based : Algoritmi di stima utilizzando come parametro l'RSSI

Data la sua facilità di misura, l'RSSI è il parametro più utilizzato per stimare la distanza tra due nodi nelle reti WPAN. Nei sistemi di localizzazione Range-Based il parametro RSSI viene utilizzato sia durante la fase operativa per stimare la posizione del nodo incognito per mezzo della relazione (12) o formulazioni più complesse, sia durante la fase della calibrazione per stimare i parametri descritti nel modello del segnale utilizzato quali path-loss, il coefficiente di polarizzazione d'antenna, attenuazione dovuta alle pareti nel caso NLOS, etc...

Una volta note le distanze tra le ancore e l'Unknwon è possibile stimare la posizione del nodo applicando uno tra i seguenti algoritmi:

- *Min-Max*
- *Trilaterazione*
- *Multilaterazione Atomica*

### 2.2.1 Min-Max

L'algoritmo illustrato in questa sezione è stato ideato da *Andreas Savvides* [9]. L'algoritmo Min-Max è un metodo molto diffuso là dove le potenzialità *hardware* (memoria e velocità di calcolo) disponibili del sistema sono limitate, ovvero nei casi dove viene richiesta una buona precisione di stima a basso costo computazionale [10].

Infatti, nonostante la sua semplicità di realizzazione e il suo ridotto costo computazionale, permette di ottenere stime comunque vicine a quelle che si otterrebbero applicando l'algoritmo della multilaterazione come mostrato in [10].

L'idea principale su cui si fonda l'algoritmo è quella di costruire dei quadrati per ogni punto di riferimento. La posizione di ogni ancora è il centro di un quadrato il cui lato è il doppio della distanza stimata tra nodo incognito e l'ancora stessa. In reti WPAN costituite da nodi basati su specifiche IEEE 802.15.4 o IEEE 802.11 è possibile stimare la distanza tra nodi utilizzando il parametro RSSI come descritto in precedenza. Come illustrato in Fig.6 costruendo tre quadrati è possibile stimare la posizione del nodo incognito come centro del quadrilatero formato dall'intersezione dei tre.

Dunque il quadrato per l'a-esimo nodo fisso  $(x_a, y_a)$  a distanza  $d_a$  è definito dalla seguente relazione :

$$[x_a - d_a, y_a - d_a] \times [x_a + d_a, y_a + d_a] \quad (13)$$

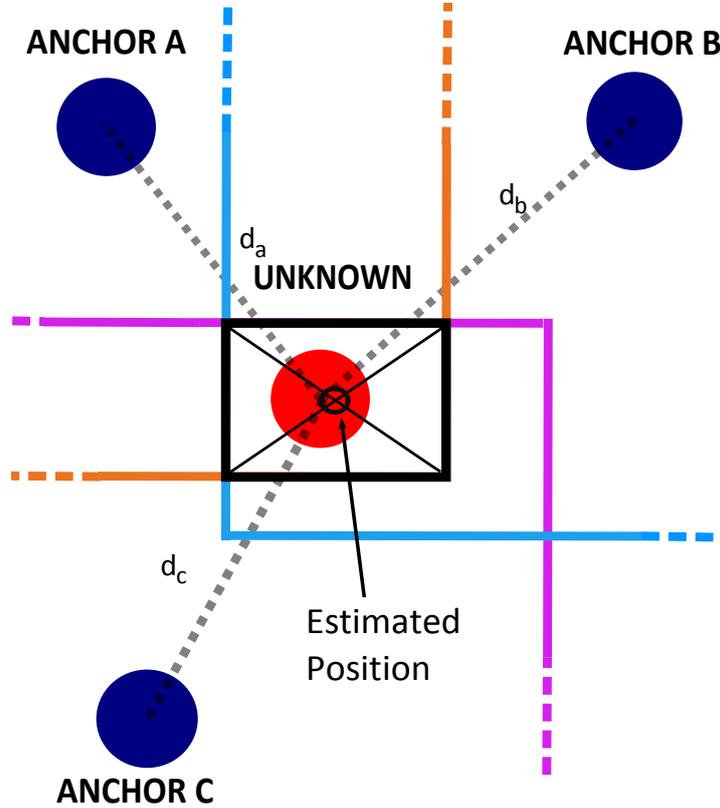


Figura 6: Localizzazione applicando il metodo Min-Max.

Gli estremi del quadrilatero di intersezione sono calcolati prendendo i massimi e i minimi di tutte le coordinate:

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \quad (14)$$

Data l'imprecisione della stima della distanza tra i nodi, dovuta ai problemi legati all'utilizzo del parametro RSSI, ed essendo l'algoritmo di per se non preciso, la stima della posizione dell'Unknown sarà affetta da errore. Per valutare la bontà della stima effettuata risulta possibile calcolare il valore di un parametro di bontà detto Residuo, dato

dalla somma degli scarti tra le distanze misurate e le distanze calcolate tra le ancore e la posizione stimata  $(x_u, y_u)$  ovvero:

$$residuo = \frac{\sum_{i=1}^n \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} - d_i}{n} \quad (15)$$

in cui:

- $n$  è il numero di nodi di riferimento che sono nel raggio di copertura del nodo incognito.

### 2.2.2 Trilaterazione

La trilaterazione è un algoritmo che permette di trovare, note le distanze da tre Ancore della rete, la posizione di un nodo di coordinate non note a priori. Il principio su cui si fonda l'algoritmo è il seguente :

- si consideri il problema di geometria piana descritto in Fig.7-a. Osservando la Fig.7-b risulta evidente che, tracciando tre circonferenze di raggio pari alla distanza Ancore-Unknown centrate nelle rispettive ancore, il punto di intersezione di esse corrisponde alla posizione del nodo incognito. Risulta dunque possibile ottenere le coordinate dell'Unknown calcolando le coordinate del punto P.

Da notare che nel piano bidimensionale sono necessarie tre circonferenze per stimare univocamente la posizione dell'oggetto di interesse. Generalmente l'intersezione di due circonferenze genera infatti due punti, quindi ambiguità nella stima. L'impiego dell'algoritmo della trilaterazione può essere esteso allo spazio tridimensionale calcolando il punto di intersezione della superficie di tre sfere di raggio noto. Il GPS si serve di tale algoritmo per stimare la posizione del nodo incognito.

Facendo riferimento alla Fig.(7) viene analizzato il problema dal punto di vista matematico. Per poter ottenere le coordinate del punto P deve essere risolto il seguente sistema di equazioni :

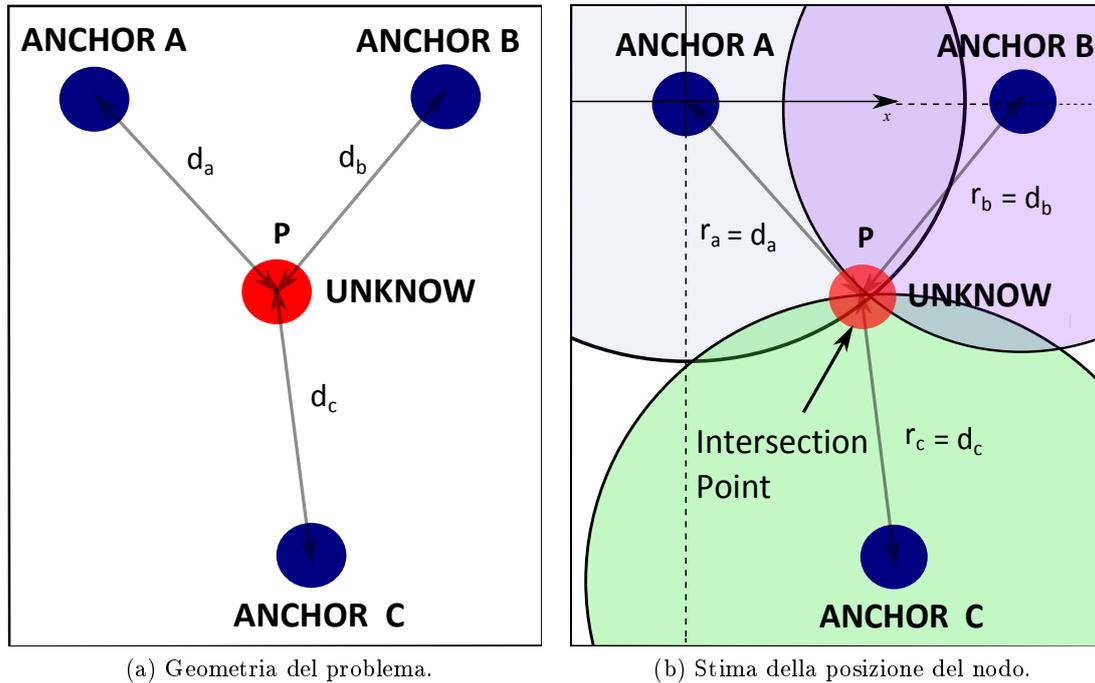


Figura 7: Trilaterazione

$$\begin{cases} r_a^2 = (x - x_a)^2 + (y - y_a)^2 \\ r_b^2 = (x - x_b)^2 + (y - y_b)^2 \\ r_c^2 = (x - x_c)^2 + (y - y_c)^2 \end{cases} \quad (16)$$

dove :

- $r_a, r_b, r_c$  sono i raggi delle tre circonferenze che coincidono con le distanze  $d_i$  ricavate dai valori degli RSSI
- $(x_i, y_i)$  sono le coordinate delle Ancore in un sistema di riferimento fissato

Nell'ipotesi in cui  $x_a = 0, y_a = 0, y_b = 0$  la risoluzione del sistema di equazioni (16) viene semplificata. Sottraendo infatti la seconda equazione dalla prima e risolvendo rispetto  $x$  si ottiene:

$$x = \frac{r_a^2 - r_b^2 + x_b^2}{2 \cdot x_b} \quad (17)$$

ottenuto il valore della coordinata  $x$ , sottraendo la prima equazione dalla terza si ottiene:

$$y = \frac{r_a^2 - r_c^2 + x_c^2 + y_c^2}{2 \cdot y_c} - \frac{x_c}{y_c} \cdot x \quad (18)$$

dalla quale sostituendo nella (17) la(18) si ottiene il valore della coordinata  $y$ .

Eliminando le ipotesi iniziali ( $x_a = x_b = y_b = 0$ ) il sistema (16) resta risolubile anche se la sua risoluzione diventa più complessa, inoltre aggiungendo la variabile  $z^2$  si crea un sistema tra sfere le cui soluzioni sono le coordinate del loro punto di intersezione nel piano tridimensionale .

Purtroppo, nelle applicazioni reali, i casi da risolvere sono più complessi del caso ideale affrontato in Fig.(7). Il legame RSSI-Distanza infatti non è deterministico anzi, fenomeni aleatori modificano il valore dell’RSSI ricevuto come descritto nella sezione 2.1.1. Detto questo, i raggi delle circonferenze ottenuti dalla relazione dell’RSSI-Distanza non sempre genereranno circonferenze intersecanti in un solo punto a causa delle distorsioni e dei disturbi che rendono il modello descritto dalla relazione (12) lontano dalla realtà. In generale gli errori fatti sulla stima dei raggi possono essere anche considerevoli e possono generare sia circonferenze che si intersecano in più punti come mostrato in Fig.(8-a) sia circonferenze che non si intersecano in nessuno punto Fig.(8-b).

Nella pratica sono numerose le casistiche che si possono affrontare, ma per quanto riguarda l’applicazione dell’algoritmo della trilaterazione è bene riassumere tutte le possibilità in due situazioni:

- *Tutti i Cerchi sono intersecanti*: In questo caso deve essere verificata la condizione  $l_{ij} < r_i + r_j$  per  $\forall i, j$  dove  $l_{ij}$  è la distanza reale tra  $Anchor_i$  e  $Anchor_j$  mentre  $r_i, r_j$  sono i raggi delle circonferenze ottenuti dalla stima delle distanze tra ancore e Unknown. Questo è il caso pseudo-ottimo, ovvero il caso in cui tutti i cerchi si intersecano tra loro. I cerchi intersecandosi generando delle aree di decisione individuate da triangoli con vertici i punti di intersezione dei cerchi. La stima delle posizione viene attuata individuando la zona di decisione di estensione minore e mediando le coordinate del triangolo costruito su essa. In Fig.9 viene illustrato un esempio del

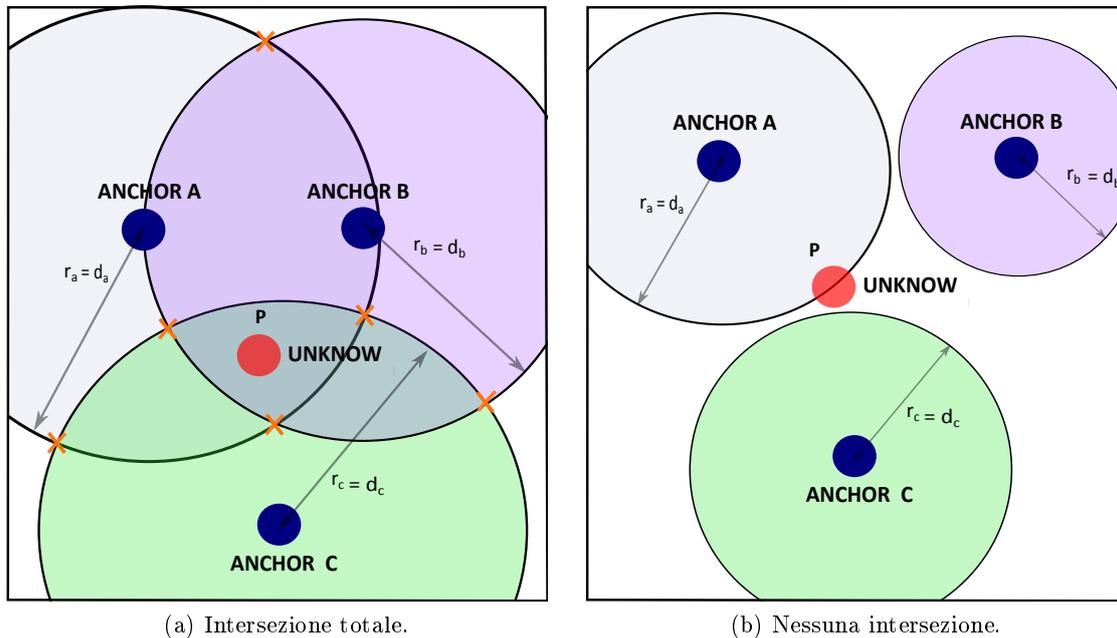


Figura 8: Trilaterazione, possibili intersezioni dei cerchi.

metodo di stima descritto. In figura i cerchi si intersecano in sei punti differenti individuati dalle lettere dell'alfabeto. Successivamente, avendo il triangolo  $\widehat{EGF}$  area minore, viene stimata la posizione del nodo mediando le sue coordinate.

- *I Cerchi sono disgiunti e/o tangenti* : Se una o più coppie di cerchi soddisfano questa condizione  $l_{ij} \geq r_i + r_j$  significa che non tutti i cerchi si intersecano tra loro. Ad esempio può accadere che i cerchi non si intersechino oppure che se ne intersechino solo due. Dato che non sarebbe efficiente trattare singolarmente ogni casistica diversa da quella precedentemente illustrata, è stato ricercato un metodo unico per stimare la posizione del nodo da applicare ad ogni casistica. Infatti nel caso in cui due o più cerchi siano tangenti o disgiunti tra loro vengono aumentati, secondo un certo criterio, i raggi delle tre circonferenze fino a che queste non si incontrano in un punto. Dopodiché viene stimata la posizione del nodo andando a calcolare le coordinate del punto di intersezione infatti. In questo modo, anche se con errore, viene comunque fornita una stima della posizione in qualunque situazione. In Fig.10 è illustrato un esempio del metodo descritto. In figura, inizialmente i cerchi generati dalle Ancore

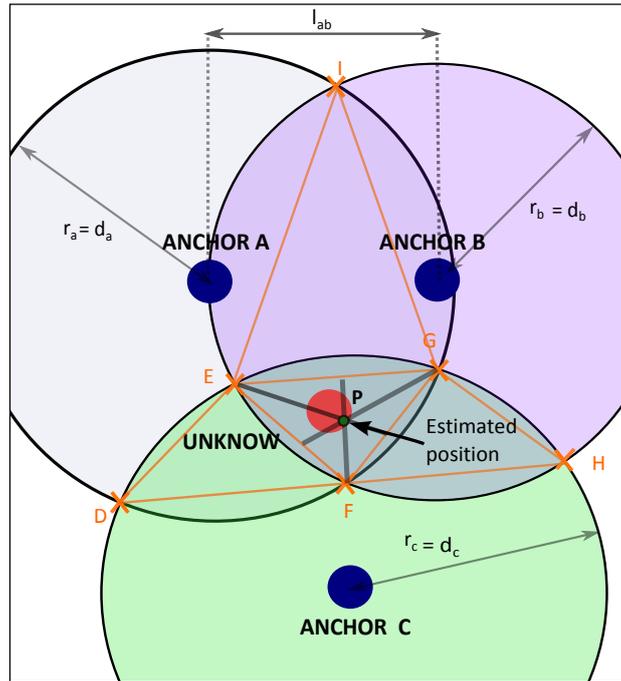


Figura 9: Trilaterazione, soluzione nel caso che cerchi si intersecano.

A,B,C non si intersecano. Successivamente vengono aumentati i raggi dei cerchi in modo tale da stimare fornire la stima delle coordinate del nodo.

Per valutare la bontà della stima effettuata risulta possibile calcolare il valore *residuo* come nel metodo *Min-Max*.

Occorre notare che nel secondo caso illustrato, nonostante le misure dei raggi siano errate, viene ugualmente fornita una stima della posizione. In tale caso il valore del residuo sarà elevato e la stima non sarà dunque affidabile.

### 2.2.3 Multilaterazione

L'algoritmo della multilaterazione è l'estensione del concetto della trilaterazione applicato ad un numero di sfere/circonferenze  $N > 3$ . Per migliorare l'accuratezza della stima può essere utile considerare le misure delle distanze rispetto a tutti i nodi di riferimento rilevati dal nodo mobile. Avendo a disposizione le misure rispetto  $n$  Ancore ( $n \geq 3$ ) si ottiene il seguente sistema di equazioni :

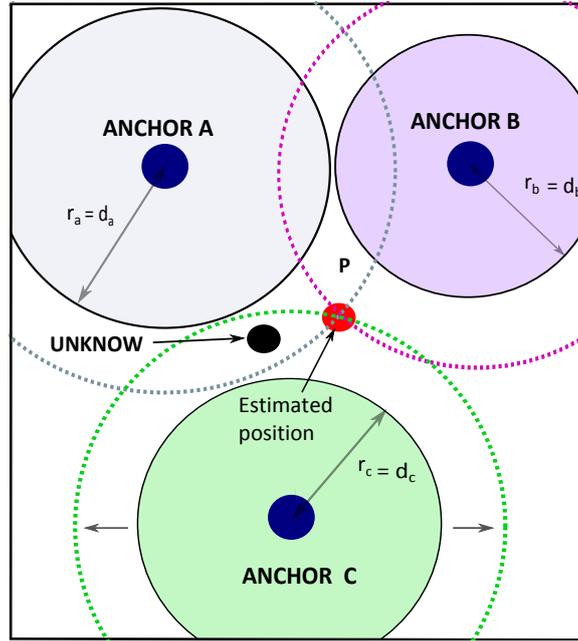


Figura 10: Trilaterazione, espansione dei raggi.

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \\ \vdots \\ (x - x_n)^2 + (y - y_n)^2 = r_n^2 \end{cases} \quad (19)$$

La  $i$ -esima equazione rappresenta la circonferenza di raggio  $r_i$  centrata sull' $i$ -esima ancora  $(x_i, y_i)$ . Il sistema (19) può essere linearizzato sottraendo l'ultima equazione dalle prime  $n - 1$  ottenendo :

$$\begin{cases} (x_1^2 - x_n^2) - 2 \cdot (x_1 - x_n)x + y_1^2 - y_n^2 - 2 \cdot (y_1 - y_n)y = r_1^2 - r_n^2 \\ (x_2^2 - x_n^2) - 2 \cdot (x_2 - x_n)x + y_2^2 - y_n^2 - 2 \cdot (y_2 - y_n)y = r_2^2 - r_n^2 \\ \vdots \\ (x_{n-1}^2 - x_n^2) - 2 \cdot (x_{n-1} - x_n)x + y_{n-1}^2 - y_n^2 - 2 \cdot (y_{n-1} - y_n)y = r_{n-1}^2 - r_n^2 \end{cases} \quad (20)$$

Riordinando opportunamente i termini del sistema (20) è possibile riportarlo nella

forma  $\underline{A}\underline{x} = \underline{b}$  dove :

$$\underline{A} = \begin{cases} -2(x_1 - x_n) & -2(y_1 - y_n) \\ -2(x_2 - x_n) & -2(y_2 - y_n) \\ \vdots & \vdots \\ -2(x_{n-1} - x_n) & -2(y_{n-1} - y_n) \end{cases} \quad \underline{b} = \begin{cases} r_1^2 - r_n^2 - y_1^2 + y_n^2 \\ r_2^2 - r_n^2 - y_2^2 + y_n^2 \\ \vdots \\ r_{n-1}^2 + r_n^2 - y_{n-1}^2 + y_n^2 \end{cases} \quad (21)$$

Poiché la matrice  $\underline{A}$ , essendo rettangolare, non può essere invertita, per risolvere il sistema di equazioni si ricorre alla matrice pseudo inversa di  $\underline{A}$  :

$$\underline{A}^+ = \left( \underline{A}^\top \underline{A} \right)^{-1} \underline{A}^\top \quad (22)$$

Se la matrice pseudo inversa esiste, è possibile risolvere il sistema lineare come :

$$\underline{x} = \left( \left( \underline{A}^\top \underline{A} \right)^{-1} \underline{A}^\top \right) \underline{b} \quad (23)$$

dove  $\underline{x}$  è il vettore che contiene le coordinate del punto stimato. Anche in questo caso un *residuo* ampio è indice che la stima effettuata è soggetta a forti errori. Se il valore del *residuo* supera una soglia fissata la stima viene scartata.

### 2.3 Tecniche Range-Free: Algoritmi di stima

Nel corso degli anni, una parte della ricerca dedicata allo sviluppo di sistemi di localizzazione *indoor*, ha implementato algoritmi che non utilizzano le informazioni ottenute dalla relazione RSSI-Distanza (da qui il nome *Range-Free*). Come illustrato nella sezione 2.1.1, data l'incertezza del legame RSSI-Distanza la localizzazione è spesso imprecisa. I ricercatori hanno tentato di sviluppare delle tecniche che utilizzano solo il valore dell'RSSI per cercare di ottenere risultati migliori rispetto ai metodi precedentemente illustrati.

Nelle tecniche Range-Free la fase della calibrazione ricopre un ruolo fondamentale. Durante la calibrazione viene suddiviso l'ambiente di lavoro in posizioni di riferimento come ad esempio una griglia. Successivamente vengono memorizzate in un data base le coordinate delle posizioni di riferimento denominate "*posizioni di calibrazione*" e le rispettive misure dell'RSSI ottenute dalle antenne in fase di ricezione del segnale trasmesso dal nodo di posizione incognita nelle suddette posizioni. L'operazione descritta serve a creare una *mappa* (Fingerprint) della distribuzione del valore di RSSI ricevuto nelle varie posizioni dell'ambiente in cui è installata la rete WSN. Tale mappa sarà utilizzata durante la fase di localizzazione insieme all'algoritmo per stimare la posizione del nodo incognito. Inizializ-

zare un sistema di localizzazione Range-Free richiede dunque tempi maggiori rispetto alle tecniche descritte in precedenza a fronte di un'accuratezza maggiore. Occorre notare che, nelle tecniche Range-Free, non è necessaria la conoscenza delle coordinate delle Ancore del sistema per stimare la posizione del nodo, ma è tuttavia necessaria la conoscenza delle coordinate delle posizioni di calibrazione. In questi sistemi il software che la stima della posizione del nodo è detto “*motore per la localizzazione*” o semplicemente “*Engine*”.

I principali algoritmi di localizzazione *fingerprint-based* sono i seguenti :

- *K-Nearest-Neighbor (kNN)*
- *Metodi Bayesiani*
- *Support Vector Machine (SVM)*
- *Multidimensional Vector Regression*

### 2.3.1 *K-Nearest Neighbor*

In questa tecnica la fase della calibrazione si compone delle seguenti azioni :

- L'ambiente osservato viene suddiviso in una griglia. Ogni centro delle celle che compongono la griglia corrispondono ad una posizione di riferimento.
- Da ogni posizione di riferimento un nodo trasmette un segnale in broadcast un numero di volte  $n$  fissato.
- Le ancore nel raggio di copertura del nodo trasmittente misurano i valori di RSSI dalle  $n$  segnalazioni. Al termine delle  $n$  segnalazioni ogni ancora media i valori di RSSI misurati.
- Successivamente i dati vengono raccolti da un *server* che si occupa di creare un *database*. Per ogni posizione di calibrazione vengono dunque registrati su un server l'identificativo delle ancore che hanno ricevuto il segnale, la media degli RSSI ricevuti e le coordinate della posizione .

Nel KNN [15] ,durante la localizzazione, il nodo di posizione incognita trasmette un segnale in broadcast. Le ancore che ricevono il segnale misurano il valore di RSSI. Successivamente

viene creato il vettore  $\underline{p}$  di RSSI misurati dalle ancore nel raggio di copertura del nodo. In base al vettore  $\underline{p}$  e agli identificativi delle ancore si cercano nel database le  $K$  posizioni candidate. La  $i$ -esima posizione viene scelta come candidata se l'errore quadratico medio tra il vettore  $\underline{p}$  e il vettore contenuto nel database associato a quella posizione minore di una soglia fissata. Per stimare la posizione dell'Unknown vengono mediate le posizioni dei  $K$  candidati :

$$(\hat{x}, \hat{y}) = \left( \frac{1}{K} \right) \cdot \sum_{i=1}^K (x_i, y_i) \quad (24)$$

La media può essere pesata secondo i valori delle distanze ottenuti.

Il fattore  $K$  assume il suo valore in base alla topologia della rete e alla mappa della zona da monitorare per ottenere le prestazioni migliori. Da notare che, variando la dimensione delle celle della griglia durante la fase della Calibrazione, l'accuratezza dell'algoritmo si modifica. Infatti più una griglia è fitta maggiore sarà l'accuratezza del sistema a discapito del tempo speso durante la fase della Calibrazione e durante il calcolo della posizione.

### 2.3.2 Metodi Bayesiani

Nei metodi Bayesiani viene trattato il problema della localizzazione come se fosse un problema di decisione statistica nel quale i dati registrati durante la fase di calibrazione sono utilizzati per costruire un modello statistico del valore di RSSI ricevuto da ogni posizione di calibrazione. In seguito saranno illustrati i concetti necessari a comprendere questo metodo.

Si assuma di avere un spazio di decisione  $L$   $n$ -dimensionale, dove  $l_i$  con  $i = 1, 2, \dots, n$  rappresenta la  $i$ -esima possibile posizione assunta dal nodo incognito. Al fine di stimare la posizione dell'Unknown viene scelta la posizione  $l_i$  secondo la seguente regola :

$$se \quad P(l_i/\underline{s}) > P(l_j/\underline{s}) \quad per \quad j = 1, 2, \dots, n \quad j \neq i \quad (25)$$

dove :

- $\underline{s}$  è il vettore di RSSI ottenuto durante la fase operativa dalla posizione da stimare. Ogni componente di  $s$  è associata ad un'ancora distinta.
- $P(l_i/\underline{s})$  è la probabilità di essere nella posizione  $l_i$  noto di aver ricevuto il segnale  $\underline{s}$ , detta probabilità a posteriori.

Come in ogni problema di decisione, la probabilità a posteriori non è predicibile. Dunque viene utilizzato il teorema di Bayes per scrivere la  $P(l_i/\underline{s})$  in un'altra forma ovvero :

$$P(l_i/\underline{s}) = \frac{P(\underline{s}/l_i) \cdot P(l_i)}{P(\underline{s})} = \frac{P(\underline{s}/l_i) \cdot P(l_i)}{\sum_{k=1}^n P(\underline{s}/l_k) P(l_k)} \quad (26)$$

Assumendo che la probabilità di trovarsi in una data posizione all'interno della WSN abbia una distribuzione uniforme, ovvero che le  $P(l_i)$  con  $i = 1, 2, \dots, n$  siano equiprobabili la (25) si trasforma nella (27). Viene scelta per cui la posizione  $l_i$  se:

$$P(\underline{s}/l_i) > P(\underline{s}/l_j) \quad \text{per } j = 1, 2, \dots, n \quad j \neq i \quad (27)$$

La funzione di verosimiglianza  $P(\underline{s}/l_i)$  viene stimata durante la fase Calibrazione e può essere aggiornata durante la fase operativa. Utilizzando la (27) vengono associate al nodo incognito solo posizioni discrete ovvero quelle impiegate per la fase della calibrazione. Facendo una media pesata delle posizioni di riferimento (28) è possibile ottenere una stima più accurata:

$$(\hat{x}, \hat{y}) = \sum_{i=1}^n (P(l_i/\underline{s}) \cdot (x_i, y_i)) \quad (28)$$

Da notare che in condizioni reali l'ipotesi di avere la variabile  $l$  con distribuzione uniforme non è del tutto vera. Infatti in un ambiente di lavoro, come un ufficio, la probabilità che una persona si trovi in certe zone della stanza sarà diversa rispetto altre zone. Ad esempio la probabilità che una persona si trovi nella posizione in cui è presente un tavolo sarà bassa. D'altro canto l'ipotesi di distribuzione uniforme è la più generale possibile e, anche nel caso in cui la  $P(L_i)$  fosse nota per  $\forall i$  la disuguaglianza espressa dalla 27 sarebbe pesata solo da delle costanti.

Per ottenere un'accuratezza adeguata risulta di fondamentale importanza stimare correttamente la funzione di verosimiglianza. Come evidenziato in [11], in letteratura esistono numerosi metodi per affrontare questa problematica, tra questi i più utilizzati sono i metodi *Kernel* e i metodi *Istogramma*.

Nei metodi Kernel viene stimata la funzione di verosimiglianza per ogni posizione  $l$  del *training set*. Facendo riferimento al caso monodimensionale ovvero un solo una ancora ed applicando il metodo alla  $l$ -esima posizione la funzione di verosimiglianza stimata risulta essere:

$$P(\underline{s}/l) = \frac{1}{n_l} \sum_{s_i=l} K(s; s_i) \quad (29)$$

dove:

- $n_l$  è il numero di campioni dell'RSSI collezionati durante la fase di calibrazione nella  $l$ -esima posizione
- $K(\cdot; o_i)$  è la funzione Kernel

Nel sistema descritto in [12] viene utilizzata come funzione  $K(\cdot)$  una Gaussiana:

$$K_{GAUSS}(s; s_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s - s_i)^2}{2\sigma^2}\right) \quad (30)$$

dove:

- $\sigma$  è la deviazione standard della funzione. Il valore di  $\sigma$  determina la larghezza della funzione.

Durante la fase operativa, ricevuto un valore di RSSI, viene applicata la (29) all'intero Training Set. Successivamente applicando la (28) viene stimata la posizione del nodo di coordinate incognite.

Nel caso  $n$ -dimensionale, assumendo l'indipendenza delle osservazioni, la probabilità globale  $p(\underline{s}/l)$  risulta essere il prodotto delle  $n$  probabilità individuali calcolate rispetto ad ogni ancora. Da notare che è possibile ottenere il metodo KNN come caso limite del metodo descritto sopra ponendo la varianza della funzione Kernel nulla.

Il metodo dell'istogramma offre la possibilità di stimare la funzione di verosimiglianza nel modo seguente: si assuma che la variabile osservata sia monodimensionale e che valori di massimo e di minimo assunti dalla variabile in questione siano noti. Si supponga inoltre di avere a disposizione  $n$  osservazioni  $(x_1, x_2, \dots, x_n)$  della variabile di cui si vuole stimare la densità di probabilità. L'intero range di valori possibili assunti dalla variabile (dal minimo al massimo) deve essere diviso in intervalli non sovrapposti detti *bins*. Ogni intervallo è definito nel modo seguente :

$$[x_0 + mh, x_0 + (m + 1)h] \quad (31)$$

dove:

- $m$  è un intero
- $h$  rappresenta la larghezza del bin

Detto questo la funzione di verosimiglianza viene stimata nel modo seguente:

$$\hat{f}(x) = \frac{(\text{Numero di elementi } X_i \text{ nel bin che contiene } x)}{n (\text{larghezza del bin che contiene } x)} \quad (32)$$

I risultati mostrati in [12] evidenziano che le prestazioni migliori in termini di errore medio sulla stima della posizione si ottengono utilizzando il metodo dell'istogramma . Inoltre in [12], viene mostrato nei test sperimentali che con entrambe le tecniche di stima della  $p(s/l)$  si ottiene un'accuratezza di stima della posizione migliore rispetto l'utilizzo del metodo KNN . In letteratura sono molti i prototipi di sistemi di localizzazione sviluppati secondo questo metodo tra i quali quelli descritti in [12] e [13]. Occorre notare che lo stesso programma Ekahau utilizza in parte un approccio Bayesiano [14].

### 2.3.3 Support Vector Machine

Le *Support vector machine* (SVM) sono state sviluppate negli anni '90 da Vladimir Vapnik e il suo team presso i laboratori Bell AT&T [49]. L'algoritmo *Support Vector*, su cui si fondano le SVM, è una generalizzazione dell'algoritmo *Generalized Portrait* sviluppato in Russia nei primi anni '60 e appartiene agli algoritmi della categoria *Statistical learning theory*. Una SVM è un classificatore binario che apprende il confine fra osservazioni appartenenti a due diverse classi Fig.11.

In vengono sono illustrate le osservazioni di due tipi variabili differenti separate “linearmente” nello spazio. Ciò significa che tracciando una retta è possibile classificare le

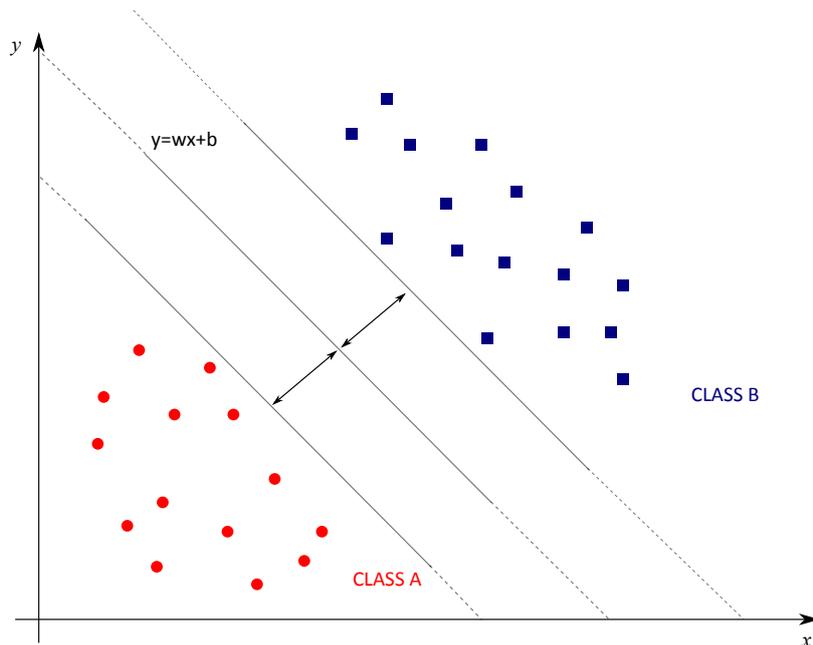


Figura 11: Dati linearmente separabili.

osservazioni come appartenenti a due classi distinte. Lo scopo della SVM è quello di trovare, tra tutti gli iperpiani che classificano correttamente le due classi, l'iperpiano che ha norma minima, cioè margine massimo rispetto ai punti delle classi distinte.

Utilizzando le support vector machine è possibile classificare anche variabili le cui osservazioni non sono linearmente separate nello spazio. Questa tecnica pertanto permette di mappare le osservazioni di due variabili distinte non linearmente separabili in uno spazio alternativo in cui queste sono linearmente separate Fig.12..

Le SVM sono impiegate nei sistemi di localizzazione per processare i dati di calibrazione in modo tale da rendere il problema di stima assimilabile ad un problema di classificazione. Nel documento [40] viene descritta questa tecnica facendo riferimento al problema della localizzazione.

Si supponga  $l$  il numero delle osservazioni nel training set. Si consideri inoltre la coppia  $(\hat{x}_i, y_i)$  in cui  $i = 1, 2, \dots, l$ . In un problema di classificazione, applicato ai sistemi di localizzazione Range-Free,  $\hat{x}_i$  rappresenta il vettore degli RSSI misurati mentre  $y_i$  è una variabile che assume valore 1 se la posizione corrisponde al punto di riferimento selezionato nel training set, -1 altrimenti. L'algoritmo descritto ha come obiettivo quello di costruire una mappa  $\hat{x}_i \rightarrow y_i$  attraverso la minimizzazione dell'errore empirico:

$$R_{emp}(\theta) = \frac{1}{l} \sum_{i=1}^l \|y_i - f(x_i, \theta)\| \quad (33)$$

dove

- $l$  è il numero di osservazioni
- $y_i$  sono le *label*
- $x_i$  sono i vettori contenenti i valori degli RSSI
- $f$  è la funzione che minimizza il rischio

La soluzione del problema matematico sopra riportato, esaurientemente descritta in [40], permette di ricavare una funzione che trasforma i dati di RSSI del training set in uno spazio in cui sono linearmente separabili. Tali sistemi sono anche noti come classificatori a massimo margine, poiché allo stesso tempo minimizzano l'errore empirico di classificazione (33) e massimizzano il margine geometrico.

Per meglio comprendere come tale tecnica possa essere utilizzata nei sistemi di localizzazione si osservi la Fig.12. Si supponga che i cerchi in figura siano associati a valori di RSSI (di calibrazione) e le due classi a due posizioni di calibrazione differenti. Ricevuto un valore di RSSI durante la localizzazione è possibile trasformare tale valore nello spazio dei segnali separato (tridimensionale in figura) e associarlo ad una posizione di calibrazione. Nel documento [40] viene mostrato un sistema di localizzazione implementato che utilizza tale algoritmo. Una valutazione delle prestazioni di tale sistema sarà effettuata nella sez. 3.

### 2.3.4 Multidimensional Vector Regression

Lo sviluppo e l'applicazione di questo algoritmo sono descritti in [42]. Il sistema descritto è stato implementato nel modo seguente :

- Durante la fase di calibrazione vengono misurati i valori di RSSI da delle posizione di riferimento nell'ambiente di localizzazione.

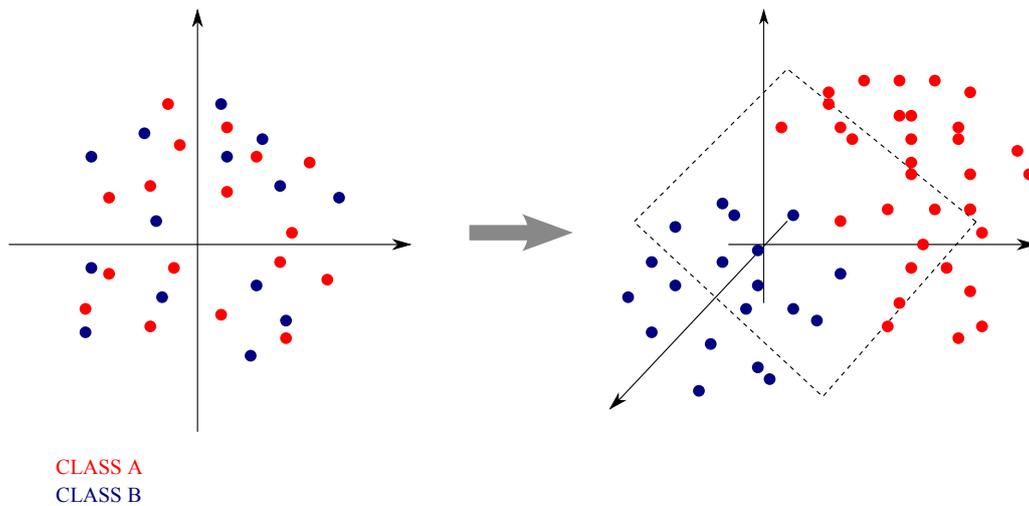


Figure 12: SVM, trasformazione.

- Noti i campioni di RSSI del training set e le posizioni di riferimento da cui sono stati misurati, viene applicato l'algoritmo KCCA (*Kernel Canonical Correlation Analysis*). Tale algoritmo permette di calcolare dei vettori (*Canonical Vectors*) che permettono di trasformare lo spazio dei segnali e lo spazio delle coordinate in due spazi a correlazione massima. Tale algoritmo opera quindi un pre-processing ai dati ottenuti durante la fase della calibrazione. La trattazione matematica di tale algoritmo è descritta in [42].
- Ottenute delle misure di RSSI durante la fase operativa, come prima operazione tali misure vengono trasformate utilizzando i vettori canonici. Dopodiché viene applicato l'algoritmo del KNN tra il vettore osservato trasformato e i campioni del training set proiettati nello spazio a massima correlazione.

## 2.4 Parametri per valutare le prestazioni di un sistema di localizzazione

I parametri utilizzati per valutare le prestazioni di un sistema di localizzazione sono molteplici. Il valore dell'accuratezza di un algoritmo non è sufficiente per definirne la bontà ma occorre considerare anche altri fattori. È inutile infatti avere una buona accuratezza se il sistema non è scalabile. Un buon sistema è quindi caratterizzato da un *trade-off* delle seguenti caratteristiche :

- **Accuratezza:** Si tratta del parametro più importante che caratterizza un sistema di localizzazione ed è associato al valore medio dell'errore della stima, ovvero la media delle distanze euclidee tra la posizione stimata e la posizione reale del nodo incognito. La bontà del valore di questa quantità è relativa all'ambiente in cui è applicato l'algoritmo. Risulta infatti inutile avere un'accuratezza di 1 m se l'obiettivo è localizzare in una stanza 2mx2m. Spesso è possibile migliorare il valore dell'accuratezza di un algoritmo sottraendo la *bias* dal suo valore, ossia un *offset sistematico* (errore) da cui è affetto il sistema.
- **Precisione:** L'accuratezza considera solamente il valore dell'errore medio della stima. La precisione è invece un parametro legato alla varianza della stima. La differenza tra queste due quantità è mostrata in Fig.13: in figura (a) è mostrato un caso estremo di stima precisa ma poco accurata visto che l'errore medio è elevato (*bias* elevata), mentre in figura (b) si illustra un tipico esempio di stima poco precisa ma accurata dato che il valore medio dell'errore è:  $E\{e\} = E\{p - \hat{p}\} = p - E\{\hat{p}\} = 0$  dove  $p$  è la posizione reale e  $\hat{p}$  è la stima.

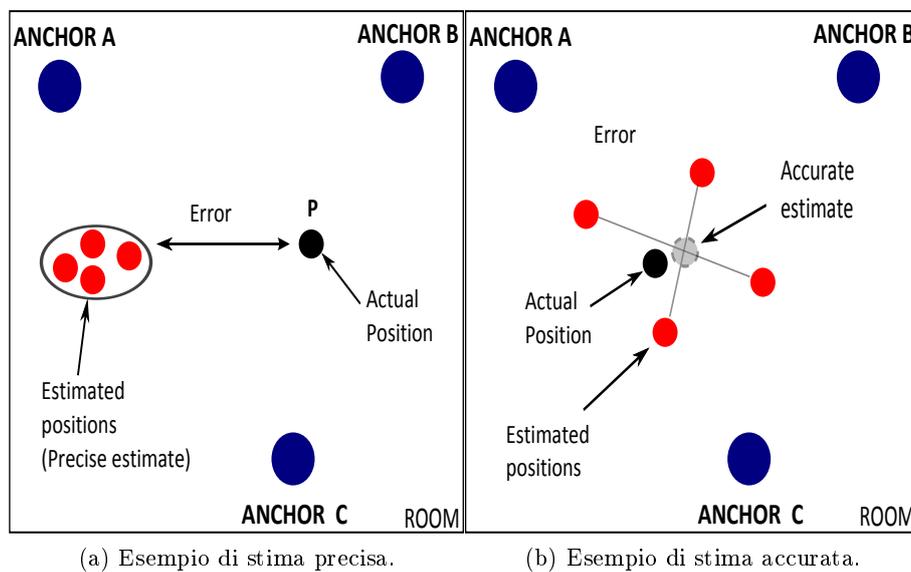


Figura 13: Differenza tra stima precisa e stima accurata.

- *Scalabilità*: Le prestazioni di un sistema di localizzazione devono essere quanto più indipendenti dalla dimensione della rete sia in termini geografici sia in termini di densità del numero di nodi (*Anchor/Unknow*).
- *Robustezza*: Un sistema di localizzazione deve essere robusto, ovvero non deve dipendere fortemente dalle condizioni dell'ambiente in cui viene utilizzato.
- *Complessità*: Il sistema non deve essere complesso dal punto di vista hardware (consumo energetico,..etc) e software (costo computazionale).
- *Costo*: Quest'ultimo fattore è fortemente correlato al precedente ed è molto importante, risulta infatti inutile creare un sistema efficiente se poi non è possibile commercializzarlo a causa dei costi troppo elevati. Come detto in precedenza l'obiettivo del presente elaborato è quello di sviluppare un sistema di localizzazione su una rete di nodi a basso costo.

### 3 Scelta dell'Algoritmo di localizzazione

Dopo un'analisi preliminare, è stato deciso di scegliere un algoritmo di localizzazione tra quelli presenti nelle tecniche Range-Free sezione 2.2. Tale decisione è stata dettata sia dalle caratteristiche hardware delle schede utilizzate per implementare la rete WSN sia dai fenomeni di multi-path, shadowing e assorbimento che rendono la potenza di un segnale a radio frequenza ricevuto in ambiente indoor fortemente variabile.

Per capire meglio le motivazioni che hanno portato a tale decisione, è stato eseguito un test misurando i valori di RSSI ricevuti tra due schede ZigBee (vedi sez.7) posizionate nel corridoio (24 m x 5 m) del Polo di Sistemi Logistici dell'Università di Pisa Fig.14 .

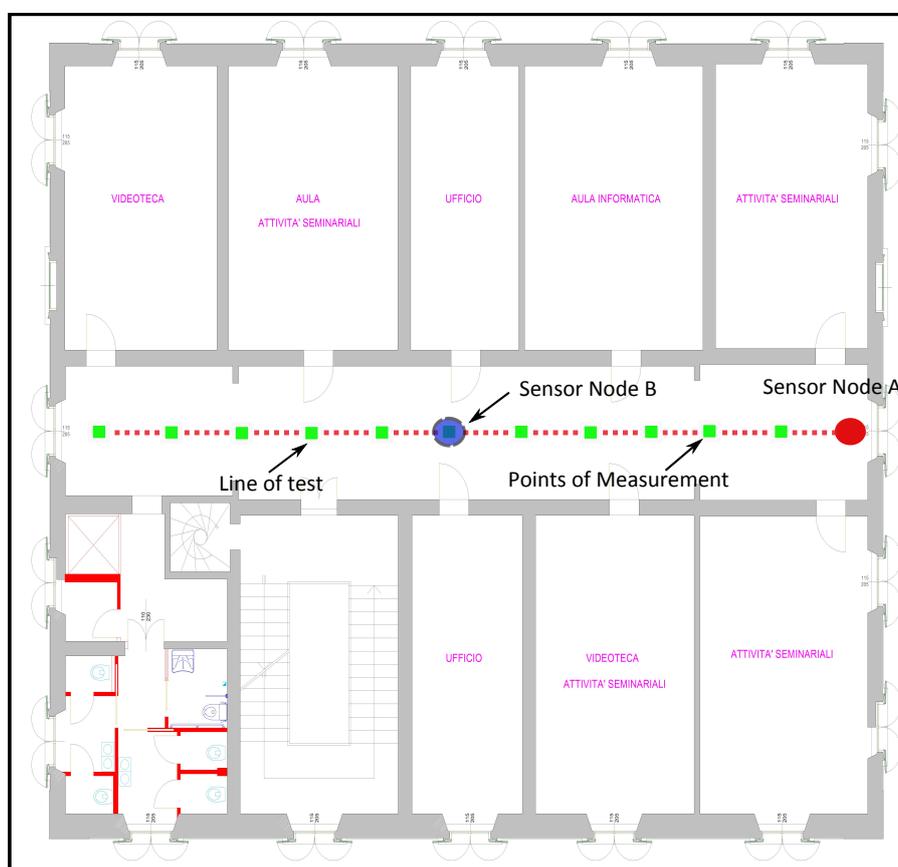


Figura 14: Test, misura dell'RSSI tra due nodi.

In figura è possibile osservare le posizioni di misura e la disposizione delle schede ZigBee, la distanza tra due posizioni differenti era un metro.

Le misure sono state effettuate con la scheda ZigBee B in tenuta in mano da una persona. Per ogni posizione di riferimento la persona è stata fatta orientare in cinque direzioni diverse, così da mostrare gli effetti dovuti ai fenomeni del multipath, shadowing e polarizzazione di antenna sul valore di RSSI misurato ad una distanza fissata. La valutazione di questi effetti è importante perché, durante la fase di localizzazione, la direzione e l'angolo in cui è posto un *tag* rispetto alle Ancore del sistema non sono noti a priori.

I valori di RSSI misurati durante il test sono illustrati in Tabella 1.

ANCHOR ID	DISTANCE [m]	MEASURE 1 [dBm]	MEASURE 2	MEASURE 3	MEASURE 4	MEASURE 5
ID0000000000000007	1	-53	-48	-61	-61	-56
ID0000000000000007	2	-67	-52	-71	-63	-61
ID0000000000000007	3	-61	-61	-70	-65	-65
ID0000000000000007	4	-59	-64	-72	-77	-73
ID0000000000000007	5	-64	-66	-80	-75	-74
ID0000000000000007	6	-76	-66	-77	-67	-76
ID0000000000000007	7	-77	-84	-84	-85	-80
ID0000000000000007	8	-79	-80	-77	-79	-81
ID0000000000000007	9	-76	-70	-84	-77	-75
ID0000000000000007	10	-78	-70	-88	-78	-80
ID0000000000000007	11	-81	-71	-88	-85	-89
ID0000000000000007	12	-69	-76	-86	-77	-74
ID0000000000000007	13	-79	-80	-85	-81	-76
ID0000000000000007	14	-86	-79	-90	-75	-77
ID0000000000000007	15	-73	-75	-79	-82	-84
ID0000000000000007	16	-86	-81	-78	-74	-79
ID0000000000000007	17	-76	-80	-81	-79	-88
ID0000000000000007	18	-72	-73	-75	-75	-73
ID0000000000000007	19	-75	-86	-76	-75	-80

Tabella 1: RSSI misurati in ogni posizione di riferimento.

Come è possibile osservare nei dati mostrati, i valori di RSSI misurati in ogni punto di riferimento hanno una grande varianza, ovvero la differenza tra il valore massimo e il valore minimo di RSSI misurati dalla stessa posizione è elevata.

Applicare l'algoritmo della trilaterazione utilizzando valori di RSSI affetti da tale ordine di disturbo, porterebbe ad avere uno stimatore di posizione con varianza elevata, quindi grande indecisione sulla bontà della stima.

Si supponga infatti di voler stimare la distanza tra due schede dal valore massimo e valore minimo di RSSI misurati alla distanza di 19 metri illustrati in tabella. Utilizzando

i dati in tabella è possibile stimare il valore di  $n$  (pathloss) dalla relazione (11) nel modo seguente:

- si consideri la distanza  $d_0 = 1$  m;
- si medino i valori di RSSI in dBm misurati alla distanza  $d_0$  e si associ questo valore alla variabile  $P_0$ . Dunque il valore di  $P_0 = -56$  dBm;
- per la stima del parametro pathloss si selezioni ad esempio la distanza di 13 metri e si medino i valori di RSSI misurati a tale distanza. Si associ questo valore al parametro RSSI dell'equazione (11). Il parametro risulta essere  $\text{RSSI} = 80$  dbm.

Ottenute le variabili necessarie al calcolo del parametro, il valore della pathloss risulta essere:

$$n = - \left( \frac{-80 + 56}{10 \cdot \log_{10}(13)} \right) = 2,15$$

Ottenuto il valore di  $n$  è possibile stimare la distanza tra i due nodi utilizzando come valori di RSSI:

- $\text{RSSI}_1 = 86$  dBm
- $\text{RSSI}_2 = 75$  dBm

Dunque i valori della distanza stimati utilizzando la 12 sono :

- $d_1 = 10^{-\left(\frac{-86+56}{10 \cdot 2,15}\right)} \cong 25$  m
- $d_2 = 10^{-\left(\frac{-75+56}{10 \cdot 2,15}\right)} \cong 8$  m

Quindi, con due misure effettuate dalla stessa posizione e in istanti di tempo che si differenziano di pochi secondi, sono state stimate due distanze differenti di 17 metri sui 19 metri effettivi. Osservando questi risultati è possibile fare le seguenti considerazioni :

- La differenza tra le due stime è molto elevata considerando che le misure sono state effettuate in un corridoio ampio e con pochi ostacoli.
- A parità di differenza in potenza, in questo esempio 11 dBm, la differenza tra le stime dei raggi aumenta all'aumentare della distanza. Questo fenomeno è dovuto alla non linearità della relazione tra le due quantità.

Con l'esempio precedente si dimostra che, in questo tipo di applicazioni, utilizzare un algoritmo di stima come la trilaterazione porterebbe ad avere un'elevata varianza della stima ed errori elevati sulla stessa anche in un piccoli ambienti. Questa è una caratteristica da evitare in un sistema di localizzazione.

Data la necessità di scartare gli algoritmi di stima appartenenti alle tecniche Range-Based per i motivi descritti in precedenza, le opzioni di scelta si sono concentrate tra le tecniche Range-Free. In Tabella 2 sono mostrate le prestazioni dei principali sistemi di localizzazione fino ad oggi implementati.

System/ Solution	Wireless Technologies	Position Algorithm	Accuracy	Complexity	Robustness	Cost
Microsoft RADAR [36,37]	WLAN	KNN Viterbi-like Algorithm	3~5 m	Moderate	Good	Low
Horus [38,39]	WLAN	Probabilistic Method	2 m	Moderate	Good	Low
DIT [40,41]	WLAN	MLP,SVM	3 m	Moderate	Good	Low
Ekahau	WLAN	Probabilistic Method	1 m	Moderate	Good	Low

Tabella 2: Confronto tra i sistemi di localizzazione esistenti. [35]

Analizzando i dati riportati in tabella , tratti dal *Survey* sui sistemi di localizzazione [35], è possibile osservare che la differenza del livello di accuratezza tra i metodi probabilistici e le SVM è piccola, mentre la differenza tra i metodi probabilistici e il KNN inizia a essere considerevole. Tale fatto viene messo in evidenza anche in [12].

I sistemi sono stati confrontati valutando i test e i risultati disponibili sui documenti tecnici dei principali sistemi di localizzazione implementati ad oggi implementati [12], [40], [42].

Nonostante che le prestazioni, in termini di accuratezza, dei sistemi localizzazioni implementati con gli algoritmi Support Vector Machine o Multidimensional Vector Regression siano dell'ordine dei tre metri [40], [42](all'incirca pari al metodo Bayesiano), valutando i test e i risultati riportati negli articoli scientifici analizzati si è deciso di implementare un algoritmo di stima Bayesiano, pensando che fosse il miglior compromesso tra facilità di realizzazione, costo computazionale e accuratezza dell'algoritmo.

Per comprendere meglio l'implementazione del sistema di localizzazione nel capitolo seguente sarà illustrato il protocollo ZigBee ponendo particolare attenzione alla parte dedicata allo sviluppo di sistemi di localizzazione ZigBee.

## Parte II

# ZIGBEE

L'obiettivo del presente capitolo è quello di illustrare la tecnologia ZigBee selezionata per l'implementazione del sistema di localizzazione.

## 4 Il Protocollo

### 4.1 Introduzione

ZigBee è il nome di una specifica per un insieme di protocolli di comunicazione ad alto livello basati sullo standard di comunicazione IEEE 802.15.4.

Per definizione dei suoi stessi ideatori, lo standard ZigBee viene presentato come *“uno standard di comunicazione wireless bidirezionale a bassissimo costo e a bassissimi consumi”* [16], sviluppato con lo scopo di essere utilizzato *“nell’elettronica di consumo, nell’automazione di abitazioni ed edifici, nel controllo industriale, nelle applicazioni mediche e nei giocattoli”* [16].

Gli standard e le specifiche ZigBee vengono redatte dai membri di un’organizzazione no-profit denominata ZigBee Alliance. In Fig.15 sono mostrate alcune tra le aziende più importanti che fanno parte di ZigBee Alliance.



Figura 15: Le aziende principali che fanno parte di ZigBee Alliance.[19]

## 4.2 Caratteristiche ZigBee

Gli obiettivi di ZigBee Alliance si riassumono nel cercare di diffondere in larga scala l'uso delle reti ZigBee nello sviluppo di sistemi di controllo e automazione [17]. Le caratteristiche principali che si evidenziano in una rete ZigBee sono infatti:

- *Basso costo*
- *Affidabilità e possibilità di Auto-Riconfigurazione*
- *Flessibilità*
- *Facilità di implementazione*
- *Bassi consumi di potenza*
- *Sicurezza*

Tali caratteristiche soddisfano a pieno le esigenze richieste ad una rete WSN dedicata allo sviluppo di sistemi di controllo e monitoraggio facili da implementare, sicuri e a basso costo.

### 4.2.1 Dati Tecnici

Le principali caratteristiche tecniche dei nodi ZigBee, illustrate nel documento [17], sono le seguenti:

- *Banda di trasmissione:* Le schede ZigBee trasmettono informazione nelle bande ISM (*Industrial, Scientific and Medical*) 868.3 MHz (USA), 904÷924 MHz (EU) e 2.400÷2.480 GHz. La banda ISM è il nome assegnato dall'Unione Internazionale delle Telecomunicazioni (ITU) ad un insieme di porzioni dello spettro elettromagnetico riservate alle applicazioni radio non commerciali, per uso industriale, scientifico e medico. In commercio sono più diffuse le schede ZigBee che trasmettono nella banda 2.405÷2.480 GHz perché tale frequenza è valida a livello mondiale. I sistemi sviluppati con tali schede possono essere dunque commercializzati in tutto il mondo. Nel sistema implementato sono state utilizzate nodi ZigBee che trasmettono nell'intorno di 2.4 GHz.

- *Canali*: Nell'intorno di 2.4 GHz sono presenti 16 canali, ogni canale ha una banda di 5 MHz. Nell'intorno di 915 MHz sono presenti 10 canali ciascuno di 2 MHz mentre a 868.3 MHz è presente un unico canale.
- *Data rate*: Il data rate massimo di una scheda ZigBee è 250 Kb/s a 2.4 GHz, 40 Kb/s nell'intorno di 915 MHz e 20 Kb/s a 868.3 MHz .
- *Tipologia della rete*: Il protocollo ZigBee permette di costituire reti del tipo stella, albero o mesh. Le reti mesh hanno una struttura magliata in cui un nodo può comunicare con tutti i nodi adiacenti.

Esposte le principali caratteristiche del protocollo ZigBee, risulta utile effettuare un confronto tra i principali protocolli wireless presenti in commercio. In tabella 3 sono illustrate le principali caratteristiche tecniche dei protocolli di trasmissione IEEE 802.11.b (utilizzato nelle reti Wi-Fi), Bluetooth e ZigBee.

Caratteristica	802.11.b	Bluetooth	ZigBee
<b>Applicazioni Tipiche</b>	Reti di calcolatori	Trasmissioni audio	Reti di sensori
<b>Data Rate</b>	11 Mb/s	1 Mb/s	250 Kb/s
<b>Copertura outdoor</b>	100 m	10 m	70 – 300 m
<b>Max num Nodi</b>		32	7>65000
<b>Corrente Assorbita In Trasmissione</b>	430 mA	40 mA	35 mA
<b>Corrente Assorbita In Standby</b>	20 mA	200 µA	3 µA - 200 µA
<b>Durata Batterie</b>	Ore	Giorni	Anni
<b>Complessità Protocollo</b>	Molto Complesso	Complesso	Semplice
<b>Sicurezza</b>	Autenticazione SSID	Crittografia a 64, 128 bit	Crittografia AES A 128 bit

Tabella 3: Confronto tra le tecnologie wireless a banda stretta.

Osservando i dati in tabella è possibile dedurre che il protocollo ZigBee è più adatto per implementare reti di sensori rispetto gli altri protocolli.

Per poter implementare un sistema di controllo e monitoraggio in un ambiente di dimensioni considerevoli, è possibile che siano necessari un numero di nodi sensore elevato. Le reti che si possono costituire con i protocolli IEEE 802.11.b e Bluetooth sono composte da un numero molto inferiore di nodi rispetto ZigBee. Tale aspetto limita l'utilizzo di tali tecnologie in questo di applicazioni.

Altri aspetti importanti sono il data rate e il consumo dei nodi. I protocolli IEEE 802.11.b e Bluetooth sono caratterizzati da un data rate in trasmissione molto elevato rispetto ZigBee. In una rete di sensori non è necessario avere un data rate in trasmissione elevato. Si pensi ad esempio che un pacchetto ZigBee contenente un dato di temperatura ha la dimensione di 20 byte. Inoltre, se fosse necessario implementare un sistema a batteria, il sistema costituito dalla rete ZigBee avrebbe una durata molto superiore.

Tali aspetti evidenziano che tecnologie come Wi-Fi o Bluetooth sono adatte per altri tipi di applicazione come accesso a internet wireless, trasferimento di musica o filmati.

### 4.3 Stack del protocollo

La specifica ZigBee è una standard che definisce uno stack protocollare, ossia una pila di protocolli, che consente l'interoperabilità di dispositivi wireless (nodi ZigBee) con un basso costo, basso consumo energetico e basso data rate. Lo stack ZigBee è fondato sullo standard IEEE 802.15.4, il quale definisce le regole per la comunicazione tra nodi del livello Fisico e MAC della pila del protocollo. Le specifiche ZigBee rendono standard il livello di Rete, il livello per la Sicurezza e il livello Applicazione della pila.

In Fig.16 è mostrato lo stack protocollare ZigBee.

Ogni strato dello stack protocollare fornisce un insieme specifico di servizi necessari al corretto funzionamento dello strato successivo. Sarà fatta una breve descrizione di ogni strato che compone lo stack ZigBee [17]:

- *Physical Layer (PHY)*: essendo lo strato più basso, quindi più vicino alla trasmissione fisica dei segnali, è responsabile di una serie di funzioni tra cui l'attivazione e disattivazione della Radio per la ritrasmissione, dello *spreading* del segnale trasmesso, del calcolo dell'LQI (*Link Quality Indicator*) dei pacchetti ricevuti, dell'ED (*Energy Detect*) e del CCA (*Clear Channel Assessment*) parametri che verranno descritti nel capitolo successivo.

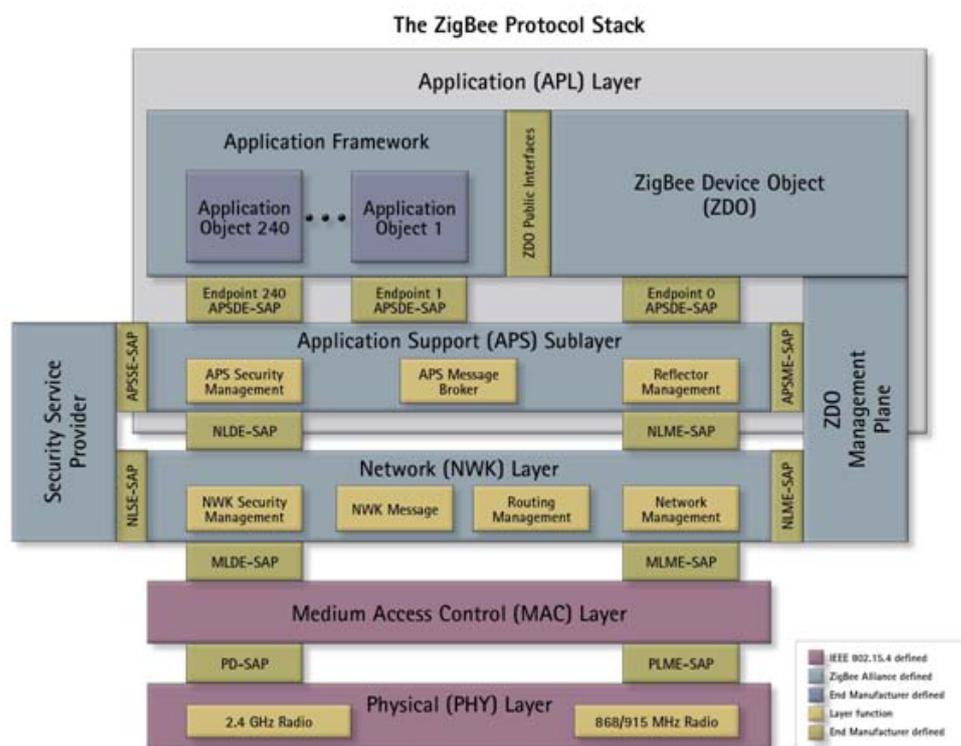


Figura 16: I livelli dello standard ZigBee. [17]

- *Medium Access Control Layer (MAC)*: è lo strato responsabile di fornire una comunicazione affidabile tra un nodo e il suo immediato vicino, evitare le collisioni tra pacchetti con l'algoritmo CSMA/CA e migliorare l'efficienza di comunicazione. Il livello MAC è anche responsabile della segmentazione dei pacchetti e della verifica dell'integrità dei pacchetti ricevuti (CRC).
- *Network Layer*: I compiti assegnati a questo livello comprendono la creazione della rete, l'assegnazione degli indirizzi, la gestione del routing dei pacchetti all'interno della rete in base all'indirizzo a 16 bit di un nodo detto *NWK address*, l'applicazione della sicurezza nella comunicazione e l'implementazione della route *discovery*.
- *Security Service Provider*: Questo protocollo fornisce dei meccanismi di sicurezza per i livelli che usano la crittografia (NWK e APS) [16]. In particolare sono previste molteplici soluzioni per implementare un grado di sicurezza di rete variabile a seconda

del particolare tipo di rete. Per la crittografia è utilizzato l'AES-128 (*Advanced Encryption Standard*) un algoritmo di cifratura a blocchi.

- *Application Layer (APL)*: È lo strato a livello più alto nella pila protocollare. Al livello applicazione sono definite le regole standard per implementare le applicazioni illustrate nella sezione 1.1 in Fig.(1) a cui sono dedicate le reti ZigBee.

#### 4.4 Livello Applicazione

L'obiettivo di ZigBee Alliance è quello di rendere standard la comunicazione ZigBee; l'organizzazione sta infatti tentando di definire delle regole di comunicazione per ogni applicazione, in modo che anche nodi ZigBee costruiti da aziende diverse possano interagire tra loro se utilizzati per implementare il medesimo sistema.

Per raggiungere tale obiettivo ZigBee Alliance definisce quindi come formare i pacchetti da trasmettere al livello applicazione, come identificare i vari campi del pacchetto e come identificare un scheda che appartiene ad una certa applicazione. In linea generale i campi specificati sono i seguenti:

- *Application Profile*
- *Device*
- *End Point*
- *Cluster*
- *Attributi*
- *Comandi*

##### 4.4.1 Application Profile

L'Application Profile definisce per quale applicazione è destinata la rete. Ogni profilo definito da ZigBee Alliance è detto profilo pubblico ed è identificato con campo a 16-bit detto *Profile ID*. I profili pubblici servono a garantire l'interoperabilità tra schede di aziende diverse destinate al solito ambiente di applicazione. I profili pubblici attualmente disponibili sono mostrati in Fig.17.



Figura 17: I profili applicazione pubblici attualmente previsti da ZigBee Alliance. [19]

All'interno di ogni profilo sono definite le regole per implementare delle applicazioni destinate all'ambiente definito dal profilo. In figura è possibile osservare che ZigBee Alliance ha reso standard l'utilizzo per esempio delle schede ZigBee in sistemi dedicati al profilo *Home Automation*. All'interno di tale profilo sono redatte e rese standard le regole per implementare sistemi di controllo destinati alle abitazioni.

Rimane comunque la possibilità di definire profili privati con opportuni numeri identificativi, anche se ZigBee Alliance consiglia di adottare questa soluzione solo nel caso in cui le funzionalità di interesse non siano già definite nei profili pubblici.

#### 4.4.2 Cluster

All'interno di ogni singolo profilo applicazione esistono più funzioni che un dispositivo può svolgere [29]. Per riportare un esempio semplice all'interno del profilo Home Automation un dispositivo può svolgere più funzioni (applicazioni), come controllare l'illuminazione dell'ambiente oppure la temperatura.

Un cluster quindi è la singola collezione di dati (*Cluster attribute*) e di messaggi (*Cluster commands*) relativi ad una specifica funzionalità di un profilo applicazione. Nelle specifiche dei cluster di un'applicazione sono riportati i formati e gli identificativi dei messaggi da associare ai pacchetti di una specifica funzione. I cluster sono identificati da un identificativo a 16 bit il *Cluster ID*.

#### 4.4.3 Attributi

Un attributo di un cluster è un'entità dati che rappresenta una grandezza fisica, uno stato, un dato da leggere o scrivere [29]. Gli attributi vengono comunicati ad altri dispositivi utilizzando i comandi con gli attributi inclusi. Anche gli attributi vengono identificati con campo a 16 bit (*Attribute ID*).

#### 4.4.4 Device ID

Nodi con diverse finalità possono far parte dello stesso cluster di una data applicazione. Per riconoscere il tipo di dispositivo viene utilizzato un campo a 16 bit detto *Device ID*.

#### 4.4.5 EndPoint

In una scheda ZigBee è possibile implementare fino a 240 applicazioni ognuna raggiungibile da un altro nodo tramite una “porta” detta *endpoint* [29]. Ogni endpoint è infatti individuato dall'*EndPoint ID* un numero a 8 bit. In Tabella4 sono illustrati i valori che può assumere tale identificativo.

Endpoint	Destinazione
0x00	ZDO
0x01 – 0xF0	Endpoint a disposizione Dell'utente
0xF1 – 0xFE	Riservati
0xFF	Rappresenta tutti gli Endpoint

Tabella 4: I valori che può assumere l'identificativo di un EndPoint. [29]

Resi standard i parametri descritti in questa sezione, un qualunque nodo ZigBee che riceve una pacchetto trasmesso secondo gli standard è in grado di capire a quale applicazione è destinato tale messaggio, se necessario è in grado di ricevere l'informazione contenuta in esso inoltrandola al corretto endpoint, elaborarla ed eventualmente ritrasmetterla ad un altro nodo ZigBee.

Ulteriori chiarimenti verranno nella sezione 5 in cui saranno descritti l'applicazione, il cluster e gli attributi definiti da ZigBee Alliance necessari a sviluppare un sistema di localizzazione conforme agli standard ZigBee.

#### 4.5 La Rete ZigBee

Ogni rete ZigBee è individuata da un indirizzo a 16 bit detto *PAN ID*. Ogni scheda ZigBee ha un indirizzo univoco MAC a 64 bit detto *MAC Address* dalla quale, quando si collega ad una rete, gli viene associato un indirizzo di rete a 16 bit *NWK Address*. I nodi che costituiscono una rete sono detti nodi e possono essere di tre tipi:

- *ZigBee Coordinator (ZC)*: Questo è il dispositivo responsabile dell'inizializzazione e del controllo della rete. È il dispositivo che decide la PAN ID della rete e assegna i NWK Address ai nodi della stessa. Questo dispositivo ha inoltre la possibilità di operare da ponte tra più reti. Durante la vita della rete immagazzina continuamente informazioni sulla rete e può agire come deposito di chiavi di sicurezza. Da notare che all'interno di una rete può esistere solo un coordinatore.
- *ZigBee Router (ZR)*: Questi dispositivi servono ad estendere la dimensione della rete ZigBee. Non hanno le stesse capacità del coordinatore ma comunque si occupano dello smistamento del traffico all'interno della rete. Inoltre, possono auto configurare il routing in caso di necessità.
- *ZigBee End Device (ZED)*: È il dispositivo meno complesso. Tale nodo è in grado di collegarsi ad una rete già formata, ed eventualmente estesa, da dispositivi delle categorie precedenti. Tale dispositivo può trasmettere e ricevere messaggi, ma non può eseguire nessuna operazione di routing. Per questo motivo un End Device deve essere necessariamente collegato ad un Router o a un Coordinatore. L'End Device è caratterizzato dal fatto che ha la parte radio che si disattiva (modalità *sleep*) a intervalli di tempo regolari. L'obiettivo di tale nodo è infatti quello di rilevare il dato da trasmettere, trasmetterlo e spegnersi per poi riattivarsi in un momento seguente. La parte radio viene disattivata per risparmiare al massimo il consumo energetico del nodo.

## 5 Sistema di Localizzazione ZigBee

### 5.1 Introduzione

ZigBee Alliance ha implementato lo standard per creare un sistema di localizzazione all'intero del profilo Telecom Application. Le direttive per formare un sistema di localizzazione sono fornite dalla compagnia in due documenti [30], [31].

Il primo documento redatto fornisce tutte le specifiche per creare un sistema di localizzazione de-centralizzato, ovvero un sistema in cui la stima della posizione del nodo viene eseguita direttamente sul nodo stesso.

Nel secondo documento redatto sono state estese le specifiche ZigBee introdotte in [30] per poter implementare un sistema di localizzazione centralizzato.

Come mostrato in Fig.(18), in un sistema di localizzazione centralizzato i dati vengono raccolti dal Gateway per poi essere trasmessi ad un server per essere utilizzati dal motore di localizzazione per stimare la posizione del nodo. In questo caso i nodi della rete servono solamente a raccogliere le informazioni necessarie per la localizzazione del nodo.

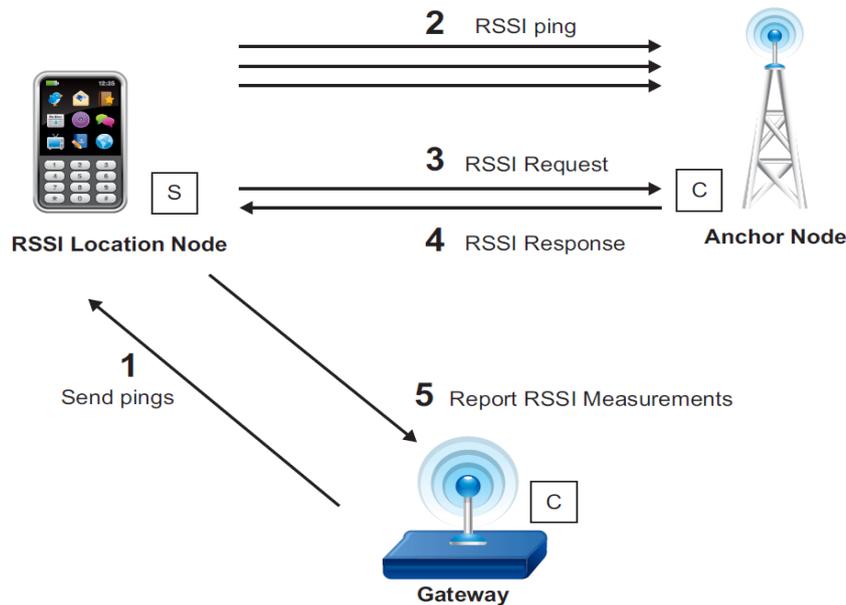


Figura 18: Sistema di comunicazione centralizzato previsto nel Telecom Application ai fini della localizzazione . [31]

Il sistema di comunicazione descritto in Fig.18 segue il seguente schema logico [31]:

1. Il Gateway invia un messaggio denominato *Send Ping* al Location Node per richiedere la posizione del nodo.
2. Il Location Node trasmette in *broadcast* con raggio = 1 (*time to live*) *n*-messaggi di *Rssi Ping*. Il Time to Live (TTL) rappresenta il numero di nodi della rete che un pacchetto deve attraversare.
3. Le Ancore che ricevono il messaggio di Ping misurano il valore di RSSI
4. Il Location Node trasmette un messaggio in broadcast di *RSSI Request* per ricedere il valore di RSSI misurato dalle Ancore
5. Le Ancore rispondo con un messaggio di risposta denominato *RSSI response*.
6. Il Location Node trasmette le misura al Gateway che inoltrerà le informazioni al server/motore di localizzazione.

## 5.2 Specifiche : Profilo , Cluster , Dispositivo

Prima di analizzare il formato dei messaggi ZigBee risulta essere utile specificare l'identificativo del profilo Telecom e l'identificativo del cluster RSSI Location:

- Profile ID = 0x0107 (Telecom Application)
- ClusterID = 0x000b (RSSI Location)
- Device ID = 0x500 (Anchor), Device ID = 0x501 (Location Node), Device ID = 0x502 (Gateway)

## 5.3 Specifiche : Payload Messaggi

In questa sezione analizzeremo le specifiche descritte in [30] e [31] illustrando il formato dei messaggi ZigBee, gli identificativi e analizzando il loro uso, ovvero analizzeremo il contenuto del cluster RSSI Location. La comunicazione tra due sensori di una rete ZigBee avviene tramite lo scambio di messaggi denominati ZCL Frame secondo il modello *client-server*.

Secondo la logica ZigBee [30], nella comunicazione tra due dispositivi, il server è l'entità che registra i dati mentre il client è l'entità che li richiede per poterli manipolare.

### 5.3.1 Server

#### ATTRIBUTI

In Tabella 5 viene mostrato l'insieme di attributi definiti in [30] e [31] e i rispettivi identificativi.

Attribute Set Identifier	Description
0x000	Location Information
0x001	Location Settings
0x002 – 0xffff	Reserved

Tabella 5: Location Attribute Sets. [30].

#### Location Information Attribute Set

Il set di attributi Location Information contiene gli attributi riportati in Tabella 6 .

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	LocationType	8-bit Data	0000xxxx	Read only	-	M
0x0001	LocationMethod	8-bit Enumeration	0x00 – 0xff	Read only	-	M
0x0002	LocationAge	Unsigned 16-bit Integer	0x0000 – 0xffff	Read only	-	O
0x0003	QualityMeasure	Unsigned 8-bit Integer	0x00 – 0x64	Read only	-	O
0x0004	NumberOfDevices	Unsigned 8-bit Integer	0x00 – 0xff	Read only	-	O

Tabella 6: Attributi del Location Information Attribute Set. [30].

**Location Type Attribute:** Il Location Type è un attributo di un byte. Il significato dei singoli bit che compongono questo attributo è mostrato in Tabella 7.

Bit Field (Bit Numbers)	Meaning	Values
0	Absolute	1 – Absolute location 0 – Measured location
1	2-D	1 – Two dimensional 0 – Three dimensional
2-3	Coordinate System	0 – Rectangular (installation-specific origin and orientation) 1-3 – Reserved
4-7	Reserved	-

Tabella 7: Location Type Attribute. [30].

I campi di questo attributo hanno il seguente significato:

- *Absolute*: se è posto a 1 significa che la posizione di un nodo è conosciuta, altrimenti vuol dire che la posizione è stata calcolata.
- *2-D* : Questo bit indica se la localizzazione è bidimensionale o tridimensionale. Nel caso in cui è posto a uno il valore della terza coordinata deve essere settato con il valore 0x8000.
- *Coordinate System*: Indica la geometria del sistema di riferimento utilizzato per esprimere le coordinate. Come è possibile osservare in tabella il valore di questo campo è sempre 0.

**Location Method Attribute :** specifica il metodo utilizzato dal sistema di localizzazione per stimare la posizione del nodo. I campi dell'attributo Location Method sono mostrati in Tabella 8. Il metodo *centralized* è stato aggiunto come estensione nel documento [31].

<b>Value</b>	<b>Method</b>	<b>Description</b>
0x00	Lateration	A method based on RSSI measurements from three or more sources.
0x01	Signposting	The location reported is the location of the neighboring device with the strongest received signal.
0x02	RF fingerprinting	RSSI signatures are collected into a database at commissioning time. The location reported is the location taken from the RSSI signature database that most closely matches the device's own RSSI signature.
0x03	Out of band	The location is obtained by accessing an out-of-band device (that is, the device providing the location is not part of the ZigBee network).
<b>0x04</b>	<b>Centralized</b>	<b>If the location is performed in a centralized way (e.g. by the GW). Different from the previous since here the device performing the localization is part of the ZigBee network.</b>
0x05 - 0x3f	-	Reserved
0x40 - 0xff	-	Reserved for manufacturer specific location methods.

Tabella 8: Location Method Attribute. [31]

**Location Age Attribute:** indica da quanto tempo è stata effettuata la stima della posizione.

**Quality of Measure Attribute:** è un indicatore della qualità della stima. La metrica con cui viene valutato questo è a descrizione di colui che implementa il sistema di localizzazione.

**Number Of Device Attribute:** indica il numero di ancore impiegate per effettuare l'ultima stima.

#### **Location Settings Attribute set**

Il set Location Settings contiene gli attributi illustrati in Tabella 9.

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	Coordinate1	Signed 16-bit integer	0x8000–0x7fff	Read / Write	-	M
0x0011	Coordinate2	Signed 16-bit integer	0x8000–0x7fff	Read / Write	-	M
0x0012	Coordinate3	Signed 16-bit integer	0x8000–0x7fff	Read / Write	-	O
0x0013	Power	Signed 16-bit integer	0x8000–0x7fff	Read / Write	-	M
0x0014	PathLossExponent	Unsigned 16-bit integer	0x0000–0xffff	Read / Write	-	M
0x0015	ReportingPeriod	Unsigned 16-bit integer	0x0000–0xffff	Read / Write	-	O
0x0016	CalculationPeriod	Unsigned 16-bit integer	0x0000–0xffff	Read / Write	-	O
0x0017	NumberRSSIMeasurements	Unsigned 8-bit integer	0x01–0xff	Read / Write	-	M

Tabella 9: Attributi Location Setting. [30]

**Coordinate 1,2,3 Attribute:** è rappresentato da un campo del tipo signed 16-bit integers e rappresenta il valore della coordinata stimata. Se il valore della coordinata non è noto, il campo coordinate deve essere posto con il valore 0x8000 [30].

**Power Attribute:** specifica il valore della potenza media  $P_0$ , espresso in dBm, ricevuta alla distanza di un metro dal trasmettitore. Anche in questo caso il valore 0x8000 indica che questo parametro non è noto.

**PathLoss Exponent Attribute:** indica il valore dell'esponente path-loss espresso nell'equazione (6). Il valore 0xFFFF indica che il valore di tale attributo non è noto.

**Reporting Period Attribute:** indica il periodo di tempo che intercorre tra due stime successive nel caso di *report* automatico. Se questo parametro è posto a zero significa che il Location Node non riporta automaticamente le informazioni necessarie alla stima.

**Number of RSSI Measurements Attribute:** indica il numero di RSSI misurati da una stessa antenna utilizzati per stimare la posizione del nodo.

## Comandi Generati

In Tabella 10 sono mostrati i comandi generati lato Server e di conseguenza ricevuti

la Client. Da notare che i comandi evidenziati sono stati introdotti nel documento successivo [31]. Il contenuto dei comandi trasmessi, dove non specificato, corrisponde ad una combinazione degli attributi sopra esposti.

<b>Command Identifier Field Value</b>	<b>Description</b>	<b>Mandatory/Optional</b>
0x00	Device configuration response	M
0x01	Location data response	M
0x02	Location data notification	M
0x03	Compact location data notification	M
0x04	RSSI Ping	M
<b>0x05</b>	<b>RSSI Request</b>	<b>O</b>
<b>0x06</b>	<b>Report RSSI Measurements</b>	<b>O</b>
<b>0x07</b>	<b>Request Own Location<sup>a</sup></b>	<b>O</b>
0x08-0xff	Reserved	

Tabella 10: Comandi lato server. [31]

**Device Configuration Response Command:** Tale comando viene trasmesso in risposta alla richiesta *Get Device Configuration* generata dal client. Il formato del pacchetto è mostrato in Fig.19.

<b>Octets</b>	1	0 / 2	0 / 2	0 / 2	0 / 1	0 / 2
<b>Data Type</b>	Enumeration	Signed Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer
<b>Field Name</b>	Status	Power	Path Loss Exponent	Calculation Period	Number RSSI Measurements	Reporting Period

Figura 19: Formato del pacchetto Device Configuration Response. [30]

Il campo STATUS indica se la risposta ha avuto successo o meno. Se il valore di questo campo corrisponde a NOT\_FOUND vuol dire che il nodo che ha risposto non è in grado di fornire i dati richiesti.

**Location Data Response Command:** Tale comando viene trasmesso in risposta alla richiesta di informazione sulla locazione del nodo e sui parametri del canale. Il formato del pacchetto è mostrato in Fig.20.

<b>Octets</b>	1	0 / 1	0 / 2	0 / 2	0 / 2	0 / 2	0 / 2	0 / 1	0 / 1	0 / 2
<b>Data Type</b>	Enumeration	Data	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer	Enumeration	Unsigned Integer	Unsigned Integer
<b>Field Name</b>	Status	Location Type	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent	Location Method	Quality Measure	Location Age

Figura 20: Formato del pacchetto Location Data Response. [30]

Se il bit dell'attributo Location Type corrispondente ad Absolute è posto a 1 i campi Location Method, Quality of Measure e Location Age non devono essere trasmessi [30].

**Location Data Notification Command:** viene trasmesso periodicamente da un nodo per annunciare le proprie informazioni di posizione e sul canale. Il periodo di trasmissione corrisponde al valore dell'attributo Reporting Period. La trasmissione di questo pacchetto può anche avvenire in modalità broadcast ma con TTL unitario per evitare congestione della rete. Il formato del pacchetto è mostrato in Fig.21.

Da notare che il formato del pacchetto è identico al comando precedente.

<b>Octets</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>0 / 2</b>	<b>2</b>	<b>2</b>	<b>0 / 1</b>	<b>0 / 1</b>	<b>0 / 2</b>
<b>Data Type</b>	Data	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer	Enumeration	Unsigned Integer	Unsigned Integer
<b>Field Name</b>	Location Type	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent	Location Method	Quality Measure	Location Age

Figura 21: Formato del pacchetto Location Data Notification. [30]

**Compact Location Data Notification Command:** è identico al comando precedente esclusi Power, Path Loss Exponent e il campo Location Method.

**RSSI Ping Command:** il comando viene trasmesso per far misurare il valore di RSSI alle ancore . La trasmissione di questo pacchetto può anche avvenire in modalità broadcast con TTL unitario. Il formato di tale pacchetto è mostrato in Fig.22.

<b>Octets</b>	1
<b>Data Type</b>	Integer
<b>Field Name</b>	Location Type

Figura 22: Formato del pacchetto RSSI Ping. [30]

**RSSI Request Command:** come illustrato in Fig.18 tale comando viene trasmesso dal Location Node per richiedere le misure di RSSI effettuate dalle Ancore. In [31] viene consigliato di trasmettere questo messaggio in modalità broadcast sempre con raggio (TTL) unitario. Tale comando non possiede payload.

**Report RSSI Measurements Command:** tale messaggio, contenente tutte le informazioni sulle misure, viene trasmesso dal Location Node al Gateway. Queste informazione vengono successivamente inoltrate al motore di localizzazione. Il formato del pacchetto è mostrato in Fig.23.

in cui:

<b>Octets</b>	<b>8</b>	<b>1</b>	<b>N</b>
Data Type	IEEE Address	Unsigned 8-bit integer	Variable
Field Name	Measuring Device	N Neighbors	Neighbors Info

Figura 23: Formato del pacchetto Report RSSI Measurements. [31]

- Measuring Device è l'indirizzo del Location Node.
- N-Neighbors è il numero di ancore che hanno effettuato la misura.
- Il campo Neighbors Info è illustrato nella figura seguente

<b>Octets:</b>	<b>8</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>1</b>
Data Type	IEEE Address	Signed16-bit integer	Signed16-bit integer	Signed16-bit integer	Signed8-bit integer	Unsigned8-bit integer
FieldName	Neighbour	X	Y	Z	RSSI	NumberRSSIMeasurements

Figura 24: Struttura Neighbors Info. [31]

Nel formato del pacchetto illustrato in figura, l'indirizzo rappresentato dal campo Neighbour è l'indirizzo dell'ancora che ha misurato il valore di RSSI.

**Request Own Location Command:** viene trasmesso al motore di localizzazione dal nodo che vuole conoscere la propria posizione. Il formato del pacchetto è mostrato in Fig.25.

### 5.3.2 Client

#### ATTRIBUTI

Il Client non ha attributi [30].

<b>Octets:</b>	<b>8</b>
DataType	IEEE Address
FieldName	IEEE address of the blind node

Figura 25: Formato del pacchetto Request Own Location. [31]

## Comandi Generati

In Tabella 11 sono mostrati i comandi generati lato Client. Anche in questo caso i comandi evidenziati sono stati introdotti nel documento [31]. Il contenuto dei comandi trasmessi, dove non specificato, corrisponde ad una combinazione degli attributi sopra esposti.

<b>Command Identifier Field Value</b>	<b>Description</b>	<b>Mandatory/Optional</b>
0x00	Set Absolute Location	M
0x01	Set device configuration	M
0x02	Get device configuration	M
0x03	Get location data	M
<b>0x04</b>	<b>RSSI Response</b>	<b>O</b>
<b>0x05</b>	<b>Send Pings<sup>a</sup></b>	<b>O</b>
<b>0x06</b>	<b>Anchor node announce</b>	<b>O</b>
0x07-0xff	Reserved	-

Tabella 11: Comandi lato Client. [31]

**Set Absolute Location Command:** è utilizzato per impostare le coordinate (note) di un nodo conosciuto. Il formato del pacchetto è illustrato in Fig.26.

<b>Octets</b>	2	2	2	2	2
<b>Data Type</b>	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer
<b>Field Name</b>	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent

Figura 26: Formato del pacchetto Device Configuration Response. [30]

**Set Device Configuration:** è utilizzato per impostare i parametri di un nodo (Pathloss, Calculation Period, etc). Il formato del pacchetto è mostrato in Fig.27.

<b>Octets</b>	2	2	2	1	2
<b>Data Type</b>	Signed Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer
<b>Field Name</b>	Power	Path Loss Exponent	Calculation Period	Number RSSI Measurements	Reporting Period

Figura 27: Formato del pacchetto Set Device Configuration. [30]

**Get Device Configuration:** tale comando è utilizzato per richiedere i parametri di localizzazione di un nodo. Il formato del pacchetto è mostrato in Fig.28.

<b>Octets</b>	1
<b>Data Type</b>	IEEE Address
<b>Field Name</b>	Target Address

Figura 28: Formato del pacchetto Set Device Configuration. [30]

Il campo Target Address contiene l'indirizzo a 64-bit del nodo. Un nodo alla ricezione di questo comando genera una Device Configuration Response.

**Get Location Data:** è utilizzato per richiedere la posizione di un nodo e i parametri di canale. Tale pacchetto può essere trasmesso in modalità unicast, multicast o broadcast [30]. Nel caso di trasmissione broadcast, per evitare congestione, il TTL deve essere unitario. Il formato del pacchetto è mostrato in Fig.29.

<b>Bits</b>	3	1	1	1	1	1	8	0 / 64
<b>Data Type</b>	8-bit Bitmap						Unsigned Integer	IEEE address
<b>Field Name</b>	Reserved	Compact Response	Broadcast Response	Broadcast Indicator	Re-calculate	Absolute Only	Number Responses	Target Address

Figura 29: Formato del pacchetto Get Location Data. [30]

Il campo *Absolute only* specifica che tipo di informazioni sono richieste, ovvero se le coordinate del nodo devono essere note a priori o calcolate. Quando un nodo riceve questo tipo di comando genera in risposta un comando di Location Data Response [30].

**RSSI Response:** viene trasmesso in risposta al comando RSSI Request [31]. Il formato del pacchetto è mostrato in Fig.30.

<b>Octets</b>	<b>8</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>1</b>
Data Type	IEEE address	Signed 16-bit integer	Signed 16-bit integer	Signed 16-bit integer	Signed 8-bit integer	Unsigned 8-bit integer
FieldName	Replying Device	X	Y	Z	RSSI	Number RSSI Measurements

Figura 30: Formato del pacchetto RSSI Response. [31]

dove:

- Replaying Device è l'indirizzo a 64 bit del nodo che ha effettuato la misura.
- X,Y,Z sono le coordinate (se note) dell'ancora che ha fatto la misura.
- RSSI è il valore di RSSI misurato.

**Send Ping:** nel momento in cui un nodo riceve questo tipo di messaggio inizia a trasmettere i pacchetti per far misurare alle Ancore del sistema i valori di RSSI [31]. Il formato del pacchetto è mostrato in Fig.31.

Octets	8	1	2 <sup>a</sup>
Data Type	IEEE Address	Unsigned 8-bit integer	Unsigned 16-bit integer
Field Name	Target Address	NumberRSSIMeasurements	CalculationPeriod

Figura 31: Formato del pacchetto Send Ping. [31]

**Anchor Node Announce:** Un'ancora trasmette tale comando nel momento in cui si registra alla rete ZigBee del sistema di localizzazione. [31]. In questo messaggio sono contenute le informazioni del nodo stesso. Il formato del pacchetto è mostrato in Fig.32.

Octets	8	2	2	2
Data Type	IEEE Address	Signed 16-bit integer	Signed 16-bit integer	Signed 16-bit integer
Field Name	Anchor node IEEE address	X	Y	Z

Figura 32: Formato del pacchetto Anchor Node Announce. [31]

## 6 BeeStack : Trasmissione e Ricezione dei pacchetti ZigBee

### 6.1 Software Freescale

Freescale fornisce all'utente un insieme di software volti alla programmazione, gestione e test delle reti ZigBee. In particolare saranno descritti due software che si sono rivelati utili per lo sviluppo del sistema di localizzazione:

- *Beekit*: Beekit è un programma che, attraverso una procedura guidata, permette di creare un software denominato BeeStack implementa il protocollo ZigBee sui chip MC1322x. Durante la procedura guidata è possibile selezionare l'applicazione a cui è dedicato il dispositivo tra alcune applicazioni già implementate da Freescale come ad esempio Home Automation, è possibile selezionare il tipo di sensore (Sensor Node, etc...) e il tipo di dispositivo da implementare (Coordinatore, etc...). Attraverso tale procedura è inoltre possibile selezionare il tipo di sicurezza nella comunicazione, la PAN ID a cui aggregarsi e i canali da utilizzare. Le ulteriori possibilità offerte da tale programma sono descritte in modo esauriente in [25]. Dopo aver creato il BeeStack con la procedura guidata sopra descritta, il firmware viene esportato e visualizzato tramite l'utilizzo di un programma di compilazione come IAR o CODEWARRIOR. Tali programmi permettono di visualizzare il BeeStack nel linguaggio di programmazione C. Visualizzando il codice nel linguaggio di programmazione C, risulta possibile applicare delle modifiche al firmware per modificare le applicazioni implementate con la procedura guidata oppure per creare delle applicazioni personali che non si atten- gono agli standard ZigBee come descritto in [26] (*Custom Application*). Il programma di compilazione offre inoltre la possibilità di caricare il codice sulle schede ZigBee attraverso l'interfaccia JTAG e eseguire il debug del codice stesso. È importante os- servare che il software caricato sulle schede ZigBee è da considerarsi quasi un sistema operativo; una parte del software BeeStack infatti è dedicato al multitasking attra- verso la gestione di uno scheduler denominato *Event Scheduler*. Questo permette di far eseguire alla scheda più compiti (*task*) "contemporaneamente".
- *Test Tool*: È un software di Freescale che permette di caricare il BeeStack nel formato *.bin* sulla scheda dalla porta USB. Test Tool permette inoltre di fare altre operazio- ni tra le quali quella di visualizzare i pacchetti ricevuti e trasmessi da una scheda connessa al PC tramite porta USB. Questo programma fornisce quindi un'interfaccia grafica per il modulo ZTC (*ZigBee Test Client*) di Freescale [27]; il modulo che gesti- sce la trasmissione dei pacchetti ZigBee su seriale aggiungendo un opportuno header

come mostrato in Fig.33. Il campo *Opgroup* e *Opcode* identificano lo specifico comando mentre *length* indica la lunghezza del pacchetto escluso l'header. Nel campo *Data* invece è contenuto il pacchetto ZigBee ricevuto. Nella trasmissione al PC viene aggiunto, all'inizio del pacchetto, un campo *Sync* che indica l'inizio del pacchetto (valore = 0x02) e in coda al pacchetto aggiungono il *CRC* ovvero un campo per la checksum. Ulteriori dettagli tecnici di Test Tool e ZTC sono disponibili in [27], [28].

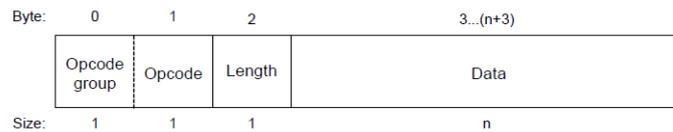


Figura 33: Il formato del pacchetto ZTC. [28]

## 6.2 Trasmissione e ricezione dei pacchetti ZigBee

In questa fase del presente elaborato si rende opportuno fare un'analisi a livello applicazione delle funzioni, all'interno del BeeStack, che permettono di trasmettere e ricevere i comandi e gli attributi descritti nelle specifiche ZigBee.

Il firmware BeeStack, generato con la procedura guidata descritta in precedenza, si presenta in IAR come un insieme di file e cartelle implementati secondo le specifiche ZigBee Fig.34. All'interno di questi file sono definite le variabili e le funzioni necessarie al corretto funzionamento del dispositivo. Il firmware è implementato secondo lo stack protocollare ZigBee.

Il file BeeApp.c descrive il funzionamento del nodo ZigBee al livello applicativo; al suo interno sono implementate funzioni e chiamate di funzioni dedicate a l'inizializzazione del dispositivo, trasmissione e ricezione di pacchetti. Le primitive che eseguono queste operazioni sono denominate:

- ***void BeeAppInit ()***: Questa funzione serve a inizializzare il dispositivo. Tra le operazioni che esegue inizializza le variabili, la radio e i led.

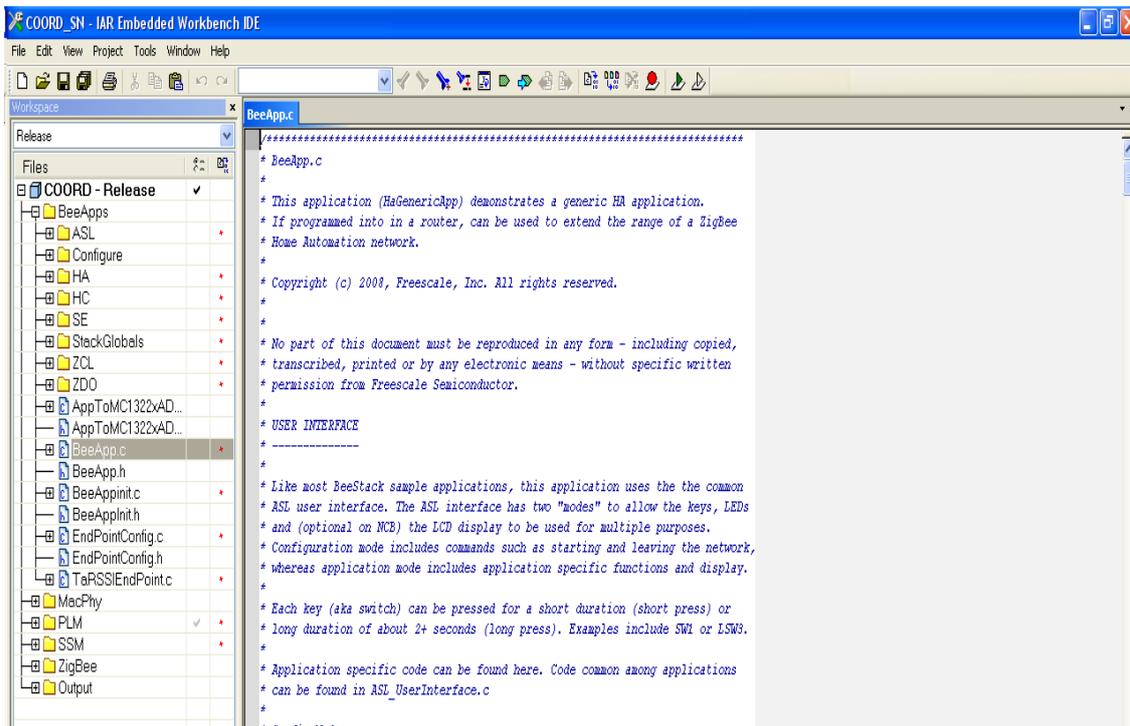


Figura 34: Come si presenta il BeeStack visualizzato in IAR.

- ***void BeeAppDataRequest ()***: La funzione BeeAppDataRequest si occupa della trasmissione dei messaggi ZigBee [29]. La primitiva, per trasmettere un pacchetto, necessita delle informazioni contenute nella seguente struttura :

```
typedef struct zbApsdeDataReq_tag {
    zbAddrMode_t    dstAddrMode;    /* indirect, group, direct-16, direct-64 */
    zbApsAddr_t     dstAddr;        /* short address, long address or group (ignored on indirect mode) */
    zbEndPoint_t    dstEndPoint;    /* destination endpoint (ignored if group mode) */
    zbProfileId_t   aProfileId;     /* application profile (either private or public) */
    zbClusterId_t   aClusterId;     /* cluster identifier */
    zbEndPoint_t    srcEndPoint;    /* source endpoint */
    uint8_t         asduLength;     /* length of payload (of this fragment if fragmented) */
    uint8_t         *pAsdu;        /* pointer to payload */
    zbApsTxOption_t txOptions;     /* options on transmit (to ACK or not to ACK, that is the question... */
    uint8_t         radiusCounter; /* radius */
    zbApsConfirmId_t confirmId;    /* identifier for the confirm */
    zbTxFragmentedHdr_t fragmentHdr; /* set to all 0x00s if not fragmented transaction */
} zbApsdeDataReq_t;
```

I campi della struttura mostrata sono illustrati nella tabella seguente:

- ***void BeeAppDataIndication ()*** : Questa funzione a livello applicazione si occupa

Campo	Tipo di dato	Formato	Descrizione
dstAddrMode	zbAddrMode	uint8	Tipo di indirizzamento
dstAddr	zbApsAddr	union	Indirizzo del nodo destinatario
dstEndPoint	zbEndPoint	uint8	Endpoint del destinatario
ProfileId	zbProfileId	uint16	Identificativo del Profilo
ClusterId	ZbClusterId	uint16	Identificativo del Cluster
srcEndpoint	zbEndPoint	uint8	EndPoint del Mittente
asduLenght		uint8	Dimensione del comando ZigBee
*pAsdu		uint8	Puntatore al payload (vedi sez. 4)
txoption	zbApsTxoption	uint8	Indica se è presente la trasmissione con Ack
RadiusCounter		uint8	È il tilme to live del pacchetto
confirmId	zbApsConfirmId	uint8	Indicatore del messaggio di conferma
zbRxFragmentedHdr	fragmentedHdr	struttura	Intestazione per la frammentazione

Tabella 12: Parametri Apsde-DataRequest .

della ricezione dei messaggi [29]. Al livello applicativo, ogni pacchetto ricevuto ZigBee ha la seguente struttura:

```

typedef struct zbApsdeDataIndication_tag
{
    zbAddrMode_t          dstAddrMode;    /* address mode */
    zbNwkAddr_t           aDstAddr;       /* destination address or group */
    zbEndPoint_t         dstEndPoint;     /* destination endpoint */
    zbAddrMode_t         srcAddrMode;     /* address mode - note: always 16-bit */
    zbNwkAddr_t          aSrcAddr;        /* source address or group (never 64-bit) */
    zbEndPoint_t         srcEndPoint;     /* source endpoint */
    zbProfileId_t        aProfileId;     /* profile ID (filtered automatically by APS) */
    zbClusterId_t        aClusterId;     /* cluster ID (no filter applied to clusters) */
    uint8_t              asduLength;     /* length of payload (or this fragment if fragmented) */
    uint8_t              *pAsdu;         /* pointer to payload */
    zbStatus_t           status;         /* status field of the incoming frame processing */
    bool_t               fWasBroadcast;   /* was the packet groupcast or broadcast? */
    zbApsSecurityStatus_t fSecurityStatus; /* was the packet secure? At which layers? */
    uint8_t              linkQuality;     /* link quality from network layer */
    zbTime_t             rxTime;         /* vendor specific time packet was received (always 0) */
    zbRxFragmentedHdr_t  fragmentHdr;    /* only present if txOptions & gApsTxOptionFragmentationRequested_c */
} zbApsdeDataIndication_t;

```

I campi della struttura sono illustrati nella tabella seguente:

Occorre notare che le informazioni dei campi *status*, *wasBroadcast*, *fSecurity*, *linkQuality* e *rxTime* non sono contenute direttamente nel pacchetto trasmesso (vedi tabella 12). Tali informazioni vengono fornite al livello applicazione dal nodo che riceve il messaggio.

- ***void BeeAppDataConfirm ()***: Indica al nodo sorgente se il pacchetto trasmesso è stato ricevuto correttamente dal nodo destinatario.

Campo	Tipo di dato	Formato	Descrizione
dstAddrMode	zbAddrMode	uint8	Tipo di indirizzamento
dstAddr	zbApsAddr	union	Indirizzo del nodo destinatario
dstEndPoint	zbEndPoint	uint8	Identificativo dell'Endpoint a Cui è destinato il pacchetto
srcAddrMode	zbAddrMode	uint8	Tipo di indirizzamento
srcAddr	zbApsAddr	union	Indirizzo del nodo sorgente
srcEndPoint	zbEndPoint	uint8	Identificativo dell'Endpoint a Cui è destinato il pacchetto
ProfileId	zbProfileId	uint16	Identificativo del Profilo
ClusterId	ZbClusterId	uint16	Identificativo del Cluster
asduLenght		uint8	Dimensione del comando ZigBee
*pAsdu		uint8	Puntatore al payload (vedi sez. 4)
status	zbStatus	uint8	Indica se il messaggio è stato Ricevuto correttamente
WasBroadcast		bool	Indica se il messaggio è stato Trasmesso in modalità broadcast
fSecurityStatus	zbApsSecurityStatus		Indica con che livello di sicurezza è Stato trasmesso il messaggio ricevuto
linkQuality		int8	È un indicatore della qualità Del canale vedi sez. 6
rxTime	ZbTime	uint16	Indica in che istante è stato ricevuto un pacchetto
zbRxFragmentedHdr	fragmentedHdr	struttura	Intestazione del pacchetto se frammentato

Tabella 13: Parametri Apsde-Dataindication.

In particolare il contenuto delle funzioni BeeAppDataIndication e BeeAppDataRequest dipende dal tipo di applicazione implementata.

## 7 Schede Utilizzate nel Progetto

### 7.1 Tipi di Schede

Nel progetto sono state utilizzate delle schede di sviluppo fornite da Freescale una delle più importanti aziende membra di ZigBee Alliance e delle schede realizzate dal Dipartimento di Ingegneria dell'Informazione di Pisa impiegando comunque i chip forniti da . Le schede fornite da Freescale sono di tre tipologie:

- **Network Node**
- **Sensor Node**
- **LowPower Node**

Tali schede hanno tutte installato il chip MC 13224 delle Freescale. Il chip MC13224 viene presentato da Freescale come un dispositivo compatibile *Platform-in-Package* per lo standard 2.4 GHz IEEE 802.15.4 [24]. Nel chip è presente un processore ARM-7TDMI a 32 bit, ha una memoria RAM di (96 KB), una ROM (80KB) e una flash da (128KB). Ulteriori dettagli tecnici sono disponibili consultando [http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=MC13224V](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13224V) oppure nel manuale [24].

Le schede utilizzate si differenziano tra loro per dimensione, quantità e tipo di sensori integrati. Dato che per lo sviluppo del sistema di localizzazione non sono stati utilizzati i sensori di tali schede (perché non è un sensore che misura il valore di RSSI ma il chip), in questa sezione non verrà effettuata una descrizione dettagliata di tali dispositivi, ma verrà unicamente illustrata in Fig.35 la scheda Sensor Node per fornire un'idea di come tali dispositivi si presentano all'utilizzatore.

Come è possibile osservare in figura, la scheda ha quattro led corrispondenti a quattro switch e un pulsante di reset . Nella scheda sono presenti un'antenna integrata, un connettore USB, un connettore JTAG per programmare la scheda ed eseguire il debug e un ingresso audio. Come sensori sono presenti un sensore accelerometrico triassiale XYZ, un sensore di temperatura e un sensore di pressione. La scheda offre le possibilità di essere alimentata da corrente elettrica, con due batterie stilo (3 V) o tramite connessione USB.

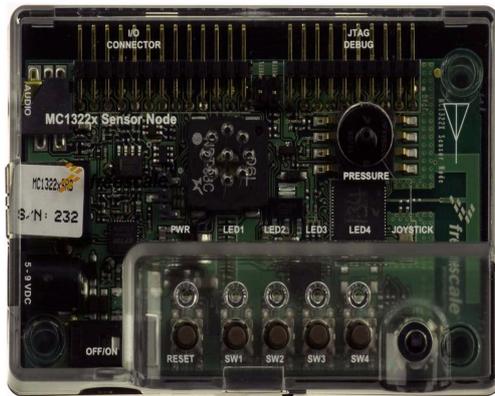


Figura 35: La scheda Freescale Sensor Node. [21]

Per maggiori specifiche sulle schede citate è possibile fare riferimento ai documenti [20], [21], [22].

## 7.2 MC13224: Misura dell'RSSI

Il significato fisico associato al nome RSSI (*Receive Signal Strength Indicator*) è stato esaurientemente illustrato nella sezione 2.1.1. Prima di poter analizzare il sistema implementato si rende però necessario chiarire come tale quantità viene misurata da un pacchetto ZigBee ricevuto. Occorre da notare che per ogni tipo di casa costruttrice e per ogni tipo di chip ZigBee la misura di tale parametro può essere definita in modo differente. Dato che nel sistema sviluppato sono state utilizzate schede ZigBee con il chip MC 13224 della Freescale, si rende opportuno illustrare è possibile ottenere il valore di RSSI impiegando tale chip.

Per i chip Freescale MC13224 il valore dell'RSSI è direttamente legato al valore del parametro *link quality indicator* (LQI) come dichiarato dalla stessa azienda nel documento [33]. Prima di mostrare la relazione che lega i due parametri è opportuno chiarire cosa rappresenta fisicamente la quantità LQI e come tale parametro viene misurato.

Ad esempio per il chip MC1323x il parametro LQI viene definito in [34] come la misura dell'energia media del segnale in banda base durante la ricezione del pacchetto. La misura viene effettuata al livello fisico in un tempo pari a  $128 \mu\text{s}$  Fig.36.

Il tempo pari a  $128 \mu\text{s}$  corrisponde al tempo necessario alla ricezione di 8 simboli come redatto nel documento [34]. Occorre notare che la misura dell'LQI viene effettuata solo su

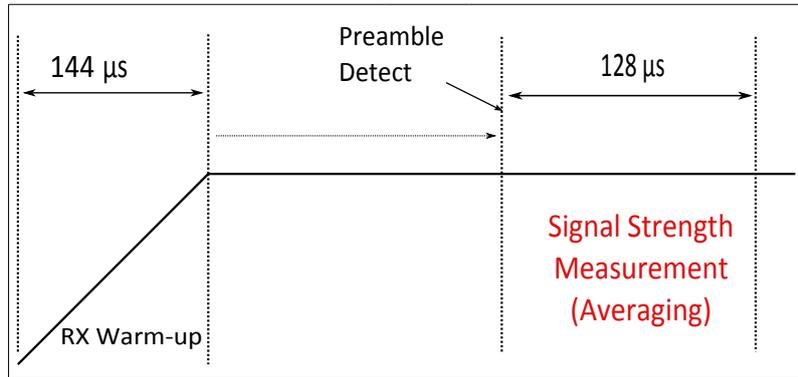


Figura 36: Intervallo di tempo in cui viene misurato il valore dell'LQI. [34]

pacchetti ZigBee. La misura viene infatti eseguita una volta ricevuto il preambolo come mostrato in Fig.36. Il valore del parametro LQI viene normalmente utilizzato per dare ai livelli superiori un'informazione sulla qualità del pacchetto ricevuto.

Per quanto riguarda i chip della famiglia MC1322x, la relazione che lega il parametro LQI al parametro RSSI, viene direttamente fornita nel documento [24]. Nel manuale del chip è redatto che al parametro LQI deve essere associata una variabile di tipo intero 8-bit senza segno come illustrato in tabella 13 sez. 5.2 .

Il valore dell'LQI viene registrato nella variabile definita come un numero esadecimale nel range 0x00÷0xFF dove:

- Il valore 0x00 corrisponde ad un valore di potenza pari a -100 dBm.
- Il valore 0xFF corrisponde ad un valore di potenza pari a 0 dBm.

Definito il parametro LQI, l'equazione che esprime la relazione tra RSSI e LQI è:

$$\text{InputPower(dBm)} = (\text{LQI}_{dec}/3) - 100 \quad (34)$$

in cui:

- **InputPower** è il valore di RSSI ricevuto in dBm.
- **LQI<sub>dec</sub>** è il valore dell'LQI espresso in base dieci.

In questa sezione è stato messo in evidenza come poter ottenere il valore dell'RSSI da un pacchetto ricevuto in una rete di sensori ZigBee Freescale con chip MC1322x.

## Parte III

# SISTEMA DI LOCALIZZAZIONE IMPLEMENTATO

Nel presente capitolo sarà analizzato il sistema di localizzazione implementato detto anche **sistema ZigBee**, andando ad illustrare in modo dettagliato l'intero sviluppo del sistema. In particolare saranno analizzate le seguenti fasi dell'implementazione:

- Implementazione del cluster RSSI Location e del sistema di comunicazione.
- Implementazione dell'algoritmo.

## 8 Implementazione del Sistema di Comunicazione e del Cluster RSSI Location

È stato implementato il sistema di comunicazione nella rete ZigBee per poter permettere la trasmissione e ricezione dei messaggi necessari alla calibrazione del sistema e alla fase di localizzazione. Il sistema di comunicazione sviluppato fa riferimento al modello **centralizzato** mostrato in Fig.18.

Nelle tecniche Range-Free la fase della calibrazione è necessaria per l'addestramento del motore di localizzazione. Durante tale fase, si raccoglie una quantità considerevole di informazioni proporzionali alle dimensioni dell'ambiente da monitorare e al numero di ancore presenti nel sistema. Considerando che la memoria flash del chip utilizzato (MC13224) è di 128 KB e che la dimensione del BeeStack è circa di 80 KB, rimarrebbe uno spazio di memoria insufficiente per caricare sulla scheda Location Node sia il file di calibrazione, sia il file dell'algoritmo di stima. La decisione di implementare un sistema centralizzato è stata dunque dettata dalle caratteristiche hardware dei nodi utilizzati.

Per realizzare il sistema descritto in Fig.18 è stato implementato il cluster RSSI Location del profilo Telecom Application ed un firmware differente per ogni tipo di nodo (Ancora, Gateway, etc..).

## 8.1 Implementazione del Cluster RSSI Location del T.A.

Inizialmente è stato implementato nel BeeStack il cluster RSSI Location appartenente al profilo applicazione Telecom. Freescale infatti non ha ancora implementato in Bee-Kit la possibilità di creare un progetto selezionando il Telecom Application come profilo applicazione.

In Fig.37 viene illustrata la fase iniziale della creazione del BeeStack in cui è possibile selezionare il profilo Applicazione (*Project types* in figura) e il tipo di sistema da implementare (*OnOffLight, Thermostat, etc..*). In figura è possibile osservare che tra i project types non è presente il profilo T.A.

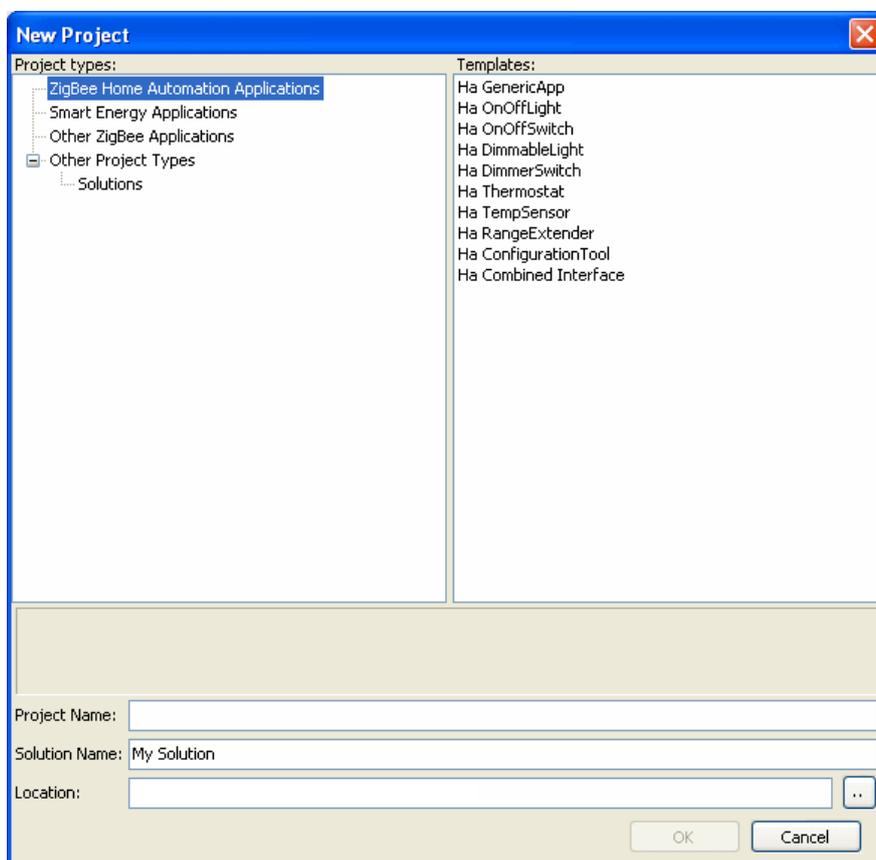


Figura 37: Beekit, scelta dell'applicazione.

Creato il BeeStack ed esportato il firmware in IAR, sono state effettuate tutte le modifiche necessarie ad implementare il Cluster RSSI Location descritto nella sezione 5. Occorre notare che il cluster RSSI Location è stato implementato allo stesso modo per tutti i tipi di nodo come dettato dalle specifiche ZigBee [30], [31]. In particolare:

- è stato definito il ProfileID del profilo applicazione Telecom.

```
/* Telecom Application Profile - little endian (0x0107) */  
#define gTaProfileId_c 0x0701 /* 0x0107, little endian order */  
#define gTaProfileIdb_c 0x07,0x01 /* in byte form */
```

- sono stati aggiunti i DeviceID per le Ancore, il Location Node e il Gateway:

```
#define gTaRSSI_Anchor_Node_c 0x0500 /* Device ID RAN */  
#define gTaRSSI_Location_Node_c 0x0501 /* Device ID RLN*/  
#define gTaRSSI_Location_Gateway_c 0x0502 /* Device ID RLG*/
```

- è stato implementato il Cluster RSSI Location descritto nella sezione 5.3 andando a definire gli identificativi degli attributi, gli identificativi dei comandi e andando a creare le strutture dei messaggi ZigBee da trasmettere. Di seguito viene illustrato un esempio di una struttura creata; tale struttura corrisponde al formato del pacchetto del comando di RSSI Response:

```

typedef struct zclCmdRSSI_RSSIresponse_tag /* Dal Maso 18/09/2011*/
{
zIeeeAddr_t ReplyingDevice;
int16_t X;
int16_t Y;
int16_t Z;
int8_t RSSI;
uint8_t NumberRSSIMeasrments;
}zclCmdRSSI_RSSIresponse_t;

```

## 8.2 Sistema di Comunicazione

In questa fase sono state implementate le funzioni dedicate alla trasmissione e alla ricezione dei comandi e degli attributi necessari a implementare il sistema di comunicazione tra i nodi sia per la fase di calibrazione sia per quella di localizzazione. Tale operazione è stata effettuata andando a modificare in modo opportuno le primitive BeeAppDataIndication e BeeAppDataRequest dei seguenti nodi:

- **Gateway:** è il nodo che si occupa di inoltrare i dati di RSSI al motore di localizzazione del sistema.
- **Ancora:** è il dispositivo che esegue le misure di RSSI.
- **Location Node:** è il nodo da localizzare.
- **Calibration tag:** tale dispositivo è utilizzato per effettuare la calibrazione del sistema.

In particolare è stato creato un file BeeApp.c (file che regola il funzionamento al livello applicazione del nodo) differente per ogni tipo di nodo per implementare il sistema descritto nella Fig.38 secondo le specifiche ZigBee descritte nella sezione 5.

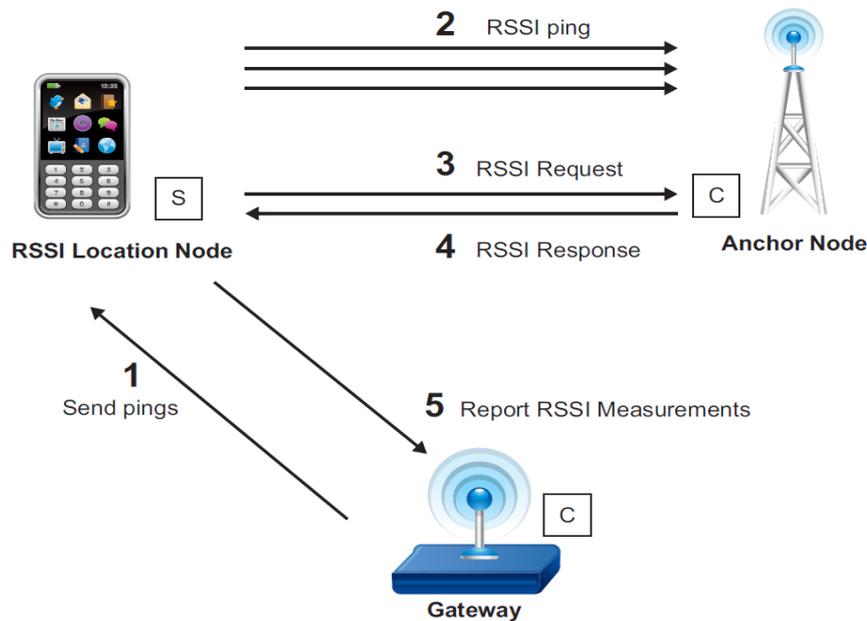


Figura 38: Sistema di comunicazione. [31]

### 8.2.1 Gateway

La funzione di Gateway è stata attribuita al coordinatore della rete ZigBee creata. Il coordinatore è stato infatti collegato tramite porta USB al PC in cui è stato implementato il sistema di localizzazione.

Facendo riferimento alla Fig.38, la funzione del Gateway all'interno del sistema di localizzazione è quella di trasmettere il comando *SendPing* verso il Location Node nel momento in cui si vuole conoscere la posizione del nodo. Tale operazione è stata implementata creando una primitiva denominata *SendPing\_App\_Req*, la quale si occupa di trasmettere il comando *SendPing* al Location Node.

Per quel che riguarda i comandi ricevuti, il Gateway deve riconoscere i pacchetti del tipo *ReportRSSIMeasurement* e trasmettere tali pacchetti dalla porta seriale al PC. Per svolgere tale funzione, durante la procedura guidata della creazione del BeeStack, è stata abilitata la comunicazione tramite porta seriale della scheda. Nel file *BeeApp.c* è stata dunque modificata la primitiva *BeeAppDataIndication* in modo tale che, qualora il contenuto del pacchetto ricevuto venisse attribuito al comando di *Report*, tale pacchetto venga trasmesso al PC .

### 8.2.2 Ancora

Questo tipo di nodo è stato sviluppato come Router. Tale decisione è stata dettata dal fatto che, per localizzare un nodo in un qualsiasi istante, la radio della scheda ZigBee dell'ancora deve essere sempre attiva per poter misurare i valori di RSSI. Tale aspetto ridurrebbe l'autonomia del sistema di localizzazione se le ancore del sistema fossero alimentate a batteria. Tuttavia è stato considerato che in molti ambienti *indoor* è possibile eliminare tale problematica alimentando le ancore con rete elettrica.

Facendo riferimento alla Fig.38, la primitiva `BeeAppDataIndication()` è stata modificata in modo da permettere al nodo di ricevere sia i pacchetti di RSSI Ping che quelli di RSSI Request. Le operazioni che il nodo deve svolgere alla ricezione di questo tipo di comandi sono le seguenti:

- *RSSI Ping*: Il nodo deve registrare il valore dell'LQI dal pacchetto ricevuto.
- *RSSI Request*: Il nodo deve trasmettere in risposta un comando *RSSI Response* contenente il valore di RSSI misurato e le informazioni del nodo stesso (indirizzo dell'ancora, coordinate se presenti, etc...).

Avendo illustrato come il nodo ancora deve trattare i comandi in ricezione, occorre prestare particolare attenzione al seguente problema che si è manifestata durante i primi test: il Location Node e il Calibration tag trasmettono il comando di RSSI Request in **broadcast**; se le ancore trasmettono contemporaneamente in risposta il comando di RSSI Response verso il Location Node, in ricezione ha luogo il fenomeno delle Collisioni. In un ambiente con un'alta densità di ancore (maggiore di tre di copertura del nodo che trasmette il comando di RSSI Request), è stato verificato che tale fenomeno provoca un aumento della perdita di pacchetti e un aumento del tempo necessario a ricevere i pacchetti di RSSI Response perché il firmware implementato sulle schede tenta di gestire le collisioni con delle ritrasmissioni automatiche.

Per la risoluzione di tale problema si è pensato di assegnare un ritardo di risposta al comando *RSSI Request* differente per ogni ancora del sistema (ritardo dell'ordine di 150 ms).

Nel codice è stato possibile risolvere tale problema utilizzando le primitive definite da ZigBee nella sezione Timer del documento [43] denominate `CallBack`. Queste funzioni permettono di richiamare una primitiva desiderata dopo un tempo definito dall'utente :

```

void delayrep()
{
  /*Send Report*/
  TMR_StartSingleShotTimer(appTimerId,175, MyTimerCallBack);
}
void MyTimerCallBack(tmrTimerID_t timerId)
{
  (void)timerId;    /* timer ID not used in this case */
  RSSIResponse_App_Req( RSSI,addrMode_data,aNwkAddr_data,transactionId);
}

```

Nella parte di codice illustrata la funzione *delayrep()* è una primitiva Timer che richiama la funzione *MyTimerCallBack* dopo un tempo pari a 175 ms. Tale funzione si occupa della trasmissione del comando RSSI Response verso il Location Node.

### 8.2.3 Location Node

Il Location Node è il nodo da localizzare all'interno della rete. Occorre notare che tale nodo, essendo indossato da persone o oggetti, non può essere alimentato con l'alimentazione elettrica, quindi per permettere una maggiore durata delle batterie è necessario impostarlo come End Device.

Il Location Node deve poter ottenere e trasmettere al motore di localizzazione le informazioni necessarie alla stima della propria posizione. Per come è stato implementato il sistema, il Location Node avvia la procedura per fornire tali informazioni in tre casi differenti:

- alla ricezione del comando Send Ping. In tal modo un utente di un sistema di monitoraggio, in un momento di necessità, avrebbe la possibilità conoscere le coordinate di un nodo di interesse facendo trasmettere, attraverso ad esempio un'interfaccia grafica utente-motore di localizzazione, al nodo un messaggio di SendPing dal Gateway.
- in modo automatico. Utilizzando le funzioni del tipo myTimerCallBack è possibile far ottenere al Location Node le informazioni necessarie alla localizzazione in modo

automatico con un intervallo di tempo impostato dall'utente. Questa modalità è utile nelle applicazioni dove è necessario monitorare la posizione di un nodo in modo continuo.

- al premere di uno switch se presente sul nodo ZigBee. Tale modalità si è rivelata utile stato per eseguire i test del sistema implementato.

Facendo riferimento alla Fig.38, il sistema di comunicazione dedicato alla trasmissione dei valori di RSSI al motore di localizzazione è stato implementato nel modo seguente:

- Il Location Node trasmette un messaggio di *RSSI Ping* in broadcast (TTL = 1) per far misurare i valori di RSSI alle ancore. La funzione che si occupa della trasmissione del comando è la primitiva denominata *RSSIPing\_App\_Req()*.
- Dopo un tempo  $t_{RQ}$  ( $t_{RQ} = 1.3$  s nel nostro sistema) dalla trasmissione del pacchetto *RSSI Ping* il Location Node trasmette il comando di *RSSI Request* (TTL = 1) automaticamente. Il valore di RSSI misurato dalle ancore viene dunque richiesto dopo ogni RSSI Ping trasmesso.
- Dopo un tempo  $t_{RP}$  ( $t_{RP} = 1.7$  s nel nostro sistema) necessario a ricevere i messaggi di *RSSI Response* (valori di RSSI misurati) dalle ancore del sistema viene avviata una procedura che permette di trasmettere automaticamente i valori di RSSI misurati dalle ancore al Gateway attraverso i messaggi di *ReportRSSIMeasurement*. Occorre notare che la durata del tempo  $t_{RP}$  dipende dal numero di ancore che trasmettono il valore di RSSI. Nei test effettuati è stato verificato che un tempo  $t_{RP} = 1.7$  s è sufficiente a ricevere un valore di RSSI da quattro ancore diverse.

Nei test effettuati (cap 5) sarà evidente, che per migliorare l'accuratezza della stima della posizione del nodo, il numero di *RSSI Ping* effettuati per ogni posizione da stimare dovrà essere un numero  $n > 1$ . A tal fine la procedura descritta in precedenza viene eseguita automaticamente un numero  $n$  di volte alla ricezione di un comando di *Send Ping*, al premere di uno switch o automaticamente ogni intervallo di tempo impostato dall'utente.

### 8.3 Calibration tag

Il Calibration tag è il nodo che l'utente deve utilizzare per effettuare la calibrazione del sistema. Dato che per tale operazione vengono misurati dei valori di RSSI da delle posizioni di riferimento è necessario utilizzare un nodo con switch. Deve essere infatti l'utente che esegue la calibrazione a decidere quando far misurare i valori di RSSI alle antenne del sistema.

Nell'implementazione di un sistema di localizzazione con tecniche Range-Free occorre prestare particolare attenzione al tempo richiesto all'utente per effettuare la calibrazione. Dato che tale sistema può essere installato in ambienti di dimensioni considerevoli, è opportuno ridurre al minimo possibile il tempo di calibrazione. Risulta infatti inefficiente richiedere all'utente tempi dell'ordine delle ore/giorni per effettuare tale operazione.

Pertanto si è tentato di implementare il sistema di comunicazione per effettuare la calibrazione nel modo più efficiente possibile tentando di ridurre il tempo richiesto da tale fase al minimo possibile. Per raggiungere tale obiettivo sono stati effettuati molteplici tentativi.

Inizialmente il sistema di comunicazione era implementato nel modo seguente: l'utente, per poter trasmettere ogni pacchetto *RSSI Ping*, era obbligato nelle posizioni di calibrazione a premere il pulsante associato a tale comando. Successivamente, per richiedere le misure effettuate dalle antenne e inoltrarle poi al Gateway, era necessario premere lo switch associato alla richiesta di *RSSI Request*.

Questo sistema si è però rivelato lento e macchinoso. Se ad esempio l'utente avesse voluto calibrare il sistema richiedendo cinque misure di RSSI per ogni posizione di calibrazione, sarebbe stato costretto a prestare particolare attenzione al numero di *RSSI Ping* e *Request* effettuati in ogni posizione di calibrazione. La calibrazione sarebbe stata un'operazione pesante dal punto di vista dell'utente.

Per risolvere l'inconveniente sopra descritto il firmware della scheda è stato sviluppato nello stesso modo del Location Node, ovvero sono state apportate le seguenti modifiche al sistema di comunicazione:

- Utilizzando le funzioni di Timer e Callback, si è reso automatico il processo di trasmissione dei pacchetti di *RSSI Ping*. L'utente era dunque in grado di trasmettere un numero prestabilito ( $n$ ) di pacchetti *RSSI Ping* premendo il pulsante associato a tale comando una singola volta per ogni posizione di calibrazione.
- Dopo ogni trasmissione del comando *RSSI Ping* si è resa automatica anche la trasmissione dei relativi comandi di *RSSI Request* e di *Report RSSI Measurement* eliminando

così la necessità di premere il pulsante associato al comando *RSSI Request*.

Il sistema sviluppato benché automatico richiedeva tempi di calibrazione troppo elevati. Ad esempio, in un ambiente caratterizzato da tre Ancore e cinque misure di *RSSI Ping* per ogni posizione di calibrazione, l'utente sarebbe dovuto rimanere fermo in ogni posizione per 5/6 minuti. Considerando che in un ambiente di dimensioni ridotte (circa 80 m<sup>2</sup>) il numero di punti di calibrazione possono essere maggiori di trenta, il tempo necessario a calibrare il sistema sarebbe dell'ordine delle ore, un tempo non accettabile.

Un ulteriore tentativo di ridurre il tempo di calibrazione è stato il seguente seguente:

- La trasmissione dei comandi di *RSSI Ping* e di *RSSI Request* veniva eseguita in modo automatico come descritto in precedenza. I messaggi di Request erano salvati in modo ordinato dal Location Node per ogni posizione durante la fase di calibrazione.
- Una volta terminate le posizioni di calibrazione, i messaggi di *Report* venivano inoltrati automaticamente al Gateway. L'avvio dell'inoltro dei pacchetti avviene premendo un tasto.

I risultati ottenuti dai test effettuati con il sistema descritto mostravano che il tempo impiegato dall'utente nella fase di calibrazione fosse stato ridotto di un fattore dieci nonostante che il tempo di calibrazione fosse pressoché identico al metodo precedente. Una volta effettuate le misure per ogni posizione di calibrazione, l'utente aveva infatti la possibilità di far trasmettere tali misure dal Location Node al Gateway senza che fosse necessaria la sua presenza. Il sistema implementato aveva però una grande limitazione. Per i problemi legati alla memoria delle schede utilizzate il numero massimo di misure che si potevano registrare nella scheda era millecinquecento. Considerando un sistema costituito da quattro ancore e dieci richieste di misura per ogni posizione di calibrazione il numero massimo di tali posizioni sarebbe stato trentasette. Anche se non è ancora stata fornita un'idea del numero di posizioni di calibrazione in funzione dell'ambiente da monitorare, avere un limite alle dimensioni della calibrazione avrebbe reso il sistema non scalabile.

Per risolvere tale problema il sistema è stato riorganizzato nel modo seguente:

- È stato imposto che durante la fase della calibrazione il Calibration tag debba essere collegato ad un PC portatile in modo da poter essere trasportato dall'utente durante

la calibrazione. Tale imposizione può essere giustificata dal fatto che in un sistema la calibrazione è unica e che i parametri tempo di calibrazione e scalabilità siano molto più importanti della comodità di esecuzione. Il Calibration tag, essendo direttamente collegato ad un PC, non deve utilizzare né registrare i valori di RSSI né utilizzare i messaggi di Report per inviarli al motore di Localizzazione.

- Terminata la fase della calibrazione, il file generato sul PC portatile può essere trasferito al PC del motore di localizzazione in molteplici modi (attraverso internet, penna USB, etc).

In questa maniera il sistema è stato reso anche scalabile. Occorre notare che il tempo di calibrazione di questa implementazione corrisponde al tempo richiesto all'utente per calibrare il sistema nell'implementazione precedente.

## 9 Implementazione del motore di localizzazione

La presente sezione ha come obiettivo quello di illustrare le fasi dell'implementazione del motore di localizzazione.

Data la possibilità di sviluppare tale sistema su un Server (sistema di comunicazione centralizzato), il sistema è stato implementato nel linguaggio di programmazione ad alto livello denominato Python.

Il Python è un linguaggio di programmazione caratterizzato da un'ottima efficienza in termini di computazione numerica [45]. Per tale linguaggio vengono creati dagli sviluppatori dei moduli aggiuntivi da installare nella *directory* del programma che permettono di estenderne le potenzialità.

Per sviluppare il sistema di localizzazione sono stati utilizzati i seguenti moduli aggiuntivi:

- *pyserial*: è un modulo che permette al Python di gestire i dati in ingresso dalla porta seriale [46].
- *numpy*: è il modulo del Python dedicato al calcolo scientifico (vettoriale, matriciale, etc..) [47].
- *matplotlib*: è un modulo grafico di Python. Permette di creare grafici e trattare le immagini.

Per l'implementazione del sistema di localizzazione sono stati sviluppati due programmi:

- *Calibrazione*: deve essere avviato durante la fase di calibrazione del sistema.
- *Localizzazione*: deve essere avviato durante la fase di localizzazione.

### 9.1 Calibrazione

L'obiettivo di tale fase è quello di ottenere le informazioni necessarie a pilotare il motore di localizzazione. Tale operazione deve essere eseguita correttamente al fine di ottenere la massima accuratezza della stima. Per come è stato implementato il sistema, per evitare problemi al motore di localizzazione descritti nella sezione 10.3, l'utente che effettua tale operazione deve prestare particolare attenzione alla posizione delle ancore evitando che

siano poste dietro oggetti rimovibili che creano una forte attenuazione del segnale ricevuto come armadi, etc...(ancore in ombra) .

L'utente, per effettuare tale operazione, deve avviare sul PC collegato al Calibration tag il programma *Calibrazione*. Tale programma permette di creare i file denominati "*Calibrazione.dat*" e "*Anchor.dat*". Il file "*Calibrazione.dat*" contiene le misure di RSSI effettuate da ogni posizione di calibrazione e gli identificativi delle Ancore che hanno misurato da tale posizione. Eseguiti ad esempio un numero  $n$  di *RSSI Ping* in ogni punto di calibrazione dal Location Node, le informazioni sono salvate con la seguente struttura:

- *NumeroAnchor*: nella prima linea della struttura salvata nel file viene registrato il numero  $k$  di Ancore che hanno misurato almeno un valore di RSSI dalla posizione di riferimento in esame.
- *AnchorID,Misure*: per ognuna delle  $k$  linee successive viene registrato l'identificativo dell'Ancora  $k$ -esima e il numero  $n_k$  di misure effettuate dal nodo in tale posizione.

Un esempio del file creato durante la fase di calibrazione è mostrato in Figura.39. In tale esempio, nella prima posizione, hanno misurato due Ancore. Nelle due righe successive sono riportati i gli identificativi di tali Ancore e i valori di RSSI misurati.

```

2
0000000000000000,-53,-49,-58,-58,-57
00000000000000004,-62,-63,-60,-59,-62
2
0000000000000000,-61,-70,-63,-60,-70
00000000000000004,-75,-61,-61,-77,-77
2
0000000000000000,-61,-63,-64,-66,-75
00000000000000004,-74,-64,-63,-61,-61
2
0000000000000000,-68,-72,-58,-71,-68
00000000000000004,-65,-63,-68,-77,-69
2
0000000000000000,-77,-69,-76,-81,-72
00000000000000004,-77,-65,-57,-62,-73
2
0000000000000000,-71,-63,-80,-74,-66
00000000000000004,-70,-76,-63,-65,-65
2
0000000000000000,-64,-60,-66,-79,-81
00000000000000004,-79,-80,-74,-68,-70

```

Figure 39: Esempio del file Calibration.dat.

Il file “*Anchor.dat*” contiene invece gli identificativi di tutte le Ancore della rete. Un esempio del file è illustrato in Fig.40 .

```

0000000000000000
0000000000000004
0000000000000002
0000000000000001

```

Figure 40: Esempio del file Anchor.dat.

Il diagramma di flusso del programma che si occupa della creazione dei file sopra descritti è mostrato in Figura 41.

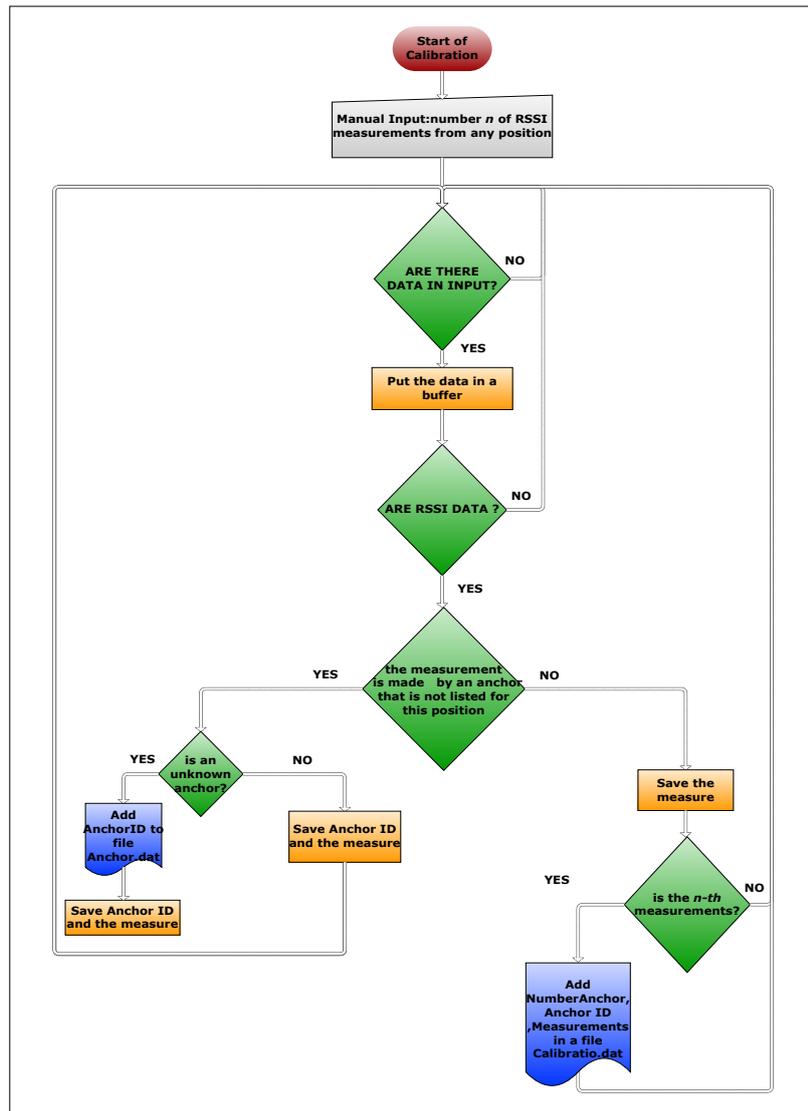


Figure 41: Diagramma di flusso del programma Calibrazione.

Analizzando il diagramma di flusso illustrato in figura è possibile effettuare la seguente descrizione:

1. Inizialmente, per la corretta esecuzione del programma, viene richiesto da linea di comando il numero di *RSSI Ping* che saranno effettuati dal Calibration Tag in ogni

posizione della griglia di calibrazione verso le ancore della rete. Tale viene registrato in una variabile denominata *NumberLQI*.

2. Successivamente è presente un loop di attesa di dati in ingresso dalla porta seriale collegata al Calibration tag. Se sono presenti dati in ricezione, i dati ricevuti vengono registrati in un buffer.
3. Se i dati ricevuti da seriale corrispondono al pacchetto di *RSSI Request*, da tale messaggio vengono estratte le informazioni necessarie a creare la struttura descritta in precedenza (AnchorID, MisuraRSSI). È possibile identificare un messaggio di *RSSI Request* dai particolari valori che assumono i campi *OpcodeGroup*, *OpCode*, *Lenght*, *ProfileID*, *ClusterID* e *CommandID* di tale pacchetto. In particolare *OpcodeGroup* = 0x9E, *OpCode* = 01, *Lenght* = 33, *ProfileID* = 0x0701, *ClusterID* = 0x000B, *CommandID* = 0x04.
4. Dopo un numero *NumberLQI* di *RSSI Ping* la struttura completa viene aggiunta nel file *Calibration.dat*, mentre ogni nuova ancora viene registrata nel file *Anchor.dat*.
5. Completate le posizioni di calibrazione l'utente termina il programma di calibrazione dalla linea di comando del PC collegato al Calibration tag.

I file creati vengono utilizzati successivamente per la creazione di un database necessario per pilotare il motore di localizzazione.

## 9.2 Localizzazione

Il programma *Localizzazione* deve essere installato sul computer collegato al Gateway ed è suddiviso nelle seguenti fasi:

- *Inizializzazione del motore di localizzazione* : Questa operazione viene eseguita un'unica volta all'avvio del programma e viene utilizzata per creare il database in modo da permettere al Location Engine di utilizzare i dati contenuti nei file di calibrazione *Anchor.dat* e *Calibration.dat* al fine di stimare la posizione del nodo.
- *Motore per la localizzazione* : In questa fase viene attivato il Location Engine per fornire la stima della posizione del nodo ogni qualvolta sono ricevuti dei dati di RSSI misurati dalla porta seriale.

### 9.2.1 Inizializzazione del motore di localizzazione

L'obiettivo di questa parte del software è quello di creare un database. In generale il termine database indica un insieme di archivi collegati secondo un particolare modello logico e in modo tale da consentire la gestione dei dati stessi da parte di applicazioni software dedicate. In particolare, per il presente sistema, creare un database efficiente dai file "Calibrazione.dat" e "Anchor.dat" è necessario per ridurre il tempo di elaborazione richiesto al motore per stimare la posizione del nodo. Il diagramma di flusso che descrive la creazione del database è illustrato in Figura :

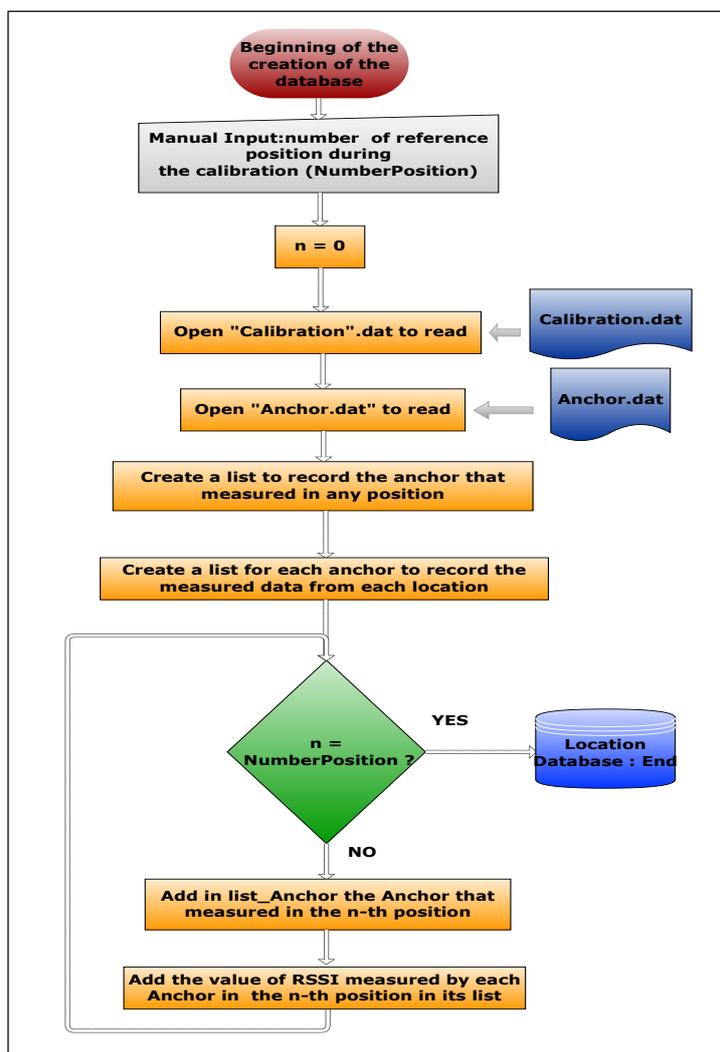


Figure 42: Diagramma di flusso della creazione del database.

Come illustrato in figura, in questa fase vengono estratti i dati dal file Calibration.dat e dal file Anchor.dat. Ad ogni struttura di dato estratta dal file Calibration.dat, corrispondente alle misure effettuate in una posizione di calibrazione, viene associato un indice in ordine crescente denominato *indice di posizione*.

Per organizzare i dati in modo efficiente si è pensato di utilizzare delle strutture particolari che il linguaggio Python permette di creare, dette *Dizionari*. Un dizionario rappresenta una collezione "non ordinata" di oggetti. Gli oggetti sono identificati univocamente da una chiave (generalmente una stringa). Ogni elemento del dizionario è rappresentato da una coppia (chiave : valore), la chiave serve per accedere all'elemento e recuperare il valore ricercato. Il valore associato ad una chiave può essere anch'esso una lista o un dizionario.

In particolare, nella fase di Inizializzazione, si creano due tipi differenti di dizionario differenti:

- *Anchor*: tale dizionario è unico; le chiavi sono gli indici delle posizioni di riferimento della griglia di calibrazione mentre i valori del dizionario sono la liste degli identificativi delle ancore che hanno misurato almeno un valore di RSSI. Per ogni posizione di riferimento(chiave) viene dunque creata una lista(valore) contenente gli identificativi delle Ancore che hanno misurato almeno un valore di RSSI nella posizione indicata dall'indice di posizione. In tal modo si viene a creare una tabella in cui per ogni posizione di riferimento sono specificati gli identificativi delle Ancore che hanno effettuato almeno una misura.
- *RSSIAnchor*: per ogni Ancora del sistema viene creato un dizionario in cui le chiavi sono gli indici delle posizioni di calibrazione dalla quale tale ancora ha effettuato almeno una misura(valore). Il contenuto del dizionario per ogni indice di posizione(chiave) è una lista (valore) delle misure di RSSI effettuate per ogni specifica posizione di calibrazione. In tal modo se in un ambiente di localizzazione sono presenti  $n$  Ancore saranno creati  $n$  dizionari.

Una volta terminata la parte di programma descritta, si ottiene un database da poter utilizzare nel sistema di localizzazione.

Per completare la fase di Inizializzazione è necessario fornire al motore di localizzazione un file denominato *Coordinate.dat* che contiene le coordinate delle posizioni di calibrazione. Le coordinate devono essere ordinate nel file secondo l'ordine degli indici di posizione. Un esempio di tale file è mostrato in Fig.43.

0.0 , 0.0
0.0 , 1.0
0.0 , 2.0
0.0 , 3.0
1.0 , 0.0
1.0 , 1.0
1.0 , 2.0
1.0 , 3.0

Figura 43: Esempio del file Coordinate.dat.

In figura, su ogni riga sono presenti le coordinate (x,y) in metri delle posizioni di calibrazione. Tale modo di procedere risulta troppo macchinoso per essere implementato in un sistema di localizzazione commerciale . Nel capitolo dedicato agli sviluppi futuri saranno presentate delle possibili soluzioni alternative.

### 9.2.2 Motore per la Localizzazione

La seconda parte del programma si occupa di stimare le coordinate del Location Node ogni qualvolta vengono trasmessi al Gateway dei dati di RSSI misurati durante la fase di localizzazione. I dati di RSSI ricevuti, insieme ai dati estrapolati dal database, vengono utilizzati dal Location Engine per stimare le coordinate del nodo.

Il Location Engine è stato implementato secondo il metodo Bayesiano descritto nella sezione 2.3.3.

Nel sistema implementato, la probabilità condizionata  $P(\underline{s}/l)$ , viene stimata secondo il metodo della funzione Kernel:

$$P(\underline{s}/l) = \frac{1}{n_l} \sum_{i=1}^{n_l} K(s; s_i) \quad (35)$$

in cui

- $n_l$  è il numero di campioni dell’RSSI collezionati durante la fase di calibrazione nella  $l$ -esima posizione di calibrazione.
- $K(\cdot; s_i)$  è la funzione Kernel.

- $s_l$  è l'iesimo valore di RSSI misurato nella l-esima posizione di calibrazione.
- $s$  è il valore di RSSI misurato durante la fase di localizzazione.
- $l$  è la posizione di calibrazione.

In [12] viene specificato che i dati di calibrazione possono essere elaborati in due modi differenti al fine di ottenere la stima della densità di probabilità dal metodo Kernel. In particolare i dati possono essere trattati nei modi seguenti:

- *Cluster*: I dati collezionati durante la fase di calibrazione vengono raggruppati in cluster. In ogni singolo cluster sono registrati i valori di RSSI ottenuti dalla medesima posizione. Se sono dunque presenti  $n$  posizioni di calibrazione, alla fine di tale operazione saranno creati  $n$  cluster. I dati di ogni cluster distinto vengono utilizzati per stimare la densità di probabilità con il metodo Kernel. Se ad esempio in un cluster sono presenti  $k$  misure di RSSI, la densità di probabilità sarà stimata come somma di  $k$  funzioni Kernel.
- *Single*: Tale modalità di trattare i dati consiste nel considerare anche i dati appartenenti ad un cluster come misurati da posizioni diverse. Se ad esempio in un cluster sono presenti  $k$  misure di RSSI, saranno stimate  $k$  densità di probabilità ognuna costituita da un'unica funzione Kernel. Nel metodo descritto se fossero stati registrati  $k$  RSSI in ognuna delle  $n$  posizioni di calibrazione sarebbero stimate  $n \cdot k$  densità di probabilità e sarebbero dunque calcolate  $n \cdot k$  probabilità.

Dopo una serie di test sperimentali, in [12] viene messo in evidenza che con il secondo metodo si ottiene un errore medio sulla stima della posizione inferiore. Per verificare quanto redatto in [12], sono state implementate entrambe le modalità di trattare i dati del training set.

Occorre notare che, come messo in evidenza nella sezione 2.3.3, in ogni posizione si è fatta l'ipotesi di indipendenza tra le misure di RSSI effettuate dalle Ancore del sistema. Se ad esempio in una posizione misurano valori di RSSI tre Ancore, saranno stimate tre  $p(\underline{s}/l)$ . La  $p(\underline{s}/l)$  sarà il prodotto delle tre probabilità stimate:

$$p_{tot}(k) = \prod_{i=0}^N p_i(s/l) \quad (36)$$

Dove  $N$  è il numero di Ancore che hanno misurato nella posizione  $k$ -esima.

Durante la fase della localizzazione, la stima della posizione del Location Node avviene nel modo seguente:

- Il Location Node trasmette dalla posizione incognita un numero  $n$  di RSSI Ping.
- Il Location Engine fornirà per entrambi i metodi Kernel implementati un numero  $n$  di stime  $(\hat{x}, \hat{y})$

La stima della posizione del nodo fornita all'utente si ottiene come media delle  $n$  stime effettuate :

$$(\hat{x}, \hat{y}) = \sum_{i=1}^n (\hat{x}_i, \hat{y}_i) \quad (37)$$

I due algoritmi di stima implementati sono stati indicati come:

- **Cluster Method**
- **Single Method**

In Fig.44 viene illustrato un esempio della stima della posizione del Location Node. L'ambiente mostrato in figura rappresenta la planimetria di due stanze del Polo di Sistemi Logistici di Livorno. Il quadrato verde rappresenta la posizione reale del nodo mentre il triangolo giallo rappresenta la stima della posizione del nodo ottenuta mediando le coordinate di dieci singole stime illustrate con i cerchi rossi.

Successivamente, per tentare di migliorare l'accuratezza del sistema, si è pensato di trattare i dati di RSSI ricevuti anche in modo differente:

- I dati di RSSI ottenuti dagli  $n$  RSSI Ping effettuati dalla posizione incognita vengono mediati per ogni ancora.
- Dopodiché viene stimata la posizione del nodo con entrambi i metodi Kernel utilizzando i valori di RSSI mediati. La posizione del nodo viene dunque fornita eseguendo un'unica stima.

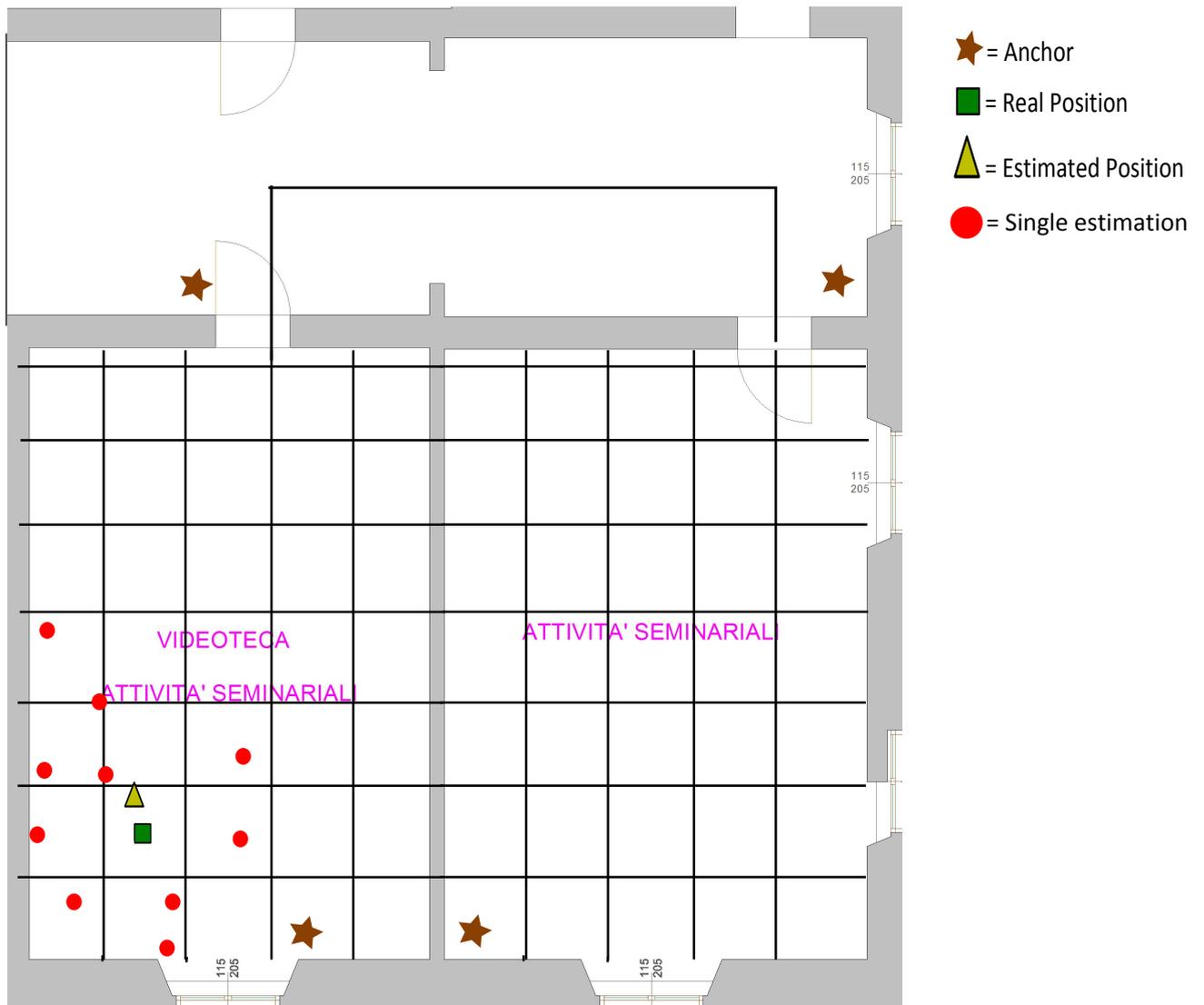


Figura 44: Esempio di localizzazione con i metodi Single e Cluster.

Quest'ultimi metodi di stima implementati sono stati indicati come:

- **Cluster RSSI Average Method**
- **Single RSSI Average Method**

### 9.3 Esempio di Applicazione

Per chiarire i concetti esposti nella presente capitolo viene riportato un esempio di implementazione del sistema sviluppato. Si supponga di voler installare il sistema in un

ambiente:

- L'utente che deve eseguire la calibrazione deve stare attento che le ancore del sistema non siano poste dietro oggetti rimovibili(ancore in ombra). Successivamente l'utente deve eseguire la calibrazione utilizzando il Calibration tag, un Pc portatile e avviando il programma "Calibrazione".
- Terminata la fase di calibrazione, l'utente deve creare il file *Coordinate.dat* e deve trasferire i file di calibrazione al Pc in cui è installato l'Engine per poterlo inizializzare.
- Avviato il programma "Localizzazione" e creato il database, si avvia il motore di localizzazione.
- Si consideri ad esempio di ottenere dal Location Node le AnchorID e le misure di RSSI effettuate da due Ancore del sistema.
- Il database creato viene utilizzato nel modo seguente: Come prima operazione vengono selezionate le posizioni in cui viene considerato probabile che si trovi il Location Node e vengono eliminate le restanti. Tale selezione viene effettuata andando a considerare nel dizionario Anchor solo gli indici delle posizioni che contengo nella lista delle ancore entrambi gli AnchorID delle ancore fornite dal Location Node. Le posizioni candidate sono riportate in una lista denominata *Candidate*. Il file *Candidate* viene utilizzato per andare a prelevare nel file *Coordinate.dat* solo le coordinate delle posizioni probabili da utilizzare nell'equazione (28) per stimare la posizione del Location Node con i quattro metodi descritti. Tale logica di decisione rende più efficiente il motore di localizzazione andando a ridurre la dimensione del problema da far risolvere al Location Engine. La logica di decisione descritta ha tuttavia due problemi: se la calibrazione viene eseguita dall'utente in modo scorretto (ancore in ombra), ovvero durante la fase di localizzazione un'ancora aumenta considerevolmente il proprio raggio di copertura in una zona oppure viene aggiunta dopo la calibrazione un'ancora al sistema, le posizioni nel raggio di copertura di tale ancora non saranno più selezionate. Il secondo problema viene risolto andando a considerare, durante la fase di localizzazione, solo le ancore presenti nel file *Anchor.dat*, mentre per evitare i problemi legati alla prima è richiesto all'utente che esegue la calibrazione di prestare attenzione alla posizione delle ancore.

Nel capitolo dedicato alle verifiche sperimentali saranno illustrati i test eseguiti con i quattro metodi descritti e saranno riportati i risultati in termini di prestazioni del sistema al variare dei parametri principali.

## Parte IV

# SISTEMA DI LOCALIZZAZIONE EKAHOU

## 10 Overview del Sistema RTLS

Ekahau è l'azienda leader nel settore dell'offerta di sistemi di localizzazione Real Time Location System (RTLS) per reti wireless basate su tecnologia Wi-fi. Come dichiarato dalla stessa azienda in [50], l'obiettivo di Ekahau inc. è quello fornire il miglior sistema di localizzazione per persone e oggetti in termini di facilità di implementazione, costo e prestazioni. Il prodotto offerto sul mercato dall'azienda è denominato Ekahau-RTLS e ha un prezzo di circa 3000/4000 \$. Tale prodotto è costituito da un insieme di elementi che permettono di localizzare tag Wi-fi, cellulari o computer portatili connessi ad una rete Wifi.

Il kit Ekahau RTLS è costituito dagli elementi illustrati in tabella 14 uniti ai rispettivi manuali di uso per l'utente.

Component	Description	Supported Platforms
 Ekahau Positioning Engine™	Server software for location tracking, Wi-Fi tag configuration and management. HTTP / XML API for applications.	Windows® XP, Windows® 2003 Server
Ekahau Site Survey™	Application for creating and editing Positioning models	Windows® XP, Windows® 2003 Server
 Ekahau Vision™	End-user application for grouping, locating and viewing the location of people and assets in real time, monitor events, and invoke notifications or alarms.	Windows® XP, Windows® 2003 Server
Ekahau Wi-Fi Tags	Wi-Fi location tag	-

Tabella 14: Elementi del kit Ekahau RTLS. [51]

Nel presente capitolo saranno illustrati brevemente gli elementi del kit Ekahau RTLS mostrati in tabella.

## 10.1 Ekahau Positioning Engine

L'Ekahau Positioning Engine (EPE) è il software che si occupa della stima delle coordinate del Location Node. L'EPE permette inoltre di eseguire il tracking di un nodo mobile Wi-Fi, ovvero permette di seguire gli spostamenti di un nodo selezionato all'interno della rete.

L'installazione di tale programma avviene tramite una procedura guidata contenuta in un pacchetto di installazione fornito nel kit RTLS. Come illustrato in Fig.45, durante tale procedura è possibile selezionare l'indirizzo IP da associare al Positioning Engine, le porte TCP e le porte UDP per poter accedere al software da remoto.



Figura 45: Set up del server. [51]

Installato il programma, l'EPE si presenta all'utilizzatore con un'interfaccia utente web browser. La pagina principale con cui viene visualizzato l'Ekahau Positioning Engine dall'utente è mostrata in Fig.46.



Figura 46: Ekahau Positioning Engine, pagina principale. [51]

Nella pagina web dell'EPE illustrata in figura è possibile accedere alle seguenti informazioni:

- **System:** Fornisce un panoramica sullo stato dello stesso. In tale pagina è possibile accedere anche al file di Log, ovvero il file in cui sono contenute le informazioni sull'attività del sistema inclusi errori e warnings.
- **Tags:** Vengono visualizzati i tag attivi (presenti nella rete) nel sistema di localizzazione, è possibile visualizzare lo stato di un tag (livello delle batterie, da quanto tempo non fornisce posizione sulla propria locazione) e comunicare essi.
- **Group:** Viene visualizzata la lista dei tag che appartengono ad un gruppo. I tag possono essere assegnati ad un gruppo se appartengono alla stessa applicazione,
- **Config :** Da tale pagina è possibile configurare i tag in funzione delle applicazioni di impiego degli stessi. In particolare i tag possono essere configurati in modo che aggiornino la posizione automaticamente in "tempo reale", durante o dopo un movimento oppure al premere di uno switch presente sul tag.

- **Models:** Nella pagina citata è possibile visualizzare le mappe di calibrazione create utilizzando il programma Ekahau Site Survey. In tale pagina è possibile selezionare la mappa di calibrazione da far utilizzare al Location Engine per stimare le coordinate dei nodi (Active model).

## 10.2 Ekahau Site Survey

Ekahau Site Survey (ESS) è il software utilizzato per eseguire la calibrazione del sistema di localizzazione. Attraverso l'uso di tale programma si crea un modello (Positioning Model), costituito dagli RSSI ricevuti dai vari Access Point della rete Wi-Fi, con cui si addestra il Location Engine.

Nel kit RTLS, insieme al pacchetto di installazione dell'ESS, viene fornita una scheda di rete esterna NIC 300 Wi-Fi Adapter da poter utilizzare per effettuare la calibrazione in modo da ottenere le prestazioni migliori.

In Fig.47 è mostrata l'interfaccia utente del programma dopo aver eseguito un'operazione di calibrazione.

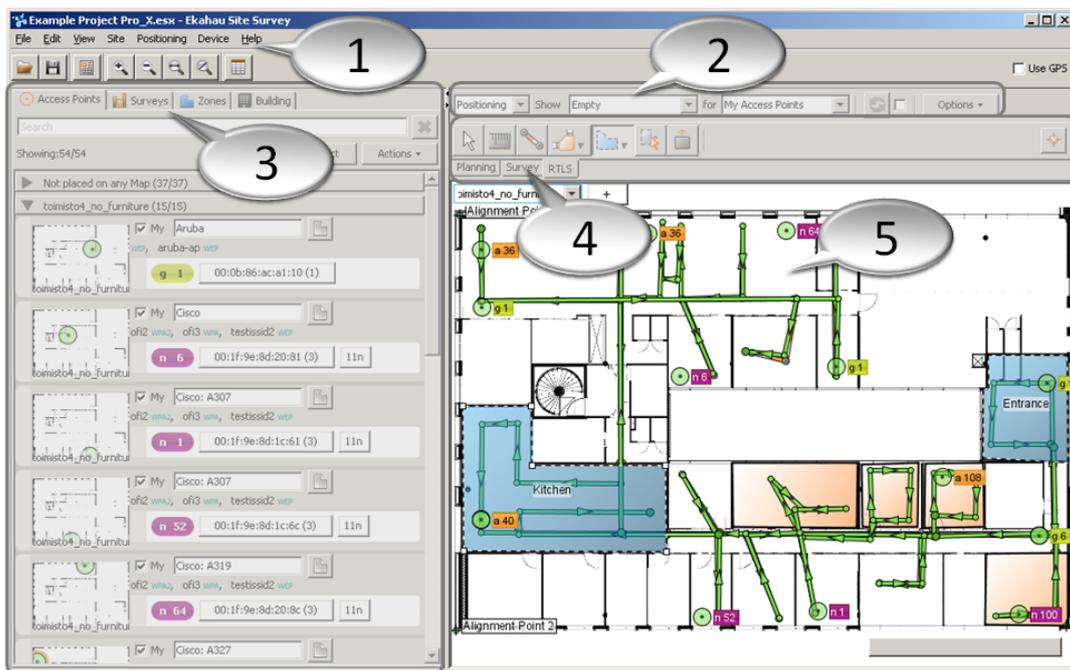


Figura 47: Ekahau Site Survey. [52]

Facendo riferimento al manuale utente [52] e alla figura mostrata, la calibrazione del sistema di localizzazione Ekahau avviene eseguendo la seguente procedura:

- Nella pagina principale dell'ESS è possibile caricare e visualizzare la planimetria (indicata nel programma con il nome mappa) dell'ambiente in cui è installata la rete Wi-Fi. L'utente deve indicare al programma le dimensioni reali dell'ambiente di localizzazione. Tale operazione avviene ad esempio indicando le dimensioni reali di una parete di una stanza visualizzata sulla mappa.
- Successivamente la fase della calibrazione procede andando a indicare nella mappa se ci sono zone (stanze) con superficie superiore a 25 m<sup>2</sup> e "abbastanza libere da ostacoli" denominate **Open Space**. In figura, tali zone, sono rappresentate dalle stanze in rosso. Nelle stanze che non sono state selezionate come Open Space è possibile invece tracciare delle linee indicate con il nome di **Rails**. I Rails servono ad indicare il percorso più probabile che un utente può eseguire durante uno spostamento. Ad esempio se viene tracciata una linea da una stanza al corridoio indica che l'utente, per uscire dalla stanza, è più probabile che attraversi la porta della stanza invece che un muro. Una volta definiti questi parametri l'utente può eseguire la calibrazione.
- L'utente, per eseguire la calibrazione, deve camminare con il PC in mano lentamente e con velocità costante all'interno delle stanze in cui è necessario implementare la calibrazione. L'utente, mentre esegue la calibrazione, deve indicare il percorso che sta eseguendo (indicato in verde in figura) nella mappa dell'ESS. Durante il percorso vengono registrati i valori di RSSI misurati dalle Ancore del sistema.
- Una volta terminata la calibrazione, l'ESS genera un file di calibrazione che può essere trasmesso all'EPE.

## **Ekahau Vision**

Tale programma permette di monitorare, sulla mappa inviata dall'ESS all'EPE, le posizioni e gli spostamenti dei tag Wi-Fi.

## Parte V

# VERIFICHE SPERIMENTALI : TEST E CONFRONTI PRESTAZIONALI

## 11 Ambiente dei test

Nel presente capitolo saranno illustrati i test effettuati con il sistema di Ekahau e con il sistema di localizzazione implementato prestando particolare attenzione ai risultati ottenuti in termini di accuratezza della stima e di tempo di calibrazione. Dato che le prestazioni di un sistema di localizzazione dipendono da come viene eseguita la calibrazione, per entrambi i sistemi è stato valutato l'errore medio di localizzazione (accuratezza) effettuando diverse tipologie di calibrazione. Sarà infine illustrato un confronto tra le prestazioni dei due sistemi analizzati.

Nella figura seguente si mostra la planimetria dell'ambiente dei test.

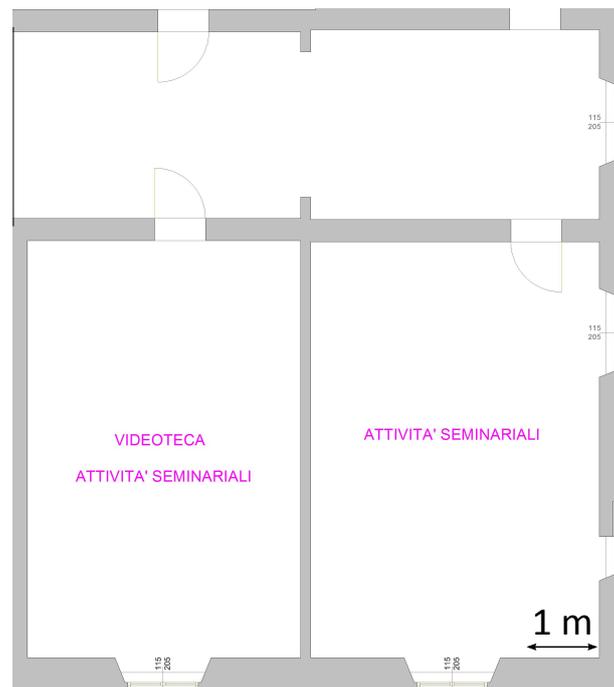


Figura 48: Ambiente dei test effettuati per la localizzazione.

Come illustrato in figura, l'ambiente utilizzato per eseguire i test è costituito da due uffici ed un corridoio che coprono complessivamente un'area di  $90 \text{ m}^2$   $9 \text{ m} \times 10 \text{ m}$  presso il Polo Universitario di Sistemi Logistici di Livorno.

## 12 Test Ekahau

Per verificare le prestazioni del sistema di localizzazione di Ekahau sono state eseguite tre tipologie di calibrazione differenti:

- *Calibrazione 1*: dopo aver creato una rete Wi-Fi costituita da tre Access Point Netgear ProSafe WAG102 Dual Band la calibrazione, illustrata in Fig.49a, è stata eseguita camminando lungo il perimetro delle stanze.
- *Calibrazione 2*: è stato rimosso l'Access Point della rete Wi-Fi indicato in Fig.49a mentre il calibrazione è stato il medesimo della *Calibrazione 1*.
- *Calibrazione 3*: sono stati utilizzati nuovamente tre Access Point per costituire la rete Wi-Fi ed è stato modificato il percorso di calibrazione come illustrato in Fig.49b.

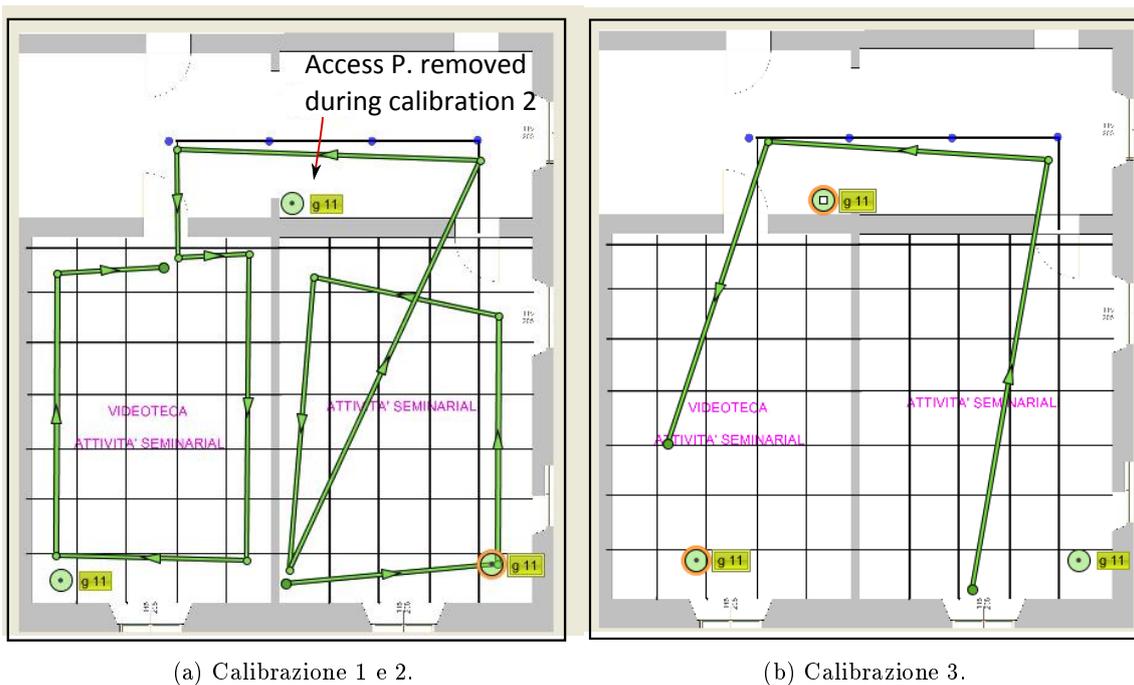


Figure 49: Calibrazioni effettuate nei test del sistema Ekahau.

Nelle figure illustrate i cerchi verdi rappresentano i nodi (Access Point) della rete Wi-Fi mentre le linee verdi rappresentano il percorso di calibrazione effettuato.

In Tabella 15 sono riportati i tempi di calibrazione, in secondi, impiegati per effettuare i due percorsi di calibrazione mostrati in Fig.49.

	Calibration 1 and 2	Calibration 3
Calibration time (s)	130	50

Tabella 15: Tempo di calibrazione nei test effettuati.

Come illustrato in tabella, il tempo necessario a calibrare il sistema di localizzazione di Ekahau per un'area di 100 m<sup>2</sup> è dell'ordine del minuto.

Al termine di ogni calibrazione effettuata è stato utilizzato un test set costituito da quindici posizioni di coordinate note (differenti da quelle di calibrazione) in cui è stato posto il Location Node in modo da verificare le prestazioni in termini di accuratezza del sistema RTLS. In ogni posizione nota sono state fatte eseguire dall'Engine quattro stime di posizione per un totale di sessanta stime. Per ogni posizione di riferimento è stato calcolato l'errore medio di stima secondo :

$$\text{Error}(k) = \left(\frac{1}{4}\right) \sum_{i=1}^4 \sqrt{(\hat{x}_i - x_k)^2 + (\hat{y}_i - y_k)^2} \quad (38)$$

in cui:

- $k$  è l'indice della posizione di test set;
- $i$  è l'indice delle stime eseguite in ogni posizione di test set;
- $(\hat{x}_i, \hat{y}_i)$  sono le coordinate dell' $i$ -esima posizione stimata;
- $(x_k, y_k)$  sono le coordinate note della  $k$ -esima posizione di test set.

L'errore medio totale è stato infine calcolato come:

$$\text{Average Error} = \left(\frac{1}{15}\right) \sum_{i=1}^{15} \text{Error}(i) \quad (39)$$

Nel grafico illustrato in Fig.50 sono riportati i risultati ottenuti con le tre tipologie di calibrazione.

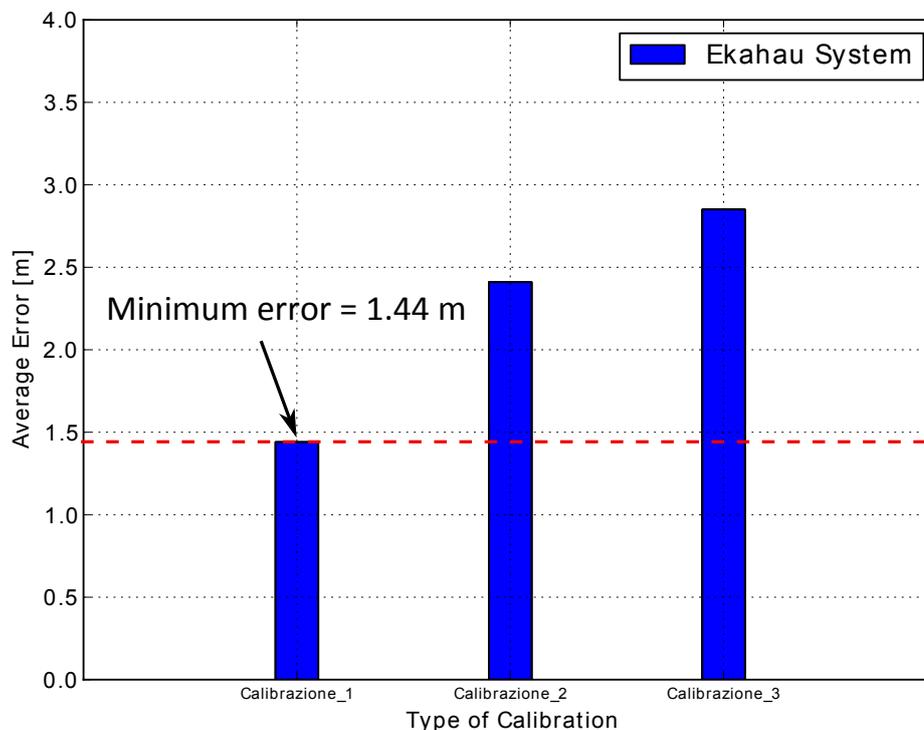


Figura 50: Errore medio del sistema Ekahau.

I risultati mostrano che con la Calibrazione 1 si riesce ad ottenere una localizzazione con un errore medio di **1,44 metri**.

Si può notare inoltre come diminuisce il valore dell'accuratezza del sistema al diminuire sia del numero di nodi della rete posti nella solita superficie di localizzazione, sia del tempo di calibrazione impiegato dall'utente per calibrare il sistema. Analizzando il grafico dei risultati si può notare che:

- rimuovendo un nodo della rete l'errore medio è aumentato di 1 metro (*Calibrazione\_2*);
- mantenendo lo stesso numero di nodi, eseguendo però una calibrazione meno pesante, l'accuratezza del sistema è diminuita di 1.46 metri (*Calibrazione\_3*).

## 13 Test del Sistema di localizzazione ZigBee

Per verificare le prestazioni del sistema implementato è stato utilizzato un test set costituito da trenta posizioni di riferimento per ogni calibrazione effettuata. I test sono stati eseguiti ponendo il valore della deviazione standard dello stimatore della densità di probabilità  $P(s/l)$  della relazione (30) pari a 0.8. A seguito di varie prove tale valore ha permesso di ottenere le prestazioni migliori in termini di accuratezza del sistema implementato.

### 13.1 Test 1

Nel primo test la calibrazione è stata effettuata creando una rete ZigBee costituita da quattro ancore e suddividendo l'ambiente di localizzazione in una griglia 1m x 1m Fig.51.

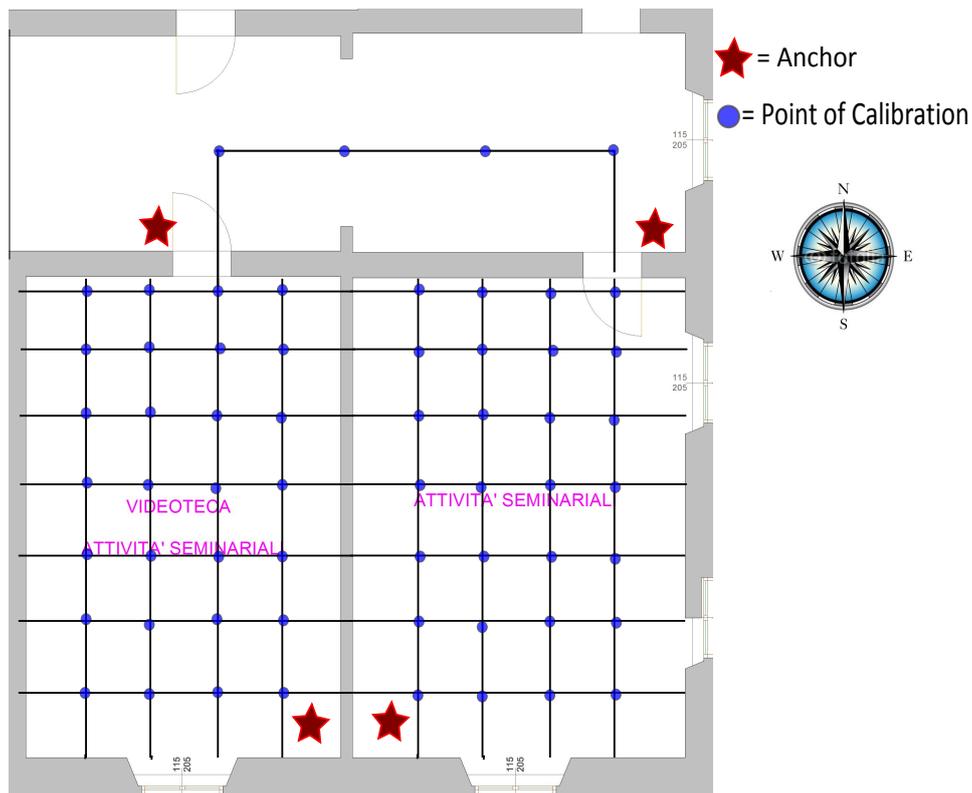


Figura 51: Prima Calibrazione del sistema ZigBee.

In figura i punti di calibrazione sono indicati con i cerchi blu mentre le ancore poste nell'ambiente di localizzazione sono individuate dalle stelle rosse. La calibrazione è stata eseguita effettuando dieci *RSSI Ping* in ogni posizione di calibrazione e orientando il calibration tag sempre nella medesima direzione (Nord). Il tempo di calibrazione è stato di 38 minuti.

Nel tentativo di verificare le massime prestazioni del sistema implementato anche il Location Node è stato orientato verso Nord in ogni posizione del test set. In Fig.53 e in Fig.53 sono illustrati i risultati in termini di errore medio ottenuti con i quattro metodi descritti nella sezione 10.2.

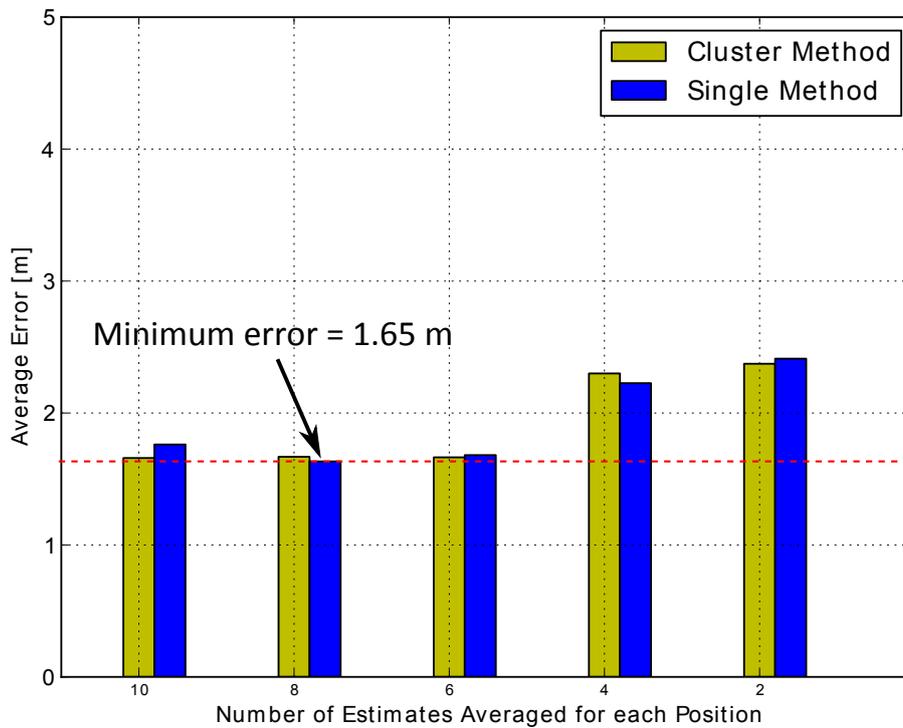


Figura 52: Errore medio degli algoritmi di localizzazione Cluster Method e Single Method.

In particolare viene mostrato l'andamento del valore dell'errore medio al variare del numero di stime di posizione mediate per fornire le coordinate del Location Node. I risultati ottenuti sono pressoché identici per entrambe le tecniche di stima e, come aspettato, l'accuratezza del sistema si riduce al diminuire del numero di stime mediate per ogni posizione. L'errore medio minimo (**1.65 metri**) è stato ottenuto con il metodo **Single Method** (vedi sez. 10.2) eseguendo otto stime per ogni posizione di di riferimento.

Occorre notare che il tempo impiegato dal Location Node per ottenere le informazioni necessarie ad eseguire le otto stime è di 87 s: un tempo così elevato impedisce di utilizzare il sistema in applicazioni in cui è necessaria una localizzazione in tempo reale.

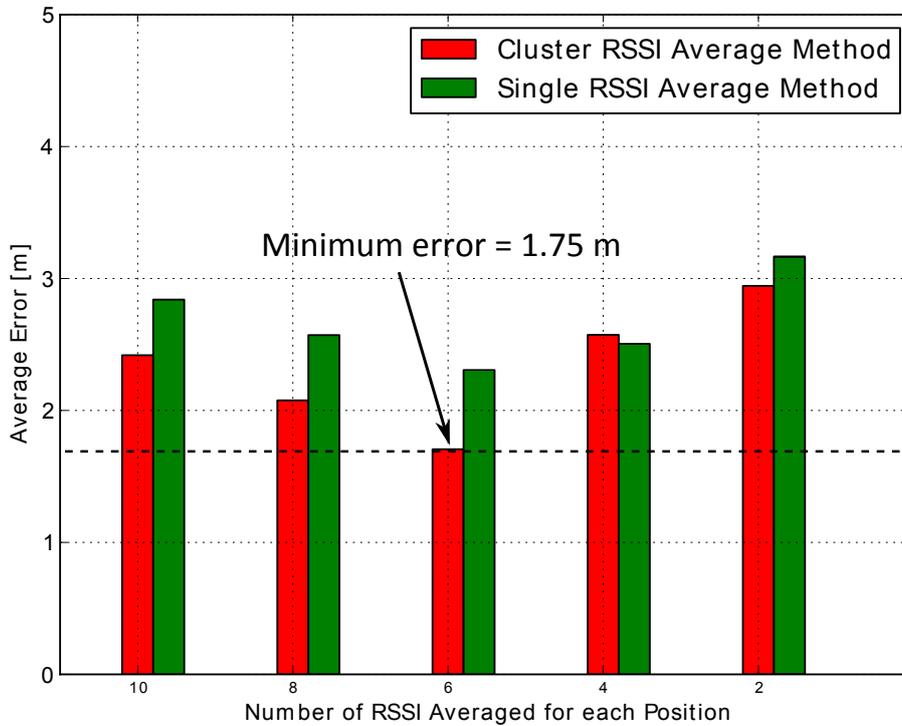


Figura 53: Errore medio degli algoritmi di localizzazione Cluster and Single RSSI Averaged Methods.

In Fig.53 sono mostrati i risultati in termini di errore medio al variare del numero di RSSI mediati per ogni posizione di test ottenuti con gli altri due metodi implementati (vedi sez.10.2). In figura è possibile osservare che l'andamento del valore dell'errore medio non dipende dal numero di RSSI mediati in ogni posizione, si ottiene infatti circa lo stesso errore sia con dieci che con due RSSI mediati. È inoltre evidente come l'accuratezza ottenuta con tali algoritmi sia peggiore rispetto a quella ottenuta con i metodi precedenti.

Dati i risultati del primo test si è deciso utilizzare nei test successivi solo le tecniche di stima **Single Method** e **Cluster Method**.

## 13.2 Test 2

Nel secondo test effettuato, la calibrazione è stata eseguita considerando solo i punti di calibrazione ai bordi delle stanze Fig.(54). Tale operazione è stata eseguita per verificare la possibilità di effettuare una calibrazione più rapida mantenendo una ragionevole accuratezza del sistema.

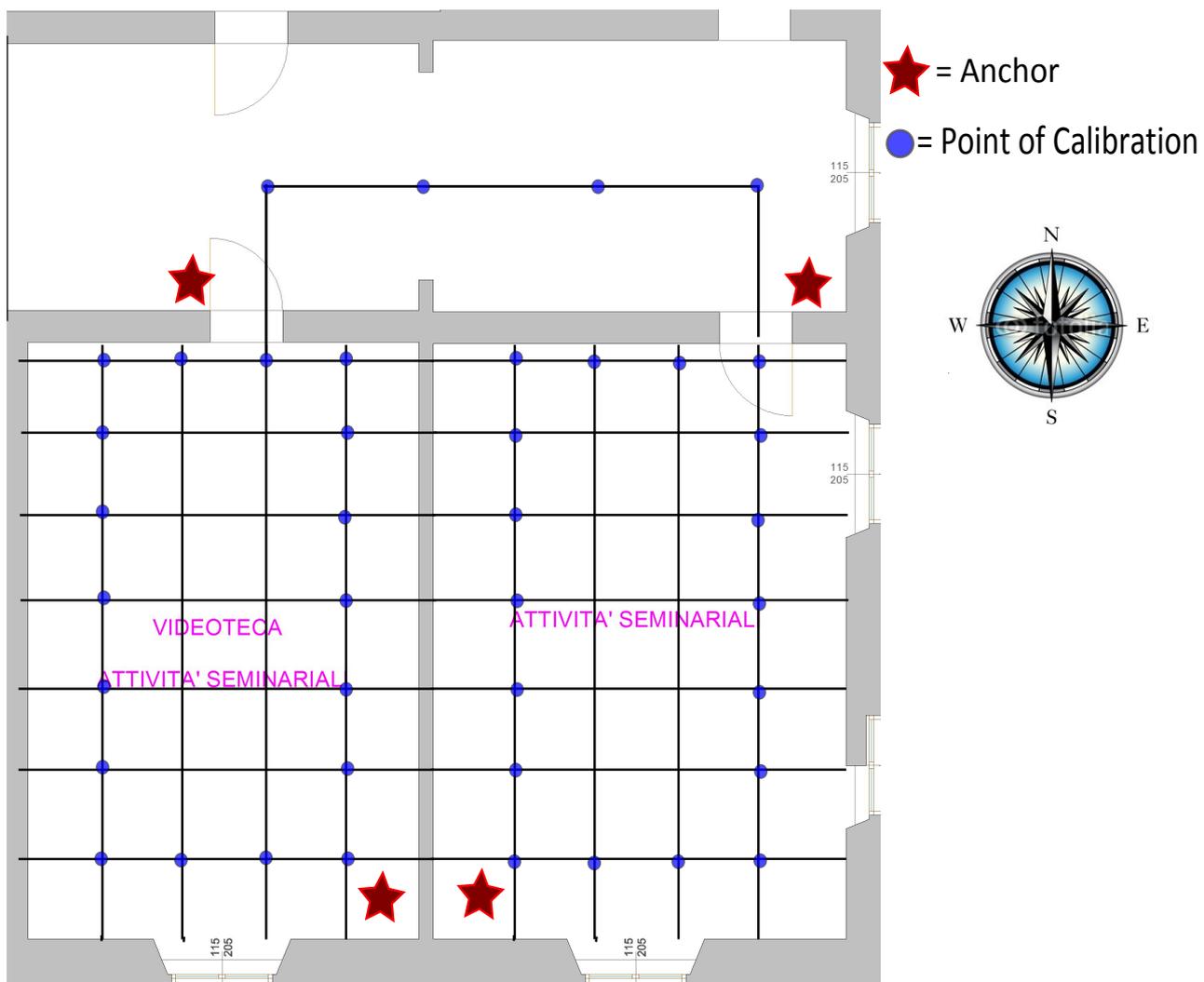


Figura 54: Seconda calibrazione del sistema ZigBee.

La calibrazione è stata eseguita effettuando dieci RSSI Ping e orientando il Calibration tag verso Nord in ogni posizione di calibrazione. Il tempo di calibrazione è stato di 24 minuti.

In Fig.55 sono illustrati i risultati ottenuti in termini d'errore medio. Come è possibile osservare, nonostante si siano dimezzate le posizione calibrazione, l'errore medio di localizzazione del sistema è aumentato solamente di 0.24 metri. Pertanto, nonostante una riduzione del tempo di calibrazione (circa la metà), l'accuratezza del sistema continua ad essere soddisfacente (**1.89 metri**).

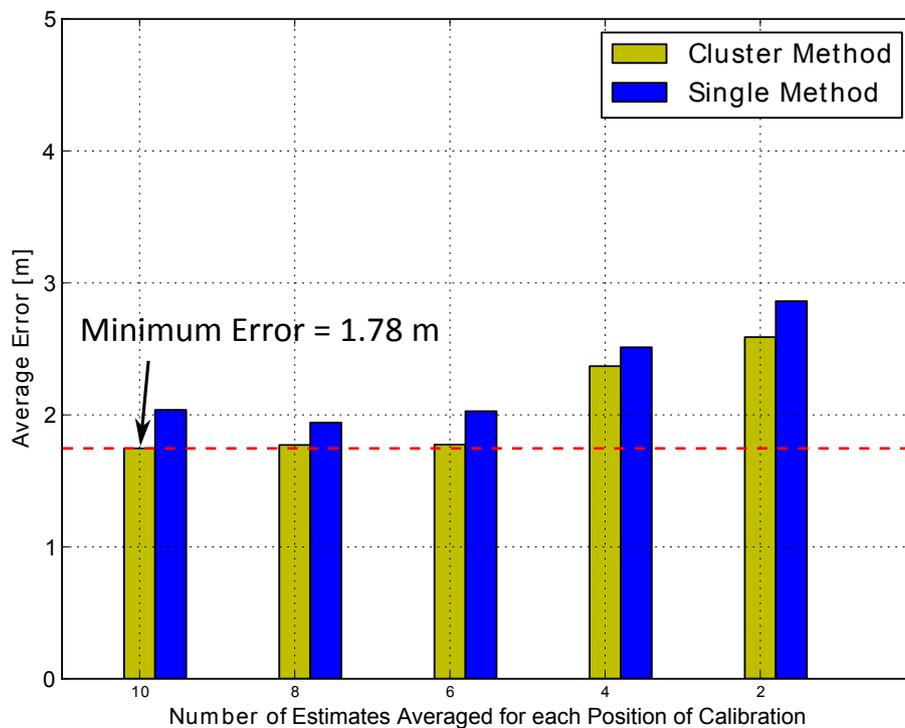


Figura 55: Errore medio degli algoritmi di localizzazione Cluster Method e Single Method.

### 13.3 Test 3

I test fin'ora effettuati, come si è visto, hanno presupposto di orientare le schede di calibrazione e localizzazione sempre allo stesso modo. In un'applicazione reale del sistema

implementato è impensabile ipotizzare che durante la localizzazione il Location Node sia sempre orientato come il Calibration tag nella fase di calibrazione. Dato che il Location Node è indossato ad esempio da persone che lavorano, durante la localizzazione il nodo sarà orientato in modo casuale.

Per verificare le prestazioni del sistema in condizioni realistiche è stato eseguito un Test nel modo seguente:

- il sistema è stato calibrato orientando il Calibration tag nelle posizioni di calibrazione come se una persona camminasse lungo il perimetro delle stanze (ad esempio come le frecce in Fig.49a. I punti di calibrazione sono stati i medesimi illustrati in Fig.54.
- Per valutare l'errore medio di calibrazione in ogni posizione del test set sono state fornite quattro stime della posizione del Location Node orientando la scheda in una direzione diversa per ogni stima eseguita. Tale operazione è stata effettuata per mostrare gli effetti provocati dai fenomeni di *shadowing* e *guadagno d'antenna* sull'accuratezza del sistema.

Dai risultati ottenuti, illustrati in Fig.56, è possibile concludere che il sistema implementato è robusto a variazioni di orientazione del Location Node rispetto il Calibration tag. Si evince infatti che l'accuratezza massima del sistema si riduce solo di 0.2 metri. Confrontando i risultati mostrati in Fig.56 è possibile osservare che la tecnica **Cluster Method** si è rivelata la migliore; il sistema implementato con tale metodo ha infatti subito una riduzione media dell'accuratezza inferiore rispetto il metodo **Single Method**.

Per effettuare un'analisi completa del sistema implementato è stato necessario verificare come si modifica il valore dell'accuratezza sia al variare del numero di ancore presenti nella medesima superficie di localizzazione sia al variare del numero di *RSSI Ping* effettuati in ogni posizione di calibrazione.

A tal fine sono stati utilizzati i file di calibrazione e di localizzazione del Test 3 andando ad analizzare l'accuratezza del sistema al variare dei parametri sopra citati. Il numero di stime mediate per fornire le coordinate del nodo in ogni posizione del test set è stato fissato a sei (Accuratezza = **1.95 m** nella figura precedente).

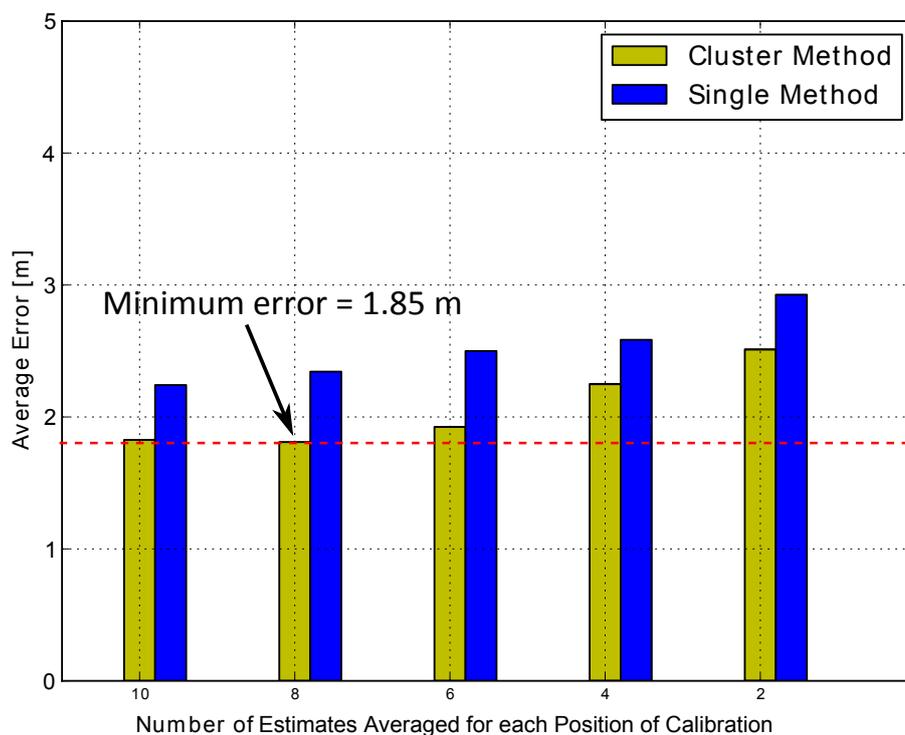


Figura 56: Errore medio degli algoritmi di localizzazione Cluster Method e Single Method.

Il primo test è stato eseguito diminuendo il numero di *RSSI Ping* effettuati in ogni posizione di Calibrazione. In Fig.57 si mostra i risultati ottenuti.

Come atteso, l'errore medio del sistema al diminuire del numero di *RSSI Ping* effettuati. Occorre notare che il valore dell'accuratezza si mantiene pressoché costante fino ad un numero di *RSSI Ping* pari a sei, mentre peggiora sensibilmente per valori inferiori (Accuratezza = **2.63 m** per due *RSSI Ping*). È dunque possibile ridurre ulteriormente il tempo di calibrazione mantenendo un valore di accuratezza accettabile. Eseguendo infatti il solito percorso di calibrazione illustrato in Fig.54 con sei *RSSI Ping* per ogni posizione invece che con dieci, il tempo di calibrazione si è ridotto a 14.5 minuti. Tale tempo è accettabile per un'area di 100 m<sup>2</sup>.

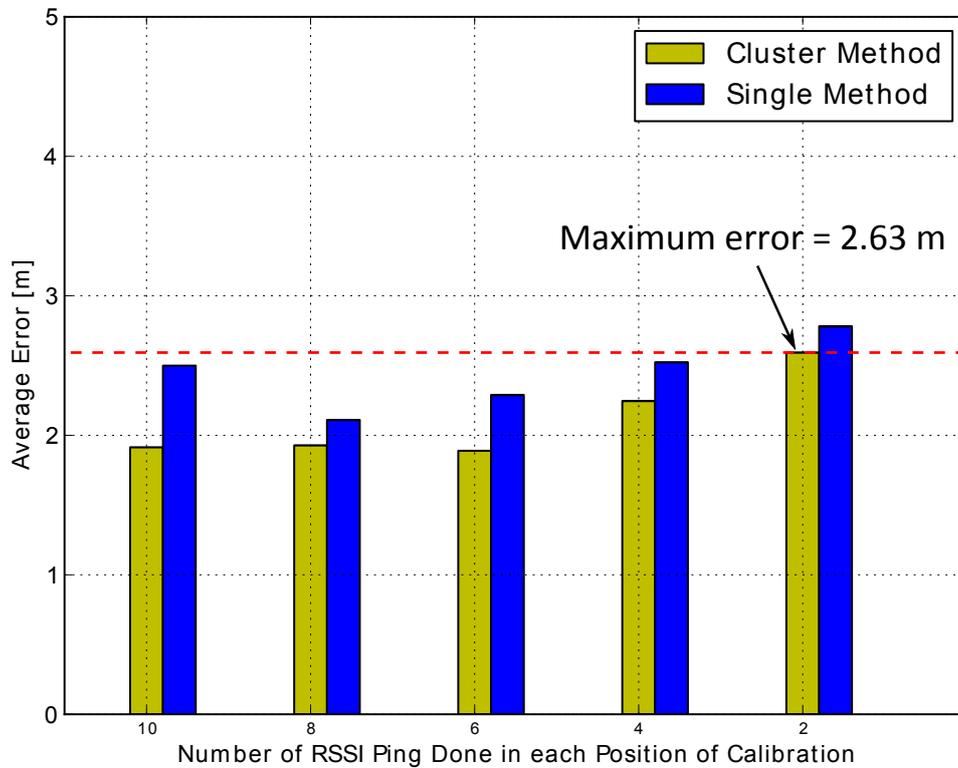


Figura 57: Errore medio, riduzione del numero di RSSI Ping.

Per quel che riguarda il test di verifica dell'andamento del valore dell'accuratezza in funzione del numero di ancore del sistema si è deciso di effettuare sia durante la calibrazione, sia durante la localizzazione un numero di *RSSI Ping* per ogni posizione pari a sei; i risultati ottenuti sono illustrati in Fig.58.

Facendo riferimento alla Fig.54, nel test le ancore sono state rimosse nel seguente ordine:

1. l'ancora posizionata in alto a destra;
2. quella posizionata in basso a sinistra;

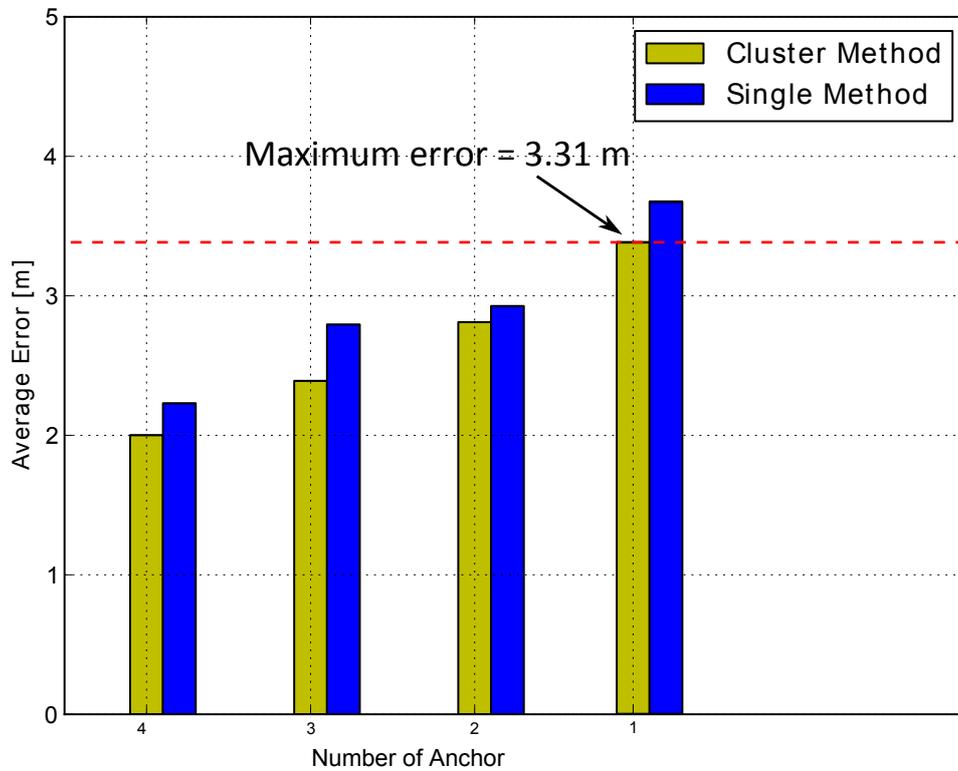


Figura 58: Errore medio, riduzione del numero di Ancore.

3. quella posizionata in alto a sinistra.

I risultati mostrano che anche con un'unica ancora si ottiene un'accuratezza di **3.31 m**.

In conclusione, dai test effettuati si evince che i migliori risultati in termini di accuratezza si ottengono utilizzando la tecnica di stima **Cluster Method**. Alla luce di quanto detto i confronti prestazionali con il sistema EkaHau verranno effettuati solamente col metodo **Cluster Method**.

## Parte VI

# CONCLUSIONI

## 14 Confronti Prestazionali e Sviluppi Futuri

### 14.1 Confronti Prestazionali

Per valutare la possibilità di sviluppare e commercializzare il sistema realizzato è opportuno effettuare un confronto prestazionale tra il sistema in questione e il sistema di Ekahau.

Per confrontare in modo coerente (parità di Ancore) i due sistemi analizzati, in termini di accuratezza e tempo di calibrazione, in Tabella 16 sono riportati sia i risultati del sistema di Ekahau ottenuti con la *Calibrazione\_1* (tre ancore) e con la *Calibrazione\_2* (due ancore), sia quelli del sistema ZigBee ottenuti per tre e due ancore (Fig.54).

	Calibration Time [s]	Accuracy 2 Anchor [m]	Accuracy 3 Anchor [m]
System of Ekahau (RTLS)	130	1.44	2.38
Our system ZigBee	864	2.85	2.23

Tabella 16: Confronti Prestazionali.

La differenza tra il tempo di calibrazione di Ekahau e quello del sistema implementato è elevata (un fattore otto in tabella): tale fatto può essere giustificato considerando che la tecnologia Wi-Fi è caratterizzata da un data rate in trasmissione (5.5 Mb/s) superiore di un fattore 20 a quello ZigBee (250 Kb/s). Tuttavia un tempo di calibrazione di circa 15 minuti ogni 100 m<sup>2</sup> di superficie è accettabile considerando che la fase di calibrazione deve essere effettuata un'unica volta. Occorre notare che il Calibration Tag impiega 3.6 s per trasmettere un *RSSI Ping* e registrare i valori di RSSI correlati. Tale valore impedisce di eseguire una calibrazione camminando perché : un uomo cammina mediamente alla velocità di 1 m/s pertanto si effettuerebbero solamente sei *RSSI Ping* ogni 3.6 m. Dai risultati illustrati nella sezione 14 si evince che sono necessari circa 12÷20 *RSSI Ping* ogni 3 m per ottenere prestazioni ragionevoli.

Le prestazioni del sistema implementato in termini di accuratezza si sono rivelate soddisfacenti; in quanto risultano inferiori a quelle di Ekahau di circa 0.5 metri. Tale differenza si mantiene approssimativamente costante in entrambe le calibrazioni effettuate e può es-

sere giustificata dal fatto che il sistema di Ekahau è un sistema affermato a cui lavorano un gran numero di persone da tempo.

La differenza sostanziale tra i due sistemi consiste nel tempo che trascorre tra due stime di posizione consecutive. Tale tempo, con la tecnologia ZigBee, è dell'ordine della decina di secondi (45 s con 6 *RSS Ping*) mentre, con il sistema Ekahau, è dell'ordine dei secondi. Tale fattore impedisce al sistema realizzato con la tecnologia ZigBee di essere utilizzato in applicazioni in cui il nodo è in continuo spostamento .

I risultati ottenuti mostrano la possibilità effettiva di realizzare un sistema di localizzazione commerciale per reti wireless costituite da nodi ZigBee.

## 14.2 Sviluppi Futuri

È possibile migliorare ulteriormente il sistema di localizzazione ZigBee implementato nel presente lavoro di tesi in modo da raggiungere molteplici obiettivi, tra cui:

- *migliorare la fase della calibrazione e la gestione dei file generati*: per rendere la fase della calibrazione più semplice si potrebbe realizzare un software, in linea con l'Ekahau Site Survey, con la possibilità di caricare e visualizzare la mappa dell'ambiente di localizzazione e di indicare i punti di calibrazione, durante tale fase, sulla stessa. Per gestire in modo più efficiente i file di calibrazione di grandi ambienti si potrebbe utilizzare un database relazionale (es. Mysql <http://www.mysql.it/>) per diminuire il tempo di elaborazione del motore di localizzazione;
- *facilitare la gestione del sistema da parte dell'utente*: si potrebbe ad esempio realizzare in un'interfaccia grafica *dell'Engine* nella quale l'utente può visualizzare sia gli identificativi di tutti i Location Node presenti nella rete, ed eventualmente inviare una *Send Ping* per richiedere la posizione di un nodo di interesse, sia una mappa per monitorare l'ambiente di localizzazione visualizzando la posizione dei Location Node presenti nella rete. *6 RSSI Ping*;
- *migliorare l'accuratezza del sistema*: a tal fine si potrebbe studiare sia la possibilità di eseguire un pre-processing (es. un filtraggio) ai dati di calibrazione, sia la possibilità di combinare in modo opportuno diverse tecniche di stima (metodi Bayesiani, Trilaterazione) in modo da migliorare l'accuratezza del sistema.

Il presente elaborato, attraverso i risultati dei numerosi test effettuati, ha dunque evidenziato come le caratteristiche del sistema di localizzazione implementato con la tecnologia ZigBee possano essere sviluppate e migliorate in modo da soddisfare le esigenze di eventuali applicazioni commerciali.

## Riferimenti bibliografici

- [1] Yun Zhou and Yuguang Fang, "Securing Wireless Sensor Networks", IEEE Communications Surveys & Tutorials, 2006.
- [2] Li Zheng, "ZigBee Wireless Sensor Networks In Industrial Applications", IEEE, International Joint Conference 2006.
- [3] Md Asdaque Hussian, Pervez Khan, Kwak Kyung Sup, "WSN Research Activities for Military Application", IEEE, 2009.
- [4] Sanchita Mal-Sarkar, Iftikhar U. Sikder and Vijay K. Konangi, "Application of Wireless Sensor Networks in Forest Fire", IEEE International Conference on Computer and Information Technology, 2010.
- [5] Oh-Heum Kwon, Ha-Joo Song and Sangjoon Park, "Anchor-Free Localization through Flip-Error-Resistant Map Stitching in Wireless Sensor Network", IEEE, 2010.
- [6] Roberto Maurizio Pellegrini, Samuela Persia, Diego Volponi and Giuseppe Marcone, "RF Propagation Analysis for ZigBee Sensor Network using RSSI Measurements", IEEE, 2011.
- [7] Hailiang Xiong, Hui Song, Zhihua Lai, Jie Zhang and Kechu Yi "A Novel Indoor Localization Scheme UWB", IEEE, 2010.
- [8] ChulYoung Park, DaeHeon Park, JangWoo Park YangSun Lee and Youngeun An, "Localization algorithm design and implementation to utilization RSSI and AOA of ZigBee", IEEE, 2010.
- [9] Andreas Savvides, Heemin Park and Mani B. Srivastava, "The Bits and Flops of the N-hop Multilateration Primitive For Node Localization Problems", 2002.
- [10] Ricardo Severino, Má rio Alves, "On a Test-bed Application for the ART-Wise Framework", Technical Report, 2006.
- [11] Silverman, B.W. "Density Estimation for Statistics and Data Analysis". Chapman and Hall, London, 1986.
- [12] Teemu Roos, Petri Myllymäki, Henry Tirri, Pauli Misikangas, and Juha Sievänen, "A Probabilistic Approach to WLAN User Location Estimation", International Journal of Wireless Information Networks, 2002.
- [13] David Madingan, Eiman Elnahrawy, Richard P.Martin, Wen-Hua Ju, P.Krishnan, and A.S. Krishnakumar, "Bayesian Indoor Positioning System", IEEE 2005.

- [14] Ekahau, "Ekahau Positioning Engine: 802.11-based Wireless LAN positioning system", EKahau Technology Document, 2002.
- [15] P.Bahl, V.N. Padmanabhan, and A. Balachandran, "Enhancements to the RADAR User Location Tracking System", Technica; Report MSR-TR-00-12, Microsoft Research, 2000.
- [16] ZigBee, "ZigBee Specification" (ZigBee Doc. 053474r17), 17 Gennaio 2008.
- [17] ZigBee, "Getting Started with ZigBee and IEEE 802.15.4", Diantree Networks, 2008.
- [18] ZigBee, "ZigBee Certification Frequently Asked Questions", FAQ's ZigBee Certification, 2006.
- [19] ZigBee, "ZigBee Alliance Official Site", <http://www.zigbee.org>.
- [20] Freescale, "1322x Network Node Reference Manual", Freescale 2010.
- [21] Freescale, "1322x Sensor Node Reference Manual", Freescale 2010.
- [22] Freescale, "1322x Low Power Node Reference Manual", Freescale 2010.
- [23] IAR system, "IAR J-Link and IAR J-Trace User Guide", IAR system 2008.
- [24] Freescale, "MC122x Reference Manual", Freescale 2010.
- [25] Freescale, "Beekit Wireless Connectivity Toolkit User's Guide", Freescale 2011.
- [26] Freescale, "Freescale BeeStack Application Development Guide for ZigBee 2007", Freescale 2011.
- [27] Freescale, "Freescale Test Tool", Freescale 2010.
- [28] Freescale, "BeeStack BlackBox ZigBee Test Client (ZTC)", Freescale 2011.
- [29] Freescale, "Freescale BeeStack Software Reference Manual for ZigBee 2007". Freescale 2008.
- [30] ZigBee Alliance, "ZigBee Cluster Library Doc. 075123r02ZB", ZigBee Alliance Maggio 2008.
- [31] ZigBee Alliance, "Telecom Applications Profile Specification", ZigBee Alliance Aprile 2010.
- [32] Freescale BeeStack, "Software Reference Manual for ZigBee 2007", Freescale 2007.
- [33] Freescale BeeStack, "MC1322x Simple Media Access Controller SMAC", Freescale 2011.

- [34] Freescale BeeStack, “*MC1323x Reference Manual*”, Freescale 2011.
- [35] Hui Liu, Student Member, IEEE, Houshang Darabi, Member, IEEE, Pat Banerjee, and Jing Liu, “*Survey of Wireless Indoor Positioning Techniques and Systems*”, IEEE TRANSACTIONS ON SYSTEM, MAN AND CYBERNETICS Novembre 2007.
- [36] Bahl and V. N. Padmanabhan, “*RADAR: An in-building RF-based user location and tracking system*,” in Proc. IEEE INFOCOM 2000, Mar., vol. 2, pp. 775–784.
- [37] P. Bahl and V. N. Padmanabhan, “*Enhancements to the RADAR user location and tracking system*,” Microsoft Corp., Tech. Rep. MSR-TR- 2000–12, Feb. 2000.
- [38] M. Youssef, A. Agrawala, and A. Udaya Shankar, “*WLAN location determination via clustering and probability distributions*,” IEEE Int. Conf. Pervasive Comput. Commun., Mar. 2003, pp. 143–151.
- [39] M. Youssef and A. K. Agrawala, “*Handling samples correlation in the Horus system*,” IEEE INFOCOM 2004, Hong Kong, vol. 2, pp. 1023– 1031, Mar. 2004.
- [40] M. Brunato and R. Battiti, “*Statistical learning theory for location fingerprinting in wireless LANs*”, Comput. Netw., vol. 47, pp. 825–845, 2005
- [41] R. Battiti, T. L. Nhat, and A. Villani, “*Location-aware computing: A neural network model for determining location in wireless LANs*,” Tech. Rep. DIT-02–0083, 2002.
- [42] Jeffrey Junfeng Pan, James T. Kwok, Qiang Yang, Senior Member IEEE, and Yiqiang Chen, “*Multidimensional Vector Regression for Accurate and Low-Cost Location Estimation in Pervasive Computing*”, IEEE Transactions on Knowledge and data Engineering Settembre 2006.
- [43] Freescale, “*Freescale Platform Reference Manual for ZigBee 2007*”, Freescale 2011 .
- [44] Freescale, “*Freescale BeeStack Application Development Guide for ZigBee 2007*”, Freescale 2011 .
- [45] Python, “*The Python Language Reference*”, Python 2012.
- [46] <http://pyserial.sourceforge.net/pyserial.html>
- [47] Numpy community, “NumPy User Guide Release 2.0.0.dev-55472ca”, 2012.

- [48] ChulYoung Park, DaeHeon Park, JangWoo Park, YangSun Lee, Youngeun An, "*Localization algorithm design and implementation to utilization RSSI and AOA ZigBee*", IEEE 2010.
- [49] Corinna Cortes, Vladimir Vapnik, "*Support Vector Networks*", Kluwer Academic Publishers 1995.
- [50] <http://www.ekahau.com/company/company-overview.html>
- [51] Ekahau, "*Ekahau Position Engine User Guide*".
- [52] Ekahau, "*Ekahau Site Survey User Guide*".
- [53] "*Special issue on Global Positioning System*", IEEE 1998
- [54] Wireless Personal Area Network Working Group, "*802.15.4 IEEE standard for the Telecommunications*", IEEE