

POLITECNICO DI TORINO

III Facoltà di Ingegneria dell'Informazione
Corso di Laurea Specialistica in Ingegneria Informatica

Tesi di Laurea Specialistica

Localizzazione in reti di sensori ZigBee

Basandosi sull'indice RSSI sia per ambienti interni che esterni



Relatore:

Prof. Giovanni Malnati

Candidato:

Alberto REALIS-LUC

ANNO ACCADEMICO 2008-2009

Sommario

La realizzazione di un sistema di localizzazione per reti di sensori ZigBee basato sul solo indice di RSSI permette di avere vantaggi in termini di costo dei dispositivi in quanto non è richiesto alcun hardware aggiuntivo per stimare la posizione dei nodi. L'indice RSSI è una stima della potenza ricevuta, ed è disponibile su ogni dispositivo ZigBee. Nel sistema vi sono tre tipi di dispositivi ZigBee: i nodi di riferimento, i nodi mobili ed il gateway.

I nodi di riferimento sono installati in posizioni note e si occupano di coprire l'intera area da monitorare, mentre i nodi mobili sono applicati ai beni da localizzare e sono liberi di muoversi, eseguono periodicamente misure di RSSI rispetto ai nodi di riferimento che vengono usate dal sistema per stimare la loro posizione. Il gateway si occupa di collegare un computer, tramite porta seriale, alla rete ZigBee. Su tale computer, il sistema prevede l'esecuzione di un servizio in grado di raccogliere ed interpretare tutte le informazioni ricevute dalla rete di sensori. Si è implementato nel servizio un'algoritmo di localizzazione che partendo dal calcolo di multilaterazione prova a migliorare le posizioni stimate dei nodi mobili facendo uso del metodo delle osservazioni indirette. Inoltre, il servizio mantiene aggiornato un database con tutte le informazioni dei nodi ed espone un'interfaccia utente tramite *web-application*, con cui l'utente può visualizzare le posizioni dei nodi, configurarli e gestire l'intero sistema. L'area monitorata dal sistema deve essere suddivisa in vari ambienti che possono essere ambienti aperti esterni oppure ambienti interni come le stanze di un edificio. Il sistema realizzato è in grado di localizzare un nodo mobile all'interno di un'edificio indicando la stanza in cui si trova, se si è installato almeno un nodo di riferimento in ogni stanza. Per ambienti più vasti od esterni il sistema è in grado di stimare le coordinate su due dimensioni, se si sono installati almeno tre nodi di riferimento che permettano di coprire interamente tale ambiente.

Il sistema ha quindi tutta una serie di vantaggi dovuta al fatto di utilizzare la sola interfaccia di rete ZigBee nella normale modalità per trasmettere dati: non è richiesto hardware aggiuntivo, non è necessario appoggiarsi al sistema GPS, il sistema può monitorare un'area composta sia da ambienti interni che esterni allo stesso tempo e si hanno anche vantaggi in termini di basso consumo di energia e di basso costo che sono caratteristiche proprie dei dispositivi ZigBee. La limitazione principale di questo sistema è la bassa precisione nella localizzazione per coordinate, ciò è dovuto al fatto che l'indice RSSI non dipende soltanto dalla distanza a cui si trova il trasmettitore ma risente fortemente anche delle riflessioni e dei disturbi presenti in ogni ambiente. Si può comunque affermare che il sistema è in grado di localizzare, con ragionevole approssimazione, oggetti di dimensioni comparabili con l'errore medio rilevato pari a 2-3 m, come ad esempio delle auto in un parcheggio.

Ringraziamenti

I ringraziamenti vanno a tutti coloro che hanno contribuito alla realizzazione di questo lavoro. Desidero innanzitutto ringraziare i tre enti che lo hanno reso possibile e cioè l'azienda *Laser s.r.l.*, l'istituto di ricerca del Politecnico di Torino: *Istituto Superiore Mario Boella* e la *Regione Autonoma Valle d'Aosta*.

Per l'istituto Boella, che ha fornito la base per avviare il progetto, desidero ringraziare in modo particolare il ricercatore Francesco Sottile che ha contribuito in modo significativo allo sviluppo del firmware sui moduli ZigBee.

Per quanto riguarda invece la *Laser* si ringraziano i colleghi Fabrizio Scarlatta, Fabrizio Falcicchio e Mauro De Biasi per gli innumerevoli passaggi verso le stazioni di Aosta, Châtillon e Pont Saint Martin.

Indice

Sommario	I
Ringraziamenti	II
1 Introduzione	1
2 Sistemi di localizzazione	3
2.1 Componenti base di un sistema di localizzazione	3
2.2 Grandezze misurabili per la localizzazione	3
2.3 Sistemi basati sul tempo	4
2.3.1 Time of Arrival	4
2.3.2 Time Difference of Arrival	5
2.4 Sistemi basati sull'intensità	6
2.4.1 Modello ad una riflessione sul terreno	9
2.5 Sistemi basati sull'interferometria	12
2.6 Sistemi basati sull'angolo di provenienza	13
2.7 Riepilogo	13
3 Algoritmi di localizzazione	14
3.1 Triangolazione	14
3.2 Trilaterazione	16
3.3 Multilaterazione	18
3.3.1 Multilaterazione atomica	18
3.3.2 Multilaterazione iterativa	20
3.3.3 Multilaterazione collaborativa	20
3.4 Min-max	22
3.5 Osservazioni indirette	23
3.5.1 Compensazione di reti	23
3.5.2 Metodo dei minimi quadrati applicato alle osservazioni indirette	23
4 Reti di sensori	27
4.1 Possibili applicazioni	27
4.2 Funzionamento	28

4.3	Protocolli di rete	28
4.4	Componenti delle reti WSN	29
4.5	ZigBee Alliance	29
4.6	Standard IEEE 802.15.4	30
4.6.1	Livello fisico	31
4.6.2	Componenti di una rete IEEE 802.15.4	32
4.6.3	Topologie di rete	32
4.6.4	Livello MAC	34
4.7	Protocollo ZigBee	37
4.7.1	Livello di rete	38
4.7.2	Livello applicazione	40
4.7.3	Stack Profile	40
4.7.4	Indirizzamento	41
4.7.5	Binding	44
4.7.6	Routing	44
4.7.7	Portabilità	46
4.7.8	Acknowledge	47
4.7.9	Scelta del canale	47
4.7.10	Configurazione del PAN ID	47
4.7.11	Memoria non volatile	47
4.7.12	Sicurezza	48
5	Il sistema realizzato	50
5.1	Requisiti del sistema	50
5.1.1	Obbiettivi	50
5.1.2	Specifiche funzionali	50
5.1.3	Specifiche hardware	54
5.1.4	Possibili applicazioni	60
5.2	Architetture	61
5.2.1	Firmware	61
5.2.2	Servizio	61
5.2.3	Database	62
5.3	Implementazione	63
5.3.1	Firmware	63
5.3.2	Servizio	73
5.3.3	Meccanismo di raccolta dati elementari	76
5.3.4	Traduzione dei dati elementari in coordinate geografiche	77
6	Analisi dei risultati sperimentali	80
6.1	Relazione tra RSSI e distanza in ambiente aperto	80
6.1.1	Descrizione dell'esperimento	80
6.1.2	Misure di RSSI rilevate	81
6.2	Verifica delle coordinate stimate dal sistema	83

6.2.1	Coordinate rilevate	84
6.3	Relazione tra RSSI e distanza in ambiente interno	85
6.3.1	Descrizione dell'esperimento	85
6.3.2	Misure di RSSI rilevate	85
7	Conclusioni	87
7.1	Considerazioni sull'RSSI	87
7.2	Considerazioni sui risultati ottenuti	88
7.3	Considerazioni sull'installazione del sistema	88
7.4	Miglioramenti futuri al sistema	89
	Bibliografia	90

Capitolo 1

Introduzione

Le reti di sensori ed in particolare lo standard ZigBee stanno riscuotendo un successo sempre più ampio in svariati ambiti, grazie a questi sensori che comunicano tra di loro tramite onde radio, è possibile monitorare lo stato di beni o di ambienti, a seconda di dove vengono posizionati. Lo scopo di questo lavoro è la realizzazione di un prototipo di un sistema di localizzazione per reti di sensori ZigBee. In svariati contesti, in cui si hanno sensori applicati a beni liberi di muoversi, può essere utile monitorare la posizione di tali moduli, inoltre sarebbe interessante avere questa possibilità facendo uso solamente dell'indice RSSI. L'RSSI (*Received Signal Strength Indicator*) è un'indicatore della potenza ricevuta, nelle reti di sensori viene rilevato ogni volta che si riceve un pacchetto. Risulta quindi particolarmente interessante la realizzazione di un sistema di localizzazione basato esclusivamente sull'RSSI poiché, a differenza di altri sistemi di localizzazione esistenti, non è necessario hardware aggiuntivo per la localizzazione (come sensori ad ultrasuoni o array di antenne), è sufficiente infatti la sola interfaccia di rete, peraltro già presente su tutti i moduli ZigBee. Questo permette di ottenere grandi vantaggi dalle caratteristiche proprie di ZigBee: basso costo dei dispositivi e basso consumo di energia degli stessi. Nel sistema realizzato vi sono due tipologie di moduli ZigBee: i nodi di riferimento e i nodi mobili. I nodi di riferimento permangono nella stessa posizione nota e vengono usati dai nodi mobili, liberi di muoversi, per fare misure di RSSI. Il sistema traduce le misure di RSSI in distanze e tramite algoritmi di multilaterazione fa una stima delle posizioni dei nodi mobili. L'indice RSSI è soggetto a forti variazioni dipendenti dall'ambiente in cui ci si trova quindi la stima della posizione che ne deriverà non potrà essere molto precisa, in questo progetto si vuole anche verificare sul campo tale precisione. Se ad esempio si determina un'errore medio di 3 m nella localizzazione si può dire che il sistema può essere usato per individuare la posizione di un'auto in un parcheggio ma non per individuare la posizione di oggetti più piccoli. In altri casi, se ci si trova all'interno di un'edificio, per localizzare un'oggetto può essere sufficiente sapere in quale stanza si trova, il prototipo realizzato implementa anche questa modalità di localizzazione. Infine occorre rendere facilmente accessibili tutti i dati: il sistema realizzato permette di gestire la rete di sensori e di visualizzare la posizione di tutti i nodi tramite una *web-application*. Il risultato finale di questo lavoro è un sistema di localizzazione funzionante, interamente accessibile e configurabile in remoto da *web-application*, capace di localizzare i sensori sia per stanze, in ambienti interni, sia con le coordinate, in ambienti esterni. In ambienti

aperti, l'errore medio riscontrato è di 2,5 m quindi si può dire che il sistema è in grado di localizzare, con ragionevole approssimazione, oggetti delle dimensioni di un'automobile. Nel secondo capitolo vengono illustrati i principi di funzionamento dei sistemi di localizzazione, vengono elencate le grandezze misurabili utili per il calcolo della posizione. Vengono poi messi a confronto alcuni sistemi di localizzazione esistenti. Nel capitolo 3 viene illustrato il funzionamento di vari algoritmi di localizzazione, viene quindi spiegato come stimare la posizione in modi diversi a seconda dei dati che si hanno a disposizione. Vengono inoltre riportati alcuni strumenti matematici che si sono rivelati particolarmente utili per costruire l'algoritmo di stima delle posizioni nel sistema realizzato. Nel capitolo 4 si descrivono più dettagliatamente le reti di sensori, viene data particolare enfasi allo *stack* ZigBee la cui conoscenza è stata fondamentale per la scrittura del firmware dei moduli. Nel capitolo 5 viene descritto il sistema realizzato, a partire dalle specifiche, si spiega il funzionamento del firmware in esecuzione sui moduli per poi descrivere le mansioni del servizio incaricato di raccogliere i dati, calcolare le posizioni e renderle accessibili tramite *web-application*. Infine negli ultimi due capitoli vengono presentati e commentati i risultati ottenuti dal prototipo realizzato.

Capitolo 2

Sistemi di localizzazione

2.1 Componenti base di un sistema di localizzazione

In ogni sistema di localizzazione si possono individuare due componenti fondamentali:

- **Punti mobili:** corrispondono agli oggetti che si vuole localizzare, la loro posizione non è quindi nota a priori, sono liberi di muoversi all'interno dell'area dove è stato installato il sistema di localizzazione.
- **Punti di riferimento:** vengono installati in posizioni note e vi restano stabilmente, vengono utilizzati dai punti mobili per calcolare le proprie posizioni. L'area in cui si vuole installare il sistema di localizzazione dovrà quindi essere interamente “coperta” da punti di riferimento.

2.2 Grandezze misurabili per la localizzazione

Per poter localizzare e quindi stimare la posizione di oggetto mobile occorre raccogliere alcune misure rispetto ad altri punti di riferimento le cui posizioni sono note a priori. Tali misure sono volte a scoprire le distanze dei punti di riferimento oppure le direzioni e quindi gli angoli in cui si trovano rispetto al nostro punto mobile di cui si desidera stimare la posizione. Un sistema di localizzazione può misurare le seguenti grandezze per stimare la posizione:

- Tempo di viaggio
- Intensità
- Fase (interferometria)
- Angolo di provenienza

2.3 Sistemi basati sul tempo

Se si riesce a misurare il tempo impiegato da un segnale per propagarsi da una stazione trasmittente ad una stazione ricevente è possibile stimare la distanza tra le due stazioni. Le onde radio nell'aria si propagano con una velocità di poco inferiore a quella della luce ($c = 299792468$ m/s) e dato che la luce impiega 3,3 ns per percorrere un metro per ottenere una precisione di 30 cm bisognerebbe avere un sistema in grado di rilevare intervalli di tempo fino a un nanosecondo. E' di questo tipo il sistema GPS (che però ha a che fare con distanze molto più elevate), non può essere usato in ambienti indoor in quanto i segnali provenienti dai satelliti non riescono ad oltrepassare i muri degli edifici. Una tale precisione è disponibile al momento sui dispositivi UWB (*Ultra Wide Band*). I sistemi UWB sono i più precisi ma anche i più costosi. Invece nel sistema *Cricket* sviluppato da *Crossbow*, per fare le misure, si è deciso di usare segnali più lenti rispetto alle onde radio e cioè gli ultrasuoni. Pur essendo più economico rispetto ai sistemi UWB, *Cricket* presenta problemi sulle lunghe distanze proprio per il fatto di usare gli ultrasuoni e questo si traduce nella necessità di installare una “maglia” di nodi di riferimento più fitta.



Figura 2.1. Sensore UWB utilizzato come punto di riferimento. *Ubisense* serie 7000.

2.3.1 Time of Arrival

Con *Time of Arrival* (ToA) detto anche *Time of Flight* (ToF) si intende il tempo impiegato da un segnale radio trasmesso da un singolo trasmettitore per raggiungere un singolo ricevitore remoto. Tenendo conto del dielettrico presente tra trasmettitore e ricevitore e conoscendo la velocità di propagazione del segnale (che sarà vicina a quella della luce), si potrà stimare la distanza tra trasmettitore e ricevitore. Nel momento in cui si riescono a misurare tempistiche molto piccole questo metodo si dimostra molto accurato. Dato che si ha a che fare con la velocità della luce si otterranno risultati migliori misurando distanze elevate. Ovviamente la sincronizzazione tra le due stazioni radio è di fondamentale importanza, la stazione ricevente misurerà il tempo con un suo orologio interno che dovrà essere perfettamente sincronizzato con

quello della stazione trasmittente. Queste misure vengono utilizzate dagli algoritmi di trilaterazione che facendo uso delle distanze tra una stazione radio mobile, di posizione imprecisata, e almeno altre tre stazioni radio fisse, di posizione nota, permettono di calcolare la posizione della stazione mobile su di un piano.

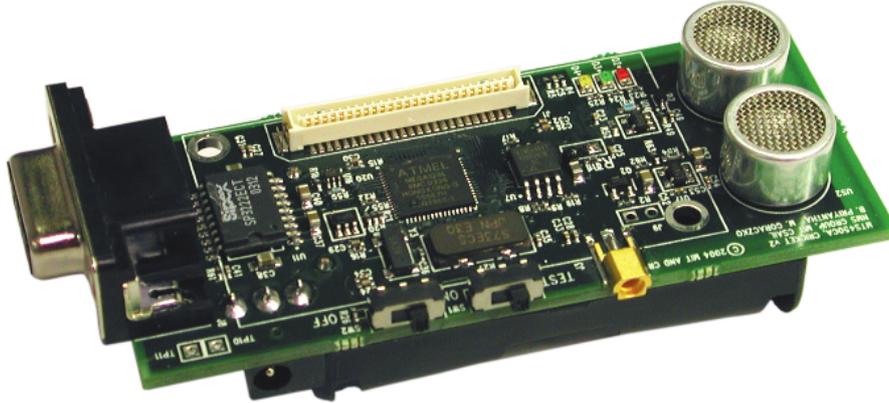


Figura 2.2. Dispositivo per la localizzazione Cricket MCS410CA. Si possono notare sulla destra il trasmettitore e il ricevitore di ultrasuoni.

2.3.2 Time Difference of Arrival

E' una tecnica per misurare il tempo di propagazione di un segnale quando si hanno a disposizione dispositivi in grado di trasmettere segnali con velocità di propagazione differenti. Un nodo che può trasmettere sia onde radio (RF) che ultrasuoni (US) trasmette prima un segnale radio e immediatamente dopo un segnale ad ultrasuoni. Dato che il suono viaggia molto più lentamente rispetto alle onde radio i nodi nelle vicinanze si vedranno arrivare prima il segnale RF e solo dopo un certo tempo vedranno arrivare anche il segnale US. Considerando il segnale RF istantaneo si può dedurre da questa differenza di tempo, che sarebbe il tempo di propagazione del segnale US, la distanza a cui si trova il nodo trasmittente.

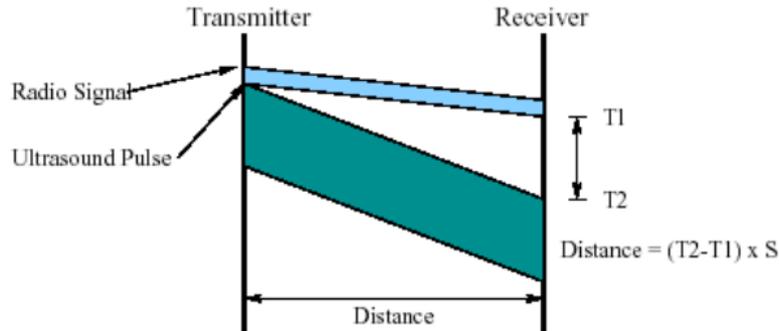


Figura 2.3. Diagramma a traliccio che illustra il funzionamento della tecnica TDoA (*Time Difference of Arrival*)

2.4 Sistemi basati sull'intensità

Valutando l'intensità di un segnale ricevuto è possibile stimare la distanza della stazione che l'ha trasmesso. Nel caso delle reti di sensori e nelle reti WiFi si usa l'indice RSSI (*Received Signal Strength Indicator*) che indica una stima della potenza del segnale ricevuto in dBm. Per correlare la potenza di un segnale ricevuto alla distanza dalla quale è stato trasmesso si può usare l'equazione di Friis:

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

Dove:

- P_R : è la potenza all'antenna ricevente espressa in Watt
- P_T : è la potenza all'antenna trasmittente espressa in Watt
- G_T : è il guadagno dell'antenna trasmittente
- G_R : è il guadagno dell'antenna ricevente
- λ : è la lunghezza d'onda: $\lambda = c/f$ dove c è la velocità della luce (299792458 m/s) e f è la frequenza.
- d : distanza in metri
- n : è una costante di propagazione del segnale, detto anche esponente di propagazione e dipende dall'ambiente in cui ci si trova

Dato che l'indice RSSI è solitamente espresso in dBm per convertire una potenza da Watt a dBm si usa la seguente formula:

$$P[\text{dBm}] = 10 \cdot \log_{10}(P[\text{W}] \cdot 10^3)$$

Ambiente	n
Free space	2,0
Retail store	2,2
Grocery store	1,8
Office, hard partitions	3,0
Office, soft partitions	2,6
Metalworking factory, line of sight	1,6
Metalworking factory, obstructed line of sight	3,3

Tabella 2.1. Alcuni esempi dei valori che potrebbe assumere n in alcuni ambienti

Nel caso dei dispositivi ZigBee ChipCon 2430 utilizzati nel sistema descritto nei capitoli successivi, si ha una potenza d'uscita pari a 1 mW (0 dBm), i guadagni d'antenna si possono considerare uguali a 1 e la frequenza si può considerare pari a 2441,75 MHz (a metà della banda dei 2,4 GHz) immaginando quindi di voler calcolare la potenza ricevuta a 100 m di distanza in ambiente aperto ($n = 2,0$) si ha:

$$P_R = P_T \frac{G_T G_R \lambda^2}{(4\pi)^2 d^n} = 1 \cdot 10^{-3} \cdot \frac{1 \cdot 1 \cdot \left(\frac{299792458}{2441,75 \cdot 10^6} \right)^2}{(4\pi)^2 \cdot 100^2} = 9,546 \cdot 10^{-12} \text{ W} = -80,2 \text{ dBm}$$

Con la stessa formula ad un metro di distanza si ottiene -40,2 dBm.

La stessa funzione la si ottiene più semplicemente con questa formula:

$$RSSI = -(10n \log_{10} d - A)$$

Dove A è la potenza del segnale ricevuto, in dBm, alla distanza di riferimento di un metro. Dalla formule precedenti è quindi possibile ricavare la distanza a cui si trova la stazione che ha trasmesso un segnale con un dato RSSI. Purtroppo l'indice RSSI non è molto affidabile, infatti è molto influenzato dall'ambiente in cui ci si trova, risente fortemente delle onde riflesse e dei disturbi; quindi le misure di distanze che ne deriveranno non saranno accurate.

Quando il sistema riesce a misurare le distanze di una stazione mobile rispetto ad almeno tre stazioni di riferimento può provare a stimarne la posizione con algoritmi di trilaterazione o multilaterazione. Ma data la forte varianza di tali misure di distanza, in alcuni sistemi, la posizione della stazione mobile viene considerata uguale alla posizione della stazione di riferimento che viene rilevata più vicina. Per realizzare sistemi di questo tipo si usano reti di sensori *ZigBee* che hanno il principale vantaggio in un basso costo oppure reti WiFi, che pur avendo un costo superiore alle reti di sensori, si possono appoggiare a infrastrutture già esistenti (come una serie di *access point* da usare come stazioni di riferimento).

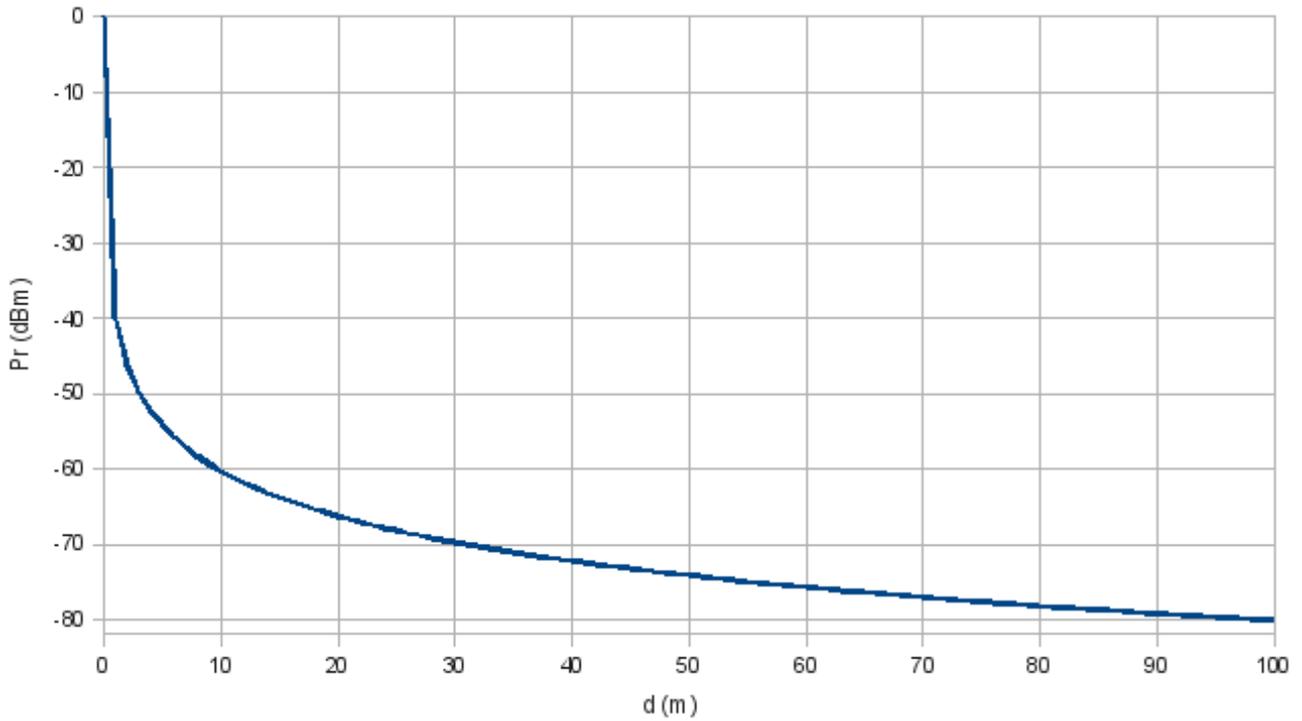


Figura 2.4. Andamento ideale della potenza ricevuta con l'aumentare della distanza in ambiente aperto. Grafico disegnato applicando la formula di Friis con le specifiche dei moduli ZigBee CC2430.



Figura 2.5. Moduli ZigBee *ChipCon* 2430 prodotti da *Texas Instruments*. Possono essere programmati sia per essere punti di riferimento sia per essere punti mobili in un sistema di localizzazione basato sull'RSSI.



Figura 2.6. Dispositivi commerciali che fanno uso di reti WiFi. A sinistra l'*Ekahau* T201, a destra un modello *AeroScout* a forma di badge.

2.4.1 Modello ad una riflessione sul terreno

In un qualunque collegamento radio reale le onde vengono riflesse e ostruite da tutti gli oggetti illuminati dall'antenna trasmittente. Calcolare il raggio di trasmissione in un ambiente reale può risultare molto complesso e sarebbero necessarie grandi risorse in termini di potenza di calcolo. Inoltre in alcuni ambienti potrebbero esserci degli oggetti in movimento o delle interferenze radio che andrebbero solo ad aumentare la complessità del sistema. Dato che nella pratica è molto difficile tener conto di tutte le riflessioni e delle interferenze (e inoltre per una data misura di RSSI si potrebbero avere più distanze corrispondenti) di solito ci si accontenta di usare la formula: $RSSI = -(10n \log_{10} d - A)$ opportunamente calibrata scegliendo i parametri A ed n in base all'ambiente in cui ci si trova. E' comunque interessante costruire un modello per il caso reale più semplice possibile: due stazioni radio ad una certa altezza dal suolo, in campo aperto senza nessun oggetto nei paraggi, tale situazione potrebbe per esempio verificarsi in mezzo ad un campo da calcio. In questo caso si avrebbe una sola riflessione sul suolo tra le due stazioni radio.

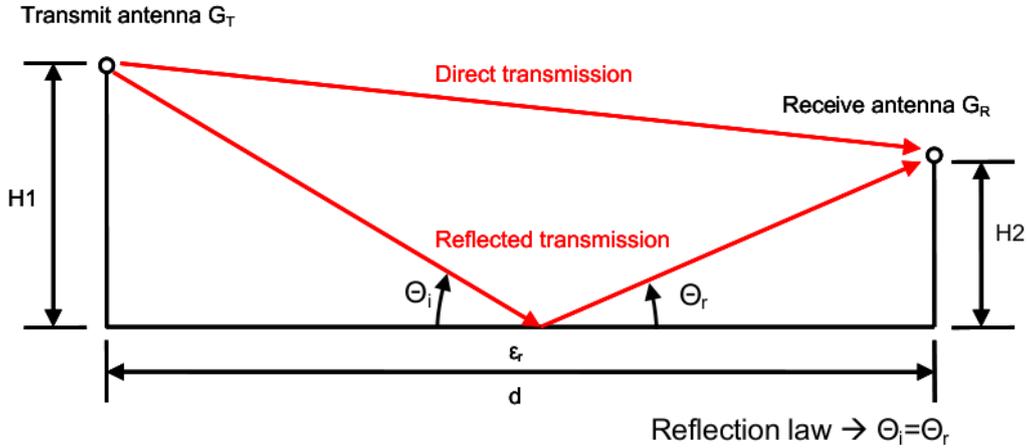


Figura 2.7. Il caso più semplice di riflessione, vi è una sola possibile riflessione nel punto in cui gli angoli θ sono uguali.

Le due onde che giungono al ricevitore possono sommarsi costruttivamente o distruttivamente a seconda della loro differenza di fase. L'ampiezza e la fase dell'onda diretta variano con la distanza percorsa. L'ampiezza dell'onda riflessa dipende dalla distanza totale percorsa e dal coefficiente di riflessione (Γ) relativo all'onda prima e dopo la riflessione.

Coefficiente di riflessione

Quando un'onda radio incidente colpisce una giunzione tra due differenti dielettrici, una porzione della sua energia viene riflessa, mentre l'energia restante oltrepassa la giunzione. La porzione riflessa dipende dalla polarizzazione del segnale, dall'angolo di incidenza, e dai dielettrici ($\epsilon_r, \mu_r, \sigma$). Assumendo che entrambe le sostanze abbiano una permeabilità $\mu_r = 1$ e che uno dei due dielettrici sia spazio libero si possono calcolare Γ_v e Γ_h che sarebbero i coefficienti di riflessione, rispettivamente per una polarizzazione verticale e orizzontale, calcolati con le equazioni di Fresnel:

$$\Gamma_v = \frac{\epsilon_r \sin \theta_i - \sqrt{\epsilon_r - \cos^2(\theta_i)}}{\epsilon_r \sin \theta_i + \sqrt{\epsilon_r - \cos^2(\theta_i)}}$$

$$\Gamma_h = \frac{\sin \theta_i - \sqrt{\epsilon_r - \cos^2(\theta_i)}}{\sin \theta_i + \sqrt{\epsilon_r - \cos^2(\theta_i)}}$$

Sapendo che la due stazioni radio si trovano alle altezze h_1 e h_2 dal suolo e conoscendo la distanza d tra i due pali che le sostengono, l'angolo di incidenza al terreno θ si può calcolare con la seguente formula:

$$\theta = \tan^{-1} \left(\frac{h_1 + h_2}{d} \right)$$

La distanza percorsa dall'onda diretta sarà:

$$d_d = \sqrt{|h_1 - h_2|^2 + d^2}$$

Mentre quella percorsa dall'onda riflessa:

$$d_r = \sqrt{(h_1 + h_2)^2 + d^2}$$

La differenza di fase è:

$$\Delta\phi = \frac{2\pi(d_r - d_d)}{\lambda}$$

Quindi immaginando di usare una polarizzazione verticale la potenza totale ricevuta da entrambe le onde sarà data da:

$$P_R = P_T \frac{G_T G_R \lambda^2}{(4\pi)^2 d_d^n} + \cos(\Delta\phi) \cdot P_T \frac{G_T G_R \lambda^2}{(4\pi)^2 d_r^n} \Gamma_v$$

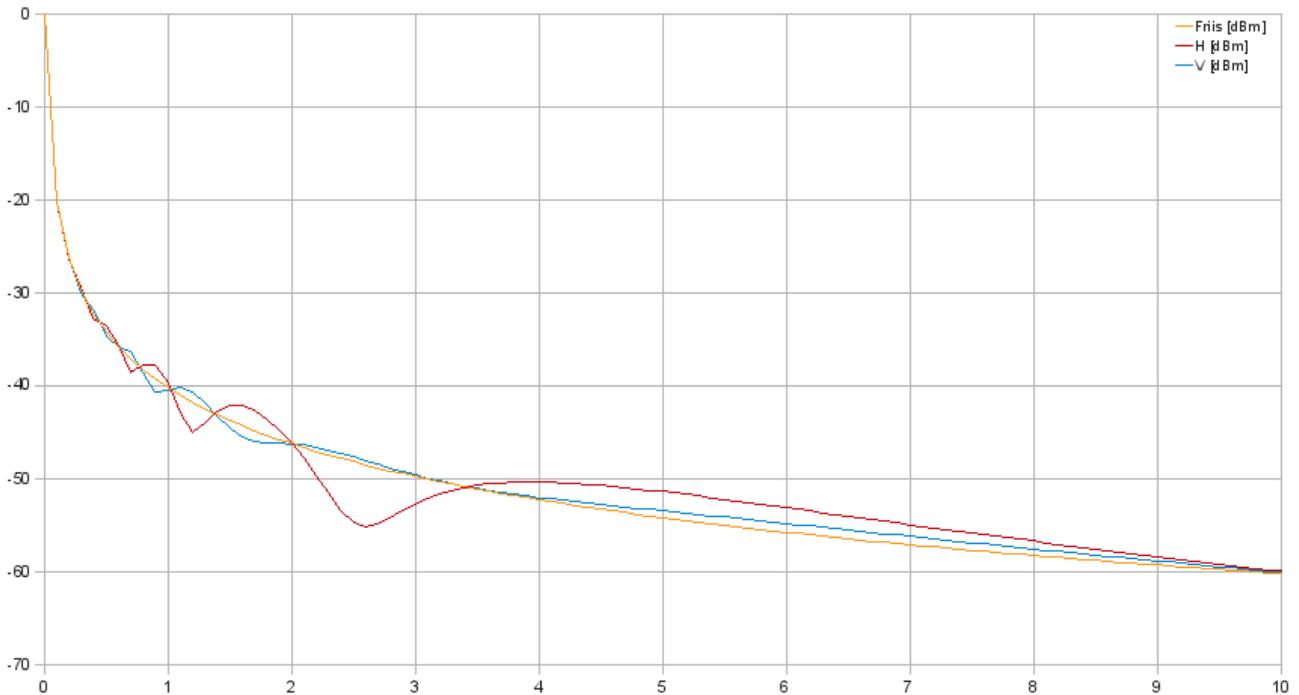


Figura 2.8. La funzione di Friis messa a confronto con i modelli ad una riflessione sul suolo con polarizzazione verticale e orizzontale. Per disegnare le curve si è immaginato che le due stazioni radio fossero entrambe ad un'altezza di 40 cm da terra, operanti ad una frequenza di 2441,75 MHz con $\epsilon = 18$. Come si può notare la polarizzazione orizzontale risente molto di più della riflessione.

2.5 Sistemi basati sull'interferometria

Se si hanno due antenne separate da una distanza d , su cui le onde radio incidono con un angolo θ si ottiene che il segnale deve percorrere distanze diverse per raggiungere entrambe le antenne. Questa differenza si traduce in una differenza di fase $\Delta\Phi$ del segnale ricevuto tra le due antenne. Con questa formula si può calcolare la direzione di arrivo del segnale:

$$\theta = \sin^{-1} \left(\frac{\lambda \Delta\Phi}{2\pi d} \right)$$

Dove λ è la lunghezza d'onda del segnale.

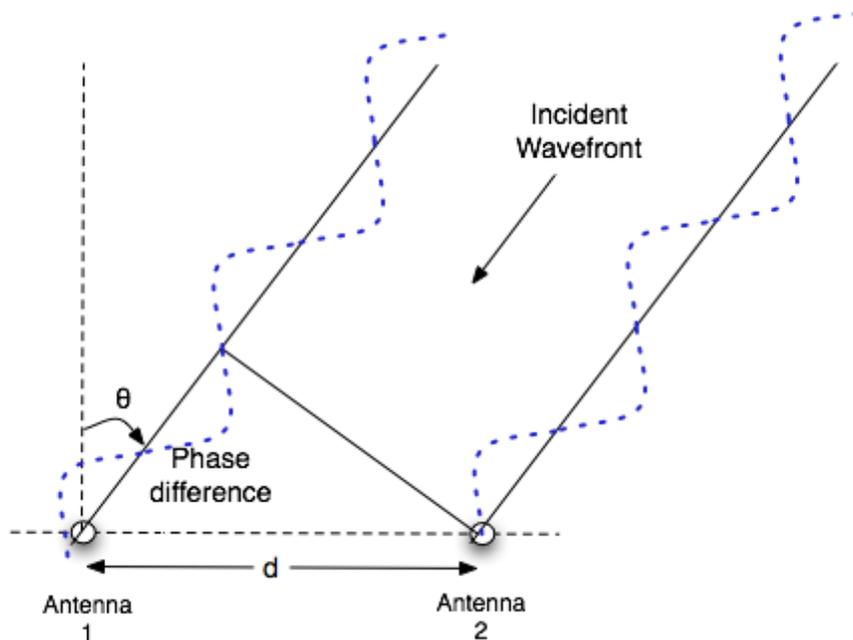


Figura 2.9. Principio dell'interferometria di fase

E' su questo principio che si basano alcuni sistemi di localizzazione, come per esempio RIPS (*Radio Interferometric Positioning System*), essi permettono di ottenere una buona precisione ma possono insorgere problemi nel caso il segnale giunga a destinazione da più percorsi attraverso varie riflessioni. Infatti nel caso in cui il segnale viene ricevuto da percorsi multipli (*multipath*) può risultare difficile capire quale sia la giusta differenza di fase su cui fare i calcoli. In un ambiente indoor si hanno chiaramente molte riflessioni e il sistema, può quindi trovarsi di fronte a problemi computazionalmente molto pesanti non così immediati da risolvere.

2.6 Sistemi basati sull'angolo di provenienza

Per rilevare l'angolo di provenienza occorre un'antenna direzionale oppure un'array di antenne, che, come si è visto prima, grazie all'interferometria di fase, permette calcolare l'angolo di provenienza. Questi sistemi, detti anche AoA (*Angle of Arrival*), richiedono antenne più complesse o la capacità di rilevare piccole differenze di fase, caratteristiche solitamente non presenti sui moduli più comuni delle reti di sensori. Conoscendo gli angoli di provenienza dei segnali trasmessi dai punti di riferimento, è possibile calcolare la posizione del punto mobile con algoritmi di triangolazione.

2.7 Riepilogo

Sistema	Principio	Precisione	Costo utente	Costo infrastruttura	Criticità
Reti ZigBee	Onde radio RSSI	1-5 m	basso	basso	precisione
Reti WiFi	Onde radio RSSI	1-5 m	basso	medio	precisione, costo
UWB	Onde radio ToA	30 cm	basso	alto	costo
Cricket	Ultrasuoni TDoA	50 cm	basso	medio	costo
RIPS	Interferometria	20-50 cm	basso	basso	multipath

Tabella 2.2. Tabella riassuntiva dei sistemi di localizzazione indoor citati precedentemente

Come si può notare i sistemi che fanno uso dell'RSSI non richiedono hardware aggiuntivo, oltre quello necessario per instaurare una rete senza fili, e quindi tendono ad avere un costo più basso. Tuttavia gli altri sistemi possono risultare molto più precisi ed affidabili. Per costo dell'infrastruttura si intende il costo necessario per coprire l'area dove si vuole installare il sistema di localizzazione con i moduli che faranno da punti di riferimento. I punti di riferimento dovranno essere installati con una densità compatibile con la loro portata. Invece per costo utente si intende il costo dei singoli moduli da usare come punti mobili che dovranno essere localizzati dal sistema.

Capitolo 3

Algoritmi di localizzazione

3.1 Triangolazione

Con triangolazione si intende il processo per determinare la posizione di un punto misurando gli angoli che puntano verso di esso da due punti di posizione nota agli estremi di una linea che sarebbe la base del triangolo così formato.

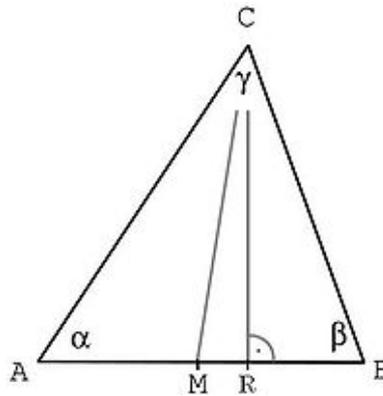


Figura 3.1. Un triangolo su cui viene applicata la triangolazione

La distanza AB tra i due angoli noti α e β si può calcolare nel modo seguente:

$$AB = \frac{CR}{\tan \alpha} + \frac{CR}{\tan \beta}$$

Dove CR è la distanza tra il terzo vertice del triangolo e la linea che unisce i vertici A e B , in pratica è l'altezza del triangolo, quindi per calcolare tale distanza:

$$CR = \frac{AB}{\frac{1}{\tan \alpha} + \frac{1}{\tan \beta}}$$

Tornano utili anche le classiche leggi della trigonometria:

$$\frac{\sin \alpha}{BC} = \frac{\sin \beta}{AC} = \frac{\sin \gamma}{AB}$$

Dato che la distanza AB è nota possiamo ora calcolare AC e BC :

$$AC = \frac{AB \sin \beta}{\sin \gamma}$$

$$BC = \frac{AB \sin \alpha}{\sin \gamma}$$

L'altezza RC può essere calcolata anche grazie ai seni degli angoli α e β :

$$RC = BC \sin \alpha = AC \sin \beta$$

Per evitare di calcolare prima AC e BC :

$$RC = \frac{AB \sin \alpha \sin \beta}{\sin \gamma} = \frac{AB \sin \alpha \sin \beta}{\sin (\alpha + \beta)}$$

Gli algoritmi di triangolazione fanno ampio uso di queste regole della trigonometria, per poi tradurre il risultato in coordinate valide rispetto al sistema di riferimento adottato dal sistema di localizzazione in uso.

3.2 Trilaterazione

La trilaterazione è un metodo per trovare l'intersezione della superficie di tre sfere i cui centri e raggi sono noti. Per quanto riguarda i sistemi di localizzazione i centri delle sfere sono dati dalle coordinate dei punti di riferimento, mentre i raggi sono le distanze rilevate dal punto mobile rispetto ai punti di riferimento nelle vicinanze. L'intersezione delle sfere individua quindi la posizione del punto mobile. In un sistema di riferimento installato su di una superficie piana, come per esempio un piano di un edificio, eliminando l'asse z , il problema si semplifica nel trovare l'intersezione di tre circonferenze. Assumiamo che le tre sfere abbiano i centri sul piano $z = 0$ e trasliamo l'origine del nostro sistema di riferimento nel centro della prima circonferenza.

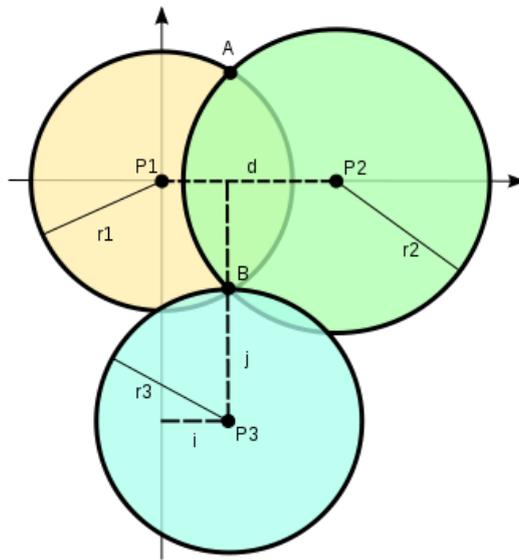


Figura 3.2. Il piano $z = 0$ mostra i centri delle tre sfere: $P1$, $P2$ e $P3$ di raggio: $r1$, $r2$ e $r3$; e le distanze: d , i , e j . B è il punto di intersezione delle tre sfere.

Possiamo quindi iniziare con il mettere a sistema le tre equazioni delle tre sfere:

$$\begin{cases} r_1^2 = x^2 + y^2 + z^2 \\ r_2^2 = (x - d)^2 + y^2 + z^2 \\ r_3^2 = (x - i)^2 + (y - j)^2 + z^2 \end{cases}$$

Sottraendo la seconda equazione dalla prima e risolvendola in x si ottiene:

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

Sostituendo nell'equazione della prima sfera si ottiene l'equazione di una circonferenza che sarebbe l'intersezione tra le prime due sfere:

$$y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{4d^2}$$

Sostituendo $y^2 + z^2 = r_1^2 - x^2$ nell'equazione della terza sfera e risolvendo in y si ottiene:

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x$$

Ora che abbiamo le coordinate x e y possiamo riutilizzare l'equazione della prima sfera per trovare la coordinata z :

$$z = \pm\sqrt{r_1^2 - x^2 - y^2}$$

Si può quindi notare che il problema può non avere soluzione nel caso di un numero negativo sotto radice oppure può averne una o due. Nel caso di due soluzioni può essere utile avere anche una quarta circonferenza qualora non fosse possibile si possono usare altri stratagemmi per evitare ambiguità. Ad esempio il sistema GPS, che fa uso di algoritmi di trilaterazione, quando ha a disposizione solo tre satelliti che portano a due soluzioni assume che quella corretta sia quella più vicina alla superficie terrestre poiché è improbabile che l'utente si trovi nello spazio al di fuori delle orbite dei satelliti. Questo metodo di calcolare la posizione si complica di parecchio quando si hanno degli errori di misura sui raggi delle sfere: in questo caso si può considerare due raggi per ogni sfera uno massimo e uno minimo le soluzioni saranno date da tutti i punti appartenenti al volume dato dall'intersezione dei tre "gusci".

3.3 Multilaterazione

Come si può evincere dal suo nome la multilaterazione è il caso generico della trilaterazione in cui si possono avere le distanze non più rispetto a soli 3 punti di riferimento ma questa volta rispetto ad n punti di riferimento. In un sistema di localizzazione in cui le misure di distanza rispetto ai punti di riferimento sono soggette ad errori, può essere molto utile ricorrere a questa tecnica di calcolo, che permette di tenere in considerazione le misure rispetto a tutti i punti di riferimento rilevati dal punto mobile.

3.3.1 Multilaterazione atomica

Con multilaterazione atomica si intende quando tale calcolo viene applicato ad un singolo punto mobile per stimarne la posizione, ha un requisito fondamentale: il nodo mobile deve avere nelle vicinanze almeno tre nodi di riferimento con cui può comunicare direttamente con un solo salto. Immaginiamo di avere le misure di distanza rispetto a n punti di riferimento di coordinate note: ne risulta un sistema di n equazioni. L' i -esima equazione non è altro che la circonferenza con centro di coordinate pari al suo i -esimo punto di riferimento (x_i, y_i) e di raggio r_i pari alla distanza rilevata verso tale punto di riferimento.

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

Il sistema può essere linearizzato sottraendo l'ultima equazione dalle prime $n - 1$ equazioni:

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

Riordinando i termini si possono costruire le matrici per risolvere il sistema nella forma $Ax = b$ dove:

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

Dato che la matrice A non è quadrata si può risolvere il sistema con la seguente formula:

$$x = (A^T \cdot A)^{-1} \cdot A^T \cdot b$$

Dove con x si intende un vettore di due elementi contenente ascissa e ordinata della soluzione (x_m, y_m) . Se le misure sono perfette, tutte le circonferenze si intersecheranno in un solo punto:

la soluzione del problema. Nel caso molto più realistico in cui le misure non sono perfette e dove quindi si potrebbero avere circonferenze che si intersecano in più punti dando origine a un'area di possibili soluzioni oppure potrebbero non intersecarsi affatto, la multilaterazione fornisce la soluzione che più si avvicina a quella corretta, che quindi tenderà ad essere più vicina al centro della circonferenza più piccola e man mano più distante dai centri dalle altre. Per valutare la bontà della soluzione appena calcolata si può calcolare il residuo tra le distanze misurate r_i e le distanze che si avrebbero dalla posizione stimata (x_m, y_m) :

$$residuo = \frac{\sum_{i=1}^n \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} - r_i}{n}$$

Un residuo molto ampio è indice di un sistema di equazioni inconsistente e questo è dovuto a misure soggette a forti errori, quando il residuo supera la portata massima dei dispositivi conviene scartare il risultato e ripetere le misure. Il calcolo di multilaterazione appena visto può essere facilmente implementato su *Matlab/Octave* con la seguente funzione:

```
%Funzione per il calcolo di multilaterazione
%Ingressi:      x: vettore delle ascisse dei punti di riferimento
%              y: vettore delle ordinate dei punti di riferimento
%              r: vettore delle distanze rilevate dai punti di riferimento
%
%Uscite:        xm: stima dell'ascissa del punto da localizzare
%              ym: stima dell'ordinata del punto da localizzare
%              residuo: residuo del calcolo di multilaterazione

function [xm,ym,residuo] = multilaterate(x,y,r)
xm=0;
ym=0;
residuo=-10; %residuo del calcolo di multilaterazione
n=length(x); %n è il numero delle equazioni
if(n>=3 & length(y)==n & length(r)==n)
    A=zeros(n-1,2); %creazione matrice A (n-1 x 2)
    b=diag(A);      %creazione vettore colonna b
    for i=1:n-1
        A(i,1)=2*(x(i)-x(n));
        A(i,2)=2*(y(i)-y(n));
        b(i)=x(i)^2-x(n)^2+y(i)^2-y(n)^2+r(n)^2-r(i)^2;
    end
    m=inv(A.'*A)*A.'*b; %m vettore dei risultati
    somma=0;
    for i=1:n %sommatoria per il calcolo del residuo
        somma=somma+sqrt((x(i)-m(1))^2+(y(i)-m(2))^2)-r(i);
    end
    residuo=somma/n;
    xm=m(1);
    ym=m(2);
end
```

3.3.2 Multilaterazione iterativa

Con multilaterazione iterativa si intende il processo eseguito su di un singolo elaboratore, quindi centralizzato, che si occupa di raccogliere tutte le misure da una rete di sensori e di ripetere la multilaterazione atomica su ogni nodo mobile di cui ancora non si conosce la posizione. Dato che non è detto che tutti i nodi mobili possano rispettare il requisito necessario per la multilaterazione si può usare quest'approccio: nel momento in cui si è stimata la posizione di un nodo mobile questo viene considerato come punto di riferimento con la posizione appena stimata. In tal modo si aggiungono punti di riferimento al sistema che possono aiutare a stimare la posizione dei nodi mobili che non avevano abbastanza punti di riferimento nelle vicinanze.

3.3.3 Multilaterazione collaborativa

Ci sono dei casi in cui un singolo nodo mobile non ha abbastanza nodi di riferimento nei suoi paraggi in cui si trovano invece altri nodi mobili di posizione indeterminata nelle sue stesse condizioni. In pratica si possono venire a creare delle situazioni in cui alcuni nodi mobili vedono meno di tre nodi di riferimento nelle loro vicinanze (non è quindi possibile stimare la loro posizione con la multilaterazione atomica) ma nella rete ogni nodo “vede” almeno tre altri nodi o mobili o di riferimento. I nodi possono collaborare tra di loro per provare a determinare la posizione dei nodi mobili per i quali non è stato possibile trovarla iterando con la multilaterazione.

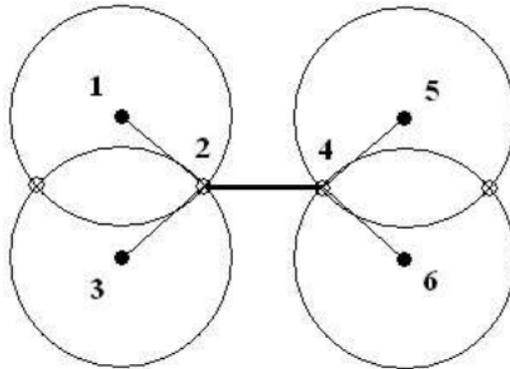


Figura 3.3. Esempio di multilaterazione collaborativa ben riuscita: i nodi 1, 3, 5 e 6 sono di posizione determinata mentre i nodi 2 e 4 sono di posizione indeterminata. Le intersezioni tra le circonferenze rappresentano le possibili posizioni di 2 e 4. Le linee che uniscono i nodi rappresentano le coppie di nodi partecipanti. In questo caso la distanza rilevata tra 2 e 4 permette di risolvere il sistema.

Per capire il funzionamento di questo algoritmo occorre definire due concetti:

- **Nodo partecipante:** è un nodo di cui si conosce la posizione (un nodo di riferimento oppure un nodo mobile di cui si è appena calcolata la posizione) oppure è un nodo di posizione indeterminata con almeno tre vicini partecipanti.

- **Coppia di nodi partecipanti:** è una coppia di nodi partecipanti che sono in grado di comunicare direttamente in cui uno è di posizione determinata e l'altro è di posizione indeterminata oppure entrambi di posizione indeterminata.

Con ogni coppia di nodi partecipanti si può quindi costruire un sistema di equazioni che potrebbero portare a stimare la posizione di tutti i nodi.

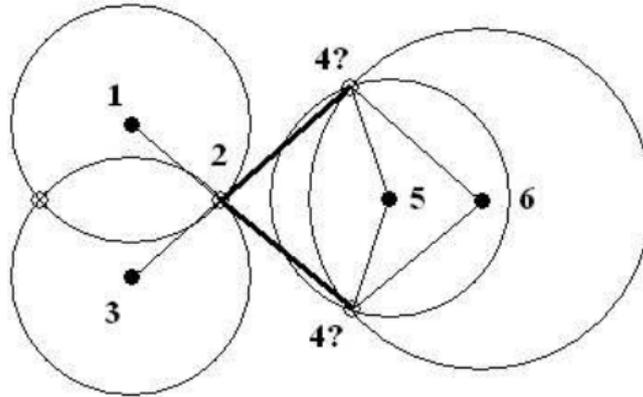


Figura 3.4. Esempio di multilaterazione collaborativa non riuscita: come prima i nodi 1, 3, 5 e 6 sono di posizione determinata mentre i nodi 2 e 4 sono di posizione indeterminata. In questo caso si può trovare la posizione del nodo 2 ma la distanza tra 2 e 4 non permette di trovare la posizione di 4.

3.4 Min-max

I calcoli matriciali appena visti per fare la multilaterazione si traducono in numerose operazioni *floating point* che risultano molto costose in termini computazionali qualora sia necessario stimare le posizioni di numerosi nodi mobili frequentemente, in un sistema centralizzato o che risultano inappropriate da eseguirsi direttamente sull'hardware limitato dei nodi della rete in un sistema distribuito. Il metodo min-max, ideato da *Andreas Savvides* molto più semplice permette di avere delle stime comunque vicine a quelle che si avrebbero con la multilaterazione. L'idea principale si basa sul costruire dei quadrati (al posto delle circonferenze) per ogni punto di riferimento. Ogni punto di riferimento è al centro di un quadrato il cui lato è il doppio della distanza rilevata rispetto a quel punto di riferimento; quindi il quadrato per l' a -esimo punto di riferimento (x_a, y_a) , rilevato a distanza r_a può essere così rappresentato:

$$[x_a - r_a, y_a - r_a] \times [x_a + r_a, y_a + r_a]$$

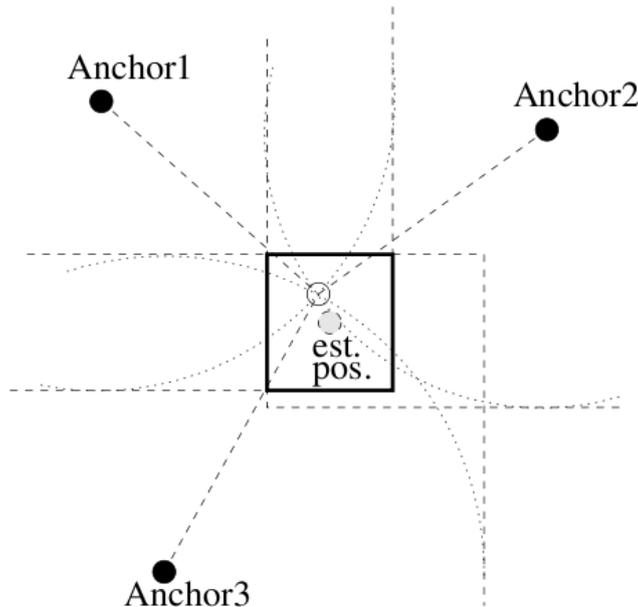


Figura 3.5. Il centro del quadrilatero formato dall'intersezione dei tre quadrati da una stima della posizione del nodo mobile che non differisce di molto dall'intersezione delle circonferenze che si avrebbe con la multilaterazione.

L'intersezione di questi quadrati determina un'altro quadrilatero i cui estremi possono essere facilmente calcolati prendendo i massimi e i minimi di tutte le coordinate:

$$[\max(x_i - r_i), \max(y_i - r_i)] \times [\min(x_i + r_i), \min(y_i + r_i)]$$

La stima della posizione non è altro che il centro del quadrilatero di intersezione e quindi la media delle coordinate dei due angoli precedenti. Anche qui è possibile calcolare il residuo per valutare la bontà di tale stima.

3.5 Osservazioni indirette

Viene qui di seguito illustrato il metodo delle osservazioni indirette, strumento che, come si vedrà in seguito, viene usato dal prototipo realizzato per cercare di migliorare le stime delle coordinate dei punti mobili.

3.5.1 Compensazione di reti

Un insieme di punti di coordinate note (punti di riferimento) o presunte, un insieme di misure (angoli, distanze ecc), uno schema geometrico costituiscono una rete.

Il metodo comunemente seguito nella compensazione di reti è quello chiamato delle osservazioni indirette o per variazione di coordinate. Ciò significa che le incognite da ricercare non sono le coordinate x, y ma le correzioni $\delta x, \delta y$ da apportare ai valori presunti x, y delle coordinate dei punti stessi. Ciò avviene mediante l'applicazione ripetuta (iterazione) di un opportuno algoritmo di calcolo finché le correzioni $\delta x, \delta y$ hanno valori trascurabili cioè il sistema converge. Il procedimento di compensazione da usare è quello che fa ricorso al metodo statistico dei minimi quadrati attuato secondo l'algoritmo delle osservazioni indirette.

3.5.2 Metodo dei minimi quadrati applicato alle osservazioni indirette

Il metodo dei minimi quadrati parte dall'ipotesi che le misure contengano solo errori accidentali che seguono la distribuzione normale e che le misure siano indipendenti fra loro.

Per determinare i valori di m grandezze incognite $x_1 x_2 \dots x_m$ si misurano r grandezze $M_1 M_2 \dots M_r$ legate alle prime da n relazioni analitiche dette equazioni generatrici:

$$f_i(x_1, x_2 \dots x_m; M_1, M_2 \dots M_r) = 0$$

$i = 1, 2 \dots n$

A causa degli errori di misura si ottengono le equazioni generate:

$$f_i(x_1, x_2 \dots x_m; M_1, M_2 \dots M_r) = \nu_i$$

Con $n > m$ equazioni, che devono essere lineari, maggiore del numero di variabili si ha un sistema sovradeterminato che non ha una soluzione unica. In questo caso si cerca la soluzione migliore, cioè quella di minima varianza. Si risolve ricorrendo al metodo dei minimi quadrati applicato agli scarti ν_i che stabilisce:

In un sistema di osservazioni dello stesso ordine di precisione, il valore più probabile X_0 è quello che rende minima la somma dei quadrati degli scarti delle singole osservazioni X_i da X_0 .

$$\sum_i (x_0 - x_i)^2$$

Quindi si determinano i valori più probabili delle incognite, cioè quelli che rendono minima la somma dei quadrati degli scarti:

$$\sum_i \nu_i^2 = \min$$

Principio dei minimi quadrati

Se ν_i sono le n variabili stocastiche con densità di probabilità:

$$f(\nu_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{\nu_i}{\sigma_i}\right)^2}$$

La probabilità che avvengano eventi contemporaneamente è data dalla probabilità composta:

$$f(\nu_1, \nu_2, \dots, \nu_n) = \left(\frac{1}{\sigma_i \sqrt{2\pi}}\right)^n \cdot \prod_{i=1}^n \frac{1}{\sigma_i} \cdot e^{-\frac{1}{2} \cdot \sum_{i=1}^n \left(\frac{\nu_i}{\sigma_i}\right)^2}$$

La $f(\nu_1, \nu_2, \dots, \nu_n)$ ha un massimo quando l'esponente $-\frac{1}{2} \cdot \sum_{i=1}^n \left(\frac{\nu_i}{\sigma_i}\right)^2$ raggiunge un minimo cioè:

$$\sum_{i=1}^n \left(\frac{\nu_i}{\sigma_i}\right)^2 = \min$$

Se introduciamo una costante arbitraria σ_0 la condizione di minimo non viene alterata:

$$\sum_{i=1}^n \frac{\sigma_0^2 \cdot \nu_i^2}{\sigma_i^2} = \min$$

Per tener conto di misure con diversa precisione si introduce una funzione peso detta peso dell'osservazione così definita:

$$P_i = \frac{\sigma_0^2}{\sigma_i^2}$$

Così la condizione dei minimi quadrati diventa:

$$\sum_{i=1}^n (P_i \cdot \nu_i^2) = \min$$

In forma vettoriale:

$$\nu^T \cdot P \cdot \nu = \min$$

Un'osservazione che abbia $\sigma_i = \sigma_0$ ha peso 1 pertanto σ_0^2 è detta varianza dell'unità peso. Il nostro sistema sovradeterminato è quindi del tipo:

$$f_i(x_1, x_2, \dots, x_m; M_1, M_2, \dots, M_r) = \nu_i$$

Per procedere dobbiamo tener conto di due cose:

- Non tutte le equazioni del sistema, per la diversa precisione delle misure, contano allo stesso modo occorre quindi pesarle.
- Non tutte le equazioni si presentano in forma lineare, occorre linearizzarle.

Linearizzazione delle equazioni generate

La funzione f in un intorno del punto x_0 dove è derivabile può essere rimpiazzata dalla sua linearizzata nel punto x_0 :

$$l_{x_0}(x) = f(x_0) + f'(x_0)(x - x_0)$$

Nel nostro caso dobbiamo linearizzare le equazioni generate: $f_i(x, M) = \nu_i$ se le quantità misurate M non sono eccessivamente disperse (solo errori accidentali) anche le x non differiranno eccessivamente, sarà quindi possibile trovare dei valori approssimati x_0 delle incognite x e operare un cambiamento di variabile del tipo $x = x_0 + \delta x$ sviluppabile in serie di Taylor arrestato al primo ordine ottenendo così per una generica equazione del sistema:

$$f_i(x_0, M) + \frac{\partial f_i}{\partial x_1} \cdot \delta x_1 + \frac{\partial f_i}{\partial x_2} \cdot \delta x_2 + \dots + \frac{\partial f_i}{\partial x_m} \cdot \delta x_m = \nu_i$$

Dove il primo termine è una quantità nota $f_i(x_0, M) = l_i$ le δx_k sono le nuove incognite i cui coefficienti $\frac{\partial f_i}{\partial x_k}$ (calcolati in x_0) costituiscono la matrice A .

Dalla linearizzazione si ottiene un sistema lineare di n equazioni in m incognite δx e n incognite ν di forma:

$$A \cdot \delta x + l = \nu$$

Analisi ponderale delle equazioni generate

Il problema si può ricondurre alla determinazione della varianza della misura, a sua volta riconducibile alla precisione dello strumento di misura.

Il peso da assegnare a ciascuna equazione sarà inversamente proporzionale alla varianza della misura cioè del termine noto l . Infatti si dimostra che la varianza di l è uguale alla varianza degli scarti δ delle equazioni e quindi i pesi di queste saranno inversamente proporzionali alle varianze dei termini noti l . La matrice dei pesi sarà una matrice diagonale con i pesi pari a:

$$P_k = \frac{1}{\sigma_k^2}.$$

$$P = \begin{bmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_n \end{bmatrix}$$

Il sistema sottodeterminato $A\delta x + l = \nu$ associato alla condizione di minimo pesato della somma degli scarti al quadrato, dà luogo ad un sistema esattamente determinato detto sistema normale:

$$A \cdot \delta x + l = \nu$$

$$\sum_{i=1}^n (P_i \cdot \nu_i^2) = \min$$

La cui soluzione è data da:

$$\delta x = \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_n \end{bmatrix} = -(A^T \cdot P \cdot A)^{-1} \cdot A^T \cdot P \cdot l$$

La stima delle varianze delle incognite δx è data dai termini diagonali h_{ii} della matrice quadrata $H = (A^T P A)^{-1}$ moltiplicati per il termine σ_0^2 la varianza dell'unità peso data dalla sommatoria degli scarti pesati al quadrato divisi per la ridondanza del sistema ovvero:

$$\sigma_0^2 = \frac{\sum_{i=1}^n (P_i \cdot \nu_i^2)}{n - m} = \frac{\nu^T \cdot P \cdot \nu}{n - m}$$

Dove n è il numero delle equazioni (misure) e m è il numero delle incognite.

$$\sigma_{x_j}^2 = \sigma_0^2 \cdot h_{jj}$$

Inoltre le correlazioni C_{ik} fra le incognite (la covarianza di x_j e x_k) sono ottenute dagli elementi non diagonali di H : h_{jk}

$$C_{jk} = \sigma_0^2 \cdot h_{jk}$$

Ellisse d'errore

Con la varianza e la covarianza delle incognite si può tracciare l'ellisse d'errore.

I semiassi a e b e l'angolo ϕ (antiorario) che il semiasse maggiore forma con l'asse delle ascisse sono dati da:

$$a = \left| \sqrt{\frac{1}{2} \cdot (\sigma_{x_j}^2 + \sigma_{x_k}^2) + \frac{1}{2} \cdot \sqrt{(\sigma_{x_j}^2 - \sigma_{x_k}^2)^2 + 4C_{jk}^2}} \right|$$

$$b = \left| \sqrt{\frac{1}{2} \cdot (\sigma_{x_j}^2 + \sigma_{x_k}^2) - \frac{1}{2} \cdot \sqrt{(\sigma_{x_j}^2 - \sigma_{x_k}^2)^2 + 4C_{jk}^2}} \right|$$

$$\phi = \frac{1}{2} \tan^{-1} \left(\frac{-2 \cdot C_{jk}}{\sigma_{x_k}^2 - \sigma_{x_j}^2} \right)$$

Capitolo 4

Reti di sensori

Le *Wireless Sensor Network* sono reti costituite da dispositivi solitamente alimentati a batteria che comunicano tra di loro tramite onde radio. Questi dispositivi possono avere a bordo sensori di vario genere come per esempio sensori di temperatura, di pressione, accelerometri, sensori di luce ecc. Recentemente, gli studi svolti sulle reti di sensori wireless hanno conosciuto un grande sviluppo e un forte interesse da parte del mondo scientifico. La loro crescita è stata soprattutto motivata dalla necessità di poter disporre di reti senza infrastruttura e senza un controllo centralizzato, perciò semplici ed economiche da realizzare. Generalmente le WSN prevedono la modalità di funzionamento *ad-hoc*, in cui i nodi che compongono la rete, mediante un continuo scambio di messaggi, sono in grado di determinare autonomamente le migliori modalità per cooperare e comunicare i dati rilevati all'utente finale.

4.1 Possibili applicazioni

Inutile dire che queste reti trovano molteplici applicazioni soprattutto dove vi è la necessità di monitorare parametri ambientali, dove è richiesta un'installazione rapida e dove sono assenti fonti di alimentazione permanenti. Si possono suddividere le possibili applicazioni in svariati gruppi:

- **Applicazioni militari:** rilevamento intrusioni o attacchi chimici, monitoraggio e tracciamento obiettivi.
- **Applicazioni ambientali:** tracciamento animali, monitoraggio parametri ambientali, rilevamento incendi o allagamenti.
- **Applicazioni di sorveglianza:** antifurti, sorveglianza edifici pubblici.
- **Applicazioni industriali:** sensori nei processi industriali e nelle catene di montaggio.
- **Applicazioni logistiche:** sensori per monitorare la posizione e lo stato durante lo stoccaggio di beni.

- **Applicazioni domestiche:** scambio dati per applicazioni domestiche.
- **Applicazioni *automotive*:** sensori e trasporto dati a bordo di autoveicoli.

4.2 Funzionamento

Il trasporto dei dati in queste reti avviene con strategie *multi-hop*. Ciò significa che se due dispositivi intendono scambiare dati ma non sono in grado di raggiungersi direttamente (perché sono troppo lontani o perché vi è un ostacolo che impedisce la trasmissione diretta tra di loro) dovranno appoggiarsi su altri dispositivi intermedi. Si può dire che i vari nodi della rete collaborino tra di loro per trasportare e per fare routing delle varie trasmissioni in corso, quindi è bene definire a priori quali compiti dovrà svolgere ogni nodo nella rete. Se per esempio dovrà occuparsi di inoltrare i pacchetti ricevuti da altri nodi oppure se dovrà preoccuparsi di trasmettere solo i propri pacchetti per risparmiare energia.

4.3 Protocolli di rete

Il protocollo di rete maggiormente usato dalle WSN è IEEE 802.15.4/ZigBee. ZigBee ha una velocità massima di trasferimento dati pari a 250 kbit/sec, un buon numero di unità collegabili in rete: fino a 64.000 dispositivi e una portata che può arrivare fino a 70-300 m a seconda del tipo di antenne usate. Vediamo ZigBee a confronto con gli altri standard di reti senza fili:

- Rispetto alle reti *WiFi* IEEE 802.11 ZigBee lavora a velocità di trasmissione molto più basse e con un raggio di copertura decisamente inferiore.
- Anche rispetto a Bluetooth la velocità di trasmissione è più bassa (Bluetooth 1 può raggiungere la velocità massima di 1 mbit/sec), il raggio di copertura è superiore di Bluetooth 1 (10 m) ma non è detto che lo sia anche rispetto a Bluetooth 2 (100 m). Mentre il numero massimo di unità collegabili in rete è a favore di ZigBee (Bluetooth può gestire fino a 8 dispositivi collegati tra di loro).

Le caratteristiche salienti di ZigBee sono: il basso consumo di energia, il basso costo dei dispositivi e l'alta versatilità con cui la rete riesce a riadattare rapidamente la propria topologia nel caso di cambiamenti o di spostamento dei dispositivi. Queste caratteristiche, fondamentali per le WSN, hanno fatto di ZigBee lo standard emergente.

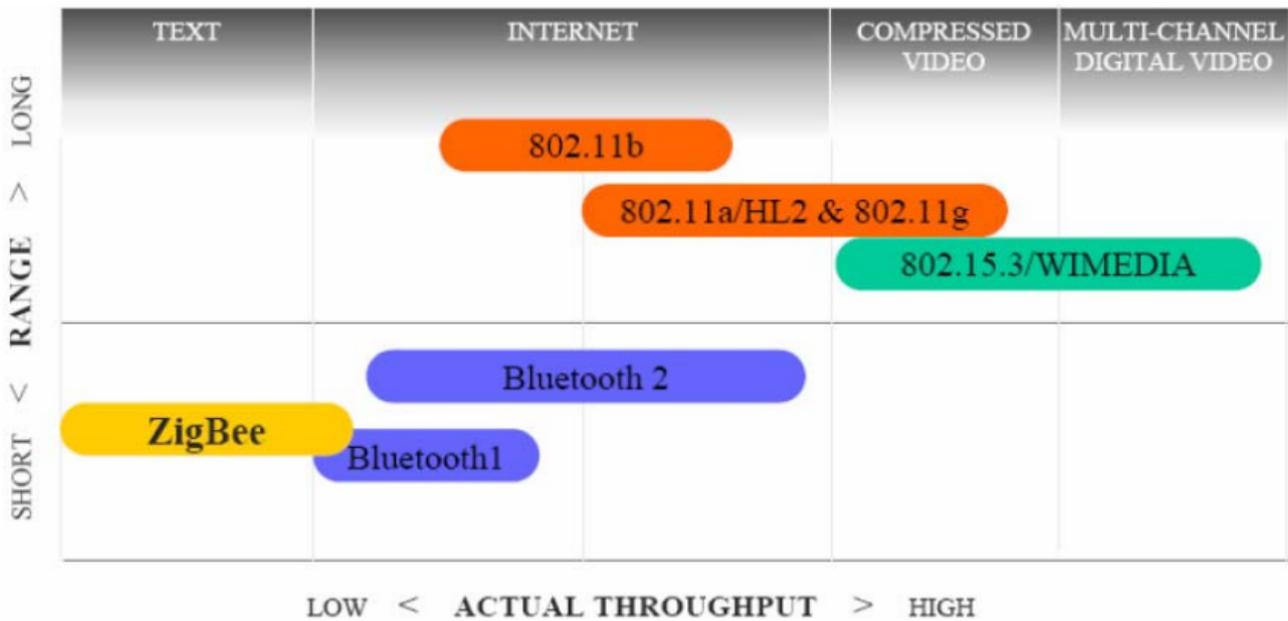


Figura 4.1. ZigBee a confronto con gli altri standard di reti senza fili. Sull'asse X del diagramma vi è la velocità di trasmissione, mentre sull'asse Y vi è il raggio massimo di trasmissione.

4.4 Componenti delle reti WSN

I componenti coinvolti in una rete WPAN (*Wireless Personal Area Network*) a basso data rate sono raggruppati in due differenti categorie a seconda delle loro funzionalità:

- **Full-Function Devices (FFD)**: sono dei dispositivi (chiamati comunemente nodi) in grado di eseguire tutte le operazioni necessarie per gestire la rete, quali: inoltrare pacchetti in una comunicazione a più salti (*multi-hop*) oppure fornire il tempo di riferimento per la sincronizzazione dei nodi.
- **Reduced-Function Devices (RFD)**: sono dei nodi più semplici rispetto ai nodi FFD, il loro compito è di trasmettere dati utili all'applicazione, non si occupano della gestione e mantenimento della rete; di conseguenza possono essere implementati usando risorse minime. Un nodo RFD può comunicare solo con un nodo FFD, mentre un nodo FFD può comunicare direttamente con più nodi RFD o altri nodi FFD.

4.5 ZigBee Alliance

Un gruppo di aziende internazionali fra i massimi esponenti nel campo delle WSN e dei sistemi *embedded* hanno fondato *ZigBee Alliance* per creare un comune standard di comunicazione che coinvolge tutti i livelli della pila protocollare, dal livello fisico al livello applicativo, e proporre

soluzioni per un insieme di applicazioni a corto raggio e a basso data rate. Il protocollo ZigBee è basato sullo standard IEEE 802.15.4 per quanto concerne i livelli fisico (PHY) e MAC mentre definisce il livello rete (NWT) e la struttura per il livello applicativo. La versione 1.0 di ZigBee è stata rilasciata nel dicembre 2004.

4.6 Standard IEEE 802.15.4

Lo standard IEEE 802.15.4 definisce un protocollo di interconnessione di dispositivi che comunicano per mezzo di onde radio e a bassa velocità, per reti WPAN (*Wireless Personal Area Network*): una comunicazione di rete semplice, affidabile, facile da installare, usata per convogliare informazioni a corto raggio (tipicamente 10 metri), per applicazioni che richiedono bassi livelli di potenza trasmessi (per avere un basso consumo di batteria) e bassi valori di *data rate*. Lo standard IEEE 802.15.4 è quindi la base ideale per le WSN, come già detto ZigBee fa uso del livello fisico (PHY) e del livello MAC (*Medium Access Control*) definiti da IEEE 802.15.4. La prima versione dello standard IEEE 802.15.4 è stata rilasciata nell'ottobre 2003, tale versione si focalizza su un insieme di applicazioni industriali, residenziali e medicali in cui la trasmissione dei dati avviene in modo non periodico, solitamente con un *duty cycle* minore del 5%. Una successiva versione dello standard è stata rilasciata nel settembre 2006 con il numero: IEEE 802.15.4B.

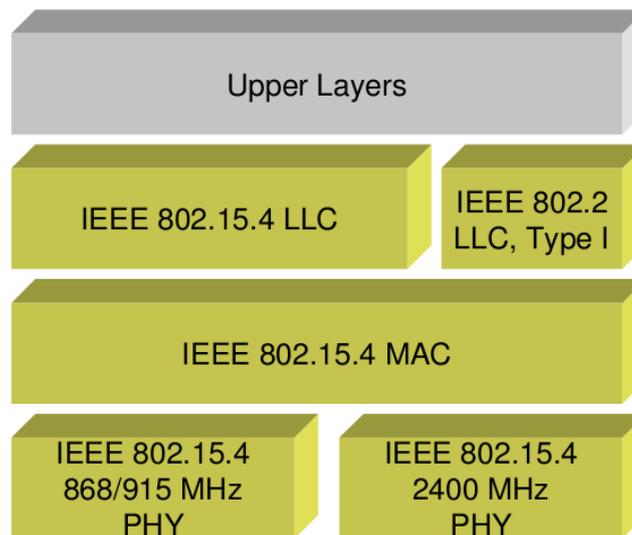


Figura 4.2. La pila protocollare dello standard IEEE 802.15.4. In basso si vedono i due blocchi fisici per le due bande utilizzate da IEEE 802.15.4, quindi si ha il livello MAC, per poi passare a LLC (*Logical Link Control*) su cui si appoggiano i livelli applicativi. I livelli definiti da ZigBee si appoggiano direttamente sul livello MAC.

4.6.1 Livello fisico

IEEE 802.15.4 definisce due livelli fisici basati sulla tecnologia *Direct Sequence Spread Spectrum* (DSSS) suddivisi a seconda della gamma di frequenze utilizzate:

1. 868/915 MHz opera nella banda ISM (*Industrial Scientific Medical*) europea (868.0 - 868.6 MHz) e nella ISM americana (902 - 928 MHz). Nella banda 868 MHz è disponibile un solo canale (il canale 0) con un data rate di 20 kbps, mentre nella banda a 915 MHz sono allocati 10 canali (dal canale 1 al canale 10), spaziatosi di 2 MHz, con un data rate di 40 kbps per ognuno.
2. 2450 MHz opera nella banda ISM (quasi mondiale) 2400 - 2483.5 MHz e supporta un data rate di 250 kbps, in aria. Questa banda è suddivisa in 16 canali (dal canale 11 al canale 26) spaziatosi di 5 MHz.

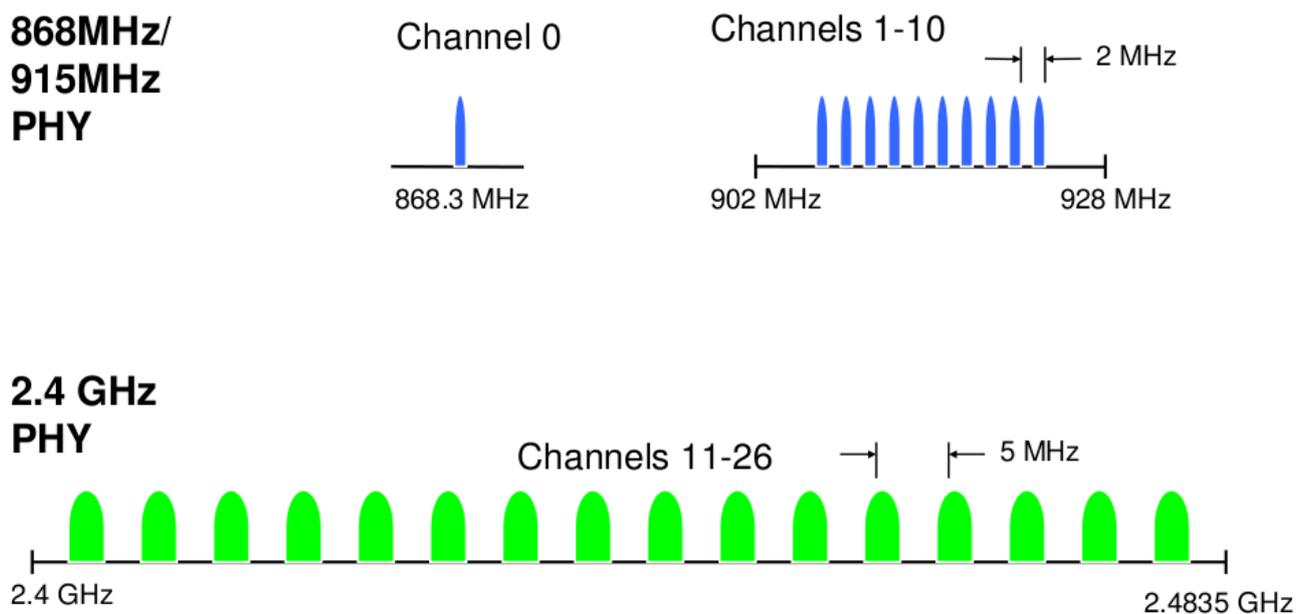


Figura 4.3. Diagrammi di frequenza che rappresentano i 27 canali utilizzabili con ZigBee.

Il livello fisico svolge inoltre le seguenti funzioni:

- **Selezione del canale.** Il livello fisico sceglie uno tra i 27 canali disponibili a seconda della richiesta dal livello MAC.
- **Attivazione e disattivazione del trasmettitore radio.** Attraverso apposite primitive, il livello MAC è capace di indirizzare una richiesta al livello fisico, chiedendo di modificare lo stato del ricetrasmittitore radio (trasmettitore attivo/disattivo, ricevitore attivo/disattivo).

- **Rilevazione del livello di potenza del segnale ricevuto.** Il livello fisico è in grado di stimare la potenza del segnale ricevuto RSSI (*Received Signal Strength Indicator*) all'interno del canale corrente con un'accuratezza di ± 6 dBm.
- **Indicazione del livello di qualità (*link quality*) per i pacchetti ricevuti.** Per ogni pacchetto ricevuto, il livello fisico esegue una caratterizzazione della potenza e/o della qualità del collegamento radio. Il risultato viene segnalato al livello MAC, specificando 8 livelli di qualità.
- **Verifica del canale libero con il protocollo CSMA-CA (*Carrier Sense Multiple Access - Collision Avoidance*).** Il tempo di rivelazione corrisponde al periodo di trasmissione di 8 simboli, durante il quale si verifica se il canale radio è impegnato da un'altra trasmissione per mezzo della misurazione di potenza su quel canale, della rilevazione di portante trasmessa, oppure della combinazione di entrambe.

4.6.2 Componenti di una rete IEEE 802.15.4

In una rete IEEE 802.15.4 WPAN, possono essere considerati tre principali componenti:

- **Nodi:** i componenti di base delle reti WPAN. Questi possono essere dei nodi FFD oppure dei nodi RFD, alimentati a batteria o da altre fonti di energia.
- **PAN Coordinator:** è unico e corrisponde al nodo FFD più importante all'interno della rete WPAN. Si occupa di inizializzare la rete, permette ad altri nodi di connettersi ad esso e li sincronizza. Data la sua importanza, il PAN Coordinator è alimentato da fonti di energia diverse da batterie.
- **Coordinators:** sono dei nodi FFD che potrebbero essere presenti all'interno delle reti WPAN con l'obiettivo di fornire un servizio di sincronizzazione attraverso l'invio di segnali chiamati *beacon*.

I nodi che lavorano in una rete WPAN a basso data rate possono usare un indirizzamento esteso a 64 bit, oppure un indirizzamento a 16 bit allocato dal PAN coordinator durante le procedure di associazione. Una singola rete WPAN a basso data rate è capace di gestire fino a 65535 nodi.

4.6.3 Topologie di rete

All'interno dello standard IEEE 802.15.4 sono previste tre possibili topologie di rete:

- **Topologia a stella:** i nodi comunicano solo direttamente con il nodo PAN coordinator. Un nodo FFD si comporta da PAN coordinator; prima di tutto inizializza la rete, dopo permette agli altri nodi (sia RFD che FFD) di connettersi alla rete.
- **Topologia peer-to-peer:** è composta da un nodo PAN coordinator e da qualsiasi altro nodo in grado di comunicare con qualsiasi altro nodo a condizione che questi siano all'interno del loro range di comunicazione. Si possono creare delle topologie di rete complesse

come quelle a maglia, note come reti *mesh*. In aggiunta, le topologie di rete peer-to-peer possono essere ad-hoc ed autoconfiguranti; permettono multi-hop, routing e migliorano l'affidabilità fornendo *multi-path routing*.

- **Topologia Cluster-Tree:** è un caso particolare di topologie peer-to-peer, in cui la maggior parte dei nodi sono FFD; i nodi RFD possono unirsi alla rete come nodi foglie alla fine di un ramo. Qualsiasi nodo FFD che fa parte della rete potrebbe assumere il ruolo di coordinatore. La rete potrebbe essere inizializzata da un nodo PAN coordinator che diventa il capo cluster ed inizia ad organizzare il primo cluster. I nodi adiacenti, se permesso, possono poi unirsi alla rete. La rete WPAN risultante può essere successivamente estesa attraverso l'elezione di nuovi capi cluster che formano nuovi cluster adiacenti al primo (questo tipo di espansione aumenta l'area di copertura ma anche il ritardo dei messaggi).

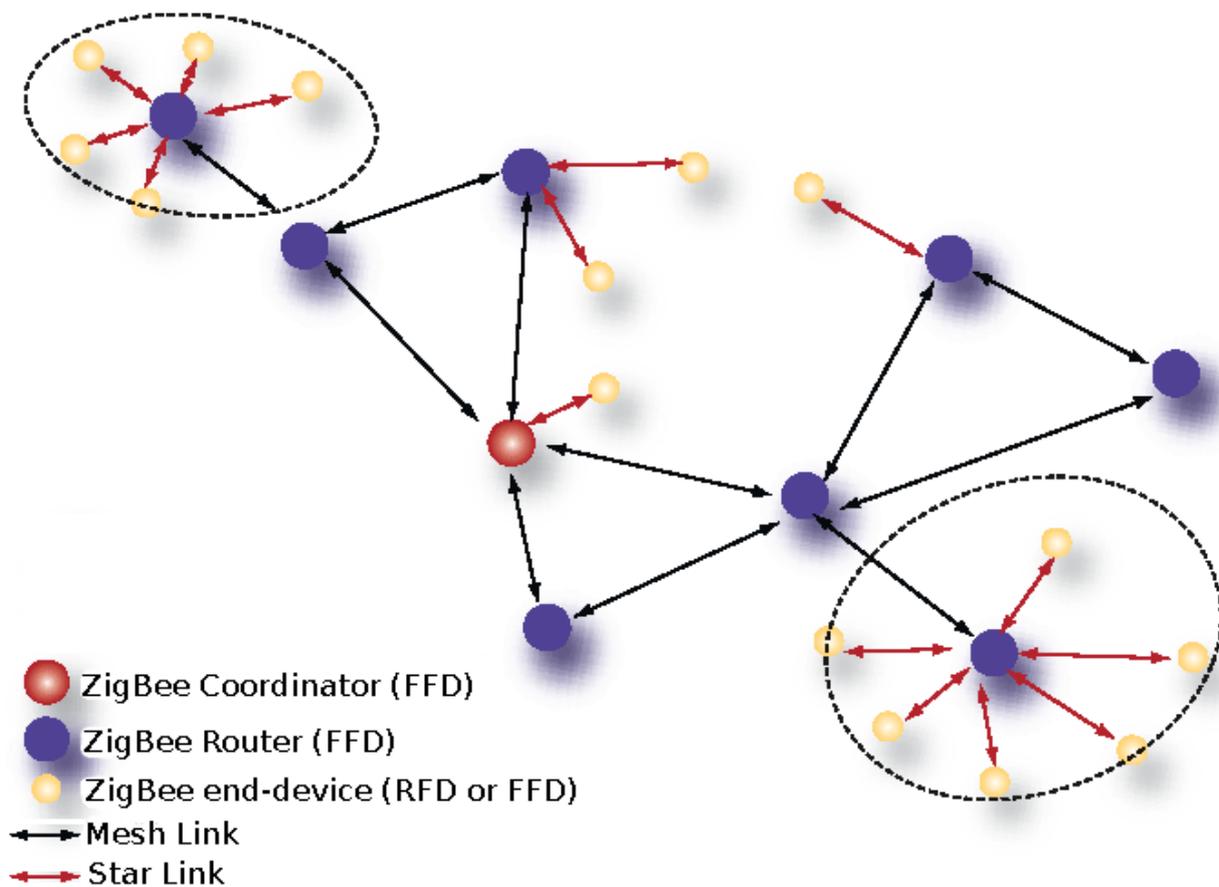


Figura 4.4. Esempio di topologia Cluster Tree.

4.6.4 Livello MAC

Secondo lo standard IEEE 802.15.4, il livello MAC fornisce due servizi di base: *MAC data service*, che permette la trasmissione e la ricezione dei dati del protocollo MAC attraverso il servizio dati del livello fisico, ed il *MAC management service*. Il livello MAC inoltre definisce quattro differenti frame: *beacon frame*, *data frame*, *acknowledgement frame*, *MAC command frames*. La lunghezza massima permessa per l'unità dati del protocollo MAC è di 127 byte; a livello fisico si può considerare un ulteriore overhead di 6 byte. Il protocollo MAC IEEE 802.15.4 definisce due differenti modalità di funzionamento:

- ***Non-beacon enabled mode***: la rete WPAN è sempre attiva e l'accesso al canale viene gestito dall'algoritmo CSMA-CA *unslotted*. Il nodo PAN coordinator (un nodo FFD) deve tenere sempre la radio accesa per essere pronto a ricevere i dati trasmessi in qualsiasi momento dai nodi RFD. I nodi RFD possono accendere la radio anche solo quando hanno qualcosa da trasmettere.
- ***Beacon enabled mode***: il nodo PAN coordinator spedisce periodicamente un segnale di beacon ai nodi della rete, solitamente ai nodi RFD. Un beacon contiene informazioni di tempo che permettono di tenere la rete sincronizzata. La comunicazione è organizzata in *superframe* i cui limiti sono delimitati da due beacon consecutivi. La durata di un superframe viene chiamata *beacon interval*. Il superframe è a sua volta suddiviso in due segmenti consecutivi:
 - un periodo attivo, sempre presente in modo tale che la sua durata, chiamata *superframe duration*, non può essere nulla;
 - un periodo inattivo opzionale, seguente il periodo attivo, che permette di risparmiare energia grazie ai dispositivi *low power mode*.

Il periodo attivo è ulteriormente suddiviso in due parti, i cui limiti comuni possono variare da un superframe al successivo:

1. in ordine troviamo prima il CAP (*Contention Access Period*), in cui l'accesso al canale è gestito dall'algoritmo CSMA-CA slottizzato;
2. di seguito al CAP troviamo un periodo opzionale chiamato CFP (*Contention Free Period*), composto da un numero di slot temporali, chiamati GTS (*Guaranteed Time Slots*), che abilitano l'accesso al canale senza contese (cioè accesso con probabilità di collisione nulla). L'accesso ai GTS è gestita dal PAN coordinator, il quale li alloca ad uno specifico nodo che, nel caso abbia dati urgenti da spedire, indirizza una richiesta per un accesso senza contese. Le allocazioni GTS ad un superframe MAC abilitano il trattamento della trasmissione dei dati per applicazioni a bassa latenza, come quelli che potrebbero incontrarsi in alcuni ambienti industriali.

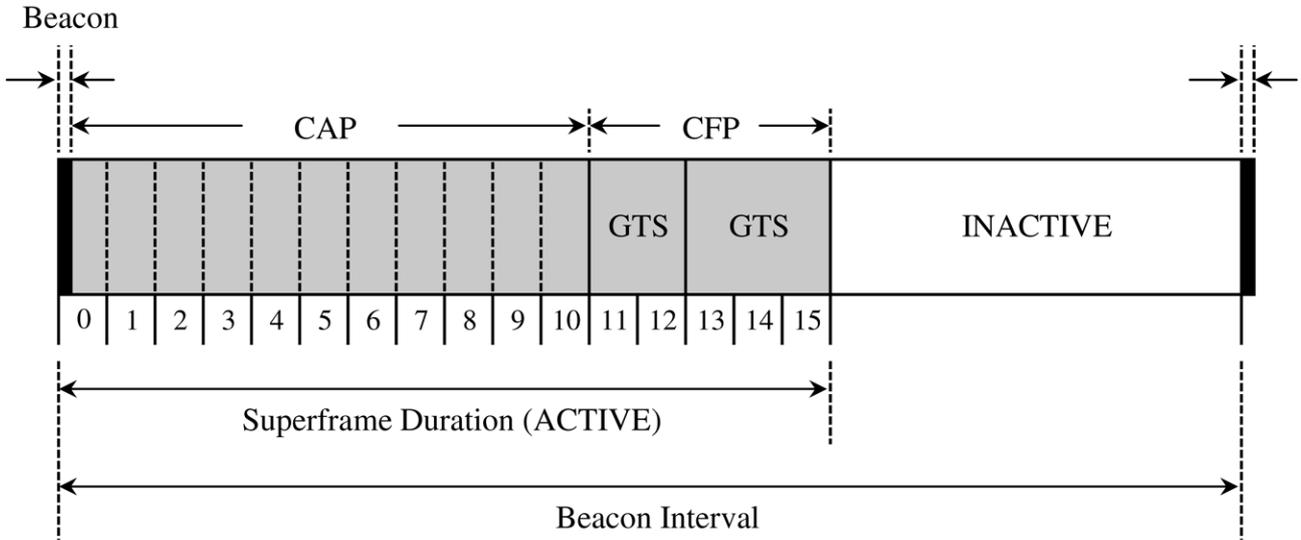


Figura 4.5. Suddivisione di un *superframe*.

Lo standard definisce tre differenti trasferimenti di dati, definiti in modo tale che questi possono essere controllati dai nodi della WPAN piuttosto che dal nodo coordinatore:

- **Trasferimento dati verso il coordinatore** (trasmissione dati diretta): viene eseguito usando un algoritmo CSMA-CA slottizzato oppure non slottizzato, a seconda della modalità di funzionamento. L'uso di ACK (*acknowledgement*) è opzionale.
- **Trasferimento dati da un coordinatore ad un nodo** (trasmissione dati indiretta): viene controllato dal nodo, il quale, prima si accorge che i dati sono pronti al nodo coordinatore per essere inviati, dopo indirizza una richiesta. Ricevendo tale richiesta il coordinatore trasmette i dati e si aspetta la ricezione degli ACK.
- **Trasferimento dati *peer-to-peer***: i nodi possono ricevere costantemente e trasmettere dati usando il CSMA-CA non slottizzato oppure sincronizzarsi tra di loro.

Le funzionalità del MAC includono anche l'inizializzazione e il mantenimento della rete PAN. Prima di inizializzare una rete PAN, una scansione attiva del canale oppure una scansione attraverso *Energy Detection* può essere eseguita da un nodo FFD per valutare la disponibilità corrente dei canali, per poi selezionare un canale logico appropriato ed un unico identificatore PAN nell'area. In particolare, durante la scansione attiva del canale, il nodo trasmette delle richieste di comandi beacon su ogni canale tra quelli specificati ed aspetta le risposte, selezionando un adatto identificatore di PAN. La scansione *Energy Detection* abilita i nodi FFD a collezionare le misure dei picchi di energia per uno specifico insieme di canali, i quali potrebbero essere utili per la selezione del canale. Il livello MAC è anche responsabile per le procedure di associazione e dissociazione che permettono ai nodi di connettersi e disconnettersi dalla rete WPAN. Un nodo FFD potrebbe annunciare la sua presenza in una rete inviando beacon: la

procedura *device discovery* può essere eseguita di conseguenza. Un nodo non connesso può iniziare l'associazione dopo una scansione attiva del canale oppure dopo una scansione passiva del canale. Durante la scansione passiva, i nodi semplicemente analizzano i beacon ricevuti e memorizzano i relativi descrittori della rete PAN. Successivamente, un nodo può scegliere un'adeguata rete PAN e, se gli è permesso, connettersi alla rete. La disconnessione può essere ottenuta semplicemente attraverso la trasmissione di un comando che notifica la disconnessione.

4.7 Protocollo ZigBee

ZigBee Alliance sviluppa uno standard wireless focalizzandosi su una combinazione semplice a due vie: a basso costo e a basso consumo di potenza. L'obiettivo principale è di dotare i dispositivi usati nell'elettronica di consumo, nella domotica, nei controlli industriali, e nei sensori per applicazioni medicali di comunicazioni wireless. Le specifiche ZigBee definiscono l'architettura della pila protocollare, specificando un insieme di livelli. Come visto precedentemente i livelli fisico e MAC sono quelli definiti da IEEE 802.15.4. ZigBee si occupa principalmente dei livelli più alti: il livello rete (NWK) e il livello applicazione (APL) che include i sottolivelli *Application Support* (APS), l'*Application Framework* (AF), il *ZigBee Device Object* (ZDO) e il *Manufacturer-defined application object*. Le specifiche contenute nella versione 1.0 di ZigBee, sono state rilasciate nel dicembre 2004. La versione 1.1 di ZigBee mira a sfruttare i miglioramenti proposti dal *TG4b*, uno dei gruppi dello standard IEEE 802.15.4, introducendo nuove capacità e aumentando la flessibilità di ZigBee.

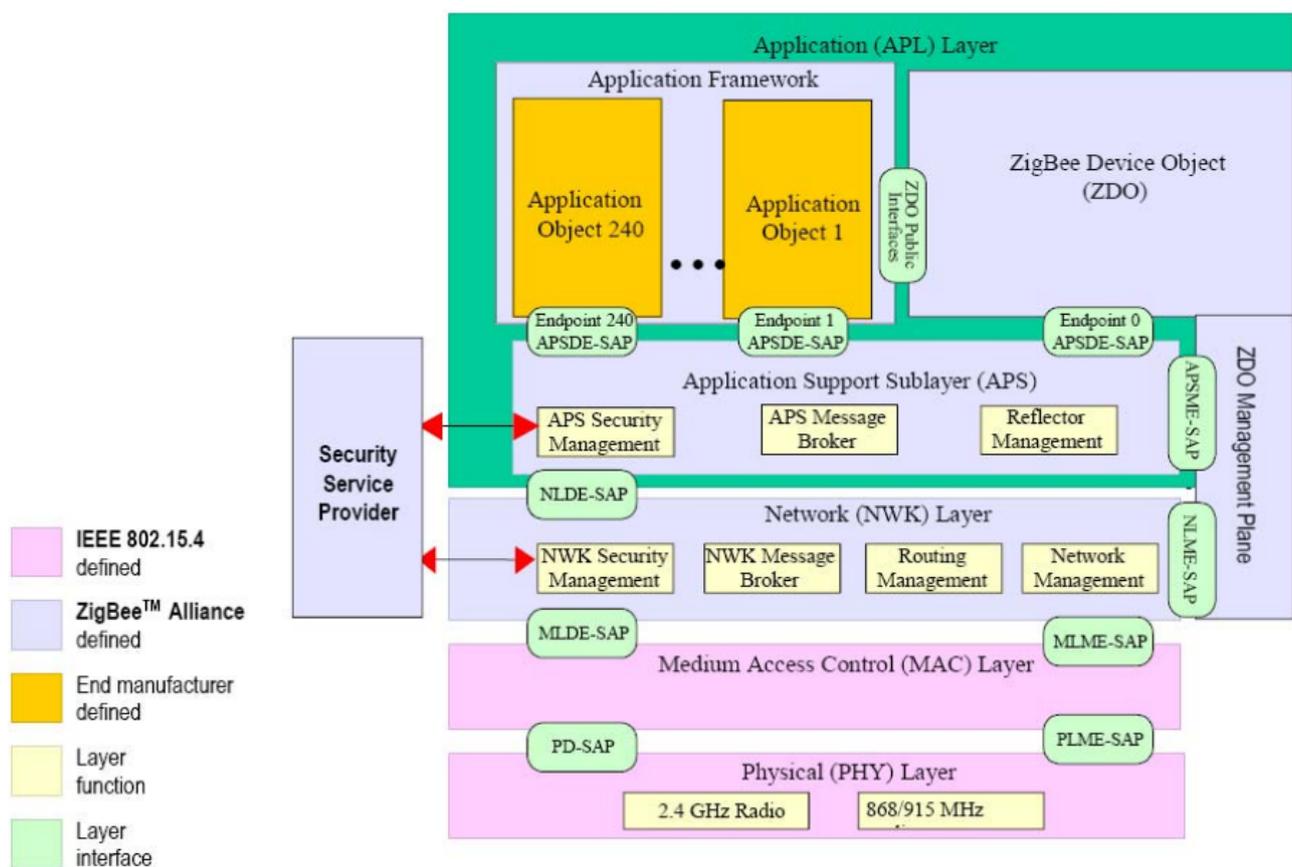


Figura 4.6. Rappresentazione completa dello *stack* ZigBee

4.7.1 Livello di rete

Il livello rete fornisce un'interfaccia tra il livello MAC di 802.15.4 e il livello applicazione. In più, esso incorpora caratteristiche ad-hoc quali il *multi-hop mesh routing* e l'autoconfigurazione. Le specifiche ZigBee definiscono tre tipi di dispositivi logici:

1. **Nodo ZigBee coordinator:** è un nodo coordinatore della rete PAN IEEE 802.15.4, responsabile per inizializzare e mantenere la rete ZigBee, permettendo anche a nuovi nodi di connettersi alla rete.
2. **Nodo ZigBee router:** è un nodo FFD IEEE 802.15.4 con l'incarico di mantenere la rete, smistare i pacchetti in transito ma non è il coordinatore.
3. **Terminale ZigBee (end-device):** è un nodo RFD o FFD IEEE 802.15.4 che partecipa alla rete ZigBee con semplici funzionalità (connessione/disconnessione dalla rete) non si occupa delle problematiche inerenti alla rete e allo smistamento del traffico.

In accordo con lo standard IEEE 802.15.4, il livello rete supporta tre tipologie di rete: star, cluster tree e mesh. Nel caso di reti cluster tree e reti mesh, i nodi ZigBee router sono impiegati per estendere la copertura di rete. Per quanto riguarda la comunicazione, le tipologie ad albero permettono di usare la modalità di funzionamento *beacon enabled mode* definito nello standard IEEE 802.15.4; tuttavia, nel caso di reti mesh in ZigBee, la trasmissione di beacon non è permessa. Il livello rete fornisce un insieme di servizi che permettono di configurare nuovi dispositivi (cioè impostazioni della pila protocollare richieste per determinate operazioni), inizializzare una nuova rete ZigBee e instradare i pacchetti nella rete. In particolare, un nodo router e il coordinatore impiegano soluzioni di routing che utilizzano alberi e tabelle di routing (come si vedrà più avanti si tratta di una versione semplificata dell'algoritmo *Ad Hoc On Demand Distance Vector Routine*). Maggiori funzionalità si riferiscono alla connessione e la disconnessione di rete, l'assegnazione di indirizzi logici di rete a nuovi nodi durante il processo di associazione, *neighbor discovery* e *route discovery*.

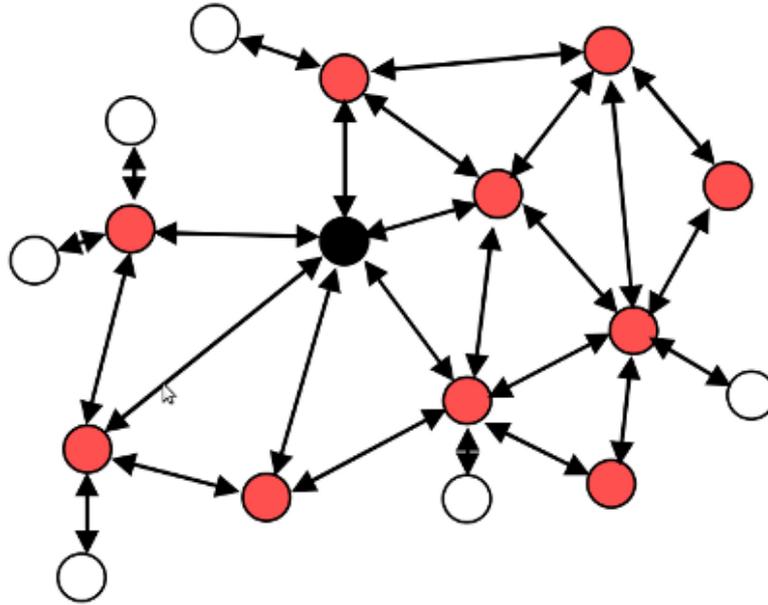


Figura 4.7. Un esempio di rete ZigBee, il *coordinator* in nero, i *router* in rosso e gli *end-device* in bianco.

Coordinator

E' il dispositivo che inizializza e la rete ZigBee. Il coordinatore innanzitutto sceglie il canale sul quale avverranno le trasmissioni, e l'identificativo di rete (PAN ID) quindi avvia la rete. Questo nodo può svolgere compiti relativi alla sicurezza e al *binding* di rete. In pratica il compito fondamentale del coordinatore è quello di configurare e avviare la rete, fatto questo si comporta come un normale router, dopo che la rete è stata avviata essa può continuare a funzionare anche senza il coordinatore.

Router

I router permettono ai nuovi dispositivi (che vengono accesi nelle loro vicinanze) di associarsi alla rete, svolgono il *multi-hop* routing e assistono la comunicazione dei loro dispositivi "figli", ovvero sottostanti ad essi nella gerarchia della rete. Solitamente ci si aspetta che i router siano sempre attivi e quindi si cerca di alimentarli in modo permanente. I router comunque possono operare anche in una modalità speciale detta "*Cluster Tree*" che permette di accendere l'interfaccia radio periodicamente per brevi intervalli di tempo, ottenendo così un notevole risparmio di energia che permette ai router di essere alimentati anche a batteria.

End-device

Gli *end-device* non hanno alcuna responsabilità di mantenimento dell'infrastruttura di rete quindi sono liberi di spegnersi e riaccendersi in qualunque momento. Infatti è soprattutto in

questi nodi periferici della rete che si cerca di ottenere il massimo risparmio di energia, proprio per la necessità di alimentarli a batteria. Solitamente anche i requisiti di memoria sono più bassi per gli *end-device*.

4.7.2 Livello applicazione

Il livello applicativo di ZigBee è composto da:

- L'**application support sub-layer** fornisce un insieme di dati e servizi di gestione da cui possono accedere lo ZDO ed il *manufacturer-defined application object*. In particolare l'APS è responsabile per il servizio di *binding*: i relativi processi abilitano due nodi, modellati come applicazioni ZigBee, ad essere logicamente connessi a livello applicazione, anche considerando aspetti di sicurezza.
- L'**application framework** è il blocco logico che contiene le applicazioni ZigBee. Al massimo possono essere gestite 240 applicazioni e riferite ad uno specifico terminale APSDE: applicazioni separate potrebbero condividere infatti la stessa radio. Il terminale speciale 0 si riferisce allo ZDO. Le applicazioni ZigBee sono capaci di accedere a due servizi dati forniti dall'application framework: il servizio KVP (*Key Value Pair*) e il *Generic Message Service*. Il primo servizio offre la capacità di acquisire e modificare i valori pertinenti agli attributi considerati nelle applicazioni ZigBee ed astratti in KVP. Il *Generic Message Service* fornisce una generalizzazione del servizio KVP, permettendo alle applicazioni di usare protocolli proprietari. Nell'ambito ZigBee, ogni tipo di applicazione è caratterizzato da una convenzione sui messaggi, sui processi e sui valori configurabili nella pila ZigBee chiamato *profile*. In particolare, sono definiti tre differenti classi di profili: *private*, *published* e *public*. I profili pubblici facilitano l'interoperabilità tra i differenti venditori interessati nelle stesse aree di applicazioni. Un profilo speciale viene anche dettagliato e chiamato *ZigBee Device Profile* (ZDP) il quale definisce le capacità supportate in tutti i dispositivi ZigBee.
- Il **ZigBee Device Object** (ZDO) è un'applicazione che include le funzionalità basilari condivise da tutti i dispositivi ZigBee e sono collocati nel livello applicazione, sopra l'APS. Il compito del ZDO consiste nell'inizializzare il livello APS, il livello rete e il *Security Service Provider*, nel definire il ruolo di uno specifico dispositivo all'interno della rete ZigBee, effettuare *device discovery* e *service discovery*, inoltre esso si occupa della gestione di sicurezza, di gestione della rete e del *binding*

4.7.3 Stack Profile

Agli occhi dello sviluppatore di firmware per dispositivi ZigBee, lo stack protocollare appare come un insieme di file sorgenti in C da compilare insieme a quelli della propria applicazione, da cui si potranno invocare le primitive desiderate come ad esempio `AF_DataRequest()` per trasmettere un pacchetto. Di seguito si farà riferimento alla configurazione dello stack per dispositivi equipaggiati con l'integrato *ChipCon CC2430/CC2431*. L'insieme dei parametri

dello stack che necessitano di essere configurati a valori specifici, viene chiamato *stack profile*. ZigBee Alliance ha una sua versione delle definizioni dello stack profile: ZigBee-2006. Per poter funzionare correttamente tutti i dispositivi presenti in una rete devono usare gli stessi parametri per lo stack profile. Lo scopo delle specifiche per lo stack profile di ZigBee Alliance è proprio quello di permettere interoperabilità tra dispositivi di produttori diversi, a patto che ogni dispositivo sia conforme alle specifiche dello stack profile ZigBee. Gli sviluppatori per applicazioni specifiche sono liberi di modificare i parametri dello stack ma la rete risultante non sarà più *ZigBee compliant* e lo stack profile sarà “*network specific*”. Lo stack profile identifier è presente anche nei beacon trasmessi da ciascun dispositivo, questo permette di determinare lo stack profile di una rete prima di associarsi ad essa. Gli stack profile “*network specific*” hanno un ID impostato a 0 mentre lo stack profile ZigBee-2006 ha l’ID impostato a 1. La definizione di questo identificativo si trova nel file *nwk_globals.h* alla voce: `STACK_PROFILE_ID`.

4.7.4 Indirizzamento

Tipi di indirizzi

In ZigBee vi sono due tipi di indirizzi: un indirizzo su 64 bit detto *IEEE address*, *MAC address* o indirizzo esteso e un indirizzo su 16 bit chiamato *network address*, *short address* o indirizzo logico. L’indirizzo su 64-bit è globalmente unico, viene assegnato dal produttore e accompagnerà ogni dispositivo per tutta la sua vita, esattamente come avviene per l’indirizzo MAC delle schede ethernet. Questi indirizzi vengono gestiti dall’IEEE e ogni produttore deve acquisirne un blocco di da poter assegnare ai propri dispositivi. Ogni dispositivo porta il proprio IEEE address assegnato dal produttore anche stampato su di un’etichetta, è possibile cambiare quest’indirizzo, ma conviene utilizzare sempre quello predefinito dal produttore che ha la garanzia di essere unico in qualunque rete si andrà ad associare il dispositivo. Invece l’indirizzo su 16 bit viene assegnato dal coordinatore ad ogni dispositivo che si associa ad una rete, viene mantenuto da tale dispositivo solo per il tempo in cui sarà associato alla rete ed è unico solo all’interno della predetta rete (un po’ come se fosse un indirizzo IP assegnato tramite DHCP in una rete privata). Lo short address viene usato non solo per identificare un’dispositivo all’interno di una singola rete ma anche per trasmettergli dati, essendo su 16 bit si può evincere che il numero massimo di dispositivi che si possono associare ad una rete ZigBee è minore di 65536. Il coordinatore ha sempre lo short address 0x0000 mentre 0xFFFF è l’indirizzo di broadcast.

Assegnamento indirizzi di rete

ZigBee usa un piano di indirizzamento distribuito, che garantisce che gli indirizzi assegnati siano unici in tutta la rete. Ogni nuovo dispositivo che si associa alla rete otterrà il suo short address univoco direttamente dal router ‘padre’, non vi è bisogno di una comunicazione attraverso tutta la rete e questo chiaramente aumenta la versatilità della rete. Questo schema di indirizzamento si basa su alcuni parametri predefiniti nello stack ZigBee e quindi vengono configurati in ogni router che si associa alla rete.

- `MAX_DEPTH` determina la massima profondità della rete. Il coordinatore è a livello 0, i suoi figli saranno al livello 1 e i figli dei suoi figli saranno al livello 2 e così via. Nel caso di una rete di sensori per monitorare il traffico installata lungo il guard-rail di un'autostrada questo parametro andrà a limitare quanto fisicamente potrà essere “lunga” la rete. Lo stack profile ZigBee 2006 definisce questo parametro a 5.
- `MAX_CHILDREN` determina il massimo numero di figli che ogni router, o il coordinatore può avere. Lo stack profile ZigBee 2006 definisce questo parametro a 20.
- `MAX_ROUTERS` determina il massimo numero di figli router che ogni nodo router o il coordinatore può avere, questo valore deve essere chiaramente minore o uguale di `MAX_CHILDREN` e la differenza `MAX_CHILDREN - MAX_ROUTERS` determinerà lo spazio di indirizzamento che quel nodo router potrà dedicare a nodi *end-device*.

Lo sviluppatore può modificare questi parametri per meglio adattarsi al tipo di applicazione richiesta, occorre però verificare la validità dei nuovi valori, visto che non si può avere più di 2^{16} indirizzi. Se questi parametri vengono modificati lo stack non sarà più conforme allo standard *ZigBee 2006* ma sarà *network-specific* e quindi sarà necessario impostare a `NETWORK_SPECIFIC` la definizione `STACK_PROFILE_ID` presente nel file `nwk_globals.h`.

Indirizzamento

Per trasmettere dei dati in una rete ZigBee, a livello applicativo, generalmente si usa la funzione `AF_DataRequest()` gli indirizzi su 16 bit sono definiti nella struttura:

```
typedef struct {
    union {
        uint16 shortAddr;
    } addr;
    afAddrMode_t addrMode;
    byte endPoint;
} afAddrType_t;
```

Come si può notare è necessario specificare, oltre all'indirizzo su di rete su 16 bit, anche l'*address mode* e l'*end-point*. L'*address mode* è necessario per stabilire se la trasmissione sarà *unicast*, *multicast*, *broadcast* o indiretta.

Unicast

Un pacchetto *unicast* viene spedito ad un singolo destinatario, è il modo di indirizzamento più semplice. Sempre se si conosce l'indirizzo di rete su 16 bit del destinatario, in tal caso si potrà spedire un pacchetto impostando `addrMode` a `Addr16Bit`.

Multicast

Un pacchetto *multicast* un ha più di un destinatario ovvero un gruppo di dispositivi identificato da un indirizzo di gruppo sempre su 16 bit. In questo caso `shortAddr` conterrà l'indirizzo di gruppo e `addrMode` sarà impostato a `AddrGroup`. Ovviamente prima di poter fare trasmissioni in *multicast* i gruppi devono essere definiti nella rete, per far ciò si usa la funzione `aps_AddGroup()`.

Broadcast

I destinatari di un pacchetto *broadcast* sono tutti i dispositivi presenti in rete. Si tratta di un caso particolare di trasmissione *multicast*. `addrMode` va impostato a `AddrBroadcast` mentre l'indirizzo `shortAddr` va impostato con un indirizzo dei seguenti indirizzi di *broadcast*:

- `0xFFFF`: indica tutti i dispositivi presenti in rete, anche quelli in stato *sleep*, che riceveranno il messaggio al loro risveglio.
- `0xFFFD`: indica tutti i dispositivi presenti in rete eccetto quelli in stato *sleep*, in pratica il messaggio verrà ricevuto da tutti i dispositivi che hanno l'interfaccia radio accesa in quel momento.
- `0xFFFC`: indica tutti i dispositivi presenti in rete con funzionalità di *router*, in questo caso il messaggio non verrà inoltrato agli *end-device*.

Trasmissione indiretta

Questo modo di indirizzamento si usa quando non si conosce ancora l'indirizzo del destinatario, in questo caso `addrMode` va impostato a `AddrNotPresent` e quindi non è necessario specificare l'indirizzo di destinazione. Nei livelli sottostanti dell'stack si provvederà a risolvere l'indirizzo di destinazione tramite le tabelle di *binding* presenti sul dispositivo mittente. Si parlerà di *binding* più avanti. Dopodiché il pacchetto verrà trattato come un qualunque altro pacchetto *unicast*.

Ottenere gli indirizzi

Nello stack vi sono alcune utilissime funzioni che permettono di conoscere l'indirizzo del dispositivo sul quale ci si trova o l'indirizzo del dispositivo padre:

- `NLME_GetShortAddr()` ritorna l'indirizzo di rete corrente su 16 bit.
- `NLME_GetExtAddr()` ritorna l'indirizzo esteso corrente su 64 bit.
- `NLME_GetCoordShortAddr()` ritorna l'indirizzo di rete del nodo padre su 16 bit.
- `NLME_GetCoordExtAddr()` ritorna l'indirizzo esteso del nodo padre su 64 bit.

4.7.5 Binding

Il binding è il meccanismo che permette di controllare il flusso di messaggi da un'applicazione ad un'altra o ad un gruppo di applicazioni. Come visto precedentemente il binding permette di inviare un pacchetto senza conoscerne l'indirizzo di destinazione ma bensì l'applicativo verso il quale dovrà essere inviato. Il livello APS si occuperà di trovare l'indirizzo di destinazione dalla sua tabella di binding, per poi inoltrare il messaggio a tale indirizzo che quindi potrebbe essere anche un indirizzo di gruppo. A differenza del vecchio stack ZigBee 1.0 che memorizzava tutte le associazioni di binding nel solo coordinatore, ZigBee 2006 memorizza le associazioni di binding in ogni nodo, ogni dispositivo ha quindi la sua tabella di binding. Lo stack mette a disposizione tutta una serie di funzioni per gestire il binding. Occorre anche definire i vari tipi di messaggi che dovranno essere scambiati tra i vari dispositivi. Il *ClusterID* è l'identificativo su 16 bit che permette di riconoscere il tipo di pacchetto. Per esempio si può definire un pacchetto di richiesta dei valori dei sensori a bordo del nodo destinatario assegnandogli un determinato *ClusterID* e il relativo pacchetto di risposta contenente i valori richiesti che avrà un altro *ClusterID* a questo punto, in fase di inizializzazione, si dovrà fare il binding nei vari nodi a seconda se si vuole che questi siano in grado di richiedere o rispondere alla richiesta dei valori dei sensori definita precedentemente. Così ogni nodo, a seconda dell'applicativo che dovrà eseguire, potrà essere abilitato o meno a rispondere a tale richiesta, anche se quest'ultima è stata spedita in broadcast, ma non solo: anche gli altri nodi dopo aver aggiornato le proprie tabelle di binding, per l'indirizzamento indiretto, sapranno a chi inoltrare le richieste di quel tipo.

4.7.6 Routing

In ZigBee, così come nelle reti a maglia il routing è decentralizzato e più nodi collaborano tra di loro per svolgere questo compito. Il routing è del tutto trasparente a livello applicativo: quando si spedisce un messaggio, quest'ultimo viene passato ai livelli più bassi dello stack, ma l'applicativo non è in alcun modo responsabile del trovare il percorso migliore per raggiungere la destinazione passando attraverso altri nodi. Una delle caratteristiche chiave di ZigBee è la capacità di trovare rapidamente percorsi alternativi qualora un collegamento dovesse venire a mancare.

Protocollo di routing

ZigBee usa, come protocollo di routing, un'implementazione di AODV (*Ad hoc On demand Distance Vector*), semplificato per le reti di sensori. Quando un nodo router riceve un pacchetto *unicast* controlla se il destinatario è uno dei suoi figli, in tal caso lo ritrasmette direttamente al nodo di destinazione. Altrimenti se il router trova l'indirizzo del destinatario nella propria tabella di routing provvederà a inoltrare il pacchetto verso il *next hop address* letto dalla tabella di routing. Infine se il router non trova alcuna *active routing table entry* per l'indirizzo di destinazione si vedrà costretto a procedere con una *route discovery*, in attesa che la tabella di routing venga quindi aggiornata con il nuovo percorso, il pacchetto viene memorizzato

temporaneamente in un buffer dopodiché potrà essere inoltrato al prossimo router verso la sua destinazione. Come già detto gli *end-device* non si occupano di routing e quando hanno qualcosa da trasmettere lo inoltrano semplicemente verso il proprio nodo padre, che, essendo un router, si occuperà quindi di inoltrare il pacchetto nella giusta direzione. Il modo in cui vengono assegnati gli indirizzi di rete permette di riconoscere facilmente i vari 'rami' del percorso per ogni destinazione, e questo permette di minimizzare le dimensioni delle tabelle di routing.

Ricerca dei percorsi migliori

Route discovery è la procedura con cui i nodi della rete collaborano per trovare nuovi percorsi attraverso la rete, può essere iniziata da un qualsiasi nodo router ed è sempre diretta verso un particolare indirizzo di destinazione. In pratica il meccanismo di *route discovery* trova tutti i possibili percorsi verso una destinazione e prova a stabilire qual'è il migliore. *Route selection* è il meccanismo con cui si sceglie il percorso con il costo più basso. Ogni nodo tiene costantemente traccia dei costi di ogni collegamento verso tutti i suoi vicini. Solitamente il costo di ogni collegamento è una funzione inversa della potenza del segnale ricevuto da quel singolo nodo. Sommando i costi di tutti i collegamenti che formano un percorso si ottiene il costo di quel percorso (*route cost*). Per scoprire i possibili percorsi verso un dato indirizzo, il router richiedente inizia a mandare in broadcast dei pacchetti di *route request* (RREQ) verso i suoi vicini. Quando un nodo riceve un pacchetto RREQ, provvede ad aggiornare il campo contenente il costo del percorso, aggiungendoci il costo dell'ultimo collegamento, dopodiché lo reinoltra in broadcast. In questo modo il pacchetto RREQ trasporta la somma dei costi di tutti i collegamenti che ha attraversato, gli inoltri si susseguono attraverso l'intera rete finché non si raggiunge il nodo con l'indirizzo desiderato. Al destinatario giungeranno molti pacchetti RREQ provenienti da strade diverse e quindi con costi diversi, il destinatario potrà ora scegliere il percorso più conveniente e rispondere al richiedente con un pacchetto RREP (*Route Reply*). Il pacchetto RREP viaggia a ritroso sul percorso migliore, i router che se lo passano ne approfittano per aggiornare le proprie tabelle di routing con il nuovo percorso appena scoperto. Anche il richiedente, quando riceve il pacchetto RREP, aggiorna la sua tabella di routing, il percorso è ora pronto e si può procedere a trasmettere dati per il nodo di destinazione.

Mantenimento dei percorsi

Qualora un router non ricevesse più gli ACK su un determinato collegamento (ciò significa che si è persa la connettività su quel collegamento) procede a fare *route discovery* verso la destinazione per cercare di aggirare il collegamento mancante. Se capita che da quel nodo non vi è più modo di raggiungere la destinazione il router deve invalidare i percorsi esistenti che fanno uso di quel collegamento, questo viene fatto inviando un pacchetto RERR verso gli altri router interessati che annulleranno i percorsi interrotti, fino al nodo sorgente che farà a sua volta *route discovery* per trovare percorsi alternativi che partono da esso, attraverso altri 'rami' della rete.

Scadenza dei percorsi

Come già detto le tabelle di routing contengono un record per ogni percorso stabilito, se per un certo periodo di tempo non vengono più spediti pacchetti su un dato percorso questo viene marcato come scaduto. I percorsi scaduti non vengono cancellati dalla tabella di routing finché non è necessario spazio per memorizzare percorsi nuovi. Il tempo massimo di validità di un percorso senza che questo venga usato può essere regolato dallo sviluppatore andando a modificare il valore della definizione `ROUTE_EXPIRY_TIME`, tale valore è espresso in secondi. Per disabilitare la scadenza dei percorsi basta impostare `ROUTE_EXPIRY_TIME` a 0.

Routing table

Ogni record presente sulle tabelle di routing (presenti a loro volta su ogni router e sul coordinatore) contiene: l'indirizzo di destinazione, l'indirizzo del prossimo router verso la destinazione (ovvero il *next-hop address*), e lo stato del collegamento. Tutti i pacchetti diretti verso un certo indirizzo presente nella tabella di routing verranno inoltrati al *next-hop address*. Le tabelle di routing hanno una dimensione prefissata che può essere definita dallo sviluppatore andando a modificare il valore di `MAX_RTG_ENTRIES`.

Route discovery table

I router coinvolti in una o più *route discovery* devono mantenere anche una tabella per memorizzare temporaneamente le informazioni delle *route discovery* in corso. Ogni riga della *route discovery table* contiene le informazioni per una singola *route discovery*, di conseguenza la dimensione massima di questa tabella (`MAX_RREQ_ENTRIES`) determina il numero massimo di *route discovery* che un nodo può fare contemporaneamente.

4.7.7 Portabilità

In ZigBee 2006 è stata aggiunta questa caratteristica, ovvero quando un *end-device* perde il collegamento con il router padre prova a riassociarsi alla rete cercando un altro router padre. Un *end-device* che ha perso il collegamento con il nodo padre e quindi con il resto della rete viene detto orfano. Per determinare se il collegamento è stato perso si tiene conto del numero di errori di trasmissione durante il normale scambio di messaggi, se si supera un valore limite definito in `MAX_POLL_FAILURES_RETRIES` si considera il collegamento perso. Regolando il valore di `MAX_POLL_FAILURES_RETRIES` lo sviluppatore può quindi regolare la sensibilità agli errori di trasmissione che determineranno la perdita del collegamento. Con un numero alto si avrà quindi una bassa sensibilità, ma si avrà un tempo più lungo prima che il dispositivo affetto da molti errori di trasmissione riesca a riassociarsi alla rete facendo uso di un collegamento migliore. E' la funzione `ZDO_SyncIndicationCB()` che si occupa di fare il *rejoin* alla rete, questo procedimento prima esegue un *orphan-scan* alla ricerca di un nuovo potenziale padre per poi riassociarsi alla rete tramite il nuovo padre facendo uso del *network rejoin command*.

4.7.8 Acknowledge

Per i pacchetti non broadcast vi sono due tipi di acknowledge: *end-to-end acknowledgement* (APS ACK) e *single-hop acknowledgement* (MAC ACK). I MAC ACK sono sempre attivi di default e sono sufficienti a garantire un alto grado di affidabilità della rete. Per aggiungere ulteriore affidabilità alla rete si possono usare anche gli APS ACK. Gli APS ACK sono gestiti dal livello APS dello stack protocollare ZigBee, in pratica il dispositivo mittente mantiene una copia del pacchetto appena spedito finché non riceve il corrispondente APS ACK proveniente dal destinatario che quest'ultimo spedisce quando avrà ricevuto il messaggio. Qualora il mittente non ricevesse l'APS ACK entro un certo timeout (configurabile alla definizione `APSC_ACK_WAIT_DURATION_POLLED`) riproverà a spedire il messaggio per un certo numero di volte (`APSC_MAX_FRAME_RETRIES`). Questa funzionalità può essere attivata facendo uso del campo `options` ogni qualvolta si farà uso della funzione `AF_DataRequest()` per spedire un messaggio.

4.7.9 Scelta del canale

Ogni dispositivo ha (nel file di configurazione `fw8config.cfg`) un'impostazione `DEFAULT_CHANLIST` che controlla il modo in cui sarà scelto il canale. Il coordinatore, che deve scegliere su quale canale creare una nuova rete, cercherà di scegliere il canale con meno rumore, mentre i *router* e gli *end-device* cercheranno i canali su cui vi siano reti esistenti a cui associarsi.

4.7.10 Configurazione del PAN ID

E' possibile predefinire a priori il PAN ID della rete che verrà creata, nel caso di nodo coordinatore, o alla quale ci si assocerà direttamente, nel caso di *router* o *end-device*; per farlo basta impostare un valore compreso tra 0 e 0x3FFF alla voce `ZDO_CONFIG_PAN_ID` del file di configurazione `f8wConfig.cfg`. Se si vuole disabilitare questa caratteristica lasciare `ZDO_CONFIG_PAN_ID` impostato a 0xFFFF.

4.7.11 Memoria non volatile

Per la rete

Un device ZigBee ha molte informazioni di stato che necessitano di essere memorizzate nella memoria non volatile così potranno essere recuperate in caso di mancanza di alimentazione. In alcuni casi potrebbe non essere più possibile riassociarsi alla rete se queste informazioni dovessero venire perse. Per poter fare uso della memoria non volatile è necessario includere `NV_RESTORE` tra le opzioni di compilazione. Sarebbe bene abilitare sempre la memoria non volatile. Il livello ZDO è responsabile di memorizzare e caricare dalla memoria non volatile le tutte le informazioni vitali del livello rete che comprendono: il *Network Information Base* (NIB, si tratta di tutte le informazioni necessarie per gestire il livello rete), la lista dei nodi figli e dei nodi padre, la tabella di *binding* e anche alcune informazioni relative alle impostazioni di sicurezza usate. Quando un nodo viene acceso, si avrà un'inizializzazione molto più rapida se tutte queste informazioni possono essere semplicemente caricate dalla memoria non volatile.

Per l'applicativo

La memoria non volatile torna molto utile anche per memorizzare informazioni riguardanti l'applicazione che viene eseguita da ogni dispositivo. Con le funzioni `osal_nv_read()` e `osal_nv_write()` è possibile leggere e scrivere nella memoria permanente. Ad ogni elemento presente in memoria NV va associato un identificatore univoco compreso tra 0x0201 e 0xFFFF che andrà specificato ogni qualvolta si accederà a tale memoria.

4.7.12 Sicurezza

La sicurezza in ZigBee è basata sulla cifratura AES in modalità operativa CCM per una sicurezza implicita. ZigBee offre le seguenti caratteristiche di sicurezza:

- Sicurezza d'infrastruttura
- Controllo degli accessi (Autenticazione)
- Sicurezza dei dati (Riservatezza)

Configurazione

Per avere una rete sicura occorre compilare il firmware di ogni device con la direttiva del preprocessore `SECURE` impostata a 1. Anche questa impostazione si può trovare nel file `f8wConfig.cfg` e sempre nello stesso file va impostata una chiave di default su 128 bit alla voce `DEFAULT_KEY`. Si può scegliere un valore casuale su 128 bit come chiave di default che può essere preconfigurata su ogni dispositivo oppure può essere configurata solo sul coordinatore che poi la trasmetterà a tutti gli altri dispositivi della rete a seconda di come viene impostata l'opzione `gPreConfigKeys` nel file `nwk_globals.c`. Chiaramente se si sceglie di distribuire la chiave via etere vi sarà un momento di vulnerabilità ogni volta che un dispositivo si assocerà alla rete visto che quest'ultimo dovrà ricevere la chiave in chiaro e nel frattempo qualcuno potrebbe facilmente intercettarla.

Controllo degli accessi

Sul coordinatore è possibile implementare qualsivoglia politica di controllo degli accessi dato che quest'ultimo è informato ogni volta che un dispositivo richiede di associarsi alla rete. Per esempio si può permettere l'associazione e quindi l'accesso alla rete a tutti i nuovi dispositivi che mentre vengono accesi hanno un pulsante premuto. Oppure si può semplicemente accettare o no nuovi nodi a seconda del loro IEEEaddr. Da notare che gli indirizzi IEEE possono essere facilmente riprogrammati e quindi un attaccante potrebbe facilmente associarsi alla rete conoscendo gli indirizzi che vengono accettati. Lo sviluppatore può implementare la politica di controllo accessi desiderata nella funzione `ZDSecMgrDeviceValidate()` che si trova nel file `ZDSecMgr.c`.

Aggiornamento chiavi

Per mantenere la rete sicura è necessario cambiare ogni tanto la chiave di cifratura, anche questo compito è svolto dal coordinatore. Come prima lo sviluppatore può implementare una propria politica di aggiornamento chiavi a seconda del livello di sicurezza desiderato. Una delle politiche più semplici è quella di aggiornare la chiave periodicamente allo scadere di un timeout oppure si può procedere a fare l'aggiornamento quando l'utente preme un bottone sul nodo coordinatore. L'aggiornamento della chiave avviene grazie a due funzioni; la prima è `ZDSecMgrUpdateNwkKey()` che trasmette in broadcast la nuova chiave, tutti i nodi della rete la memorizzano come chiave alternativa ma continuano ad utilizzare quella vecchia finché il coordinatore non esegue la seconda funzione: `ZDSecMgrSwitchNwkKey()` che con un altro messaggio in broadcast fa passare tutti i nodi della rete all'uso della chiave alternativa.

Capitolo 5

Il sistema realizzato

5.1 Requisiti del sistema

5.1.1 Obiettivi

Il progetto ha come obiettivo principale la realizzazione di un prototipo che usa le reti di sensori wireless (WSN) ZigBee per la localizzazione di oggetti mobili. L'applicazione prevede un certo numero di nodi fissi dislocati all'interno di un area di osservazione o di interesse, questi ultimi cooperano tra di loro per determinare la posizione di un nodo mobile della rete. Il nodo mobile, identificato come l'oggetto da monitorare, si muove nell'area controllata e dialoga con i nodi fissi della rete, consentendo al sistema di stabilire con ragionevole approssimazione la sua posizione. L'interazione tra il nodo mobile e i nodi fissi è di fondamentale importanza per ottenere la stima delle distanze reciproche (basandosi sull'indice RSSI), utili a sua volta per determinare la posizione del nodo mobile mediante l'uso di algoritmi di trilaterazione o di multilaterazione. Lo scopo finale del progetto è la realizzazione di un dimostratore che permetta di verificare sul campo la fattibilità o meno di un sistema di localizzazione di questo tipo. Il prototipo risultante sarà la base per sviluppare sistemi di localizzazione per applicazioni reali con esigenze specifiche, (come per esempio un consumo minimo di energia da parte dei nodi mobili alimentati a batteria o la sicurezza dei dati trasmessi via etere) alle quali, il sistema, dovrà essere adattato in seguito. E' quindi, al momento, fuori dagli obiettivi di questo progetto l'ingegnerizzazione e l'ottimizzazione del sistema per farlo rispondere a requisiti di robustezza ed industrializzazione dettati da eventuali processi produttivi.

5.1.2 Specifiche funzionali

Descrizione generale del sistema

Il sistema deve essere in grado di fornire al personale autorizzato:

- le informazioni di stato e di posizione dell'oggetto in tempo reale

- storico delle informazioni di stato e di posizione indicando il tempo in cui sono state misurate
- la posizione di tutti gli oggetti di un certo tipo in un determinato momento
- informazioni relative a situazioni anomale in termini di stato dell'oggetto o situazione ambientale
- informazioni relative alla WSN, quali stato dei nodi ed anomalie di funzionamento

Il sistema è suddiviso in 4 blocchi funzionali:

1. **WSN**: la rete dei sensori.
2. **Object Controller**: si occupa di gestire ed indirizzare le informazioni da e verso la WSN.
3. **Back-end**: si occupa di memorizzare le informazioni relative agli oggetti da monitorare.
4. **Front-end**: applicazione web, tramite la quale l'utente può visionare le informazioni rilevate dall'intero sistema.

WSN

I nodi della rete di sensori WSN possono essere suddivisi in:

- nodi mobili
- nodi fissi o di riferimento
- nodo gateway

Nodi mobili I nodi mobili WSN vengono collocati in corrispondenza degli oggetti mobili, chiamati anche beni, da localizzare e/o tracciare. Ogni nodo mobile presenta un'interfaccia wireless ZigBee attraverso la quale può comunicare con gli altri nodi della rete, in particolare con i nodi fissi e con il nodo gateway. Poiché l'algoritmo di localizzazione utilizzato per la realizzazione del dimostratore è principalmente basato su stime di distanze ricavate a partire da misure di potenze del segnale radio ricevuto RSSI (*Received Signal Strength Indicator*, indicatore del livello di potenza), ogni nodo mobile della WSN misura i valori di RSSI ad ogni ricezione dei pacchetti trasmessi dai nodi di riferimento vicini (chiamati anche nodi adiacenti). Pertanto ad intervalli di tempo regolari ΔT , i nodi mobili trasmettono verso il nodo gateway, i valori di RSSI con i corrispondenti identificatori dei nodi fissi rispetto i quali è stata effettuata ogni misura. Per realizzare la funzione di tracciabilità dello stato dei beni, tutti i nodi mobili trasmettono verso il GW, ad intervalli di tempo regolari ΔT , anche i valori di stato dei beni monitorati. ΔT può assumere valori diversi a seconda del bene da tracciare, inoltre può essere modificato dal back-end in fase di funzionamento del sistema su richiesta dell'utente tramite front-end. Attraverso l'interfaccia di rete, il nodo mobile può ricevere delle richieste di invio istantaneo dei valori di RSSI o dei sensori.

Nodi di riferimento I nodi fissi WSN, chiamati anche “nodi ancora” o “nodi di riferimento”, vengono collocati all’interno dell’edificio in particolari punti di osservazione, stabiliti in modo tale da ottenere una buona copertura wireless ed in modo da massimizzare la precisione di localizzazione. Una volta che i nodi fissi sono stati posizionati è necessario misurare le loro posizioni e memorizzarle all’interno del blocco Object Controller, il quale, come verrà spiegato più avanti, si occupa di stimare la posizione dei nodi mobili. I nodi fissi inviano su richiesta ai nodi mobili sequenze di pacchetti su cui i nodi mobili faranno le misure di RSSI. I nodi di riferimento non sono associati ad un bene ma bensì ad un luogo, come i nodi mobili essi potranno inviare informazioni di stato che quindi saranno riferite al luogo in cui si trovano (come per esempio: la temperatura, l’intensità luminosa presenti in una data stanza).

Nodo gateway Il nodo gateway è unico all’interno della rete ed è un nodo particolare che funge da tramite tra la rete WSN e l’Object Controller, per questo motivo presenta due interfacce:

- Un’interfaccia wireless ZigBee verso la rete WSN.
- Un’interfaccia seriale RS232 verso il computer che esegue l’Object Controller.

Il gateway si occupa di: ricevere dalla WSN e inoltrare su linea seriale tutti i dati destinati all’Object Controller e viceversa di ricevere i comandi e le configurazioni per i nodi, provenienti dall’Object Controller, su linea seriale e di inoltrarle verso il nodo destinatario nella WSN.

Location Service

Dei blocchi funzionali visti in precedenza: Object Controller, Back-end e Front-end vanno a formare il servizio che sarà eseguito dal computer al quale verrà collegato il cavo seriale proveniente dal gateway.

In pratica Location Service dovrà occuparsi di:

- Ricevere e decodificare i dati provenienti dal gateway su linea seriale.
- Calcolare le posizioni dei nodi mobili a partire da misure di RSSI.
- Memorizzare e gestire tutte le informazioni ricevute in un database.
- Mantenere nel DB uno storico delle posizioni dei nodi mobili e delle misure dei sensori
- Visualizzare, su richiesta, all’utente specifiche informazioni tramite interfaccia web.
- Permettere all’utente di configurare e gestire tutti i nodi della WSN tramite interfaccia web.

Location Service sarà quindi formato da quattro elementi principali:

- **Packet Decoder** si occupa di decodificare/codificare i pacchetti da e verso il gateway su linea seriale.

- **Location Engine** è incaricato di calcolare la posizione di un nodo conoscendo le misure di RSSI rispetto ai nodi fissi nelle sue vicinanze.
- **DB Manager** mantiene aggiornato il database inserendovi le informazioni appena arrivate.
- **Front-end** è l'interfaccia web che permette di interagire con l'intero sistema, anche in remoto, in un linguaggio facilmente comprensibile all'utente.

L'utente tramite interfaccia web potrà:

- Visualizzare l'ultima posizione rilevata dei nodi mobili.
- Richiedere al sistema di localizzare immediatamente uno specifico nodo mobile.
- Visualizzare gli ultimi valori dei sensori rilevati su di un nodo specifico.
- Richiedere ad un nodo di inviare immediatamente i valori dei suoi sensori.
- Visualizzare le localizzazioni di un determinato nodo mobile in un determinato intervallo di tempo.
- Visualizzare l'andamento del valore di uno specifico sensore di un nodo in un un determinato intervallo di tempo.
- Associare un bene da tracciare ad un nodo mobile.
- Configurare ogni nodo della rete WSN.
- Configurare l'elenco delle stanze dell'edificio in cui è installato il sistema.
- Gestire i vari elementi nel DB come nodi fissi, nodi mobili, beni, stanze.

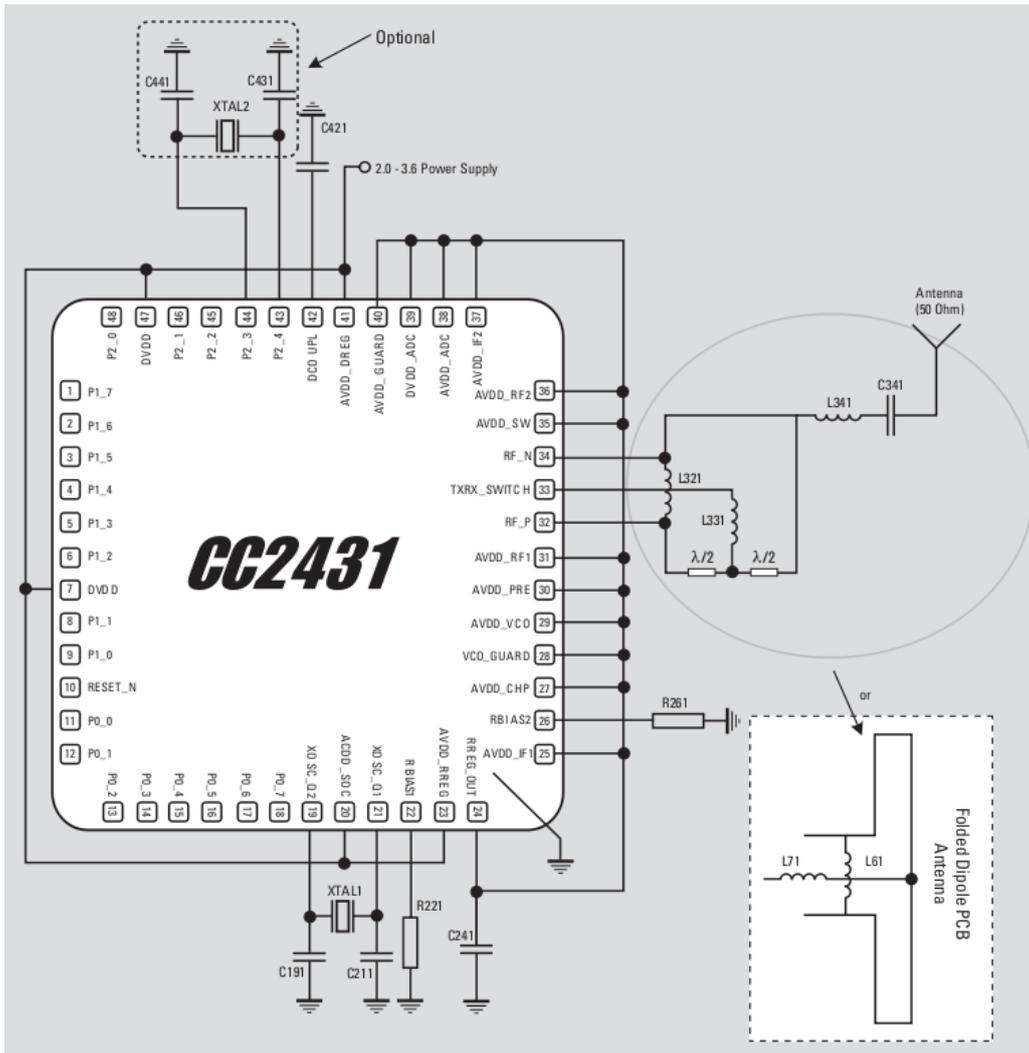


Figura 5.1. Schema dei collegamenti all'integrato CC2431

5.1.3 Specifiche hardware

Il sistema deve innanzitutto rispondere al requisito fondamentale di un basso costo e basso consumo dei dispositivi e questa motivazione ha portato l'azienda a scegliere i dispositivi CC2430 e CC2431 prodotti da ChipCon (ora parte di Texas Instruments). Questa scelta ci costringe ad utilizzare l'indice RSSI per stimare le distanze tra i nodi.

Caratteristiche tecniche

Questi integrati fanno uso al loro interno di una CPU 8051 con 8 KB di RAM, 128 KB di memoria flash non volatile e sul chip troviamo già implementate in hardware molte funzionalità utili come per esempio: un dispositivo di cifratura AES, convertitori ADC, supporto per una porta seriale e accesso ai sensori collegati al dispositivo. Inoltre la versione CC2431 rispetto alla versione CC2430 possiede un motore di localizzazione hardware interno basato sull'indice RSSI.

Parametro	Valore	Condizioni
Temperatura d'esercizio	da -40 a 85 °C	
Tensione d'alimentazione	da 2,0 a 3,6 V	
Consumo in ricezione	27 mA	Con MCU a velocità massima (32 MHz)
Consumo in trasmissione	25 mA	Con MCU a velocità massima (32 MHz)
Consumo in power mode 2	0,9 μ A	Timer di sleep attivo
Consumo in power mode 3	0,6 μ A	Wake-up solo su interrupt esterni
Tempo di wake-up	54 μ s	
Tempo commut. RX-TX	450 μ s	
Intervallo frequenze RF	2400-2483,5 MHz	Programmabile in step di 1 MHz
Radio bit rate	250 kbps	Come da specifiche IEEE 802.15.4
Sensibilità ricevitore	-94 dBm	
Potenza d'uscita nominale	0 dBm	In trasmissione
Errore location engine	< 3 m	Motore di localizzazione hardware

Tabella 5.1. Tabella riassuntiva dei sistemi delle principali caratteristiche dei dispositivi ChipCon 2431

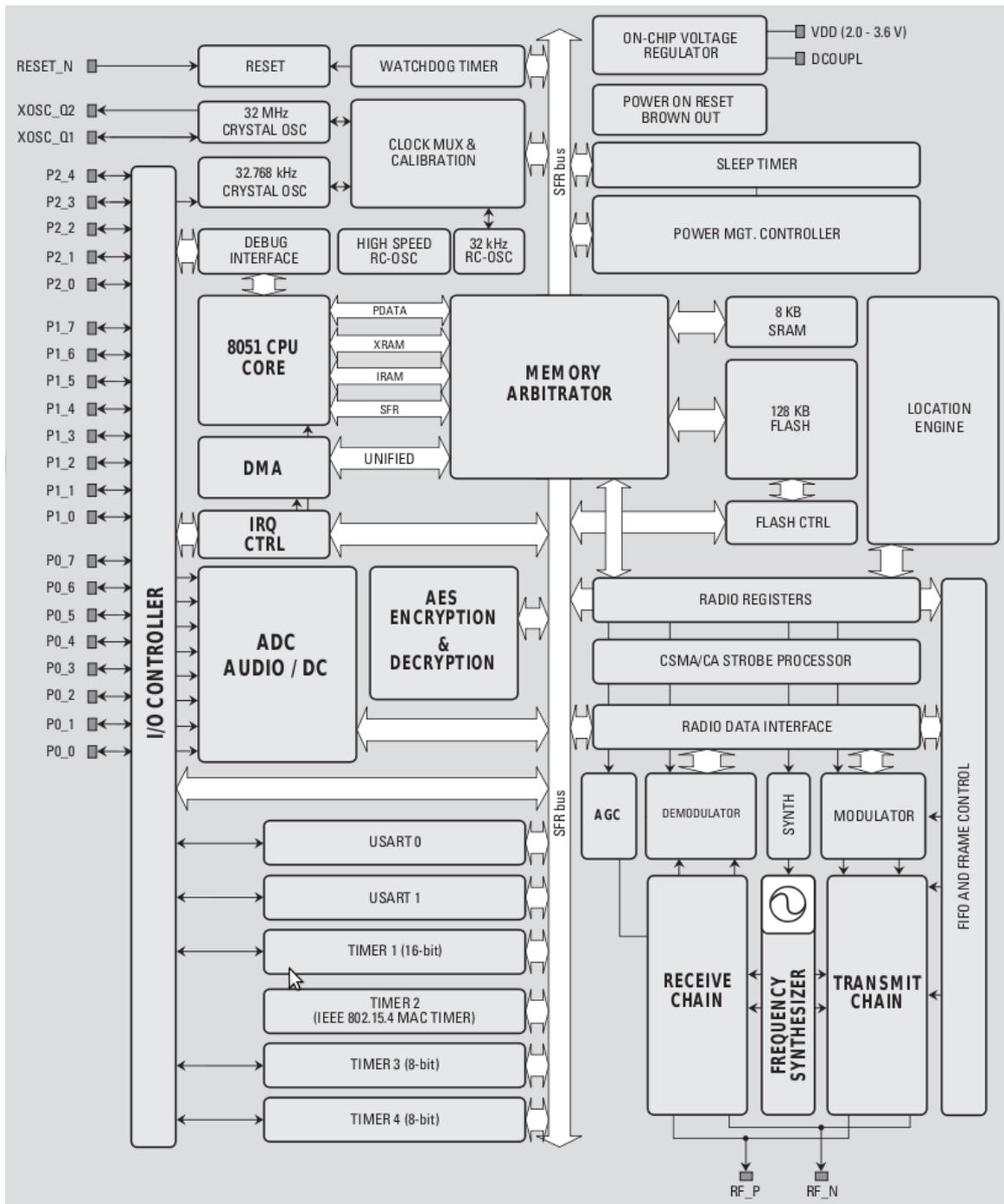


Figura 5.2. Diagramma a blocchi dell'integrato CC2431

Hardware utilizzato

Texas Instruments fornisce dei kit di valutazione che permettono agli sviluppatori di iniziare lo sviluppo di applicazioni prototipo per CC2430/CC2431. Per questo progetto si è usato il kit CC2431 ZDK Pro. Tale kit comprende una scheda SmartRF04EB (*Evaluation Board*), 2 CC2431DB (*Demonstration Board*) e 4 CC2431BB (*Battery Board*).

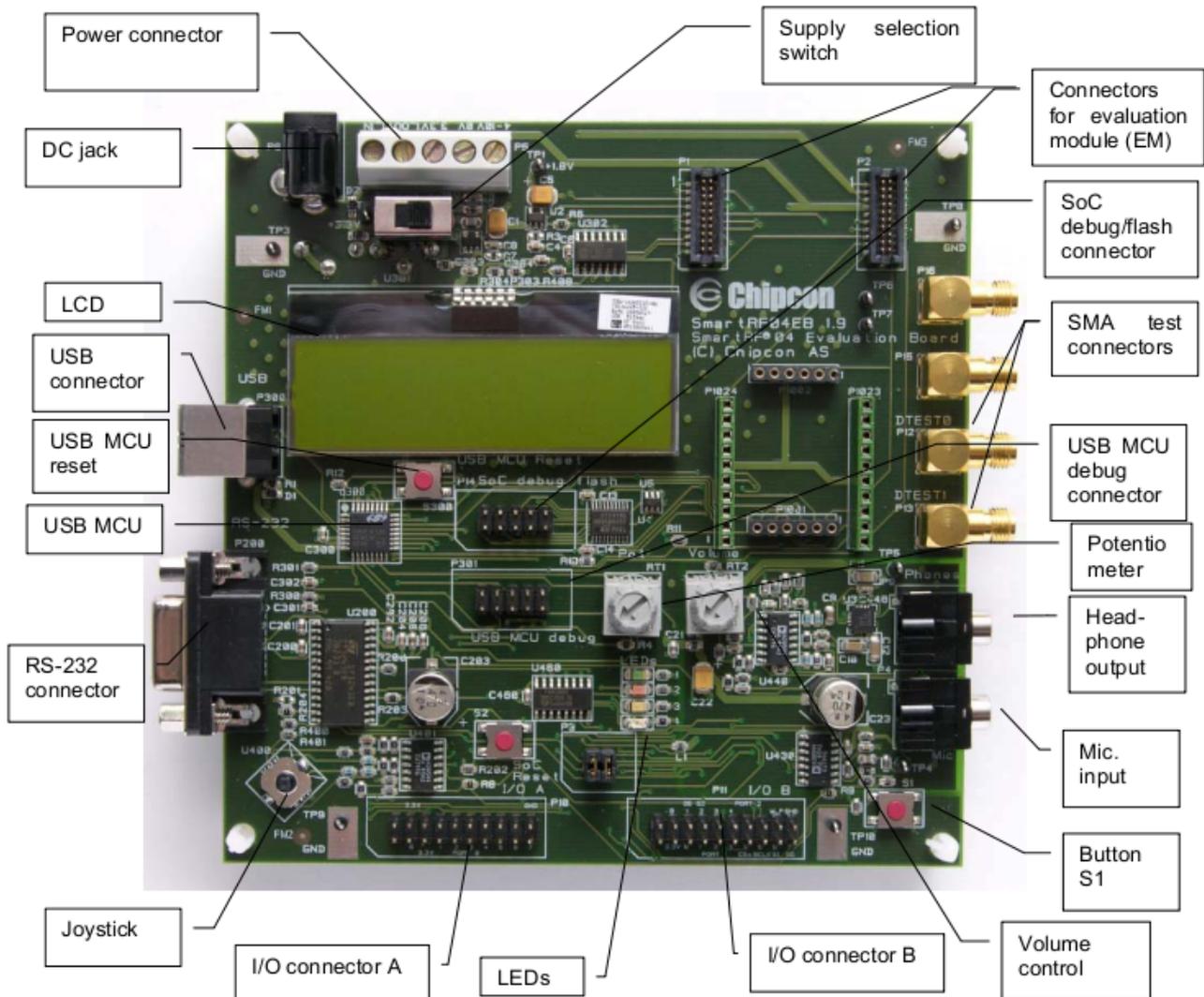


Figura 5.3. Dettaglio della scheda SmartRF04EB. In alto a destra si può notare l'alloggiamento dove deve essere inserito il modulo CC2430EM o CC2431EM che comprende l'integrato CC2430 o CC2431 con l'antenna. Tramite porta USB questa scheda può essere usata per programmare le memorie flash dei moduli che vi vengono inseriti. Sotto tale scheda c'è un alloggiamento per una batteria da 9 V.

Evalutation Board

E' la scheda principale, può essere alimentata in svariati modi: con una batteria da 9 V, con un'alimentatore da 4 V a 10 V collegato agli appositi morsetti oppure con il 5 V presente sulla connessione USB. Ha a bordo numerose caratteristiche aggiuntive:

- Interfaccia USB
- Interfaccia seriale RS-232
- Input devices: un joystick, due push button e due potenziometri
- Output devices: 4 led ed un display da 2 righe di 16 caratteri
- Interfaccia audio: un jack per microfono ed un jack per altoparlante

Si è deciso di utilizzare questa scheda come coordinatore di rete e come gateway, ovvero come punto di comunicazione della rete ZigBee con il mondo esterno grazie all'interfaccia seriale.

Demostration Board

Anche questo dispositivo dispone di una connessione USB e di un joystick, dispone anche di due push button e di due led. Questa scheda può essere alimentata con due batterie di tipo AA, attraverso un alimentatore da 4 a 12 V collegato all'apposito jack oppure tramite connessione USB, la piastra ha a bordo alcuni sensori:

- Un accelerometro su 2 assi
- Un sensore di temperatura
- Un potenziometro
- Un sensore di luce

Questa scheda ha a bordo in modo permanente l'integrato CC2430 non può quindi disporre del motore hardware di localizzazione, che invece è presente sul CC2431. L'antenna integrata sulla piastra non permette di fare buone misure per la localizzazione, per il fatto che non è omnidirezionale, per ovviare a questo problema si è deciso di usare questa scheda come punto di riferimento singolo per piccole stanze in cui per localizzare un oggetto è sufficiente sapere che si trova in quella stanza facendo a meno delle coordinate. Per le stanze piccole dove non è necessario avere una localizzazione con le coordinate (calcolate con un algoritmo di localizzazione) è sufficiente un solo nodo di riferimento per stanza.

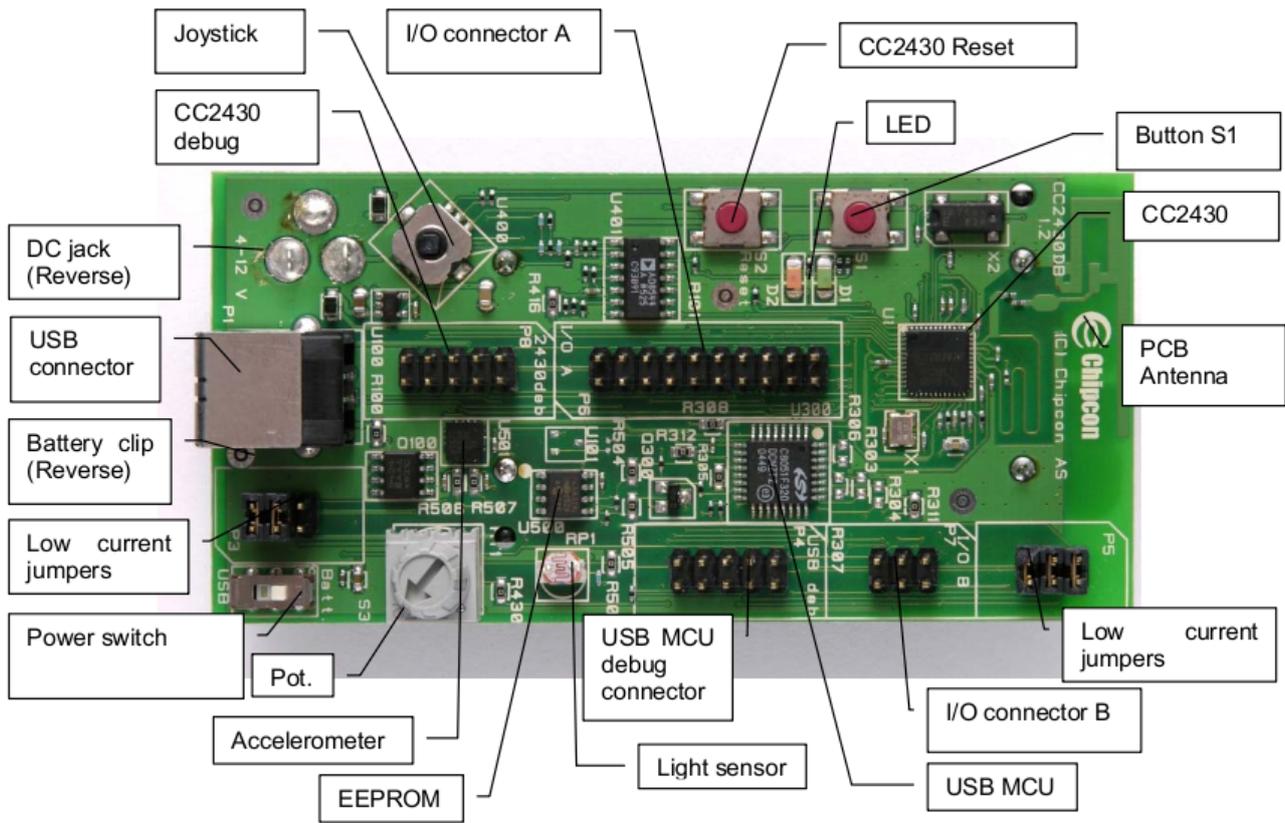


Figura 5.4. Dettaglio della Demostration Board. Sulla scheda vi è direttamente saldato un CC2430, da notare sulla destra l'antenna integrata sulla piastra. Sotto la scheda vi è l'alloggiamento per due batterie di tipo AA.

Battery Board

E' la scheda più semplice, l'alimentazione è fornita esclusivamente da due batterie di tipo AA alloggiato sotto la piastra. Oltre all'alloggiamento per il modulo CC2430EM o CC2431EM troviamo sulla scheda un led e un push button. L'antenna è quella già presente sui moduli ed è decisamente migliore (omnidirezionale) di quella della *Demonstration Board*. Il dispositivo non dispone di una connessione USB e quindi per programmare i moduli bisogna ricorrere all'*Evaluation Board*. Nel prototipo si è scelto di usare queste schede con il modulo CC2431EM per i nodi mobili da localizzare, quindi anche con l'aiuto del motore di localizzazione hardware. Queste schede vanno bene anche per essere usate come nodi di riferimento per ambienti più vasti ove è necessaria una localizzazione con le coordinate.

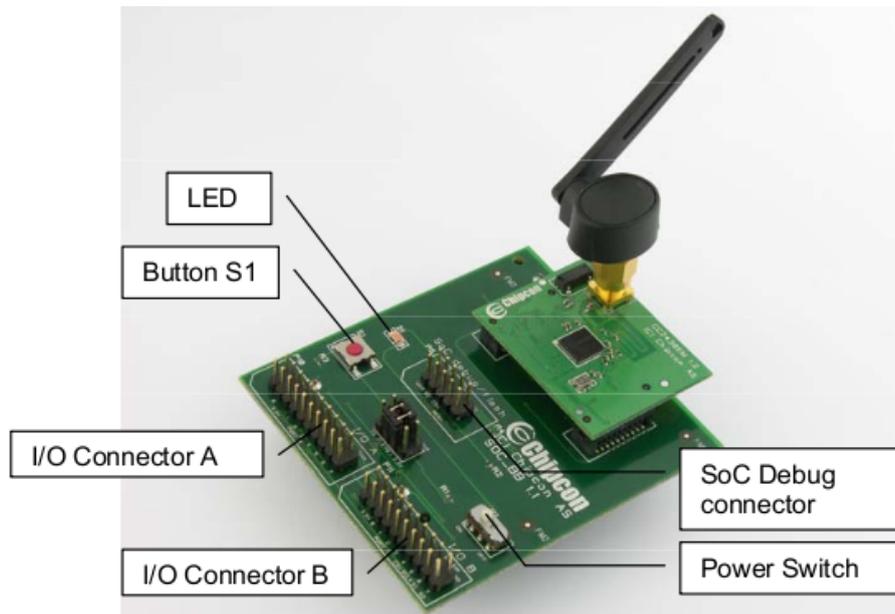


Figura 5.5. Dettaglio della Battery Board. In questo caso sulla scheda vi è installato un modulo CC2430EM con la sua antenna. Sotto la piastra vi è l'alloggiamento per due batterie di tipo AA.

5.1.4 Possibili applicazioni

Presso l'azienda sono stati previsti da subito due possibili applicazioni per il prototipo realizzato.

Ambito ospedaliero

Particolare interesse si riserva ad applicazioni di questa natura in ambito ospedaliero dove, nel caso di alcuni oggetti particolarmente critici nel processo clinico, può essere di grande importanza conoscere la loro localizzazione all'interno della struttura e il loro stato all'interno del processo. Alcuni esempi:

- Bombole di gas medicale: dove sono? Sono cariche o vuote?
- Defibrillatori: dove sono? Sono in ricarica? Sono in manutenzione?
- Accessori di reparto condivisi: letti automatici, apparecchi per disabilità: dove sono? Sono liberi? Sono prenotati? Da chi? Quando si libera il primo?

I nodi mobili del sistema verrebbero quindi associati ad apparecchiature biomedicali, il sistema permetterebbe di realizzare:

- la rintracciabilità delle apparecchiature mobili ed attrezzature di supporto

- la manutenzione programmata delle apparecchiature elettromedicali
- il controllo delle variazioni di stato dei beni
- il controllo delle condizioni ambientali in cui i beni si trovano

In questo caso si avrà un calcolo della posizione decisamente più incentrato su una localizzazione per ambienti del tipo: l'oggetto richiesto si trova in una data stanza dell'edificio.

Ambito portuale

Un'altra possibile applicazione è data dalla necessità di localizzare le merci scaricate dalle navi in un porto. Le merci possono essere container o autovetture importate. In questo caso il sistema lavorerà in ambienti esterni molto più vasti. Per poter fare le misure di RSSI in modo più accurato ed avere una copertura più vasta saranno quindi necessarie antenne di migliore qualità, non integrate sulle schede dei dispositivi. L'algoritmo di localizzazione sarà decisamente più incentrato sul calcolo delle coordinate che poi dovranno venire tradotte in un'informazione più facilmente comprensibile da chi dovrà andare fisicamente a recuperare le merci, come per esempio un numero di parcheggio.

5.2 Architetture

L'implementazione del sistema è suddivisa in due parti: la scrittura del firmware per i moduli ZigBee e lo sviluppo di un servizio per il PC collegato al gateway.

5.2.1 Firmware

Per quanto riguarda il firmware non ci sono molte scelte: deve essere scritto in C e compilato con il compilatore *IAR Systems*. Si è deciso di usare lo stack ZigBee 1.4.3 fornito da Texas Instruments dato che rispetto all'altra versione provata, la 1.4.0, permette una migliore gestione della memoria non volatile. Per poter compilare correttamente programmi usando lo Z-stack 1.4.3 è necessario usare la versione 7.30B del compilatore IAR.

5.2.2 Servizio

Invece, per quanto riguarda il servizio si è deciso di scriverlo in java in modo da poter avere un unico programma che si occupa di decodificare le informazioni ricevute da linea seriale, mantenere aggiornato il DB e rendere accessibile il tutto all'utente tramite una web-application. Il lato client è costituito da servlet java che costruiscono dinamicamente le pagine HTML che verranno visualizzate dall'utente. Si è scelto di usare il server Tomcat versione 6.0 per eseguire e rendere disponibile il servizio ai client. Queste scelte sono state dettate dal desiderio dell'azienda di evitare all'utente l'installazione di qualsiasi software (gli sarà necessario solo un browser), inoltre questa soluzione ha una grande portabilità anche dal lato server in quanto può essere

installata sia su macchine Linux che Windows. Per sviluppare il servizio si è usata l'IDE *Eclipse Enterprise Edition* che mette a disposizione tutta una serie di funzionalità, molto utili nel nostro caso, per lo sviluppo e il deploy di servlet java.

Librerie aggiuntive

Per poter compilare correttamente il servizio è necessario referenziare le seguenti librerie:

- **Jama**, *Java Matrix*: per eseguire calcoli matriciali in Java.
- **MySQL Connector**: per connettersi al database.
- **javax.comm**: per poter accedere alla porta seriale.

5.2.3 Database

Per il database si è scelto *MySQL* che anche in questo caso assicura portabilità sia su macchine Linux che Windows. Il database può essere eseguito sulla stessa macchina che esegue il servizio oppure su di una differente alla quale il servizio possa collegarsi. E' necessario creare uno schema di nome: `locationdb` e l'utente `locationService` che verranno utilizzati dal servizio per accedere alle tabelle e mantenerle aggiornate.

5.3 Implementazione

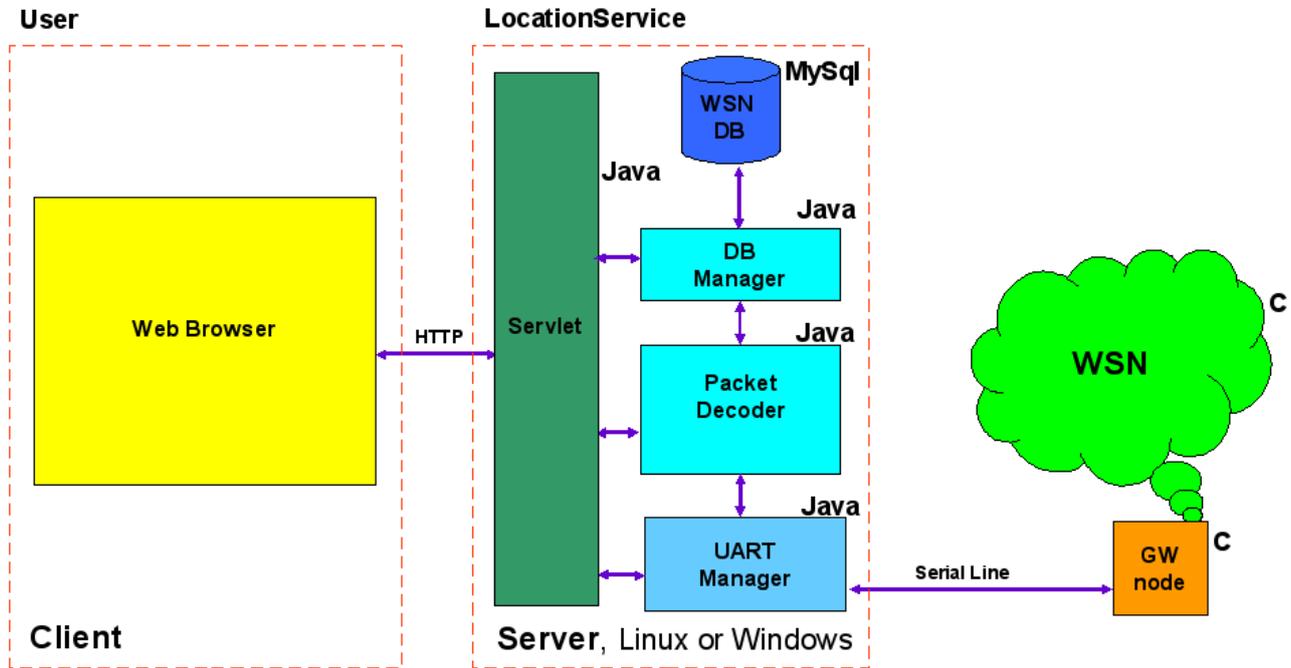


Figura 5.6. Rappresentazione schematica dell'intero sistema realizzato. I due rettangoli tratteggiati in rosso racchiudono tutte le componenti che saranno eseguite rispettivamente sul PC dell'utente e sul server. In basso a destra il gateway che si interfaccia tra la linea seriale e il resto della rete di sensori WSN. Per ogni elemento sviluppato è indicato il linguaggio in cui è stato scritto. Come si può notare il client non è altro che un normale PC in grado di connettersi al server utilizzando un browser.

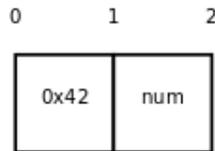
5.3.1 Firmware

Sui moduli ZigBee si ha un tipo di programmazione *event-driven* ovvero guidata dagli eventi, non c'è un utente che impartisce dei comandi ma bensì è il programma che esegue determinate operazioni a seconda di quello che succede. Immaginando di scrivere un nuovo programma con lo stack ZigBee, la prima cosa da fare è una funzione incaricata di processare i vari eventi. Ogni qualvolta si scatena un evento, lo stack richiama tale funzione passandogli l'identificativo dell'evento appena accaduto. Alcuni esempi di eventi sono: il ricevimento di un pacchetto via etere, il ricevimento di una sequenza di byte sulla linea seriale, lo scadere di un timer, un pulsante che viene premuto ecc.

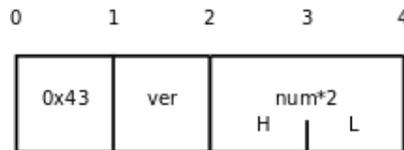
Gateway

Come già detto il gateway si occupa di fare da tramite tra la porta seriale del PC su cui sarà in ascolto il servizio e la rete ZigBee, tutti i nodi della rete che necessitano di inviare un pacchetto

al servizio lo inoltrano verso il gateway. Si è pensato di implementare la possibilità di fare una sorta di *ping* al gateway tramite porta seriale per verificarne il funzionamento, così che il servizio abbia la possibilità di capire se si è collegato alla porta seriale corretta. Per fare il *ping* del gateway occorre inviargli su linea seriale due byte il primo deve essere uguale a `0x42` mentre il secondo è un numero casuale:

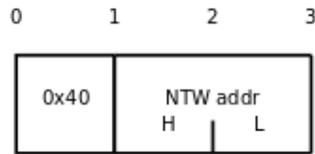


Il gateway risponde nel modo seguente:

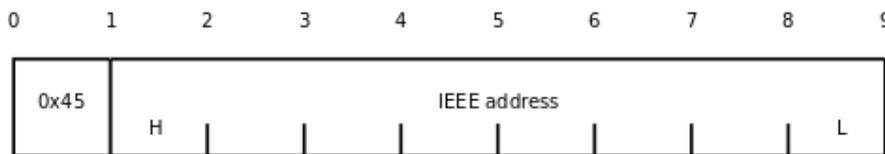


Il primo byte di tutte le sequenze da e per il gateway su linea seriale identifica sempre il tipo di pacchetto: `0x43` identifica la risposta ad una richiesta di *ping*, `ver` è un numero codificato come `unsigned int` su un byte indicante la versione del firmware in esecuzione sul gateway, mentre `num*2` non è altro che il numero casuale di prima moltiplicato per 2. `num*2` è codificato su 2 byte come `unsigned int` su 16 bit i cui primi 8 sono la parte più significativa. A seconda della risposta, il servizio potrà assumere o meno di essere collegato al gateway con la versione del firmware desiderata. Di tutti i pacchetti ricevuti dal gateway sull'interfaccia ZigBee solo alcuni sono effettivamente destinati alla linea seriale, il gateway si occupa quindi di selezionarli tramite *clusterID*. Nel momento in cui il gateway ha ricevuto un pacchetto da inoltrare sulla linea seriale innanzitutto ne estrae il campo *data*: la sequenza di byte così ottenuta non può ancora essere inviata alla porta seriale, occorre aggiungere le informazioni su che tipo di pacchetto si tratta e sul mittente. Alla sequenza di byte viene quindi anteposto una sorta di *header* di 3 byte; il primo viene ricavato con gli 8 bit meno significativi del *clusterID*, seguito da altri due byte che rappresentano l'indirizzo del mittente in formato `unsigned int` su 16 bit dove il primo byte rappresenta gli 8 bit più significativi. A questo punto la sequenza di byte viene delimitata con i caratteri di *BEGIN* e *END* uguali rispettivamente a `0x3C`, `0x3E`; se all'interno della sequenza capitano i caratteri `3C` o `3E` vengono fatti precedere dal carattere *ESCAPE* pari a `0x2F`, se capita di dover trasmettere `2F` lo si ripete due volte. Così delimitata la sequenza di byte può essere trasmessa sulla linea seriale. Grazie ai delimitatori il servizio potrà quindi distinguere tra un pacchetto e l'altro. I pacchetti che invece vengono trasmessi dal servizio su linea seriale verso il gateway devono essere ritrasmessi tutti, eccetto la richiesta di *ping*, verso la rete di sensori. I pacchetti inviati dal servizio hanno lo stesso formato appena visto: il primo byte ne identifica il tipo, i due byte successivi rappresentano lo *short address* del destinatario, segue poi la sequenza di byte da inviare verso la WSN. Per ogni sequenza di byte ricevuta su linea seriale il gateway costruisce un pacchetto ZigBee, a seconda dei dati contenuti nell'*header* sceglierà

quale *clusterID* usare, il tipo di indirizzamento e l'indirizzo del destinatario. Il campo *data* viene riempito con byte restanti della sequenza appena ricevuta e a questo punto il pacchetto è pronto per essere inviato via etere.



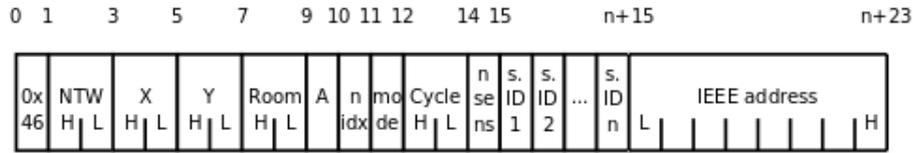
Questa è una richiesta di annuncio per il nodo della rete con *short address* pari a *NTW addr*, tale richiesta è identificata con 0x40, è solo un *header* senza altri dati, infatti il gateway in questo caso costruirà un pacchetto ZigBee con il campo *data* vuoto, con *clusterID* impostato a 0x0040, indirizzato verso *NTW addr* in *unicast*. Dato che spegnendo e riaccendendo i nodi può capitare che cambino *short address* si è deciso di implementare la richiesta di annuncio tramite *long address* che rimane sempre invariato:



Questa richiesta, identificata con 0x45 è l'unica che al primo byte del tipo viene fatto seguire l'*IEEE address* su 8 byte anziché il *network address*. Il gateway traduce tale richiesta in un pacchetto ZigBee con *clusterID* uguale a 0x0045, contenente l'*IEEE address* su 8 byte nel campo *data* che viene inviato in *broadcast*, il nodo con quell'*IEEE address*, se presente in rete, risponderà con il suo annuncio. Grazie a questa richiesta il servizio appena avviato può scoprire i *network address* dei nodi presenti in rete a partire da un elenco di *IEEE address*. In realtà lo stack mette a disposizione anche altre possibilità per risolvere il *network address* di un nodo di cui si conosce solo l'*IEEE address* ma qui si voleva provare a trasmettere in *broadcast*.

Reference Node

Si presume che i nodi di riferimento siano installati in posizioni note e statiche, con a disposizione fonti di alimentazione permanenti. Si è quindi deciso di impostare i nodi di riferimento come router, quindi essi saranno sempre attivi ed incaricati di smistare e trasportare il traffico di rete. Di conseguenza sono i nodi di riferimento a portare la copertura all'intera zona che si vuole monitorare, dovranno essere installati in modo da poter sempre comunicare direttamente con almeno un altro nodo di riferimento ed almeno un nodo di riferimento dovrà essere in grado di comunicare direttamente con il gateway, altrimenti il sistema non potrebbe funzionare. Appena acceso, o se ne riceve la richiesta, il nodo di riferimento invia al gateway il pacchetto di annuncio, che ha come *clusterID* 0x0046. Questa è la sequenza di byte risultante in uscita dal gateway verso la linea seriale:



Dove:

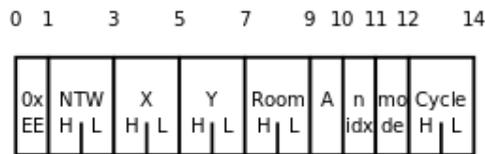
- 0x46 è l'identificatore di pacchetto di annuncio per i nodi di riferimento, viene aggiunto dal gateway.
- NTW è lo *short address* del mittente, aggiunto dal gateway.
- X è l'ascissa configurata sul nodo.
- Y è l'ordinata configurata sul nodo.
- Room è l'identificativo della stanza di appartenenza.
- A è il parametro A (in dBm senza segno) relativo alla stanza di appartenenza.
- $n\ idx$ è un indice che rappresenta il valore del parametro n relativo alla stanza di appartenenza.
- $mode$ può essere 0 o 1, indica se il nodo è in modalità automatica.
- $Cycle$ è il tempo in decimi di secondo, allo scadere del quale, il nodo in modalità automatica, invia periodicamente i valori dei propri sensori.
- $n\ sens$ è il numero di sensori a bordo.
- $s.\ ID$ sono gli identificatori dei sensori presenti sul nodo, questo campo è ripetuto per ogni sensore presente sul nodo.
- $IEEE\ address$ è il *long address* del nodo in questo caso i primi 8 bit sono quelli meno significativi.

X e Y sono numeri `unsigned int` su 16 bit che rappresentano le coordinate assegnate al nodo di riferimento, sono espresse in quarti di metro, quindi dato che il massimo numero intero rappresentabile su 16 bit è $2^{16} = 65536$, le coordinate non potranno essere maggiori di $65536/4 = 16384\ m$ e nemmeno negative.

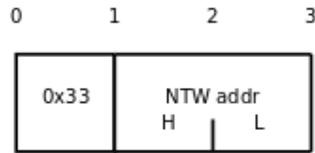
n_index	n	n_index	n
0	1,000	16	3,375
1	1,250	17	3,500
2	1,500	18	3,625
3	1,750	19	3,750
4	1,875	20	3,875
5	2,000	21	4,000
6	2,125	22	4,125
7	2,250	23	4,250
8	2,375	24	4,375
9	2,500	25	4,500
10	2,625	26	4,625
11	2,750	27	5,000
12	2,875	28	5,500
13	3,000	29	6,000
14	3,125	30	7,000
15	3,250	31	8,000

Tabella 5.2. Tabella de valori del parametro n assegnati all'indice `n_index`. Tale indice è un numero intero compreso tra 0 e 31.

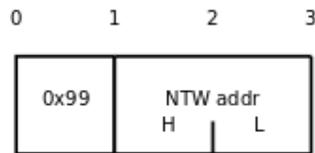
Quando il servizio riceverà l'annuncio del *reference node* potrà verificare la corrispondenza tra *network address* e *IEEE address* del nodo ed eventualmente aggiornare l'indirizzo di rete, inoltre potrà verificare che le impostazioni del nodo corrispondano a quelle presenti nel database. Il servizio può configurare uno specifico nodo di riferimento inviando su linea seriale il seguente pacchetto:



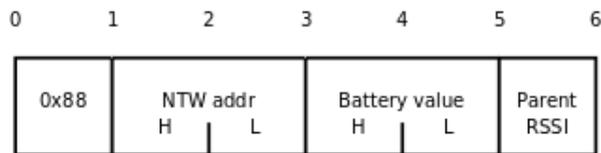
0xEE è l'identificatore per i pacchetti di configurazione per i nodi di riferimento, mentre NTW è lo *short address* del nodo di riferimento da configurare, seguono tutti gli altri dati da configurare sul nodo nello stesso formato visto nel pacchetto di annuncio. Il gateway, utilizzando 0x00EE come *clusterID*, inoltra tale pacchetto al destinatario che provvederà ad aggiornare la propria configurazione anche in memoria non volatile in modo che alla prossima riaccensione non necessiti di essere riconfigurato. Dopo essersi riconfigurato il *reference node* risponde al gateway con un pacchetto di *acknowledge* per indicare che ha recepito la nuova configurazione:



Dove `NTW addr` è l'indirizzo di rete del nodo di riferimento a cui si vuole richiedere i valori correnti dei sensori. Un'altra richiesta che è possibile fare ad un nodo di riferimento, ma anche ad un nodo mobile, è la richiesta di diagnostica:



A cui corrisponde la seguente risposta:

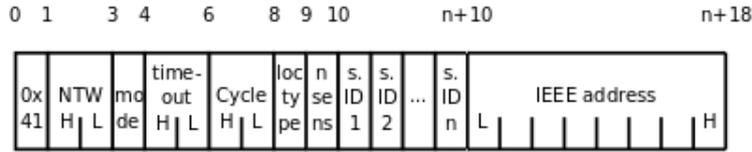


`Battery value` non è altro che il valore letto dal sensore di tensione d'alimentazione, mentre `Parent RSSI` è l'RSSI sull'ultimo *hop* del percorso in rete verso quel nodo. Grazie a questo pacchetto è possibile monitorare lo stato delle batterie dei sensori e se sono sotto una buona copertura.

Blind Node

Il nodo mobile è stato configurato come *end-device*, lo si è immaginato come dispositivo che viene applicato ai beni da monitorare e quindi dovrà essere alimentato a batteria, di conseguenza dovrà consumare il meno possibile. In modo simile ai nodi di riferimento, sui nodi mobili, se vengono impostati in modalità automatica, vi è un timer che permette di inviare periodicamente le misure di RSSI rispetto ai nodi di riferimento nei paraggi insieme con i valori dei sensori a bordo del nodo mobile stesso. La differenza principale rispetto ai nodi di riferimento è che prima dello scadere del timer i nodi mobili sono in stato di *sleep* proprio per poter mantenere l'interfaccia radio spenta e quindi consumare il meno possibile. Il allo scadere del timer (tale tempo viene indicato con `cycle` nel codice) il nodo mobile si risveglia e fa partire un altro timer più breve (tempo indicato nel codice come `timeout`) quindi inizia a fare le misure di RSSI e resta in attesa delle risposte dai reference node nei paraggi. Per tutta la durata del secondo timer l'interfaccia radio è quindi accesa, il *blind node* può anche ricevere messaggi a lui destinati, che gli erano stati spediti mentre era in stato di *sleep*, tali messaggi vengono inoltrati dall'ultimo router che l'avevano visto attivo. Chiaramente `timeout` deve essere molto più breve di `cycle`: un `timeout` di 5 secondi è più che sufficiente, mentre `cycle` potrebbe avere una

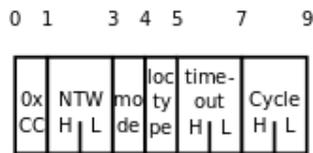
durata di anche un'ora. Anche i nodi mobili, esattamente come i nodi di riferimento, appena vengono accesi, oppure su richiesta, si annunciano con questo pacchetto:



Dove:

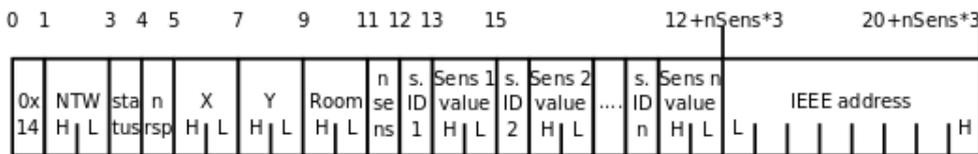
- 0x41 è l'identificatore di pacchetto di annuncio per i nodi mobili, viene aggiunto dal gateway.
- NTW è lo *short address* del mittente, aggiunto dal gateway.
- mode può essere 0 o 1, indica se il nodo è in modalità automatica.
- timeout è il valore di timeout, espresso in decimi di secondo.
- Cycle è il valore di cycle, espresso in decimi di secondo.
- loc type può essere 0 o 1, indica se si sta usando il motore di localizzazione hardware presente sui nodi piuttosto che inviare le misure di RSSI.
- n sens è il numero di sensori a bordo.
- s. ID sono gli identificatori dei sensori presenti sul nodo, questo campo è ripetuto per ogni sensore presente sul nodo.
- IEEE address è il *long address* del nodo in questo caso i primi 8 bit sono quelli meno significativi.

Vediamo ora la composizione del pacchetto di configurazione per i nodi mobili:



0xCC identifica il pacchetto di configurazione per i nodi mobili, con cui è possibile configurare se il nodo dovrà lavorare in modalità automatica, se dovrà fare uso del motore di localizzazione a bordo dei nodi e le tempistiche *timeout* e *cycle*. Dopo essersi configurato il nodo mobile risponde con un *acknowledge*:

Non è possibile fare tale richiesta in modalità automatica, in quanto il *blind node* la riceverebbe solo al suo risveglio, momento in cui deve già svolgere le misure. Se invece, il nodo mobile viene configurato per calcolarsi autonomamente la sua posizione, facendo uso del motore di localizzazione hardware presente sui moduli CC2431, al posto di inviare le misure al gateway le carica nel motore di localizzazione per il calcolo della posizione. Come visto precedentemente ogni nodo di riferimento deve essere configurato oltre che con la sua posizione ma anche con i parametri A ed n dell'ambiente in cui si trova, questi parametri sono richiesti anche dal motore di localizzazione. Nelle risposte dei *reference node* che trasportano le misure RSSI vi sono anche i parametri A ed n configurati su quel nodo. Il nodo mobile assume di trovarsi nella stessa stanza del nodo di riferimento che ha risposto con l'RSSI più alto, di conseguenza caricherà nel motore hardware solo le misure dei nodi di riferimento presenti in quella stanza con i relativi parametri A ed n . Non avrebbe senso tener conto anche degli altri nodi di riferimento, rilevati molto lontani, solo perché i loro segnali vengono attenuati dai muri. Il motore hardware può funzionare solo se si hanno a disposizione almeno 3 nodi di riferimento, se non ci sono almeno tre nodi di riferimento nella stanza in cui ci si trova il calcolo di localizzazione non viene avviato e come posizione stimata viene indicata solo la stanza. Al termine del calcolo viene inviata la posizione stimata con la seguente sequenza di byte:

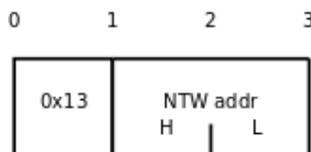


Dopo l'*header* aggiunto dal gateway, si ha:

- **status**: può essere 0 o 1 ed indica se il calcolo di localizzazione è andato a buon fine.
- **n rsp**: indica il numero di *reference node* usati per il calcolo di localizzazione.
- **X e Y**: sono le coordinate della posizione stimata espresse in quarti di metro.
- **Room**: l'identificatore della stanza in cui il nodo mobile assume di trovarsi.

Come nel pacchetto RSSI segue l'elenco dei valori dei sensori presenti sul *blind node* ed infine il *long address* codificato in modalità *bigendian*.

In *polled mode* il pacchetto XY appena visto può essere richiesto in tal modo:



ID	ClusterID	Descrizione	Da:	A:
0x42	-	Richiesta ping gateway	Servizio	GW
0x43	-	Risposta ping gateway	GW	Servizio
0x40	0x0040	Richiesta annuncio con <i>short address</i>	Servizio	RN o BN
0x45	0x0045	Richiesta annuncio con <i>long address</i>	Servizio	RN o BN
0x46	0x0046	Annuncio <i>reference node</i>	RN	Servizio
0x41	0x0041	Annuncio <i>blind node</i>	BN	Servizio
0xEE	0x00EE	Invio configurazione <i>reference node</i>	Servizio	RN
0x11	0x0011	ACK configurazione ricevuta <i>reference node</i>	RN	Servizio
0xCC	0x00CC	Invio configurazione <i>blind node</i>	Servizio	BN
0xDD	0x00DD	ACK configurazione ricevuta <i>blind node</i>	BN	Servizio
0x21	0x0021	Richiesta misure RSSI e sensori	Servizio	BN
0x22	0x0022	Misure RSSI e sensori	BN	Servizio
0x13	0x0013	Richiesta localizzazione XY	Servizio	BN
0x14	0x0014	Stima posizione XY	BN	Servizio
0x33	0x0033	Richiesta valori sensori	Servizio	RN
0x44	0x0044	Valori sensori	RN	Servizio
0x99	0x0099	Richiesta valori diagnostica	Servizio	RN o BN
0x88	0x0088	Valori diagnostica	RN o BN	Servizio

Tabella 5.4. Tabella riassuntiva di tutti i possibili pacchetti scambiati tra il servizio e la rete di sensori.

5.3.2 Servizio

Essendo scritto in java, il servizio vede le varie componenti del sistema come oggetti: si ha l'oggetto `Node` che rappresenta tutti i *reference node*, l'oggetto `BlindNode` che rappresenta i nodi mobili, è un'estensione dell'oggetto `Node`, l'oggetto `Room` che descrive le varie stanze, l'oggetto `Good` che può essere associato ad un `BlindNode` descrive il bene monitorato da tale nodo, infine un oggetto `Sensor` che descrive un singolo sensore. Ogni oggetto `Node` o `BlindNode` potrà referenziare più oggetti `Sensor` che descrivono i sensori che hanno a bordo. Inoltre vi sono tutta una serie di oggetti che descrivono i pacchetti ricevuti dal gateway, ogni volta che viene ricevuta una sequenza di byte dalla porta seriale, con l'*header* viene costruito un'oggetto `Packet` che poi viene esteso con l'oggetto del tipo di pacchetto corrispondente, utilizzando le tutte altre informazioni che seguono l'*header*. Appena avviato il servizio verifica di essere collegato sulla porta seriale al gateway con una richiesta di *ping* e quindi verifica l'esistenza delle tabelle nel database. Il servizio mantiene tutti gli oggetti in memoria facendo uso di *hash map*, questo garantisce velocità nel recuperare i dati, ogni volta che i dati vengono modificati il servizio provvede ad aggiornare anche il database. Il database contiene una tabella per ogni oggetto utilizzato dal servizio: per esempio c'è una tabella dei nodi, una delle stanze, una dei beni monitorati ecc. Il servizio legge dal database prevalentemente in fase di avvio per caricare tutti i dati del sistema. Invece per quanto riguarda lo storico delle localizzazioni e dei valori

dei sensori, tali informazioni vengono lette direttamente dal database ogni volta che vengono richieste. Dopo aver caricato l'elenco dei nodi di riferimento dal database, il servizio verifica che siano effettivamente attivi richiedendo a ciascun nodo di riferimento di annunciarsi. In fase di inizializzazione, gli annunci vengono richiesti con l'*IEEE address* poiché i nodi potrebbero aver cambiato *network address* (se sono stati riaccesi) dall'ultima esecuzione del servizio. Invece per quanto riguarda i nodi mobili vengono considerati attivi appena si riceve il loro il primo pacchetto, dato che solitamente lavorano in modalità automatica. Una volta che il sistema è avviato l'utente, tramite servlet può elencare i nodi e configurarli: quando viene modificata una configurazione di un nodo il sistema invia la richiesta al gateway dopodiché aggiornerà i dati di quel nodo nell'*hash map* e nel database solo quando riceverà l'*acknowledge* corrispondente dal gateway.

RefNodes List

List of all RefNodes

Name	IEEEaddress	NTWaddr	Location	Position	Vbatt	Edit
A8	00124b00000505be	21587	Prato	10.00 ; 17.25	2.57	Details
A6	00124b00000505bc	25907	Prato	20.00 ; 0.00	2.57	Details
A5	00124b00000505b7	20726	Prato	0.00 ; 0.00	2.58	Details
D1	00124b000001086b	25906	Garage	10.00 ; 5.75	2.57	Details
D2	00124b0000010872	20725	Ufficio	10.00 ; 1.25	2.56	Details

Operations on all reference nodes

Request diagnostic for all reference nodes

Get all diagnostic

Figura 5.7. La schermata della *web-application* nella quale vengono elencati tutti i nodi di riferimento presenti nel sistema, cliccando sui bottoni “*Details*” è possibile accedere alla pagina di configurazione di ciascun nodo.

L'invio di pacchetti verso il gateway è implementata con una coda FIFO, quando viene inviato un pacchetto a cui corrisponderà una risposta la coda viene fermata in attesa della risposta. Tramite servlet l'utente può gestire l'intero sistema, ad esempio è possibile gestire le stanze, aggiungerne di nuove, cancellare quelle esistenti o modificarne i parametri A ed n di ogni stanza. Se viene richiesto di cancellare una stanza il servizio chiederà all'utente in quale altra stanza collocare gli eventuali nodi di riferimento presenti in quella stanza dopodiché

riconfigurerà automaticamente tali nodi con la nuova stanza. Anche se vengono modificati i parametri A ed n di una stanza il sistema provvede a riconfigurare automaticamente i *reference node* presenti in quella stanza.

BlindNode
 Name:
 IEEE address: 00124b0000060769
 Short address: 31073
 Auto mode: Polled - Automatic
 Localization algorithm: Centralized - Distributed
 Timeout (sec)
 Cycle time (sec)

Current Location
 Room: Ufficio
 Position X,Y: 10.00 m ; 1.25 m

Sensors
 Sensors on node: 3
 Temperature: 17.00 °C
 Battery Voltage: 2.56 V

Diagnostic
 Battery voltage: 2.56 V
 Parent RSSI: -48 dBm
 Received last packet at: Tue 31/Mar/2009 10:17:11.658

Delete node
 Ensure that the node is switched off before deleting.

Figura 5.8. Questa schermata della *web-application* mostra invece la pagina di configurazione di un nodo mobile, come si può vedere è possibile impostare le tempistiche e le modalità di funzionamento. Con i bottoni “*Localization History*” e “*Values History*” è possibile accedere allo storico delle posizioni rilevate e dei valori di ogni sensore presente sul nodo.

5.3.3 Meccanismo di raccolta dati elementari

I dati elementari che vengono usati per calcolo della posizione, sia in modo centralizzato sul servizio, che in modo distribuito con il motore di localizzazione hardware su ogni nodo mobile, sono: le misure di RSSI dal nodo mobile da localizzare rispetto ai nodi di riferimento con cui riesce a comunicare direttamente e i parametri dell'ambiente A ed n in cui ci si trova. Per quanto riguarda i parametri A ed n devono essere misurati dall'utente, grazie ad essi il sistema potrà essere calibrato per ogni tipo di ambiente, sia in ambienti esterni che interni. Invece le misure di RSSI vengono effettuate dai nodi, vediamo come avviene questo procedimento. Nel momento in cui un *blind node* deve misurare gli RSSI, perché o si è appena risvegliato in modalità automatica oppure perché ha appena ricevuto una richiesta di RSSI, inizia a inviare una serie di pacchetti detti *blast*. I *blast* sono pacchetti vuoti che vengono inviati in *broadcast* con l'impostazione `AF_SKIP_ROUTING` che fa in modo che facciano un solo *hop* evitando quindi che vengano propagati per l'intera rete dai router. Infatti vogliamo fare le misure di RSSI rispetto ai *reference node* con cui il nodo mobile può comunicare direttamente. Nel sistema realizzato vengono inviati 8 *blast*. I nodi di riferimento nelle vicinanze ricevono i *blast*, può capitare che non tutti nodi di riferimento ricevano tutti i *blast* inviati dal *blind node*. Per questo motivo è il nodo mobile a richiedere le misure di RSSI ai *reference node* subito dopo aver terminato di inviare i *blast*, anche tale richiesta viene inviata in *broadcast* con l'opzione `AF_SKIP_ROUTING` impostata. I nodi di riferimento memorizzano l'RSSI misurato per ogni singolo *blast* ricevuto, quando ricevono la richiesta di misure dal nodo mobile calcolano la media e σ^2 degli RSSI precedentemente memorizzati, nel fare questi calcoli si tiene conto anche dell'RSSI della richiesta di misure appena ricevuta essendo anch'essa proveniente con un solo *hop* dal nodo mobile. Dunque l'RSSI in *uplink* (dal nodo mobile verso i nodi di riferimento) non è altro che la media degli RSSI dei *blast* e della richiesta di misure:

$$RSSI_{up} = \frac{\sum_{i=1}^n RSSI_i}{n}$$

Per avere un indice sulla varianza di tali misure si calcola anche σ^2 :

$$\sigma^2 = \frac{\sum_{i=1}^n (RSSI_i - RSSI_{up})^2}{n - 1}$$

Appena calcolati questi due valori il *reference node* li invia al nodo mobile che li aveva richiesti. Il nodo mobile riceverà quindi le risposte da tutti i nodi di riferimento nelle vicinanze che hanno ricevuto almeno un *blast*. Per ogni risposta ricevuta il *blind node* memorizza $RSSI_{up}$ e σ^2 e anche $RSSI_{dw}$: l'RSSI in *downlink* che sarebbe l'RSSI rilevato sul ricevimento della risposta stessa. $RSSI_{dw}$ a differenza di $RSSI_{up}$ non è una media ma bensì una misura singola. Tutti questi dati vanno a formare il pacchetto RSSI destinato al servizio, visto precedentemente, oppure verranno dati in pasto al motore di localizzazione hardware. I valori di σ^2 non sono richiesti dal motore di localizzazione interno ma sono invece molto utili all'algoritmo di localizzazione presente sul servizio per cercare di perfezionare la stima della posizione.

5.3.4 Traduzione dei dati elementari in coordinate geografiche

L'algoritmo usato dal motore di localizzazione hardware presente sui nodi è proprietario: di conseguenza non è al momento possibile sapere come funziona. Di seguito viene illustrato l'algoritmo di localizzazione implementato sul servizio che viene richiamato ogni volta che si riceve un pacchetto RSSI proveniente dal gateway. Per tradurre le misure di RSSI in distanze l'algoritmo ha bisogno anche dei parametri ambientali A ed n . Nel database, nella tabella delle stanze, sono memorizzati i parametri A ed n di ogni stanza e anche la massima distanza misurabile in quella stanza. Per ogni pacchetto RSSI ricevuto il servizio va innanzitutto a leggere la prima misura di RSSI che è sempre la maggiore e assume che il nodo mobile si trovi nella stessa stanza del nodo di riferimento che ha originato l'RSSI più forte quindi potrà andare a recuperare i parametri A ed n di quella stanza.

Selezione delle misure

Prima di iniziare a fare i calcoli è necessario selezionare le misure appena ricevute: il servizio controlla subito i valori di σ^2 , tutte le misure che ce l'hanno troppo alto vengono scartate. Nel servizio si è scelto di porre questo limite per $\sigma^2 > 10$, 10 è un valore molto alto di σ^2 , si è scelto di lasciare un margine così ampio proprio perché nella pratica le misure sono quasi sempre soggette a forte varianza e come si vedrà in seguito l'algoritmo tiene conto anche del valore di σ^2 di ogni misura per stimare la posizione. Nella pratica si è provato a frapporre oggetti in movimento tra i nodi mentre venivano fatte le misure di RSSI, ne sono risultati valori di σ^2 nettamente più elevati rispetto ad una situazione statica. Le misure di RSSI che hanno passato questa prima selezione possono proseguire: per ogni misura viene calcolato l'RSSI medio tra RSSI in *uplink* e *downlink*:

$$RSSI = \frac{RSSI_{up} + RSSI_{dw}}{2}$$

Dopodiché si possono trasformare i valori di RSSI così calcolati in distanze con la seguente formula:

$$d = 10^{\frac{A - RSSI}{10 \cdot n}}$$

Che sarebbe la formula inversa di quella citata nel paragrafo sui sistemi basati sull'intensità nel capitolo sulla localizzazione:

$$RSSI = -(10n \log_{10} d - A)$$

Dall'elenco delle stanze, per la stanza in cui si assume di trovarsi, oltre ad A ed n (appena usati per calcolare le distanze) viene caricato anche il valore della massima distanza misurabile in quella stanza. La massima distanza misurabile in ogni stanza deve essere configurata dall'utente nel momento in cui vengono inserite le stanze nel sistema. Ora che si hanno le distanze il servizio procede con una seconda selezione: vengono scartate tutte le misure di distanza maggiori della massima distanza misurabile in quella stanza. Con le misure rimanenti si può procedere al calcolo della posizione. Se rimane una sola misura il servizio non potrà fare altro che indicare la sua posizione come appartenente alla stessa stanza dove si trova il nodo di riferimento che ha originato tale misura. Si è immaginato che venga installato almeno un nodo di riferimento per

ogni stanza. Se rimangono 2 misure la posizione viene calcolata facendo la media pesata, sulle distanze rilevate, delle coordinate dei due nodi di riferimento. Se invece passano la selezione almeno 3 misure il servizio procede con il calcolo di multilaterazione.

Calcolo di multilaterazione

Per la stima delle coordinate si è deciso di usare l'algoritmo di multilaterazione atomica esattamente come descritto nel paragrafo 3.3.1. Dato che è necessario fare calcoli matriciali il servizio fa uso della libreria *Jama*. Oltre alla stima delle coordinate: (x_0, y_0) viene calcolato anche il residuo sul calcolo di multilaterazione.

Selezione dei risultati

Il residuo permette di valutare di quanto le distanze misurate differiscono da quelle che si hanno tra i nodi di riferimento e il punto stimato. Il servizio esegue una selezione dei risultati ottenuti dal calcolo di multilaterazione basandosi sul valore del residuo. Si è deciso di accettare un residuo massimo di 2: tutti i risultati con un residuo maggiore di 2 vengono scartati.

Correzione della stima di posizione

I risultati che hanno passato la selezione del residuo vengono processati per un'ulteriore correzione per la quale si tiene conto anche dei valori di σ^2 . In pratica si vuole assegnare un peso maggiore alle misure che hanno un valore di σ più basso. Questa volta si avranno come incognite: δx e δy che sono le correzioni da sommare ai valori delle coordinate presunte precedentemente: x_0 e y_0 . Il procedimento di compensazione da usare è quello che fa ricorso al metodo statistico dei minimi quadrati attuato secondo l'algoritmo delle osservazioni indirette. Si costruisce quindi un sistema per calcolare gli scarti sulle coordinate tenendo conto delle differenze tra le distanze misurate e quelle ricalcolate tra il punto stimato (x_0, y_0) e i punti di riferimento. Si ha quindi la matrice A_1 di due colonne con le componenti in x e y , il vettore colonna l dei termini noti, che contiene le differenze tra i quadrati delle distanze ricalcolate e quelle misurate d :

$$A_1 = \begin{bmatrix} 2(x_0 - x_1) & 2(y_0 - y_1) \\ 2(x_0 - x_2) & 2(y_0 - y_2) \\ \vdots & \vdots \\ 2(x_0 - x_n) & 2(y_0 - y_n) \end{bmatrix} \quad l = \begin{bmatrix} (x_1 - x_0)^2 + (y_1 - y_0)^2 - d_1^2 \\ (x_2 - x_0)^2 + (y_2 - y_0)^2 - d_2^2 \\ \vdots \\ (x_n - x_0)^2 + (y_n - y_0)^2 - d_n^2 \end{bmatrix}$$

Il servizio costruisce anche la matrice diagonale dei pesi P , che assegna un peso, inversamente proporzionale alle varianze, ad ogni equazione. Per fare questo è necessario ricalcolare i valori della deviazione standard σ_l rapportati alle distanze ottenute dalle misure di RSSI:

$$\sigma_l = -2d \cdot \sigma$$

$$P = \begin{bmatrix} \frac{1}{\sigma_{l1}^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_{l2}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_{ln}^2} \end{bmatrix}$$

Infine per calcolare δ si applica la seguente formula:

$$\delta = -(A_1^T \cdot P \cdot A_1)^{-1} \cdot A_1^T \cdot P \cdot l$$

δ è un vettore colonna di due elementi che sono i valori da sommare alle coordinate per compensarle:

$$\delta = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \quad \begin{array}{l} x = x_0 + \delta x \\ y = y_0 + \delta y \end{array}$$

Quando si ha $\sigma^2 = 0$ il servizio impone $\sigma^2 = 0,01$ per evitare di dividere per zero. Volendo è possibile provare a iterare ripetendo la correzione sui nuovi x e y così ottenuti, ma nella pratica si è visto che le iterazioni successive non portano a significativi miglioramenti rispetto alla prima, si ottengono dei valori di δ molto piccoli, praticamente ininfluenti. Questo accade nella gran parte dei casi quando la funzione converge, però non è certo che iterando si converga verso un punto, se la funzione diverge iterare sarebbe solo dannoso in quanto porterebbe il punto stimato ad allontanarsi rapidamente dalla stima iniziale. Si è visto sperimentalmente che nella maggioranza dei casi la funzione converge e quindi applicando la correzione si ha un miglioramento. Si è quindi deciso di iterare la correzione una sola volta.

Capitolo 6

Analisi dei risultati sperimentali

6.1 Relazione tra RSSI e distanza in ambiente aperto

Come si è visto, il sistema, per tradurre le misure di RSSI in distanze fa uso della relazione: $RSSI = -(10n \log_{10} d - A)$. L'esperimento descritto in seguito ha lo scopo di verificare tale equazione e la stima dei parametri A ed n sul campo.

6.1.1 Descrizione dell'esperimento

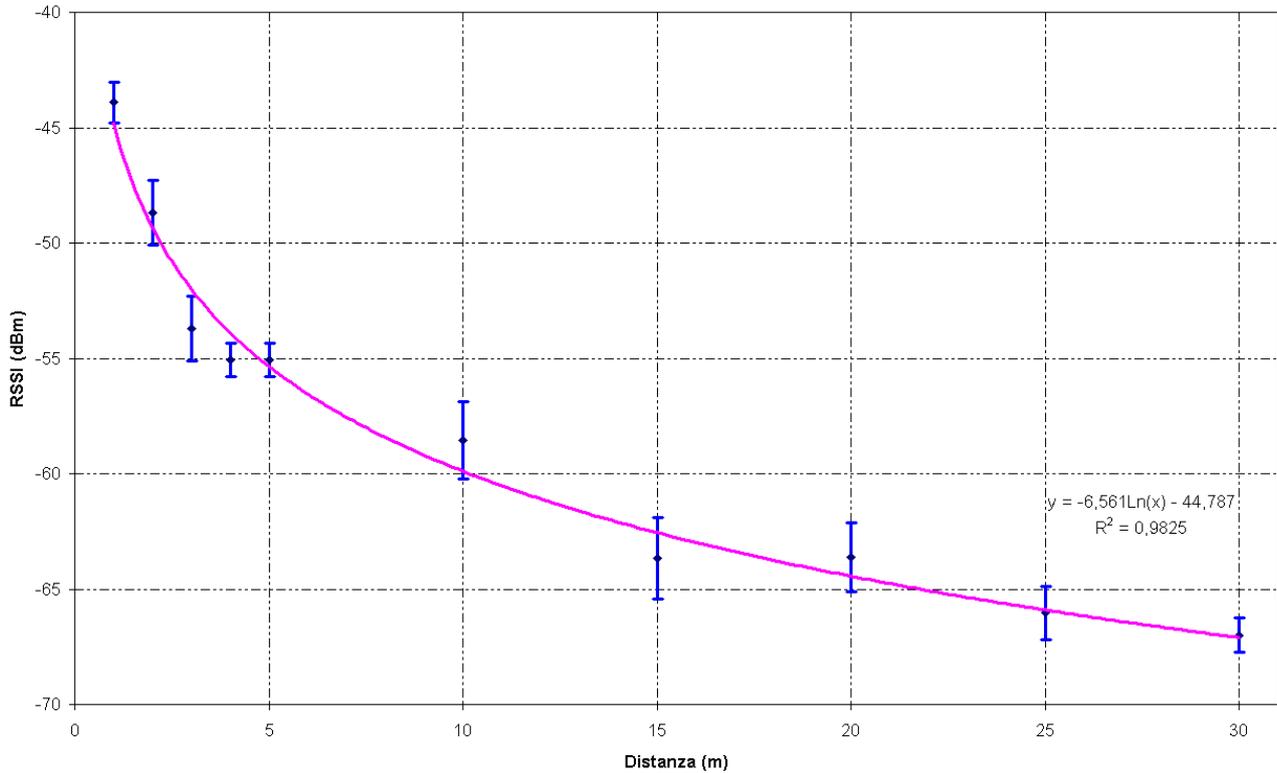
Le misure sono state effettuate in campo aperto (un prato) senza ostacoli nel raggio di 60 metri, i moduli sono stati posti ad un'altezza di 30 cm da terra. Durante la prova si sono usati due moduli CC2431 montati su *BatteryBoard*, quindi con l'antenna esterna, uno programmato come *reference node* lasciato sempre fermo, mentre l'altro programmato come *blind node* e spostato a varie distanze note dal punto di riferimento. Per ogni ogni distanza il nodo mobile, in modalità automatica, continua a ripetere le misure di RSSI rispetto al singolo nodo di riferimento e le invia al gateway. Il gateway è collegato ad un computer su cui è in esecuzione il servizio opportunamente modificato per collezionare le misure di RSSI. In questo modo si possono ripetere svariate misure ad ogni distanza, nell'esperimento si sono effettuate 35 misure di RSSI ad ogni distanza.

6.1.2 Misure di RSSI rilevate

d (m)	RSSI (dBm)	σ
0	-20,000	0,000
1	-43,904	0,298
2	-48,694	0,467
3	-53,694	0,467
4	-55,059	0,239
5	-55,059	0,239
10	-58,559	0,561
15	-63,649	0,588
20	-63,600	0,497
25	-66,029	0,382
30	-67,000	0,246

Tabella 6.1. Tabella riassuntiva delle misure, per ogni distanza della prima colonna, si sono effettuate 35 misure. Il valore nella seconda colonna è la media delle 35 misure effettuate, mentre il numero nella terza colonna è la deviazione standard.

Con i dati in tabella possiamo provare a tracciare un grafico e a fare un'interpolazione logaritmica tra i punti:



Come si può notare la linea di tendenza non intercetta le barre d'errore date da 3σ alle distanze di 3 e 4 metri, ma è comunque molto vicina agli estremi superiori, ciò è probabilmente dovuto alle riflessioni. Per il resto, la linea di tendenza rispetta le barre d'errore e tende ad avvicinarsi sempre di più ai punti misurati con l'aumentare della distanza, probabilmente questo si deve al fatto che su distanze maggiori, in campo aperto, l'influenza delle riflessioni tende a farsi sentire sempre meno. La funzione interpolata ha la seguente equazione:

$$y = -6,561 \cdot \ln x - 44,787$$

Che grazie alla proprietà dei logaritmi:

$$\ln x = \frac{\log_{10} x}{\log_{10} e}$$

Può essere ricondotta nella forma: $RSSI = -(10n \log_{10} d - A)$ con $A = -44,787$ dBm e $n = 1,511$. Si può quindi dire che l'equazione precedente descrive abbastanza bene l'andamento del valore dell'RSSI con l'aumentare della distanza, ma nonostante il valore di R^2 pari a 0,9825 l'esperimento dimostra anche che anche in campo aperto l'RSSI viene influenzato dalle riflessioni. Negli ambienti chiusi l'effetto delle riflessioni è chiaramente molto più accentuato, si può cercare di ovviare al problema delle riflessioni cercando i valori di A ed n per ogni ambiente. Ecco quindi che la scelta dei parametri ambientali diventa fondamentale per migliorare la precisione del sistema.

6.2 Verifica delle coordinate stimate dal sistema

Si è svolto un secondo esperimento per verificare l'accuratezza delle coordinate calcolate dal sistema e per controllare il funzionamento dell'algoritmo di compensazione implementato. Si sono installati tre nodi di riferimento facendo uso dei moduli CC2431 installati su *battery board*. Sempre in campo aperto e sempre a 40 cm da terra. Trovandosi nello stesso luogo dell'esperimento precedente si sono usati i parametri ambientali $A = -44,787$ dBm ed $n = 1,511$. I tre nodi di riferimento vengono posizionati in posizioni note che formano un triangolo equilatero di lato 20 m, configurati con le seguenti coordinate: $(0; 0)$, $(20; 0)$ e $(10; 17,32)$. Si è quindi provveduto a spostare in varie posizioni note un nodo mobile, si tratta anche in questo caso di un CC2431 con l'antenna esterna montato su *battery board*, posizionandolo sempre a 40 cm da terra. Per ogni posizione si prova a localizzare il nodo mobile con il sistema, collezionando le stime fatte in prima battuta con il calcolo di multilaterazione e la stime finali compensate.



Figura 6.1. Il luogo dove sono state effettuate le prove.

6.2.1 Coordinate rilevate

Pos. reale	Pos. stimata	Err. stima	Pos. compensata	Err. comp.
10,00; 8,00	11,60; 7,97	1,60	11,59; 8,98	1,86
10,00; 15,32	10,00; 15,79	0,47	10,00; 15,81	0,49
10,50; 16,45	7,71; 14,51	3,39	9,62; 15,18	1,54
12,50; 13,00	7,71; 14,41	4,99	14,23; 10,79	2,81
10,00; 7,32	10,00; 6,73	0,59	10,00; 7,42	0,10
5,00; 8,66	-0,44; 12,52	6,67	5,64; 8,71	0,64
4,00; 6,93	1,83; 9,91	3,69	3,22; 6,14	1,11
10,00; 19,32	10,00; 19,40	0,08	10,00; 19,32	0,00

Tabella 6.2. Nella prima colonna ci sono le coordinate reali, nella seconda le stime provenienti dal calcolo di multilaterazione e nella terza gli errori commessi dal calcolo di multilaterazione rispetto ai punti reali. Nella quarta colonna troviamo invece la stima finale compensata delle coordinate e nell'ultima vi è l'errore commesso dalla stima finale rispetto al punto reale.

Come si può notare nei primi due casi la correzione tende a peggiorare lievemente la stima iniziale ma nel complesso permette di migliorare, anche significativamente, i risultati. Durante le varie prove si è visto oscillare l'errore commesso dal sistema fino a massimi di 5 m. Sperimentalmente, dalle prove effettuate, si può stimare un errore medio di 2,5 m. Questo secondo esperimento mette in evidenza anche l'importanza di avere antenne il più possibile omnidirezionali. Purtroppo le antenne in dotazione non sono del tutto omnidirezionali infatti ruotando il nodo mobile su se stesso, mantenendolo nella medesima posizione si può osservare come le distanze rilevate rispetto ai nodi di riferimento cambiano anche sensibilmente sulle lunghe distanze. (Nel primo esperimento i nodi venivano allontanati sempre con lo stesso orientamento.)

6.3 Relazione tra RSSI e distanza in ambiente interno

Si è ripetuto il primo esperimento all'interno di una stanza di un edificio per verificare come cambia la relazione tra RSSI e distanza in ambienti interni. Per poi provare a stimare i parametri ambientali anche in quest'ultimo caso.

6.3.1 Descrizione dell'esperimento

Le misure sono state effettuate esattamente come nel primo esperimento solo a distanze più corte avendo a disposizione una distanza massima in linea retta inferiore rispetto al primo esperimento.

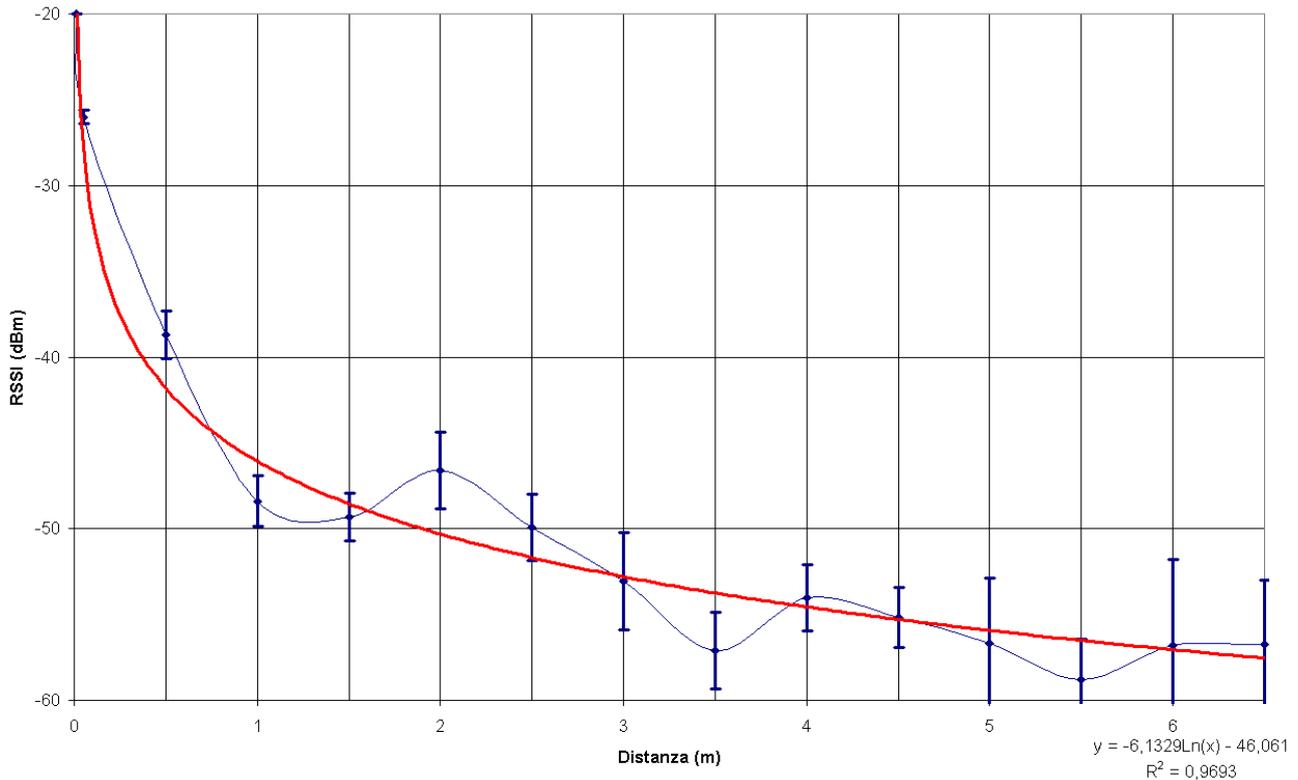
6.3.2 Misure di RSSI rilevate

d (m)	RSSI (dBm)	σ
0,01	-20,000	0,000
0,05	-26,017	0,131
0,50	-38,691	0,465
1,00	-48,397	0,493
1,50	-49,319	0,470
2,00	-46,583	0,746
2,50	-49,929	0,636
3,00	-53,083	0,946
3,50	-57,103	0,736
4,00	-54,043	0,647
4,50	-55,191	0,580
5,00	-56,706	1,270
5,50	-58,780	0,786
6,00	-56,786	1,667
6,50	-56,721	1,232

Tabella 6.3. Tabella riassuntiva delle misure, per ogni distanza della prima colonna, si sono effettuate 35 misure. Il valore nella seconda colonna è la media delle 35 misure effettuate, mentre il numero nella terza colonna è la deviazione standard.

Questa volta si nota chiaramente l'effetto delle riflessioni, da questi risultati si può affermare che il sistema non è adatto a fare una localizzazione per coordinate in stanze piccole, con una diagonale inferiore a 10 m, per il fatto che si avrebbero errori sulle stime delle distanze troppo elevati relativamente alle dimensioni delle stanze. In ambienti di questo tipo conviene configurare il sistema a localizzare i nodi mobili per stanza, limitandosi cioè ad indicare solamente in quale stanza si trovano. Si può avere un miglioramento nel calcolare le stime delle distanze ricavando i parametri ambientali dalla funzione interpolata, quindi come nel primo esperimento con

i dati in tabella possiamo provare a tracciare un grafico e a fare un'interpolazione logaritmica tra i punti:



La funzione interpolata ha la seguente equazione: $y = -6,1329 \cdot \ln x - 46,061$ riconducendola nella forma: $RSSI = -(10n \log_{10} d - A)$ si ottiene $A = -46,061$ dBm e $n = 1,412$. I parametri ambientali appena ricavati differiscono da quelli ottenuti in ambiente aperto, questo conferma che la scelta di tali parametri, a seconda dell'ambiente in cui ci si trova, è importante per cercare di stimare il meglio possibile le distanze partendo da misure di RSSI, volendo comunque evitare di tener conto di tutte le riflessioni che si avrebbero in ogni ambiente.

Capitolo 7

Conclusioni

7.1 Considerazioni sull’RSSI

L’indice RSSI viene calcolato dallo stack ZigBee ogni volta che viene ricevuto un pacchetto, tale indice viene calcolato basandosi sul numero di errori di trasmissione rilevati nel pacchetto appena ricevuto, in quanto ci si aspetta un numero di errori più elevato per potenze ricevute più basse. Di conseguenza l’RSSI non è la potenza reale ricevuta ma, come dice il suo acronimo stesso, un’indice della stessa. Inoltre come visto nelle prove realizzate l’RSSI, così come anche la potenza ricevuta reale, sono soggetti alle riflessioni. Nella realtà risulta estremamente complesso costruire un modello che tenga conto delle riflessioni che si verrebbero a creare in ogni ambiente; ed anche se fosse possibile basterebbe un cambiamento minimo per alterare completamente il modello realizzato. Ad esempio basterebbe aprire o chiudere una porta. Tener conto anche delle riflessioni non basta: nella realtà si ha a che fare con tutta una serie di altri fattori che possono influenzare l’indice RSSI come, ad esempio: oggetti in movimento o disturbi. Per quanto riguarda i disturbi si pensi a quelli dovuti al funzionamento di motori elettrici o alternatori o anche solo alla presenza di altre reti senza fili che operano su frequenze vicine come peraltro le reti *WiFi* e *Bluetooth*. Ci si deve quindi accontentare di stimare le distanze con la seguente relazione:

$$RSSI = -(10n \log_{10} d - A)$$

Adattandola ad ambienti diversi scegliendo accuratamente i valori dei parametri ambientali A ed n che è importante stimare con apposite misure come visto precedentemente. Sostanzialmente la prima conclusione a cui si può arrivare è che l’indice RSSI, pur avendo grandi vantaggi in termini di costi per il fatto che non richiede alcun hardware aggiuntivo oltre all’interfaccia di rete già presente sui sensori, non è un buon mezzo per localizzare i sensori in modo preciso, con un’errore inferiore ad un metro.

7.2 Considerazioni sui risultati ottenuti

Nonostante la debolezza intrinseca dell'indice RSSI si può comunque affermare che il prototipo realizzato raggiunge gli obiettivi prefissati. Per la localizzazione in ambienti aperti o comunque vasti con le coordinate il sistema ha un errore medio di 2,5 *m* con dei massimi di 5 *m*, questi dati ottenuti sperimentalmente sono stati confermati da un punto di vista più teorico provando a calcolare i semiassi dell'ellisse d'errore (vedere paragrafo 3.5.2) su di un campione di misure. Quindi il sistema può essere utilizzato per localizzare con le coordinate, in ambienti aperti o molto vasti, oggetti di grandi dimensioni come per esempio autovetture o container. Per quanto riguarda la localizzazione per stanze, collocando un nodo di riferimento per ogni stanza che si vuole coprire, si sono ottenuti buoni risultati, grazie alla presenza dei muri il sistema non ha quasi mai avuto dubbi ad individuare la stanza in cui si trovava il nodo mobile.

Inoltre il sistema implementa tramite *servlet* java un'interfaccia utente con cui è possibile gestirlo rapidamente da remoto e configurarlo in ogni suo aspetto; questa caratteristica è sicuramente un punto di forza del prototipo realizzato che quindi si può prestare come valida base di partenza per la realizzazione di sistemi di localizzazione per applicazioni più specifiche.

7.3 Considerazioni sull'installazione del sistema

La scelta dei punti in cui installare i nodi di riferimento è fondamentale per ottenere un'accuratezza maggiore nella localizzazione. Occorre innanzitutto stabilire in quali luoghi si vuole una localizzazione con le coordinate piuttosto che una localizzazione per stanze. Nei luoghi aperti o in ambienti chiusi di grandi dimensioni (di un ordine di grandezza maggiore di quello dell'errore del sistema), come per esempio fabbriche o capannoni, si può avere una localizzazione per coordinate a patto di installare almeno tre nodi di riferimento e di configurarli con i parametri *A* ed *n* specifici per l'ambiente in cui ci si trova. Inoltre i nodi di riferimento devono essere posizionati in modo da coprire interamente l'area che si desidera monitorare e dato che svolgono anche le mansioni di router devono essere alimentati in modo permanente e devono sempre poter comunicare con almeno uno degli altri router. Per migliorare le stime delle posizioni i nodi di riferimento dovrebbero essere installati tutti alla stessa altezza da terra il più possibile simile a quella a cui si troveranno i nodi mobili da tracciare. Installare molti nodi di riferimento è un buon modo, seppur costoso, per migliorare significativamente l'accuratezza delle stime ed incrementare l'affidabilità della rete.

Conviene invece impostare il sistema per una localizzazione per stanze dove si hanno più stanze di dimensioni comparabili con l'errore commesso dal sistema (dove quindi non avrebbe senso la stima delle coordinate su di un sistema di riferimento relativo alla stanza stessa). In molti casi è sufficiente un solo nodo di riferimento per stanza, ed anche qui è importante posizionare strategicamente i nodi di riferimento, in particolare occorre evitare di installarli in prossimità di porte o altre aperture che danno verso altre stanze coperte dal sistema, poiché in questo modo, i nodi che si trovano nell'altra stanza potrebbero rilevare degli RSSI più alti per un nodo di riferimento che non è quello della loro stanza.

Un'altro aspetto importante da non sottovalutare per la messa in funzione del sistema è la scelta delle antenne dei moduli, soprattutto laddove si desidera una localizzazione per coordinate le antenne devono essere di buona qualità ma soprattutto il più possibile omnidirezionali.

7.4 Miglioramenti futuri al sistema

Un modo per cercare di migliorare le stime delle distanze a partire dall'RSSI è quello di fare le misure di RSSI trasmettendo a livelli di potenza diversi. Nel sistema realizzato si è sempre trasmesso alla potenza massima consentita pari a 1 mW; però lo stack ZigBee permette anche di trasmettere a livelli di potenza inferiori. Si potrebbero quindi inviare i *blast* usati per fare le misure di RSSI a potenze via via crescenti fino ad arrivare a quella massima consentita. L'RSSI potrebbe quindi essere calcolato con una media pesata che tenga conto delle varie potenze a cui sono stati trasmessi i *blast*, ciò permetterebbe di avere le stime delle distanze più accurate. Un altro miglioramento, di grande utilità, da apportare al sistema consiste nell'implementare una procedura di calibrazione automatica, da richiamare ogni qualvolta l'utente voglia inserire un nuovo ambiente o stanza nel sistema. Si è vista l'importanza dei valori corretti dei parametri ambientali A ed n per una migliore stima delle distanze, conviene quindi facilitare la misura di tali parametri. La procedura di calibrazione automatica dovrebbe richiedere all'utente di porre a varie distanze note un dato nodo mobile rispetto ad un dato nodo di riferimento, posizionato nell'ambiente per cui si vogliono calcolare i parametri ambientali. Tale procedura dovrebbe dunque raccogliere tutta una serie di misure di RSSI a distanze note, per poi provare ad interpolare una funzione di tipo logaritmico, da cui ricavare A ed n .

Bibliografia

- [1] *M. A. Spirito, F. Sottile*
“**Standard 802.15.4 e ZigBee**”
Istituto Superiore Mario Boella
- [2] *Shashank Tadakamadla*
“**Indoor Local Positioning System For ZigBee, Based on RSSI**”
Mid Sweden University
- [3] *Guido Albertengo*
Lucidi del corso “**Reti radiomobili**”
Politecnico di Torino
- [4] *Tor-Inge Kvakrsrud*
Design Note DN018 “**Range Measurement in an Open Field Environment**”
Texas Instruments Inc.
- [5] *Andreas Savvides, Chin-Chien Han, Mani B. Strivastava*
“**Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors**”
University of California, Los Angeles
- [6] *Koen Langendoen, Niels Reijers*
“**Distributed Localization Algorithms**”
Delft University of Technology, Olanda
- [7] *Nancy Lynch*
“**Advanced Distributed Algorithms: Localization**”
Massachusetts Institute of Technology
- [8] *Andreas Savvides, Heemin Park, Mani B. Strivastava*
“**The n -Hop Multilateration Primitive for Node Localization Problems**”
University of California, Los Angeles

- [9] *Prof. Giorgio Vassena*
“**Il metodo delle osservazioni indirette**”
Corso di Topografia A - Università di Brescia
- [10] *S. Viscomi*
“**PREGEO L’elaborazione del libretto delle misure**”
<http://www.topografi.it>
- [11] “**Il principio dei minimi quadrati**”
Università IUAV di Venezia
- [12] “**Z-Stack Developer’s Guide**”
Texas Instruments Inc.