

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA

SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA
Corso di Laurea Magistrale in Ingegneria Elettronica e delle Telecomunicazioni

**PROGETTO ED IMPLEMENTAZIONE
DI PROTOCOLLI A BASSO CONSUMO
ENERGETICO PER RETI WIRELESS**

Elaborato in
TEORIA DELL'INFORMAZIONE E CODICI LM

Relatore:
Chiar.mo Prof.Ing.
Marco Chiani

Presentata da:
Luca Delvecchio

Correlatore:
Prof.Ing.
Andrea Giorgetti

Sessione III
Anno Accademico 2012/2013

*Per trovare la giusta via,
bisogna avere il coraggio di perdersi. . .*

Abstract

“I computer del nuovo millennio saranno sempre più invisibili, o meglio embedded, incorporati agli oggetti, ai mobili, anche al nostro corpo. L’intelligenza elettronica sviluppata su silicio diventerà sempre più diffusa e ubiqua¹. Sarà come un’orchestra di oggetti interattivi, non invasivi e dalla presenza discreta, ovunque”. [Mark Weiser, 1991]

La visione dell’ubiquitous computing, prevista da Weiser, è ormai molto vicina alla realtà e anticipa una rivoluzione tecnologica nella quale l’elaborazione di dati ha assunto un ruolo sempre più dominante nella nostra vita quotidiana. La rivoluzione porta non solo a vedere l’elaborazione di dati come un’operazione che si può compiere attraverso un computer desktop, legato quindi ad una postazione fissa, ma soprattutto a considerare l’uso della tecnologia come qualcosa di necessario in ogni occasione, in ogni luogo e la diffusione della miniaturizzazione dei dispositivi elettronici e delle tecnologie di comunicazione wireless ha contribuito notevolmente alla realizzazione di questo scenario.

La possibilità di avere a disposizione nei luoghi più impensabili sistemi elettronici di piccole dimensioni e autoalimentati ha contribuito allo sviluppo di nuove applicazioni, tra le quali troviamo le WSN (Wireless Sensor Network), ovvero reti formate da dispositivi in grado di monitorare qualsiasi grandezza naturale misurabile e inviare i dati verso sistemi in grado di elaborare e immagazzinare le informazioni raccolte.

La novità introdotta dalle reti WSN è rappresentata dalla possibilità di effettuare monitoraggi con continuità delle più diverse grandezze fisiche, il che ha consentito a questa nuova tecnologia l’accesso ad un mercato che

¹onnipresente

prevede una vastità di scenari indefinita. Osservazioni estese sia nello spazio che nel tempo possono essere inoltre utili per poter ricavare informazioni sull'andamento di fenomeni naturali che, se monitorati saltuariamente, non fornirebbero alcuna informazione interessante.

Tra i casi d'interesse più rilevanti si possono evidenziare:

- segnalazione di emergenze (terremoti, inondazioni)
- monitoraggio di parametri difficilmente accessibili all'uomo (frane, ghiacciai)
- smart cities (analisi e controllo di illuminazione pubblica, traffico, inquinamento, contatori gas e luce)
- monitoraggio di parametri utili al miglioramento di attività produttive (agricoltura intelligente, monitoraggio consumi)
- sorveglianza (controllo accessi ad aree riservate, rilevamento della presenza dell'uomo)

Il vantaggio rappresentato da un basso consumo energetico, e di conseguenza un tempo di vita della rete elevato, ha come controparte il non elevato range di copertura wireless, valutato nell'ordine delle decine di metri secondo lo standard IEEE 802.15.4. Il monitoraggio di un'area di grandi dimensioni richiede quindi la disposizione di nodi intermedi aventi le funzioni di un router, il cui compito sarà quello di inoltrare i dati ricevuti verso il coordinatore della rete. Il tempo di vita dei nodi intermedi è di notevole importanza perché, in caso di spegnimento, parte delle informazioni raccolte non raggiungerebbero il coordinatore e quindi non verrebbero immagazzinate e analizzate dall'uomo o dai sistemi di controllo. Lo scopo di questa trattazione è la creazione di un protocollo di comunicazione che preveda meccanismi di routing orientati alla ricerca del massimo tempo di vita della rete.

Nel capitolo 1 vengono introdotte le WSN descrivendo caratteristiche generali, applicazioni, struttura della rete e architettura hardware richiesta.

Nel capitolo 2 viene illustrato l'ambiente di sviluppo del progetto, analizzando le piattaforme hardware, firmware e software sulle quali ci appoggeremo per realizzare il progetto. Verranno descritti anche alcuni strumenti utili per effettuare la programmazione e il debug della rete.

Nel capitolo 3 si descrivono i requisiti di progetto e si realizza una mappatura dell'architettura finale.

Nel capitolo 4 si sviluppa il protocollo di routing, analizzando i consumi e motivando le scelte progettuali.

Nel capitolo 5 vengono presentate le interfacce grafiche utilizzate utili per l'analisi dei dati.

Nel capitolo 6 vengono esposti i risultati sperimentali dell'implementazione fissando come obiettivo il massimo lifetime della rete.

Indice

| | | |
|----------|---|-----------|
| 1 | Wireless Sensor Networks | 1 |
| 1.1 | Hardware | 2 |
| 1.2 | Standard IEEE 802.15.4 | 4 |
| 1.2.1 | Livelli | 5 |
| 1.2.2 | Obiettivi | 10 |
| 1.2.3 | Topologie di rete | 12 |
| 2 | Ambiente di sviluppo | 15 |
| 2.1 | Piattaforma Hardware | 15 |
| 2.1.1 | Il sistema The Guardian | 16 |
| 2.1.2 | Convertitore RS232 ↔ TCP/IP | 18 |
| 2.1.3 | Convertitore UART ↔ RS232 | 18 |
| 2.2 | Piattaforma Firmware | 20 |
| 2.3 | Piattaforma Software | 21 |
| 2.4 | Strumenti di debugging | 21 |
| 3 | Progetto | 23 |
| 3.1 | Requisiti | 23 |
| 3.1.1 | Requisiti funzionali | 23 |
| 3.1.2 | Requisiti di sistema | 24 |
| 3.1.3 | Requisiti non funzionali | 24 |
| 3.2 | Analisi dei problemi | 27 |
| 3.2.1 | Analisi dei consumi | 27 |
| 3.2.2 | Analisi della potenza di trasmissione | 30 |
| 3.2.3 | Analisi dei protocolli di routing | 33 |

| | | |
|----------|--|-----------|
| 3.3 | Architettura logica | 38 |
| 4 | Implementazione Firmware | 41 |
| 4.1 | Gestione associazioni | 41 |
| 4.2 | Gestione inserimento/esclusione nodi | 42 |
| 4.3 | Gestione raccolta dati e sincronizzazione nodi | 44 |
| 4.4 | Protocollo di routing | 46 |
| 4.4.1 | Protocollo Simil-MCSP | 47 |
| 4.4.2 | Protocollo Simil-LEACH | 51 |
| 4.4.3 | Protocollo di Flooding | 52 |
| 4.5 | Invio dati al server | 53 |
| 4.6 | Gestione rete da remoto | 54 |
| 4.7 | Configurazione nodi rete | 54 |
| 5 | Implementazione Server e Software Grafico | 57 |
| 5.1 | Server TCP/IP | 58 |
| 5.2 | Terminale di controllo della rete | 58 |
| 5.3 | Interfaccia grafica | 59 |
| 6 | Risultati sperimentali | 61 |
| 6.1 | Risultati attesi | 61 |
| 6.1.1 | Scenario A | 63 |
| 6.1.2 | Scenario B | 65 |
| 6.1.3 | Lifetime stimato | 67 |
| 6.2 | Risultati ottenuti | 69 |
| 6.2.1 | Scenario A | 72 |
| 6.2.2 | Scenario B | 75 |
| 6.2.3 | Lifetime stimato | 78 |
| 7 | Conclusioni | 79 |
| A | Configurazione TCP Converter | 81 |
| B | Batteria al piombo | 83 |

Acronimi

| | |
|--------------|--|
| AC | Accumuled Cost |
| AODV | Ad Hoc On-Demand Distance Vector |
| FFD | Fully Function Device |
| ISM | Industrial, Scientific and Medical |
| LEACH | Low Energy Adaptive Clustering Hierarchy |
| MCSP | Minimum Cost Survivor Path |
| OSAL | Operative System Abstraction Layer |
| PAN | Personal Area Network |
| RFD | Reduced Function Device |
| WSN | Wireless Sensor Network |

Elenco delle figure

| | | |
|-----|--|----|
| 1.1 | Esempio di rete WSN | 2 |
| 1.2 | Diagramma a blocchi di un nodo per reti WSN | 4 |
| 1.3 | Protocolli standard wireless | 5 |
| 1.4 | Modello di riferimento ISO/OSI | 6 |
| 1.5 | Pannello fotovoltaico | 11 |
| 1.6 | Topologie di rete 802.15.4 | 12 |
| 2.1 | Diagramma a blocchi del sistema The Guardian | 16 |
| 2.2 | Dispositivi Texas Instruments | 17 |
| 2.3 | The Guardian | 18 |
| 2.4 | Convertitore RS232 - TCP/IP | 19 |
| 2.5 | Convertitore RS232 - UART | 19 |
| 2.6 | IAR Embedded Workbench | 20 |
| 2.7 | SmartRF05EB | 21 |
| 2.8 | CC2531 | 22 |
| 2.9 | Esempio di interfaccia utente del Packet Sniffer | 22 |
| 3.1 | Misura di corrente attraverso il multimetro | 28 |
| 3.2 | Misura di corrente con resistenza di Shunt, schema | 28 |
| 3.3 | Misura di corrente con resistenza di Shunt, oscilloscopio | 29 |
| 3.4 | Impostazione potenza di trasmissione | 30 |
| 3.5 | Analisi consumi, Output Power = -18 dBm | 31 |
| 3.6 | Analisi consumi, Output Power = 4,5 dBm | 32 |
| 3.7 | Riduzione del guadagno in ricezione | 33 |
| 3.8 | Scenari di test MCSP [1] | 36 |
| 3.9 | Prestazioni MCSP, % nodi presenti VS cicli di esecuzione [1] | 36 |

| | | |
|------|---|----|
| 3.10 | Architettura del progetto | 38 |
| 3.11 | Scenario A di test | 40 |
| 3.12 | Scenario B di test | 40 |
| 4.1 | Trama TDM | 44 |
| 4.2 | Schema della comunicazione nella rete | 45 |
| 4.3 | Modalità di associazione | 49 |
| 4.4 | Configurazione nodi | 55 |
| 5.1 | Struttura multi-thread del software | 57 |
| 5.2 | Interfaccia grafica | 59 |
| 5.3 | Terminale di comando, avvio | 60 |
| 5.4 | Terminale di comando, ricezione dati | 60 |
| 6.1 | Scenario A | 63 |
| 6.2 | Scenario B | 65 |
| 6.3 | Grafico di dispersione dei valori di tensione rilevati in una settimana | 70 |
| 6.4 | Grafico dei valori di tensione previsti per i prossimi 300 giorni | 71 |
| 6.5 | Disposizione nodi, Peryton, Scenario A | 72 |
| 6.6 | Grafico Simil-MCSP, scenario A | 73 |
| 6.7 | Grafico Simil-LEACH, scenario A | 73 |
| 6.8 | Grafico Flooding | 74 |
| 6.9 | Disposizione nodi, Peryton, Scenario B | 75 |
| 6.10 | Grafico Simil-MCSP, scenario B | 76 |
| 6.11 | Grafico Simil-LEACH, scenario B | 77 |
| 6.12 | Grafico Flooding | 77 |
| 6.13 | Grafico di confronto tra i vari protocolli | 78 |
| B.1 | Tensione di fine scarica | 84 |
| B.2 | Tempo di scarica in funzione della corrente di scarica | 84 |

Elenco delle tabelle

| | | |
|------|--|----|
| 2.1 | Livelli di tensione UART - RS232 | 19 |
| 3.1 | Corrente assorbita The Guardian | 30 |
| 4.1 | Messaggio inviato dal sink nel SuperSlot | 46 |
| 4.2 | Messaggio inviato dal nodo nel SuperSlot | 46 |
| 6.1 | Consumo medio per ciclo, Protocollo Simil-MCSP | 62 |
| 6.2 | Consumo medio per ciclo, Protocollo Simil-LEACH | 62 |
| 6.3 | Consumo medio per ciclo, Protocollo Flooding | 62 |
| 6.4 | Consumo medio per periodo, Scenario A, Simil-MCSP | 63 |
| 6.5 | Consumo medio per periodo, Scenario A, Simil-LEACH | 64 |
| 6.6 | Consumo medio per periodo, Scenario A, Flooding | 64 |
| 6.7 | Consumo medio per periodo, Scenario B, Simil-MCSP | 66 |
| 6.8 | Consumo medio per periodo, Scenario B, Simil-LEACH | 66 |
| 6.9 | Consumo medio per periodo, Scenario B, Flooding | 67 |
| 6.10 | Lifetime stimato per un nodo RFD, MCSP | 67 |
| 6.11 | Lifetime stimato per un nodo RFD, LEACH | 68 |
| 6.12 | Lifetime stimato per un nodo FFD in ambiente con 1 nodo FFD | 68 |
| 6.13 | Lifetime stimato per un nodo FFD in ambiente con 2 nodi FFD | 68 |
| 6.14 | Lifetime stimato per un nodo FFD in ambiente con 4 nodi FFD | 69 |
| 6.15 | Regressione lineare per la stima dei coefficienti della retta di previsione | 70 |
| 6.16 | Dati rilevati manualmente, Simil-MCSP, scenario A | 72 |
| 6.17 | Dati rilevati manualmente, Simil-MCSP, scenario B | 75 |

Capitolo 1

Wireless Sensor Networks

Una Wireless Sensor Network nella sua più semplice forma può essere definita come una rete di dispositivi che possono acquisire informazioni dall'ambiente e possono comunicarle tramite collegamenti wireless. I dati vengono indirizzati, attraverso connessioni multi-hop, ad un sink (o coordinatore) che solitamente è collegato ad altre reti, ad esempio Internet, attraverso un gateway. I nodi possono essere fermi o in movimento, possono conoscere o meno la loro posizione, possono essere omogenei tra loro oppure di diverso tipo [1].

Lo sviluppo di soluzioni basate sulle Wireless Sensor Network è molto recente ed è considerato come uno scenario multidisciplinare poiché richiede conoscenze di sensori e signal processing, radio, networking, programmazione di microcontrollori, algoritmi per il risparmio energetico, gestione di database e di programmazione di server per la raccolta dei dati.

In questo capitolo viene descritto l'hardware necessario per realizzare un dispositivo in grado di far parte di una rete WSN, analizzandone i vantaggi e i limiti, e viene definito lo standard protocollare 802.15.4 con alcuni accenni alle caratteristiche dei vari livelli ISO/OSI ed alle topologie di rete disponibili.

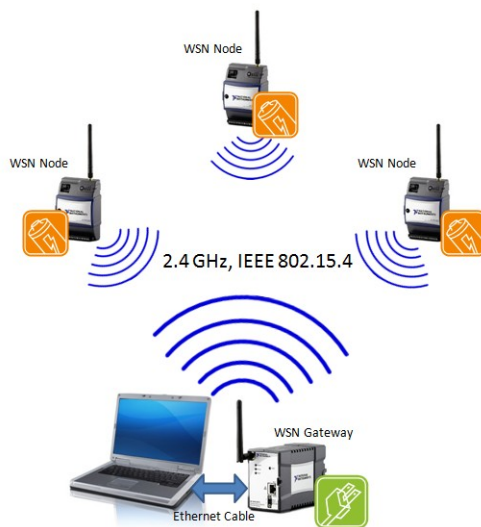


Figura 1.1: Esempio di rete WSN

1.1 Hardware

Un nodo appartenente ad una rete WSN deve essere un dispositivo di piccole dimensioni, a basso costo, a basso consumo che consente la lettura di sensori che non richiedono una data rate elevata. Ad esempio, la cattura video con un sensore ottico CMOS di risoluzione elevata non è realizzabile con una rete WSN.

I componenti necessari per la realizzazione di un nodo WSN sono un microcontrollore, un dispositivo di trasmissione wireless basata secondo lo standard 802.15.4, una serie di sensori opportunamente interfacciati attraverso i relativi circuito di condizionamento, possibili attuatori e l'alimentazione.

- Unità computazionale: la gestione del nodo è affidata ad un componente in grado di gestire il flusso delle operazioni di ogni nodo, come ad esempio la lettura dei dati attraverso i sensori collegati, l'invio/ricezione di dati, la gestione delle operazioni di risparmio energetico e il calcolo degli algoritmi di routing. I componenti ideali sono i microcontrollori, ovvero dispositivi a singolo chip per sistemi embedded, che a differenza dei microprocessori, i quali integrano sul chip solo la logica di elaborazione, sono dei sistemi completi e integrano

memorie, pin di I/O, timer, interrupt controller, prevedono modalità di risparmio energetico e l'interfacciamento con sensori esterni.

- Unità di memoria: è necessaria una memoria di tipo EEPROM o flash per la memorizzazione del codice ed è altrettanto necessaria una memoria RAM per le variabili temporanee. Solitamente sono integrate nel microcontrollore.
- Unità di comunicazione: è costituito da un transceiver ad onde radio che possa trasmettere a 868 MHz, 915 MHz o 2.4 GHz in base allo standard IEEE 802.15.4.
- Unità di sensing: i sensori possono essere di tre tipi: passivi omnidirezionali, passivi direzionali e attivi. I sensori passivi sono quelli nei quali la misurazione del parametro in esame avviene tramite la misurazione di una grandezza elettrica passiva (resistenza, capacità) alimentata da sorgenti esterne di energia. Misurando una grandezza passiva il sensore non modifica in nessun modo l'oggetto della misurazione.

Si ha omnidirezionalità se la grandezza misurata non varia a seconda della direzione in cui la si misura; ad esempio, sensori di temperatura ed umidità sono omnidirezionali. Sono direzionali invece i sensori di immagine, come le fotocamere, che restituiscono risultati differenti in base al loro posizionamento [2].

I sensori attivi forniscono in uscita un segnale elettrico attivo (tensione, corrente) ottenuto mediante trasformazione di energia (meccanica, termica, luminosa, ecc.) in forma elettrica.

- Unità di alimentazione: solitamente si utilizza alimentazione a batterie, di tipo diverso e di capacità diversa a seconda dell'hardware e dell'applicazione da realizzare. Uno degli obiettivi delle WSN è la realizzazione di reti aventi tempi di vita molto lunghi e, data la durata limitata delle batterie, solitamente si propende all'integrazione delle stesse con fonti energetiche ausiliare, come l'energia solare attraverso le celle fotovoltaiche, le vibrazioni meccaniche attraverso elementi

piezoelettrici/ magnetici o il flusso di aria tramite minuscole turbine basate su MEMS.

Riassumendo le reti WSN presentano i pregi di essere realizzate con dispositivi piccoli, a basso costo, portabili ed autoalimentati, ragion per cui la loro installazione è possibile praticamente in ogni angolo del pianeta.

La necessità di mantenere i consumi ridotti, allo scopo di avere elevati tempi di vita, comporta tuttavia una potenza di trasmissione limitata e di conseguenza un range ridotto di copertura wireless. Infine la limitata capacità di memorizzazione dati non consente di lavorare con grandi quantità di informazioni.

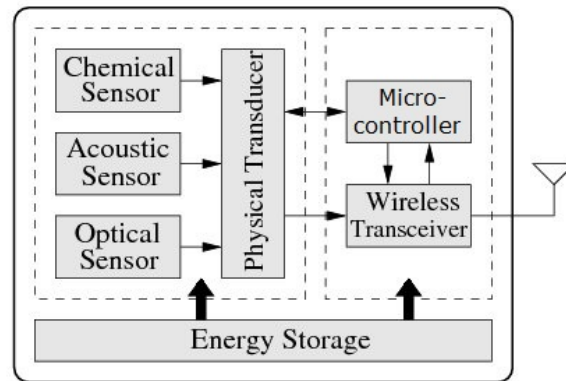


Figura 1.2: Diagramma a blocchi di un nodo per reti WSN

1.2 Standard IEEE 802.15.4

I requisiti di connettività di una rete di sensori wireless potrebbero essere soddisfatti anche utilizzando protocolli noti e divenuti uno standard commerciale, quali 802.11 (Wi-Fi) e 802.15.1 (Bluetooth). Esaminando però nel dettaglio gli obiettivi di una Wireless Sensor Network, dovremo eliminare la possibilità di utilizzo di entrambi gli standard rispettivamente a causa degli elevati consumi energetici per il Wi-Fi e del range ridotto (10m) di copertura per il Bluetooth.

Nel corso dell'anno 2003 è stato definito il protocollo standard IEEE 802.15.4 per la frequenza 868 MHz con una data rate massima di 20 kbps.

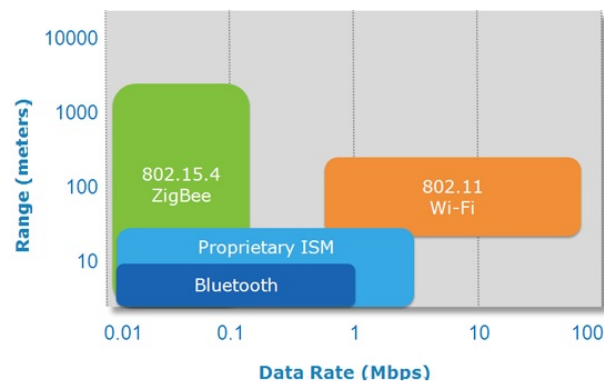


Figura 1.3: Protocolli standard wireless

Negli anni successivi, fino al 2006, il protocollo è stato sviluppato aggiungendo anche le frequenze di trasmissione 915 MHz e 2.4 GHz e incrementando la data rate fino a 250 kbps. Le bande di lavoro citate sono bande ISM quindi ad accesso libero. Il range di trasmissione definito nel protocollo è variabile tra i 10 e i 75 metri in base all'hardware in analisi.

1.2.1 Livelli

Attraverso il modello OSI (Open Systems Interconnection) possiamo caratterizzare l'architettura del protocollo 802.15.4. La divisione dei livelli non è definita e limpida come in altre architetture, in quanto per scelta progettuale si decide di consentire la sovrapposizione tra di essi, in particolare tra livello 2 e livello 3. Questo perchè nonostante la divisione in livelli consenta di ottenere ottime prestazioni in ognuno di essi, grazie all'approccio cross-layer si riescono a massimizzare le prestazioni della rete riducendo i consumi energetici. Alcuni dei documenti qui proposti rappresentano protocolli di comunicazione che sfruttano le interazioni tra livelli:

- livello MAC e livello fisico [3],
- livello rete e livello fisico [4],
- livello di trasporto e livello fisico [5].

Nei vari esperimenti proposti in questa trattazione elaboreremo un protocollo cross-layer che si basa sull'integrazione tra livello MAC e livello di

trasporto creando percorsi di routing basati sul numero di hop, sulla carica della batteria e sulla qualità del link senza alcuno scambio di informazione a livello applicativo.

I livelli fondamentali per una rete WSN sono il secondo e il terzo. I layer rimanenti assumono un'importanza minore. I livelli di sessione, presentazione e applicazione spesso e volentieri non sono nemmeno definiti.



Figura 1.4: Modello di riferimento ISO/OSI

Livello 1: fisico

Il livello fisico (PHY) fornisce servizi come l'attivazione e disattivazione dei moduli radio, energy detection (ED), link quality indication (LQI), selezione del canale, clear channel assessment (CCA), ovvero rilevazione dell'energia del canale per determinare se è già in corso una trasmissione, e ovviamente l'invio e la ricezione dei pacchetti sul mezzo di trasporto fisico. La trasmissione si basa sulla modulazione di codice DSSS (Direct Sequence Spread Sprecctum) nelle bande:

- 868 MHz: 20Kbps - 1 (2003) / 100Kbps - 3 (2006)
- 915 MHz: 40Kbps - 10 (2003) / 250Kbps - 30 (2006)
- 2.45 GHz: 250Kbps - 16 canali

Livello 2: collegamento

Il livello Data Link è composto da due sottolivelli: LLC e MAC. Il primo, detto Logical Link Control layer, è il collegamento con i livelli superiori e il livello fisico e consente l'interoperabilità tra diversi tipi di rete. Il suo uso è molto limitato, dato che anche altri protocolli del livello 3 si occupano della cooperazione con altri sistemi.

Il livello MAC (Medium Access Control) riveste invece un ruolo molto importante e si occupa di tre funzioni fondamentali:

- l'assemblaggio dei dati in frame tramite l'aggiunta di un header, contenente informazioni sull'indirizzo, e un trailer, contenente informazioni per la correzione degli errori
- la scomposizione dei frame ricevuti per estrarre informazioni sull'indirizzo e la correzione degli errori
- la gestione dell'accesso al mezzo di trasmissione condiviso

L'obiettivo principale per la realizzazione di un buon livello MAC per reti WSN è il risparmio energetico. Ci sono diversi fattori che contribuiscono allo spreco energetico, come l'eccessivo overhead, l'idle listening, le collisioni di pacchetti e l'overhearing, ovvero la ricezione e la decodifica di pacchetti destinati ad altri nodi. La ritrasmissione di pacchetti persi a causa di collisioni è un'ennesima fonte di spreco, nonché di degrado delle prestazioni. L'obiettivo di un MAC è quello di impedire le situazioni appena descritte. Ci sono tre gruppi principali di protocolli MAC:

- Scheduled-based
- Contention-based
- ibridi

I primi regolano l'accesso al canale tramite una schedulazione, preassegnata ad ogni nodo, affinché un solo nodo alla volta possa occupare il mezzo trasmissivo. I protocolli Contention-based non effettuano alcuna preallocazione delle risorse e l'accesso al canale condiviso è garantito on-demand. Se due nodi tentano l'accesso contemporaneo, si verifica una collisione. Il compito

di questo tipo di MAC è quello di minimizzare, e non di evitare completamente, la manifestazione di queste collisioni. I protocolli ibridi uniscono i vantaggi delle due tecniche.

I protocolli MAC presentano meccanismi differenti per minimizzare la probabilità di collisione, l'overhearing e l'overhead per il controllo di accesso al canale al fine di ridurre i consumi. Il problema delle collisioni viene risolto abitualmente ricorrendo ad algoritmi che consentano l'accesso esclusivo al canale tra i vari contendenti. Il metodo classico per evitare l'overhearing è la disattivazione del modulo radio quando non è necessario [2].

Livello 3: rete

Il livello di rete si occupa principalmente del routing. Il routing ha un ruolo importantissimo, dato che lo scopo di una WSN è quello di convogliare tutti i dati presenti rilevati da ogni sensore presente nella rete in un unico punto, il sink, dove risiede un sistema in grado di visualizzare i dati ricevuti o di inoltrarli verso reti di altro tipo come ad esempio Internet. Non sempre è possibile raggiungere il sink con una singola trasmissione perchè il range di copertura di questa tecnologia è nell'ordine delle decine di metri, mentre alcune applicazioni possono prevedere il monitoraggio di aree molto più vaste.

Per raggiungere il sink con il minor costo energetico è necessario che i protocolli di routing individuino i cammini minimi tra i nodi e la destinazione finale. Oltre alla ricerca del cammino a numero minimo di hop i protocolli di routing potrebbero tenere in considerazione anche la carica della batteria dei nodi a cui si vuole inviare un dato o la qualità del link trasmissivo. Se ad esempio un nodo è particolarmente scarico o il canale non è ottimale per la trasmissione si può scegliere di percorrere una strada alternativa.

La dinamicità della rete, la densità dei nodi, le risorse hardware limitate e la necessità di mantenere limitati i consumi, rendono complicato lo sviluppo di protocolli di routing. Uno degli aspetti negativi in particolare è la necessità di mantenere la latenza di trasmissione limitata anche per l'invio in multi-hop dei pacchetti provenienti da i nodi più lontani.

Livello 4: trasporto

La natura data-centric delle WSN combinata con le risorse hardware limitate rende il protocollo TCP inutilizzabile in questo ambito, perchè la gestione del traffico all'interno della rete può dipendere dal tipo di dati trasportati. Ad esempio per i pacchetti provenienti dai nodi e diretti al sink, che seguono quindi il forward path, si può applicare un principio di eventreliability, ovvero non ha importanza che tutti i pacchetti arrivino a destinazione, ma serve che arrivino almeno quelli necessari per un corretto monitoraggio dell'evento. Al contrario, nel percorso inverso (reverse path), è richiesto che tutti i messaggi spediti dal sink giungano ai nodi destinatari. Questo perchè i messaggi mandati dal sink contengono informazioni di controllo dell'attività del nodo, come richieste dei dati o istruzioni di programmazione.

Livelli 5,6,7: sessione, presentazione, applicazione

Protocolli di questo tipo per una generica WSN non sono ancora diffusi, sebbene in passato ne siano stati proposti alcuni, tra cui Sensor Management Protocol (SMP), Task Assignment e Data Advertisement Protocol (TADAP) e Sensor Query Data Dissemination (SQDDP).

Il motivo della mancata diffusione è la diversità dei requisiti che ogni installazione richiede. Un esempio classico di applicazione richiede il monitoraggio periodico di tutti i sensori, un altro potrebbe prevedere il monitoraggio solo di una parte di sensori, altri ancora potrebbero richiedere periodi di campionamento diversi. La vastità di possibili implementazioni rendono quindi non facile la realizzazione di uno standard.

SMP fornisce operazioni software per compiti di amministrazione della rete, come data aggregation, clustering, sincronizzazione, localizzazione, accensione sensori, monitoraggio e riconfigurazione dei nodi, autenticazione e distribuzione di chiavi nelle comunicazioni sicure.

TADAP gestisce l'assegnazione di "interessi" riguardo un attributo o un evento e la pubblicizzazione dei dati disponibili.

SQDDP fornisce all'utente le interfacce per avviare e rispondere a query e si occupa di assegnare identificativi ai nodi in base ad attributi o alla loro locazione.

Il discorso è diverso per le reti ZigBee. Infatti questo standard definisce anche il livello applicativo, che è composto dall'Application Support Sublayer (APS), dal Zigbee Device Object (ZDO) e da oggetti applicativi definiti dal produttore [2].

1.2.2 Obiettivi

È importante considerare che non è possibile creare un prototipo di WSN in grado di gestire tutte le applicazioni realizzabili; sarà però possibile analizzare alcuni parametri comuni a tutte le reti.

- Quality of Service (QoS): i parametri QoS adottati tradizionalmente come ritardo, banda minima e jitter, sono trascurabili perché i pacchetti trasmessi sono di ridotte dimensioni e, grazie ai meccanismi di analisi del canale, è possibile accedere al medesimo in maniera esclusiva.
- Tolleranza ai guasti: un nodo potrebbe esaurire la batteria, subire danni fisici o perdere il collegamento wireless con gli altri nodi. Tale guasto non deve però compromettere il comportamento del resto della rete che deve continuare a monitorare i nodi rimanenti.
- Lifetime: i nodi sono alimentati da batterie che non è possibile o non è conveniente sostituire. L'obiettivo di una WSN è quindi quello di rimanere attiva più a lungo possibile. Per questo motivo il risparmio energetico assume un ruolo fondamentale. L'applicazione di fonti energetiche per la ricarica dei nodi, come le celle fotovoltaiche (vedi figura 1.5) possono allungare il tempo di vita della rete, ma un buon algoritmo che preveda l'ottimizzazione del risparmio energetico è fondamentale in tutte le reti.

La definizione di lifetime non è univoca, ma dipende dall'applicazione in esame: a volte si indica con lifetime il tempo entro il quale il primo nodo della rete esaurisce la propria energia, a volte corrisponde all'istante in cui il 50% dei nodi vengono persi, altre volte indica il momento in cui un'area particolare della rete non è più monitorata da alcun nodo. Riassumendo, il lifetime dipende dall'obiettivo dell'applicazione realizzata.



Figura 1.5: Pannello fotovoltaico

- Scalabilità: una WSN può comprendere un numero elevato di nodi e la sua architettura deve essere in grado di controllare una rete che assicuri la connessione per tutti i nodi.
- Dinamicità: nuovi nodi possono entrare e altri possono uscire dalla rete in qualsiasi momento e i link tra nodi potrebbero cadere o degradarsi in seguito a variazioni ambientali o allo spostamento dei nodi stessi. Le reti devono potersi autoconfigurare per permettere la connessione di tutti i nodi presenti.
- Programmabilità: i nodi devono essere in grado di poter cambiare i propri compiti in qualunque momento, ovvero devono poter essere riprogrammabili.
- Risorse limitate: Le risorse hardware sono limitate in un nodo di piccole dimensioni e a basso consumo. Gli algoritmi devono avere una ridotta occupazione di memoria (footprint), ovvero devono avere una complessità tale da poter essere eseguibili da CPU poco potenti.

1.2.3 Topologie di rete

All'interno di una rete 802.15.4 distinguiamo 3 tipi di dispositivi:

- PAN Coordinator (sink, gateway)
- FFD (Fully Function Device, router)
- RFD (Reduced Function Device, end node)

Il PAN Coordinator è il nodo a cui devono giungere tutti i dati e che deve potersi interfacciare con il mondo esterno. I nodi router possono interagire con qualsiasi altro nodo in visibilità radio, a prescindere dalla modalità di funzionamento di quest'ultimo. Gli RFD invece sono stati concepiti come nodi a ridotta complessità e pertanto essi possono comunicare soltanto con gli FFD.

Le più comuni topologie di rete utilizzate nelle reti di sensori wireless sono:

- Stella: un nodo, detto centro-stella, funge da coordinatore della rete. Un qualsiasi nodo per comunicare con un altro nodo invia il messaggio al coordinatore, il quale lo inoltra al destinatario. Tutti i messaggi quindi transitano attraverso il centro stella, e un messaggio compie al massimo due hop. È la topologia più semplice, in quanto le informazioni di routing sono statiche, e minimali, e ogni nodo comunica solamente con il coordinatore. La semplicità di questa topologia va a discapito dell'impossibilità di realizzare reti che sfruttino il multi-hop. La dimensione della rete è legata quindi al range di copertura radio

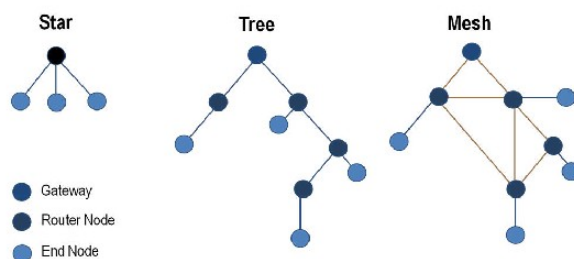


Figura 1.6: Topologie di rete 802.15.4

dei dispositivi. Un altro svantaggio risiede nel fatto che la rete non è robusta, nel momento in cui il coordinatore non svolge completamente i suoi compiti.

- Mesh: Nelle reti mesh o peer to peer, ogni nodo può potenzialmente comunicare con gli altri presenti nel raggio di copertura radio.

Si parla di topologia “full mesh” se ogni nodo può comunicare con tutti gli altri, di “partial mesh” in caso contrario.

Questa topologia viene definita di tipo multi-hop, in quanto consente l’arrivo a destinazione di un messaggio attraverso l’inoltro dello stesso da parte di nodi intermedi. In questo modo è possibile realizzare reti di dimensioni arbitrariamente grandi. Rispetto alla topologia a stella si ha il vantaggio di avere percorsi ridondanti, che consentono l’aumento dell’affidabilità e della robustezza, richiedendo però l’utilizzo dei algoritmi più complessi.

- Albero - Nella topologia ad albero i nodi si diramano dal PAN Coordinator con una struttura nella quale si ha un padre e un numero definito di figli, i quali potranno svolgere la funzione di padre per i nodi posti al livello inferiore.

L’invio di un messaggio da parte di un nodo prevede la risalita dell’albero fino alla radice, ovvero il sink, che svolge la funzione di punto di raccolta dei dati e coordinatore della rete. Dato il tipo di struttura i nodi hanno un carico di lavoro che aumenta al diminuire della loro profondità.

Il vantaggio di questa topologia, rispetto a quella mesh, è la creazione di percorsi a cammino minimo che consentono lo sviluppo di sistemi di gestione meno complessi. Rispetto alla topologia a stella invece il vantaggio è dato dalla possibilità di creazione di reti arbitrariamente grandi, utilizzando il multi-hop.

Capitolo 2

Ambiente di sviluppo

In questo capitolo si analizzeranno tutte le periferiche e le piattaforme utilizzate per la stesura dell'elaborato.

Per la parte hardware si descriverà il dispositivo utilizzato come nodo all'interno della rete e i componenti che hanno consentito l'esportazione dei dati raccolti in un PC Server.

La descrizione della piattaforma firmware concerne lo stack protocollare TIMAC, messo a disposizione dalla Texas Instruments come base della programmazione dei nodi e il software utilizzato per la programmazione del firmware IAR Embedded Workbench.

L'ambiente di sviluppo del software che permette la visualizzazione grafica dei dati è descritto come ultimo elemento. A causa della compatibilità con i diversi sistemi operativi, della semplicità di utilizzo e della licenza open-source la scelta del linguaggio di programmazione è ricaduta su Java e quella dell'IDE su Eclipse.

2.1 Piattaforma Hardware

Le topologie di rete implementabili nel protocollo 802.15.4 sono 3: stella, mesh o albero. In questo elaborato si utilizzerà una topologia di tipo ad albero, che, rispetto alle reti a stella, presenta il vantaggio di realizzare trasmissioni multihop, mentre rispetto alle reti mesh vanta la possibilità di creare percorsi a cammino minimo con una complessità inferiore..

Individuata la topologia, avremo la necessità di identificare 3 tipi di dispositivi per realizzare le funzioni di PAN Coordinator, Fully Function Device (FFD) e Reduced Function Device (RFD).

Il PAN coordinator avrà il compito di gestire la rete e di inoltrare i dati ricevuti verso un PC con connessione TCP/IP. Un interfacciamento tra Ethernet-TCP/IP e WSN è necessario, perciò si prevede l'integrazione di componenti che possano consentire la comunicazione.

Il dispositivo FFD, avendo compiti di inoltre oltre a quelli semplici di sensing, sarà soggetto a consumi maggiori rispetto ai dispositivi RFD quindi potrebbe richiedere una batteria di capacità più elevata.

La composizione hardware lato WSN è la medesima per tutti i nodi e si basa sul dispositivo *The Guardian*, illustrato in seguito.

2.1.1 Il sistema The Guardian

Il dispositivo utilizzato per la realizzazione dell'elaborato è il *The Guardian* della società I.Co. Innovative Company. Il diagramma a blocchi in figura 2.1 mostra l'architettura del nodo.

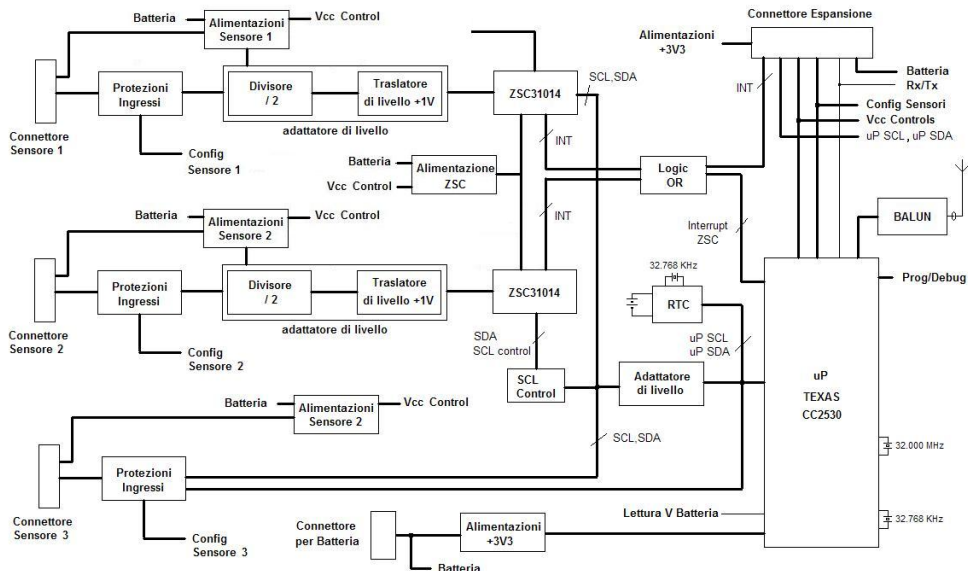


Figura 2.1: Diagramma a blocchi del sistema The Guardian

- System-on-a-chip CC2530F256: Si basa sulla famiglia CC2530 di Texas Instruments, che unisce un microcontrollore 8051 a un trasceiver compatibile con lo standard 802.15.4. In particolare il componente impiegato è il CC2530F256, dotato di una memoria Flash di 256 KB. Altre funzionalità di questo dispositivo necessarie alla realizzazione dell'applicazione finale sono le porte di I/O e il moduli di trasmissione UART e I2C.







| | System-on-chip Small footprint High integration Low cost | Co-processor Flexible Easy to use Reduced time to market | Dual-chip Ultra low-power or High performance |
|------------------------------------|--|--|---|
| Complete ZigBee® Solution | Application CC2530  | Any Processor (e.g. MSP430 or Stellaris ARM)  | MSP430 or Stellaris™ ARM  |
| | Protocol stack  | CC2530ZNP ZigBee co-processor with embedded stack and uart/spi/usb interface  | |
| | Radio | | CC2520  |
| RF front end (optional) | CC2590 / CC2591 | CC2590 / CC2591 | CC2590 / CC2591 |

Figura 2.2: Dispositivi Texas Instruments

- Serial real-time clock (RTC): Ha il compito di memorizzare l'ora indicata e aggiornarla. È un dispositivo a margine di errore e consumi minimi e comunica in I²C con il microcontrollore.
- Convertitore A/D di precisione: Viene utilizzato in sostituzione dei convertitori A/D interni al microcontrollore a causa della risoluzione più elevata e perché presenta intrinsecamente meccanismi di compensazione della temperatura. Comunica in I²C con il microcontrollore.
- Regolatori di tensione: Necessari alla corretta alimentazione e protezione da sovralimentazioni dei componenti della scheda a 3V3 o 5V.
- Sensori: Alle due porte d'ingresso possono essere collegati sensori di ogni tipo. La scheda è in grado di alimentare i sensori fino a una tensione massima di 5V.

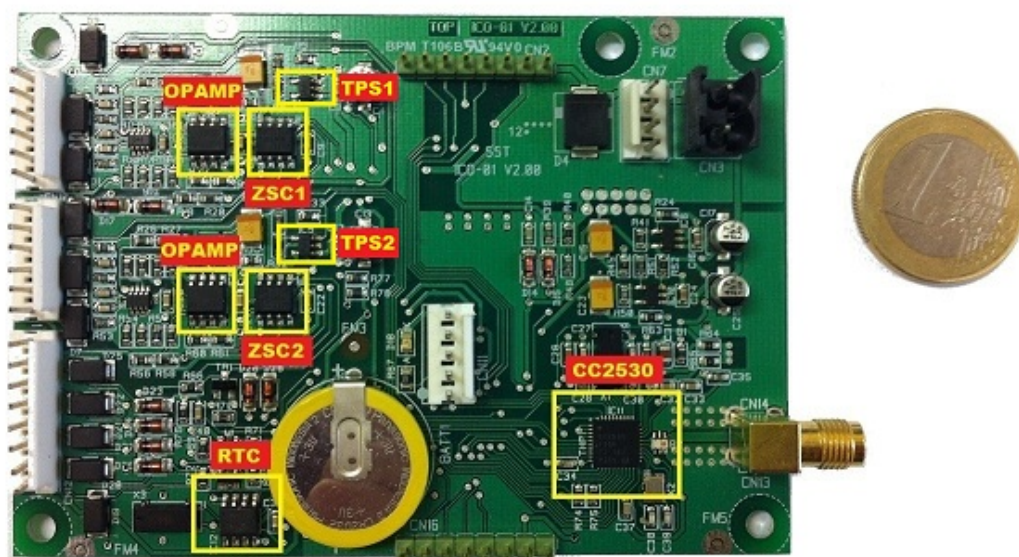


Figura 2.3: The Guardian

2.1.2 Convertitore RS232 ↔ TCP/IP

Il convertitore individuato è il Grid Connect GC-ATC-1000 che esegue la conversione RS232 ↔ TCP/IP. Il dispositivo dovrà attivare una connessione con un server e scambiare dati, da un lato mediante il protocollo TCP/IP, dall'altro attraverso il protocollo RS232. Il microcontrollore CC2530F256 non è interfacciabile direttamente con il protocollo seriale RS232, ragion per cui adotteremo un ulteriore convertitore.

Nell'appendice A verrà descritta la configurazione del convertitore.

2.1.3 Convertitore UART ↔ RS232

Il convertitore è un semplice traslatore di livello realizzato su una striboard inserendo un dispositivo MAX3232 opportunamente configurato. Consente l'interfacciamento del convertitore RS232-TCP/IP al dispositivo di comunicazione seriale UART del microcontrollore.



Figura 2.4: Convertitore RS232 - TCP/IP

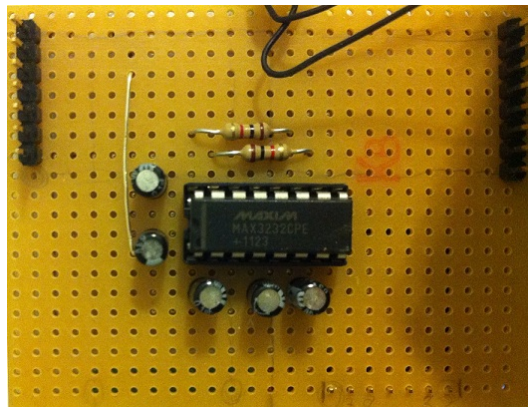


Figura 2.5: Convertitore RS232 - UART

| | '1' | '0' |
|-------|-----|-----|
| RS232 | -5V | 5V |
| CMOS | +3V | 0V |

Tabella 2.1: Livelli di tensione UART - RS232

2.2 Piattaforma Firmware

L'ambiente di sviluppo del firmware per la programmazione dei nodi è IAR Embedded Workbench for 8051. Il produttore, Texas Instruments, fornisce supporto per lo standard 802.15.4 a livello applicativo con software come Z-Stack, RemoTi o SimpliciTI. I software elencati però non consentono un'elevata personalizzazione quindi si è scelto di utilizzare lo stack applicativo TIMAC che fornisce tutte le librerie MAC per accedere al modulo radio, settare il canale e la potenza di trasmissione e gestire la trasmissione dei pacchetti. È disponibile anche un piccolo sistema operativo Operative System Abstraction Layer (OSAL) che consente di gestire timer, memoria ad allocazione dinamica, riavvio del sistema e altri importanti aspetti.

Anche se non è l'obiettivo di questa trattazione, l'utilizzo dello stack protocollare TIMAC consente anche la realizzazione di PAN Coordinator alimentati a batteria e non da rete fissa, in quanto non ci si deve attenere necessariamente alla trasmissione di dati in modalità beacon-enabled sviluppata ad esempio nelle reti ZigBee.

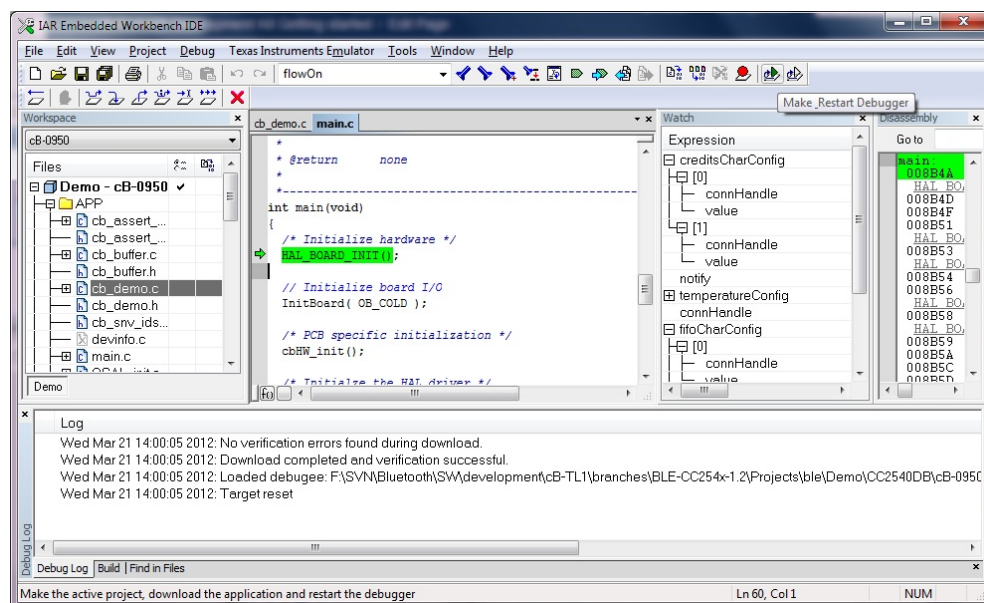


Figura 2.6: IAR Embedded Workbench

2.3 Piattaforma Software

Il linguaggio di programmazione del server di ricezione dei dati scelto è Java, poiché è completamente indipendente dal sistema operativo sottostante, nella versione 1.7. Viene utilizzato come IDE di sviluppo Eclipse.

L'interfaccia grafica, così come il server, viene sviluppata in Java.

Sviluppi futuri dell'applicazione prevederanno strutture a database per la gestione dei dati e interfacce di presentazione atte alla visualizzazione dei medesimi su browser. L'obiettivo della trattazione però è semplicemente l'analisi dei dati ricevuti, quindi ci si limiterà alla realizzazione di una applicazione Java contenente l'interfaccia grafica che ne consenta la visualizzazione.

2.4 Strumenti di debugging

Il debug, così come la programmazione, del firmware dei nodi 'The Guardian' della WSN è stato possibile grazie alla scheda di programmazione SmartRF05EB e al software IAR Embedded Workbench for 8051.



Figura 2.7: SmartRF05EB

L'analisi dei pacchetti 802.15.4 scambiati tra i nodi è realizzabile attraverso il dispositivo CC2531 configurato con firmware "Packet Sniffer for IEEE 802.15.4", collegato tramite porta USB ad un qualsiasi PC con OS Windows. Nel PC sarà necessario installare i driver relativi al CC2531 e il software di visualizzazione Packet Sniffer, fornito anch'esso da Texas Instruments.

Un altro software utilizzato sempre interfacciabile con il dispositivo CC2531 è Perytons Analyzer, utilizzato per la visualizzazione dei grafi ad albero della rete.



Figura 2.8: CC2531

| P.nbr | Time (us) | Length | Frame control field | | | | Sequence number | Dest. PAN | Dest. Address | Source Address | MAC payload | NWK Frame control field | | | | | | |
|------------|-----------|-----------|---------------------|-----|-----|---------|-----------------|-----------|---------------|----------------|----------------------------|----------------------------|---------|---------|----|-----|-----|---|
| RX | +0 | =0 | Type | Sec | Pnd | Ack.req | PAN_compr | 0xD5 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 0F 9F | Type | Version | DR | GA | Sec | | |
| | | | Cmd | 0 | 0 | 0 | 0 | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 | |
| P.nbr RX 2 | +15584322 | =15584322 | 29 | 0 | 0 | 0 | 1 | 0xD6 | 0xFAE3 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 0F 9F | Type | Version | DR | GA | Sec | |
| | | | | | | | | | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 |
| P.nbr RX 3 | +15070740 | =30655062 | 29 | 0 | 0 | 0 | 1 | 0xD7 | 0xFAE3 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 10 9F | Type | Version | DR | GA | Sec | |
| | | | | | | | | | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 |
| P.nbr RX 4 | +15067174 | =45722236 | 29 | 0 | 0 | 0 | 1 | 0xD8 | 0xFAE3 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 11 9F | Type | Version | DR | GA | Sec | |
| | | | | | | | | | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 |
| P.nbr RX 5 | +15069306 | =60791542 | 29 | 0 | 0 | 0 | 1 | 0xD9 | 0xFAE3 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 12 9F | Type | Version | DR | GA | Sec | |
| | | | | | | | | | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 |
| P.nbr RX 6 | +15069415 | =75860957 | 29 | 0 | 0 | 0 | 1 | 0xDA | 0xFAE3 | 0xFFFF | 0x0000 | 09 10 FC FF 00 00 01 13 9F | Type | Version | DR | GA | Sec | |
| | | | | | | | | | | | | FD 72 01 00 4B 12 00 08 60 | Cmd | 0x2 | 0 | 0 | 0 | 0 |

Figura 2.9: Esempio di interfaccia utente del Packet Sniffer

Capitolo 3

Progetto

In questo capitolo saranno discusse le caratteristiche del progetto. I requisiti funzionali specificano quali funzioni deve avere il sistema agli occhi dell'utente finale. I requisiti di sistema sono quelli che nascono dalla necessità di interfacciamento verso altri sistemi a cui è necessario adeguarsi per una perfetta sinergia. I requisiti non funzionali sono quelli che sono necessari al corretto comportamento del programma ma non ne aumentano le funzionalità.

A seguito della disamina dei requisiti saranno discussi e spiegati gli eventuali problemi riscontrati durante il test del sistema.

Infine si provvederà alla stesura dell'architettura logica finale del sistema, motivando le scelte progettuali.

3.1 Requisiti

3.1.1 Requisiti funzionali

Il progetto di una rete WSN agli occhi di un utente finale prevede semplicemente la capacità di visualizzare le misurazioni effettuate dai sensori installati e la possibilità di non dover intervenire sulla rete già installata, il che richiede da una parte la necessità di sviluppare metodi per gestire la dinamicità della rete ed eventualmente il controllo di essa da remoto, dall'altra la ricerca di un elevato lifetime in modo da evitare ogni tipo di

manutenzione sui dispositivi, dalla sostituzione delle batterie dei dispositivi al più classico spegnimento/riaccensione a causa di malfunzionamenti.

Il progetto prevede una rete in grado di effettuare un monitoraggio ogni 5 minuti e consentire in real time ad un software installato in un PC Server la visualizzazione dei dati aggiornati ricevuti dalla rete WSN.

3.1.2 Requisiti di sistema

Il progetto prevede sia la realizzazione della rete WSN sia l'implementazione di un software, seppur semplice, che realizzi la funzione di server per la ricezione dati e la funzione di interfaccia grafica per la visualizzazione dei dati. Dovendo implementare la rete dalle fondamenta, gli unici requisiti di sistema sono dati dai protocolli di comunicazione utilizzati dai dispositivi, ovvero 802.15.4 per la comunicazione tra i nodi e TCP/IP per il collegamento al PC server.

3.1.3 Requisiti non funzionali

Risparmio energetico

Una rete WSN prevede nodi con alimentazione a batteria, quindi con un tempo di vita determinato. È necessario perciò implementare meccanismi che consentano di massimizzare la durata delle batterie, sfruttando la capacità dei micro-controllori di entrare in una modalità di basso consumo energetico (sleep).

In base all'architettura della rete si possono definire diverse metodologie per il risparmio energetico. Alcune architetture consentono ai nodi di rimanere sempre in sleep ed essere svegliati solo su richiesta, attraverso interazioni del livello MAC con il livello fisico o attraverso la presenza di due canali radio. Altre architetture si basano sulla posizione dei nodi; se il sistema rileva due nodi vicini sarà sufficiente effettuare la misurazione solamente su uno di essi. Altre architetture eseguono una sorta di TDMA, ovvero si assegna ad ogni nodo uno slot temporale periodico in cui trasmettere e per il tempo rimanente il nodo entrerà in modalità di risparmio energetico.

Nella rete trattata si utilizza quest'ultima tecnica. Si imposta un parametro di progetto che rappresenta il periodo di campionamento dei sen-

sori all'interno della rete e si realizza un diagramma di flusso che prevede periodicamente:

- campionamento del sensore,
- invio del messaggio al coordinatore della rete,
- ingresso nella modalità risparmio energetico fino al successivo campionamento del sensore

I nodi intermedi con funzionalità di routing devono consentire l'inoltro verso il PAN Coordinator dei pacchetti ricevuti, quindi devono essere accesi nello slot temporale in cui i nodi associati trasmettono. Per poter preservare la durata della batteria questi nodi non possono rimanere sempre accesi. Si prevede un diagramma di flusso che periodicamente e in maniera sincronizzata con i nodi comporta:

- l'abilitazione della parte radio per la durata di uno slot temporale di lunghezza dipendente dal numero di nodi presenti,
- l'ingresso nella modalità risparmio energetico fino al successivo slot di trasmissione

Controllo della topologia della rete

Il controllo della topologia della rete sfrutta la ridondanza del numero di nodi per ridurre il numero di quelli attivi.

Se un nodo è situato molto vicino ad altri può essere sostituito da uno di essi, poiché i valori rilevati dai due adiacenti potrebbero risultare molto simili.

Grazie a questa tecnica si potrebbe ottenere un miglioramento del lifetime, ma per rilevare la propria posizione i dispositivi devono essere dotati di GPS, ragion per cui questa tecnica non è stata adottata nell'applicazione in esame. Sarebbe possibile effettuare un controllo grossolano grazie al rilevamento della potenza di trasmissione RSSI, ma a causa di una localizzazione poco precisa è stato scelto di non implementare il controllo di topologia.

Routing

Requisiti necessari per un ottimo protocollo di routing sono la possibilità di interconnessione tra tutti i nodi appartenenti alla rete, l'aggiornamento

dinamico dei path e la ricerca del percorso minimo, magari adattando la potenza di trasmissione al link che si ha a disposizione. Un protocollo di routing con queste caratteristiche consente di evitare la perdita di dati all'interno della rete e trasmettere i pacchetti con il minimo costo in energia possibile.

L'algoritmo di routing che si vuole realizzare deve garantire un lifetime della rete elevato, dove per lifetime si intende, in questa applicazione, il tempo entro il quale una percentuale di nodi definita rimane accesa e invia i propri dati al coordinatore. Per garantire questa condizione bisogna realizzare il protocollo di routing scegliendo come nodi intermedi quelli aventi maggiore energia e possibilmente anche quelli più vicini per minimizzare la potenza in trasmissione e conseguentemente i consumi.

Aggregazione dati

L'architettura della rete, che prevede la lettura dei dati ogni 5 minuti, non necessita di alcuna forma di aggregazione dati sia nei nodi figli, sia nei nodi intermedi. Questo perché si vuole avere una visualizzazione in tempo reale o in ogni caso entro i 5 minuti successivi del dato campionato. Se i requisiti della rete prevedessero un campionamento dei sensori ogni 5 minuti con una visualizzazione degli stessi ogni 30, i meccanismi di aggregazione contribuirebbero al risparmio energetico. Dovendo inviare i dati ad intervalli di 30 minuti, sarebbe possibile mantenere inattiva la parte radio del dispositivo per tempi più lunghi, diminuendo i consumi.

Sincronizzazione nodi

La sincronizzazione dei nodi è fondamentale per garantire una corretta temporizzazione della trama TDMA con la quale i nodi accedono al canale per inviare i messaggi al coordinatore, al fine di evitare collisioni nel mezzo wireless.

La sincronizzazione temporale è altrettanto importante perché i valori misurati devono essere legati ad un orario, essendo letti periodicamente. I nodi sono dotati di un Real Time Clock interno che ne consente la lettura con

ottima precisione, ma essendo questi componenti soggetti a deriva temporale necessitano di un meccanismo di sincronizzazione.

Memoria limitata

Le risorse limitate dei microcontrollori utilizzati rappresentano un fattore importante. È fondamentale considerare, in fase di programmazione, che non sarà possibile allocare tabelle o database di grandi dimensioni e che la loro allocazione nella memoria flash comporterà tempi di accesso ai dati elevati quindi una latenza nella trasmissione dei messaggi maggiore.

3.2 Analisi dei problemi

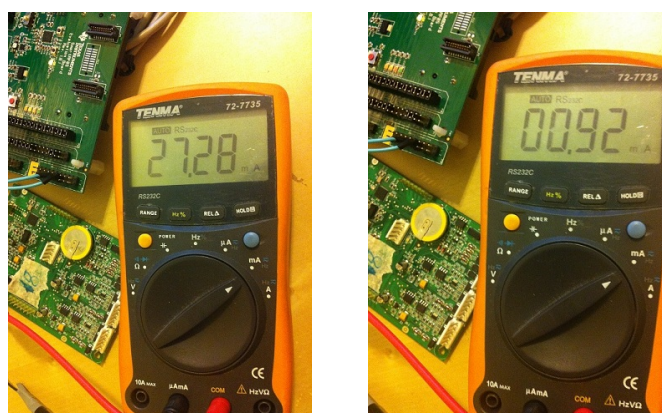
3.2.1 Analisi dei consumi

In questa sezione si esegue una disamina sui consumi effettivi del dispositivo *The Guardian* in modo da poter eseguire un calcolo teorico dei consumi.

Il nodo integra un convertitore A/D di precisione che richiede un'alimentazione a 5V. La tensione necessaria viene fornita attraverso un regolatore di tensione che può essere abilitato attraverso un pin di I/O del microcontrollore. Infatti l'accensione del secondo regolatore di tensione oltre a quello del micro-controllore a 3V3, implica dei consumi non trascurabili. Le modalità di analisi sono state due:

- multimetro in serie nella configurazione di amperometro
- oscilloscopio attraverso l'inserimento di una resistenza di Shunt

In un primo momento si è eseguita la misura con il multimetro, ma la velocità di elaborazione limitata del dispositivo non ha consentito di rilevare con precisione i consumi nelle brevi fasi di trasmissione poiché il multimetro rileva la media temporale dei valori letti.



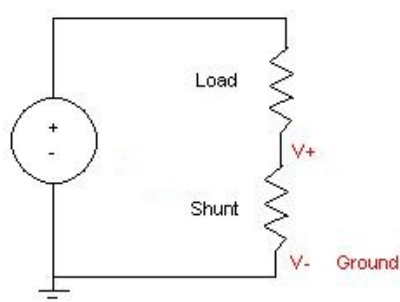
(a) Device On

(b) Sleep

Figura 3.1: Misura di corrente attraverso il multimetro

Per analizzare nel dettaglio la corrente assorbita si è deciso di utilizzare l'oscilloscopio, attraverso il quale però non è possibile effettuare misure di corrente. Risulta necessario l'inserimento di una resistenza di Shunt dimensionata con il valore di 100Ω .

A dispositivo spento nello Shunt non circola corrente quindi la tensione sulla resistenza è nulla, a dispositivo acceso un assorbimento di corrente da parte del nodo corrisponderà ad una caduta di tensione sullo Shunt proporzionale.

**Figura 3.2:** Misura di corrente con resistenza di Shunt, schema

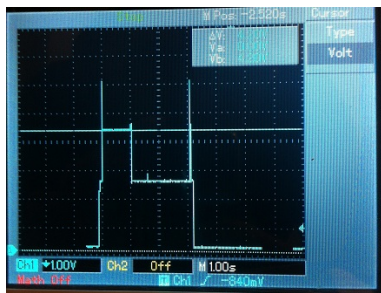
La scelta del valore [100Ω , 0.25 W] è stata fatta in base alle grandezze in gioco. La corrente da misurare sarà nell'ordine dei mA quindi si deve scegliere un valore di resistenza che rappresenti un trade-off tra l'elevata

caduta di tensione, se si sceglie una resistenza elevata, che può impedire l'accensione del nodo, e l'elevata dissipazione per effetto Joule, provocata dalla scelta di una resistenza troppo bassa.

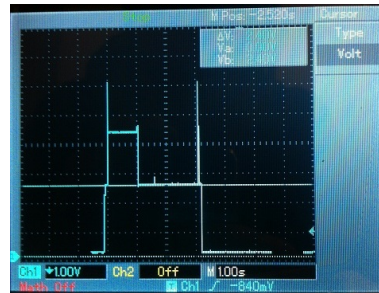
Alimentando il nodo con un alimentatore regolabile impostato a 9V otteniamo una potenza dissipata di 0.09 W compatibile con la resistenza scelta.

Non si esegue taratura dello strumento ma si nota che la misura differisce di un offset costante di 1 mA rispetto alle misurazioni eseguite con il multimetro, che prendiamo come riferimento perché realizza una misura più accurata.

$$W = V^2/R = 9/100 = 0.09W$$



(a) Radio ON, 5V ON



(b) RADIO ON, 5V OFF

Figura 3.3: Misura di corrente con resistenza di Shunt, oscilloscopio

Nell'esempio raffigurato il nodo abilita la parte radio appena si risveglia dallo sleep, riceve un pacchetto di sincronizzazione e accende il regolatore di tensione a 5 V per poter alimentare i sensori e il convertitore A/D. In questa fase il consumo è di 45 mA. Il campionamento dei sensori avviene dopo 1 secondo per poter garantire la lettura di un valore stabile. In seguito vengono spenti i regolatori a 5 V mentre la parte radio del dispositivo rimane accesa e invia i dati assorbendo una corrente di 25 mA. Al termine dello slot di invio un messaggio avverte i nodi che possono entrare nella fase di sleep, durante la quale il consumo scende a 920 μA.

| | | |
|----------|----------|---------|
| Radio ON | Radio ON | Sleep |
| 5V ON | 5V OFF | |
| 45 mA | 27 mA | 0.92 mA |

Tabella 3.1: Corrente assorbita The Guardian

3.2.2 Analisi della potenza di trasmissione

Analizzando i requisiti di progetto di routing si è considerato di poter adattare la potenza di trasmissione in base alla qualità del link trasmissivo, ma le i test effettuati hanno sconsigliato questa soluzione, in quanto diminuendo la potenza in trasmissione non si ottiene la riduzione dei consumi attesa o comunque non rilevante rispetto al costo di implementazione di un algoritmo di routing che si adatti alla qualità del link in esame.

La programmazione della potenza di trasmissione nel dispositivo CC2530 avviene attraverso il registro TX_POWER.

Dimostriamo che variando il valore del registro i consumi non hanno variazioni rilevanti monitorando la corrente assorbita con l'oscilloscopio. Verifichiamo l'effettiva variazione della potenza trasmessa servendoci del software Packet Sniffer che consente l'analisi della potenza ricevuta attraverso il parametro RSSI.

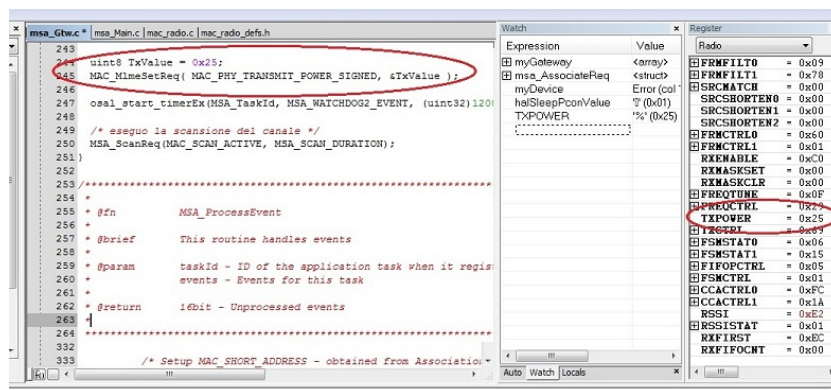


Figura 3.4: Impostazione potenza di trasmissione

Il protocollo utilizzato per l'analisi prevede l'attesa di un messaggio dal coordinatore. Completata la ricezione il nodo attende il suo turno di trasmissione entrando in sleep e successivamente provvede all'accensione del

regolatore 5V che consente l'acquisizione del segnale. La trasmissione della risposta avviene con un ritardo di 1,3 secondi rispetto al messaggio inviato dal coordinatore, come si può vedere nella figura 3.5.

È stato eseguito un test nel quale di volta in volta si varia la potenza in uscita per verificare che la corrente assorbita nell'istante della trasmissione non abbia variazioni rilevanti. Nelle figure che seguono saranno presentati due casi in particolare:

- -18 dBm
- +4.5 dBm

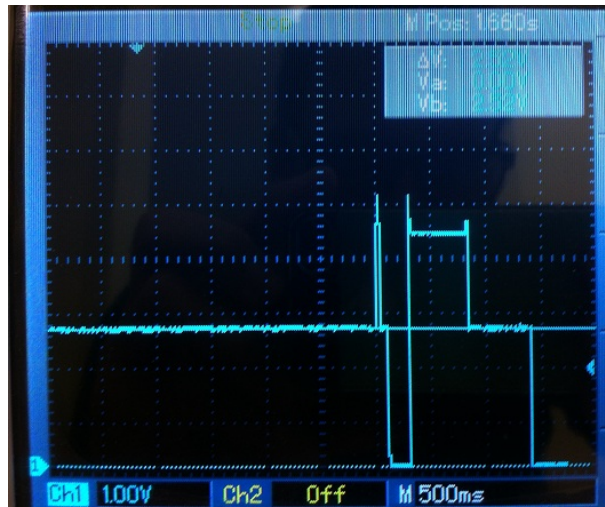


(a) Screenshot Oscilloscopio

| Pn.br. | Time (ms) | Length | Frame control field | Sequence number | Dest. PAN Address | Dest. Address | Source Address | MAC payload | RSSI (dBm) | FCS |
|--------|-----------|--------|--|-----------------|-------------------|---------------|----------------|--|------------|-----|
| RX 599 | +10605 | 18 | Type Sec Pnd Ack.req PAN_compr DATA 0 0 1 1 | 0x5D | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -78 | OK |
| RX 600 | +0 | 5 | Type Sec Pnd Ack.req PAN_compr ACK 0 0 0 0 | 0x5D | | | | | -76 | OK |
| RX 601 | +1302 | 27 | Type Sec Pnd Ack.req PAN_compr DATA 0 0 1 1 | 0xA1 | 0x11CC | 0x0100 | 0x0102 | 04 00 02 01 02 00 00 00 00 00 00 00 00 13 CD 00 | -75 | OK |
| RX 602 | +1 | 5 | Type Sec Pnd Ack.req PAN_compr ACK 0 0 0 0 | 0xA1 | | | | | -78 | OK |

(b) Screenshot Packet Sniffer

Figura 3.5: Analisi consumi, Output Power = -18 dBm



(a) Screenshot Oscilloscopio

| P.nbr. | Time (ms) | Length | Frame control field | Sequence number | Dest. PAN Address | Dest. Address | Source Address | MAC payload | RSSI (dBm) | FCS |
|--------|-----------|--------|--------------------------------|-----------------|-------------------|---------------|----------------|-------------------------|------------|-----|
| RX | +9301 | 18 | Type Sec Pnd Ack.req PAN_compr | 0x53 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -81 | OK |
| 743 | =6323288 | 5 | ACK 0 0 0 0 | 0x53 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -81 | OK |
| RX | +0 | 5 | Type Sec Pnd Ack.req PAN_compr | 0x53 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -58 | OK |
| 744 | =6323289 | 5 | ACK 0 0 0 0 | 0x53 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -58 | OK |
| RX | +1303 | 27 | Type Sec Pnd Ack.req PAN_compr | 0x26 | 0x11CC | 0xFFFF | 0x0102 | 04 00 02 01 02 00 00 00 | -56 | OK |
| 745 | =6324592 | 27 | DATA 0 0 1 1 | 0x26 | 0x11CC | 0xFFFF | 0x0102 | 00 00 00 00 00 15 C8 00 | -56 | OK |
| RX | +1 | 5 | Type Sec Pnd Ack.req PAN_compr | 0x26 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -94 | OK |
| 746 | =6324593 | 5 | ACK 0 0 0 0 | 0x26 | 0x11CC | 0xFFFF | 0x0100 | 00 00 00 | -94 | OK |

(b) Screenshot Packet Sniffer

Figura 3.6: Analisi consumi, Output Power = 4,5 dBm

Dall'analisi dei test si denota che, a discapito di una diminuzione della potenza di uscita, i consumi rimangono invariati. Il comportamento del dispositivo diverso dalle aspettative potrebbe essere causato dall'assorbimento di corrente necessario all'alimentazione della parte radio in ricezione. Per avvalorare questa ipotesi si esegue un test che prevede la diminuzione della sensibilità del ricevitore radio diminuendo il guadagno degli amplificatori in ricezione nel dispositivo CC2530. Si nota che effettivamente si ha una riduzione di corrente di 4 mA, da 27 e 23. Questo vantaggio comporta però la diminuzione drastica delle prestazioni in termini di range di copertura, quindi in possibili applicazioni future bisogna valutare bene il trade-off offerto.



Figura 3.7: Riduzione del guadagno in ricezione

3.2.3 Analisi dei protocolli di routing

In questa sezione verranno esaminati vari tipi di protocolli di routing, sviluppati o descritti solo a livello teorico, per comprendere quale è la migliore soluzione realizzabile per il sistema descritto in questa trattazione.

Una prima classificazione dei protocolli di routing, di seguito elencati, viene realizzata in base al metodo di memorizzazione dei percorsi.

- Proattivi: viene mantenuta la topologia di tutta la rete, che è aggiornata periodicamente. Tutti i nodi sanno come raggiungere ogni altro nodo ad ogni istante.
- Reattivi: costruiscono un cammino quando necessario e lo mantengono finché è valido e hanno una determinata latenza ogni volta che si deve trasmettere un pacchetto ad un nodo mai raggiunto prima.
- Ibridi: all'interno di un cluster il routing è di tipo proattivo, mentre tra i vari cluster il routing è di tipo reattivo.

In una rete WSN i dispositivi possono avere diversi compiti, quali PAN Coordinator, nodi FFD a funzionalità complete e RFD a funzionalità ridotte. Un'ulteriore classificazione viene realizzata in base ai ruoli che assumono i nodi all'interno della rete.

- Protocolli Flat
- Protocolli Gerarchici
- Protocolli Location-Based

Protocolli flat

Nei protocolli flat tutti i nodi hanno lo stesso ruolo e seguono sempre determinate regole. Dovendo eseguire le medesime operazioni questi protocolli si basano su un approccio data-centric / data-dissemination. Data-centric perché i dati provenienti da tutti i nodi vengono indirizzati verso il sink e data-dissemination perché i dati provenienti dal sink vengono distribuiti all'interno della rete consentendone la ricezione da parte di tutti i nodi.

Riportiamo di seguito alcuni esempi.

- Flooding

È un protocollo elementare che prevede l'inoltro dei messaggi in broadcast verso tutti i vicini. La semplicità si paga con consumi energetici elevati.

- Gossiping

L'algoritmo è simile al flooding ma evita la congestione della rete effettuando l'inoltro solamente verso uno dei vicini, scelto casualmente.

- Sensor Protocol for Information via Negotiation.

Viene scelto soprattutto nelle reti aventi nodi non rappresentati da un identificativo univoco (ID). La trasmissione di un messaggio da parte del coordinatore prevede in precedenza l'invio in broadcast di un'identificazione del messaggio. Solo i nodi interessati all'informazione risponderanno al coordinatore, richiedendo l'invio dei dati finali.

- Directed diffusion

Ogni dato è contraddistinto da una coppia attributo/valore. L'identificazione dei nodi è legata all'attributo mentre il valore definisce un gradiente che ha il compito di indirizzare il pacchetto. La propagazione dell'identificazione crea tra i nodi i gradienti relativi al pacchetto che corrisponde all'interesse. Una volta completata la diffusione dell'identificazione, i dati vengono inoltrati secondo il percorso migliore, definito dall'andamento del gradiente stesso nella rete. È un protocollo proattivo perché prima dell'invio del dato prevede l'instaurazione di un percorso.

- Ad Hoc On-Demand Distance Vector (AODV)

È un protocollo di routing reattivo ed è implementato nell'architettura ZigBee. Ogni nodo mantiene in memoria le informazioni sul next hop, realizzando una tabella di routing locale. Se il next hop non è conosciuto, viene eseguito il processo di route discovery. Quando un nodo effettua tale operazione invia in broadcast un comando di richiesta route che contiene l'indirizzo del mittente, l'indirizzo del destinatario e un costo del path. Ogni volta che il comando si propaga nella rete i nodi, che ricevono e ritrasmettono il pacchetto, incrementano il campo del costo del path e aggiungono una entry alla loro tabella di route discovery. Il destinatario potrà ricevere il comando da più nodi; sarà suo compito analizzare quello avente percorso di costo minimo e rispondere al mittente con il pacchetto di route reply instradandolo correttamente.

- Minimum Cost Survivor Path (MCSP)

MCSP è un protocollo di tipo flat basato sull'algoritmo classico di Bellman-Ford e articolato su due fasi: la prima prevede la distribuzione in broadcast di un pacchetto che consenta ai nodi di definire i percorsi a costo minimo, la seconda prevede la trasmissione attraverso il next hop definito in fase di routing.

Nella prima fase si invia un messaggio contenente un valore chiamato Accumulated Cost (AC), ottenuto dalla stima di parametri come link quality, energia disponibile nel nodo e numero di hop. Ogni nodo calcola il proprio costo, lo somma a quello del pacchetto ricevuto e trasmette nell'etere le proprie informazioni.

Un nodo può ricevere diversi messaggi che indicano diversi cammini e, scegliendo il pacchetto avente AC minore, otterrà automaticamente il cammino a costo minimo. I cammini aventi costo maggiore vengono semplicemente scartati, come nell'algoritmo di Viterbi.

Analizzando la figura 3.8, possiamo identificare due scenari, dove il nodo sink si trova esternamente o internamente alla rete. I risultati dei test mostrano che il protocollo MCSP ha prestazioni migliori quando

il sink è interno alla rete, mentre sono comunque soddisfacenti quando è esterno.

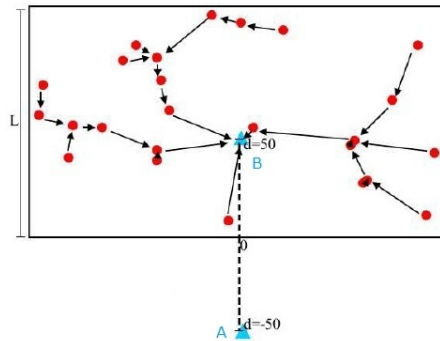


Figura 3.8: Scenari di test MCSP [1]

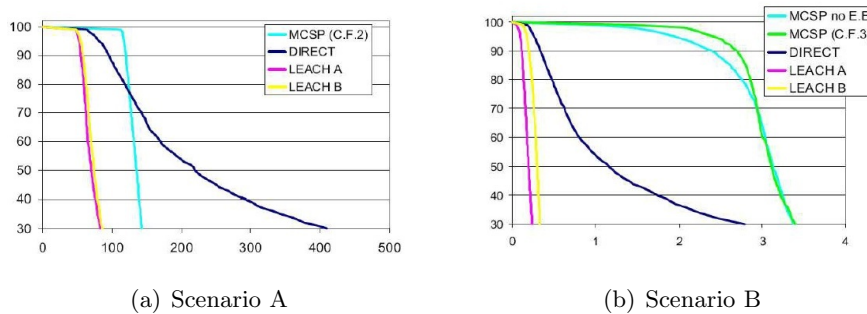


Figura 3.9: Prestazioni MCSP, % nodi presenti VS cicli di esecuzione [1]

Protocolli gerarchici

La rete viene organizzata in base al ruolo che i nodi possono assumere al suo interno. Alcuni nodi infatti, possono prendere le funzioni di cluster-head e creare una sottorete. Quando un nodo appartenente alla sottorete deve comunicare, lo farà attraverso il suo cluster-head, che potrà inviare il messaggio al destinatario, se appartenente alla stessa sottorete, attraverso il passaggio da un altro cluster-head in caso contrario. Così facendo si riduce notevolmente l'overhead in fase di routing.

- Low Energy Adaptive Clustering Hierarchy (LEACH)
È forse il più famoso protocollo di clustering e si realizza attraverso

due fasi: la fase di setup e la fase di steady.

La prima è necessaria per l'assegnazione del compito di cluster head ad alcuni nodi interni alla rete. I rimanenti nodi si uniscono al cluster head più vicino per minimizzare i consumi in trasmissione. Dopo le associazioni, i cluster-head possono mantenere in memoria le tabelle di routing riguardanti tutta la rete. Il compito di un cluster head sarà trasmettere un pacchetto al cluster head del destinatario. Sarà poi quest'ultimo ad indirizzare il pacchetto all'interno della sua rete verso il destinatario scelto. La fase di setup si ripete periodicamente, per effettuare una rotazione dei clusterhead, dato che sono i dispositivi che consumano una quantità maggiore di corrente rispetto ai nodi figli.

Nella seconda fase, quella di steady vengono invece trasmessi i dati. I nodi figli, che rilevano i dati dei sensori, trasmettono verso i cluster head. Come nella fase di steady, sarà il cluster head ad indirizzare i dati verso il sink, dopo averli aggregati se necessario.

- Pegasis

Il protocollo Pegasis è un miglioramento del LEACH. Il vantaggio si realizza attraverso la creazione di un meccanismo a catena, per cui i nodi dovranno trasmettere dati solamente ai loro vicini, aggregandoli lungo le catene, fino a raggiungere il nodo leader, incaricato di trasmetterli al sink. Il nodo con le funzioni di leader viene continuamente alternato per garantire uniformità nei consumi.

Protocolli location-based

I nodi sfruttano le informazioni provenienti da un dispositivo di geolocalizzazione come il GPS per inviare i dati ottimizzando l'energia. In questa trattazione non li esaminiamo in quanto non disponiamo dell'hardware adeguato per realizzare protocolli di questo tipo.

3.3 Architettura logica

L'architettura di rete prevede l'utilizzo di 9 nodi e un PC Server.

- 1 Nodo PAN Coordinator (sink, gateway)
- 8 Nodi FFD/RFD

Il nodo PAN Coordinator viene interfacciato al PC Server per consentire archiviazione ed elaborazione dei dati ricevuti e la gestione della rete.

Gli altri nodi si occupano di raccogliere e inoltrare i dati raccolti verso il coordinatore. In base al tipo di protocollo di routing i nodi possono assumere le funzioni di RFD se svolgono un semplice sensing o di FFD se svolgono funzioni di coordinamento e inoltro.

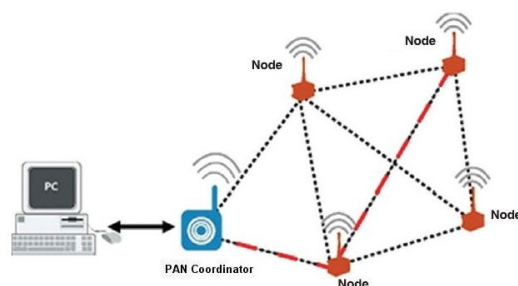


Figura 3.10: Architettura del progetto



(a) PAN Coordinator



(b) Nodo RDF/FFD

L'obiettivo della tesi è lo sviluppo di un nuovo protocollo di routing basato sulla ricerca della migliore prestazione in termini di lifetime della rete, dove come definizione di lifetime si intende entro quanto tempo una parte significativa della rete si spegne e di conseguenza non trasmette più i dati dei relativi sensori. Il parametro in esame sarà quindi la tensione ai capi delle batterie che alimentano i nodi.

I protocolli di routing realizzati sono:

- un protocollo flat con un algoritmo simile al MCSP. Il ruolo dei nodi all'interno di questo protocollo è statico, quindi saranno configurati 5 nodi come RFD e 3 nodi come FFD
- un protocollo gerarchico simile al protocollo Leach, nel quale i nodi saranno in grado arbitrariamente di lavorare come RFD o FFD.
- un protocollo di flooding, rappresentante il caso peggiore

Per i tre protocolli è previsto il test in due diversi ambienti.

Nello Scenario A viene disposto un solo nodo FFD per ogni sottorete creata, creando un ambiente dove non è possibile effettuare routing.

Nello scenario B, i nodi interni alla rete avranno a disposizione più percorsi che verranno scelti arbitrariamente in base a parametri come numero di hop, energia dei nodi e qualità del link, potendo quindi effettuare routing.

L'implementazione dei protocolli e i motivi che hanno spinto alla realizzazione di alcune modifiche rispetto alle descrizioni teoriche sono spiegati nel paragrafo 4.4.

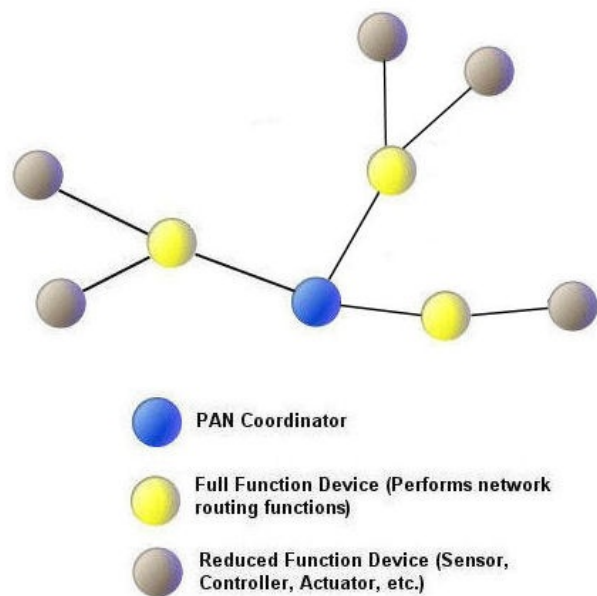


Figura 3.11: Scenario A di test

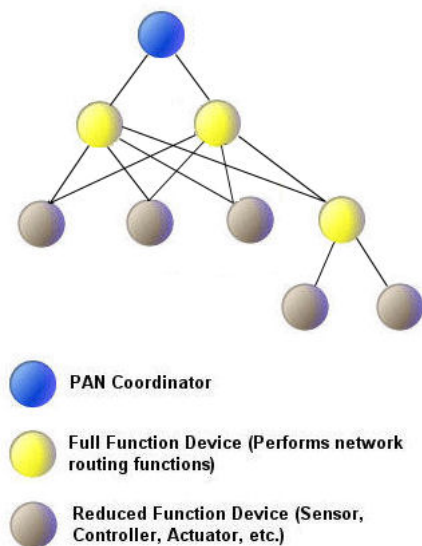


Figura 3.12: Scenario B di test

Capitolo 4

Implementazione Firmware

Le caratteristiche del protocollo 802.15.4 prevedono l'instaurazione di una comunicazione a partire dal meccanismo di beaconing. Il beacon è una delle 4 strutture di dati definite nel 802.15.4, oltre ai frame di DATA, ACK e MAC Command.

Esistono due modalità di comunicazione: beacon-enabled e beacon-disabled. La prima prevede l'invio periodico da parte del PAN Coordinator del beacon ad intervalli nell'ordine dei ms, richiedendo quindi un dispositivo alimentato da rete fissa. L'abilitazione dei beacon è utile nei contesti in cui si possono avere velocità di trasferimento elevate, come ad esempio nel collegamento tra periferiche di un pc; contrariamente si può utilizzare la modalità beacon-disable. In quest'ultima modalità il beacon viene utilizzato solo per l'associazione, mentre per lo scambio di messaggi tra due dispositivi non è richiesto. Ciò consente un notevole risparmio di energia e prevede la possibilità di alimentare i nodi PAN Coordinator a batteria. Per i motivi sopra descritti si è scelto di utilizzare questa configurazione.

4.1 Gestione associazioni

L'associazione dei nodi alla rete è realizzata attraverso lo scambio di un pacchetto di beacon. Un nodo, al suo avvio, effettua una scansione del mezzo trasmissivo richiedendo un pacchetto che indichi la presenza di altri nodi all'interno della rete. I coordinatori presenti vedono la richiesta e

rispondono inviando il beacon, un messaggio contenente dati quali il canale di comunicazione, il proprio indirizzo e alcune informazioni sul tipo di protocollo utilizzato (meccanismi di sicurezza, eventuale payload). La trasmissione dei beacon è implementata completamente a livello MAC e non consente l'impostazione di un ritardo per la realizzazione di trame TDMA, quindi se più coordinatori sono presenti nella rete, questi si contenderanno il canale e nel caso sia occupato, metteranno in pratica l'algoritmo di back-off per la ritrasmissione dei dati.

Una volta ricevuti i vari pacchetti di beacon, modificati rispetto allo standard al fine di inserirvi le informazioni di routing, il nodo che si vuole associare alla rete sceglie quale coordinatore gli consente di effettuare il percorso a costo minimo e invia una richiesta di associazione. Un primo routing avviene in questa fase e sfrutta l'approccio cross-layer del protocollo 802.15.4 tra i livelli 2 e 3. La risposta alla richiesta di associazione consente l'impostazione dei parametri di rete a livello MAC e quindi l'abilitazione del dispositivo radio all'invio di messaggi all'interno della rete.

Inizialmente si era pensato di sviluppare tutti gli algoritmi di routing sfruttando i beacon a livello MAC, ma i vincoli descritti hanno indirizzato la nostra scelta sulla realizzazione del routing a livello applicativo.

Per l'analisi della dissociazione da un coordinatore distinguiamo due sotto-casi. Il primo è quello in cui un nodo, una volta entrato a far parte di una rete, attraverso un aggiornamento dei percorsi di routing, identifichi un padre che gli consenta di compiere una trasmissione verso il nodo sink a costo minore di quella attuale. In tal caso il nodo si dissocia dal padre per eseguire una richiesta al nuovo nodo. Il secondo caso si verifica quando invece un nodo viene spostato, si spegne o viene eliminato dalla rete. Il nodo non potrà più notificare al nodo padre la volontà di dissociarsi, perciò sarà compito del nodo padre, come indicato nella sezione 4.2, rilevare l'assenza del nodo figlio.

4.2 Gestione inserimento/esclusione nodi

L'inserimento o esclusione di nodi devono essere realizzata con modalità plug&play, ovvero senza la necessità di riavviare la rete una volta in-

serito/escluso un nodo. Tale procedura richiede l'analisi separata dei due casi.

Il caso dell'inserimento di un nuovo dispositivo è abbastanza semplice perché, come detto, all'avvio un nodo è programmato per effettuare la scansione del canale e, una volta rilevata la presenza di un coordinatore vi si associa.

Si può riscontrare un ritardo nell'associazione nel caso in cui il nodo entrante non possa comunicare con il PAN Coordinator (livello 0), sempre alimentato quindi sempre disponibile a soddisfare tali richieste. In questo caso il nodo si dovrà legare ad un altro di livello superiore allo zero, alimentato a batteria e quindi acceso solamente per brevi intervalli. Il nodo entrante dovrà attendere la fase di aggiornamento delle tabelle di routing, nella quale tutti i nodi rimangono attivi per qualche secondo, tempo utile a garantire le associazioni dei nodi che volessero entrare a far parte della rete.

L'inserimento può essere realizzato in qualsiasi istante perché il coordinatore prevede l'invio di pacchetti contenenti lo stato della rete e i tempi di sleep che ogni nodo deve rispettare e perciò ogni nodo che si associa, alla ricezione di tale pacchetto, avrà a disposizione tutte le informazioni necessarie alla sua sincronizzazione con gli altri nodi.

Nel protocollo sviluppato si inviano la durata dello slot di trasmissione, la durata dello slot di inattività, la data attuale e lo stato della rete (regime di trasmissione o di aggiornamento delle tabelle di routing). In sviluppi futuri dell'applicazione si potrebbero settare con questa procedura altri parametri, ad esempio il tempo di stabilizzazione dei sensori prima dell'invio del dato, oppure si potrebbero realizzare richieste personalizzate per ogni nodo, andando a rilevare un tipo di sensori ogni 5 minuti e un altro ogni 20 minuti.

Il caso dell'esclusione del dispositivo è un po' più complesso da gestire. Il coordinatore di un dispositivo che viene estromesso dalla rete deve riconoscere che il nodo escluso non effettua più trasmissioni e quindi eliminarlo dalla lista. Il suddetto controllo è realizzato attraverso una variabile booleana che analizza se il nodo ha effettuato trasmissioni o meno nell'arco temporale tra due aggiornamenti della tabella di routing e, in caso negativo, si procede all'eliminazione.

4.3 Gestione raccolta dati e sincronizzazione nodi

La raccolta dei dati segue un paradigma data-dissemination / data-centric, ovvero il nodo sink effettua una richiesta dati che viene disseminata in broadcast in tutta la rete. I nodi, una volta ricevuto questo messaggio, attendono un tempo determinato dal loro ID nella rete, quindi univoco, per avere accesso esclusivo al canale e trasmettono il loro dato con modalità multihop verso il sink, quindi data-centric. Al termine della trasmissione i nodi entrano nella fase di risparmio energetico, nella quale vi rimangono fino al successivo intervallo di trasmissione del sink. Il tempo di inattività quindi deve essere fornito dal nodo sink stesso. Al risveglio, tenuto conto del tempo di guardia, il sink invierà attraverso il meccanismo di data-dissemination la successiva operazione da svolgere. Questo metodo consente, oltre alla raccolta dei dati, una sincronizzazione completa di tutta la rete, sia a livello di trama TDM sia a livello di sincronizzazione temporale grazie all'invio all'interno del messaggio dell'orario attuale.

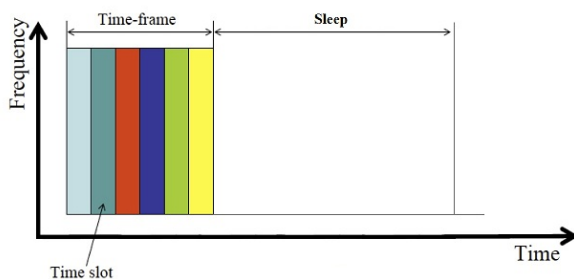


Figura 4.1: Trama TDM

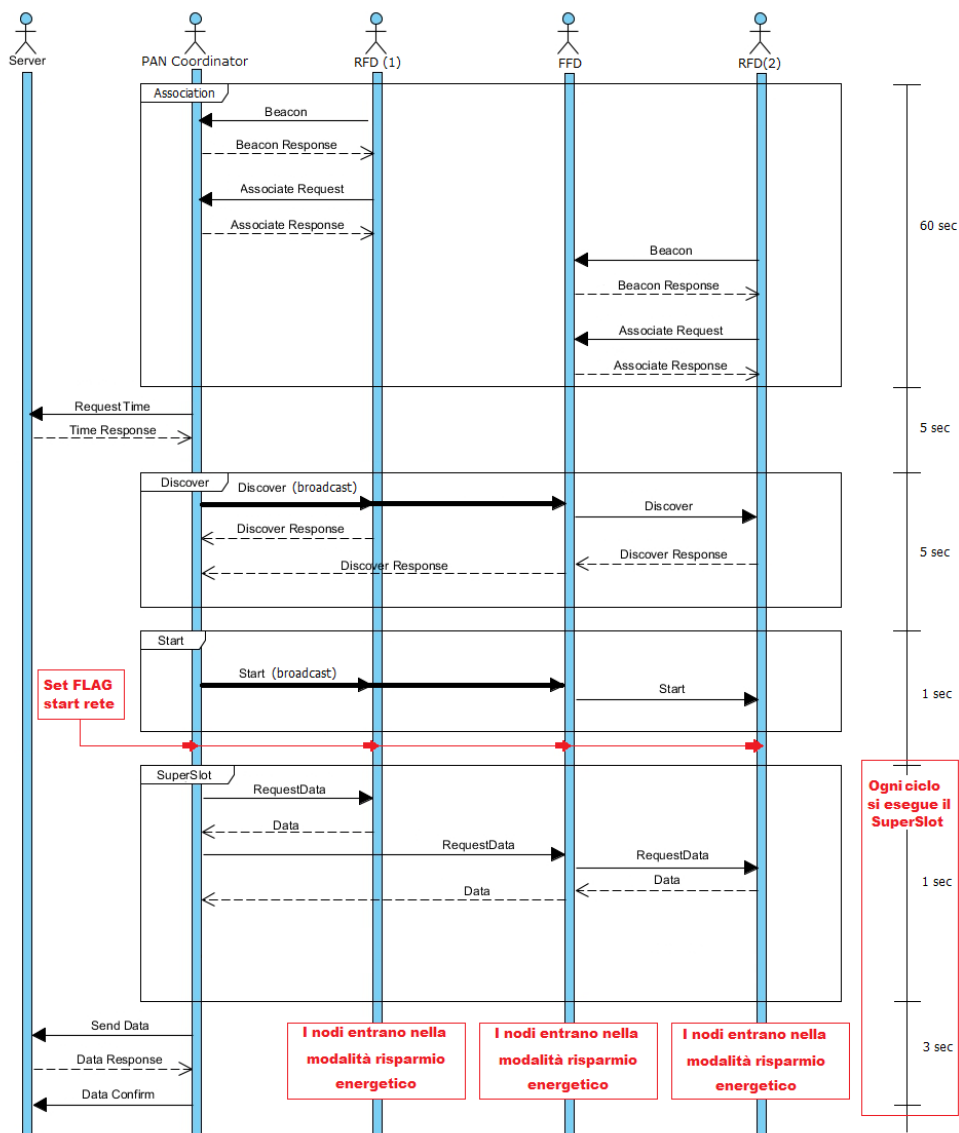


Figura 4.2: Schema della comunicazione nella rete

In seguito sono mostrati i messaggi inviati rispettivamente dal sink verso i nodi e viceversa.

| ID Msg | Date | Time | Network State | Sleep Time |
|--------|------|------|---------------|------------|
| 1B | 3B | 3B | 1B | 2B |

Tabella 4.1: Messaggio inviato dal sink nel SuperSlot

| ID Msg | ID Node | Port | Type | Date | Time | Value | Battery | Link |
|--------|---------|------|------|------|------|-------|---------|------|
| 1B | 2B | 1B | 1B | 3B | 3B | 2B | 2B | 1B |

Tabella 4.2: Messaggio inviato dal nodo nel SuperSlot

4.4 Protocollo di routing

In questa sezione sono descritti i protocolli di routing adottati e vengono spiegati motivi che hanno spinto alla scelta di tali algoritmi.

La rete sarà applicata in un ambiente dinamico, per cui sarebbe consigliato l'utilizzo di protocolli reattivi anziché di protocolli proattivi, i quali mantengono una topologia di tutta la rete e sono più adatti per applicazioni statiche. Di contro, i protocolli reattivi richiedono l'accensione di tutti i nodi della rete ogni volta che uno di essi, in mancanza del successivo hop, deve ricercare il migliore percorso. L'accensione di tutti i nodi comporta un notevole dispendio di energia, perciò si è deciso di implementare protocolli proattivi, che memorizzano al loro interno le informazioni di routing e le aggiornano periodicamente, ma locali, ovvero che memorizzano solamente il next hop. Solo il PAN Coordinator avrà una visione completa di tutta la rete.

Eliminando in partenza i protocolli basati sulla localizzazione, in quanto l'hardware fornito non è dotato di GPS o dispositivi simili, si dovranno concentrare le analisi sui protocolli flat o gerarchici.

All'interno dei protocolli flat sarà possibile scartare quelli basati sull'advertising. La ragione di questa scelta deriva dal fatto che i nodi della rete hanno un identificativo univoco e che la rete che si vuole realizzare prevede

la richiesta dei dati a tutti i sensori, e non solo ad una parte di essi. Non sarà necessario quindi richiedere ai nodi della rete informazioni come la loro identità o se sono interessati ad un determinato tipo di messaggio.

Si è scelto un protocollo MCSP, invece di AODV, perchè dalle ipotesi fatte un protocollo proattivo locale consente un maggiore risparmio energetico rispetto ad un protocollo reattivo. Questi ultimi, qualora un nodo non riuscisse a trasmettere al next hop predefinito, prevedono l'invio di un messaggio in broadcast richiedendo la risposta di tutti i nodi. Un protocollo come MCSP invece, prevedendo due fasi, quella di trasmissione e quella di aggiornamento delle tabelle di routing, non richiede tale specifica e consente di definire uno slot temporale entro il quale tutti i nodi, accesi, calcoleranno i nuovi next hop. Per quanto riguarda i protocolli flat si sceglie quindi MCSP, aggiornato con piccole modifiche.

Tra i protocolli gerarchici si sceglie LEACH, anch'esso leggermente modificato al fine di migliorare l'efficienza energetica. La scelta non ricade su Pegasus perché, come indicato nella sezione 3.2.2, la potenza di trasmissione non incide sui consumi, quindi aggregare dati tra i nodi vicini invece di inviarli direttamente al coordinatore di riferimento è solamente uno spreco di risorse temporali ed energetiche.

Come base di confronto per i test implementati, si realizza un protocollo di flooding che individua il caso peggiore.

In tutti i protocolli la scelta di non effettuare data aggregation porta all'inoltro diretto di ogni messaggio dal nodo che ha effettuato sensing verso il sink attraversando tutti i nodi che creano il percorso a costo minimo. La scelta è motivata dalla natura real-time della rete in esame (sezione 3.1.3) e dall'analisi dei consumi in trasmissione che evidenzia come l'invio di più messaggi non sia un elemento rilevante ai fini dei consumi.

4.4.1 Protocollo Simil-MCSP

Il protocollo prevede la distinzione tra due tipologie di nodi: RFD, incaricati solo di compiere sensing, e FFD, incaricati di gestire sottoreti e inoltrare dei dati.

Associazione

Un primo meccanismo di routing si realizza in fase di associazione, come descritto in 4.1.

La bontà di questa tecnica risiede nel fatto che ogni nodo prima di diventare coordinatore e poter inviare il beacon, deve associarsi ad un padre. Aiutandosi con la figura 4.3, si immagina la struttura a livelli, nella quale la trasmissione è consentita solo tra livelli adiacenti.

Il primo nodo che può assumere il ruolo di coordinatore ed essere abilitato all'invio del beacon è al livello 0, dove abbiamo il PAN Coordinator.

Se un nodo posto a livello 3 si accende ed effettua la scansione del canale non potrà comunicare con il PAN Coordinator perché fuori dalla sua copertura radio, quindi rieseguirà periodicamente la scansione.

Un nodo a livello 2 che si accende ed effettua la scansione, scoprirà che nel suo raggio di copertura sarà presente il nodo a livello 3 che, non essendo associato a nessun padre, non risponderà alla richiesta di associazione. In questo modo si garantisce che i nodi di ogni livello si associno solamente con nodi di livello inferiore.

Un nodo a livello 1 inserito in seguito potrà comunicare con il PAN Coordinator a livello 0, quindi vi si assocerà e sarà abilitato a rispondere alle richieste provenienti dal livello 2. Questa procedura consente di evitare possibili legami a nodi che non possono comunicare con il sink, il destinatario dei messaggi di tutti i nodi.

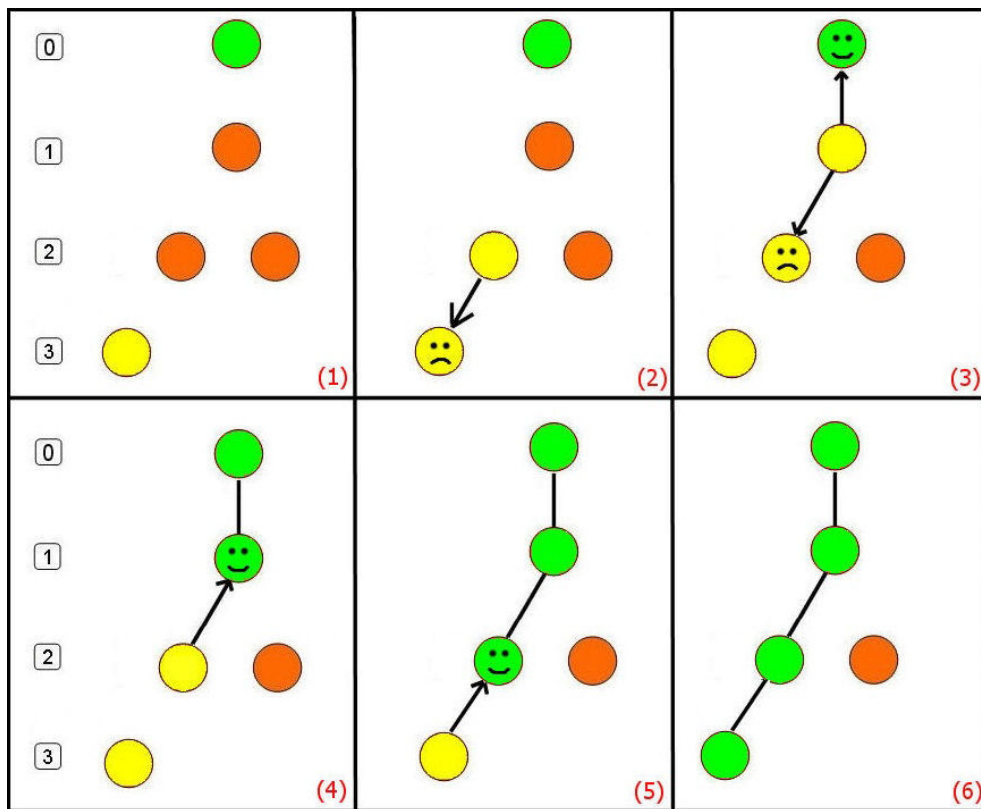


Figura 4.3: Modalità di associazione

Flusso di controllo

In questa configurazione i nodi abilitati gestire le associazioni saranno solamente i nodi FFD, mentre i nodi RFD potranno solamente associarsi ai padri posti all'interno della loro copertura radio. Una volta realizzato ciò, il sink avvia il flusso di controllo indicando periodicamente, attraverso un messaggio disseminato per la rete, il compito da eseguire, che esso sia la trasmissione o l'aggiornamento delle tabelle di routing.

La trasmissione dei dati avviene ogni 5 minuti mentre l'aggiornamento delle tabelle di routing avviene ogni 30 minuti.

La fase di aggiornamento si realizza inviando una Route Request in broadcast da parte ogni nodo. Coloro che riceveranno la richiesta risponderanno inviando un messaggio contenente numero di hops, tipo di alimentazione del nodo e stato della batteria. Grazie alla rilevazione del link quality in ricezione si ottiene un quarto parametro utile alla definizione del percorso a costo minimo.

Ogni nodo impiega circa 1,5 secondi per questa operazione. Un RFD, che deve semplicemente aggiornare la propria tabella rimarrà acceso solo 1,5 secondi, mentre un FFD che deve rispondere alle Route Request di tutti i nodi rimarrà acceso per 1.5 secondi moltiplicato per il numero di nodi presenti nella rete. L'implementazione di reti di grandi dimensioni potrebbe portare ad elevati tempi di aggiornamento, ma raggruppando le varie reti in sottogruppi caratterizzati da diversi PAN ID si può risolvere questo problema.

Novità

Rispetto al caso MCSP l'algoritmo proposto si differenzia nella definizione della funzione costo in quanto:

- analizza se il dispositivo è alimentato e nel caso massimizza il valore della batteria
- non considera l'energia del percorso accumulata ma solamente l'energia dell'ultimo nodo. Prendendo in esame l'energia accumulata tra due nodi con carica [100%, 20%] e due nodi aventi carica [60%, 50%], il calcolo dell'energia accumulata farebbe cadere la scelta sul percorso

con più energia totale, quindi il primo, correndo il rischio di scaricare totalmente il nodo con il 20% di energia.

Il costo di ogni percorso viene calcolato realizzando una combinazione lineare dei parametri in esame.

$$CF_i = \alpha \sum_{i=0}^{i-1} H_i + \beta E_{i-1} + \gamma LQ_{(i,i-1)} \quad (4.1)$$

dove CF indica il Cost Function, $\sum_i H_i$ indica il numero di hops accumulato, E indica l'energia normalizzata del nodo padre e può essere espressa calcolando la tensione della batteria rispetto al valore massimo (se il dispositivo è alimentato tale valore si massimizza a 1) e LQ indica la qualità del mezzo trasmissivo normalizzata

I coefficienti α, β, γ possono essere dimensionati in base all'importanza che si vuole dare ai vari parametri. Nel caso in esame diamo maggiore rilevanza al numero di salti; è fondamentale infatti che un messaggio venga inviato al sink con il minimo numero di salti possibile. L'energia residua e il link quality, in questo ordine, completano la definizione del costo. La configurazione finale testata con ottimi risultati risulta $[\alpha = 3, \beta = 3, \gamma = 1]$. Nel caso di applicazioni indoor, dove i cammini multipli portano a risultati molto variabili, il link quality può essere trascurato settando $\gamma = 0$.

4.4.2 Protocollo Simil-LEACH

Il protocollo Simil-LEACH è stato costruito sulla base del protocollo MCSP, formato anch'esso da due fasi, una di setup della rete e una di steady nella quale avvengono le trasmissioni dei dati dei sensori verso il sink.

La differenza principale tra i due algoritmi risiede nel ruolo che svolgono i nodi. Nel protocollo Simil-MCSP ogni nodo ha il suo compito definito, mentre in questo protocollo tutti i nodi possono operare in entrambe le modalità, RFD o FFD. Tale miglioramento consente la possibile comunicazione tra tutti i nodi, introducendo vantaggi multipli.

- in MCSP un nodo posto nel raggio di copertura di un RFD non poteva essere inserito nella rete perché l'associazione era possibile solo con nodi FFD, mentre in Simil-LEACH ogni nodo è visibile agli altri

- in MCSP un nodo trasmette verso il sink solo attraverso gli FFD, mentre in Simil-LEACH l'inoltro è abilitato per tutti i nodi

Potendo operare in entrambe le modalità il protocollo Simil-LEACH consente di ruotare i ruoli di FFD e RDF all'interno della rete, consentendo un consumo più uniforme.

Flusso di controllo

Le modalità di associazione, di invio dei messaggi e di selezione del percorso a cammino multiplo sono uguali al protocollo Simil-MCSP e indicate nella sezione 4.4.1.

Rispetto a tale implementazione, come detto, il dispositivo deve essere in grado di rilevare se deve effettuare sensing o se deve effettuare anche funzioni di inoltro.

Il controllo sul sensing del nodo è possibile attraverso la lettura della configurazione iniziale del dispositivo (sez. 4.7), mentre l'attivazione delle funzioni di inoltro è controllabile determinando se al dispositivo sono associati dei figli. In caso positivo il nodo rimarrà acceso per tutta la durata dello slot di trasmissione, mentre in caso negativo invierà il proprio messaggio nel relativo slot e ritornerà in sleep.

Novità

A differenza del protocollo LEACH classico non è il PAN Coordinator che effettua la scelta dei vari cluster-head ma sono i vari nodi che scelgono autonomamente il coordinatore migliore. Questo meccanismo riduce l'overhead causato dallo scambio di messaggi tra PAN Coordinator e cluster-head.

La definizione della funzione costo, così come nel Simil-MCSP, si realizza molto semplicemente attraverso il calcolo di una combinazione lineare di fattori, il che comporta una notevole facilità di costruzione dell'algoritmo di routing e di calcolo del next hop.

4.4.3 Protocollo di Flooding

Il protocollo di flooding prevede l'inoltro in broadcast di tutti i messaggi ricevuti. I messaggi inviati dal sink vengono disseminati nella rete come nei

protocolli precedenti. A differenza di questi, i messaggi inviati dai nodi non seguono alcun instradamento preferito ma vengono inoltrati in broadcast fino al raggiungimento del sink. Il lato negativo di un protocollo così semplice è indicato dai consumi elevati, in quanto ogni nodo rimane acceso per tutta la durata dello slot time.

4.5 Invio dati al server

L'invio dei dati verso il server viene realizzato attraverso il dispositivo UART del microcontrollore. Il trasferimento dei dati verso l'UART è a cura del DMA, anch'esso integrato.

Nello sviluppo di questa comunicazione sono stati riscontrati diversi problemi. Il primo è rappresentato dalla dimensione del buffer di memoria del DMA, che consente lo spostamento di massimo 256 byte. Nel caso sia necessario inviare messaggi di dimensioni maggiori, questi andranno perciò frazionati. Un altro problema è rappresentato dalla possibile interruzione della comunicazione con il server. In questa condizione il convertitore TCP/IP consente comunque di immagazzinare una quantità di dati pari a 10KB. Si deve sviluppare un metodo che rilevi le possibili interruzioni nella comunicazione e interrompa l'invio dei dati verso il convertitore TCP/IP superato il limite dei 10KB. La comunicazione potrà riprendere quando verrà notificata la riattivazione della stessa attraverso la ricezione di un messaggio di risposta o di un byte di keep alive.

Un'ultima considerazione necessaria è dovuta alla possibile presenza di pacchetti flottanti nella rete, che possono giungere al destinatario in ritardo. Si provvede quindi alla sincronizzazione tra PAN Coordinator e PC Server inserendo nell'header del messaggio TCP/IP due byte contenenti il Sequence Number. Il SeqN viene inviato dal PAN Coordinator al Server e la risposta di quest'ultimo deve contenere lo stesso SeqN. Un messaggio contenente SeqN minore viene considerato più vecchio e quindi viene scartato. Una volta completata la trasmissione si incrementa il valore del SeqN.

4.6 Gestione rete da remoto

Il nodo sink comunica attraverso il protocollo TCP/IP con un PC Server per l'invio periodico dei dati e la visualizzazione di questi ultimi attraverso un'interfaccia grafica. Il PC Server però può anche svolgere una funzione di controllo della rete, trasmettendo messaggi al sink. Nel protocollo in esame, ad esempio, è possibile effettuare un riavvio della rete comandato da remoto. Il nodo sink provvederà nel successivo slot di trasmissione ad inviare un messaggio di reset a tutti i nodi e sarà quindi possibile riavviare la rete. Sfruttando questa tecnica è possibile inviare messaggi di vario tipo ai nodi. Ad esempio, è possibile anche la riprogrammazione dei nodi da remoto, la richiesta di determinati flag per l'analisi del corretto funzionamento e altre possibili applicazioni.

4.7 Configurazione nodi rete

La configurazione iniziale della rete prevede l'assegnazione di un identificativo univoco (ID) per ogni nodo della dimensione di due byte. Tale identificativo è necessario prima di tutto per il riconoscimento del nodo ed in secondo luogo per la temporizzazione della trama TDM.

Il nodo con ID = 0x0001 sarà infatti il primo a trasmettere, il nodo 0x0002 trasmetterà per secondo, con un ritardo di 100 ms e così di seguito. Anche l'aggiornamento delle tabelle di routing, che richiede un tempo di 1,5 secondi, sarà regolamentato nello stesso modo.

La configurazione iniziale prevede inoltre l'impostazione del numero di sensori da campionare e/o di attuatori da controllare. Il nodo TheGuardian infatti supporta, oltre ai due ingressi che portano ai convertitori A/D di precisione, altre porte I/O che possono essere utilizzate come ingressi analogici oppure come uscite digitali.

I due byte di identificativo e il byte indicante la configurazione dei sensori devono essere memorizzati nella flash del dispositivo. Utilizzando il programma SmartRF Flash Programmer è possibile accedere velocemente a 16 byte della memoria che saranno utilizzati come byte di configurazione.

Nella figura 4.4 è mostrato un esempio di configurazione dove si programma il nodo con ID=1 e un numero di sensori collegati pari 5.

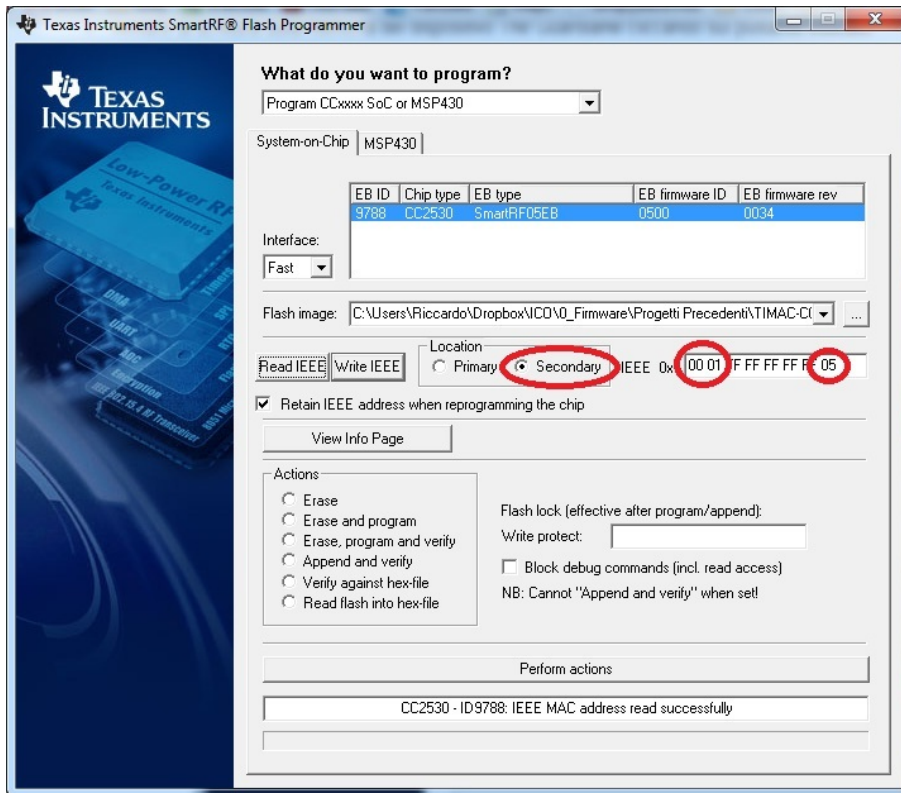


Figura 4.4: Configurazione nodi

Capitolo 5

Implementazione Server e Software Grafico

Il server di ricezione è una componente fondamentale per una rete WSN in quanto consente di salvare e quindi immagazzinare tutti i dati provenienti che altrimenti, a causa della ridotta capacità di memorizzazione dei dispositivi che compongono la rete, andrebbero persi.

L'interfaccia grafica realizzata consente la lettura dello storico dei dati immagazzinati e la rappresentazione di essi in grafici al fine di favorire l'analisi dell'andamento della tensione ai capi delle batterie dei nodi e trarre quindi conclusioni sul lifetime della rete.

Il software elaborato, per quanto non sia complesso, sfrutta l'architettura multi-thread, ovvero implementa tre diversi thread paralleli, uno per la lettura dati dalla Socket, uno per la lettura da terminale di comandi che possono essere inviati alla rete e uno per l'aggiornamento periodico dei grafici.

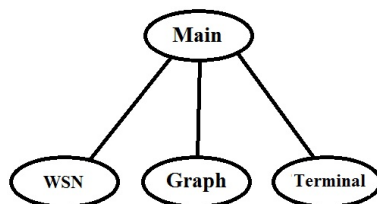


Figura 5.1: Struttura multi-thread del software

5.1 Server TCP/IP

Il server è realizzato utilizzando le librerie che Java fornisce a supporto della comunicazione su rete. Si realizza quindi un thread che gestisca una `ServerSocket` sulla porta 10000, scelta casualmente, e analizzi i dati in ingresso. Il byte indicante il tipo di messaggio consente di individuarne il contenuto e le operazioni da eseguire:

- Messaggio di tipo richiesta data e orario: all'avvio il PAN Coordinator esegue una richiesta di questo tipo per sincronizzare i Real Time Clock dei nodi con l'orario attuale fornito dal PC Server.
- Messaggio di tipo dati: il server memorizza i dati in un file di testo e invia, come conferma della ricezione del messaggio, la data attuale in modo da consentire la sincronizzazione temporale. Si esegue un controllo su eventuali pacchetti non consegnati all'interno della rete inserendo due byte indicanti il Sequence Number. Il SeqN viene inviato dal PAN Coordinator e il server non deve fare altro che spedire la risposta con lo stesso SeqN. Un pacchetto ricevuto dal PAN Coordinator con SeqN più vecchio di quello attuale viene ritenuto invalido e quindi si scarta.
- Messaggio di tipo conferma ricezione data: sfruttando un meccanismo simile al 3-Way Handshake il PAN Coordinator conferma al server di aver ricevuto il messaggio contenente la data attuale.

5.2 Terminale di controllo della rete

Il terminale di controllo consente l'invio di messaggi al PAN Coordinator della rete. Al momento sono utilizzati solamente tre comandi:

- Reset Soft. Il coordinatore attende l'istante in cui tutti i nodi siano attivi per eseguire un riavvio della rete.
- Reset Hard. Il coordinatore viene riavviato immediatamente.
- Keep Alive. Viene inviato un byte di Keep Alive per verificare la corretta comunicazione TCP/IP.

5.3 Interfaccia grafica

L'interfaccia grafica rappresenta la parte esteriore del software e prevede l'inserimento di 8 grafici, ognuno rappresentante la carica della batteria del rispettivo nodo. I grafici sono stati realizzati utilizzando le librerie gratuite JFreeChart. L'aggiornamento in tempo reale dei grafici è stato possibile grazie all'implementazione di un thread che viene eseguito periodicamente ogni minuto, analizza il file dove i dati vengono salvati e apporta gli eventuali aggiornamenti ai grafici.

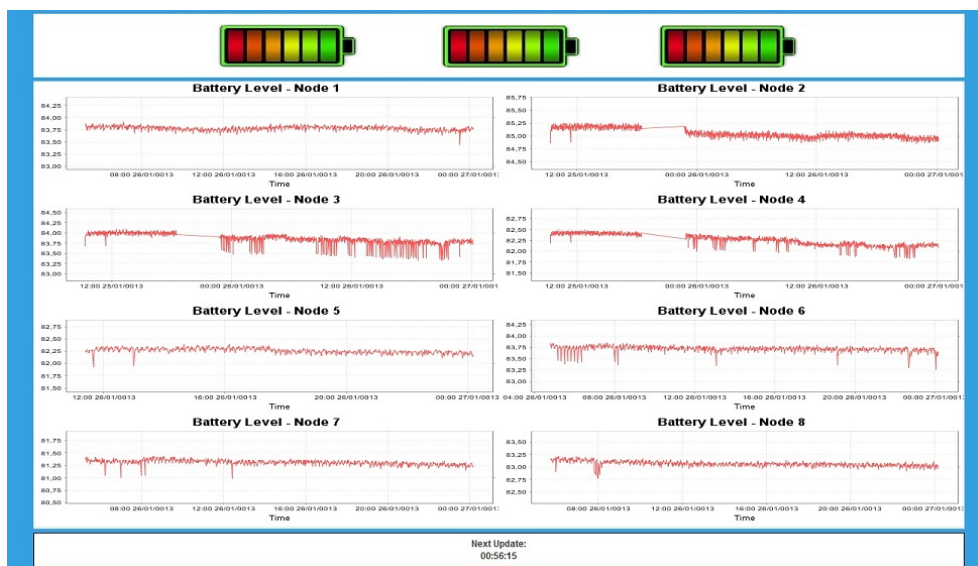
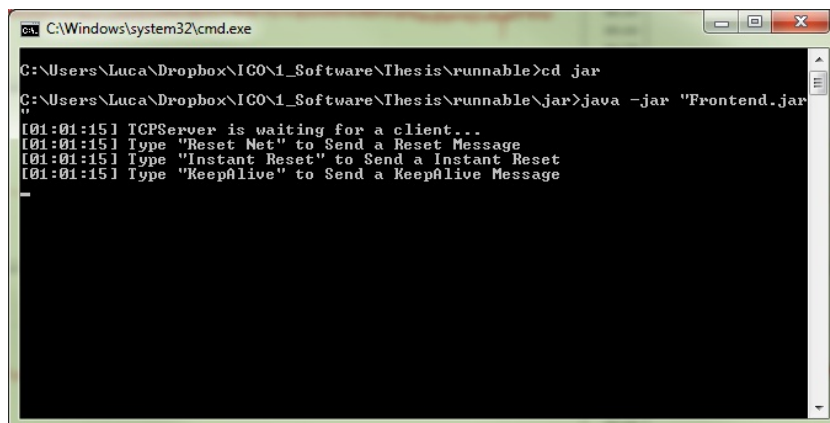
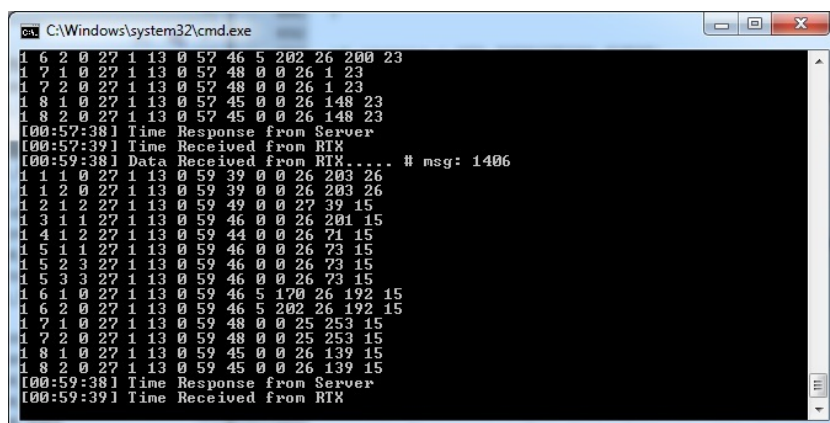


Figura 5.2: Interfaccia grafica



```
C:\Windows\system32\cmd.exe
C:\Users\Luca\Dropbox\ICO\1_Software\Thesis\runnable>cd jar
C:\Users\Luca\Dropbox\ICO\1_Software\Thesis\runnable\jar>java -jar "Frontend.jar"
[01:01:15] TCPServer is waiting for a client...
[01:01:15] Type "Reset Net" to Send a Reset Message
[01:01:15] Type "Instant Reset" to Send a Instant Reset
[01:01:15] Type "KeepAlive" to Send a KeepAlive Message
```

Figura 5.3: Terminale di comando, avvio



```
C:\Windows\system32\cmd.exe
1 6 2 0 27 1 13 0 57 46 5 202 26 200 23
1 7 1 0 27 1 13 0 57 48 0 0 26 1 23
1 7 2 0 27 1 13 0 57 48 0 0 26 1 23
1 8 1 0 27 1 13 0 57 45 0 0 26 148 23
1 8 2 0 27 1 13 0 57 45 0 0 26 148 23
[00:57:38] Time Response from Server
[00:57:39] Time Received from RTX
[00:59:38] Data Received from RTX.... # msg: 1406
1 1 1 0 27 1 13 0 59 39 0 0 26 203 26
1 1 2 0 27 1 13 0 59 39 0 0 26 203 26
1 2 1 2 27 1 13 0 59 49 0 0 27 39 15
1 3 1 1 27 1 13 0 59 46 0 0 26 201 15
1 4 1 2 27 1 13 0 59 44 0 0 26 71 15
1 5 1 1 27 1 13 0 59 46 0 0 26 73 15
1 5 2 3 27 1 13 0 59 46 0 0 26 73 15
1 5 3 3 27 1 13 0 59 46 0 0 26 73 15
1 6 1 0 27 1 13 0 59 46 5 170 26 192 15
1 6 2 0 27 1 13 0 59 46 5 202 26 192 15
1 7 1 0 27 1 13 0 59 48 0 0 25 253 15
1 7 2 0 27 1 13 0 59 48 0 0 25 253 15
1 8 1 0 27 1 13 0 59 45 0 0 26 139 15
1 8 2 0 27 1 13 0 59 45 0 0 26 139 15
[00:59:38] Time Response from Server
[00:59:39] Time Received from RTX
```

Figura 5.4: Terminale di comando, ricezione dati

Capitolo 6

Risultati sperimentali

Nella sezione 3.3 è stato descritto il sistema elaborato e gli obiettivi che si vogliono raggiungere.

In questa sezione inizialmente si analizzano i consumi dei nodi con i tre diversi protocolli e si fornisce una panoramica su quali dovrebbero essere i lifetime attesi.

In seguito si rappresentano i test eseguiti analizzando l'andamento della tensione della batteria. Purtroppo non è stato possibile effettuare test di lunga durata, perciò si analizzerà, per ogni protocollo e per ogni ambiente, l'andamento dei test per una settimana e il lifetime verrà stimato in base alla pendenza della retta che si otterrà attraverso la regressione lineare dei punti ricavati. La scarica della tensione ai capi della batteria nel tempo si può approssimare come lineare, come si può vedere in figura 5 del documento [17].

6.1 Risultati attesi

Il controllo di flusso ideato è molto semplice. Le operazioni che i nodi devono svolgere sono principalmente due, sensing e inoltro dati.

Nell'architettura prevista tutti i dispositivi realizzano l'operazione di sensing e inviano il messaggio contenente lo stato della batteria. Nel caso in cui un nodo debba eseguire forwarding rimane acceso per tutta la durata dello

slot temporale per consentire l'inoltro di messaggi verso il sink. Una volta eseguiti i vari compiti, i nodi entrano in modalità di risparmio energetico.

L'analisi dei consumi ci consente di ricavare un consumo medio per ogni ciclo della durata di 5 minuti, nelle varie fasi di invio dati, tempi di guardia, forwarding, sleep o aggiornamento dei percorsi di routing.

L'aggiornamento dei percorsi di routing è previsto ogni 30 minuti e in tale occasione non sarà necessario effettuare la trasmissione dei dati verso il sink. Realizzando un ciclo ogni 5 minuti e volendo aggiornare le tabelle ogni 30 minuti, otterremo un comportamento periodico ogni 6 cicli.

Attraverso l'analisi del comportamento periodico si potrà definire un consumo medio nel tempo. Il consumo sarà poi confrontabile con i dati sperimentali, per verificare se è attendibile.

| CICLO SEND | | | Consumo medio (mA) | CICLO ROUTING (RFD) | | | Consumo medio (mA) |
|--------------------|-----------|--------------|--------------------|--|-----------|--------------|--------------------|
| Action | Time (ms) | Current (mA) | | Action | Time (ms) | Current (mA) | |
| TG | 100 | 27,28 | 0,97677 | route | 1500 | 27,28 | 1,0518 |
| read | 200 | 46,54 | | sleep | 298500 | 0,92 | |
| send | 100 | 27,28 | | CICLO ROUTING (FFD) | | | |
| TG | 100 | 27,28 | | route | 12000 | 27,28 | |
| sleep | 299500 | 0,92 | | sleep | 288000 | 0,92 | 1,9744 |
| CICLO FORWARD/SEND | | | Consumo medio (mA) | CICLO SLEEP | | | Consumo medio (mA) |
| Action | Time (ms) | Current (mA) | | Action | Time (ms) | Current (mA) | |
| TG | 100 | 27,28 | 1,07343 | TG | 100 | 27,28 | Consumo medio (mA) |
| read | 200 | 46,54 | | sleep | 299900 | 0,92 | |
| forwar | 1300 | 27,28 | | CICLO SEND <td rowspan="2">Consumo medio (mA)</td> | | | Consumo medio (mA) |
| sleep | 298400 | 0,92 | | TG | 100 | 27,28 | |
| | | | | read | 200 | 46,54 | 0,97677 |
| | | | | send | 100 | 27,28 | |
| | | | | TG | 100 | 27,28 | |
| | | | | sleep | 299500 | 0,92 | |
| | | | | sleep | 298400 | 0,92 | |

Tabella 6.1: Consumo medio per ciclo, Protocollo Simil-MCSP

| CICLO SEND | | | Consumo medio (mA) | CICLO FORWARD/SEND | | | Consumo medio (mA) |
|------------|-----------|--------------|--------------------|---------------------|-----------|--------------|--------------------|
| Action | Time (ms) | Current (mA) | | Action | Time (ms) | Current (mA) | |
| TG | 100 | 27,28 | 0,97677 | TG | 100 | 27,28 | Consumo medio (mA) |
| read | 200 | 46,54 | | read | 200 | 46,54 | |
| send | 100 | 27,28 | | forwar | 1300 | 27,28 | 1,07343 |
| TG | 100 | 27,28 | | sleep | 298400 | 0,92 | |
| sleep | 299500 | 0,92 | | CICLO ROUTING (FFD) | | | |
| | | | route | 12000 | 27,28 | | |
| | | | sleep | 288000 | 0,92 | 1,9744 | |
| | | | | CICLO SLEEP | | | Consumo medio (mA) |
| | | | | TG | 100 | 27,28 | |
| | | | | sleep | 299900 | 0,92 | 0,92879 |

Tabella 6.2: Consumo medio per ciclo, Protocollo Simil-LEACH

| CICLO FORWARD/SEND | | | Consumo medio (mA) | CICLO ROUTING (FFD) | | | Consumo medio (mA) |
|--------------------|-----------|--------------|--------------------|--|-----------|--------------|--------------------|
| Action | Time (ms) | Current (mA) | | Action | Time (ms) | Current (mA) | |
| TG | 100 | 27,28 | 1,07343 | route | 12000 | 27,28 | Consumo medio (mA) |
| read | 200 | 46,54 | | sleep | 288000 | 0,92 | |
| forward | 1300 | 27,28 | | CICLO SEND <td rowspan="2">Consumo medio (mA)</td> | | | |
| sleep | 298400 | 0,92 | | TG | 100 | 27,28 | |
| | | | | | read | 200 | 46,54 |
| | | | | send | 100 | 27,28 | |
| | | | | sleep | 299500 | 0,92 | |

Tabella 6.3: Consumo medio per ciclo, Protocollo Flooding

6.1.1 Scenario A

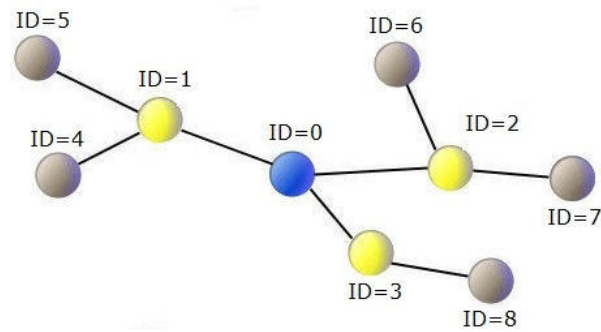


Figura 6.1: Scenario A

Lo scenario descritto rappresenta una situazione statica dove il routing non è sfruttato nelle sue potenzialità perchè i nodi figli possono comunicare con il sink solamente con un nodo intermedio.

| FASE | | Minuto | | | | | | Consumo medio / 30min |
|------|---|--------|-----|-----|-----|-----|----|-----------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 2 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 4 | S | S | S | S | S | R1 | 0,9895 |
| | 5 | S | S | S | S | S | R1 | 0,9895 |
| | 6 | S | S | S | S | S | R1 | 0,9895 |
| | 7 | S | S | S | S | S | R1 | 0,9895 |
| | 8 | S | S | S | S | S | R1 | 0,9895 |

| FASE | Consumo medio (mA) |
|----------------------|--------------------|
| F/S Forward/Send | 1,073 |
| S Send | 0,977 |
| R1 1 node Routing | 1,052 |
| R2 All nodes Routing | 1,974 |

Tabella 6.4: Consumo medio per periodo, Scenario A, Simil-MCSP

| FASE | | Minuto | | | | | | Consumo medio/30min |
|------|---|--------|-----|-----|-----|-----|----|---------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 2 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 4 | S | S | S | S | S | R2 | 1,1432 |
| | 5 | S | S | S | S | S | R2 | 1,1432 |
| | 6 | S | S | S | S | S | R2 | 1,1432 |
| | 7 | S | S | S | S | S | R2 | 1,1432 |
| | 8 | S | S | S | S | S | R2 | 1,1432 |

| FASE | | Consumo medio (mA) |
|------|-------------------|--------------------|
| F/S | Forward/Send | 1,073 |
| S | Send | 0,977 |
| R1 | 1 node Routing | 1,052 |
| R2 | All nodes Routing | 1,974 |

Tabella 6.5: Consumo medio per periodo, Scenario A, Simil-LEACH

| FASE | | Minuto | | | | | | Consumo medio/30min |
|------|---|--------|-----|-----|-----|-----|----|---------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 2 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 4 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 5 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 6 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 7 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 8 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |

| FASE | | Consumo medio (mA) |
|------|-------------------|--------------------|
| F/S | Forward/Send | 1,073 |
| S | Send | 0,977 |
| R1 | 1 node Routing | 1,052 |
| R2 | All nodes Routing | 1,974 |

Tabella 6.6: Consumo medio per periodo, Scenario A, Flooding

Si prevede il consumo medio più elevato nel protocollo di Flooding, dove i nodi ritrasmettono tutti i messaggi in broadcast e rimangono tutti attivi per tutta la durata dello slot temporale.

In questo particolare scenario, i nodi RFD possono comunicare con il sink attraverso un solo FFD, i protocolli Simil-MCSP e Simil-Leach assumono comportamenti molto simili.

In entrambi i protocolli tutti i messaggi provenienti dai nodi RFD devono transitare per i dispositivi FFD, quindi saranno questi ultimi i primi a scaricarsi.

6.1.2 Scenario B

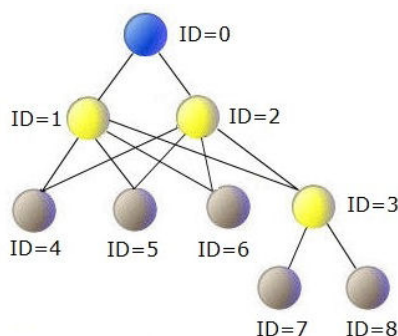


Figura 6.2: Scenario B

In questo scenario la presenza di più coordinatori allo stesso livello consente ai nodi figli di scegliere quale coordinatore ha il percorso di costo minore e associarsi ad esso.

Mentre il coordinatore scelto dai nodi rimane acceso per garantire l'inoltro dei messaggi verso il sink, gli altri coordinatori posti allo stesso livello potranno entrare in sleep, riducendo notevolmente i consumi.

Quando il nodo scelto si scarica e nella rete vengono rilevati coordinatori con più energia, l'aggiornamento dei percorsi di routing causerà l'associazione e l'assegnazione del compito di forwarding a questi ultimi, consentendo al nodo scarico di entrare in sleep. Questa tecnica consente di uniformare i consumi dei nodi all'interno della rete.

| FASE | | Minuto | | | | | | Consumo medio/30min | Distr. uniforme dei consumi |
|------|---|--------|-----|-----|-----|-----|----|---------------------|-----------------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 | 1,1832 |
| | 2 | S | S | S | S | S | R2 | 1,1432 | |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 | |
| | 4 | S | S | S | S | S | R1 | 0,9895 | |
| | 5 | S | S | S | S | S | R1 | 0,9895 | |
| | 6 | S | S | S | S | S | R1 | 0,9895 | |
| | 7 | S | S | S | S | S | R1 | 0,9895 | |
| | 8 | S | S | S | S | S | R1 | 0,9895 | |

| FASE | | Consumo medio (mA) |
|------|-------------------|--------------------|
| F/S | Forward/Send | 1,073 |
| S | Send | 0,977 |
| R1 | 1 node Routing | 1,052 |
| R2 | All nodes Routing | 1,974 |

Tabella 6.7: Consumo medio per periodo, Scenario B, Simil-MCSP

Attraverso l'analisi si evidenzia il problema principale del protocollo Simil-MCSP. I ruoli dei dispositivi all'interno della rete infatti sono statici, quindi i nodi RFD a livello 2 non possono essere utilizzati dai nodi a livello 3 per realizzare inoltri, ma questi ultimi si devono servire solamente del FFD posto nel loro range di copertura, ovvero il numero 3. Quest'ultimo quindi sarà il primo a scaricarsi.

Un protocollo che consenta ai dispositivi di selezionare il ruolo, RFD o FFD, in base alle necessità migliorerebbe ancora le prestazioni.

| FASE | | Minuto | | | | | | Consumo medio/30min | Distr. uniforme dei consumi |
|------|---|--------|-----|-----|-----|-----|----|---------------------|-----------------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 | 1,1832 |
| | 2 | S | S | S | S | S | R2 | 1,1432 | |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 | |
| | 4 | S | S | S | S | S | R2 | 1,1432 | |
| | 5 | S | S | S | S | S | R2 | 1,1432 | |
| | 6 | S | S | S | S | S | R2 | 1,1432 | |
| | 7 | S | S | S | S | S | R2 | 1,1432 | |
| | 8 | S | S | S | S | S | R2 | 1,1432 | |

| FASE | | Consumo medio (mA) |
|------|-------------------|--------------------|
| F/S | Forward/Send | 1,073 |
| S | Send | 0,977 |
| R1 | 1 node Routing | 1,052 |
| R2 | All nodes Routing | 1,974 |

Tabella 6.8: Consumo medio per periodo, Scenario B, Simil-LEACH

Grazie al protocollo Simil-Leach risolviamo il problema descritto sopra. Dai dati possiamo dedurre come i consumi dei dispositivi siano più uniformi.

| FASE | | Minuto | | | | | | Consumo medio/30min |
|------|---|--------|-----|-----|-----|-----|----|---------------------|
| | | 5 | 10 | 15 | 20 | 25 | 30 | |
| ID | 1 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 2 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 3 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 4 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 5 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 6 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 7 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |
| | 8 | F/S | F/S | F/S | F/S | F/S | R2 | 1,2232 |

| FASE | | Consumo medio (mA) |
|------|-------------------|--------------------|
| F/S | Forward/Send | 1,073 |
| S | Send | 0,977 |
| R1 | 1 node Routing | 1,052 |
| R2 | All nodes Routing | 1,974 |

Tabella 6.9: Consumo medio per periodo, Scenario B, Flooding

Il protocollo di flooding prevede, come nello scenario A, un consumo uniforme di tutti i nodi, fino allo spegnimento completo della rete.

6.1.3 Lifetime stimato

In base al consumo medio, per ogni tipologia di nodo si analizza il possibile tempo di vita in un'architettura con alimentazione a batteria al piombo [6 V, 4.4 Ah]. Il ridottissimo consumo del dispositivo, che non richiede correnti di spunto elevate, consente di sfruttare la batteria al piombo fino alla tensione di scarica, valutata in 5.2 V, tenendo in considerazione però il rischio di danneggiare la batteria.

$$h = \text{Capacità batteria} / \text{Consumo medio}$$

| ANALISI DURATA EFFETTIVA BATTERIA | | | |
|-----------------------------------|-------------------------|----------|----|
| TheGuardian | Corrente assorbita | 0,9895 | mA |
| | Tensione della batteria | 6 | V |
| | Capacità della batteria | 4,4 | Ah |
| | Potenza del carico | 0,005937 | W |
| | Energia immagazzinata | 26,4 | Wh |
| | Ore utilizzo teoriche | 4446,69 | h |
| | Giorni utilizzo teorici | 185,28 | gg |
| | Mesi utilizzo teorici | 6,18 | mm |
| | Anni utilizzo teorici | 0,51 | h |




Tabella 6.10: Lifetime stimato per un nodo RFD, MCSP

| ANALISI DURATA EFFETTIVA BATTERIA | | | |
|-----------------------------------|-------------------------|-------------|----|
| TheGuardian | Corrente assorbita | 1,143166667 | mA |
| | Tensione della batteria | 6 | V |
| | Capacità della batteria | 4,4 | Ah |
| | Potenza del carico | 0,006859 | W |
| | Energia immagazzinata | 26,4 | Wh |
| | Ore utilizzo teoriche | 3848,96 | h |
| | Giorni utilizzo teorici | 160,37 | gg |
| | Mesi utilizzo teorici | 5,35 | mm |
| | Anni utilizzo teorici | 0,45 | h |




Tabella 6.11: Lifetime stimato per un nodo RFD, LEACH

| ANALISI DURATA EFFETTIVA BATTERIA | | | |
|-----------------------------------|-------------------------|-------------|----|
| TheGuardian | Corrente assorbita | 1,223166667 | mA |
| | Tensione della batteria | 6 | V |
| | Capacità della batteria | 4,4 | Ah |
| | Potenza del carico | 0,007339 | W |
| | Energia immagazzinata | 26,4 | Wh |
| | Ore utilizzo teoriche | 3597,22 | h |
| | Giorni utilizzo teorici | 149,88 | gg |
| | Mesi utilizzo teorici | 5,00 | mm |
| | Anni utilizzo teorici | 0,42 | h |




Tabella 6.12: Lifetime stimato per un nodo FFD in ambiente con 1 nodo FFD

| ANALISI DURATA EFFETTIVA BATTERIA | | | |
|-----------------------------------|-------------------------|-------------|----|
| TheGuardian | Corrente assorbita | 1,183166667 | mA |
| | Tensione della batteria | 6 | V |
| | Capacità della batteria | 4,4 | Ah |
| | Potenza del carico | 0,007099 | W |
| | Energia immagazzinata | 26,4 | Wh |
| | Ore utilizzo teoriche | 3718,83 | h |
| | Giorni utilizzo teorici | 154,95 | gg |
| | Mesi utilizzo teorici | 5,17 | mm |
| | Anni utilizzo teorici | 0,43 | h |




Tabella 6.13: Lifetime stimato per un nodo FFD in ambiente con 2 nodi FFD

| ANALISI DURATA EFFETTIVA BATTERIA | | | |
|-----------------------------------|-------------------------|-------------|----|
| TheGuardian | Corrente assorbita | 1,163166667 | mA |
| | Tensione della batteria | 6 | V |
| | Capacità della batteria | 4,4 | Ah |
| | Potenza del carico | 0,006979 | W |
| | Energia immagazzinata | 26,4 | Wh |
| | Ore utilizzo teoriche | 3782,78 | h |
| | Giorni utilizzo teorici | 157,62 | gg |
| | Mesi utilizzo teorici | 5,25 | mm |
| | Anni utilizzo teorici | 0,44 | h |




Tabella 6.14: Lifetime stimato per un nodo FFD in ambiente con 4 nodi FFD

6.2 Risultati ottenuti

Per analizzare l'effettivo lifetime della rete si è scelto di esaminare l'andamento della tensione ai capi delle batterie.

I primi test sono stati effettuati rilevando la tensione manualmente attraverso l'utilizzo di un multimetro in configurazione di voltmetro.

Una volta implementato il software necessario alla ricezione dei dati per il PC Server si è preferito lasciare il compito delle rilevazioni al convertitore A/D del microcontrollore e trattare semplicemente i dati ricevuti attraverso analisi statistiche.

In una settimana di test infatti, considerando che l'invio dello stato delle batterie dei nodi avviene ogni 5 minuti, si potranno ottenere circa 3000 campioni per ogni nodo.

Supportati da software come Matlab o Excel si potranno elaborare questi campioni al fine di realizzare una regressione lineare, calcolare la pendenza della retta ricavata e prevedere una stima dell'andamento della tensione nel tempo.

| node | date | time | batt_hex | batt_norm | volt | timestamp | | |
|------|------------|----------|----------|-----------|----------|-----------|----------|----------|
| | | | | | | | m | q |
| 1 | 23/01/2013 | 10:15:08 | 6917 | 0,844 | 6,130056 | 41297,427 | -0,01021 | 427,5699 |
| 1 | 23/01/2013 | 10:15:08 | 6917 | 0,844 | 6,130056 | 41297,427 | | |
| 1 | 23/01/2013 | 10:20:08 | 6919 | 0,845 | 6,131829 | 41297,431 | | |
| 1 | 23/01/2013 | 10:20:08 | 6919 | 0,845 | 6,131829 | 41297,431 | | |
| 1 | 23/01/2013 | 10:25:08 | 6918 | 0,844 | 6,130942 | 41297,434 | | |
| 1 | 23/01/2013 | 10:25:08 | 6918 | 0,844 | 6,130942 | 41297,434 | | |
| 1 | 23/01/2013 | 10:30:08 | 6917 | 0,844 | 6,130056 | 41297,438 | | |
| 1 | 23/01/2013 | 10:30:08 | 6917 | 0,844 | 6,130056 | 41297,438 | | |
| 1 | 23/01/2013 | 10:30:08 | 6917 | 0,844 | 6,130056 | 41297,438 | | |

Tabella 6.15: Regressione lineare per la stima dei coefficienti della retta di previsione

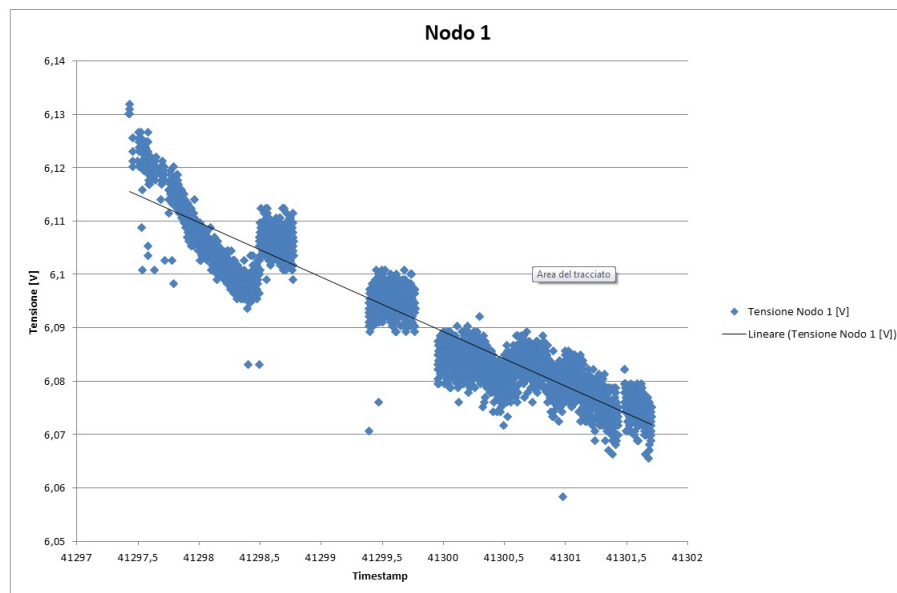


Figura 6.3: Grafico di dispersione dei valori di tensione rilevati in una settimana

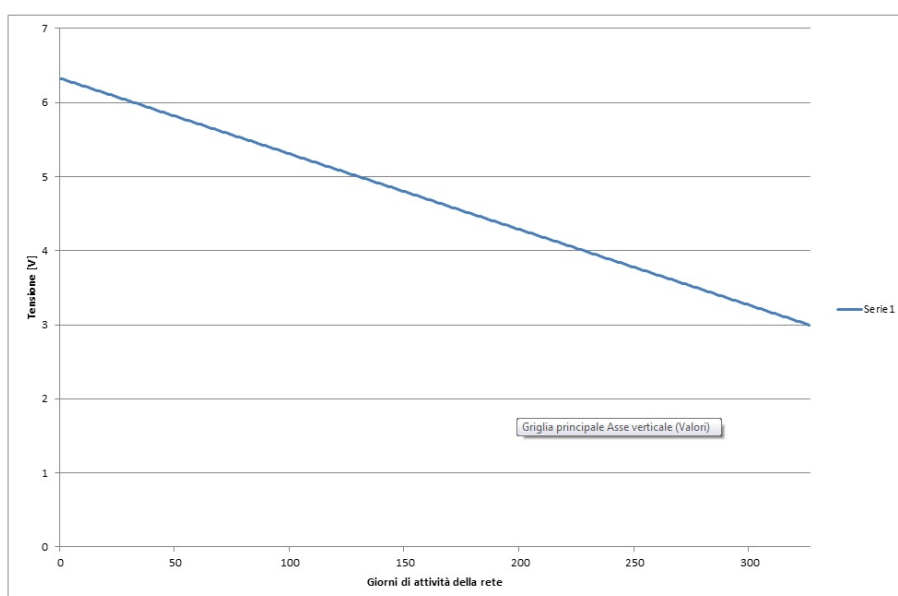


Figura 6.4: Grafico dei valori di tensione previsti per i prossimi 300 giorni

6.2.1 Scenario A

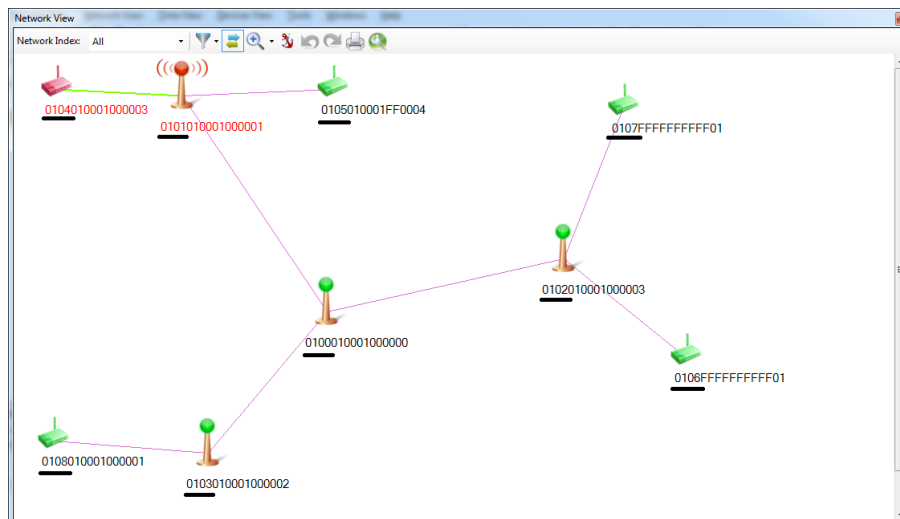


Figura 6.5: Disposizione nodi, Peryton, Scenario A

| ID | | Tensione [V] | | | | Tensione normalizzata (V) | | | | Coefficienti retta | | Angolo (°) |
|----|--|--------------|------|------|------|---------------------------|--------|--------|--------|--------------------|---------|---------------|
| | | Giorno | | | | Giorno | | | | m | q | |
| | | 1 | 3 | 5 | 7 | 1 | 3 | 5 | 7 | | | |
| 1 | | 6,27 | 6,24 | 6,22 | 6,20 | 0,8636 | 0,8599 | 0,8567 | 0,854 | -0,0016 | 0,86499 | -0,09194 |
| 2 | | 6,22 | 6,19 | 6,17 | 6,15 | 0,8567 | 0,8526 | 0,8501 | 0,8477 | -0,00149 | 0,85774 | -0,08523 |
| 3 | | 6,25 | 6,22 | 6,20 | 6,17 | 0,8609 | 0,8567 | 0,8545 | 0,8496 | -0,0018 | 0,86266 | -0,10338 |
| 4 | | 6,29 | 6,28 | 6,27 | 6,25 | 0,8664 | 0,865 | 0,8636 | 0,8609 | -0,0009 | 0,86756 | -0,0513 |
| 5 | | 6,29 | 6,28 | 6,27 | 6,26 | 0,8664 | 0,865 | 0,8636 | 0,8623 | -0,00069 | 0,86708 | -0,03946 |
| 6 | | 6,18 | 6,17 | 6,17 | 6,15 | 0,8512 | 0,8499 | 0,8499 | 0,8471 | -0,00062 | 0,852 | -0,03551 |
| 7 | | 6,28 | 6,27 | 6,26 | 6,26 | 0,865 | 0,8636 | 0,8623 | 0,8621 | -0,00051 | 0,86528 | -0,02908 |
| 8 | | 6,37 | 6,35 | 6,34 | 6,34 | 0,8774 | 0,8747 | 0,8733 | 0,8733 | -0,00069 | 0,87741 | -0,03946 |

Tabella 6.16: Dati rilevati manualmente, Simil-MCSP, scenario A

L'ipotesi dell'andamento dei consumi è confermata dai valori rappresentati nei grafici. Il protocollo Simil-MCSP e il protocollo Simil-LEACH, inseriti in topologie di rete che non consentono l'utilizzo del routing, hanno andamenti simili, mentre nel flooding tutti i nodi si scaricano uniformemente.

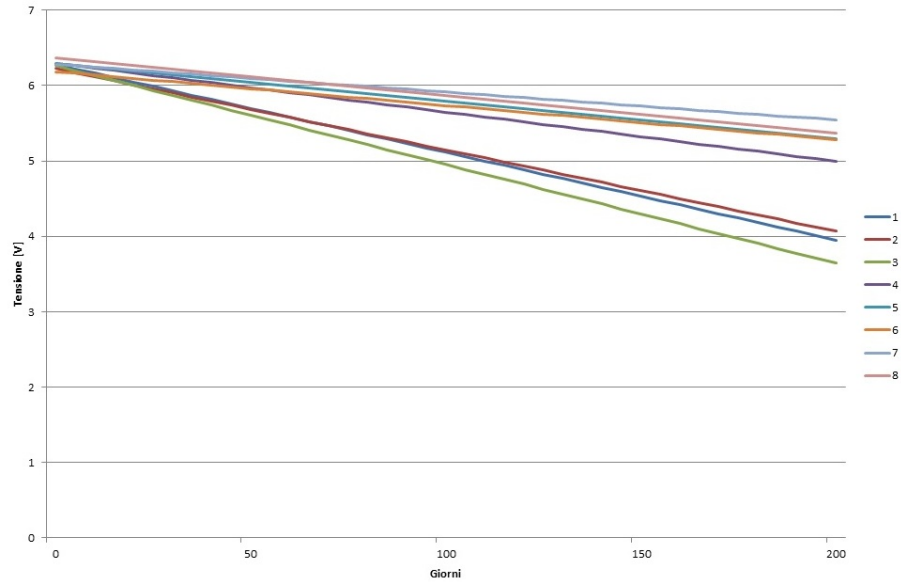


Figura 6.6: Grafico Simil-MCSP, scenario A

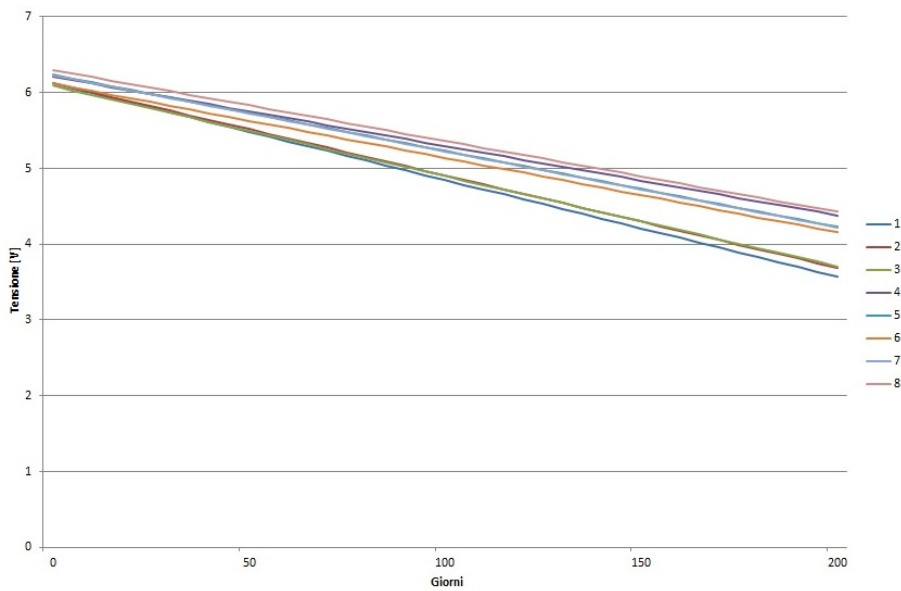


Figura 6.7: Grafico Simil-LEACH, scenario A

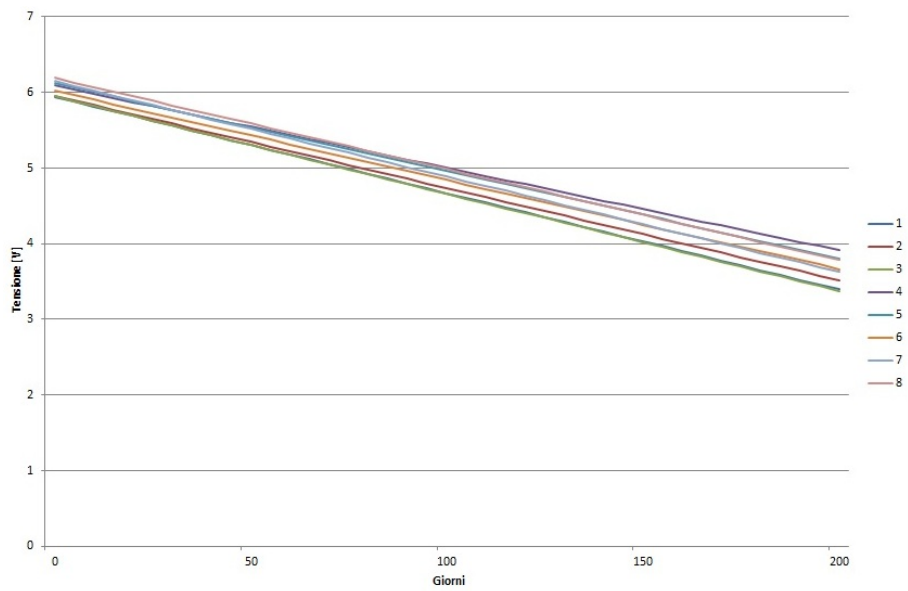


Figura 6.8: Grafico Flooding

6.2.2 Scenario B

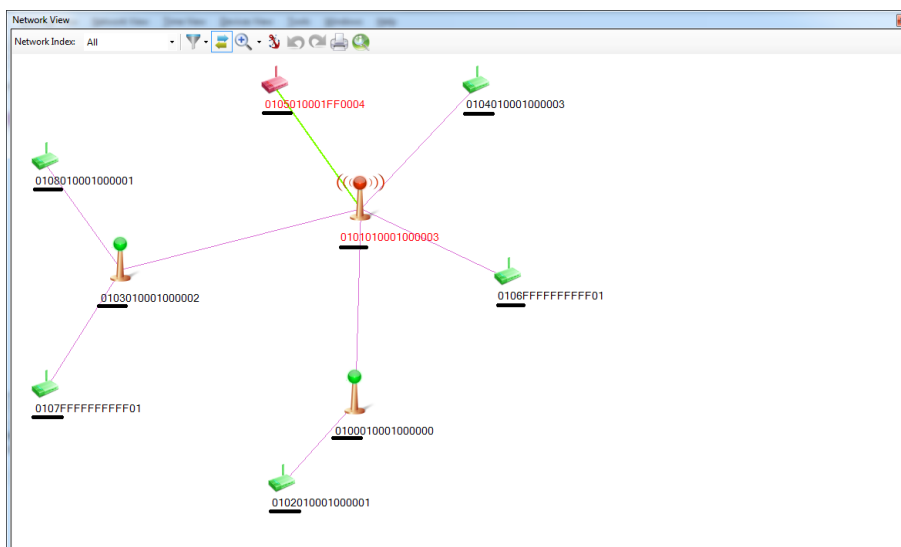


Figura 6.9: Disposizione nodi, Peryton, Scenario B

| ID | Tensione [V] | | | | Tensione normalizzata (V) | | | | Coefficienti retta | | Angolo |
|----|--------------|------|------|------|---------------------------|--------|--------|--------|--------------------|---------|---------|
| | Giorno | | | | Giorno | | | | m | q | (°) |
| | 1 | 3 | 5 | 7 | 1 | 3 | 5 | 7 | | | |
| 1 | 6,20 | 6,17 | 6,14 | 6,13 | 0,854 | 0,8499 | 0,8457 | 0,8442 | -0,0017 | 0,85514 | -0,0959 |
| 2 | 6,15 | 6,15 | 6,14 | 6,13 | 0,8477 | 0,8471 | 0,8457 | 0,8437 | -0,0007 | 0,84871 | -0,0383 |
| 3 | 6,17 | 6,14 | 6,12 | 6,09 | 0,8496 | 0,8457 | 0,8423 | 0,8388 | -0,0018 | 0,85125 | -0,1022 |
| 4 | 6,25 | 6,24 | 6,23 | 6,21 | 0,8609 | 0,859 | 0,8581 | 0,8554 | -0,0009 | 0,8618 | -0,0497 |
| 5 | 6,26 | 6,25 | 6,24 | 6,23 | 0,8623 | 0,8609 | 0,8595 | 0,8587 | -0,0006 | 0,86275 | -0,0347 |
| 6 | 6,15 | 6,14 | 6,12 | 6,12 | 0,8471 | 0,8457 | 0,843 | 0,843 | -0,0008 | 0,84773 | -0,0434 |
| 7 | 6,26 | 6,25 | 6,23 | 6,23 | 0,8621 | 0,8609 | 0,8581 | 0,8581 | -0,0007 | 0,86273 | -0,0419 |
| 8 | 6,34 | 6,32 | 6,31 | 6,30 | 0,8733 | 0,8705 | 0,8691 | 0,8678 | -0,0009 | 0,87376 | -0,0513 |

Tabella 6.17: Dati rilevati manualmente, Simil-MCSP, scenario B

Anche in questo caso l'andamento dei consumi segue quello ipotizzato. L'unica differenza si nota nel protocollo Simil-MCSP. Come si deduce dal grafico 6.10 si ha un consumo maggiore del nodo 1 rispetto al nodo 2. Tale comportamento si spiega perché il nodo 1 inizia il test con una tensione maggiore quindi i dispositivi, legandosi al nodo con maggiore energia, vi si associano mentre il nodo 2, non avendo alcun figlio, non deve inoltrare alcun dato per quasi tutta la durata del test. Solo quando la tensione delle batterie raggiunge lo stesso livello gli algoritmi di routing inizieranno ad

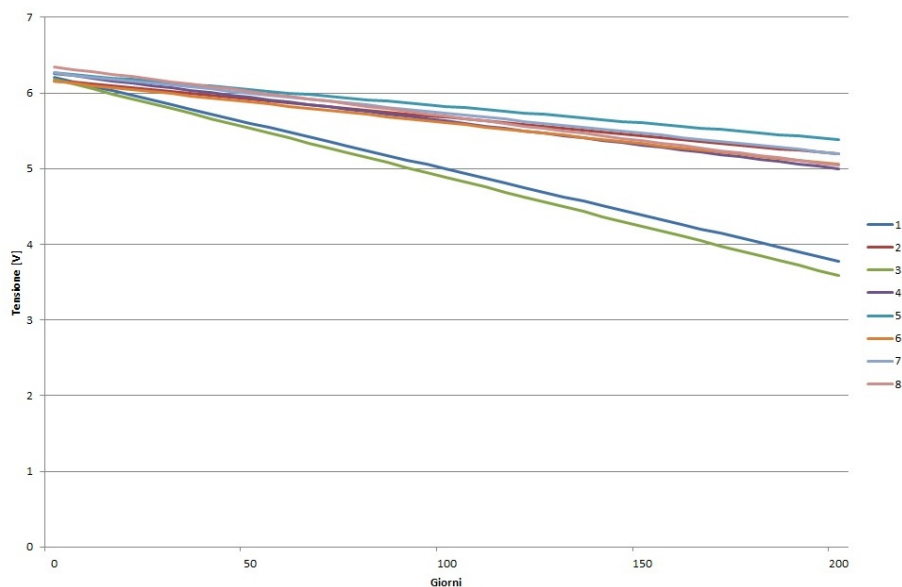


Figura 6.10: Grafico Simil-MCSP, scenario B

uniformare i consumi. Attraverso un'analisi più duratura si potrebbe notare come i consumi dei due nodi si tenderebbero allo stesso valore.

Il protocollo Simil-LEACH segue l'andamento previsto, ove i nodi [1,2] hanno il consumo più elevato, i nodi [3-6] hanno andamenti pressoché uniformi e i nodi [7,8] si comportano da RFD e hanno i consumi minori. Il protocollo di flooding assume lo stesso comportamento nonostante la variazione della topologia della rete e produce sempre una scarica uniforme di tutti i nodi.

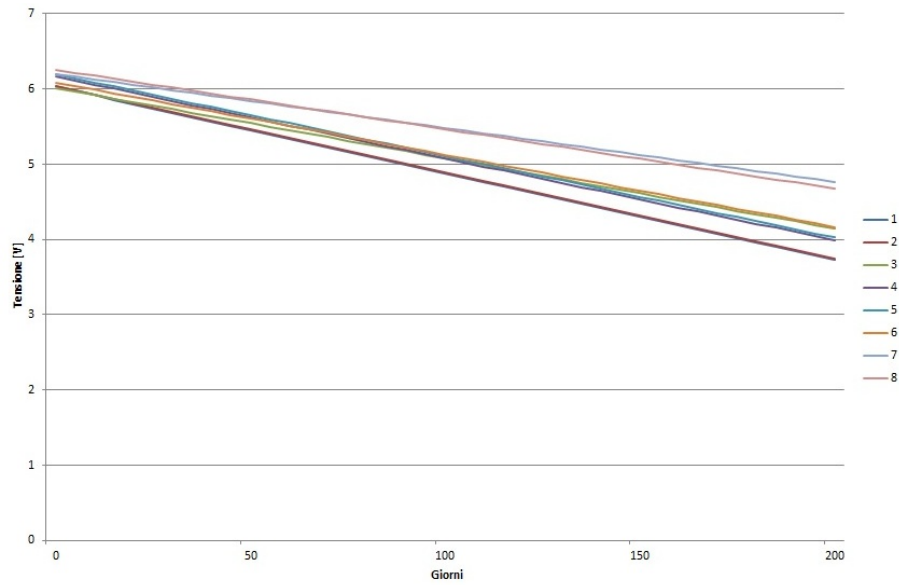


Figura 6.11: Grafico Simil-LEACH, scenario B

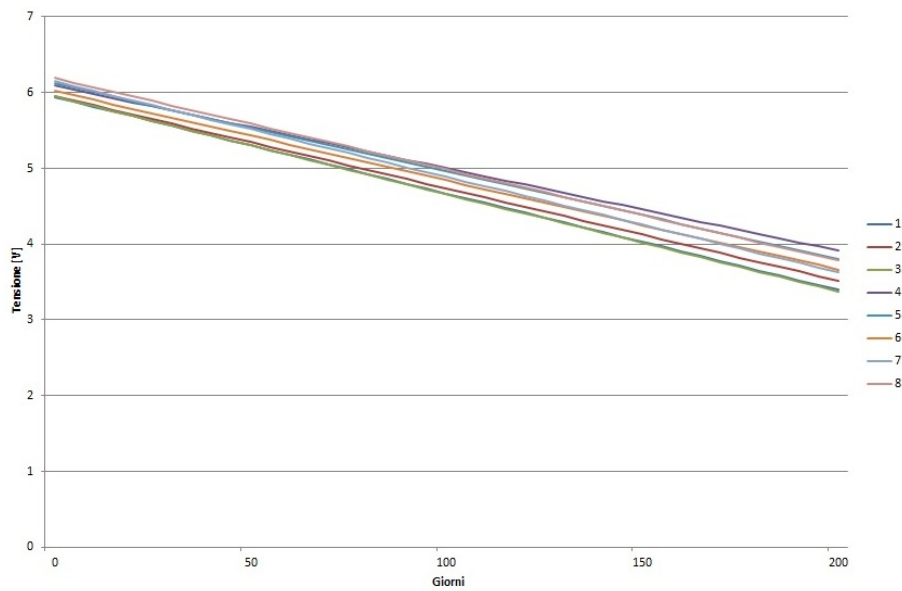


Figura 6.12: Grafico Flooding

6.2.3 Lifetime stimato

In base ai dati a disposizione si effettua un'analisi sulla percentuale di nodi che possono trasmettere dati all'interno della rete. A partire da una tensione di batteria carica (6,3 V) per tutti i nodi, si imposta una tensione al di sotto della quale i nodi vengono considerati spenti, 4.5 V (vedi appendice B) per una batteria al piombo [6V, 4.4Ah], e si analizza lo spegnimento progressivo dei nodi nella rete.

Nell'immagine si può notare come il protocollo di flooding rappresenti il caso peggiore, perché il consumo uniforme della rete porta allo spegnimento immediato di tutti i nodi. Nello scenario A, come detto, i protocolli MCSP e LEACH assumono comportamenti simili. Una volta scaricati i nodi FFD, gli RFD ancora in vita provano a trasmettere dati verso il sink, ma in assenza di un nodo che li inoltri, vengono persi. I nodi potrebbero essere ancora attivi ma non utili alla raccolta dei dati.

Il protocollo LEACH nello scenario B implementa il caso ottimale di routing applicato ad una rete. I nodi si spengono in successione ma la capacità di inoltra è garantita fino a quando i più vicini al sink non abbandonano la rete.

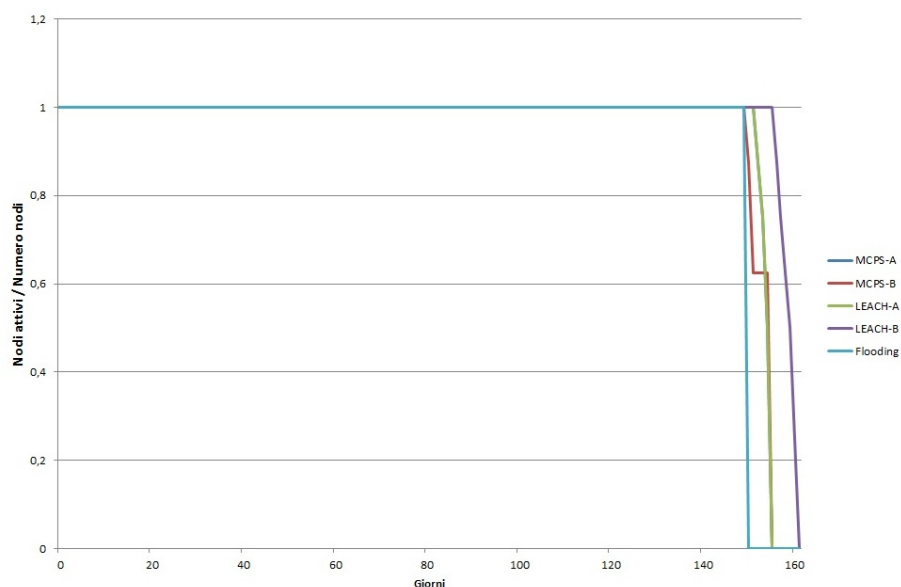


Figura 6.13: Grafico di confronto tra i vari protocolli

Capitolo 7

Conclusioni

Nella trattazione si è cercato di analizzare, riferendosi ai requisiti delle Wireless Sensor Network, un protocollo che garantisse un trade-off ottimo tra robustezza e life-time.

Uno degli obiettivi raggiunti ha riguardato la progettazione del firmware dei nodi, tenendo in considerazione tutte le problematiche che possono presentare un canale wireless ed un'architettura a microcontrollore a basso consumo, ovvero con scarse risorse di memoria e una ridotta potenza in trasmissione.

Per quanto riguarda la robustezza della rete, i test realizzati hanno mostrato un'ottima tolleranza agli imprevisti che si possono presentare nel contesto di questa tipologia di reti. Abbiamo verificato come lo spegnimento di nodi dovuto alla scarica delle batterie o all'uscita dal range di copertura venga riconosciuto correttamente, così come l'inserimento di nuovi nodi all'interno della rete. Lo sviluppo della comunicazione tra rete e server, servendosi del terminale di comando, è stato inoltre fondamentale per impedire blocchi causati dall'interruzione della comunicazione TCP/IP.

I protocolli sviluppati permettono l'estensione della rete anche con dimensioni molto elevate, sfruttando l'invio dei dati in multi-hop.

In particolare la creazione dei vari protocolli di routing ha consentito l'individuazione di varie problematiche in corso d'opera. Tra gli algoritmi sviluppati, il Simil-LEACH garantisce l'adattabilità della rete a qualsiasi ambiente di lavoro, sia attraverso la ricerca del cammino minimo, sia at-

traverso l'analisi della qualità del mezzo trasmissivo e dell'energia del nodo verso il quale sarà effettuata la trasmissione. I test effettuati mostrano che il protocollo garantisce il miglior tempo di vita, grazie alla rotazione del ruolo dei coordinatori che permette di ottenere consumi medi minori rispetto agli altri protocolli.

L'interfaccia grafica realizzata, seppur molto semplice, consente di avere una visualizzazione completa e dinamica dell'andamento della batteria dei nodi, senza la necessità di elaborare di volta in volta i dati acquisiti.

Sviluppi futuri dell'hardware utilizzato potranno consentire la gestione della modalità di "Deep Sleep" del microcontrollore, inserendo un RTC in grado di comandare interrupt, così da ottenere tempi di vita della rete molto più elevati in quanto il consumo in sleep, dai 900 μA attuali, scenderebbe almeno di un ordine di grandezza.

Inoltre, sviluppi futuri del firmware possono consentire la comunicazione all'interno della rete su canali diversi (Time Slotted Channel Hopping), evitando l'interferenza tra i vari nodi. A questo proposito, una versione aggiornata dello standard, IEEE 802.15.4e, basata su questo approccio, è in fase di definizione e potrà essere utilizzata in applicazioni future.

Appendice A

Configurazione TCP Converter

[Administrator Setting](#)
[TCP Mode](#)
[UDP Mode](#)
[UART](#)
[Reset Device](#)

Administrator Setting

| | | | |
|---------------------------------------|--|----------------------------------|---|
| Kernel Version | V1.43 2010/01/21 | | |
| MAC Address | 00:11:22:76:8C:FB | | |
| Nickname | <input type="text" value="NetUART"/> | | |
| IP Setting | | | |
| IP Address | <input type="text" value="192"/> | <input type="text" value="168"/> | <input type="text" value="1"/> . <input type="text" value="20"/> |
| Subnet Mask | <input type="text" value="255"/> | <input type="text" value="255"/> | <input type="text" value="255"/> . <input type="text" value="0"/> |
| Gateway | <input type="text" value="192"/> | <input type="text" value="168"/> | <input type="text" value="1"/> . <input type="text" value="13"/> |
| IP Configure | <input checked="" type="radio"/> Static <input type="radio"/> DHCP | | |
| Password Setting | | | |
| Username | <input type="text" value="admin"/> | max:15 | |
| Password | <input type="password" value="*****"/> | max:15 | |
| Confirm | <input type="password" value="*****"/> | | |
| <input type="button" value="Update"/> | | | |
| Load Default Setting to EEPROM | <input type="button" value="Load"/> | | |

Note:
Nickname only can use "0-9","a-z","A-Z","_","-",""

[Administrator Setting](#)
[TCP Mode](#)
[UDP Mode](#)
[UART](#)
[Reset Device](#)

TCP Control

| Item | Value |
|---------------------------------------|--|
| Telnet Server/Client | <input type="radio"/> Server <input checked="" type="radio"/> Client <input type="radio"/> Disable |
| Port Number | <input type="text" value="10000"/> |
| Remote Server IP Address | <input type="text" value="192"/> . <input type="text" value="168"/> . <input type="text" value="1"/> . <input type="text" value="13"/> |
| Client mode inactive timeout | <input type="text" value="20"/> minute (1~99,0=Disable) |
| Server mode protect timeout | <input type="text" value="60"/> minute (1~98,0=Disable,99=Can't replace) |
| <input type="button" value="Update"/> | |

[Administrator Setting](#)
[TCP Mode](#)
[UDP Mode](#)
[UART](#)
[Reset Device](#)

UDP

| Item | Value |
|---------------------------------------|---|
| Status | <input type="radio"/> Enable <input checked="" type="radio"/> Disable |
| Local Port | <input type="text" value="21"/> |
| Remote Address | IP |
| | <input type="text" value="192"/> . <input type="text" value="168"/> . <input type="text" value="2"/> . <input type="text" value="2"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| | <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| Port | |
| <input type="text" value="21"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="text" value="0"/> | |
| <input type="button" value="Update"/> | |

[Administrator Setting](#)
[TCP Mode](#)
[UDP Mode](#)
[UART](#)
[Reset Device](#)

UART Control

| Item | Setting |
|---------------------------------------|---|
| Mode | <input type="text" value="RS232"/> |
| Baudrate | <input type="text" value="19200"/> |
| Character Bits | <input type="text" value="8"/> |
| Parity Type | <input type="text" value="none"/> |
| Stop Bit | <input type="text" value="1"/> |
| Hardware Flow Control | <input type="text" value="hardware"/> |
| Delimiter | <input type="checkbox"/> Character 1: <input type="text" value="00"/> , <input type="checkbox"/> Character 2: <input type="text" value="FF"/> |
| | <input type="checkbox"/> Silent time: <input type="text" value="5"/> (1~255)*200ms |
| | <input type="checkbox"/> Drop Character |
| <input type="button" value="Update"/> | |

Appendice B

Batteria al piombo

L'analisi del lifetime richiede di stabilire fino a quando le batterie potranno erogare energia. Utilizzando una batteria al piombo [6V, 4.4Ah] dovremo definire la tensione di fine scarica dato un sistema che richiede un assorbimento di correnti nell'ordine dei mA.

La capacità di una batteria (Ah) viene espressa come il prodotto tra la corrente di scarica (A) e il tempo (h) trascorso fino al raggiungimento della tensione finale di scarica. Essa varia in relazione all'intensità della corrente erogata.

La capacità nominale (C) viene convenzionalmente definita alla scarica in 20 h, con una tensione di fine scarica di 1,75V per elemento a una temperatura tra 20/25 ° C. Nel caso di una batteria 12V-7,2Ah, per esempio, la corrente di scarica viene espressa con la formula:

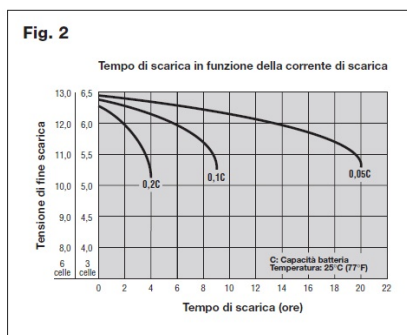
$$0,05C = 0,05 \cdot 7,2 = 0,36A$$

$$2C = 2 \cdot 7,2 = 14,4A$$

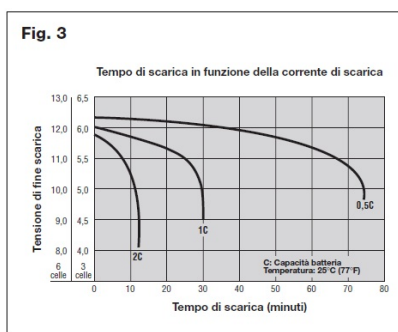
Per effetto della resistenza interna della batteria la tensione scende più rapidamente con correnti di scarica più alte (figure B.1 e B.2). Per evitare di ridurre la vita di una batteria, si raccomanda di non scaricarla oltre le tensioni minime indicate in tabella.

| Corrente di scarica | Tensione di fine scarica |
|---------------------|--------------------------|
| Fino a 0,2 C | 1,75 V/cella |
| 0,2 C - 0,5 C | 1,70 V/cella |
| 0,5 C - 1,0 C | 1,60 V/cella |
| 1,0 C - 2,0 C | 1,50 V/cella |
| 2,0 C - 3,0 C | 1,35 V/cella |
| Sopra 3,0 C | 1,00 V/cella |

Figura B.1: Tensione di fine scarica



(a)



(b)

Figura B.2: Tempo di scarica in funzione della corrente di scarica

Bibliografia

- [1] A. Conti, D. Dardari, G. Mazzini, R. Verdone, “Wireless Sensor and Actuator Networks”, Elsevier, 2008.

- [2] F. De Stefani, “Sistema di monitoraggio ambientale tramite WSN”, Supervisor Prof. Paolo Ettore Gamba, Università degli Studi di Pavia, Facoltà di Ingegneria, 2008.

- [3] M. C. Vuran, I. F. Akyildiz, “Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks”, in IEEE/ACM Transactions on networking, vol. 14, no. 2, pp. 316-329, April 2006.

- [4] J. Yuan, Z. Li, W. Yu, B. Li, “A A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks”, IEEE Journal on Selected Areas in Communications, vol.24, no.11, pp. 2092-2103, November 2006.

- [5] M. Chiang, “Balancing transport and physical Layers in wireless multihop networks: jointly optimal congestion control and power control”, IEEE Journal on Selected Areas in Communications, vol.23, no. 1, pp.104-116, January 2005.

- [6] S. Mohanty, “Energy Efficient Routing Algorithms for Wireless Sensor Networks and Performance Evaluation of Quality of Service for IEEE 802.15.4 Networks”, Supervisor Dr. Sarat Kumar Patra, Department of Electronics & Communication Engineering National Institute of Technology, Rourkela-769008, January 2010

-
- [7] K. Javed, “ZigBee suitability for Wireless Sensor Networks in Logistic Telemetry Applications”, Halmstad University, School of Information Science, Computer and Electrical Engineering (IDE), January 2006
- [8] N. Franceschini, “Sviluppo di un sistema di monitoraggio e controllo ambientale a basso consumo energetico per reti di sensori wireless”, Supervisor Ch.Mo Prof.re Luca Schenato, Università degli Studi di Padova, Dipartimento di ingegneria dell’informazione, December 2009
- [9] J.H. Chang, “Maximum Lifetime Routing in Wireless Sensor Networks”, IEEE/ACM Transactions on Networking, vol.12, no.4, pp.609-619, August 2004
- [10] R.M. Fujimoto, K.S. Perumalla and G.F. Riley, “Network Simulation”, Morgan & Claypool, 2007
- [11] N.P. Mahalik, “Sensor Networks and Configuration”, Springer, 2007
- [12] Peng, Mao-heng, You-min, “ZigBee Routing Selection Strategy Based on Data Services and Energy-balanced ZigBee Routing”, IEEE Asia-Pacific Conference on Services Computing, 2006 APSCC '06, pp.400-404, December 2006
- [13] F. Lo Piccolo, D. Battaglino, L. Bracciale, A. Bragagnini, M. Turrolla, N.B. Melazzi, “On the IP support in IEEE 802.15.4 LR-WPANs: self-configuring solutions for real application scenarios”, Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean, June 2010
- [14] V. Di Renzo, “Progetto di una rete di sensori wireless per il monitoraggio di movimenti franosi”, Tesi di Laurea Magistrale, Relatore Prof. Ing. M. Chiani, Università di Bologna, Marzo 2010.
- [15] A. Giorgetti, M. Lucchi, M. Chiani, and M. Z. Win, “Throughput per pass for data aggregation from a wireless sensor network via a UAV”, IEEE Trans. Aerospace and Electronic Systems, vol. 47, no. 4, pp. 2610–2626, Oct. 2011.

- [16] M. Lucchi and M. Chiani, “Distributed detection of local phenomena with wireless sensor networks”, in Communications (ICC), 2010 IEEE International Conference on, May 2010, pp. 1–6.
- [17] M. Chen, “Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance”, IEEE Transactions on Energy Conversion, vol.21, no.2, pp.504-511, June 2006

Un ringraziamento particolare va alla mia famiglia che amo e che mi ha sempre sostenuto moralmente ed economicamente in questi anni permettendomi di raggiungere questo obiettivo. Ringrazio tutti i miei amici ed una persona speciale perché mi hanno sempre sopportato anche se più di una volta li ho trascurati. Ringrazio inoltre il prof. Marco Chiani per la disponibilità e attenzione dimostrata nei miei confronti. Ringrazio i miei colleghi di lavoro che hanno avuto una pazienza infinita nell'attraversare con me questo anno di fatiche. Ringrazio tutti coloro che hanno condiviso con me questi anni di università tra gioie, dolori, sofferenze ed emozioni perché abbiamo creato un gruppo unico che ci ha dato la forza di arrivare a questo giorno. Ringrazio te che stai leggendo queste righe, perché se sei qui vuole dire che sei parte importante della mia vita.