Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea Magistrale in Scienze di Internet

SISTEMA PER IL MONITORAGGIO AMBIENTALE MEDIANTE UTILIZZO DI UNA RETE DI SENSORI ZIGBEE

Tesi di Laurea in Sistemi e Reti Wireless

Relatore:
Chiar.mo Prof.
Luciano Bononi

Presentata da:
Marco Notarnicola

Sessione Terza
Anno Accademico 2009/2010

Ai miei genitori ...

Introduzione

Per un secolo, dalla scoperta delle onde radio, le tecnologie della comunicazione come il telefono o il telegrafo, sono state utilizzate per connettere luoghi. Negli ultimi decenni invece gli sforzi si sono concentrati con le tecnologie *mobile* e gli *smartphone* , sul connettere persone grazie anche ai cambiamenti sociali introdotti da Internet. La prossima fase, secondo molti esperti consisterá nella connessione degli oggetti, completando in questo modo la parabola " *Places*, *People*, *Things* ".[1]

Se consideriamo quante applicazioni potrebbe avere l'interconnessione e la possibilitá di rendere "*Smart* " gli oggetti utlizzati nella vita quotidiana,ovvero renderli capaci di essere connessi alla rete e di condividere informazioni, sembra realistico pensare ad un futuro molto prossimo dove qualsiasi consumatore in grado di accedere al mercato di consumo, potrebbe arrivare a possedere dai 5 ai 10 dispositivi interconnessi che si scambiano informazioni. Frigoriferi che ordinano la spesa da soli, orologi che inviano dati biometrici in tempo reale al medico, autostrade intelligenti che informano le autovetture sulla situazione del traffico, svariati tipi di senori.

Se anche si dovesse realizzare solo parzialmente la previsione di IBM di 50 miliardi di dispositivi entro il 2020 connessi alla rete, sarebbe un ordine di grandezza superiore rispetto agli 1,5 miliardi di PC e al migliardo di cellulari collegabili a internet, presenti in tutto il mondo. Sono tantissime le applicazioni che potrebbero nascere da queste tecnologie, per questo è stato coniato il termine *"Internet of Things "*.

L'internet degli oggetti è una nuova rivoluzione della rete, gli oggetti

si rendono riconoscibili e acquisiscono intelligenza grazie al fatto di poter comunicare dati su se stessi e accedere ad informazioni aggregate da parte di altri oggetti. Tutti gli oggetti presenti nella nostra vita quotidiana possono assumere un ruolo attivo grazie al collegamento della rete.

Tuttavia saranno gli oggetti più semplici a dominare la scena, entro il 2012, ad esempio, si stima che i sensori fisici genereranno il 20% del traffico internet non video.[2]

Per favorire questo sviluppo però sarà necessaria un'infrastruttura di rete che sia in grado di gestire un traffico dati molto superiore rispetto al livello attuale, questo puó essere possibile solo con l'integrazione e la coesistenza di diverse tecnologie. Infatti é necessario segmentare le tecnologie in base alla grandezza geografica dei contesti in cui queste vengono utilizzate, partendo dall'interno delle singole case *(Personal Area Network)* fino a quelle che agiscono tra diversi continenti, *(Wide Area Network)* per adattare meglio le tecnologie agli scenari in cui devono operare. Ogni scenario infatti richiede delle caratteristiche differenti di comunicazione tra i dispositivi che ne fanno parte, come ad esempio gli ambienti domestici, dove negli ultimi anni si sono affacciati decine di nuovi dispositivi digitali potenzialmente interconnettibili in maniera wireless con possibilitá di controllo e monitoraggio delle attivitá da remoto grazie all'integrazione con la rete *Internet*.

In questi tipi di ambienti che si é avvertita l'esigenza di sviluppare nuove tecnologie che siano in grado di rendere omogeneo la comunicazione tra questi nuovi sistemi. Queste nuove tecnologie di comunicazione peró dovranno rispondere a delle esigenze ed essere in grado di offrire servizi che rispettino alcuni vincoli presenti in questi tipi di contesti, ovvero dovranno essere soprattutto *low cost*, in modo da rendere conveniente il loro utilizzo, in grado di supportare agevolmente un numero elevato di oggetti connessi dinamicamente alla stessa rete, garantire bassi consumi energetici e infine assicurare un servizio di comunicazione sicuro e affidabile.

I fattori necessari per interconnettere gli oggetti sono: l'energia e la connettività . Piè energia e connettività sono necessari e meno oggetti è possibile

connettere.

È per rispondere a queste esigenze che si stanno sviluppando nuova categoria di sistemi di comunicazioni denominata *Wireless Personale Area Network (WPAN)* che include diverse tipologie di sistemi hardware e software per la trasmissione dei dati tra i vari dispositivi utilizzando il canale radio nella maniera più efficiente possibile e con ridotti consumi operativi.

Tra le tecnologie emergenti in questo ambito che si sta affermando in questo ambito vi è sicuramente Zigbee, che si sta affermando sempre di più come protocollo standard per le applicazioni WPAN.

All'interno del presente lavoro è stata analizzata una particolare tipologia di WPAN denominata *Wireless Sensor Network* e dei diversi contesti d'uso in cui può essere utilizzata. Verrà proposta un analisi del protocollo Zigbee, a livello delle caratteristiche che compongono i livelli dello stack e infine verrà presentata la descrizione del progetto realizzato che ha avuto come obbiettivo la progettazione e realizzazione di un sistema per il monitoraggio ambientale. Per l'implementazione del sistema è stata utlizzata una WSN formata da dispositivi sensori Zigbee ed un sistema software scritto in Java che si occupa di inviare i dati ricevuti dalla rete ad un database remoto tramite il quale questi dati verrano messi a disposizione di una applicazione client che effettua la visualizzazione e il monitoraggio dei dati.

Indice

Elenco delle figure

Capitolo 1

Wireless Sensor Network

La pervasività delle comunicazioni wireless sta aumentando in un numero sempre maggiore di attività della vita umana, quindi il panorama delle tecnologie *wireless* è sempre più vasto, per questo è opportuno dividerle a seconda della loro estensione geografica, infatti come mostrato in Figura 1.1 possiamo avere:

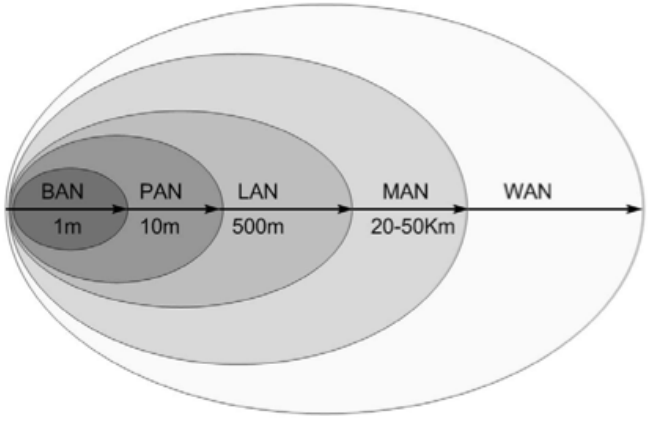

Figura 1.1: Tipologie rete per estensione geografica

- *Body Area Network* (BAN): reti limitate ad un raggio d'estensione di pochi metri di distanza dal corpo umano a caratterizzate da un basso

utilizzo di potenza. Possono essere utilizzate soprattutto in campo medico tramite l'utilizzo di sensori che rilevino dati biometrici.

- *Personal Area Network* (PAN): reti personali, tipicamente wireless anche queste caratterizzate da una bassa potensa trasmissiva.Il raggio di copertura è di qualche decine di metri.

- *Local Area Network* (LAN): reti inizialmente nate cablate e poi sviluppate in ambiente wireless. Hanno un raggio di estensione fino a 500m

- *Metropolitan Area Network* (MAN): grandi reti cablate per il trasporto di grandi flussi di dati, si tratta di reti che hanno come estensione un'area metropolitana come una città.

- *Wide Area Network* (WAN): macro reti a livello mondiale, possono essere cablate o su link satelittare, rappresentano il cuore di Internet.

La distinzione è utile per capire a quale ambito fanno riferimento le diverse tecnologie di *networking* esistenti, infatti negli ultimi anni il bisogno di mobilità e il crescente numero di dispositivi che l'elettronica di conusmo mette a disposizione della gente comune, hanno fatto sorgere un interesse particolare per le tecnologie wireless utilizzate in ambito PAN che sono in grado di far comunicare tutti questi dispositivi tra di loro all'interno dello stesso ambiente. Queste tecnologie prendono il nome di *Wireless Personal Area Network* (WPAN).

Un particolare filone di studi all'interno delle WPAN, ovvero quello relativo alla comunicazione senza fili di dispostivi, chiamati sensori, capaci di acquisire dati dall'ambiente circostante e di reagire a particolari situazioni che si presentano nell'ambiente di riferimento, ha portato alla nascita di una particolare tipologia di WPAN definita con il nome di *Wireless Sensor Network* (WSN).

1.1 Elementi WSN

Le reti WSN come abbiamo visto operano in ambienti WPAN, quindi limitati geograficamente, ma che possono essere densamente popolati di nodi, questo porta ad una maggiore complessità sia in termini di affidabilità del canale di comunicazione che deve essere in grado di ridurre al minimo gli errori dovuti alle collisioni e alle interferenze, sia in termini di costi per implementare la rete.

Di conseguenza possiamo identificare alcune caratteristiche fondamentali per una WSN:

Affidabilità : il meccanismo di comunicazione adottato in una WSN deve garantire la massima affidabilità di trasmissione dei dati, evitando al massimo le collisioni e quindi l'integrità dei dati trasmessi;

Interoperabilità: I dispositivi utilizzati all'interno di una WSN possone essere prodotti da diversi produttori, questo però non deve essere da ostacolo all'integrazione di questi dispositivi all'interno della stessa WSN.

Per questo lo standard di comunicazione adottato deve essere in grado di favorire il funzionamento di dispositivi di diversi tipi di dispositivi e deve permettere anche l'integrazione tra diverse tecnologie di comunicazione, sia wireless che wired.

Scalabilità Una WSN deve essere in grado di gestire in maniera automatica l'aumentare e il diminuire del numero dei nodi che fanno parte della rete, infatti alcuni di questi possono essere dei dispositivi mobili. Deve essere in grado di autoconfigurarsi e di gestire automaticamente la topologia della rete.

Low Cost per rendere conveniente l'adozione di una tecnologia wireless rispetto ad una cablata, bisogna rendere competitivi i prezzi dei dispositvi. Negli utlimi anni, soprattutto grazie alla diminuzione dei costi

dei chip in silicio e all'economie di scala che si possono ottenere nella produzione di massa, il costo dei dispositivi che sono in grado di costruire una WSN ha raggiunto i pochi dollari, rendendo di fatto questo tipo di tecnologie low cost.

Low Data Rate In una WSN i dispositivi non hanno bisogno di un data rate molto alto, in quanto per la natura intrinseca dei dati che questi devono gestire,(comandi on/off, indicazioni della temperatura ecc..) i dati da trasferire hanno delle quantità molto ridotte, quindi i dispositivi possono impiegare delle tecnologie di comunicazione non molto complesse e quindi anche meno costose.

Low Power Alcuni dispostivi di una WSN possono essere posizionati, soprattutto in contesti industriali, in ambienti in cui non è possibile avere una alimentazione di energia continua, quindi alcuni dispositivi possono essere alimentati a batterie.

Questo pone un problema molto importante, infatti è necessario che la durata delle batterie sia preservata al più lungo possibile, sia per limitare gli interventi di manutenzione, sia per la scomodità di doverle cambiare in dispositivi che si possono trovare all'interno di ingranaggi meccanici mobili.

Per questo motivo le tecnologie applicate nelle WSN devono applicare delle politiche di *energy harvesting* che siano in grado di limitare l'utilizzo di energia da parte dei dispositivi alimentati a batteria ed allungare in questo modo la loro autonomia.

1.2 Contesti di utilizzo delle WSN

Le reti WSN hanno diversi campi di applicazione, come ad esempio *Home Automation, Healtcare, Monitoring, etc ....* Questi sono solo alcuni dei contesti in cui si possono implementare WSN anche se certamente non sono

gli unici e con il passare del tempo saranno sempre di più i settori della vita umana in cui sarà conveniente utilizzare un approccio WSN.

In questa sezione vengono descritti alcuni dei maggiori contesti di applicazione per le WSN.

1.2.1 Home Automation

È una dei maggiori contesti di applicazioni per le WSN, infatti possono essere impiegati in diversi scenari all'interno di un ambiente domestico:

Security System consiste nel posizionamento di diversi sensori, come ad esempio, rilevatori di movimento, di fumo, apertura/chiusura delle porte. Questi dati vengono trasmessi ad un coordinatore centrale che immagazina i dati e li rende disponibili all'utente finale in diversi tipi di formati.

Meter-Reading System I contatori delle società energetiche che operano in contesti domestici, hanno la necessità di leggere i dati periodicamente per generare le fatture dei consumi, inoltre questi dati possono essere utili anche alle singole persone che in questo modo possono controllare in qualsiasi momento il loro consumi e quindi la loro spesa.

Attraverso l'implementazione di *Automatic Meter Reading*(AMR) è possibile creare una WSN che fornisca un monitoraggio e controllo remoto sui contatori residenziali del gas, elettrico, acqua, eliminando in questo modo il bisogno dell'intervento umano e aprendo lo spazio per una serie di applicazioni collegate per l'utente finale.

1.2.2 Industrial Automation

A livello industriale le WSN possono aiutare a tenere sotto controllo i processi produttivi e ad intervenire da remoto su alcune operazioni meccaniche che ne fanno parte, oltre a consentire di tenere traccia degli spostamenti di persone e merci all'interno dell'area di produzione.

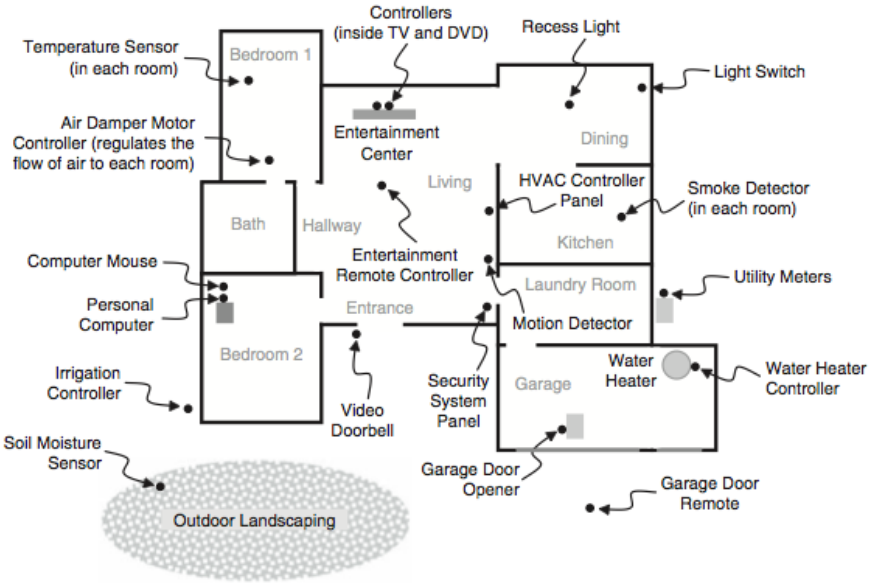

Figura 1.2: Contesto Home Automation

1.2.3 Healthcare

Una delle maggiori applicazioni per le WSN può essere rappresentato dal settore sanitario. Già oggi esistono dei dispositivi sensori in grado di rilevare i dati biomedici dei pazienti, come ad esempio la pressione sanguigna o il battito cardiaco, ed inviare i dati tramite la rete ad un centro di analisi. Questo permette la possibilità di tenere costantemente i pazienti sotto controllo senza la necessitàdell'intervento di un operatore sanitario e permette di rilevare situazioni problematiche che richiedono un intervento di prevenzione del rischio.

1.2.4 Consumer Electronics Remote Control

Negli utlimi anni si è assistito all'avvento di decine di dispositvi digitali nella vita delle persone comuni, utlizzati per i più svariati motivi, soprattutto legati a *Enterteinment*. Attraverso le WSN è possibile creare in questo tipo di contesti un ambiente di remote control che è in grado di gestire tutti

questi dispositivi attraverso un unico centro di comando wireless o addirittura remoto, sfruttando le tecnologi di Internet.

Capitolo 2

Standard WSN

Il panorama tecnologico delle WSN e piú in generale quello delle Lr-WPAN (*Low-rate Wireless Personal Area Network*)si é ampliato notevolmente negli ultimi anni, a causa soprattutto dalle difficoltá di adattare tecnologie wireless giá maturi come gli standard WI-FI(802.11) e Bluetooth (802.15.1) in contesti WSN. Oggi ci sono un gran numero di soluzioni che permettono l'interconnessione e la comunicazione dei dispositivi all'interno della stessa rete, create sia da istituti di ricerca universitari che da istituzioni commerciali, ma la tendenza é quella di convergere verso alcuni standard comuni e condivisi che permettano l'integrazione dei dispositivi di diversi produttori in modo da favorire l'adozione e la diffusione delle WSN. Le tecnologie che maggiormente si stanno affermando sono lo IEEE 802.15.4 e la sua versione commerciale definita con il nome ZigBee.

2.1 IEEE 802.15.4

É uno standard creato dallo IEEE (*International Electric Electronic Engeneer*) per reti flessibili, di basso costo, piccoli consumi energetici e bassi bit-rate. All'interno delle WSN basate sullo standard IEEE802.15.4, infatti vengono utilizzati dei dispositivi che non necessitano di un alto *throughput* e non hanno i requisiti energetici di stack protocollari piú pesanti. Lo standard

IEEE 802.15.4 implementa soltanto i livelli piú bassi della pila protocollare ISO/OSI, cioé il livello fisico e il livello MAC.

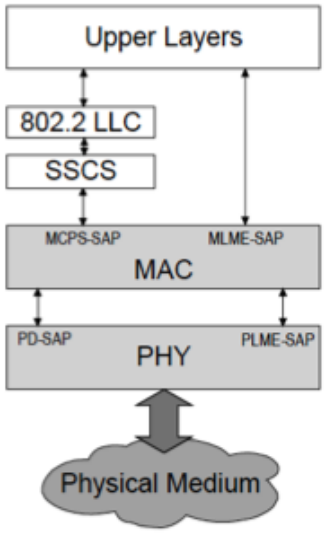

Figura 2.1: architettura IEEE 802.15.4

Lo standard definisce due tipi di dispositivi, *Full Function Device* (FFD) con funzionalitá complete e *Reduced Function Device* (RFD) con funzioni limitate. Ogni rete deve includere almeno un FFD che agisce come coordinatore della WSN, inoltre i dispositivi possono operare anche come *devices* normali. Gli RFD svolgono invece operazioni molto semplici, non devono elaborare o spedire grandi quantitá di dati e possono rimanere inattivi quando non hanno necessitá di comunicazione. Un FFD puó comunicare sia con altri FFD che con gli RFD, mentre questi ultimi possono comunicare solo con altri RFD. In questo modo possiamo costruire una WSN quando almeno 2 dispositivi comunicano all'interno dello stesso *Personal Operating Space* (POS), utilizzando lo stesso canale fisico.

2.1.1 Topologie di rete

Lo standard IEEE 802.15.4 puó operare essenzialmente con due topologie di rete: *Star Topolgy e Peer-to-Peer topology* .

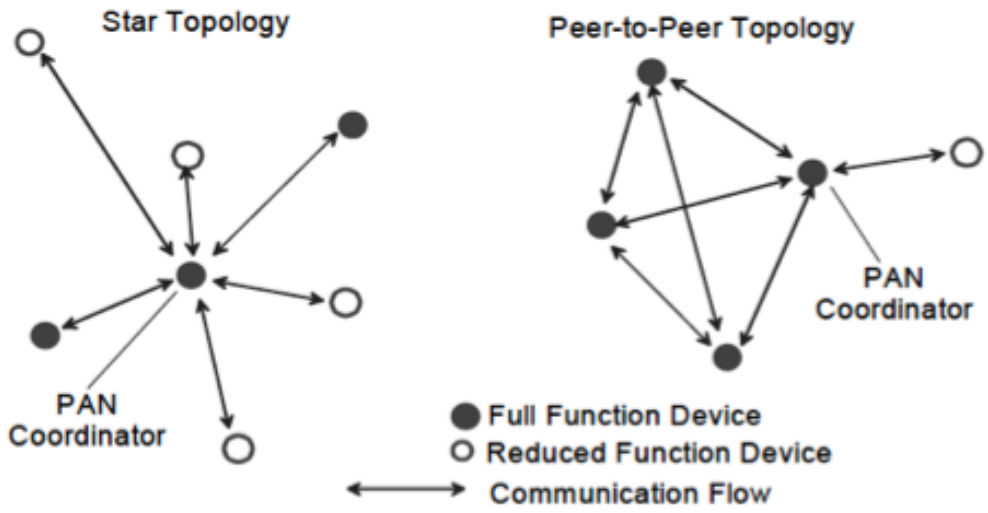

Figura 2.2: Topologie reti 802.15.4

- *Star* : In queste reti, la comunicazione é stabilita tra i *devices* e un singolo *controller* centrale, chiamato *PAN coordinator*, tutti i dispositivi presenti all'interno della rete comunicano direttamente e solo con il coordinatore;
- *Peer-to-Peer* : Anche in questa rete esiste solo un coordinatore, ma gli altri dispositivi possono comunicare tra di loro senza dover passare dal coordiantore; Questa tipologia é meno gerarchica e permette di implementare scenari autoconfiguranti o reti ad-Hoc, inoltre permette anche la comunicazione *multi-hop*, fornendo una maggiore affidabilitá mediante percorsi multipli tra sorgente e destinatario
- *Cluster-tree* : É una particolare tipo di rete P2P dove molti dispositivi sono FFD e coordinano i singoli *cluster*, assumendo il compito di

cluster-head(CLH) e dispositivi che fanno parte dei cluster possono essere considerati come nodi figli e agganciarsi al CLH. Uno solo di questi FFD diventa il coordinatore globale della rete.

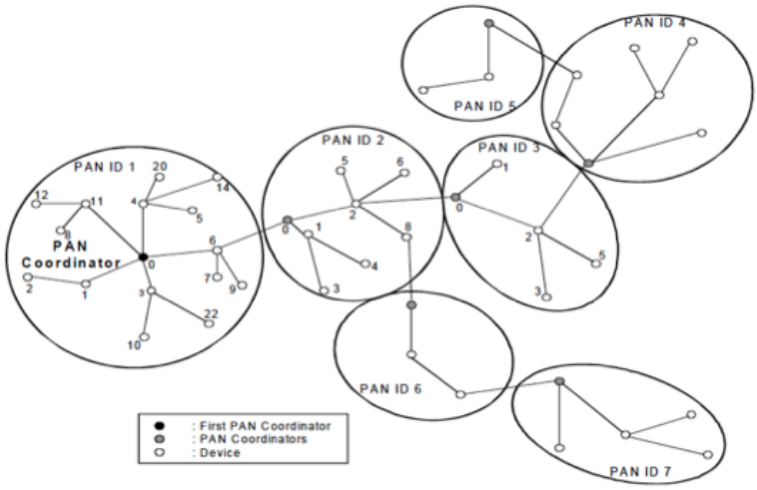

Figura 2.3: Topologie cluster-tree

2.1.2 PHY layer

Il livello fisico dello standard IEEE802.15.4, fornisce l'interfaccia tra il canale di trasmissione e il livello MAC superiore e fornisce i seguenti servizi:

- Attivazione e disattivazione del transceiver
- Rilevazione dell'energia del canale in uso(*Energy Detection ED*)
- Rilevamento della qualitá del collegamento *Link Quality LQ*
- Selezione della frequenza di comunicazione
- Trasmissione e ricezione dati

	MODULAZIONE	UTILIZZO	DATA RATE	CANALI
2.4 GHz	O-QPSK	Ovunque	250 kbps	16
915 MHz	BPSK	Americhe	40 kbps	10
868 MHz	BPSK	Europa	20 kbps	1

Figura 2.4: IEEE 802.15.4 Frequencies and Data Rates

Lo standard offre tre opzioni di banda di frequenza per il livello PHY. Per tutte e 3 le frequenze viene utilizzata la tecnica *Direct Sequence Spread Spectrum* (DSSS).

Il data rate é di 250Kbps nella banda dei 2.4GHz, 40Kbps a 915MHz e 20Kbps nella banda a 868MHz.

A loro volta le frequenze all'interno di queste bande, sono suddivise in canali logici. Esiste un solo canale tra 868 e 868.6MHz, 10 canali tra 902 e 928MHz e 16 canali tra 2.4 e 2.4835GHz, come mostrato nella figura 2.4.

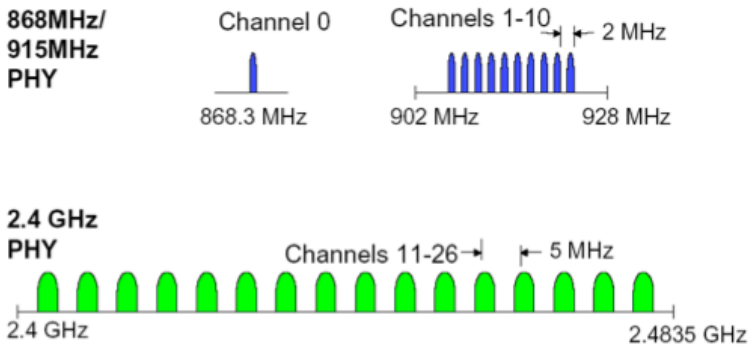

Figura 2.5: IEEE 802.15.4 channels

L'alto data rate possibile a 2.4GHz é attribuibile al tipo di modulazione utilizzata per questa banda. I bassi data rate possono essere convertiti invece

in una migliore sensitivitá e in una migliore copertura, mentre gli alti data rate significano soprattutto alti *throughput*, basse latenze o bassi *duty cycle*.

Architettura PHY layer

Il PHY *layer* é composto da un'unitá di comando generalmente chiamato PHY *Layer Management Entity*(PLME), la quale provede a offrire un'interfaccia, attraverso le primitive definite, con il livello MAC. In queste unitá vengono generate e mantenute le strutture dati per offrire il servizio. Come mostrato dalla struttura rappresentata in figura 2.5, il livello fisico offre due servizi accessibili dai *Service Access Point*(SAP):

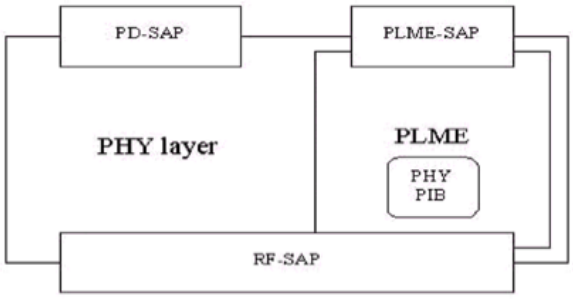

Figura 2.6: IEEE 802.15.4 PHY layer

- PHY *Data Service* accessibile attraverso il PHY Data SAP (PD-SAP)
- PHY *Management Service* accessibile attraverso il PLME-SAP

Il livello fisico quindi fornisce l'accesso al canale radio attraverso il RF-SAP mentre attraverso PD-SAP e PLME-SAP fornisce il servizio di scambio dei dati con il protocollo MAC.

Receiver Energy Detection (ED)

La misurazione di ED del livello fisico viene effettuata per essere utilizzata dai livelli superiore come informazione per la scelta del canale. In particolare si tratta di una stima della potenza segnale ricevuto.

Il risultato dell'analisi ED viene riportato come un intero a 8 bit. Il minimo valore di ED (0) indica che la potenza ricevuta é inferiore a 10dB sopra uno specifico *receiver sensivity*.

Link Quality Indication (LQI)

La misura LQI caratterizza la qualiteị pacchetti ricevuti. La misurazione può essere ottenuta usando ED, *signal-to-noise ratio* o la combinazione di questi due metodi. Il massimo e il minimo valore di LQI indica bassa o alta qualità del segnali sul link.

Clear Channel Assessment (CCA)

Il CCA viene eseguito seguendo almeno uno dei seguenti tre metodi:

- *Energy above threshold.* CCA indica il link occupato quando viene rilevato un livello di energia al di sopra della soglia ED.
- *Carrier sense only.* CCA indica il link occupato solo se viene rilevato un segnale con la modulazione e lo spreading caratteristiche di IEEE 802.15.4
- *Carrier sense with Energy above threshold.* CCA indica il link occupato quando rileva le caratteristiche di modulazione e spreading di 802.15.4 e il livello di energia ricevuto superiore alla soglia ED.

Formato Physical PDU

Il PPDU è il pacchetto di livello fisico che viene modulato e spedito dal chip radio sul canale. Una parte essenziale del PPDU è il preambolo iniziale di sincronizzazione, che serve al ricevitore per acquisire la sincronizzazione di bit quando inizia un certo simbolo e di frame quando inizia un determinato campo nel pacchetto dati. I dati provenienti dal livello MAC sono all'interno di PSDU e hanno una lunghezza massima di 127 byte.

Il PPDU, come mostrato in figura 2.7 è composto da tre elementi:

Octets: 4	1	1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Figura 2.7: struttura PPDU

- **SHR (Synchronization Header):** permette al ricevitore di sincronizzarsi. Il preamblolo permette la sincronizzazione dei bit e il SFD(*Start of Frame Delimeter*) indica l'inizio del frame.
- **PHR (Physical Header):** contiene informazoni sulla lunghezza del pacchetto.
- **Payload:** trasporta le informazioni provenienti dal livello MAC.

Modulazione

Lo standard IEEE802.15.4 come abbiamo visto in precedenza utilizza tre differenti bande dello spettro con differenti data rate; Le due bande a basso trasferimento dati, 868 e 915 MHz, usano una modulazione BPSK, mentre la banda a 2.4GHz utilizza una modulazione O-QPSK.

Modulazione a 2.4 GHz : La modulazione utilizzata è una *Offset Quadrature Shift Keyng*(O-QPSK), ovvero una modulazione 16-ary quasi orthogonal che utilizza una particolare sequenza pseudo-casuale di 32 chip per rappresentare 4 bit, realizzando in questo modo uno spreading del segnale (DSSS); Il livello fisico dello standard definisce un ritmo di simboli di 62.5 Ksimboli/s dove ogni simbolo rappresenta 4 bit, per un ritmo di 250 Kbit/s.

Modulazione a 868/916 MHz : La modulazione nelle bande di 868 e 915 MHz utilizza un *Binary Phase Shift Keying* (BPSK) Sequence

DSSS con una sequenza di 15 chip. Il livello fisico dello standard IEEE802.15.4 definisce un ritmo di simboli di 20 Ksimboli/s per la banda a 868 MHz e un ritmo di simbolo di 40 Ksimboli/s per la banda a 915 MHz, dove ogni simbolo rappresenta un bit, per un ritmo di bit di 20 kbit/s nel primo caso e di 40 kbit/s nel secondo.

2.1.3 MAC Layer

Il livello MAC è sicuramente quello che implementa le funzionalità più interessanti dello standard. Questo livello fornisce due servizi: MAC *data service* che abilita la ricezione e trasmissione dei MAC PDU (MPDU), attraverso i servizi messi a disposizione dal livello fisico, e il MAC *management service* che si interfaccia con il MAC sublayer management entity(MLME) fornendo alcuni servizi di controllo.

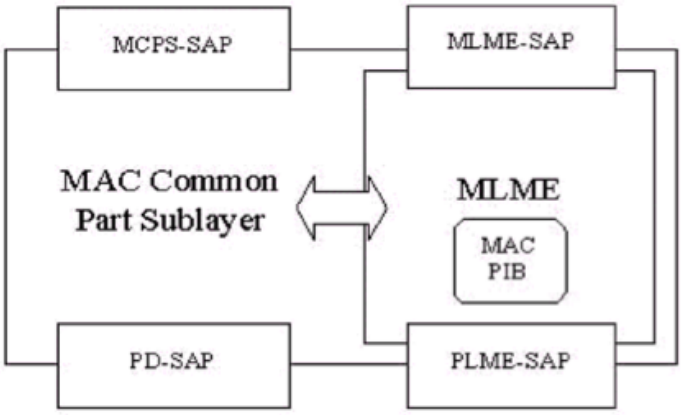

Figura 2.8: struttura MAC layer

Il livello MAC fornisce i seguenti servizi:

- Gestione dei *beacon*.
- Accesso al canale.
- Gestione GTS.

- Controllo sui frame.
- Invio pacchetti di conferma (ACK)
- Associazione/Disassociazione da una rete.

CSMA/CA

Il *Carrier Sense Multiple Access with Collision Avoidance*(CSMA/CA) è il meccanismo principale per la gestione dell'accesso al mezzo condiviso, implementato nel livello MAC dello standard IEEE 802.15.4.

Questo metodo consiste nell'ascoltare il canale prima della trasmissione dei dati e di utilizzarlo solo nel caso risulti libero, se risulta occupato il trasmettitore dovrà aspettare un periodo pseudo-casuale prima di riprovare.

Lo standard prevede anche l'utilizzo opzionale di una struttura detta *superframe* che gestisce in maniera diversa l'accesso al canale.

La gestione dell'accesso al canale mediante CSMA/CA, viene detto in modalità *beaconless*. In questa modalità tutti i dispositivi di una certa rete si contendono il canale secondo il seguente algoritmo

- Aspetto un periodo τ scelto come variabile aleatoria uniforme in $(0, 2^{BE}-1) : \tau \in U(0, 2^{BE}-1)$ dove BE è il *Backoff Exponent*.
- Ascolta il canale. (CCA)
- Se il canale è libero trasmetti.
- Se il canale è occupato incrementa BE come $BE = min(BE+1, aMaxBE)$ dove aMaxBE è il massimo valore di BE consentito. Incrementa il numero di ritrasmissioni (NB) di uno.
- Se il numero di ritrasmissioni (NB) ha superato la massima soglia prefissata (macMaxCSMABackoff) segna un fallimento, altrimenti ricomincia dall'inizio.

Superframe structure

Il livello MAC dello standard 802.15.4 prevede anche un'altra modalità di accesso al canale oltre al CSMA/CA, detta *beacon mode*.

Il *beacon* è un pacchetto particolare che viene utilizzato per creare una struttura detta *superframe* che serve a regolamentare l'accesso al canale.

Il compito di gestire la comunicazione all'interno della rete attraverso i superframe spetta unicamente al coordinatore della rete. Egli trasmette periodicamente dei beacon, che hanno il compito di sincronizzare i dispostivi che fanno parte della PAN gestita dal coordinatore, identificare la PAN e descrivere la struttura del superframe stesso.

La struttura di un *superframe* è delimitata dalla trasmissione di un *beacon frame* da parte del coordinatore, questi sono utlizzati per identificare la rete e per descrivere la natura del superframe. Il superframe può essere composto da due parti, *active period e inactive period*, durante quest ultimo il coordinatore può entrare in modalità *low power* e smettere di funzionare (sleep).

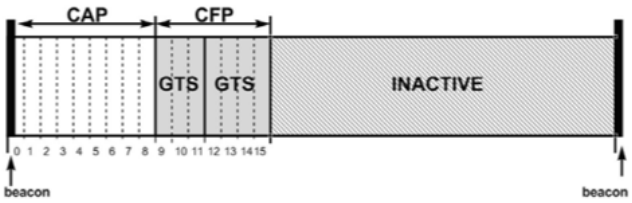

Figura 2.9: struttura superframe

La porzione attiva del superframe invece è divisa in 16 slot di durata uguale ed è composta da 3 parti: il beacon iniziale, il*Contention Access Period (CAP) e Contention Free Period (CFP).*

Il CAP inizia subito dopo la trasmissione del beacon nello slot 0 e termina subito prima dello slot di inizio del CFP, se lo spazio CFP non è allocato dal coordinatore il CAP continua per tutto la porzione attiva del superframe. All'interno del CAP, ogni dispositivo che vuole comunicare dovrà competere con gli altri dispositivi utilizzando il meccanismo CSMA/CA *slotted*, ovvero

ogni tentativo di trasmissione deve avvenire all'inizio di un nuovo slot di tempo all'interno del CAP.

Il CFP viene destinato dal coordinatore della rete a quei dispositivi che hanno bisogno di una bassa latenza o che richiedono uno specifico livello di data rate o di larghezza di banda. Questo è reso possibile utilizzando all'interno del CFP degli slot, chiamati *Guaranteed Time Slot*, in cui l'accesso al canale è garantito senza CSMA-CA.

Modelli di trasferimento dati

All'interno di una rete 802.15.4 ci sono tre tipologie prinicipali di trasferimento dati:

- dal dispositivo al coordinatore;
- dal coordinatore al dispositivo;
- tra due dispositivi;

In una rete con topologia a stella sono presenti solo i primi due, mentre in una topologia peer-to-peer possiamo trovare tutte e tre le tipologie.

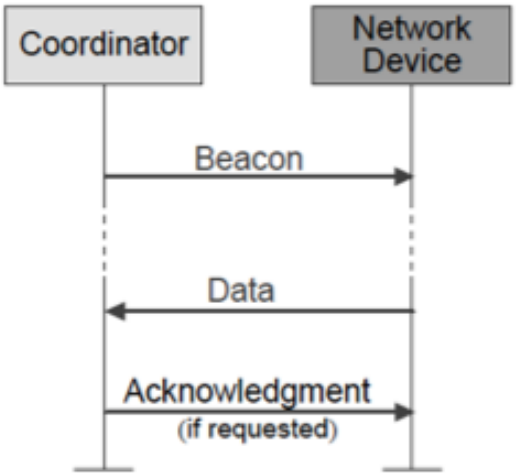

Figura 2.10: trasferimento dati verso il coordinatore con beacon

In una rete *beacon enabled* il dispositivo deve attendere l'invio di un beacon da parte del coordinatore e solo in seguito può trasmettere i dati.

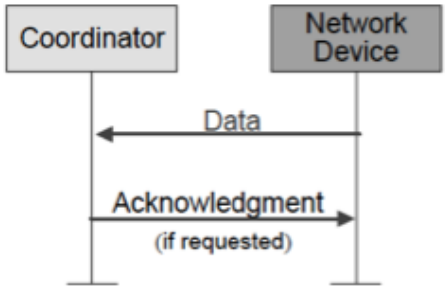

Figura 2.11: trasferimento dati verso il coordinatore senza beacon

In una rete che non è *beacon enabled* il dispositivo semplicemente trasmette i dati che vuole inviare, e rimane in attesa di un ACK opzionale dal coordinatore.

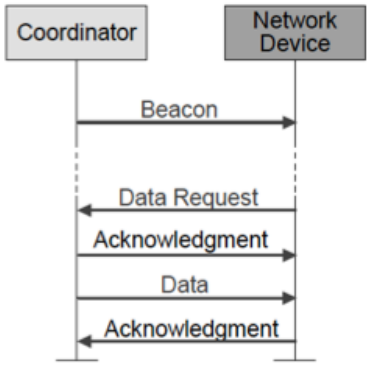

Figura 2.12: trasferimento dati verso il dispsitivo con beacon

Invece quando il coordinatore vuole inviare dati al dispositivo in una PAN *beacon enabled*, lo comunica all'interno del beacon, quindi il dispositivo che riceve il beacon invia una richiesta di dati al coordinatore, il quale trasmette prima un ACK di ricezione della richiesta e successivamente invia i dati.

In un una rete non *beacon enabled* quando il coordinatore vuole inviare dati al dispsitivo, aspetta la richiesta di dati da parte del client. Se questo

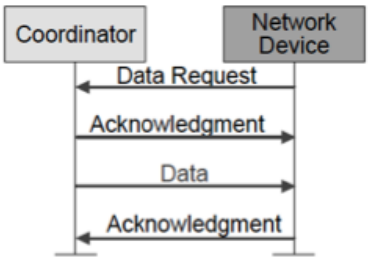

Figura 2.13: trasferimento dati verso il dispositivo senza beacon

avviene il coordinatore risponde prima con un ACK alla richiesta seguito dalla trasmissione dei dati. Se non ci sono dati da trasmettere il coordinatore segnala questo fatto al client.

MAC frame

I pacchetti o frame del livello MAC sono costituiti al massimo da 127 byte. Esistono quattro tipi di datagram :

- *Beacon Frame*: utilizzato dal coordinatore nelle rete *beacon enabled*.
- *Data Frame*: usato per il trasferimento dati.
- *Acknowledgement Frame*: utilizzato per confermare la corretta ricezione di un frame (non contiene né indirizzi né campo dati).
- *Command Frame*: usato per configurare e controllare i nodi

Il primo campo, *Frame Control Field* (FCF) a 16 bit, contiene informazioni sul tipo di pacchetto, sull'indirizzamento(determina la lunghezza dell'indirizzo a 16 o 64 bit) e altri controlli relativi alla sicurezza, alla richiesta di ACK e alla presenza o meno di altri dati da spedire.

Il secondo campo *Sequence Number*, è un campo di 8 bit che indica un numero progressivo univoco per il pacchetto. Seguono poi dei campi per gli indirizzi del mittente e del destinatario e degli identificativi delle reti a cui appartengono.

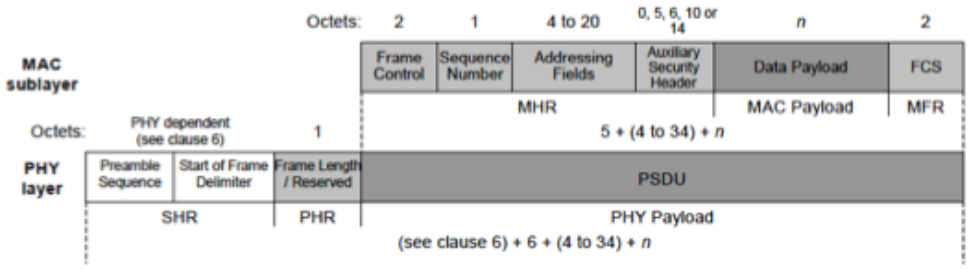

Figura 2.14: MAC DataFrame

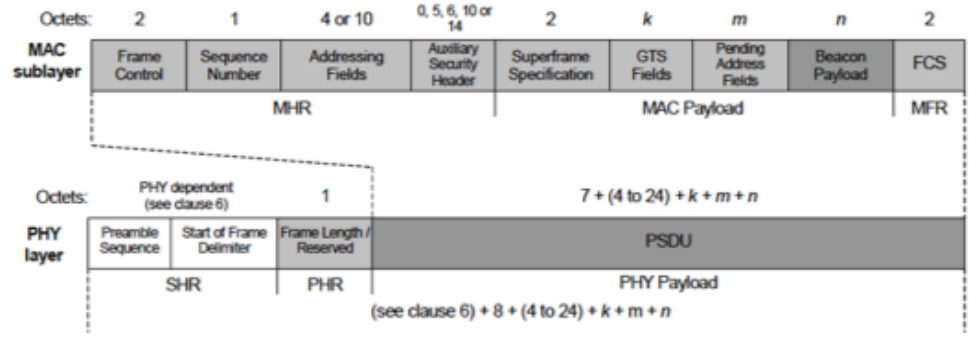

Figura 2.15: Beacon Frame

Il campo *payload* porta i dati veri e propri e infine il datagram termina con un campo *Frame Check Sequence*(FCS) che è costituito da un controllo d'errore di 16 bit di tipo CRC.

2.2 Zigbee

Lo standard di networking Zigbee è stato definito per essere adottato in quei tipi di mercati o di scenari dove non erano adattabili altre tipi di tecnologie wireless. Infatti mentre la maggior parte degli standard wireless sono stati creati per andare veloce, Zigbee è stato ideato per avere un basso data rate.

Inoltre a differenza degli altri stack, Zigbee è stato ideato per essere uno stack leggero adatto per essere implementato sui microcontrollori a 8-bit tipici dei sensori. Infatti questa tecnologia non è stata creata per la comunicazione video o per la connettività ad internet, ma il suo utilizzo è previsto soprattutto per il controllo e il monitoraggio dei sensori, anche per questo è lo standard che si sta affermando più di altri nelle WSN.

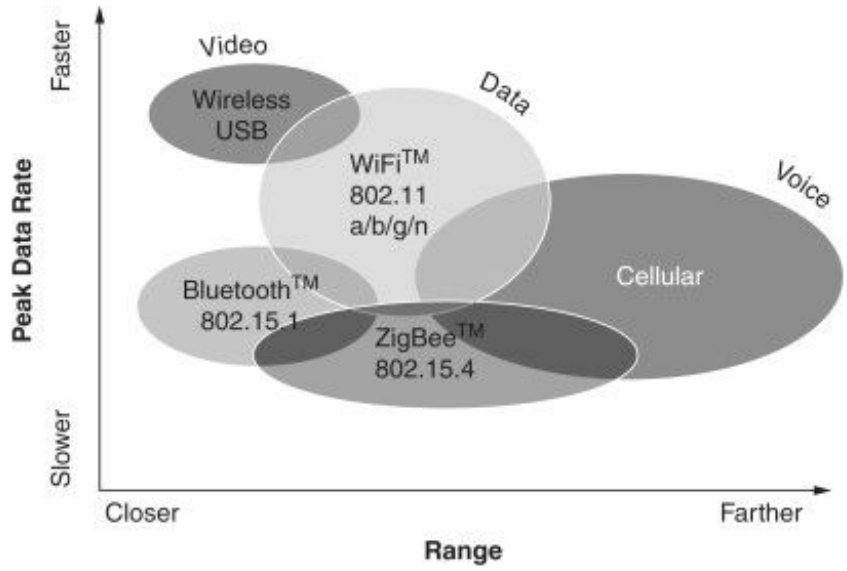

Figura 2.16: Tecnologie wireless

Infatti il mercato delle WSN ha delle necessità uniche rispetto ad altri scenari, per le quali Zigbee si adatta perfettamente, in particolare queste si riferiscono a :

- alta affidabilità
- bassi costi
- bassi consumi di energia
- sicurezza
- globale con utilizzo di frequenze libere
- scalabile

lo standard Zigbee è stato creato dalla Zigbee Alliance, ovvero un'associazione di oltre 285 aziende che lavorano insieme per favorire la creazione di una tecnologia affidabile, cost-effective, che consuma poco, basato su uno standard globale. Il loro obbiettivo è:

- definizione dei software layer
- fornire specifiche per i test e l'interoperabilità dei prodotti.
- promuovere lo standard Zigbee.
- gestire l'evoluzione di questa tecnologia.

2.2.1 Zigbee Stack

Il protocollo Zigbee si basa sui livelli PHY e MAC dello standard IEEE802.15.4, per questo motivo in questa sezione non verranno trattati e si rimanda alla sezione precedente. Ogni livello esegue uno specifico set di servizi per i livelli superiori. Ogni entità di uno specifico servizio fornisce un'interfaccia verso i livelli superiori attraverso un *Service Access Point*(SAP).

Application Layer

Il livello applicazione (APL) dello stack Zigbee è composto da APS *sublayer, Zigbee Device Object (ZDO), Application Objects* definiti dal produttore.

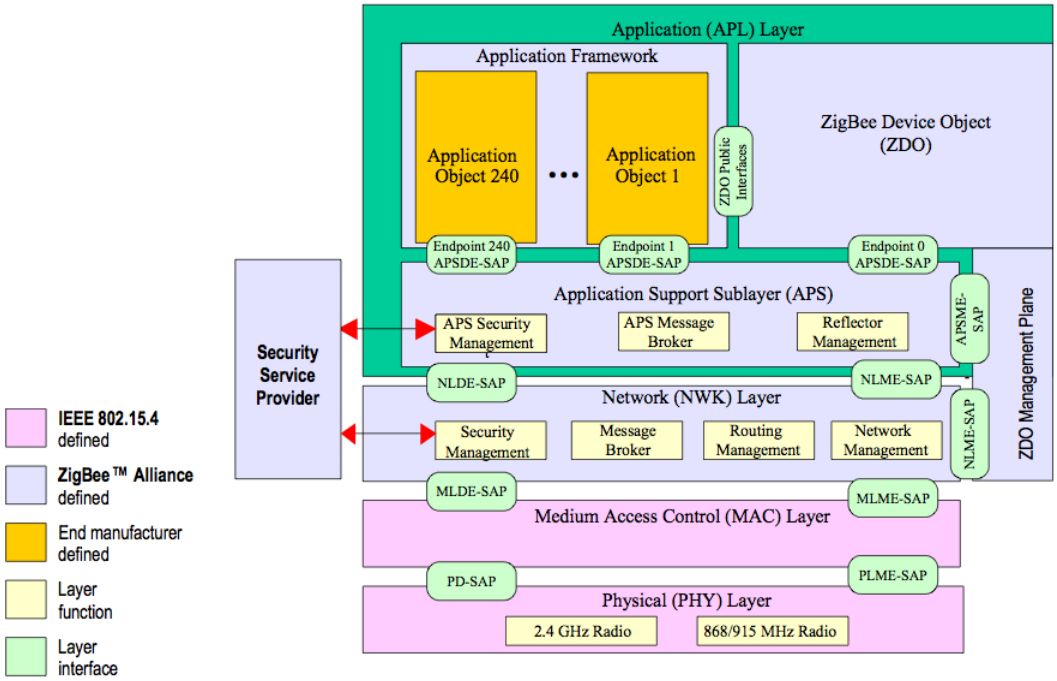

Figura 2.17: Zigbee Stack

Application Framework

Con AF si intende l'ambiente nel quale gli Application Object sono ospitati nei dispositivi Zigbee. Possono essere definiti fino a 240 distinti AO, ognuno indentificato da un indirizzo *End Point* (EP) da 1 a 240.

Esistono anche due EP addizionali che sono definiti per essere utilizzati da APSDE-SAP: EP 0 è riservato per l'interfaccia dati con ZDO e EP 255 è riservato all'interfaccia dati per il broadcast dei dati verso tutte gli AO. Gli EP 241-254 sono riservati per utilizzi futuri.

Application Objects

È un componente cha appartiene alla porzione del livello APS definito dai produttore che attualmente implementa l'applicazione.

Zigbee Device Object (ZDO)

Zigbee Device Object è un'applicazione eseguita sull'EP 0 in ogni Zigbee device.

ZDO tiene traccia dello stato del dispositvo e fornisce un'interfaccia per il Zigbee Device Profile (ZDP) ovvero uno specifico application profile che configura e mantiene Zigbee devices e i servizi sulla rete.

ZDO non interagisce solo con il livello applicazione, ma interagisce direttamente anche con il livello rete.

ZDO controlla il livello rete avvertendolo quando formare o aderire ad una rete e quando abbandonarla, inoltre fornisce un'interfaccia per network layer management serviceZDO rappresenta una classe di funzionalità base che forniscono un'interfaccia tra gli AO, i *device profile* e APS.

ZDO è responsabile dei seguenti compiti:

- Inizializza APS, NWK layer e SSP.
- Raccoglie informazioni di configurazione dagli EP per determinare e implementare *discovery, security management, network managemente e binding management*

ZDO mette a disposizione intefacce pubbliche agli AO per il controllo delle funzioni del dispositivo e della rete mediante AO.

Application Support SubLayer (APS)

APS specifica la porzione del livello applicazione che si occupa delle specifiche del servizio e dell'interfaccia delle applicazione definite dal produttore e ZDO.

Queste specifiche definiscono un *data service* che consente ad un AO di trasportare i dati e di gestire il servizio che si occupa del meccanismo di *binding*. Inoltre definisce il formato e le specifiche del APS frame.

L'obbiettivo di APS è quindi di definire le funzionalità necessarie per consentire le corrette operazioni del livello NWK e delle funzionalità richieste dagli AO definiti dai produttori di dispositivi.

APS fornisce l'interfaccia tra il livello di rete e il livello applicazione attraverso un set di servizi che sono utlizzati sia da ZDO che dagli AO. Questi servizi sono offerti attraverso due entità:

- APS *data entity* (APSDE) attraverso APSDE-SAP: fornisce un servizio di trasmissione dati tra due o più applicazioni all'interno della stessa rete.
- APS *management entity* (APSME) attraverso APSME-SAP: fornisce una varietà di servizi agli *Application Object* che includono sicurezza e il *binding* dei dispositivi. Inoltre mantiene un database degli object gestiti, chiamato APS *information base*(AIB).

APSDE fornisce un servizio dati al livello rete e al livello applicazione sia a ZDO che ad AO per consentire il trasporto di un PDU di livello applicazione tra due o più dispositivi che si trovano nella stessa rete.

APSDE si occupa dei seguenti servizi:

- **Generazione del PDU livello applicazione (NPDU)**
- **Binding**
- **Filtraggio degli indirizzi dei gruppi**
- **Trasporto affidabile**
- **Scarto dei messaggi duplicati**
- **Frammentazione e Riassemblaggio dei NWK frame**

APSME si occupa di correlare due dispositivi in base ai loro servizi e ai loro bisogni. Questo servizio viene chiamato *binding* e APSME deve essere in grado di costruire e mantenere una tabella per memorizzare queste informazioni.

In particolare si occupa:

- **Gestione binding**

- **Gestione AIB**
- **Sicurezza**
- **Gestione dei gruppi**

Network Layer

Il livello rete fornisce una serie di funzionalità per assicurare il corretto funzionamento del livello MAC IEEE802.15.4 sottostante e si occupa di fornire un servizio adeguato al livello applicazione.

Per interfacciarsi con il livello applicazione, il livello rete concettualmente è costituito da due *service entities* che si occupano delle funzionalità necessarie. Queste entità sono il *NWK layer data entity*(NLDE) che si occupa della trasmissione dei dati attraverso il SAP associato(NLDE-SAP) e il *NWK layer management entity*(NLME) che fornisce un servizio di gestione attraverso il SAP associato (NLME-SAP).

NLDE fornisce un servizio dati che consente ad una applicazione di trasportare un APDU tra due o più dispositivi all'interno della stessa rete.

NLDE fornisce i seguenti servizi:

- **Generazione del PDU livello rete (NPDU)**
- **Routing**
- **Sicurezza**

NLME fornisce un servizio di gestione che consente ad una applicazione di interagire con lo stack.

Mette a disposizione i seguenti servizi:

- **Configurazione di un nuovo dispositivo**
- **Creazione della rete**
- **Joining, Rejoining e Leaving dei dispositvi della rete**

- **Indirizzamento**
- **Individuazione dei vicini**
- **Scoperta del percorso di routing**
- **Controllo di ricezione**
- **Utilizzo dei differenti meccanismi di routing (unicast, broadcast)**

A livello rete inoltre viene generato il frame NPDU che è costituito dei seguenti componenti basici:

- NWK header che comprende Frame Control, informazioni di indirizzamento e di sequenza.
- NWK payload di lunghezza variabile, che contiene i dati.

Octets: 2	**2**	**2**	**1**	**1**	**0/8**	**0/8**	**0/1**	**Variable**	**Variable**
Frame control	Destination address	Source address	Radius	Sequence number	Destination IEEE Address	Source IEEE Address	Multicast control	Source route subframe	Frame payload
NWK Header									Payload

Figura 2.18: NPDU frame

2.2.2 Zigbee Network

Tipi di dispositivi

Una rete Zigbee può essere composta dai seguenti dispositivi:

- **Coordinatore :** è il dispositivo che crea e controlla la rete. Questo dispositivo memorizza informazioni sulla rete, incluso quelle riguardanti la sicurezza, infatti funziona anche come *trust center* e come deposito per le chiavi di sicurezza.

- **Router:** questo dispositivo estende l'area di copertura della rete, effettua operazioni di *routing* che permettono di identificare i percorsi tra sorgente e destinatario. Può connettersi direttamente al coordinatore oppure con altri router e può supportare dei dispositvi figli.

- **End Device:** questo dispositivo può ricevere e inviare messaggi, ma non può eseguire nessuna operazione di routing

Topologia Zigbee Network

Una rete Zigbee può assumere differenti topologie, le principali sono a stella e *peer-to-peer o mesh*. Quest'ultima è sicuramente la più interessante in quanto presenta una struttura a maglia di router e end devices interconnessi tra di loro. Ogni router è tipicamente connesso ad almeno due percorsi e trasmette i messaggi ad i suoi vicini.

Una rete mesh Zigbee supporta la comunicazione *multi-hop*, nella quale i dati sono passano da un device all'altro utilizzando il percorso più affidabile e il percorso più efficiente fino alla destinazione. Inoltre le reti mesh aiutano a fornire un'alta tolleranza agli errori, infatti se un dispositivo cade o sperimenta delle interferenze, la rete auto configura i percorsi utilizzando i dispositivi rimantenti.

Indirizzamento

Ogni dispositivo Zigbee dispone di due indirizzi: MAC address a 64 bit (dei quali 24 bit identificano il produttore) ed un network address di 16 bit.

Per stabilire una connessione con una nuova rete, viene utlizzato l'indirizzo MAC, una volta connesso alla rete, il nodo verrà identificato nella rete attraverso il suo network address.

Per le comunicazioni unicast viene utlizzato il MAC address specifico del nodo, mentre per le comunicazioni broadcast viene utlizzato il MAC address generico 0xFFFF.

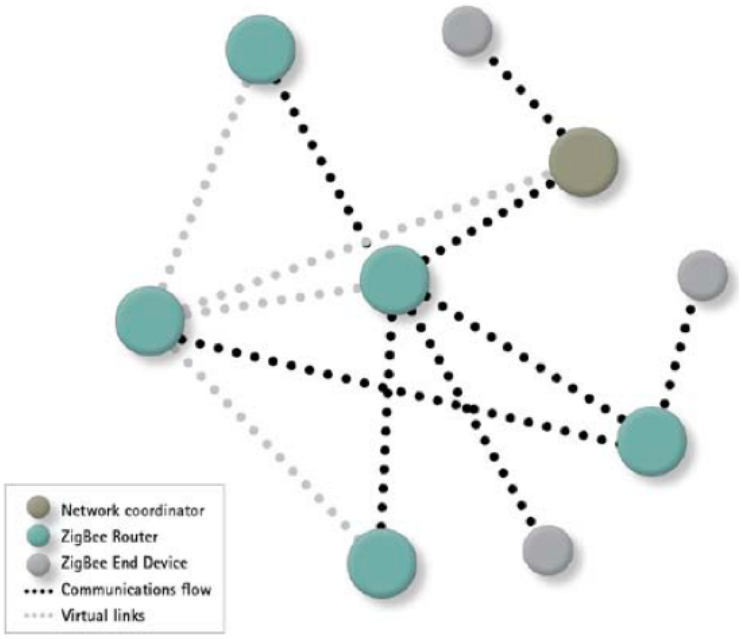

Figura 2.19: Rete Mesh

Zigbee Routing

L'algoritmo di routing implementato nelle reti Zigbee si basa sulle nozioni dell'algoritmo *Ad-Hoc Distance vector* ,ovvero un algoritmo puro di acquisizione del percorso *on-demand* nel quale ogni Zigbee router che partecipa nella consegna di un frame da una particolare sorgente ad una particolare destinazione, mantiene una tabella di routing per il percorso. Questa tabella memorizza la "distanza logica" della destinazione e l'indirizzo del prossimo router nel percorso verso la destinazione.

Il primo obbiettivo dell'algoritmo è di inviare un messaggio *discovery broadcast* solo quando necessario, di distinguere tra la gestione della topologia generale e di diffondere informazioni su i cambi di connettività locale ai nodi vicini che hanno bisogno dell'informazione. Infatti un nodo non ha bisogno di scoprire e mantenere il percorso verso un altro nodo fino a quando non ha bisogno di comunicare con questo, a meno che il primo nodo non stia

fornendo un servizio di spedizione intermedia tra altri due nodi.

Quando un nodo sorgente ha bisogno di comunicare con un altro nodo del quale non ha informazioni di routing nella sua tabella, inizia il processo *Path Discovery*.

Ogni nodo mantiene due contatori separati: *sequence number e broadcast id*. Il nodo sorgente inizia il path discovery inviando in broadcast un messaggio di *route request*(RREQ) ai suoi vicini, che include : *source add. , source sequence number, broadcast id, destination addr, destination sequence number, hop count*.I sequnce number servono per identificare l'aggiornatezza della richiesta. La coppia source addr, broadcast id identifica unicamente un RREQ, quando broadcast id viene incrementato significa che il nodo ha inviato una nuova richiesta. Quando un nodo intermedio riceve il messaggio lo passa ai suoi vicini e incrementa hop count.

Quando un RREQ viaggia da un nodo sorgente a quello di destinazione, viene memorizzato anche il *reverse path* da tutti i nodi verso la sorgente. Qusto percorso viene costruito memorizzando l'indirizzo del nodo precedente da cui si è ricevuto il RREQ. Il reverse path viene mantenuto per un tempo sufficente al RREQ di attraversare la rete e di produrre una risposta al mittente.

Quando il messaggio RREQ arriva al nodo che possiede il percorso attuale verso la destinazione, prima controlla che il RREQ è stato ricevuto nel corso di un collegamento bi-direzionale. Se il nodo non è la destinazione ma possiede il percorso verso la destinazione, determina se il percorso posseduto è attuale confrontando il numero di sequenza della destinazione nella sua tabella con quello presente nel RREQ, se il numero in possesso del router intermedio è minore di quello presente nel RREQ, non utilizzerà il percorso in suo possesso ma farà un broadcasting del RREQ altrimenti invierà un *route request reply*(RREP). Ogni RREQ ha un *time to life* scaduto il quale non può essere ritrasmesso in modo da evitare ridondanze. Quando il nodo sorgente riceve il RREP può utilizzare il percorso verso la destinazione.

2.2.3 Application Profiles, Clusters

Application Profile descrive una collezione di dispositivi impiegati per una specifica applicazione e i messaggi impiegati da questi dispositvi. Per esempio ci sono profile definiti per *Home Automation e Smart Energy* Ci sono due tipi di application profiles:

- **Public Application Profiles:** application software interoperabili tra loro sviluppati dalla Zigbee Alliance che compiono specifiche funzioni.
- **Manufacturer-Specific Profiles:** applicazioni private sviluppate da aziende per gestire un dispositivo Zigbee.

I dispositivi di uno specifico profile comunicano tra di loro attraverlo l'utilizzo di *Clusters* che contengono una serie di attributi necessari per condividere informazioni tra gli application object.

Zigbee Cluster Library (ZCL)

La ZCL è una libreria di clusters che può essere utilizzata da ogni applicazione. Questo consente ai clusters comuni di essere riutilizzati in differenti domini funzionali, per esempio, lo il cluster "illuminazione" può essere utilizzato da ogni applicazione che richiede il controllo delle luci, come HA e *Commercial Building Automation*.

I clusters contenuti in ZCL sono organizzati in un numero di differenti domini di applicazione che comprendono *Lighting, HVAC(heating, ventilation, air conditioning), measurement and sensing, security.*

Binding

Per *binding* si intende la connessione tra due EP che supportano uno specifico application profile e ogni tipo di messaggio rappresentato dal cluster all'interno di quel profile.

I *bindings* possone essere creati tra EP individuali o tra gruppi di EP, come ad esmpio luce ed interruttore, che hanno una correlazione di input e di output (lo stesso cluster ID).

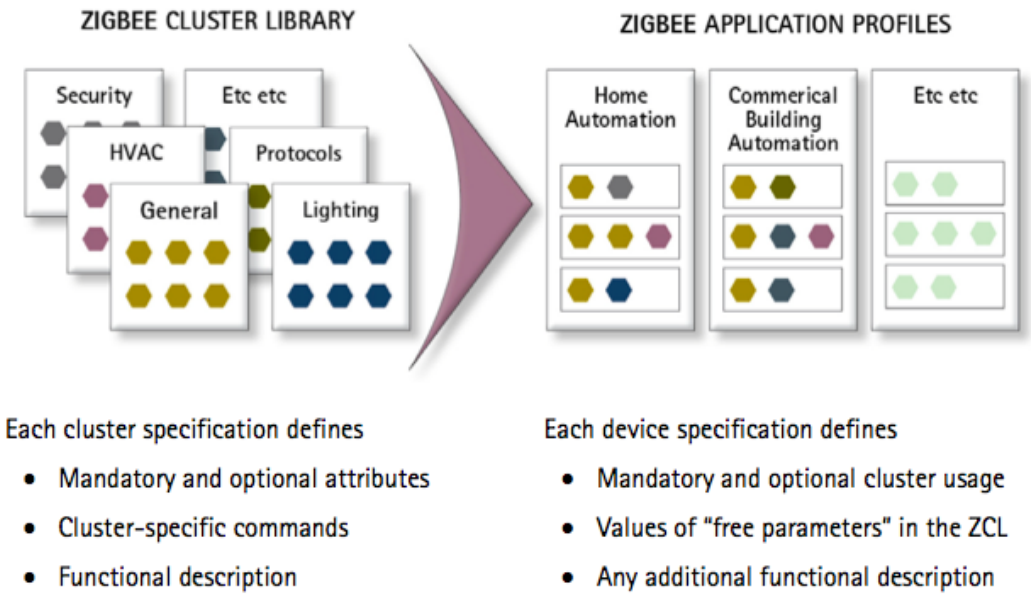

Figura 2.20: Zigbee Cluster Library

Come detto precedentemente un dispositivo Zigbee può arrivare a contenere 240 EP, per questo ogni dispositivo fisico supporta *bindings* multipli.

I vari *bindings* possono essere memorizzate nel dispositivo sorgente, per esempio un controllo remoto potrebbe memorizzare gli indirizzi e EP id di tutte le applicazioni con le quali ha bisogno di comunicare. Questo meccanismo è conosciuto con binding diretto o di sorgente

Le informazioni relative ai *bindings* possono essere anche memorizzate in una *bindings cache* con un dispositivo intermedio che fornisce una tabella di ricerca delle mappe di tutti gli endpoint di origine e di destinazione. Questo è noto come binding indiretto.

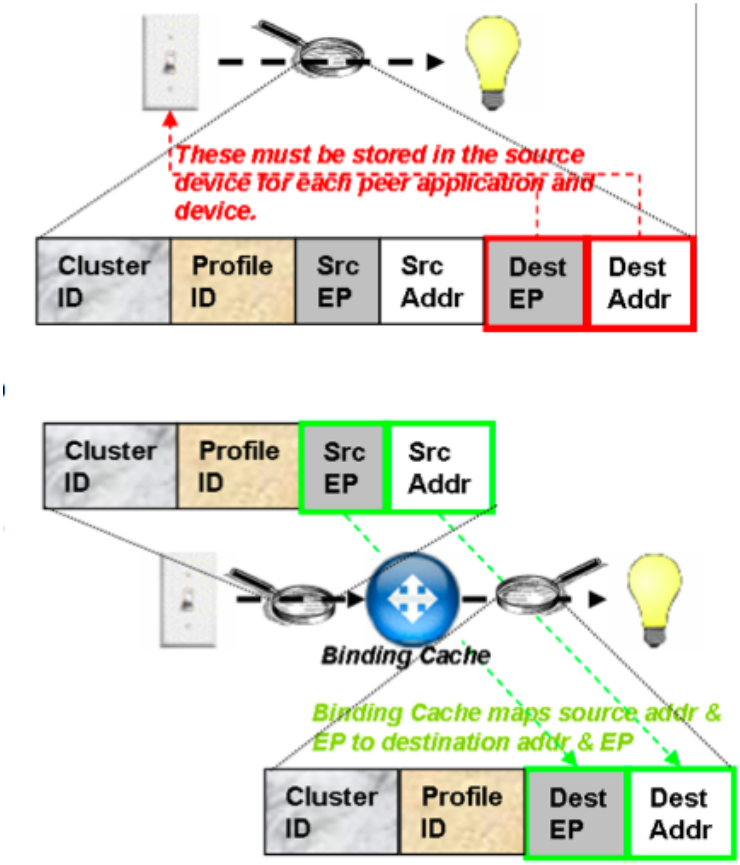

Figura 2.21: Bindings

2.2.4 Security

Lo standard Zigbee, per garantire comunicazioni sicure ed evitare l'intrusione di dispositivi esterni, supporta l'utilizzo di protocolli di criptaggio e autenticazione.

Criptaggio

Il criptaggio è la pratica di modificare un messaggio per mezzo di una permutazione. Lo standard Zigbee supporta l'uso di *Advanced Encryption Standard*(AES).

Il trasmettitore del messaggio usa un algoritmo per criptare il messaggio prima di trasmetterlo e solo il ricevente previsto, conosce come recuperare il messaggio originale. Il messaggio inizialmente non criptato prende il nome di *plaintext* mentre il messaggio criptato viene definito *chipertext*. Se il criptaggio è effettuato su un blocco di dati, l'algoritmo viene definito come *block chiper*.

Zigbee usa un block cheper a 128 bit. La pratica di criptaggio e decriptaggio viene definita criptografia.

In AES ogni algoritmo di criptaggio dei dati è associato ad una chiave. La chiave è rappresentata da un numero binario ed è tenuta nascosta ad ogni trasmissione.

Quando il trasmittente e il ricevitore utilizzano la stessa chiave, questo metodo viene definito a chiave simmetrica. Zigbee supporta solo questo metodo di criptografia.

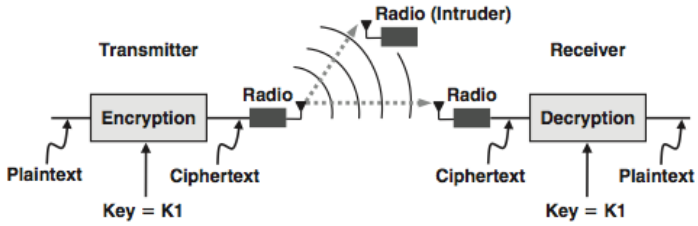

Figura 2.22: Criptaggio utilizzando una chiave simmetrica

Inoltre Zigbee fornisce dei metodi per stabilire e condividere la chiave tra due o più dispositivi nella rete.

Due tipi di chiave sono utilizzate durante una comunicazione sicura: *link key e network key*. Link key vieni condivisa tra due dispositivi e viene utlizzata in comunicazioni unicast. Network Key è condivisa tra tutti i dispositivi della rete e viene impiegata durante le comunicazioni broadcast.

Ogni rete Zigbee che implementa sistemi di sicurezza ha un dispositivo designato, chiamato *Trust Center* che distribuisce sia le link key che le network key agli altri dispositvi. Ci può essere un unico Trust center in una

rete. Il coordinatore della rete determina l'indirizzo del dispositivo che agirà da trust center all'interno della rete.

Ci sono tre metodi per un dispositivo di acquisire una link key, *preinstallation, key transport, key establishment.*

Nel metodo preinstallation, il produttore del dispositivo inserisce la chiave nel dispositivo. In questo modo quando un dispositivo partecipa ad una rete, non ha bisogno di chiedere al trust center la chiave.

In molte applicazioni questo è il metodo più sicuro per acquisire la chiave.

Nel metodo key transport, il dispositivo chiede la chiave al trust center utilizzando un comando appartenente ad APS sublayer. Il trust center può inviare le chiave attraverso una comunicazione non protetta, questo può rappresentare un momento di vulnerabilità.

Key establishment è un metodo per creare delle chiavi random in due dispositivi senza bisogno di comunicazione delle chiavi attraverso una comunicazione insicura.

Questo metodo è basato sul protocollo *Symmetric-Key Key Establishment* (SKKE). I dispositivi che vogliono utilizzare questo metodo devono essere forniti anche di una chiave comune, *Master Key*, questa viene resa disponibile attraverso preinstallation, key transport o user.entered data (es. passw). Il metodo Key establishmente può essere utlizzato solo per ricavare la link key e non per quella della rete.

Nel protocollo SKKE ci sono due dispositivi, un initiator e un responder. L'initiator stabilisce il link key utlizzando il master key disponibile e trasferisce dei dati al risponditore. Il responder utilizza i dati e ricava il link key dai dati stessi. Se la derivazione ha avuto successo i due dispositivi avranno la stessa chiave simmetrica da un utilizzare come chiave per la criptografia dei dati.

Il trust center ha due modi di operare: *commercial mode e residential mode*. In commercial mode, il trust center deve mantenere una lista dei dispositivi e delle chiavi di link, master e network, questo causa un aumento della memoria richiesta quando aumenta la dimensione della rete.

Al contrario, residential mode è progettato per essere utlizzato in applicazioni residenziali che non necessitano di un alto livello di sicurezza. In questo tipo di scenario l'unica chiave che deve essere mantenuta dal trust center è quella di rete.

Nelle reti Zigbee, ogni livello del protocollo (APS, NWK, MAC) è responsabile della sicurezza del frame del rispettivo livello. Per tutti i livelli all'interno dello stesso nodo, viene utilizzata la stessa chiave.

Autenticazione

All'interno dello standard Zigbee viene supportato sia l'autenticazione dei dispositivi che quella dei dati.

L'autenticazione dei dati è eseguita dal trust center e questa si differenzia tra commercial mode e residential mode.

In residential mode se un nuvo dispositivo aderisce alla rete e non dispone di una network key, il trust center ha bisogno di inviarla attraverso una comunicazione insicura, avviando un momento di vulnerabilità. Se il dispositivo dispone della chiave di rete, deve attendere di ricevere una *dummy network key*(tutti zero) dal trust center come parte della procedura di autenticazione, con cui può risalire all'inidirizzo del trust center e aggiornare la sua APS information base. In questo momento il dispositivo è considerato autenticato.

In commercial mode invece, il trust center non invia mai la network key attraverso una comunicazione non protetta, ma può essere inviata una master key su un link non protetto.

Una volta ricevuta la master key, il nuovo dispositivo può far partire il protocollo key establishment. Il nuovo dispositivo ha un tempo limitato per derivare la link key *apsSecurityTimeOutPeriod* scaduto il quale dovrà abbandonare la rete e ripetere associazione e autenticazione.

Una volta stabilita la link key, il trust center potrà inviare la network key attraverso un link sicuro, in questo momento il nuovo dispositivo viene considerato autenticato in commerical mode.

Per quanto riguarda l'autenticazione dei dati è importante essere sicuri che i dati in transito non siano stati modificati.

Per raggiungere questo obbiettivo, il trasmettitore accompagna il frame con uno specifico codice conosciuto come *Message Integrity Code* generato con un metodo noto sia al trasmettitore che al ricevitore. Un dispositivo non autorizzato non sarà in grado di generare questo MIC.

Il ricevitore del frame calcola il MIC e se risulta uguale a quello trasmesso dal trasmettitore, i dati si possono considerare autentici.

Il MIC viene generato utilizzando il protocollo Chaining Message Authentication Code (CCM).

CCM è stato definito per essere utlizzato insieme a AES 128 bit e condivide le stesse chiavi di sicurezza.

Dalla parte del trasmittente, il compito di CCM-AES è di criptare i dati e generare un MIC associato, il quale sarà inviato al ricevitore insieme al frame.

Il ricevitore utilizza AES-CCM per decriptare i dati e generare il proprio MIC dal frame ricevuto per compararlo con il MIC ricevuto, se sono uguali i dati saranno considerati autenticati.

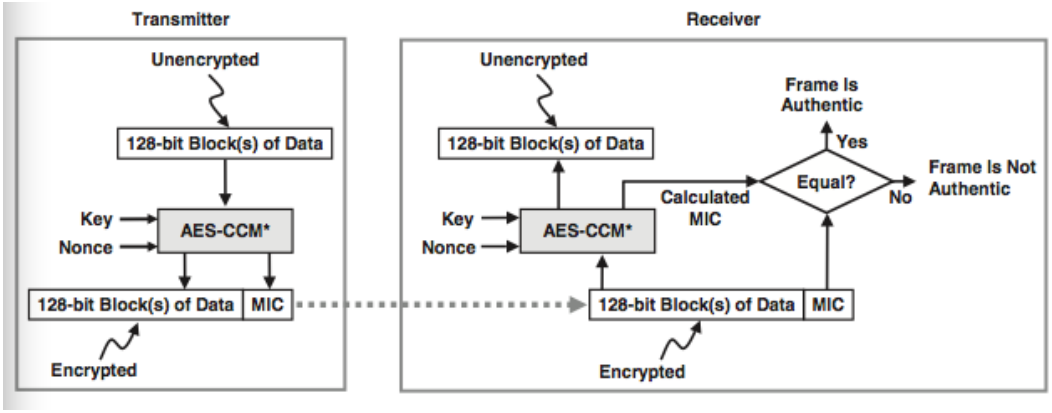

Figura 2.23: Applicazione del MIC nell'autenticazione dati.

Capitolo 3

La piattaforma Freescale 1322x

Per la realizzazione del sistema di monitoraggio, è stata utlizzata la piattaforma della Freescale 1322x.

Freescale Semiconductor, Inc. è leader mondiale nella progettazione e produzione di semiconduttori per i settori automotive, consumer, industriale, networking e wireless. Fa parte inoltre della Zigbee alliance ed è promotore della tecnologia Zigbee.

I dispositivi utlizzati per l'implementazione fisica della WSN appartengono al kit di sviluppo della Freescale denominato "Wireless Connectivity ToolKit ". Questa piattaforma di sviluppo è formata da 4 dispositivi basati sulla famiglia di microcontrollori MC1322x.

I microcontrollori MC1322x sono costituiti da un transceiver a 2.4 GHz, da un processore ARM7, accelleratore hardware sia per il livello MAC 802.15.4 che per lo standard di sicurezza AES, più una serie completa di periferiche per microcontrollori.

Figura 3.1: Freescale Wireless Connectivity Toolkit

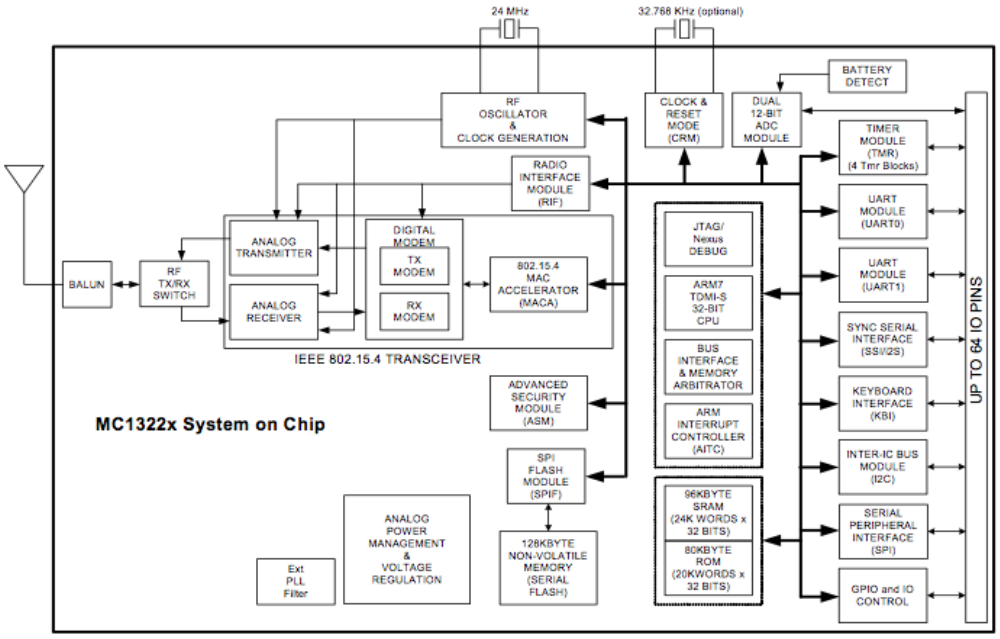

Figura 3.2: Freescale 1322x Block Diagram

Proprio la presenza di un architettura basata su ARM rende questa piattaforma ideale per l'implementazione di applicazioni Zigbee. Infatti, L'architettura ARM (Advanced RISC Machine) indica una famiglia di microprocessori RISC a 32-bit sviluppata da ARM Holdings e utilizzata in una moltitudine di sistemi embedded.

Grazie alle sue caratteristiche di basso consumo (rapportato alle prestazioni) l'architettura ARM domina il settore dei dispositivi mobili dove il risparmio energetico delle batterie è fondamentale.

All'interno dei dispositivi Freescale è presente un microprocessore al 32bit operante a 26 MHz, ovvero ARM7TDMI-S.

La ARM7TDMI è una CPU RISC (*Reduced Instruction Set Computer*) 16-bit/32-bit progettata dalla ARM, basata su architettura ARM v4T e impiegata da molte compagnie costruttrici di semiconduttori come nucleo per microcontrollori e sistemi on chip. È un core versatile studiato principalmente per dispositivi mobili e a bassa potenza. La sua caratteristica principale riguarda l'emulation in real-time.

Il codice scritto per questa CPU è eseguibile direttamente sulle CPU più recenti della famiglia ARM9.

I prodotti più famosi che usano microcontrollori o sistemi on chip basati su questa CPU sono l'iPod di Apple, il Game Boy Advance e il Nintendo DS della Nintendo, la maggior parte dei telefoni Nokia e il Lego Mindstorms NXT.

Tutti dispositivi MC1322x dispongono di 3 tipo di risorse di memoria, RAM, ROM e FLASH. La memoria RAM è di 96 KB, la ROM invece dispone di 80KB e inizialmente contiene il bootstrap, il codice di 802.15.4MAC e i drivers. La memoria FLASH invece contiene 128KB, viene utilizzata all'avvio del dispositivo per caricare e inizializzare la RAM.

3.1 1322x-SRB (Sensor Reference Board)

Il Sensor Node 1322x è un dispositivo IEEE 802.15.4 *compliant evaluation board* basato sulla tecnologia 1322x di Freescale, che può essere utilizzata per applicazioni wireless che vanno dalla semplice connettività punto-a-punto fino a delle complete reti Zigbee mesh.

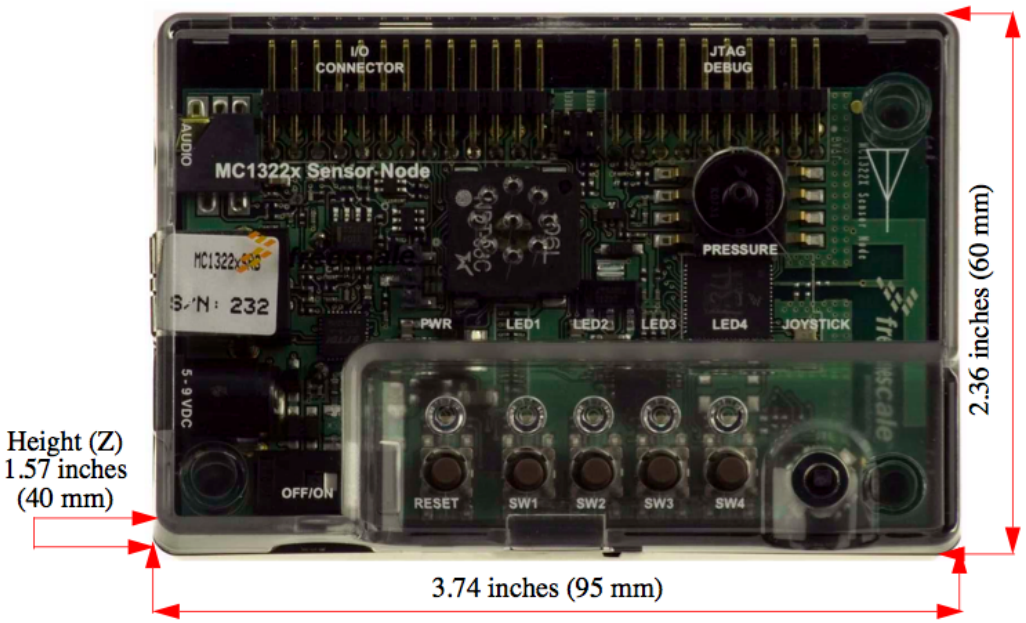

Figura 3.3: Freescale 1322x Sensor Node

Il Sensor Node 1322x fornisce le seguenti caratteristiche:

- IEEE 802.15.4 compliant wireless node con capacitdi funzionamento Zigbee con l'utlizzo dello stack BeeStack della Freescale.
- 2.4 GHz RF transceiver node integrato
- Interfaccia USB integrata compatibile con lo standard USB 2.0 e 1.1
- Audio subsystem con audio jack da 2.5mm per microfono e auricolari, amplificatore audio output per on-board speacker.

- Freescale sensore di pressione
- Sensore di temperatura
- Freescale XYZ tri-axis accelerometro
- Connettore 20-pin per lo standard JTAG
- Power management circuit con regolazione on-board per risorse multiple: può essere alimentato da interfaccia USB, DC power o da 2 batterie AA.
- Interfaccia utente composta da : 4 TACT switch direzionali con un pulsante centrale , 4 pulsanti per testare le applicazioni, 4 LED per il controllo dei processi delle applicazioni.

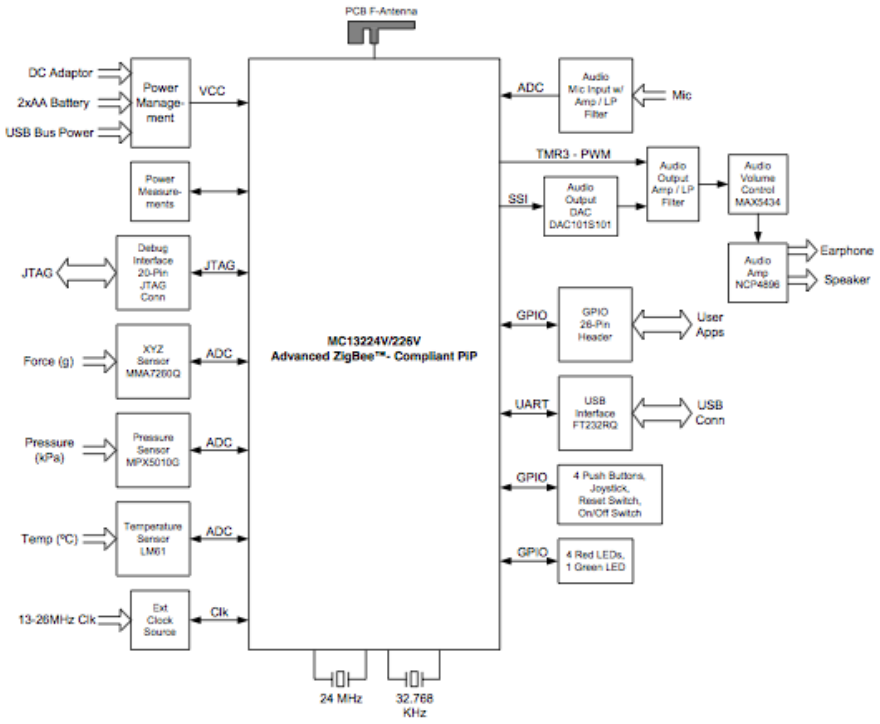

Figura 3.4: Freescale 1322x Sensor Node Block Diagram

3.1.1 Interfaccia Radio

L'interfaccia radio RF presente sul dispositivo opera sulle frequenze comprese tra i 2405 e 2480 MHz, inoltre fornisce una *balun*, ovvero un dispositivo utilizzato per l'adattamento di impedenza tra l'antenna e il circuito ricevente, integrata nella scheda radio che permette al TX/RX switch di essere connessa direttamente all'antenna. Il dispositivo radio permette di programmare la potenza di trasmissione da -30 dBm fino a +2dBm, mentre la sensitività del dispostivo in ricezione è di -95 dBm. Questo permette di ottenere una range di copertura (outdoor) della line of sight di 300 metri.

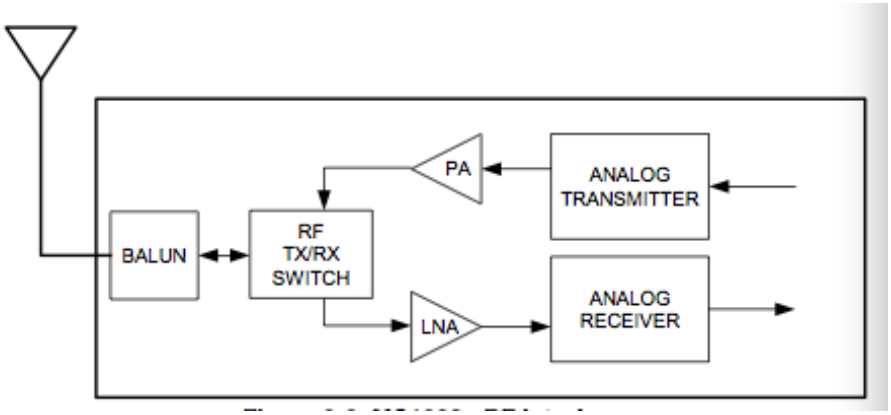

Figura 3.5: Freescale 1322x RF interface

3.1.2 Power management e Measurement

Per consentire la massima versatilità di utlizzo, il Sensor Node può essere alimentato via DC (*Direct Current*) con potenza nominale di 5Vdc, via USB o tramite 2 batterie alcaline AA.

Quando il dispositivo viene alimentato tramite DC o USB la fornitura da parte delle batterie viene automaticamente interrotta.

L'energia fornita da DC o USB viene regolata a 3.3V, inoltre tutte le fonti di energia sono isolate tramite dei diodi. Sono forniti inoltre, un interruttore on/off e dei LEDs per controllare l'alimentazione.

3.1.3 USB Interface

È disponibile un USB plug per collegare il dispositivo al PC o ad altri dispositivi. La porta è connessa a FTDI FT232R USB UART device e appare al PC come una Virtual COM Port (VCP).

3.1.4 Sensori

Il Sensor Node fornisce tre tipologie di sensori, un sensore di temperatura, un sensore di pressione e un accellerometro tre assi low-g.

Sensore Temperatura

Il sensore di temperatura integrato nel dispositivo è un *National Semiconductor* LM61BIM3 e ha un accuratezza di +- 3 gradi centigradi. Il sensore è continuamente alimentato.

Sensore di Pressione

Il sensore di pressione è un Freescale MPXV5010GC6U fully integrated device. Fornisce un output analgico proporzionale alla pressione misurata, la sua accuratezza è del 5 %. Il sensore viene continuamente alimentato, inoltre è dotato di una *axial port* per la connessione di un tubo alla fonte di pressione.

Accelerometro

L'accelerometro presente nel dispositivo è un Freescale MMA7260QR2 . Si tratta di un accellerometro specifico per 4 tipi di sensitività, 1.5g/2g/4g/6g.

Il range è selezionabile via control line e inoltre il sensore può anche essere spento durante il funzionamento del dispositivo.

3.2 1322x Network Node

Il dispositivo 1322x Network Node fornisce una piattaforma per la valutazione dei dispositivi MC1322x, lo sviluppo di applicazioni software che sfruttano le funzionalità di Zigbee e IEEE 802.15.4.

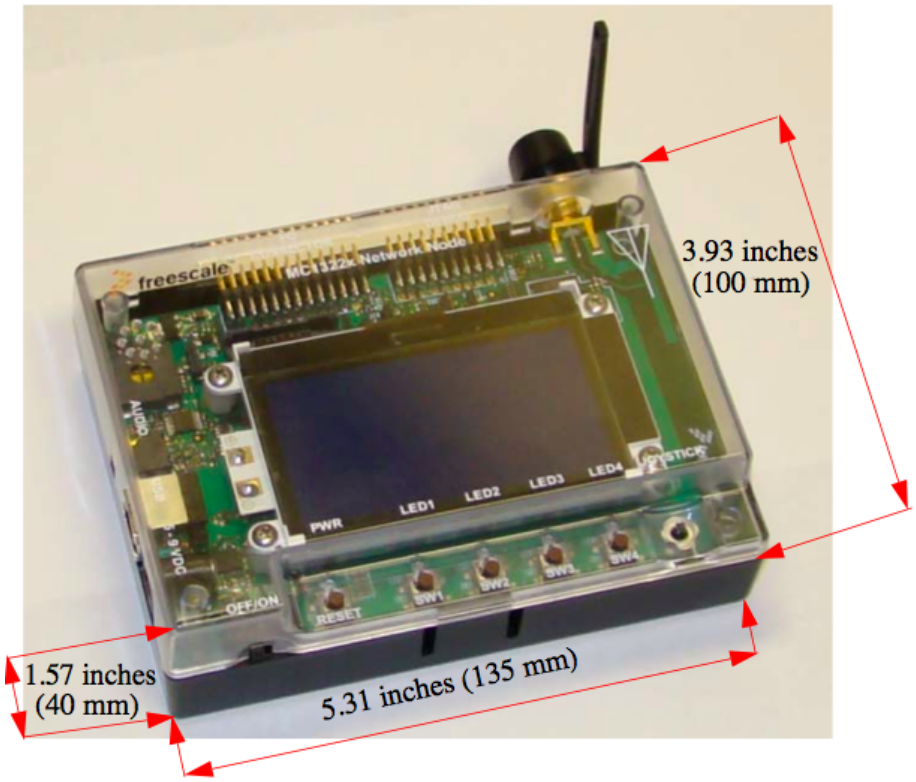

Figura 3.6: Freescale MC1322x Network Node

Il Network Node 1322x è composto dalle seguenti caratteristiche:

- IEEE 802.15.4 compliant wireless node, in grado di supportare Zigbee.
- 2.4 GHz RF transceiver.
- MCU 32-bit ARM7.
- Accelleratore hardware per 802.15.4 MAC e sicurezza AES.

- Interfaccia USB 2.0 e 1.1 compatibile.
- Schermo LCD con LED *backlight*, risoluzione 128x64.
- Audio subsystem con jack da 2.5mm per microfono e auricolari.
- 2 debug/development interface : 20-pin connector per lo standard JTAG e 38-pin MICTOR connector per NEXUS real-time debug interface.
- Power Management, con regolatore per alimentazioni multiple
- Interfaccie utenti con switches and LEDs

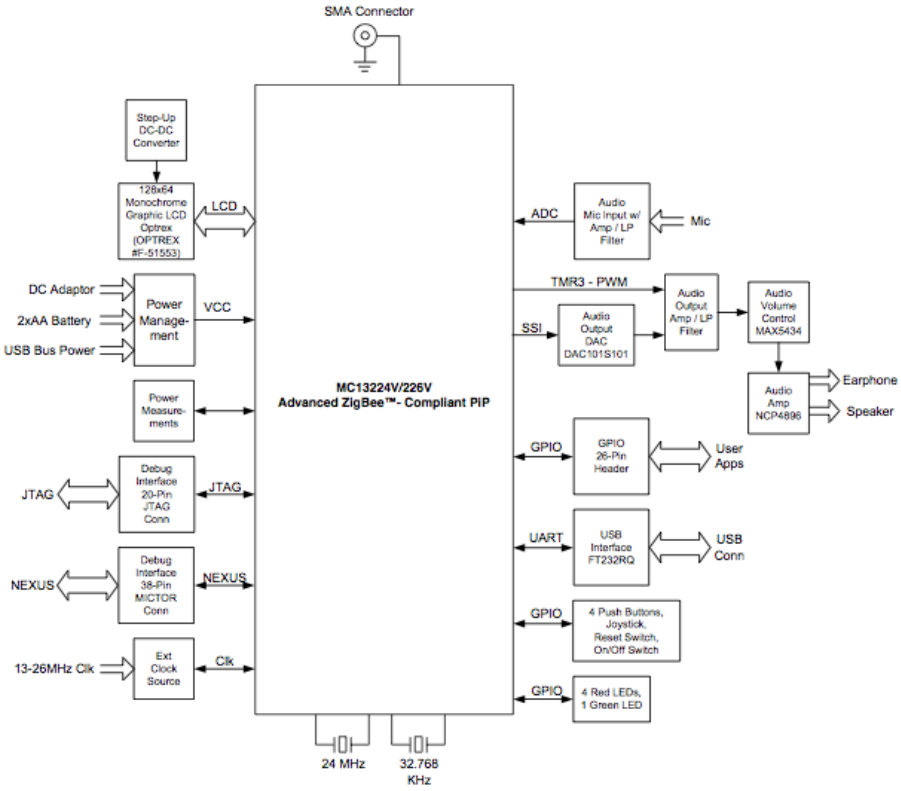

Figura 3.7: Freescale MC1322x Network Node block diagram

3.2.1 Radio Interface

L'interfaccia radio presente nel dispositivo opera sulle frequenze da 2405 a 2480 MHz, utilizza una potenza trasmissiva che tipicamente va dai 0dbm a +2dBm, mentre il receiver sesitivity è di -95 dBm, inoltre è presente un TX/RX switch che permette di utilizzare il dispositivo in modalità trasmissione o ricezione, infatti i dispositivi radio 802.15.4 compliant sono *half duplex*.

La line of sight massima, in contesti outdoor è di 300m.

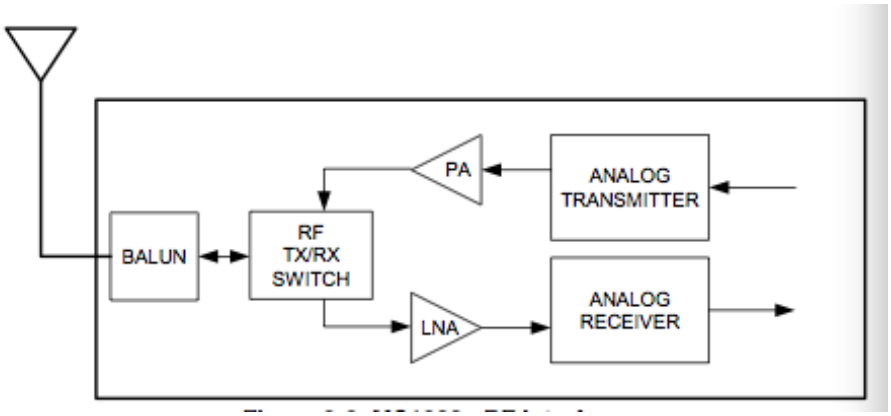

Figura 3.8: Freescale 1322x RF interface

3.2.2 Power Management

Per fornire la massima flessibilità, il Network Node può essere alimentato sia tramite DC (solitamente con un converter AC-DC), tramite USB o per mezzo delle 2 batterie alcaline AA alloggiate nel dispositivo.

Quando si utlizza l'alimentazione da DC o USB, viene interrotta automaticamente l'alimentazione da batteria. Le alimentazioni tramite DC e USB sono regolate a 3.3 V.

3.2.3 USB Interface

Per molte applicazioni o dimostrazioni è desiderabile connettere il Network Node al PC o altri device. La porta USB è fornita di un USB *receptable plug*.

La porta è connessa a FTDI FT232R USB UART device, il quale appare come una *Virtual COM port* (VCP) al PC.

I drivers sono forniti da Freescale con i moduli e attualmente sono disponibili solo per Windows OS.

3.3 Beekit

BeeKit è un'applicazione desktop, che tramite un'interfaccia grafica permette agli sviluppatori di configurare una rete wireless basata sui protocolli supportati, ovvero Beestack (versione di Freescale per Zigbee), IEEE 802.15.4 MAC e il protocollo propietario di Freescale S-MAC.

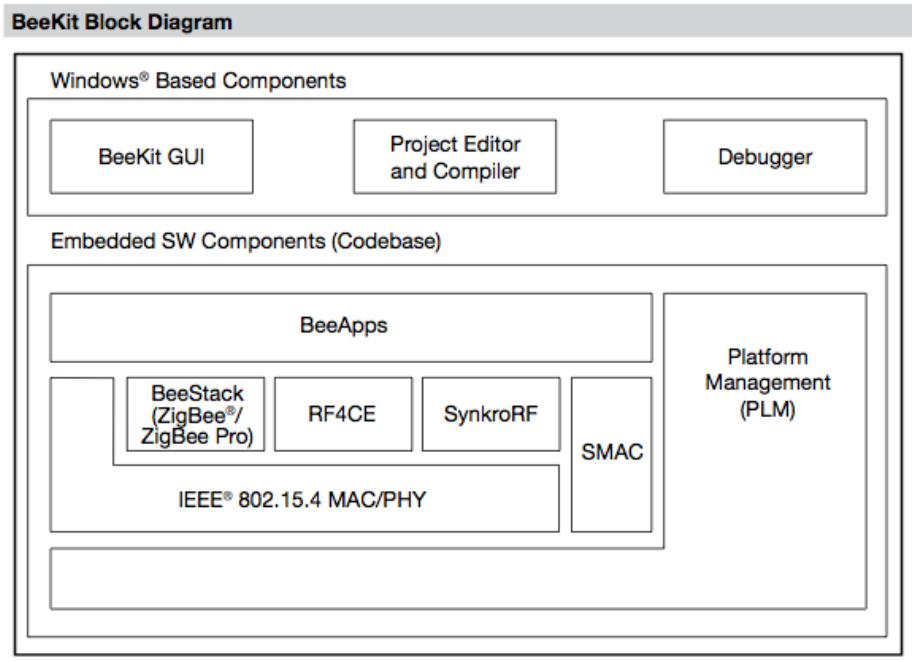

Figura 3.9: Beekit block diagram.

Beekit si occupa di creare applicazioni per i dispositivi da alcuni template, fornendo la capacità agli sviluppatori di settare le propietà (chiamate compile-time option) con le quali configurare l'applicazione e Beestack.

Il progetto creato può essere esportato come un file XML e importato in IAR Embedded Workbench IDE. IAR Embedded Workbench per ARM è un ambiente di sviluppo integrato (IDE) progettato per consentire lo sviluppo ed il debug di applicazioni embedded per il microprocessore ARM.

Il pacchetto include un compilatore C/C++ in grado di raggiungere un livello di ottimizzazione del codice molto elevato, generando così un eseguibile

per FLASH e PROM molto compatto ed efficiente.

IAR IDE prende l'output prodotto da Beekit (XML file) e lo converte in un project file creando il *source directory tree*. L'ambiente di sviluppo si occupa della compilazione di codice C e delle librerie utilizzate nel progetto e crea una immagine binaria del progetto che può essere "flashata " sulla memoria flash del dispositivo utlizzando l'interfaccia JTAG.

Capitolo 4

WSN per il monitoraggio ambientale

Come obbiettivo finale della tesi, è stato creato un sistema basato su una WSN, che utilizzando i sensori presenti nei dispositivi Freescale descritti in precedenza, sia in grado di consentire il monitoraggio ambientale, ovvero di rilevare i livelli sia di temperatura e pressione presenti nell'ambiente di utilizzo sia i dati provenienti dall'accellerometro relativi agli spostamenti.

Quindi il sistema prevederà una parte fisica, costituita dalla WSN che si occuperà della rilevazione dei dati e della loro comunicazione ed una parte software che avrà il compito di visualizzare i dati provenienti dalla rete.

4.1 Topologia

Al fine di costruire la rete di sensori per il monitoraggio, è stato pensato di implementare una topologia a stella. Infatti, nello scenario ipotizzato come contesto di utlizzo, avremo un nodo centrale che svolgerà un duplice ruolo, ovvero quello di operare come coordinatore della rete e di trasmettere i dati al PC. Il resto della rete sarà costituito da una serie di End Device che opereranno come sensori che rileveranno i dati e saranno direttamente connessi al coordinatore.

Figura 4.1: Topologia a stella con coordinatore (C) centrale e sensori (S) collegati.

Nel caso concreto della rete implementata per il sistema realizzato, questa comprenderà due nodi, il nodo coordinatore e il nodo sensore.

Il nodo coordinatore svolgerà i ruoli che gli sono attribuiti dal protocollo Zigbee, quindi si occuperà della formazione della rete e di effettuare i join ai nodi che fanno richiesta di inserimento nella rete.

Il nodo coordinatore, inoltre è collegato tramite USB al PC e quindi si occupa anche di inviare i dati ricevuti dal nodo sensore.

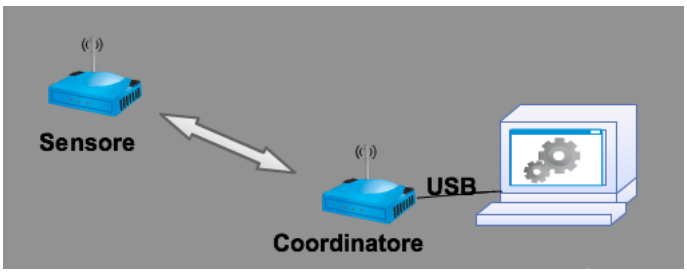

Figura 4.2: WSN

4.2 Hardware e Software utilizzati

Per l'implementazione della rete sono stati utlizzati 2 dispositivi della piattaforma Freescale precedentemente descritta. Per il coordinatore è stato utlizzato il Network Node, mentre per il dispositivo sensore è stato utilizzato il Sensor Node, visto i diversi tipi di sensori di cui dispone.

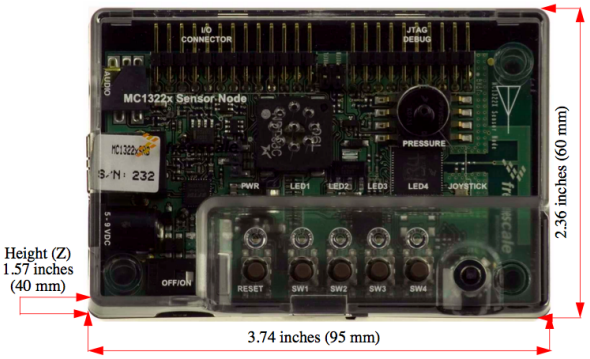

Figura 4.3: Freescale 1322x Sensor Node

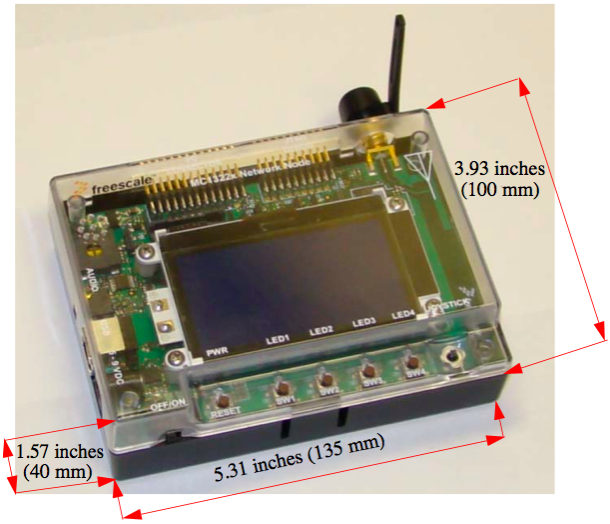

Figura 4.4: Freescale MC1322x Network Node

Per quanto riguarda il protocollo di comunicazione da implementare nei due dispositivi è stato utilizzato BeeStack, ovvero la versione Freescale delle specifiche Zigbee. Beestack implementa tutti i livelli protocollari previsti da Zigbee e svolge tutte le funzionalità dello stack originale.

Per le applicazioni utlizzate dai dispositivi all'interno della rete, è stato utlizzato il profilo Smart Energy.

La creazione dei firmware da implementare all'interno dei dispositvi è stato utilizzato il software Beekit, il quale mette a disposizione diversi tempalte di applicazioni per ogni profile.

Per la rete utilizzata dal sistema di monitoraggio, sono stati scelti 2 tipi di applicazioni:

- **Weather Station (WS):** implementato nel coordinatore, opera come gateway connettendo i dispositivi di monitoraggio con il centro servizi per l'elaborazione dei dati. WS si occupa di consegnare i dati ricevuti dai dispositivi tramite la connessione ad un HOST, tipicamente un PC, che si occuperà invece di elaborare i dati ed inviarli tramite la rete.

  WS opera anche come Zigbee coordinator e mantiene la lista degli Smart Energy devices registrati.

- **Metering Device:** utilizzato dal dispostivo sensore, utilizza uno schema timer based per mostrare continuamente i dati rilevati dai sensori.

Tramite Beekit, oltre alla creazione dell'applicazione che opererà nei dispositivi, sono state settate anche diverse configurazioni dello stack, come le spazio di indirizzamento dei dispositivi e della rete, i tipi di sensori da utilizzare, etc. I campi relativi agli indirizzi sono stati settati in modo che vengano generati automaticamente in maniera random, nel momento della formazione della rete.

Una volta creata l'applicazione con Beekit, questa viene esportato in IAR IDE, dove il codice può essere modificato. Infiene viene effettuato il debug del codice contenuto nel progetto e quindi viene effettuato il download

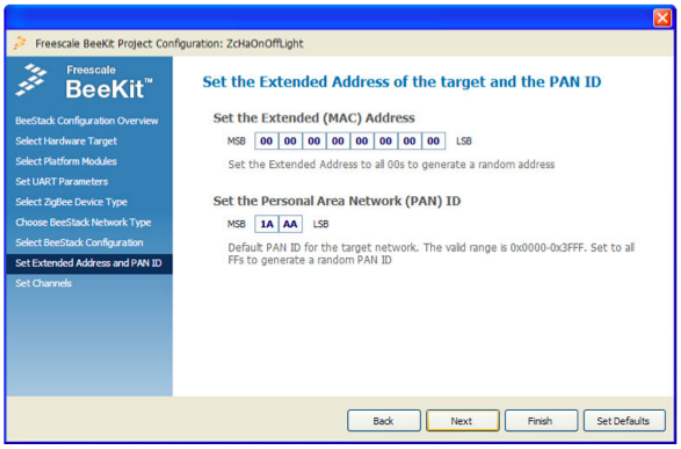

Figura 4.5: Beekit Address configuration

sulla piattaforma hardware, terminato il quale il dispositivo è pienamente funzionante.

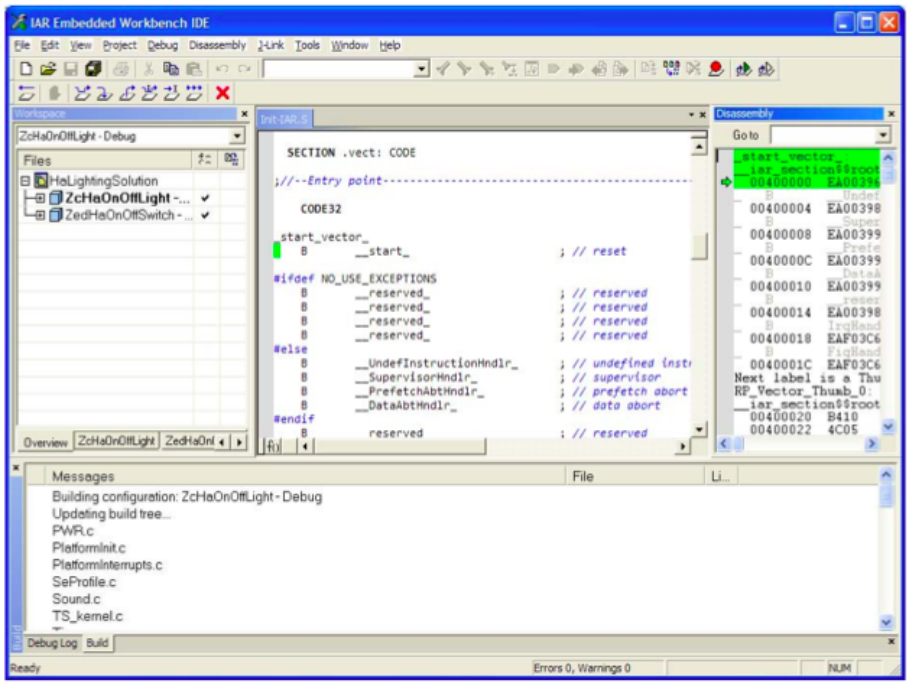

Figura 4.6: IAR System IDE

Capitolo 5

Sistema software per il monitoraggio

Al fine di dimostrare un'approccio tra la tecnologia delle WSN e le tecnologie IT, è stata creata un sistema software in Java per la memorizzazione e la visualizzazione da remoto dei dati rilevati dalla WSN Zigbee. Il sistema è composto da 2 applicazioni Desktop:

- **Z Sender:** opera sul PC connesso al coordinatore della rete attraverso il cavo USB e si occupa di ricevere i dati in ingresso sulla porta USB e di inviarli ad un database remoto per la memorizzazione.

- **Z Monitor:** opera come un'applicazione client che si connette al database a cui sono stati inviati i dati dal Sender, richiede i dati della WSN e li mostra tramite una GUI che rappresenta i valori rilevati dai sensori e dei grafici con gli andamenti registrati nel corso del tempo.

5.1 Architettura generale

L'architettura generale del sistema si presenta come mostrato in figura 5.1.

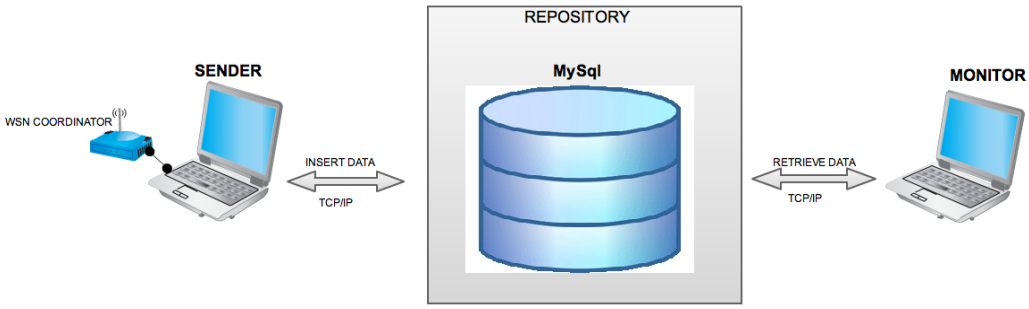

Figura 5.1: Architettura sistema software monitoraggio

le due componenti del sistema, il Sender e il Monitor condividono un database che da una parte riceve i dati dal Sender e dall'altra li invia al Monitor per la visualizzazione.

Si può quindi definire il sistema come un'architettura basata sul pattern architetturale "Passive Repository ". Infatti le due componenti non comunicano con il database in maniera sincrona e non sono direttamente collegate tra di loro da processi specifici, ognuno dei subcomponenti del sistema ignora come i dati siano prodotti o utilizzati dagli altri subcomponenti.

La scelta di utilizzare il pattern repository è stata guidata dalle seguenti motivazioni, la prima è che questo tipo di architettura rappresenta un modo efficiente di condividere grandi mole di dati, inoltre ogni sottosistema non deve preoccuparsi di come i dati sono prodotti o usati dagli altri sottosistemi, vi è una gestione centralizzata di backup, sicurezza, controllo di accesso e recovery da errori e infine è facile aggiungere nuovi sottosistemi.

Inoltre si è deciso di rendere i due sottosistemi indipendenti, in modo da rendere slegata l'attività di invio dei dati da parte del Sender da quella di visualizzazione del Monitor, infatti i dati memorizzati sul database, in questo modo potrebbero essere utlizzati da altri tipi di piattaforme, ad esempio applicazioni client server o web application o anche da altri sistemi di controllo remoto che accedendo ai dati rilevati dalla WSN possono effettuare operazioni opportunamente definite.

5.2 Tecnologie utilizzate

Di seguito verranno descritte le tecnologie utilizzate per implementare il sistema software

5.2.1 Database

Per la creazione del Database è stato utilizzata la piattaforma MySql, ovvero un RDBMS (*Relational DataBase Mangement System*) opensource che utilizza un modello relazionale per la memorizzazione dei dati, cioè i dati vengono separati in tabelle in modo da rendere la loro gestione e interrogazione piè veloce e flessibile. Tra le caratteristiche principali di un DB MySql vi sono:

- capacità di gestire grandi quantità di dati
- condivisione dei dati fra più utenti e applicazioni
- utilizzo di sistemi di protezione e autorizzazione per l'accesso ai dati stessi.

MySql opera all'interno di un MySql server che permette al DB di acquisire funzionalità di accesso tramite internet in maniera veloce e sicura, e di supportare molti programmi client e librerie, administrative tools, oltre ad un ampio range di application programming interfaces (APIs).

Il DataBase MySql utilizzato all'interno del sistema, si chiama Station ed è composto da un'unica tabella chiamata DataSensor strutturata come mostrato in figura:

I dati ricevuti dall'applicazione Sender dalla WSN vengono inseriti all'interno del DB utilizzando delle semplici query SQL.

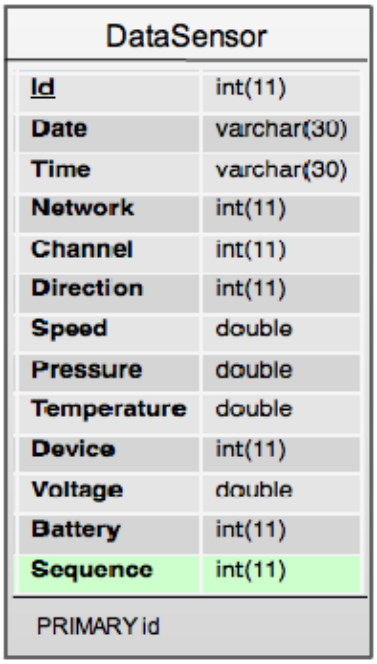

Figura 5.2: Tabella DataSensor all'interno del DB MySql

5.2.2 JDBC

Per implementare la comunicazione con il DataBase MySql remoto sono state utilizzate, sia nel Sender che nel Monitor le API JDBC.

JDBC (Java DataBase Connectivity), è un connettore per database che consente l'accesso alle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DBMS utilizzato. È costituita da una API, raggruppata nel package java.sql, che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati.È orientata ai database relazionali ed è Object Oriented.

Le API JDBC rendono relativamente semplice inviare delle query SQL a dei DB relazionali e supporta tutti i dialetti di SQL.

Il valore aggiunto delle API JDBC è che un'applicazione può accedere virtualmente a qualsiasi tipo di sorgente di dati e ed essere eseguita su ogni tipo di piattaforma hardware con la Java Virtual Machine.

In altre parole, con le API JDBC, non è necessario scrivere un programma per ogni differente DB con cui si vuole interagire. È sufficiente scrivere un singolo programma che utilizzi le API JDBC e il programma è in grado di inviare, utilizzando i drivers specifici per i diversi DB, comandi SQL o altri tipi di comandi alla sorgente di dati appropriata .

Inoltre scrivendo l'applicazione in Java, non si ha la necessità di preoccuparsi di scrivere differenti versioni dell'applicazione per essere eseguite su diverse piattaforme. La combinazione della piattaforma Java e delle API JDBC consente agli sviluppatori di scrivere il programma una volta ed eseguirlo ovunque.

Le applicazioni che utilizzano JDBC fanno uso di Connection object, che rappresentano le connessioni con il database. Una singola applicazione può avere una o più connessioni con lo stesso DB, oppure avere connessioni con diversi tipi di DB. Il modo tradizionale per stabilier una connessione con un DB è invocando il metodo DriverManager.getConnection. Questo metodo prende come argomento una stringa contenente URL del DB. DriverManager class, costituisce invece il JDBC management layer e si occupa di cercare

i dirver per la connessione con il DB presente nell'URL. Infatti la classe DriverManager mantiene una lista di Drivers registrati e quando il metodo getConnection è invocato, cerca tra i drivers presenti nella lista, fino a quando non trova quelli adatti a stabilire la connessione con il DB specificato nell'URL.

```
1 private String urlDB = "jdbc:mysql://localhost:8889/Station";
2 Connection con = DriverManager.getConnection(urlDB, "user", "psw");
```

Quando un applicazione usa la classe DriverManager per la creazione di un oggetto Connection, bisogna fornire un URL come argomento del metodo DriverManager.getConnection.

URL (Uniform Resource Locator) fornisce informazioni sulla posizione di una risorsa su internet e può essere considerato come un indirizzo.

La prima parte di un URL specifica il protocollo utilizzato per l'accesso alle informazioni (in questo caso jdbc) mentre il resto dell'indirizzo indica dove la risorsa dei dati è presente nella rete.

Una volta che la connessione è stabilita, vengono passati i comandi dai inviare al DB. JDBC non pone nessuna restrizione sui tipi di comandi SQL che possono essere inviati. Questo fornisce flessibilità, infatti consente l'utilizzo di comandi specifici per ogni DB.

JDBC fornisce tre differenti interfacce per inviare comandi SQL implementabili attraverso i metodi della classe Connection. L'interfaccie per inviare i comandi SQL e i metodi di Connection sono i seguenti:

- Statement: implementato attraverso il metodo .createStatement() fornisce il semplice invio del comando SQL (senza parametri aggiuntivi)
- Prepared Statement: utilizza prepareStatement() per precompliare comandi che vengono utilizzati spesso e che possono contenere parametri aggiuntivi.
- Called Statement: utilizzato attraverso il metodo .prepareCall () permette di richiamare una procedura da una serie di comandi memorizzati.

```
1 private String adresseDB = "jdbc:mysql://localhost:8889/Station";
2 private String nom = "root";
3 private String pwd = "root";
4 private Connection connection = null;
5 private Statement stmt = null;
6
7 Connection connection = DriverManager.getConnection(addressDB, "user", "psw");
8
9  Statement stm = connection.createStatement();
10 result = select.executeQuery("Select␣TEMPERATURE␣from␣␣DATASENSOR␣␣");
```

5.2.3 JFreeChart

All'interno dell'applicazione Monitor, che si occupa del monitoraggio dei valori rilevati dai sensori, sono stati utilizzati alcuni grafici per la visualizzazione dei dati. Per la creazione dei grafici è stata utilizzata la libreria JFreeChart, una libreria opensource che permette di creare grafice, statici e dinamici all'interno di applicazioni Java. Le librerie sono rilasciate sotto licenza GNU Lesser General Public Licence (LGPL), che ne permette l'utilizzo anche in applicazioni proprietarie.

JFreeChart è utilizzabile sia in applicazioni client che all'interno di applicazioni server, supporta molti tipi di grafici e permette funzionalità di esportazione dei grafici creati in diversi tipi di formati (PNG, PDF)

5.3 Z Sender

Si tratta di una applicazione desktop scritta in Java che opera sul PC collegato al coordinatore della WSN. Lo stream di byte che il coordinatore invia al PC tramite USB, che corrisponde ai dati inviati dal sensore, viene scritto su un file di log come una sequenza di caratteri esadecimali che viene costantemente aggiornato con i dati in arrivo dalla rete.

```
02/07/2011 08:56:57 0x01 0x01 0xCB 0x00 0x01 0x12 0x07 0x50 0x01 0x0E 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:57 0x01 0x01 0xCF 0x00 0x01 0x0F 0x07 0x28 0x01 0x10 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:57 0x01 0x01 0xD3 0x00 0x01 0x11 0x07 0x46 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xD7 0x00 0x01 0x11 0x07 0x41 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xDB 0x00 0x01 0x11 0x07 0x23 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xDF 0x00 0x01 0x11 0x07 0x2D 0x01 0x0E 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xE3 0x00 0x01 0x11 0x07 0x2D 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xE7 0x00 0x01 0x11 0x07 0x32 0x01 0x0E 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xEB 0x00 0x01 0x12 0x07 0x1E 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:58 0x01 0x01 0xEF 0x00 0x01 0x11 0x07 0x32 0x01 0x10 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0xF3 0x00 0x01 0x11 0x07 0x23 0x01 0x0E 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0xF7 0x00 0x01 0x11 0x07 0x50 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0xFB 0x00 0x01 0x11 0x07 0x41 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0xFF 0x00 0x01 0x11 0x07 0x1E 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0x03 0x00 0x01 0x11 0x07 0x32 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:56:59 0x01 0x01 0x07 0x00 0x01 0x11 0x07 0x23 0x01 0x12 0x80 0x50 0x02 0x24 0x01
02/07/2011 11:57:00 0x01 0x01 0x0B 0x00 0x01 0x11 0x07 0x28 0x01 0x12 0x80 0x50 0x02 0x24 0x01
```

Figura 5.3: File di Log con lo stream di byte ricevuto dal coordinatore

La sequenza dei caratteri rappresenta la struttura del pacchetto del livello applicazione generato da Zigbee all'interno dei dispositivi.

		Network Node Data (2 bytes)	Sensor Node Data (11 bytes)					
	Bytes 1-2	Bytes 3-4	Bytes 5-6	Bytes 7-8	Bytes 9-10	Bytes 11-12	Bytes 13-14	Byte 15
Date/Time	Network Device Info	Sequence Number Operating Channel	Wind Direction (0-360°)	Wind Speed (n.n)	Air Pressure (n.n)	Temperature	Battery Voltage (n.n)	Battery Yes/No

Figura 5.4: Struttura data frame

Quando il programma si avvia, l'utente introduce l'indirizzo del database a cui vuole inviare i dati, uesername e password del DB e autorizza la trasmissione. da questo momento si crea una connessione con il database e programma effettua una lettura periodica del file di Log, il quale viene costantemente aggiornato con lo stream di byte ricevuto sulla porta USB, i dati letti dal file di Log vengono decodificati e preparati per essere inviati al DB. Ogni 5 min il programma effettua una query di inserimento con i dati aggiornati e la invia al DB che si occuperà della memorizzazione

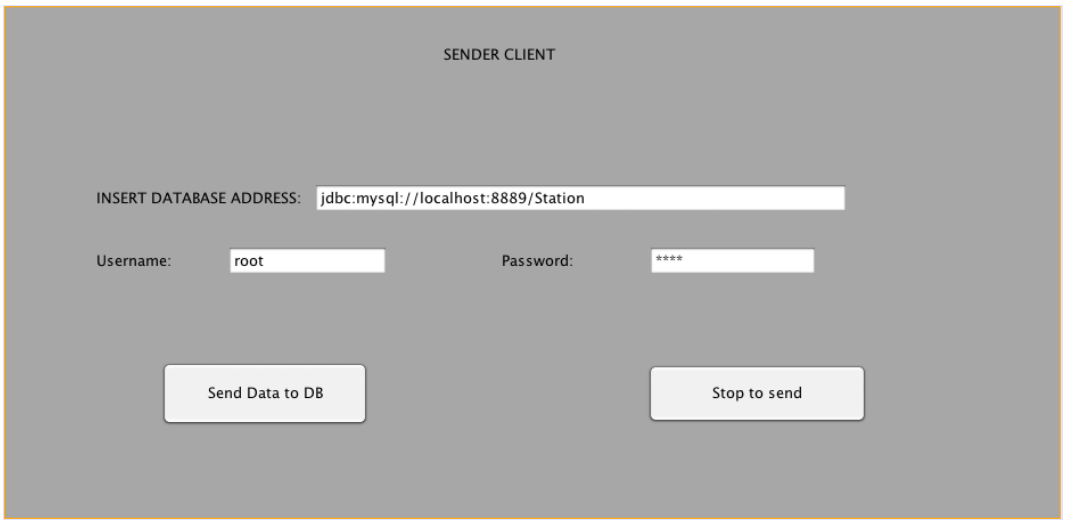

Figura 5.5: ZSender GUI

5.4 Z Monitor

ZMonitor è un applicazione che utilizza i dati memorizzati nel database per implementare un servizio di monitoraggio remoto dei dati inviati dalla WSN.

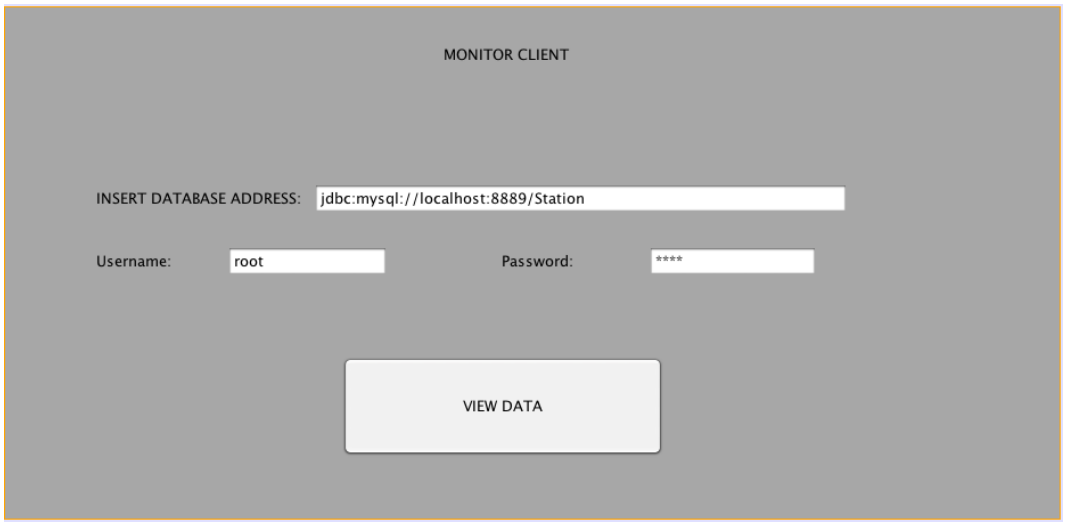

Figura 5.6: ZMonitor GUI

Il funzionamento è il seguente: all'avvio del programma l'utente inserisce l'indirizzo del database su cui sono memorizzati i dati da monitorare, username e password e autorizza la connessione con il database. Da questo momento il programma periodicamente effettua una richiesta al DB tramite una query per richiedere i dati che sono stati inviati da uno o più ZSender.

Con i dati ricevuti dal DB vengono generati dei grafici che mostrano l'andamento delle misurazioni e i livelli rilevati dalla WSN.

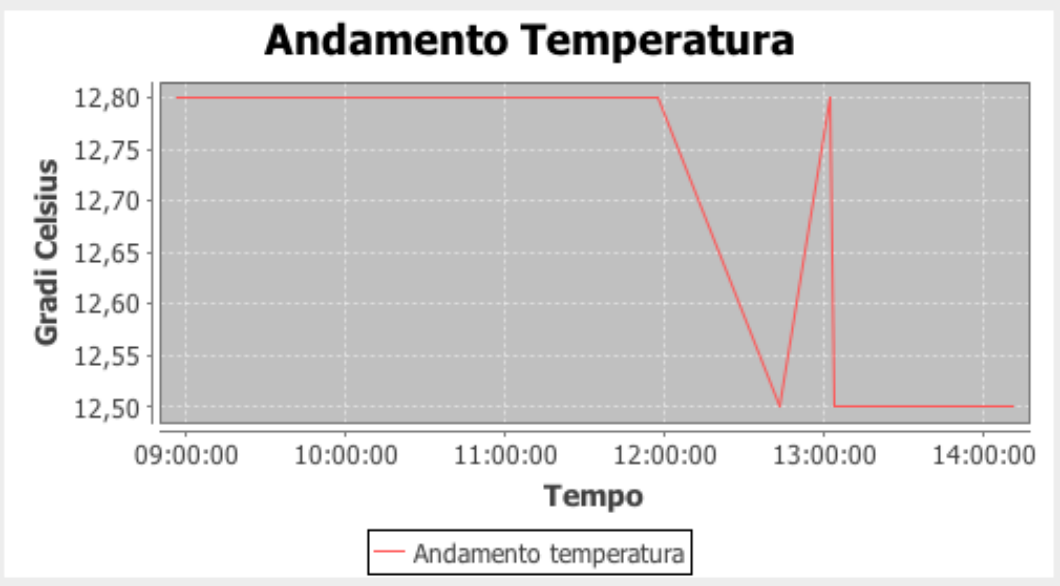

Figura 5.7: Grafico andamento temperatura

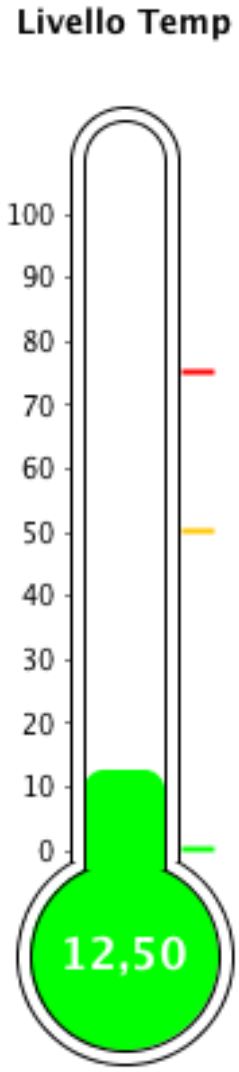

Figura 5.8: Grafico livello temperatura

Conclusioni

Internet of Things è uno scenario prossimo con cui entreremo in contatto nei prossimi 5-10 anni. Potenzialmente si tratta di un fenomeno in grado di modificare profondamente ogni tipo di attività umana e di come le persone sono abituate a interagire con l'ambiente che le circonda. Un ruolo fondamentale all'interno di questa evoluzione della rete e dei comportamenti sociali delle persone sarà svolto da reti in grado di far comunicare svariati tipi di dispositivi tra di loro ed integrarli nella rete Internet per trasmettere e ricevere dati. In questo contesto le tecnologie wireless possono diventare una tecnologia in grado di favorire questi cambiamenti, grazie all'affidabilità raggiunta e soprattutto al risparmio energetico che queste sono i grado di garantire ai dispositivi. Le *Wireless Sensor Network* possono rappresentare la soluzione per integrare gli oggetti che ci circondano quotidianamente con la rete Internet, in modo da avere a disposizione in tempo reale ed in ogni posto del pianeta, informazioni su oggetti che in qualche modo stanno lavorando per noi.

I contesti d'applicazione sono legati soprattutto ai luoghi di utilizzo dove sono più presenti connettività ed energia, come la casa o i luoghi di lavoro.

Diverse tecnologie wireless esistono oggi in ambito WSN e sono attualmente commercializzate sotto diverse etichette, ma soprattutto negli ultimi anni una in particolare, il protocollo Zigbee, si sta affermando come standard grazie all'affidabilità, i pochi costi e i bassi consumi che questo è in grado di offrire. Sempre più imprese stanno investendo nelle WSN e stanno adottando Zigbee come sistema di comunicazione per i loro prodotti.

Affinchè i dati generati dalle WSN possano essere diposnibili in rete e utilizzati per realizzare le più svariate tipologie di applicazioni, dal monitoraggio al controllo degli oggetti, vi è la necessità di integrare questi sistemi con le tecnologie IT.

In questa direzione si è cercato di indirzzare il sistema che è stato realizzato, infatti rappresenta un esempio di come i dati rilevati dalla WSN possono essere inviati tramite le tecnologie IT, sfuttando internet come infrastruttura di comunicazione, ed essere memorizzati in maniera permanente all'interno di datatabase distributi. Una volta resi disponibili i dati, questi possono essere gestiti nei modi più diversi da applicazioni che utlizzando questi dati possono offrire servizi alle persone, come nel caso dell'applicazione di monitoraggio che usa i dati rilevati in tempo reale dalla rete di sensori per mostrare all'utente i livelli delle misure rilevate.

Chiaramente il sistema rappresenta solo la base della comunicazione tra il mondo delle WSN e il mondo di internet, le potenzialità di sviluppo si muovono in diverse direzioni, infatti si potrebbe pensare sia ad una integrazione dei dati con altri tipi di piattaforme, per esempio delle web application che implementino altri tipi di servizi o ad applicazioni per dispositivi *mobile*, sia all' implementazione di nuove funzionalità di controllo dei dispositivi connessi alla WSN attraverso il protocollo Zigbee.

Le possibilità offerte dall'integrazione delle tecnologie WSN e quelle IT possono essere infinite, internet rappresenta il legame con cui questi due mondi possono entrare in contatto e condividere informazioni. Forse passerà ancora qualche tempo prima che tutto questo si realizzi in maniera compiuta e sia assimilata dalla nostra società, ma le potenzialità di queste tecnologie sono chiare fin da oggi e rivelano scenari rivoluzionari che promettono di cambiare il mondo in cui viviamo.

Bibliografia

[1] Il Sole 24 ORE-OGGETTI interconnessi , 12/09/2010.

[2] Gartner Group, 2010.

[3] Sinem Coleri Ergen -September 10, 2004.

[4] ZigBee Wireless Networks andTransceivers,Shahin Farahani, 2008.

[5] Zigbee Wireless Networking , Drew Gislason, 2008

[6] Zigbee Specification, Zigbee Alliance, 2008

[7] www.freescale.com

[8] www.oracle.com

[9] http://dev.mysql.com/doc/

[10] http://www.jfree.org

[11] www.wikipedia.org