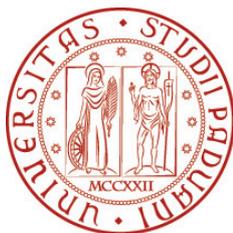


ENRICO ALBERTINI

LOCALIZZAZIONE TRAMITE TECNICHE DI
FINGERPRINTING RADIO

TESI DI LAUREA

Relatore: Ch.mo Prof. Andrea Zanella



Università degli Studi di Padova

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria delle Telecomunicazioni

27 marzo 2012

INDICE

INTRODUZIONE	vii
1 RETI DI SENSORI WIRELESS	1
1.1 Panoramica	1
1.2 Il mote	2
1.2.1 Architettura di un mote	2
1.2.2 Sistema operativo di un mote	4
1.2.3 Un esempio commerciale: TelosB	4
1.3 Applicazioni delle WSN	5
2 PRINCIPI DI MISURA PER LA LOCALIZZAZIONE	9
2.1 Triangolazione	9
2.1.1 Time of Arrival (TOA)	9
2.1.2 Received Signal Strength (RSS)	10
2.1.3 Angle of Arrival (AOA)	11
2.1.4 Altri metodi di localizzazione	12
2.2 Algoritmi basati sul fingerprinting	13
2.2.1 Metodo probabilistico	13
2.2.2 k -nearest-neighbor (kNN)	14
2.2.3 Reti neurali	14
2.2.4 Support Vector Machine (SVM)	16
2.2.5 Smallest M-vertex Polygon (SMP)	16
2.3 Prossimità	16
3 METODO FINGERPRINTING	19
3.1 Procedura operativa	19
3.1.1 Fase di misura	19
3.1.2 Fase di localizzazione	20
3.2 Esempio di implementazione	20
4 CONCLUSIONI	23
BIBLIOGRAFIA	25

ELENCO DELLE FIGURE

Figura 1.1	Architettura software ZigBee.	1
Figura 1.2	Esempio di rete ZigBee.	2
Figura 1.3	Architettura di base di un nodo.	3
Figura 1.4	Lato superiore del sensore TelosB.	5
Figura 1.5	Lato inferiore del sensore TelosB.	5
Figura 2.1	Localizzazione basata su TOA/RTOF	9
Figura 2.2	Localizzazione basata su RSS	10
Figura 2.3	Localizzazione basata su AOA	11
Figura 2.4	Localizzazione basata su TDOA	12
Figura 2.5	Esempio di rete neurale	15
Figura 3.1	Planimetria del laboratorio SigNET.	21

ELENCO DELLE TABELLE

Tabella 2.1	Errore medio nella stima con algoritmo kNN	14
Tabella 3.1	Esempio di entry corrispondente a un <i>fingerprint</i>	20

ACRONIMI

AOA	Angle of Arrival
CPU	Central Processing Unit
DSSS	Direct Sequence Spread-spectrum
GPS	Global Positioning System
HVAC	Heating, Ventilation and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks

IR	Infrared Radiation
ISO	International Organization for Standardization
kNN	k -nearest-neighbor
LOS	Line of Sight
MAC	Medium Access Control
MLP	Multilayer Perceptron
nesC	Network Embedded System C
OSI	Open Systems Interconnection
PHY	Physical
POA	Phase of Arrival
RAM	Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
ROM	Read Only Memory
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTOF	Roundtrip Time of Flight
SigNET	Special Interest Group on NETworking @ UNIPD
SMA	SubMiniature version A connector
SMP	Smallest M-vertex Polygon
SVM	Support Vector Machine
TDOA	Time Difference of Arrival
TOA	Time of Arrival
USB	Universal Serial Bus
UWB	Ultra Wide Band
VHF	Very High Frequency
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

INTRODUZIONE

Le reti di sensori wireless – Wireless Sensor Network (WSN) – sono recentemente divenute un argomento di ricerca molto sentito. L'idea base su cui si basano le WSN è che molti servizi utili possono essere forniti da una rete wireless formata da elementi economici e a basso consumo energetico chiamati *mote*. Questi piccoli dispositivi elettronici hanno una memoria, un'unità di elaborazione, dei sensori e un'interfaccia radio.

In questa tesi, verranno analizzati gli aspetti generali delle WSN, soffermandosi particolarmente sugli aspetti legati alla localizzazione, analizzando i principali algoritmi e metodi. Verrà presentato un esempio di implementazione di un sistema per la localizzazione indoor tramite il network fingerprinting.

RETI DI SENSORI WIRELESS

WSN sta per “Wireless Sensor Network”, è l’acronimo comune per indicare una rete wireless di dispositivi elettronici indipendenti chiamati mote. Questa rete può essere utilizzata per il controllo ambientale, acquisizione di dati, ecc. Ogni mote è connesso agli altri con un sistema di comunicazione wireless.

1.1 PANORAMICA

Lo standard IEEE 802.15.4 è stato concepito per regolamentare i livelli Physical (PHY) e Medium Access Control (MAC) di reti Wireless Personal Area Network (WPAN). Le specifiche ZigBee [6] sono basate su questo standard: esse sviluppano i livelli superiori del modello ISO/OSI, non coperti dallo standard, per offrire una soluzione completa di rete di trasmissione dati. Inoltre, lo standard 802.15.4 può essere utilizzato con gli usuali protocolli Internet e la specifica IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) per costruire reti Internet senza fili di tipo embedded. È importante sottolineare che lo standard IEEE 802.15.4 non è ZigBee. Come mostrato in figura 1.1, ZigBee implementa i livelli superiori dello *stack* ISO/OSI.

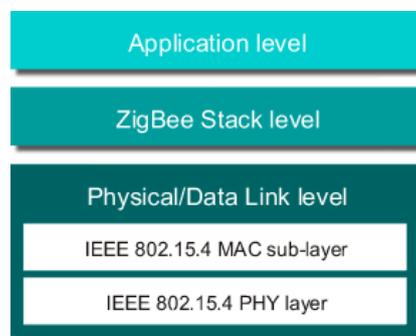


Figura 1.1: Architettura software ZigBee.

Come mostrato in figura 1.2, ci sono tre differenti tipi di dispositivo ZigBee:

- *Coordinator*: è il dispositivo più “intelligente”, costituisce la radice di una rete ZigBee e può operare da ponte tra più reti. Ci può essere un solo Coordinator in ogni rete. Questo dispositivo è inoltre in grado di memorizzare informazioni riguardo la sua rete e può agire come archivio per le chiavi di sicurezza.

- *Router*: si occupa dell'instradamento dei dati da e verso altri dispositivi.
- *End Device (mote)*: include solo le funzionalità minime per dialogare con il suo nodo parente (Coordinator o Router), non può trasmettere dati provenienti da altri dispositivi. Richiede il minor quantitativo di memoria e quindi risulta spesso più economico rispetto ai Router o ai Coordinator.

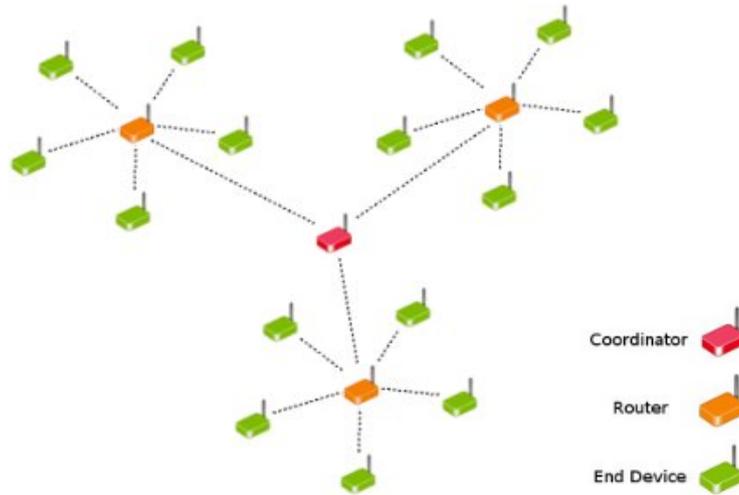


Figura 1.2: Esempio di rete ZigBee.

1.2 IL MOTE

Esistono in commercio molti tipi di *mote*, ma possiamo delineare alcuni tratti comuni:

- Basso bitrate della connessione wireless
- Bassa potenza computazionale
- Basso consumo di energia: un *mote* deve funzionare per molto tempo senza richiedere la sostituzione della batteria.

1.2.1 Architettura di un *mote*

Un nodo sensore, chiamato anche *mote*, è un nodo in una rete di sensori wireless che è in grado di eseguire alcune operazioni di elaborazione, raccogliere informazioni dai sensori e comunicare con gli altri nodi collegati alla rete.

I *mote* possono avere architetture molto differenti ma tutte con alcune caratteristiche comuni: l'unità di controllo, la memoria, l'unità di comunicazione, uno o più sensori e uno o più attuatori. L'architettura di base di un *mote* è illustrata in Figura 1.3. Di seguito analizzeremo in dettaglio le varie parti.

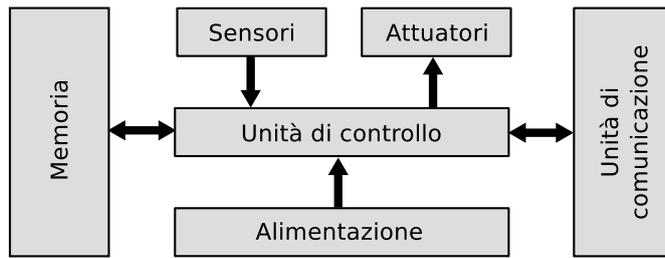


Figura 1.3: Architettura di base di un nodo.

- *Unità di controllo*: è l'unità di elaborazione centrale (Central Processing Unit (CPU)) del nodo. Il suo compito è quello di eseguire le istruzioni di un programma presente in memoria. Durante l'esecuzione del programma la CPU legge o scrive dati in memoria. Al momento i *mote* utilizzano vari tipi di microprocessori ad alta efficienza energetica, generalmente microcontrollori.
- *Memoria*: la memoria è divisa in Read Only Memory (ROM) e Random Access Memory (RAM). La ROM contiene le istruzioni che il nodo deve eseguire per funzionare (il firmware). Il termine ROM è spesso utilizzato in letteratura ma può essere frainteso: il contenuto della ROM può essere modificato (per la riprogrammazione del nodo), quindi è possibile scrivere sulla ROM ma non è generalmente fatto durante la normale attività del nodo. La RAM contiene l'informazione che l'unità di controllo sta elaborando in un determinato istante di funzionamento.
- *Unità di comunicazione*: si occupa di trasmettere e ricevere l'informazione dagli altri nodi. Generalmente il bitrate è basso se confrontato con quello di una rete di computer ma è necessario che sia così perché la trasmissione e la ricezione sono operazioni che richiedono molta energia.
- *Alimentazione*: il nodo consuma energia per elaborare e trasmettere le informazioni acquisite dai sensori. Molta energia è richiesta dall'unità di comunicazione. La batteria è la sorgente principale di potenza elettrica per il nodo.
- *Sensori*: permettono al nodo di raccogliere le informazioni dal "mondo esterno" come temperatura, pressione, umidità, ecc.
- *Attuatori*: permettono al nodo di modificare il "mondo esterno". Per esempio, potremmo avere un braccio meccanico che in caso la temperatura della stanza superi una certa soglia, apra una finestra.

1.2.2 Sistema operativo di un mote

Nella sezione precedente abbiamo avuto modo di esaminare le stringenti specifiche *hardware* che contraddistinguono un *mote*. In un ambiente così vincolato il sistema operativo deve soddisfare i seguenti requisiti:

- *Lavorare con risorse limitate*: la sua esecuzione non deve sovraccaricare il *mote*.
- *Gestire il multitasking*: i *mote* sono stati sviluppati per controllare il “mondo esterno” che genera eventi asincroni, per cui è necessario che il *mote* possa gestire più di un *task* alla volta.
- *Riusabilità del codice*: una volta terminato lo sviluppo di un programma per un tipo di *mote*, devono essere necessarie poche modifiche al codice per farlo funzionare su un altro tipo di *mote*.
- *Controllo generico dell'hardware*: visto l'enorme insieme di dispositivi che un *mote* può utilizzare, è necessario fornire strumenti semplici e generici per controllare l'*hardware*. Naturalmente un programma sviluppato per uno specifico *hardware* sarà meno portabile, ma più efficiente.
- *Basso consumo energetico*: durante la fase di sviluppo della WSN è molto noioso così è irragionevole spesso rifare solo per ricaricare i granelli, i programmi utilizzati hanno il potere di consumare meno possibile in modo da non dover frequentemente ridistribuire la rete.

TinyOS [1] è un sistema operativo Open Source, creato dall'Università di Berkeley (California, USA), che cerca di soddisfare i requisiti descritti precedentemente. TinyOS è scritto con il linguaggio ad eventi Network Embedded System C (nesC) [7], un “dialetto” del C sviluppato specificatamente per TinyOS.

1.2.3 Un esempio commerciale: TelosB

Dopo aver esaminato l'architettura generale di un *mote*, analizzeremo un esempio commerciale, il TelosB [4].

Le caratteristiche principali dei sensori TelosB sono:

- Conformità con lo standard IEEE 802.15.4/ZigBee
- Trasmettitore wireless Chipcon a 2.4 GHz
- Data rate di 250 kbps

- Microcontrollore Texas Instruments MSP430 da 8 MHz con RAM da 10 kB e Flash da 48 kB
- ROM da 45 kB
- Antenna *patch* integrata
- Interfaccia USB
- Compatibile con TinyOS
- Sensori integrati di temperatura, luminosità e umidità

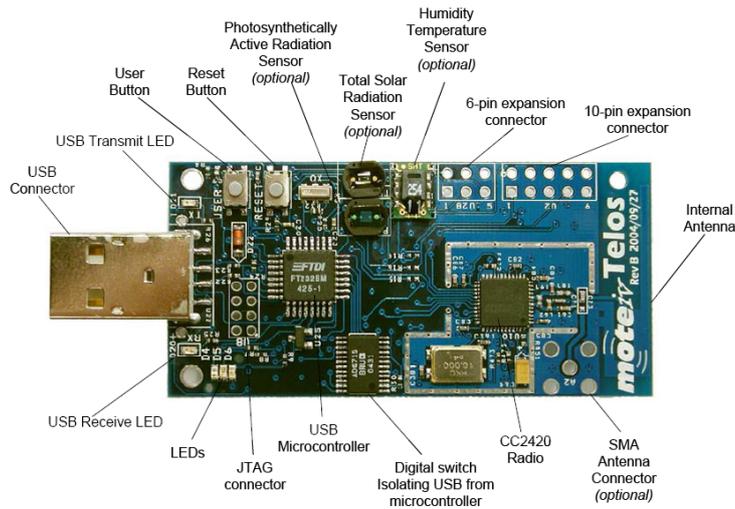


Figura 1.4: Lato superiore del sensore TelosB.

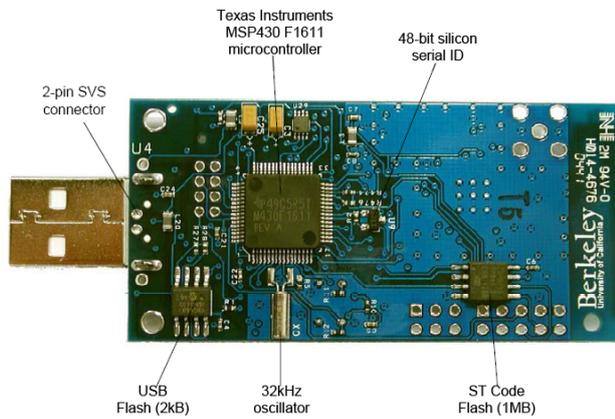


Figura 1.5: Lato inferiore del sensore TelosB.

1.3 APPLICAZIONI DELLE WSN

Una rete di sensori può essere costituita da un insieme di sensori di diverso tipo capaci di rilevare grandezze come tempera-

tura, umidità, pressione, luce, ma anche capaci di rilevare il movimento di veicoli, la composizione del terreno, livello di rumore e molte altre grandezze.

Vista la grande varietà di possibilità che le WSN offrono, possono trovare applicazione per il monitoraggio e controllo in campo militare, ambientale, sanitario, casalingo e commerciale [2].

Analizzeremo in dettaglio due esempi di utilizzo delle WSN legato alla localizzazione.

Esempio di applicazione: monitoraggio di animali

Ai fini della ricerca biologica, è molto utile monitorare gli animali per lunghi periodi e in ampie zone geografiche. Questi studi permettono di comprendere il comportamento animale e di analizzare le interazioni all'interno del loro stessa specie, così come con altre specie. Con le tecniche oggi in uso, il *tracking* è costoso e richiede *tag* di grandi dimensioni che esauriscono rapidamente la carica delle batterie.

La pratica più diffusa è quella di dotare gli animali da monitorare di collari con trasmettitore Very High Frequency (VHF), che vengono rintracciati con metodi di triangolazione utilizzando antenne direttive. In alternativa, è possibile utilizzare dei collari che sfruttano il sistema Global Positioning System (GPS), ma questa è una soluzione molto costosa e con il problema dell'alto consumo di energia.

L'utilizzo di reti di sensori wireless può migliorare notevolmente la ricerca nel settore biologico, come dimostrato dal progetto "ZebraNet" [8]. L'utilizzo del *routing multihop* permette comunicazioni a bassa potenza per la trasmissione dei dati relativi alla posizione degli animali. Inoltre, l'interazione animale può essere stimata utilizzando misurazioni *pair-wise* senza ricorrere al sistema GPS. Il risultato finale è una maggiore durata delle batterie e di conseguenza un numero minore di operazioni di manutenzione.

Esempio di applicazione: logistica

Consideriamo ora uno scenario diverso, formato da una rete di sensori in un'edificio composto da uffici, reparti di produzione e magazzino. I sensori ricoprono già un ruolo molto importante nel settore manifatturiero.

Il monitoraggio e il controllo dei macchinari industriali è tradizionalmente cablato, ma l'utilizzo dei sensori wireless riduce l'elevato costo del cablaggio e rende il reparto di produzione più dinamico.

Inoltre, i pacchi e gli oggetti da immagazzinare e tutte le apparecchiature per l'ufficio sono stati contrassegnati con sensori

quando sono arrivati nella struttura.

Questi sensori permettono il controllo delle condizioni di conservazione (temperatura e umidità) e il controllo del riscaldamento, ventilazione, e aria condizionata (Heating, Ventilation and Air Conditioning (HVAC)).

I sensori sulle attrezzature mobili segnalano la loro posizione quando il dispositivo viene perso o è necessario ritrovarlo (ad esempio, durante l'inventario), e persino contattare la sicurezza se l'apparecchiatura è in procinto di "uscire dell'edificio.

Trovare velocemente oggetti o attrezzature riduce la necessità di avere duplicati e quindi comporta un risparmio che giustifica il costo della rete di sensori wireless.

La tecnologia Radio Frequency Identification (RFID) rappresenta un primo passo nella logistica di magazzino. Le etichette RFID, rintracciabili solo quando passano a pochi metri da un lettore, rimangono irraggiungibili per la maggior parte del loro tempo di giacenza in magazzino.

Le reti di sensori wireless, tuttavia, possono essere raggiungibili e localizzate finché sono all'interno di un'area dell'ordine di una decina di metri dagli altri sensori wireless. La precisione della localizzazione cooperativa aumenta con la densità dei sensori.

PRINCIPI DI MISURA PER LA LOCALIZZAZIONE

Oggi giorno la letteratura propone vari metodi per risolvere il problema della localizzazione radio indoor, anche se tutti sfruttano lo stesso principio: un nodo *mobile* misura il segnale ricevuto da altri nodi per stimare la sua posizione corrente. Per una trattazione dettagliata delle varie tecniche si veda [9].

Non è semplice modellare la propagazione radio in un contesto indoor a causa di molteplici fattori tra cui: la presenza di un forte multipath, bassa probabilità della cosiddetta Line of Sight (LOS) e la presenza di svariati parametri ambientali (planimetria, oggetti in movimento, superfici di riflessione, ecc.).

2.1 TRIANGOLAZIONE

La triangolazione è una tecnica che permette di calcolare distanze fra punti sfruttando le proprietà dei triangoli. Può essere sfruttata in due modi. Il primo stima la posizione di un oggetto misurando le distanze da più punti di riferimento (TOA e RSS), il secondo stima la posizione di un oggetto misurando gli angoli relativi a più punti di riferimento (AOA).

2.1.1 Time of Arrival (TOA)

Questo metodo si basa sul fatto che la distanza tra due nodi è direttamente proporzionale al tempo di propagazione del segnale RF. Il ritardo di propagazione tra il nodo i e il nodo j , $T_{i,j}$, è uguale alla distanza tra il trasmettitore e il ricevitore, $d_{i,j}$, divisa per la velocità di propagazione, v_p .

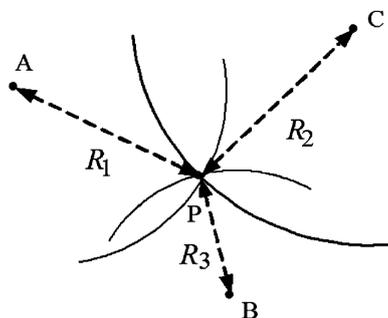


Figura 2.1: Localizzazione basata sulle misure TOA/RTOF.

Se un nodo i è in grado di determinare $T_{i,j}$, può calcolare la distanza $d_{i,j}$ che lo separa dal nodo j . Per poter determinare la posizione 2-D, il nodo i deve essere in grado di collezionare almeno tre misure da tre nodi differenti, come mostrato in figura 2.1.

Il punto chiave dei metodi basati sulla misura del tempo è la capacità del ricevitore di stimare con precisione il tempo di arrivo dei segnali LOS. Per fare ciò, è necessario che tutti i nodi della rete siano perfettamente sincronizzati. Inoltre il trasmettitore deve inviare una sorta di *timestamp* per permettere al ricevitore di conoscere con esattezza l'istante in cui è stato inviato il messaggio.

I principali errori che influiscono questa stima sono il rumore additivo e il *multipath*. Il TOA può essere misurata utilizzando differenti tecniche di misura tra cui il Direct Sequence Spread-spectrum (DSSS) [12] e il Ultra Wide Band (UWB) [5].

2.1.2 Received Signal Strength (RSS)

RSS è definito come la tensione misurata dal circuito Received Signal Strength Indicator (RSSI) del ricevitore. Spesso, l'RSS viene considerato in maniera equivalente alla potenza misurata, cioè, l'ampiezza al quadrato del segnale. Si può considerare l'RSS di un segnale acustico, RF, ecc.

I sensori wireless comunicano con gli altri sensori vicini, così l'RSS del segnale RF può essere misurato da ciascun ricevitore durante il normale trasferimento di dati senza richiedere un aumento di larghezza di banda o di energia.

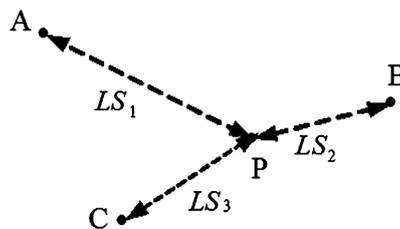


Figura 2.2: Localizzazione basata sulle misure RSS, dove LS_1, LS_2 e LS_3 rappresentano la misura di path loss.

Le misure RSS sono relativamente poco costose e di semplice implementazione a livello *hardware*. Tuttavia sono notoriamente imprevedibili e per sistemi *indoor* è difficile trovare un canale LOS tra il trasmettitore e il ricevitore.

In un canale reale, l'ambiente influisce sulla misura RSS. I principali fenomeni in questione sono il *multipath* e il *shadowing*.

Al ricevitore, giungono segnali con ampiezze e fasi differenti che vanno a sommarsi o sottrarsi in funzione della frequenza, causando un *fading* selettivo. L'effetto di questo tipo di *fa-*

ding può essere ridotto utilizzando il metodo *spread-spectrum* che media la potenza ricevuta in un ampio intervallo di frequenze.

L'altra fonte di errore nelle misure RSS, il *shadowing*, è legata all'attenuazione di un segnale a causa degli ostacoli (mobili, muri, alberi, edifici, ecc.) che incontra nel cammino tra il trasmettitore e il ricevitore.

Nello spazio libero, la potenza del segnale decade proporzionalmente a d^{-2} , dove d è la distanza tra il trasmettitore e il ricevitore. A causa dell'interferire degli oggetti nell'ambiente, la potenza del segnale ricevuto oscilla. La potenza media ricevuta attraverso un canale reale decade proporzionalmente a d^{-n} , dove n è l'esponente *path-loss*, tipicamente compreso tra 2 e 4. La potenza media a distanza d è tipicamente modellata come

$$P_L(d) = P_0 \left(\frac{d}{d_0} \right)^{-n} \quad (2.1)$$

dove P_0 è la potenza ricevuta (dBm) a una piccola distanza di riferimento d_0 .

2.1.3 Angle of Arrival (AOA)

La misura AOA è definita come l'angolo tra la direzione di propagazione di un'onda incidente e una direzione di riferimento che può coincidere con il Nord oppure con il segnale proveniente da un *beacon*. Per poter utilizzare il metodo AOA è necessario utilizzare un'antenna direttiva o un array di antenne su ogni nodo sensore. Altre tecniche per rilevare gli angoli tra i nodi sono discusso in [11] e [5].

Come mostrato in figura 2.3, il metodo AOA si basa sul fatto che la posizione di un nodo *mobile* P può essere determinata conoscendo la posizione di due punti di riferimento A e B e i relativi angoli θ_1 e θ_2 .

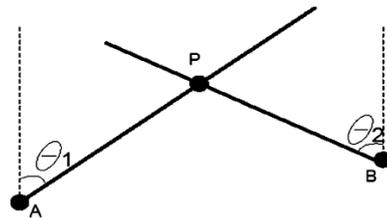


Figura 2.3: Localizzazione basata sulle misure AOA.

I principali vantaggi del metodo AOA sono il fatto che non sia necessaria nessuna sincronizzazione tra i nodi e che per determinare il posizionamento 3-D è necessario conoscere la posizione di soli tre nodi di riferimento (rispettivamente per il posizionamento 2-D è necessario conoscere la posizione di due nodi

di riferimento). D'altro canto il metodo AOA richiede *hardware* mediamente più complesso.

Naturalmente, per ottenere stime molto precise della posizione, è necessario effettuare misure accurate degli angoli. Come gli altri metodi, anche l'AOA soffre di *shadowing* e *multipath*.

2.1.4 Altri metodi di localizzazione

2.1.4.1 Time Difference of Arrival (TDOA)

Il principio base del TDOA è di misurare la differenza del tempo di arrivo di segnali provenienti dai nodi *anchor*. Considerando una coppia di nodi *anchor*, i e j , in cui (x_i, y_i, z_i) indicano le coordinate del primo e (x_j, y_j, z_j) quelle del secondo. La posizione (x, y, z) del nodo *mobile* deve rispettare l'equazione dell'iperboloide, data da:

$$R_{i,j} = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (2.2)$$

in cui $R_{i,j}$ è la differenza tra i raggi dei nodi *anchor*. Questo valore può essere stimato ed è la costante del nostro problema. In altre parole, il punto (x, y, z) deve trovarsi in un iperboloide con una differenza di raggio costante $R_{i,j}$. Si può stimare la posizione del nodo *mobile* usando le intersezioni di questi iperboloidi.

Come mostrato in figura 2.4, la localizzazione 2-D può essere effettuata tramite l'intersezione di due o più misure TDOA. Le misure TDOA effettuate sui nodi *anchor* (A, B e C) formano due iperboloidi la cui intersezione individua il nodo *mobile* P.

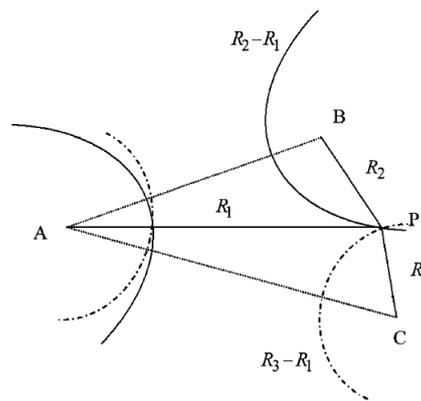


Figura 2.4: Localizzazione basata sulle misure TDOA.

2.1.4.2 Roundtrip Time of Flight (RTOF)

Utilizzando questo metodo è necessario misurare il tempo di volo di andata e ritorno dal nodo *mobile* al nodo *anchor*. Questo metodo richiede requisiti di sincronizzazione meno stringenti rispetto al metodo TOA ma introduce un nuovo problema. Il tempo di elaborazione del nodo *anchor* può essere difficile da stimare, in particolare se il servizio di localizzazione è solo uno dei molti servizi in esecuzione sulla WSN, l'elaborazione della risposta eseguita su un nodo *anchor* può essere interrotta da altri task.

2.1.4.3 Phase of Arrival (POA)

Questo metodo è anche chiamato "Received Signal Phase Method". L'idea è quella di misurare la fase del segnale ricevuto per stimare la posizione. Supponiamo che i nodi *anchor* trasmettano segnali sinusoidali puri con offset di fase nullo e stessa frequenza f . Al ricevitore si avrà

$$s_r(t) = e^{j2\pi f(t+t_0)} \quad (2.3)$$

ma t_0 è correlato alla distanza tra i nodi, più precisamente $t_0 = D_i/c$ dove c è la velocità della luce. Dalla (2.3) otteniamo

$$s_r(t) = e^{j(2\pi f t + \phi_0)} \quad (2.4)$$

dove $\phi_0 = 2\pi f t_0$. Utilizzando le due formule si possono ottenere le informazioni di distanza dalla fase, in modo da poter applicare alla fase entrambi i metodi TOA o TDOA.

2.2 ALGORITMI BASATI SUL FINGERPRINTING

La principale sfida introdotta dalle tecniche basate sul metodo *fingerprinting* è che l'intensità del segnale ricevuto può essere affetta, in un contesto di propagazione indoor, da diffrazione, riflessione e scattering.

Ci sono almeno cinque algoritmi di localizzazione basati sul *fingerprinting* che utilizzano tecniche di pattern recognition: metodo probabilistico, k -nearest-neighbor (kNN), reti neurali, Support Vector Machine (SVM) e Smallest M-vertex Polygon (SMP).

2.2.1 Metodo probabilistico

Questo metodo considera la localizzazione come un problema di classificazione. Supponiamo che il nodo *mobile* si possa trovare in n possibili posizioni L_1, \dots, L_n e che s sia il vettore contenente le misure di RSSI ottenute durante la fase di localizzazione. Indichiamo con $P(L_i|s)$ la probabilità che il nodo *mobile* sia nella posizione L_i dato il vettore dei segnali ricevuti s e con $P(L_i)$ la

NODI <i>anchor</i>	1NN	2NN	3NN	4NN	5NN	6NN
132	1.75	1.47	1.29	1.23	1.38	1.31
99	1.63	1.52	1.38	1.31	1.36	1.39
66	1.74	1.47	1.51	1.60	1.52	1.60
33	1.78	1.93	1.94	1.72	1.99	2.12
16	2.55	2.34	2.65	2.98	3.41	3.99

Tabella 2.1: Errore medio nella stima della posizione al variare del parametro k e del numero di nodi *anchor* (unità espresse in metri).

probabilità che il nodo *mobile* sia nella posizione L_i . Possiamo ottenere la seguente regola di decisione basata su una probabilità *a posteriori*:

$$\text{Scelgo } L_i \text{ se } P(L_i|s) > P(L_j|s), \\ \forall i, j = 1, \dots, n, \quad j \neq i. \quad (2.5)$$

Utilizzando la regola di Bayes e assumendo, non possedendo informazioni *a priori* sulla posizione dei nodi, che $P(L_i) = P(L_j)$, $\forall i, j$, dalla (2.5) otteniamo una regola di decisione basata sul *likelihood*:

$$\text{Scelgo } L_i \text{ se } P(s|L_i) > P(s|L_j), \\ \forall i, j = 1, \dots, n, \quad j \neq i \quad (2.6)$$

Utilizzando la (2.6) e un modello statistico appropriato è possibile determinare la posizione del nodo *mobile*.

2.2.2 *k-nearest-neighbor (kNN)*

In questo metodo vengono cercate le k corrispondenze più vicine tra la *fingerprint* misurata nella fase di localizzazione e quelle raccolte durante la fase di misura, archiviate nel database. La corrispondenza viene solitamente cercata con il metodo dei minimi quadrati. Successivamente verranno mediate le k posizioni, ottenendo una stima della posizione del nodo *mobile*. Nella tabella 2.1 vengono riportati alcuni risultati dettagliatamente analizzati in [10]. In questo approccio, k è il parametro adeguato per ottenere le migliori prestazioni.

2.2.3 *Reti neurali*

Potendo disporre di un insieme di dati per l'addestramento (*training set*) formato da esempi tipici d'ingresso (il valore RSS)

e dalle corrispettive uscite (le coordinate geografiche del nodo *mobile*), la rete neurale è in grado di imparare ad inferire la relazione che li lega. Questo paradigma di apprendimento è detto supervisionato (*supervised learning*).

Successivamente, la rete è addestrata mediante un opportuno algoritmo (tipicamente la *backpropagation*), il quale usa tali dati allo scopo di modificare i pesi ed altri parametri della rete stessa in modo tale da minimizzare l'errore di previsione relativo all'insieme d'addestramento.

Se l'addestramento ha successo, la rete impara a riconoscere la relazione incognita che lega le variabili d'ingresso a quelle d'uscita, ed è quindi in grado di fare previsioni anche laddove l'uscita non è nota a priori; in altri termini, l'obiettivo finale dell'apprendimento supervisionato è la previsione del valore dell'uscita per ogni valore valido dell'ingresso, basandosi soltanto su un numero limitato di esempi di corrispondenza (vale a dire, coppie di valori input-output).

Per i sistemi di localizzazione basati su reti neurali viene generalmente utilizzata una rete Multilayer Perceptron (MLP) con un unico strato di neuroni nascosti, come mostrato in figura 2.5.

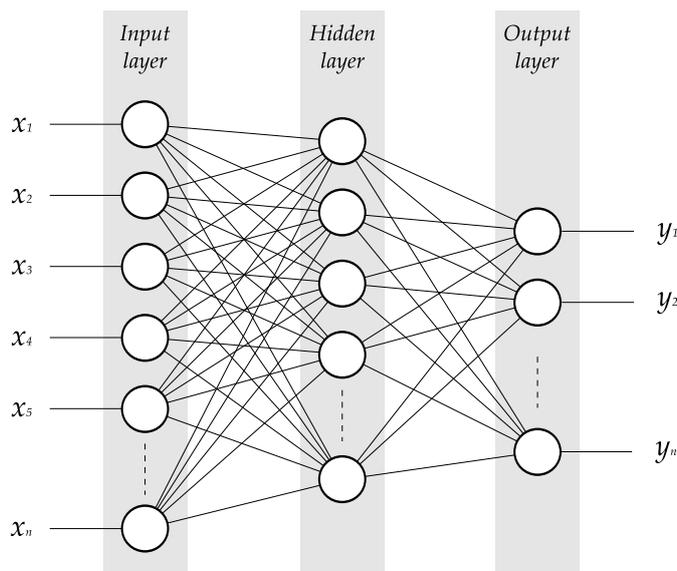


Figura 2.5: Esempio di rete neurale del tipo multilayer perceptron (MLP) con tre livelli.

Il vettore \mathbf{X} , contenente i valori di RSS, passato in ingresso alla rete è moltiplicato per il vettore dei pesi \mathbf{W}_1 dello strato di input per poi essere sommato con il *bias* dello strato di input. Assumendo N la dimensione dei vettori \mathbf{X} e \mathbf{W}_1 , si ha:

$$y_i = g_i \left(\sum_{j=1}^n w_{i,j} x_j + \vartheta_i \right) \quad (2.7)$$

dove $g_i(\cdot)$ è la funzione di attivazione del singolo neurone, $w_{j,i}$ e x_i sono rispettivamente il peso della sinapsi $n_j - n_i$ e l'ingresso i -esimo, ϑ_i è la soglia (*bias*) assegnato ad ogni strato durante la fase di apprendimento. Il risultato y_1 è passato alla funzione di trasferimento dello strato di neuroni nascosto. L'uscita è moltiplicata per il vettore dei pesi W_2 dello strato nascosto per poi essere sommato con il *bias* dello strato nascosto. L'uscita del sistema è un vettore a due o tre elementi, che contengono rispettivamente la stima delle coordinate 2D o 3D della posizione del nodo *mobile*.

2.2.4 Support Vector Machine (SVM)

Tra le tecniche più recenti la SVM è una delle più promettenti. Le macchine a vettori di supporto, chiamate anche macchine *kernel*, sono un insieme di metodi per l'analisi statistica e l'apprendimento supervisionato. Sono particolarmente adatte ad applicazioni per la regressione e la classificazione di pattern. Come mostrato in [3] e [13] le SVM sono state utilizzate con successo nella localizzazione tramite *fingerprinting*.

2.2.5 Smallest M-vertex Polygon (SMP)

Consideriamo M nodi *anchor* che vengono interrogati da un nodo *mobile*. Considerando i valori RSS ottenuti dal segnale ricevuto da ogni nodo *anchor* è possibile costruire un poligono a M -vertici. Ripetendo questa procedura si otterranno più poligoni da cui sarà necessario estrarre il più piccolo. Mediando le coordinate di questo poligono, è possibile stimare la posizione del nodo *mobile*.

2.3 PROSSIMITÀ

Gli algoritmi di prossimità forniscono informazioni indicative sulla posizione di un oggetto. Generalmente viene utilizzato un fitto reticolo di antenne, ciascuna avente una posizione nota.

Quando un oggetto mobile viene rilevato da un'antenna, la posizione dell'oggetto viene fatta coincidere con quella dell'antenna. Se un oggetto mobile viene rilevato da più di un'antenna, la posizione dell'oggetto mobile viene fatta coincidere con la posizione dell'antenna che riceve il segnale più forte. Questo metodo è relativamente semplice da implementare e può essere implementato su diversi tipi di supporti fisici. In particolare, sono spesso basati su questo metodo i sistemi che utilizzano Infrared Radiation (IR) e RFID.

Un altro esempio è l'identificazione di cella. Questo metodo

si basa sul fatto che le reti di telefonia mobile sono in grado di identificare la posizione approssimativa di un telefono cellulare conoscendo a quale cella è agganciato in un determinato momento. Il principale vantaggio di questo metodo è che già oggi è in uso e può essere supportato da tutti i telefoni cellulari.

METODO FINGERPRINTING

In questo capitolo verrà presentato un esempio di implementazione di localizzazione indoor tramite tecniche di *fingerprinting*. Il sistema è stato sviluppato dal gruppo di ricerca SigNET dell'Università di Padova.

3.1 PROCEDURA OPERATIVA

Quando implementiamo programmi per le WSN, dobbiamo affrontare due problemi di difficile risoluzione: dobbiamo scrivere algoritmi che siano efficienti sia a livello energetico sia a livello computazionale.

Lo scenario classico è formato da una rete fissa di nodi *anchor*, di cui è nota la posizione, e da un nodo *mobile* che, una volta entrato all'interno dello spazio coperto dalla rete, utilizzando il metodo *fingerprinting* riesce a localizzarsi. Come visto nel capitolo precedente, è possibile sfruttare l'indicazione di potenza dei segnali trasmessi dai nodi *anchor*, cioè il parametro RSSI.

Possiamo suddividere questo metodo in due fasi distinte: misura e localizzazione.

3.1.1 Fase di misura

Lo scopo della fase di misura è raccogliere il *fingerprint* (l'impronta digitale) di molte posizioni dell'area d'interesse. In questa fase il nodo *mobile* conosce perfettamente la propria posizione geografica (sono note le coordinate x, y, z). L'operazione base della fase di misura consiste in:

1. Il nodo *mobile* interroga tutti i nodi *anchor* nel raggio di copertura.
2. I nodi *anchor* rispondono al nodo *mobile* inviando il loro identificativo univoco. Il nodo *mobile* misura l'RSSI di ogni risposta ricevuta.
3. Le coppie *anchor* ID - RSSI vanno a formare il *fingerprint* della posizione.
4. Il *fingerprint*, insieme alle coordinate del nodo *mobile*, viene inserito in un database. La tabella 3.1 rappresenta una entry di esempio del database.

X	Y	Z	RSSI	ID_ANCHOR
0.00	0.00	0.00	-11	100

Tabella 3.1: Esempio di entry corrispondente a un *fingerprint*.

L'operazione base viene ripetuta in n posizioni diverse nello spazio in cui si intende usare questo sistema, con n da fissare. Per ottenere buone prestazioni si è costretti ad usare un passo di campionamento molto fitto. Per poter raggiungere il giusto compromesso tra precisione del sistema e tempo necessario per ultimare la fase di misura è necessario scegliere accuratamente il valore di n .

3.1.2 Fase di localizzazione

Lo scopo della fase di localizzazione è di determinare la posizione geografica del nodo *mobile* utilizzando i *fingerprint* raccolti nella fase di misura.

L'operazione base della fase di localizzazione consiste in:

1. Il nodo *mobile* interroga tutti i nodi *anchor* nel raggio di copertura.
2. I nodi *anchor* rispondono al nodo *mobile* inviando il loro identificativo univoco. Il nodo *mobile* misura l'RSSI di ogni risposta ricevuta.
3. Le coppie *anchor* ID - RSSI vanno a formare il *fingerprint* della posizione sconosciuta.
4. Il *fingerprint* ottenuto viene inviato al server contenente le informazioni ottenute durante la fase di misura.
5. Il server cerca, usando i *fingerprint* in suo possesso, di determinare la posizione del nodo *mobile*.

3.2 ESEMPIO DI IMPLEMENTAZIONE

A titolo di esempio abbiamo preso in considerazione il sistema implementato dal gruppo di ricerca SigNET. Non entreremo nei dettagli del sistema, ma descriveremo alcune prove eseguite.

Come mostrato in figura 3.1, il laboratorio è una stanza di 10x10 m con un'alta densità di nodi *anchor*, circa 48 Tmote Sky, rappresentati dai bollini verdi t1...t48. I mote sono collocati su un piano sospeso distante 60 cm dal soffitto e 240 cm dal pavimento. I nodi sono posizionati su una griglia con passo di 160 cm

sull'asse x e di 120 cm sull'asse y . I tre rettangoli rossi, indicati rispettivamente con SIG1, SIG2 e SIG3, sono dei piccoli computer utilizzati per programmare e testare i mote.

Per effettuare le misure, abbiamo utilizzato come nodo *mobile* un mote dotato di connettore SMA a cui abbiamo connesso un'antenna omidirezionale. Abbiamo deciso di utilizzare un'antenna esterna per il fatto che l'antenna *patch* integrata nel mote, essendo direttiva, potrebbe aver influenzato le nostre misure.

Per poter effettuare le misure, è stato definito un sistema locale di coordinate cartesiane, come indicato in figura 3.1.

Nella fase di misura, il sistema implementato, per ridurre le variazioni aleatorie introdotte dal canale, fa sì che il nodo *mobile* interroghi ogni nodo *anchor* eseguendo 2 volte la scansione su 4 canali, scelti tra i canali standard. Il valore RSSI preso in considerazione, è il risultato della media dei valori di RSSI ottenuti nelle varie prove. Dovendo creare una mappa completa di *fingerprint* che verranno utilizzate per determinare la posizione del nodo *mobile*, la superficie del laboratorio è stata campionata in maniera uniforme. Come indicato dai pallini rossi in figura 3.1, abbiamo eseguito le misure posizionandoci sulle intersezioni di una griglia di passo 1 m, raccogliendo 81 *fingerprint*.

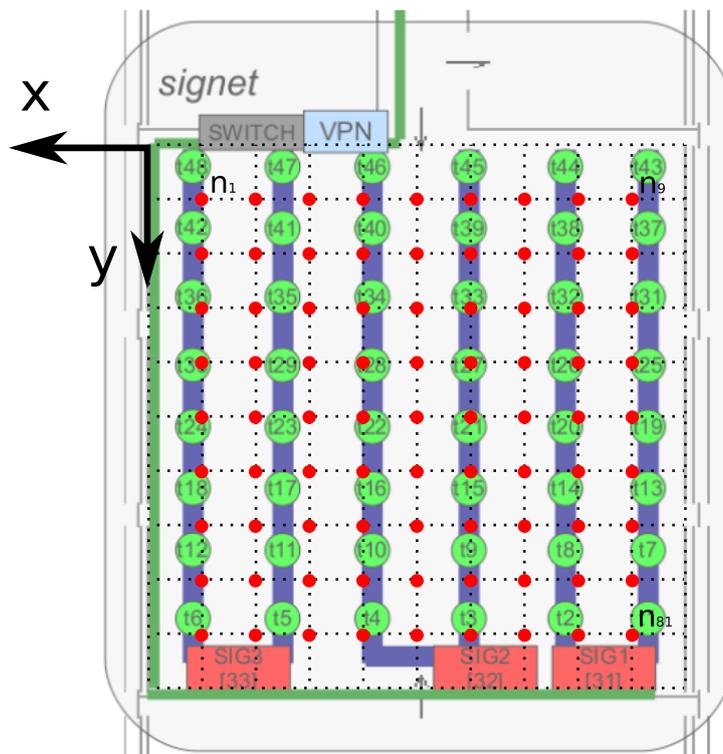


Figura 3.1: Planimetria del laboratorio SigNET. I pallini verdi indicano la posizione dei 48 nodi mentre i pallini rossi indicano le 81 posizioni in cui sono state raccolte le *fingerprint*.

Per semplificare la fase di misura, le misure sono state effet-

tuare mantenendo costante la coordinata z . Il nodo *mobile* è stato collocato approssimativamente a 90 cm dal pavimento e quindi a una distanza di 150 cm dai nodi *anchor*.

Abbiamo eseguito varie prove, utilizzando tutti i nodi disponibili o un sotto insieme, simulando aree più o meno dense. Tutti i mote utilizzavano una potenza di -1 dBm, 0.79 mW. Per determinare se le stime fossero corrette, abbiamo fissato un criterio: la misura è corretta se il sistema identifica la posizione con un errore inferiore a 2 metri, sbagliata negli altri casi. Tutte le prove sono state realizzate in uno scenario reale in cui erano presenti altri dispositivi wireless e persone che si muovevano.

Dalle prove eseguite, abbiamo visto che il sistema in media restituisce dei buoni risultati. Le condizioni migliori, con circa il 95% di stime corrette, sono date da uno scenario in cui poche persone si muovono, le misure vengono eseguite esattamente nei punti in cui avevamo precedentemente raccolto i *fingerprint*, tutti i nodi *anchor* sono attivi. Le condizioni peggiori, con circa il 55% di stime corrette, sono date da uno scenario in cui molte persone si muovono, le misure vengono eseguite nel mezzo di due punti in cui avevamo precedentemente raccolto i *fingerprint*, tutti i nodi *anchor* sono attivi. Abbiamo rilevato che la variazione di antenna e di potenza influiscono molto le misure.

CONCLUSIONI

Dalla nostra analisi emerge che le tecniche di *fingerprinting* sono una buona scelta per la localizzazione in reti radio. Nelle reti di sensori wireless, che richiedono un basso consumo di energia e algoritmi semplici, le prestazioni sono basse. Aumentando le caratteristiche *hardware* dei *mote*, in modo tale che siano capaci di trasmettere più potenza, si otterrebbero prestazioni nettamente migliori.

Uno dei principali svantaggi del metodo *fingerprinting* è la fase di misura in cui è richiesto un operatore che sposti fisicamente il nodo *mobile* e raccolga il *fingerprint* di quella posizione. Fortunatamente abbiamo constatato che questo metodo ha una buona resistenza alle variazioni dell'ambiente, per cui i *fingerprint* ottenuti in fase di misurazione possono essere utilizzati per molto tempo.

Sarebbe interessante analizzare i risultati ottenuti da sistemi che implementano il metodo *fingerprinting* utilizzando i diversi algoritmi presentati nel capitolo 2.

BIBLIOGRAFIA

- [1] Sito ufficiale del progetto TinyOS. URL <http://www.tinyos.net/>. (Citato a pagina 4)
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002. (Citato a pagina 6)
- [3] M. Brunato and R. Battiti. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Network*, 47:825–845, 2005. (Citato a pagina 16)
- [4] Moteiv Corporation. Telos (rev b) : Preliminary data-sheet (12/5/2004). URL <http://www2.ece.ohio-state.edu/~biby/ee582/telosMote.pdf>. (Citato a pagina 4)
- [5] N. S. Correal, S.Kyperountas, Q. Shi, and M.Welborn. An ultrawideband relative location system. In *IEEE Conf. Ultra Wideband Syst. Technol.*, pages 394–397, November 2003. (Citato nelle pagine 10 and 11)
- [6] William C. Craig. Zigbee: Wireless control that simply work. URL <http://www.zigbee.org/>. (Citato a pagina 1)
- [7] David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesc language: A holistic approach to network embedded systems. In *ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI)*, 2003. (Citato a pagina 4)
- [8] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebra-net. *Proc. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 96–107, October 2002. (Citato a pagina 6)
- [9] Hui L., Houshang D., Pat B., and Jing L. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transaction on Systems, Man, and Cybernetics*, 37(6):1067–1080, November 2007. (Citato a pagina 9)
- [10] Binghao Li, James Salter, Andrew G. Dempster, and Chris Rizos. Indoor positioning techniques based on wireless lan. In *LAN, First IEEE International Conferenca on Wireless Broadband and Ultra Wideband Communications*, pages 13–16. (Citato a pagina 14)

- [11] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002. (Citato a pagina 11)
- [12] B. B. Peterson, C. Kmiecik, R. Hartnett, P. M. Thompson, J. Mendoza, and H. Nguyen. Spread spectrum indoor geolocation. *Journal Institute of Navigation*, 45(2):97–102, 1998. (Citato a pagina 10)
- [13] C. L. Wu, L. C. Fu, and F. L. Lian. Wlan location determination in e-home via support vector classification. In *IEEE International Conference Network*, volume 2, pages 1026–1031, 2004. (Citato a pagina 16)

COLOPHON

Questa tesi è stata scritta con L^AT_EX 2_ε utilizzando lo stile Classic Thesis di André Miede.

Versione finale del 22 marzo 2012 alle 13:08.