Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in magistrale in Informatica

Progettazione, implementazione ed analisi di una rete mesh di sensori 6LoWPAN basata su tecnologia SPIRIT

Relatore:
Chiar.mo Prof.
Marco Di Felice

Presentata da:
Giovanni Pisana

Correlatore:
Chiar.ma Prof.ssa
Eleonora Franchi Scarselli

Sessione II
Anno accademico 2016/2017

Introduzione

Nel corso degli ultimi venti anni internet ha subito una forte crescita. Con l'era del web 3.0 dispositivi embedded anche chiamati smart object, dotati di tecnologia IP sono diventati parte integrante di questo mondo. Nasce cosi *l'Internet of Things*, un era rappresentata da oggetti che dialogano tra loro per rendere la nostra vita sempre più smart. Il termine IoT è presente ovunque, basti pensare alla quantità di dispositivi connessi ad internet che ognuno di noi possiede. Al giorno d'oggi gli ambiti applicativi sono molteplici: dalle smart home, case intelligenti che regolano da sole temperatura e illuminazione in base alle condizioni esterne, all'agricoltura con sensori disposti nel terreno al fine di monitorare e comunicare le condizioni del suolo al sistema che ne gestisce l'erogazione dell'acqua e dei fertilizzanti. Alla base di tutto ciò ci sono i sensori, congegni elettronici che rendono intelligenti gli oggetti di uso quotidiano fornendo loro un modo per comunicare con l'ambiente circostante; in questo modo si può assegnare un identità a tutto ciò che ci circonda.

Le reti di sensori wireless (WSN) sono un implementazione dell'Internet of Things, dove una serie di sensori autonomi, distribuiti in uno spazio senza la necessità di un infrastruttura, cooperano tra loro allo scopo di monitorare l'ambiente circostante tramite l'uso di grandezze fisiche come temperatura, pressione, umidità, gas. Una WSN è costituita da un insieme di nodi di media/piccole dimensioni equipaggiati di sensori/attuatori, i quali posso scambiarsi informazioni mediante comunicazioni radio sfruttando appositi protocolli di comunicazione.

Sia in ambito indoor ma soprattutto in ambito outdoor la quantità di nodi-sensori impiegata per la creazione di questo tipo di reti è molto elevata, motivo per il quale nella progettazione di questi dispositivi si deve tener conto di vincoli stringenti sul consumo energetico, tramite alimentazioni a batteria o mediante tecniche di energy harvesting.

Il progetto di tesi tratterà la creazione di una rete di sensori per un ambiente indoor tramite l'utilizzo di nodi sensori prodotti e forniti dall'azienda STMicroelettronics. La rete verrà installata all'interno di una struttura universitaria dove i nodi-sensori ne monitoreranno l'ambiente tramite misure di temperatura, umidità e pressione. La creazione della rete utilizzerà gli strumenti più innovativi nell'ambito delle WSN, ovvero, il protocollo 6LoWPAN con il quale i nodi vengono dotati di tecnologia IP per comunicare e il protocollo RPL per gestire le comunicazioni all'interno della rete.

La tesi si articola in quattro capitoli. Nel primo capitolo si parlerà dello stato dell'arte, descrivendo prima i campi applicativi per poi andare ad identificare tutti gli standard che regolamentano le Wireless Sensor Network. Il secondo capitolo descriverà le tecnologie utilizzate per il progetto di tesi per poi passare al terzo capitolo dove verranno illustrate le scelte implementative. Infine il quarto concluderà il lavoro illustrando le valutazioni sperimentali.

Indice

Elenco delle figure

Capitolo 1

Wireless sensor network - WSN

Le Wireless sensor networks o anche reti di sensori sono architetture formate da un alto numero di nodi interconnessi tra di loro il cui scopo è quello di rilevare dati e reperire informazioni circa l'ambiente in cui operano. I nodi non necessitano di alcuna infrastruttura per operare e possono essere di varie dimensioni, di diverso costo e di diverso potere computazionale ma in genere a causa delle limitate capacità energetiche questi dispositivi sono caratterizzati da poca capacità di memoria e basso potere computazionale. I nodi (sensori), spesse volte vengono alimentati con batterie esterne, motivo per il quale devono essere progettati per poter consumare minor energia possibile alternando fasi in cui i dispositivi si trovano in uno stato dormiente "sleep mode" a fasi in cui i dispositivi consumano energia.

Le WSN sono caratterizzate in base all'area geografica in cui operano e si dividono in:

1. WLAN (Wireless Local Area Network): reti che permettono la comunicazione all'interno di un area di circa un centinaio di metri (Università, Biblioteche). Un esempio di questa tecnologia è il WiFi.

2. WMAN (Wireless Metropolitan Area Network): Questa tecnologia permette la connessione di più reti in un aera metropolitana come ad esempio una città. Un esempio di questa tecnologia è il Wimax, una connessione wireless a banda larga.

3. WWAN (Wireles Wide Area Network): Questo tipo di reti coprono aree ancora più ampie come ad esempio intere nazioni tramite l'uso di sistemi satellitari o particolari antenne. Tecnologie che rientrano in questo campo sono il GSM/UMTS.

4. WPAN (Wireless Personal Area Network): in queste reti il range di comunicazione è molto corto (10/20 metri circa) e l'architettura non necessita di alcuna infrastruttura per operare. Rientrano in questa categoria i raggi infrarossi (IR) e il bluetooth. Il bluetooth opera in un range leggermente superiore rispetto gli infrarossi.

1.1 Scenari applicativi

Grazie all'innovazione tecnologica le reti di sensori stanno assumendo un ruolo importante in molti scenari. Tramite un sensore è possibile osservare grandezze come temperatura, umidità, pressione, rumore, luminosità, movimento e velocità di oggetti. Possiamo quindi categorizzare le WSN in quattro grossi settori applicativi:

- **Applicazioni militari:** L'uso di sensori per il monitoraggio di ambienti ostili permette di riconoscere anomalie nell'ambiente dovute ad eventuali attacchi chimici o batteriologici evitando di esporre a rischi inutili vite umane. Anche nell'ambito militare come in quello sanitario i sensori vengono utilizzati in accessori come tute o orologi permettendo di monitorare lo stato di salute dei militari. Altre applicazioni usano la combinazione di sensori acustici e comunicazioni wireless per rilevare e tracciare oggetti pericolosi in modo da prevenire eventuali pericoli [1, 2].

- **Monitoraggio di ambienti:** Uno degli ambiti principali in cui le reti di sensori vengono utilizzate è quello del monitoraggio di ambienti sia indoor che outdoor. Ciò porta innovazione nel mondo dell'agricoltura permettendo di controllare ampie aree geografiche tappezzandole

di sensori in modo da prevenire e curare eventuali problemi in tempo reale [5]. Sempre in questo ambito sono state utilizzate le reti di sensori per monitorare lo stato di attività dei vulcani tramite nodi dotati di microfoni il cui scopo era quello di acquisire dati durante le eruzioni [4]. Per quanto riguarda gli ambienti indoor sono state sviluppate applicazioni che rilevano la qualità dell'aria tramite l'uso di sensori di umidità, temperatura, gas e polvere in modo da identificare sostanze inquinanti che possono danneggiare la salute di coloro che stanno all'interno degli edifici [3]. Data la molteplice quantità di sensori utilizzati il fattore energia risulta fondamentale. Si cerca quindi di attuare politiche di "energy saving" per trarre energia da fonti naturali (pannelli solari).

- **Applicazioni mediche:** Le reti di sensori vengono utilizzate all'interno di strutture ospedaliere per il monitoraggio di parametri fisiologici dei pazienti [6], per la localizzazione dei dottori tramite appositi sensori applicati direttamente sulla persona e per facilitare l'allocazione delle risorse ospedaliere []. Inoltre vengono impiegati per monitorare le attività ginniche tramite l'installazione di piccoli sensori all'interno di orologi o bracciali che rilevano il battito cardiaco e le calorie consumate.

1.2 Tecnologie

Con il passaggio dalle reti cablate alle reti Wireless è nata l'esigenza di stabilire nuovi standard per le comunicazioni a basso costo, bassa velocità e basso consumo energetico. L'IEEE (Institute of Electrical and Electronics Engineers) ha introdotto una serie di standard che vanno a regolare i livelli Fisico e Mac delle tecnologie wireless attualmente in uso che di seguito andremo descriveremo.

1.2.1 Standard 802.11

L'802.11 [13], meglio conosciuto come standard Wi-Fi, racchiude una serie di protocolli di trasmissione per le reti WLAN, sotto forma di varie release rilasciate nel corso del tempo. Al livello fisico il protocollo utilizza frequenze che vanno dai 2,4 GHz ai 5 GHz con un datarate impostato a 600 Mbps; la qualità del segnale non è delle migliori e viene molto degradata in presenza di ostacoli, acqua e altri fattori che ne riducono la portata. Al livello MAC l'accesso al canale avviene tramite il CSMA/CA (Carrier Sense Multiple Access with Collision Avoidence) il quale funziona nel seguente modo. Quando una stazione vuole comunicare ascolta il canale di trasmissione; se risulta occupato rimanda la trasmissione ad un momento successivo evitando possibili collisioni; se invece risulta libero la comunicazione è permessa.

1.2.2 Standard 802.15.1

l'802.15.1 [14, **?**], meglio conosciuto come Bluetooth, è uno standard per reti low-power che fornisce un metodo economico e sicuro per scambiare informazioni tra dispositivi attraverso una frequenza radio sicura e a corto raggio. A causa del limitato range di comunicazione si presta bene per le reti PAN (Personal Area Network), piccole reti di estensione di qualche metro. Bluetooth opera nella banda ISM dei 2,4 e 2,5 GHz (ISM) usando una modulazione a frequenza (FSK) con datarate di 720Kbps. Una insieme di dispositivi sincronizzati su una stessa frequenza viene chiamata "piconet". All'interno della piconet solo uno dei dispositivi è chiamato "master" e si occupa della sincronizzazione mentre gli altri sono detti "slave".

Questa tecnologia è progredita con il passare degli anni fino a quando nel 2010 è stata introdotta una versione aggiornata sotto il nome di Bluetooth Low Energy, che si focalizza più sul basso consumo energetico che sulla velocità di trasferimento. Un dispositivo che monta questa tecnologia ha dimensioni molto ridotte e il basso consumo di batteria fa si che non abbia alcuna necessità di manutenzione periodica. Il Bluetooth Low Energy ope-

ra alla frequenza dei 2,4 GHz (ISM) e impiega un transceiver a frequency hopping per combattere le interferenze tra dispositivi in prossimità.

Ma il punto di forza di questa tecnologia sta in dispositivi chiamati Beacon, i quali usano il Bluetooth Low Energy per creare un'infrastruttura smart, orientata alla localizzazione, che i dispositivi mobili possono utilizzare per ricavare informazioni, in tempo reale. Un beacon può essere programmato per aiutare il cliente nell'acquisto in negozio o semplicemente per raccogliere informazioni sull'ambiente circostante. Questo apre la strada ad un nuovo livello di interazione con il mondo, senza la necessità di avere una connessione ad internet.

1.2.3 Dash7

La Dash7-Alliance [10], un consorzio no-profit ha fondato uno standard per comunicazioni RFID in reti di sensori low power. L'ISO/IEC 18000-7 definisce una frequenza di comunicazione di 433 MHz nella banda ISM (disponibile in tutto il mondo) che permette una migliore propagazione. La rete si basa sulla modalità master-slave, chiamati anche interrogatori e tag e si compone di quattro tipologie di dispositivi:

- Blinker: un dispositivo le cui funzioni sono limitate alla sola trasmissione dei dati.

- Endpoint (tag o slave): un dispositivo lowpower che può sia trasmettere che ricevere dati.

- Subcontroller: Simile ad un Endpoint ma con qualche funzionalità in più. Grazie ad eventi di wake-up può alternare fasi in cui risulta attivo a fasi in cui risulta in uno stato dormiente per risparmiare batteria.

- Gateway: un dispositivo che permette di connettere una rete D7A ad un altre rete. Per fare ciò non può essere mai offline.

1.2.4 LoRa

LoRa, il cui acronimo sta per Long Range, è uno standard di comunicazione wireless a lungo raggio introdotto dalla LoRa Alliance. Questo protocollo si presta bene per dispositivi dove il consumo energetico risulta un parametro importante. Il range d'azione di un dispositivo dipende molto dall'ambiente e dalla possibile presenza di ostruzioni fisiche ma in generale riesce a coprire intere città. A livello fisico opera alle frequenze di 433, 868 e 915 MHz nelle bande ISM in base alla regione in cui viene utilizzato con un datarate di 50 Kbps. La topologia utilizzata è di tipo "star-of-stars" e include tre differenti tipologie di dispositivi:

- End-device: dispositivi low-power che comunicano con le LoRa gateway.
- LoRa gateway: ricevono informazioni dai dispositivi End-device che inoltrano verso il LoRa Network server.
- Network server: questo nodo si occupa di decodificare i pacchetti in arrivo spediti dai device ed eventualmente genera quei pacchetti che devono essere rimandati indietro.

1.2.5 Standard 802.15.4

L'802.15.4 [15] è stato pensato per reti WPAN a basso bit rate costituite da dispositivi alimentati tramite batterie come ad esempio i sensori. Questo standard definisce il livello Fisico e MAC dello stack protocollare. A livello fisico offre un servizio che interviene sull'accensione e spegnimento del modulo radio permettendo al sistema di risparmiare energia ed inoltre fornisce tre possibili bande operative:

- **868-868.6 Mhz**: banda utilizzata nella maggior parte dei paesi europei con un data-rate di 20 kbps

- **902-928 Mhz**: banda utilizzata nel continente oceanico e nel continente americano con un data-rate di 40 kbps

- **2400-2483.5 Mhz**: banda utilizzabile in tutto il globo con data-rate di 250 kbps

La tecnica di modulazione utilizza la BPSK (Bynary Phase Shift Key), una delle tecniche più utilizzate e robuste nell'ambito delle reti wireless la quale associa alle diverse fasi della portante una serie di valori binari. A livello MAC offre servizi quali la possibilità di creare una rete PAN, la trasmissione dei beacon e l'accesso al canale tramite il protocollo CSMA/CA come nello standard Wi-Fi.

Solitamente per le reti di sensori si ricorre all'uso di questo standard poiché gli standard come il WiFi ad esempio vengono per lo più sfruttati per il collegamento del nodo gateway della rete di sensori verso Internet o un altro host.

ZigBee

ZigBee è il protocollo principale che utilizza lo standard 802.15.4. Creato dal consorzio ZigBee Alliance viene pensato per sistemi wireless a basso data rate e basso consumo energetico. Permette ai dispositivi di gestire il risparmio energetico tramite fasi di sleep in cui il dispositivo si spegne per risparmiare energia. ZigBee a livello Fisico e Mac si appoggia allo standard 802.15.4 mentre specifica il livello rete e quello applicativo. I dispositivi che fanno uso di questo protocollo si differenziano in tre categorie : ZigBee Coordinator, un dispositivo FFD che si occupa di formare la rete, ZigBee Router instrada i pacchetti che devono giungere a destinazione e ZigBee End Device, con capacità limitate, non può inoltrare i messaggi come i nodi router ma si limita solo a rispondere alle richieste. Questi dispositivi possono essere collegati a formare una rete a stella, mesh o a cluster-tree variando il numero di tipologie di dispositivi. Lo svantaggio di questo protocollo sta nel fatto che i dispositivi

ZigBee, possono comunicare solo con altri dispositivi della stessa famiglia, assumendo che utilizzino lo stesso profilo (Ad esempio il Bluetooth)

6LoWPAN

Il concorrente di ZigBee è rappresentato dal protocollo 6LoWPAN (IPv6 over Low power Wireless Personal Area Network). Questo protocollo è stato creato dall' Internet Engineering Task Force (IETF) e utilizza lo standard 802.15.4. Lo stack protocollare si basa su un protocollo TCP/IP adattato per essere utilizzato in dispositivi con stringenti vincoli energetici e di memoria. Il punto di forza di questo protocollo sta nel fatto che utilizza indirizzi IPv6 per identificare i dispositivi della rete, cosa che lo rende compatibile con la maggior parte delle applicazioni presenti su internet al momento. Viene introdotto infatti nello stack un livello adattivo con il quale gli indirizzi IPv6 vengono compressi per essere utilizzati all'interno delle reti di sensori. Naturalmente segue una decompressione per quei pacchetti che devono essere inviati all'esterno della rete.

Sembra quindi che 6LoWPAN sia più attraente poiché è basato su tecnologia IP, viceversa ZigBee sembra quello più popolare e più adottato.

1.3 Topologia

Prima di parlare di topologia bisogna tenere in mente uno dei punti fondamentali delle reti di sensori ovvero, l'assoluta libertà offerta nel posizionamento fisico dei nodi nello spazio oltre al fatto che in una tipica rete i sensori sono nell'ordine delle decine di unità per metro quadro. Nuovi nodi possono essere aggiunti alla rete e allo stesso tempo nodi esistenti possono guastarsi per vari motivi (malfunzionamenti, guasti, mancanza di energia). Questi fattori caratterizzano molto la scelta topologica della rete da adottare. Lo standard 802.15.4 supporta tre topologie di rete:

Stella: In questa topologia tutti gli elementi dipendono da un coordinatore (PAN) e comunicano direttamente con esso. Ogni device vede solo il

coordinatore e per dialogare con gli altri device deve passare necessariamente da esso. Uno svantaggio di questa configurazione sta nel nodo coordinatore al quale spetta il compito più importante; se il traffico verso il coordinatore è ingente, potrebbe causare un collo di bottiglia comportando una perdita di pacchetti

Mesh: In questa topologia il coordinatore è ancora presente ma la comunicazione tra i nodi non è necessariamente mediata da esso. In questo modo si possono avere percorsi rindondanti permettendo topologie e algoritmi di routing più complessi rispetto a quelli adottati nella configurazione a stella. Questo tipo di rete permette la comunicazione multi-hop e sarà quella che verrà adottata nella fase implementativa;

Cluster-Tree: Nella topologia ad albero è presente un nodo radice che svolge la funzione di coordinazione (PAN-coordinator), dei nodi intermedi (Co-ordinator) che fanno da tramite tra il coordinatore e i nodi foglia (end-device). Ogni nodo può comunicare con il proprio padre e con i propri figli.

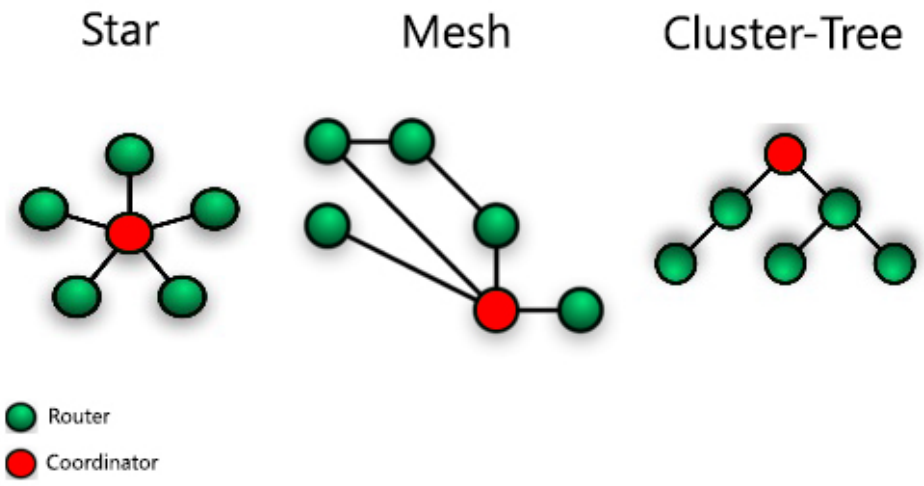

Figura 1.1: Topologie reti di sensori

Capitolo 2

Tecnologie utilizzate

Questo capitolo parlerà delle scelte progettuali che hanno portato alla realizzazione del lavoro. Verrà descritto il protocollo 6LoWPAN con la sua architettura e componenti, il protocollo di routing adottato e una descrizione generale del sistema operativo utilizzato per lo sviluppo della tesi. Tutti questi concetti serviranno a comprendere meglio le scelte adottate nella fase implementativa.

2.1 Il protocollo di rete IP

Un indirizzo IP permette di identificare un dispositivo collegato ad una rete che utilizza lo standard TCP/IP a livello rete. Inizialmente fu utilizzato l'Internet Protocol Version 4 (IPv4), che utilizza 32 bit per rappresentare un singolo indirizzo e riesce ad identificare un massimo di 2^{32} indirizzi univoci. Con il passare degli anni però i dispositivi che possono collegarsi ad internet sono aumentati in maniera esponenziale provocando l'esaurimento degli indirizzi IPv4 disponibili. Per risolvere questo problema è stata introdotta la versione 6 del protocollo IP (IPv6) il quale porta le seguenti novità e vantaggi:

- **Rappresentazione degli indirizzi:** Con IPv6 si riesce ad identificare una quantità molto più elevata di indirizzi IP; questo perché per

rappresentare un singolo indirizzo vengono utilizzati 128 bit e non 32, riuscendo cosi a rappresentare 2^{128} indirizzi univoci.

- **Eliminazione del NAT:** Il NAT (NeTwork Address Translation) è una tecnica usata per sostituire gli indirizzi IP contenuti negli header dei pacchetti. Era stato introdotto nel protocollo IPv4 per ovviare alla scarsità di indirizzi IP e quindi con il protocollo IPv6 è stato eliminato completamente.

- **Eliminazione DHCP:** Il DHCP si occupava di assegnare agli host gli indirizzi IP. Con il protocollo IPv6 non vi è più la necessità di usarlo poiché gli IP vengono auto-assegnati tramite procedure come la stateless address auto-configuration.

- **Struttura Header migliorata:** Molti campi dell'header IPv4 sono opzionali o comunque poco usati, per cui IPv6 elimina tutti quei campi non ritenuti fondamentali riducendo cosi la dimensione dell'header del pacchetto IP.

Se da un lato IPv6 ha portato novità nel mondo delle reti, da un altro lato ha portato grossi problemi di compatibilità nella comunicazione tra un dispositivo che usa IPv4 e uno che usa IPv6. Per permettere questo tipo di comunicazione sono state identificate una serie di tecniche come la Dual Stack la quale prevede l'utilizzo di un doppio stack IP per poter interpretare entrambi le versioni del protocollo IP o la tecnica di Tunnelling con la quale viene stabilito un tunnel point-to-point tra due host che vogliono comunicare. I pacchetti IPv6 vengono incapsulati alla sorgente in pacchetti IPv4 e inviati nel tunnel per poi essere decapsulati a destinazione.

2.2 6LoWPAN

Al giorno d'oggi esistono molte applicazioni che traggono beneficio dalla comunicazione Wireless. Molte di queste, fanno uso di tecnologie proprietarie che rendono difficile l'integrazione con il mondo delle reti basate su IP.

Il protocollo 6LoWPAN [8] (IPv6 over Low power Wireless Personal Area Network) permette di superare queste limitazioni in quanto utilizza il protocollo IPv6 per la comunicazione. Grazie a questo, dispositivi forniti di tecnologia IP possono facilmente connettersi a reti IP senza la necessità di appositi gateway o proxy che facciano da mediatori. Di seguito verrà descritta l'architettura di questo protocollo.

2.2.1 Architettura

Una rete di sensori Wireless può essere vista come un insieme di isole di dispositivi collegati tra di loro. Un isola viene chiamata LoWPAN ed è caratterizzata dal fatto che i dispositivi che ne fanno parte condividono lo stesso prefisso IPv6. All'interno di una LoWPAN i nodi vengono classificati in due categorie, Router e Host. I nodi Router a differenza dai nodi Host hanno la capacità di inoltrare le informazioni ricevute verso altri nodi, cosa che i nodi Host non possono fare. Inoltre possono essere presenti anche particolari tipologie di nodi Router chiamati Edge router i quali fanno da mediatori tra la LoWPAN di appartenenza ed eventuali altre reti LoWPAN presenti nell'area circostante. I nodi sono liberi di muoversi da una LoWPAN ad un altra e ciò comporta un cambio nell'indirizzo IPv6 per quanto riguarda il prefisso della rete in cui andranno a spostarsi.

Il protocollo 6LoWPAN definisce tre tipologie di reti che di seguito andremo a descrivere

Simple LoWPAN Questo tipo di rete è formata da una serie di nodi Router/Host ed è connessa ad altre eventuali reti IP tramite un Edge Router che fa da mediatore.

Extended LoWPAN Questa è una versione più complessa della Simple LoWPAN dove gli Edge Router presenti possono essere più di uno, collegati assieme tramite una backbone;

Ad-Hoc LoWPAN questo tipo di rete non è connessa ad internet e non necessita di alcuna infrastruttura per operare. Possono essere presenti

sia nodi Router che nodi Host e in alcuni casi è possibile concedere privilegi particolari ad alcuni nodi rendendoli simili ad un Edge Router in modo da svolgere funzioni quali la possibilità di generare indirizzi unique local address (ULA) e la possibiltà di effettuare la procedura di registrazione per funzionalità come la Neighbors Discovery

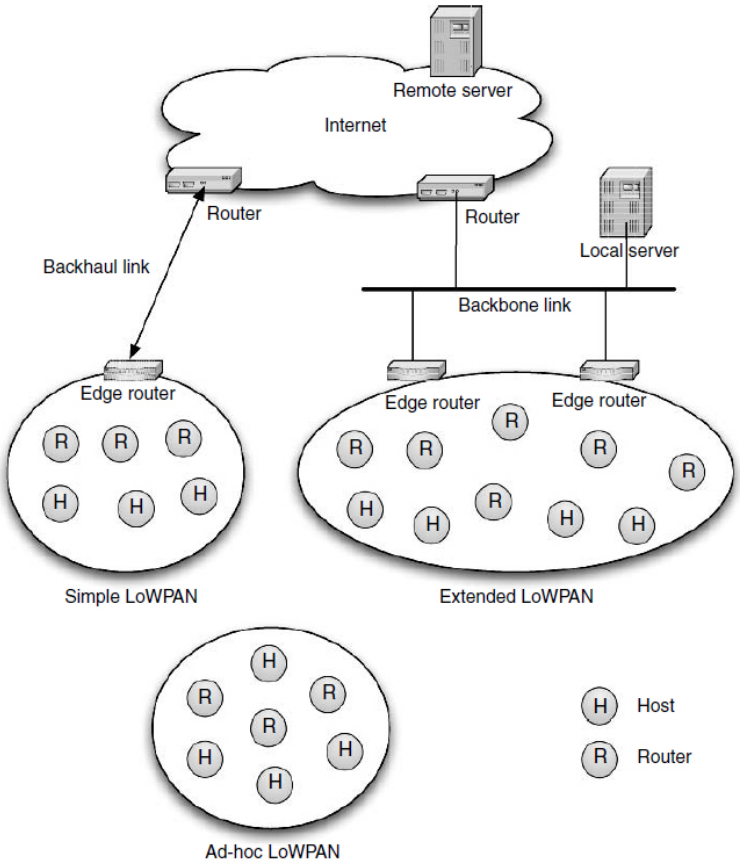

Figura 2.1: Architetture 6LoWPAN [8]

2.2.2 Stack protocollare

Lo Stack utilizzato nel protocollo 6LoWPAN è molto simile allo stack utilizzato per il protocollo IP, ma con una serie differenze limitate dalla

tecnologia in cui questo protocollo viene usato.

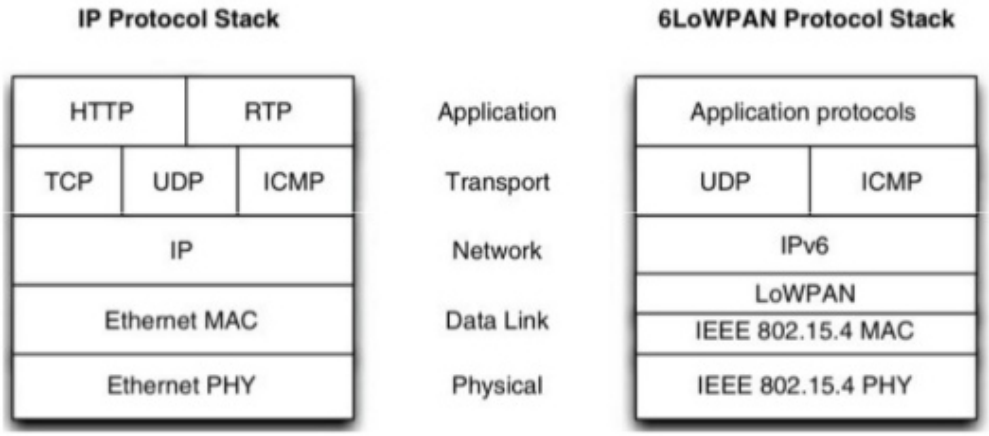

Figura 2.2: Stack 6LoWPAN

Livello fisico e MAC

Per quanto riguarda il livello Fisico 6LoWPAN fa uso dello standard 802.15.4 fissando un data-rate di 250 Kbps, frequenza che oscilla tra 200 e 2483.5 MHz e un payload massimo di 127 byte. Invece per quanto riguarda il livello MAC lo standard 802.15.4 definisce quattro tipologie di frame: data frame utilizzati per il trasferimento dei dati, beacon frame utilizzati da un nodo coordinatore per strutturare la comunicazione tra i nodi, acknowledgement frame per confermare l'avvenuta ricezione di un data frame e MAC layer command frame per abilitare alcuni servizi a livello MAC (associazione o disassociazione da un nodo coordinatore).

Livello adattivo

A differenza del tipico stack IP, viene introdotto un livello chiamato "livello adattivo" il quale fa da mediatore tra il livello MAC e il livello rete. Si occupa di tre operazioni fondamentali:

- **Frammentazione e riassemblaggio:** L'IEEE 802.15.4 fornisce una dimensione per la Maximum Trasmission Unit (MTU) di 127 byte di cui 25 byte sono destinati per l'header, footer e spazio riservato agli indirizzi e 21 byte per l'overhead, per un totale di 81 byte. D'altra parte un tipico pacchetto IPv6 ha una dimensione di 1280 byte. Di conseguenza, per problemi di dimensione un pacchetto IPv6 non può essere incapsulato in un frame 802.15.4. Per risolvere questo problema il livello adattivo effettua una frammentazione dell'indirizzo IPv6 in più di 16 frammenti.

- **Compressione dell'header:** questo meccanismo chiamato HC1 agisce sull'header del pacchetto IPv6 andando ad eliminare tutte quelle informazioni ridondanti. Esistono infatti alcuni campi nell'header IPv6 che sono comuni a tutti i dispositivi di una 6LoWPAN che possono essere eliminati andando a ridurre la dimensione del singolo pacchetto IPv6.

- **Routing:** Questo livello dello stack gestisce una delle due modalità di routing attuate dal protocollo 6LoWPAN. Il routing a questo livello è chiamato Mesh-under [17]; i pacchetti vengono inviati a livello link-layer sotto forma di Frame 802.15.4 e utilizzano indirizzi a 64 o a 16 bit. Per inviare un pacchetto esso viene frammentato alla sorgente e per raggiungere il successivo hop viaggia su cammini multipli. Se tutti i frammenti raggiungono con successo il nodo di destinazione, allora il livello adattivo del nodo di destinazione si occuperà del riassemblaggio. Se anche un frammento non arriva a destinazione il pacchetto dovrà essere ritrasmesso. Questo tipo di routing non è performante in reti con elevato numero di hop poiché la probabilità di perdita di pacchetti è elevata.

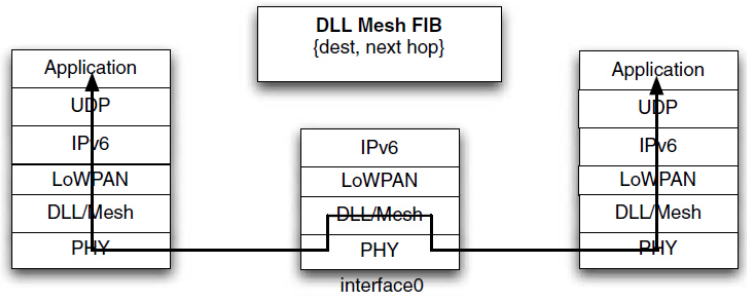

Figura 2.3: Routing Mesh-under

Livello rete

Il livello rete definisce un altra tipologia di routing chiamata Route-over. Questo tipo di routing si basa su indirizzi IPv6 e sfrutta le tabelle di routing per inviare i dati. Ogni nodo della rete viene visto come un router IP il quale ha la possibilità di inoltrare i messaggi che non sono destinati ad esso. I pacchetti vengono frammentati e riassemblati ad ogni Hop per poi arrivare a destinazione. Alla ricezione di un pacchetto il nodo controlla se è destinato ad esso, di conseguenza, se è lui il destinatario inoltra il pacchetto al livello superiore viceversa, si limita ad inoltrare il pacchetto al nodo successivo in accordo con le informazioni contenute nella tabella di routing. Questo tipo di routing è più performante in reti di grandi dimensioni con elevato numero di hop.

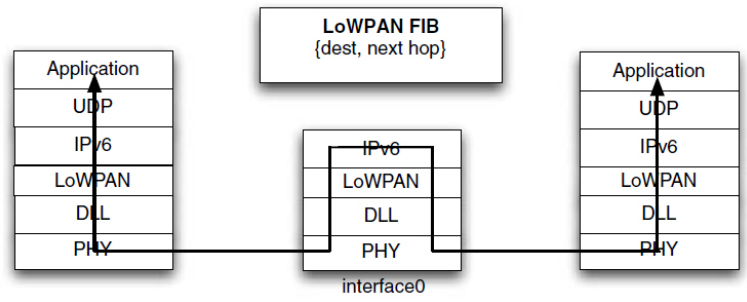

Figura 2.4: Routing Route-over

Livello trasporto

La principale differenza a livello trasporto riguarda la tipologia di protocollo utilizzato per trasmettere i dati. 6LoWPAN utilizza solo ed esclusivamente il protocollo UDP poiché il protocollo TCP, utilizzato nello standard IP, è troppo complesso per quanto riguarda performance, efficienza e complessità. Inoltre nel caso di mobilità dei nodi-sensori, quando avviene un meccanismo di handover, il livello trasporto reagisce con un cambio di indirizzo IP per uno dei due end-point. Il protocollo UDP è indifferente al cambio di indirizzo IP poiché ogni datagramma è indipendente. TCP invece non è capace di risolvere questo problema per cui la connessione potrebbe interrompersi dopo l'handover.

Livello Applicativo

A livello applicativo 6LoWPAN utilizza i socket per specifiche applicazioni. Un socket viene aperto per poter ricevere o inviare pacchetti ed ogni socket è associato al protocollo trasporto utilizzato oltre che alla porta sorgente o di destinazione. Esempio di applicazioni a questo livello sono COAP e MQTT.

- **MQTT:** Il Message Queue Telemetry Transport è un protocollo di messaggistica instantanea che utilizza un approccio di tipo publish/subscriber. Creato da IBM, fornisce una connessione verso nodi remoti con potenza di calcolo e condizioni di banda limitata. Agisce con due tipologie di dispositivi: i client e i broker. I nodi client sono collegati ad uno o più broker i quali distribuiscono i messaggi ricevuti su uno specifico topic ai client che si erano sottoscritti. Ma per andare incontro ai vincoli stringenti nell'ambito delle reti di sensori, poichè MQTT fa uso del protocollo TCP per operare, è stato sviluppato MQTT-S, una versione più leggera utilizzabile in reti di sensori low-power che fanno uso del protocollo UDP, più adatto alle reti di sensori.

I topic MQTT-S sono strutturati in modo gerarchico e vengono utilizzate delle wildcard per poter osservare la gerarchia. Quando un client invia un messaggio al broker può scegliere di renderlo persistente, in questo caso il broker provvederà a salvarlo in memoria. Ai client per garantirne la sicurezza viene chiesto di autenticarsi con username a password.

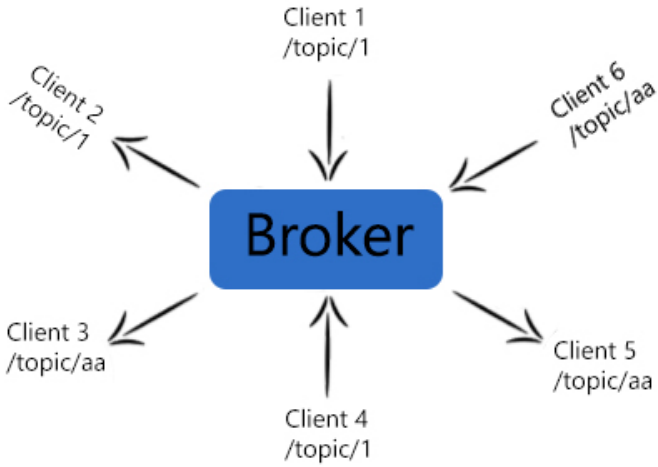

Figura 2.5: Architettura MQTT

Essendo un protocollo message-oriented, MQTT fa uso di messaggi ognuno con un header fisso, un header variabile più un payload. L'uso dell'header variabile non è obbligatorio ma serve a fornire informazioni aggiuntive. L'header di un tipico pacchetto MQTT è strutturato nel modo seguente:

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

Figura 2.6: Header MQTT

– Message Type: specifica il tipo di messaggio (Es: richiesta di connessione al Server)

  - DUP: specifica se il pacchetto è un duplicato.
  - Retain: specifica se il messaggio deve essere mantenuto persistente o no.
  - il rimanente spazio sarà occupato dal payload più l'eventuale header variabile.

- **CoAP:** simile ad HTTP, è stato creato per le esigenze dei dispositivi in cui viene usato. Questo protocollo è standardizzato dal Working Group della internet Engineering Task FOrce (IETF) ed è costituito un insieme di interfacce REST comuni con il protocollo HTTP. Opera in dispositivi caratterizzati da basse prestazioni e consumi come le reti di sensori. L'architettura è molto simile ad un client-server tipico del protocollo HTTP e si appoggia al protocollo di comunicazione di livello trasporto UDP.

  I messaggi hanno un formato composto da 4 campi: un header di 4 byte contiene le informazioni sul messaggio (versione, tipologia di messaggio, duplicato?), un token di dimensione fra 0 e 8 byte incluso dal client nella richiesta, una sequenza di opzioni e un palyload che rappresenta la parte utile del messaggio.

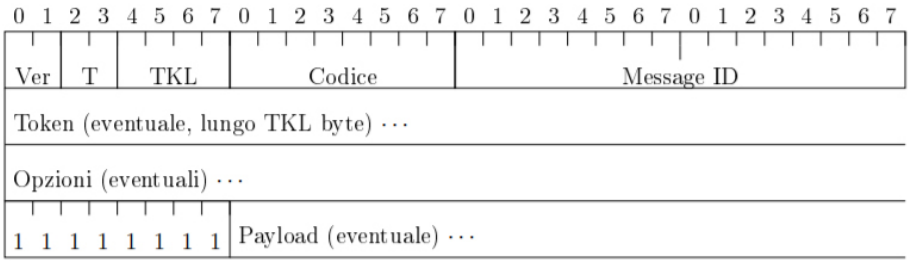

Figura 2.7: Header COAP

  I client effettuano richieste ai server, i quali rispondono inviando le informazioni richieste.

2.2.3 Meccanismo d'indirizzamento

Poiché le reti di sensori impiegano una quantità elevata di nodi sensori, nasce l'esigenza di identificare univocamente ogni nodo assegnandogli un indirizzo IP. Con la crescente domanda di indirizzi ip è stato necessario introdurre la versione 6 del protocollo IP per ovviare all'esaurimento degli indirizzi disponibili (IPv4). Con il passaggio da IPv4 a IPv6 si passa a rappresentare indirizzi non più a 32 bit ma a 128 bit permettendo in questo modo di rappresentare miliardi di indirizzi univoci.

Tre sono le categorie di indirizzi IPv6:

1. **Unicast uno-a-uno:** identifica una singola interfaccia di rete. Un pacchetto destinato ad un indirizzo unicast raggiungerà un interfaccia in modo univoco.

2. **Anycast uno-a-molti:** è simile a quello unicast con la differenza che il pacchetto, destinato ad un indirizzo anycast sarà consegnato ad una delle interfacce identificate da quell'indirizzo e raggiungerà il nodo più vicino, in base al protocollo di instradamento scelto (in genere il più vicino).

3. **Multicast uno-a-molti:** a differenza dell'anycast il pacchetto destinato ad un indirizzo di multicast verrà consegnato a tutte le interfacce a cui questo indirizzo si riferisce.

L'indirizzo può essere essere visto come due componenti distinte, un prefisso e un identificatore di interfaccia. Il prefisso in genere viene utilizzato per indicare la sottorete di appartenenza mentre l'identificatore di interfaccia identifica il dispositivo (in genere l'indirizzo MAC). Gli indirizzi unicast a sua volta possono classificarsi in link-local e global. Gli indirizzi link-local hanno validità e visibilità nell'ambito della sottorete di appartenenza e non possono essere utilizzati all'esterno per raggiungere destinazioni verso altre sottoreti. Gli indirizzi global sono dotati di un prefisso ed un identificatore di interfaccia, sono univoci e permettono la comunicazione tra le varie sottoreti.

Un altro requisito delle reti di sensori sta nella loro capacita di auto-configurarsi. I nodi per generare il proprio indirizzo IP non hanno bisogno di utilizzare supporti esterni come il DHCP o il NAT ma per fare ciò utilizzano funzionalità come la Stateless Address Autoconfiguration, un meccanismo che permette ad un nodo di generare il proprio indirizzo IP usando una combinazione di informazioni disponibili localmente (indirizzo MAC) più una serie di informazioni fornite dalla rete (prefisso). In questo modo l'indirizzo IP che verrà generato identificherà in modo univoco il dispositivo all'interno della rete di appartenenza.

2.3 Routing

Il routing è il processo con il quale si vuole determinare un cammino tra due nodi della rete che non hanno la possibilità di comunicare tra di loro. Trovandoci in un ambiente con limitate capacità energetiche, computazionali e di memorizzazione lo sviluppo di questo tipo di protocolli risulta abbastanza impegnativo e richiede molta attenzione. I protocolli di routing nelle reti di sensori vengono caratterizzati in base a come le rotte vengono determinate. Un protocollo di tipo proattivo calcola le rotte prima ancora che esse siano utilizzate, viceversa un protocollo di tipo reattivo determina la rotta solo quando viene richiesta. Inoltre esistono protocolli ibridi che utilizzano una combinazione dei due protocolli appena citati.

In base alla conoscenza che un nodo ha della rete esistono protocolli di tipo link-state e protocolli di tipo distance-vector. In quelli di tipo link-state ciascun nodo comunica a tutti i nodi della rete lo stato dei suoi collegamenti adiacenti. In questo modo viene creato un database che rappresenta la topologia completa della rete. In quelli di tipo distance-vector i nodi non sono a conoscenza delle topologia della rete ma conoscono semplicemente i collegamenti con i nodi adiacenti. Ogni nodo mantiene una tabella che associa ad ogni destinazione la stima della distanza che lo separa dalla destinazione stessa più il primo passo per raggiungerla.

Inoltre possiamo raggruppare i protocolli di routing in tre categorie principali in base alla struttura della rete:

- **Data-centric:** Questa tipologia di routing si adatta a reti di grandi dimensioni dove il numero di nodi è elevato, causando un elevata rindondanza di informazione rendendo difficile creare un sistema di indirizzamento globale. Questo tipo di protocollo usa metodi di aggregazione per l'inoltro dei dati. Esempi di routing Data-centric sono SPIN e Directed Diffusion

- **Gerarchico:** Per risolvere i problemi di scalabilità della rete e per evitare di sovraccaricare troppo gli Edge Router questo tipo di protocollo tende a creare cluster di nodi disposti in una struttura gerarchica organizzata a livelli. Il vantaggio sta nel ridurre al minimo il numero di pacchetti inviati al nodo sink, diminuendo il consumo energetico. Esempi di protocolli di questo tipo sono LEACH e PEGASIS

- **Geografico:** Questo tipo di routing per identificare i nodi si basa su informazioni relative alla loro posizione. La posizione può essere stimata usando strumenti come ad esempio il GPS. Il principale protocollo di questa tipologia e GEAR.

2.3.1 RPL

L'IETF ROLL (Routing Over Low power and Lossy networks) ha definito una serie di specifiche per la creazione di un protocollo di routing per reti di sensori low-power. È nato cosi RPL (Routing Protocol for Low power and Lossy networks), un protocollo proattivo distance vector che può essere inquadrato in parte nel contesto del routing gerarchico per il fatto che crea una struttura logica su di una una fisica già esistente.

Topologia

RPL fa uso di un grafo orientato (DODAG - Destination Oriented Directed Acyclick Graph) in cui non esistono cicli e in cui gli archi formano dei cammini che vanno dai nodi inferiori verso il nodo radice in cui il cammino termina. Il nodo radice chiamato DODAG root ha quindi solo cammini entranti. Una rete può comprendere diversi DODAG e per mantenere la coerenza tra i vari DODAG vengono considerati una serie di parametri:

- **RPLInstanceID.** Identifica un istanza di RPL dove per istanza si sta ad intendere un insieme di DODAG indipendenti e non correlati. Un nodo può appartenente ad un solo DODAG all'interno della stessa istanza.

- **DODAGID.** Identifica un particolare DODAG e una particolare DODAG root. La coppia (RPLInstanceID, DODAGID) identifica un DODAG all'interno nella rete.

- **Rank.** Questo valore identifica la posizione di un nodo rispetto la radice. Solitamente la radice ha un rank nullo e man mano che ci si allontana dalla radice il rank cresce in maniera monotona fino ad arrivare alle foglie dove il rank è massimo. Il valore del rank viene calcolato da una funzione obiettivo in base ad una determinata metrica scelta a priori.

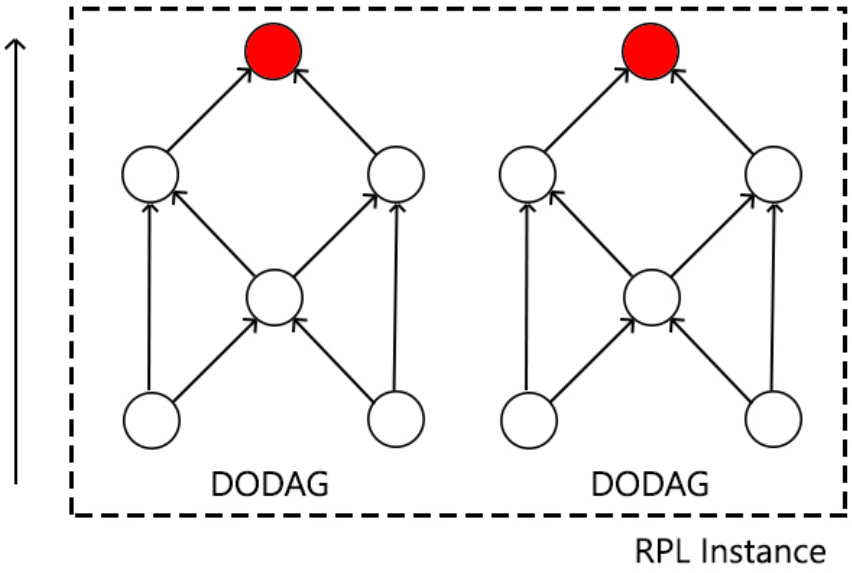

Figura 2.8: Topologia protocollo RPL

Messaggi

Per la creazione della rete e per la gestione del routing RPL utilizza una serie di messaggi, definiti all'interno di un pacchetto ICMPv6 i quali si classificano in:

- **DODAG Information Sollecitation (DIS).** Viene utilizzato per scoprire l'esistenza di nodi vicini. È mandato per sollecitare l'invio di un pacchetto DIO da un altro nodo.

- **DODAG Information Object (DIO).** Questo messaggio viene utilizzato per la formazione e per il mantenimento dei cammini verso la radice. In questo modo un nodo può scoprire l'esistenza di un istanza di RPL e apprenderne i parametri. Un nodo può ricevere più messaggi di tipo DIO dai vari nodi della DODAG. Di conseguenza dovrà scegliere il nodo migliore a cui collegarsi; ciò avviene osservando le metriche fornite dalla funzione obiettivo (numero minimo di hop, qualità del canale)

- **Destination Advertisement Object (DAO).** Questo messaggio viene utilizzato per la formazione dei camini dal nodo radice verso gli altri nodi del DODAG ed è mandato in modalità unicast da un figlio verso il genitore. Il DAO può essere confermato dal genitore con un messaggio di DAO-Ack

Formazione dei cammini

I cammini verso il nodo radice vengono costruiti tramite l'invio di un messaggio di tipo DIO tra i nodi della rete. Il messaggio conterrà una serie di campi tra i quali l'istanza di RPL e il DODAGID impostati inizialmente dal nodo radice. I nodi che ricevono questo tipo di messaggio dovranno adottare una configurazione coerente con i parametri contenuti all'interno del messaggio aggiornando il campo Rank il quale decresce man mano che si sale nel cammino verso la radice, che ha rank minimo. Poiché un nodo può ricevere più messaggi DIO da più nodi, dovrà scegliere tra un insieme di nodi genitori un solo genitore (preferred parent) a cui collegarsi. Questa scelta viene effettuata andando a confrontare i rank dei vari genitori e selezionando il genitore con rank migliore. Il rank viene assegnato in base ad una funzione obiettivo (Of) la quale stabilisce la metrica per il calcolo della distanza tra due nodi [**?**]. Questa distanza può essere considerata in termini di "hop count" (Of0) oppure utilizzando la metrica "ETX". La prima consiste nello scegliere il genitore con il valore minimo di HOP_COUNT ovvero il genitore più vicino, mentre la seconda consiste nello scegliere il genitore con valore minimo di ETX (Expected Trasmission count) che rappresenta il numero totale atteso di trasmissioni richieste per consegnare un pacchetto con successo a destinazione. Con la seconda metrica si preferisce puntare più sulla qualità del collegamento che sulla distanza.

I cammini dal nodo radice verso gli altri nodi vengono costruiti tramite i messaggi di tipo DAO in due modalità distinte:

- **Modalità non storing.** In questa modalità la tabella di routing è presente solo nel Border Router e conterrà tutti i percorsi possibili

verso un determinato nodo all'interno del DODAG. Quando il Border Router vuole inviare un messaggio al nodo N4, crea un un pacchetto inserendo all'interno di un header sorgente tutti i nodi che il messaggio dovrà raggiungere per arrivare a destinazione e invierà il messaggio al primo nodo della catena.

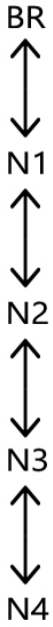

Tabella di routing	
Indirizzo di destinazione	**Indirizzo Next-hop**
IP nodo N1	----
IP nodo N2	IP nodo N1
IP nodo N3	IP nodo N2
IP nodo N4	IP nodo N3

Rotta			
IP nodo N1	IP nodo N2	IP nodo N3	IP nodo N4

Figura 2.9: Routing con modalità non-storing

- **Modalità storing.** In questa modalità, a differenza dalla precedente ogni nodo mantiene la propria tabella di routing che conterrà tutti i percorsi esistenti all'interno del proprio sotto albero. Quando un nodo vuole inviare un messaggio inserirà all'interno dell'header sorgente solo l'indirizzo del nodo di destinazione e inoltrerà il messaggio ad uno dei nodi a distanza uno nel suo sotto albero il quale, si occuperà di inoltrarlo a sua volta in base alle informazioni contenute nella propria tabella di routing.

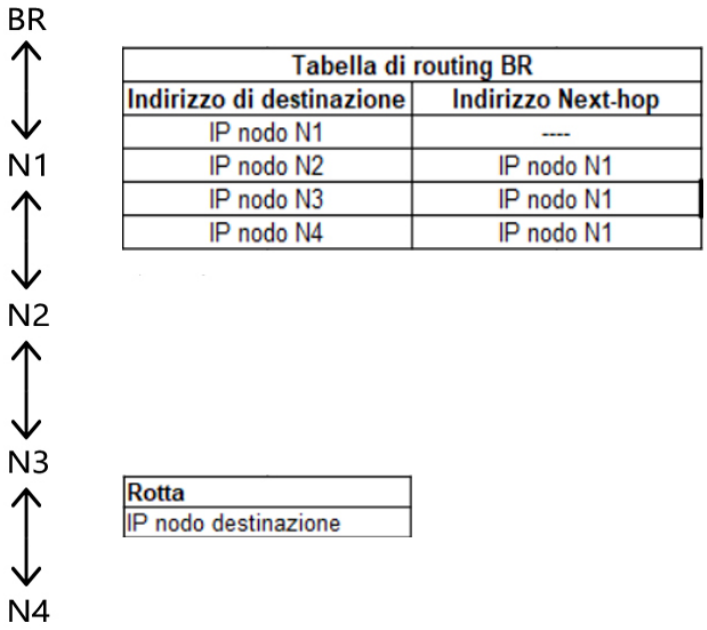

Figura 2.10: Routing con modalità storing

2.4 Il sistema operativo Contiki

Contiki è un sistema operativo progettato per soddisfare i requisiti che i dispositivi di piccole dimensioni devono soddisfare, ovvero pochi kilobyte di memoria disponibile. È completamente gratuito e il suo codice è disponibile su Github per gli sviluppatori [9]. Implementa sia lo standard IPv4 che quello IPv6 ma supporta anche varianti low-power come 6LoWPAN il quale verrà utilizzato per sviluppare il progetto di tesi. Inoltre supporta lo standard C e fa uso di "protothread", particolari tipologie di thread progettati per essere usati su sistemi con ristretti vincoli di memoria come ad esempio le reti di sensori.

Contiki rientra tra i sistemi operativi progettati secondo un approccio event-driven dove il comportamento dei processi viene gestito da un flusso di eventi all'interno di una macchina a stati finiti. La routine di gestione degli eventi (event-handler) viene invocata da un event-dispatcher e un evento in esecuzione può essere interrotto da meccanismi come ad esempio inter-

rupt hardware. Poiché gli handlers non sono preemptive l'utilizzo di sistemi event-based è consigliato nel caso in cui i processi fanno un alto uso di I/O, massimizzando cosi il rendimento della CPU

Un importante funzionalità che offre il sistema operativo sta nella possibilità di caricare i moduli a run-time. Questo facilita la riprogrammazione del codice risultando di fondamentale importanza nelle reti con un elevato numero di sensori. Molti sistemi operativi richiedono il caricamento completo dell'immagine binaria mentre Contiki permette di sostituire dinamicamente programmi e servizi rendendo il sistema flessibile e leggero. Per fare ciò utilizza l'Executable Linkable Format (ELF), uno standard utilizzato nell'ambito dei sistemi embedded che consiste in un header formato da un insieme di sezioni, una per il codice binario (.text), una per i dati allocati staticamente (.data) e una opzionale contenente la tabella dei simboli e delle stringhe (.symtab e .strtab). Per caricare un modulo ELF viene utilizzata la funzione elfloader_load() la quale ritorna ELFLOADER_OK per avvenuto caricamento del modulo o una serie di altri stati nel caso di caricamento fallito

```
ELFLOADER_BAD_ELF_HEADER      1
ELFLOADER_NO_SYMTAB           2
ELFLOADER_NO_STRTAB           3
ELFLOADER_NO_TEXT             4
ELFLOADER_SYMBOL_NOT_FOUND    5
ELFLOADER_SEGMENT_NOT_FOUND   6
ELFLOADER_NO_STARTPOINT       7
```

Figura 2.11: Stati di ELF

Capitolo 3

Implementazione

Questo capitolo illustrerà il lavoro svolto. Verranno descritte le componenti hardware utilizzate, l'ambiente di lavoro per lo sviluppo e i vari procedimenti di creazione, configurazione della rete con i relativi metodi adottati per la fase di comunicazione.

3.1 Hardware utilizzato

In questa sezione andremo ad analizzare tutte le componenti hardware utilizzate per costruire il nodo sensore che sarà utilizzato per lo sviluppo del progetto di tesi. Tutte le componenti hardware utilizzate sono state prodotte dall'Azienda STMicroelettronics.

3.1.1 STM32 Nucleo Board

Le schede Nucleo fanno parte di una famiglia di schede adatte per la progettazione di applicazioni con microprocessori ARM Cortex. In base alle capacità di memoria e alle performance che si vogliono avere esistono decine di schede appartenenti a questa famiglia.

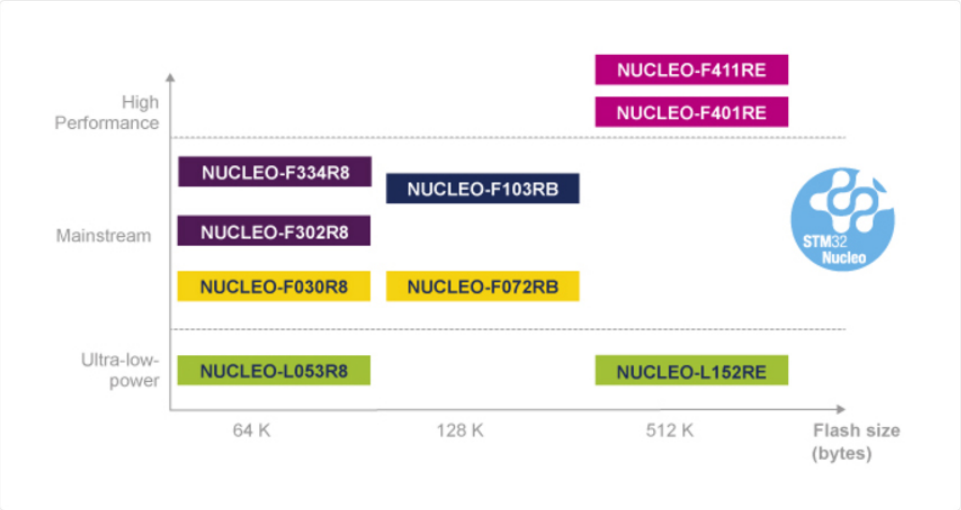

Figura 3.1: Classificazione schede Nucleo

La scheda utilizzata in questo lavoro e la NUCLEO-F401RE [10] le cui caratteristiche principali sono:

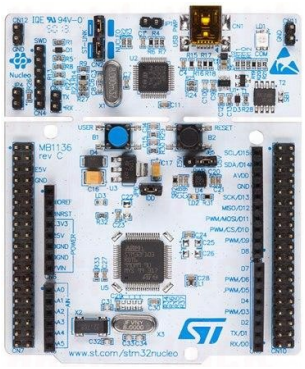

Figura 3.2: NUCLEO-F401RE

Processore: ARM Cortex M4

Alimentazione: 3.3 V, 5 V, 7 V fino a 12 V

Peso: 50 g

Header: compatibile con Arduino UNO rev.

Programmatore: ST-LINK/V2-1 che può essere usato separatamente alla scheda, utilizzato per caricare il firmware ed eseguire il debugging

Alimentazione: tramite USB o alimentazione esterna

LED: Tre di cui uno programmabile dall'utente, uno tricolore per la comunicazione e uno per l'alimentazione

Pulsanti: Due di cui uno programmabile dall'utente e uno per il reset

Inoltre ST fornisce una serie di driver a basso livello chiamati **HAL (Hardware Abstraction Layers)** che offrono un vasto set di APIs per interagire facilmente con le applicazioni dei livelli superiori. Questi driver includono una serie di moduli e ogni modulo e collegato ad una specifica periferica. (UART, USART, I^2C).

3.1.2 Spirit Sub 1GHz RF

Il modulo utilizzato per la comunicazione radio si chiama Spirit [12] ed è montato sulla board STNucleo F401RE. Le sue caratteristiche sono:

Figura 3.3: Nucleo Spirit IDS01A1

Frequenza operativa: 150-174 MHz, 300-348 MHz, 387-470 MHz, 779-956 MHz;

Modulazione: 2-FSK, GFSK, MSK, GMSK, OOK, and ASK;

Data-rate: da 1 a 500 kbps.

Consumo: 9 mA in Ricezione e 21 mA in Trasmissione con potenza massimo di +11 dBm

Output power: +16 dBm

Buffer: 96 byte per ogni buffer (TX e RX)

Ritrasmissione: automatica

Crittografia: AES 128-bit

Wake-up: interno tramite timer o esterno tramite eventi

Temperature operative: da -40 $^{\circ}C$ a 85 $^{\circ}C$

Per quanto riguarda il meccanismo di accesso al canale, Spirit fa uso del CSMA/CA, meccanismo basato sulla regola dell'ascolto del canale prima della trasmissione. Questo evita che più dispositivi usino il canale per trasmettere incrementando la probabilità di ricezione dei dati. Il CSMA non è abilitato quando il dispositivo è in modalità di ricezione. L'operazione di rilevazione del canale si basa sulla potenza del segnale ricevuto (RSSI, received signal strength indicator). Questo valore è misurato da due antenne le quali si alternano nella fase di ricezione per poter permettere ad un algoritmo di scegliere quella con il valore RSSI più alto.

Riguardo le modalità operative il modulo radio è provvisto di un controller principale che gestisce l'alternarsi di due operazioni, ricezione (RX) e trasmissione (TX). Inoltre gestisce 6 modalità operative.

SHUTDOWN è la modalità che permette di risparmiare più batteria ma in questa modalità le configurazioni e i dati salvati vanno persi. Lo stato base

è READY in cui il dispositivo è in attesa di un eventuale cambio di stato. Da questo stato si può passare allo stato di ricezione/trasmissione solo passando tramite lo stato LOCK e questa operazione viene gestita tramite un comando utente. Dopo che un dato è stato trasmesso o ricevuto si può passare allo stato base (Ready) o andare in stato di SLEEP. Se dallo stato di READY non avvengono timeout si passa allo stato STANDBY il quale permette il consumo di batteria minore in assoluto mantenendo attivi i registri e le configurazioni.

3.1.3 Modulo dei sensori

La board X-NUCLEO-IKS01A2 [12] fornisce informazioni utili ai fini del monitoraggio. È equipaggiata di connettori compatibili con Arduino UNO r3 e contiene al suo interno un accellerometro (LSM6DSL), un giroscopio (LSM303AGR), un magnetometro e sensori di umidità, temperatura (HTS221) e pressione (LPS22HB). Si interfaccia con il microcontrollore tramite il pin I^2C. Inoltre la board fornisce una libreria per lo sviluppo con esempi per tutti i sensori compatibili con il firmware STM32Cube.

In questo studio sono stati presi in considerazione solo i sensori di umidità, temperatura e pressione motivo per il quale quelli non utilizzati sono stati disattivati fisicamente nella board per permettere una consumo minore.

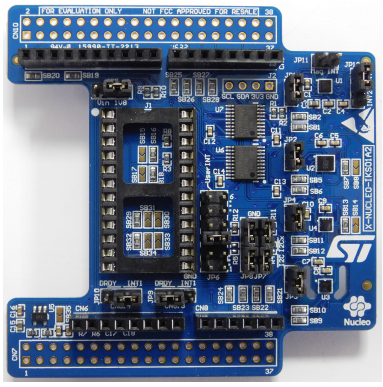

Figura 3.4: Nucleo IKS01A2

3.2 Descrizione del progetto

La topologia scelta per la progettazione di questa rete è quella a Mesh ma con una piccola modifica, ovvero, si è scelto di non considerare la differenza tra nodi Router e nodi Foglia (End Device). Nel nostro caso abbiamo voluto considerare tutti i nodi come Router, in questo modo ogni nodo ha la facoltà sia di inviare che di ricevere dati. La funzione dei nodi Router permette la comunicazione multi-hop tra due dispositivi che vogliono comunicare ma per motivi fisici si trovano troppo distanti l'uno dall'altro.

La rete è stata installata in una struttura universitaria allo scopo di monitorare i vari ambienti dove i singoli sensori sono stati collocati. Le componenti hardware per lo sviluppo sono state fornite dall'azienda STMicroelettronics in collaborazione con l'Università. I nodi-sensori utilizzati per lo studio sono cinque in totale. Tutti e cinque hanno le stesse caratteristiche sia hardware che software tranne un nodo in particolare il quale svolge una funzione in più rispetto gli altri ovvero quella di Coordinatore della rete. Il nodo Coordinatore avrà il compito di iniziare la procedura di scoperta dei dispositivi, settandosi come radice di una struttura Grafo che verrà utilizzata per il routing della rete, di conseguenza tutte le informazioni che circolano all'interno della rete hanno come destinazione il nodo Coordinatore il quale si occuperà di elaborarle ed utilizzarle.

Ogni nodo-sensore è alimentato a presa elettrica ed è dotato di un modulo che permette di acquisire informazioni su temperatura, umidità e pressione dell'ambiente circostante. Periodicamente ogni nodo invia lo stato dei propri sensori al Border Router il quale si occuperà di smistarli verso un server che li salverà e li classificherà in un apposito database. Oltre all'aggiornamento periodico da parte dei dispositivi il Border Router può richiedere le informazioni sui sensori in un determinato istante tramite un messaggio di GET che viaggerà nella rete e tramite il multi-hop riuscirà ad arrivare a tutti i nodi i quali risponderanno con le informazioni richieste.

Di seguito in figura è presente un immagine esplicativa della rete 6LoWPAN realizzata

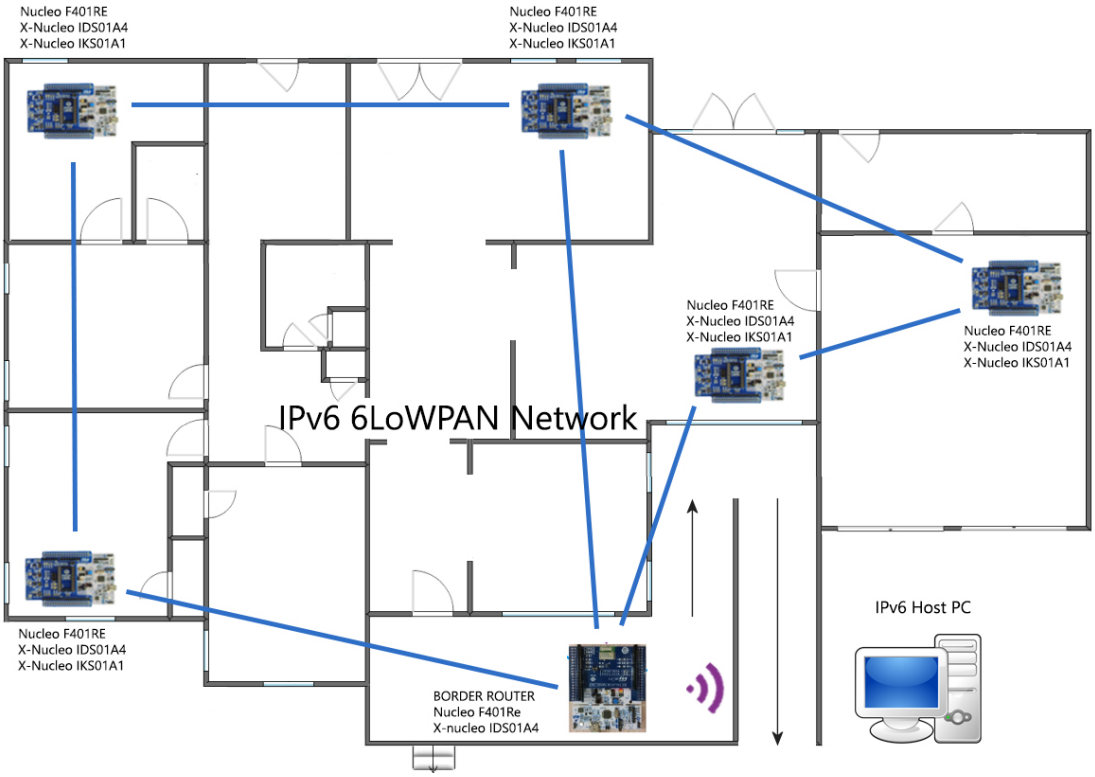

Figura 3.5: Rete 6LoWPAN

3.3 IDE utilizzato

Per implementare la rete si è scelto di utilizzare il software gratuito AC6 System Workbench, multi-piattaforma basato su Eclipse il quale supporta la programmazione su board STM32. Può essere utilizzato su Windows, Linux e OS X ed è facilmente scaricabile dal sito www.openstm32.org.

Le funzionalità che offre sono le seguenti:

- Supporto per microcontrollori STM32 Nucleo
- Compilatore GCC C/C++
- Debugger GDB

- Compatibile con i plug-in di Eclipse

Una volta scaricato e installato, il software è stato configurato per poter interagire con il sistema operativo. Contiki, adesso alla versione 3.0 è scaricabile e installabile in due modalità. La prima riguarda Instant Contiki, una macchina virtuale eseguibile su ambiente Linux che comprende i tools di sviluppo, simulatori e compilatori in un unico file eseguibile con il player VMWare. La seconda modalità consiste nell'utilizzare progetti di esempio presenti sui repository di Github. Si è scelto di adottare la seconda modalità basandoci su un progetto esempio di ST che implementa in modo semplice una comunicazione tra due nodi utilizzando il protocollo UDP.

Una volta importato il progetto all'interno del software di sviluppo come possiamo vedere in figura 3.5 è presente una configurazione organizzata in cartelle

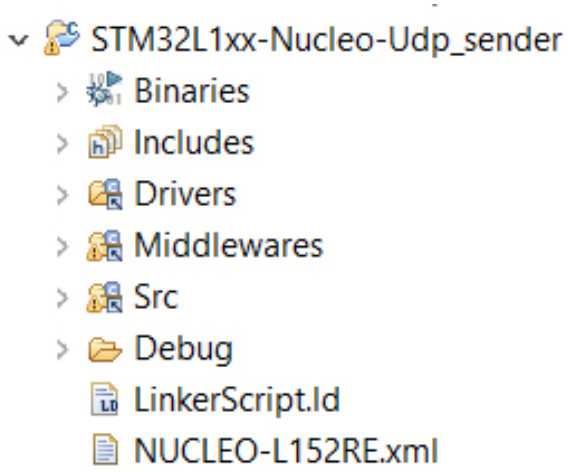

Figura 3.6: Struttura progetto

Tutte queste cartelle sono presenti nello stesso ordine in cui sono strutturate nella cartella del progetto, tranne le cartelle Binaries e Include. La cartella Binaries contiene tutti i file eseguibili (.elf), la cartella Includes contiene una lista di percorsi relativi a file inclusi nel progetto, la cartella Drivers

contiene i driver della Nucleo board (HAL), del modulo radio e dei sensori. La cartella Middlewares contiene tutti i servizi che il sistema operativo mette a disposizione (ip, rpl, timers) e infine la cartella Src contiene i file sorgenti.

3.4 Configurazione nodo

Il nodo sensore che andremo ad utilizzare sarà quindi formato dalla Nucleo board (F401RE) alla quale verrà connesso il modulo radio (Spirit Sub 1Ghz) tramite gli appositi pin. Successivamente, al di sopra del modulo radio, verrà inserita la board dei sensori (IKS01A1).

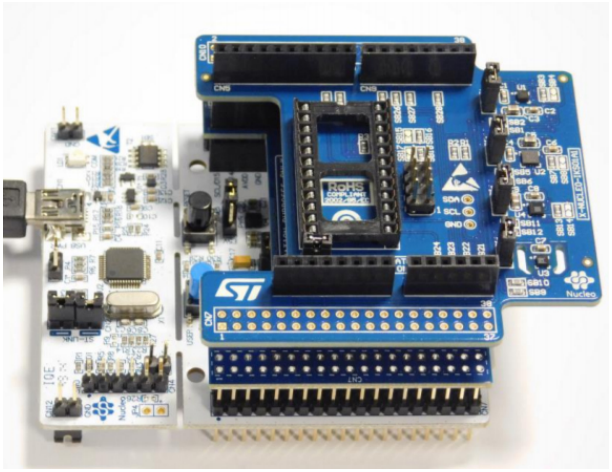

Figura 3.7: Struttura nodo-sensore

Il nodo è alimentato tramite corrente elettrica ma è possibile anche alimentarlo tramite una batteria nel caso lo si volesse utilizzare in ambienti esterni dove le prese elettriche scarseggiano. Il nodo-sensore è stato collegato tramite un cavo mini USB ad un PC, utilizzando un emulatore terminale "Tera Term" per poter visualizzare gli output in maniera comoda.

Prima di creare la rete ogni nodo deve configurare una serie di parametri per poter utilizzare il modulo radio per la comunicazione. Tramite questi parametri è possibile depotenziare il dispositivo diminuendo il POWER_DBM

che nella nostra configurazione è settato alla massima potenza. Di conseguenza è possibile anche modificare la modulazione scegliendone una tra quelle supportate da Spirit cosi come anche le soglie applicate per la trasmissione e la ricezione. La frequenza invece è l'unico valore che non può essere modificato. Nella tabella 3.1 è presente la configurazione che ho scelto di adottare nello sviluppo del progetto

$\#definePOWER\_DBM$	11.6	dBm
$\#defineCHANNEL\_SPACE$	$20e3$	–
$\#defineFREQ\_DEVIATION$	$20e3$	–
$\#defineBANDWIDTH$	$100.0e3$	–
$\#defineMODULATION\_SELECT$	FSK	–
$\#defineDATARATE$	38400	bit/s
$\#defineRSSI\_RX\_THRESHOLD$	-118.0	dBm
$\#defineRSSI\_TX\_THRESHOLD$	-107.0	dBm
$\#defineBASE\_FREQUENCY$	$868.0e6$	MHz

Tabella 3.1: spirit.h

In questa fase, poiché non è ancora possibile identificare il nodo-sensore nella rete, ogni dispositivo si auto-genera un indirizzo IPv6 (Unique local address), visibile solo all'interno della rete di appartenenza che utilizzerà nella fase successiva per essere identificato e per poter comunicare con un altro dispositivo. La generazione degli indirizzi IP avviene tramite la seguente funzione:

```
static uip_ipaddr_t *
set_global_address(void)
{
  static uip_ipaddr_t ipaddr;
  int i;
  uint8_t state;

  uip_ip6addr(&ipaddr, UIP_DS6_DEFAULT_PREFIX, 0, 0, 0, 0, 0, 0, 0);
  uip_ds6_set_addr_iid(&ipaddr, &uip_lladdr);
  uip_ds6_addr_add(&ipaddr, 0, ADDR_AUTOCONF);

  printf("IPv6 addresses: ");
  for(i = 0; i < UIP_DS6_ADDR_NB; i++) {
    state = uip_ds6_if.addr_list[i].state;
    if(uip_ds6_if.addr_list[i].isused &&
       (state == ADDR_TENTATIVE || state == ADDR_PREFERRED)) {
      uip_debug_ipaddr_print(&uip_ds6_if.addr_list[i].ipaddr);
      printf("\n");
    }
  }

  return &ipaddr;
}
```

Figura 3.8: Auto generazione indirizzo IPv6

L'indirizzo IPv6 che viene generato è un indirizzo di 128 bit costituito da un prefisso UIP_DS6_DEFAULT_PREFIX impostato al valore FD00 che occupa i primi 7 bit (i restanti 57 bit sono impostati a zero) per indentificare l'indirizzo come unicast-local e da un identificatore di interfaccia (indirizzo MAC del dispositivo) che occupa i restanti 64 bit. In questo modo ogni nodo-sensore sarà identificato da un indirizzo, univoco all'interno della rete di appartenenza.

- ESEMPIO indirizzo IPv6: fd00::500:f8ff:70d6:8f29

Come ultima operazione prima di passare alla fase successiva ogni nodo registrerà una connessione UDP a cui sarà collegata una funzione di callback la quale verrà chiamata ogni volta che un pacchetto e in transito verso la sua destinazione.

```
int simple_udp_register(struct simple_udp_connection *c,
                        uint16_t local_port,
                        uip_ipaddr_t *remote_addr,
                        uint16_t remote_port,
                        simple_udp_callback receive_callback);
```

Figura 3.9: Registrazione connessione UDP

3.5 Creazione della rete

Una volta che i nodi-sensori sono stati configurati correttamente, tramite la scoperta dei dispositivi, si passa alla fase successiva che consiste nella creazione della rete. Per questa fase viene utilizzato il protocollo RPL il quale, come detto precedentemente, permette di creare una struttura logica su di una fisica già esistente. Questa struttura è rappresentata da un grafo orientato aciclico il quale avrà come nodo radice il Border Router, un nodo specializzato all'interno della rete che si occuperà di iniziare il processo di scoperta dei dispositivi e di instradare l'informazione che circola nella rete verso l'esterno. Ogni messaggio avrà come destinazione finale il Border Router.

La procedura inizia con la seguente funzione:

```
static void
create_rpl_dag(uip_ipaddr_t *ipaddr)
{
  struct uip_ds6_addr *root_if;

  root_if = uip_ds6_addr_lookup(ipaddr);
  if(root_if != NULL) {
    rpl_dag_t *dag;
    uip_ipaddr_t prefix;

    rpl_set_root(RPL_DEFAULT_INSTANCE, ipaddr);
    dag = rpl_get_any_dag();
    uip_ip6addr(&prefix, UIP_DS6_DEFAULT_PREFIX, 0, 0, 0, 0, 0, 0, 0);
    rpl_set_prefix(dag, &prefix, 64);
    PRINTF("created a new RPL dag\n");
  } else {
    PRINTF("failed to create a new RPL DAG\n");
  }
}
```

Figura 3.10: Inizializzazione del processo di scoperta dei dispositivi

il Border Router crea la struttura DAG e si setta come nodo radice impostando il proprio rank a zero. Quando un nodo non fa parte della rete e vuole scoprirla utilizza i messaggi DIS (DODAG Information Solicitation) per sollecitare l'invio di un messaggio DIO da un altro nodo al fine di scoprire la presenza di nodi vicini. Il messaggio DIO conterrà l'indirizzo IPv6 del nodo che l'ha generato con il relativo rank, che è pari a zero per la radice e sale man mano che si scende verso le foglie. Un nodo che riceve un messaggio DIO controllerà il proprio rank con quello contenuto all'interno del messaggio e solo se risulta maggiore si collegherà ad esso inserendo l'indirizzo IPv6 nel campo genitore e aggiornando il proprio rank pari a quello del padre più un offset. Un nodo può ricevere più messaggi di tipo DIO, in questo caso ci sono più nodi candidati come genitori ma la scelta dovrà ricadere solo su uno in particolare. La scelta del "preferred parent" viene effettuata considerando la qualità del collegamento (ETX), in questo modo il genitore scelto sarà colui che avrà il valore minimo di ETX. Quando un nodo non fa ancora parte della rete e deve collegarsi per la prima volta, la metrica ETX è calcolata utilizzando il valore di RSSI (ETX=RSSI) misurato sul nodo genitore, viceversa se un nodo fa già parte della rete e deve scegliere se collegarsi ad un altro genitore o meno il valore di ETX verrà scelto in base alla qualità del link misurata come il numero totale atteso di trasmissioni richieste per consegnare un pacchetto. ETX è riassunto nella seguente formula:

$$ETX = \frac{1}{PRR\_down * PRR\_up} \tag{3.1}$$

dove PRR_down indica la qualità del link misurata sul nodo ricevente, viceversa PRR_up misura la qualità del link misurata sul nodo che invia. Il Packet Reception Rate (PPR) è calcolato nel seguente modo:

$$PRR = \frac{Numero\_dipacchetti\_ricevuti}{Numero\_di\_pacchetti\_inviati} \tag{3.2}$$

La metrica ETX potrà assumere valori che vanno da 1 (collegamento ottimo) a valori maggiori di uno che indicano collegamenti sempre peggiori.

Appena un nodo sceglie il proprio genitore, viene inizializzato un timer DAO_Timer la cui durata cresce fino a stabilizzarsi ad un valore fisso. Quando il DAO_Timer termina il nodo dovrà avvisare il proprio padre tramite l'invio di un messaggio DAO (Destination Advertisement Object). Questo messaggio permette di riempire le tabelle di routing dei dispositivi utilizzando la modalità STORING delle tabelle. Un messaggio DAO viene inviato in unicast dal nodo figlio verso il genitore il quale lo riceve e memorizza l'indirizzo IPv6 del nodo da cui l'ha ricevuto nella tabella di routing, inserendo una nuova rotta. Questo messaggio viene propagato fino alla radice la quale avrà una tabella contenente tutti i nodi della rete.

Per mantenere consistenza nella rete vengono utilizzati messaggi di tipo No-Path DAO. Questa tipologia di messaggio viene inviata in risposta ad un evento (nodo guasto o scarico) per notificare un nuovo parent.

3.6 Comunicazione Multi-hop

La comunicazione tra due nodi è possibile solo in questa fase, dove il DODAG è completo e ogni nodo del DODAG ha riempito la propria tabella di routing. Nell'ambito delle reti di sensori, dato l'ampio range d'azione dei nodi-sensore è molto probabile che due nodi non riescano a comunicare poiché la potenza radio dei dispositivi risulta troppo debole per poter comunicare a certe distanze. Per risolvere questo problema è stato quindi introdotto il concetto di multi-hop. Il multi-hop consiste in un passaparola di informazioni che permette a due nodi, lontani tra di loro, di poter comunicare.

Proprio per questo motivo si è scelto di adottare una topologia della rete che identifichi tutti i nodi come nodi router non andando a considerare la differenza fra Router e End Device. In questo modo ogni nodo che riceve un dato che non è destinato a lui potrà inoltrarlo in modo che arrivi alla giusta destinazione. L'approccio utilizzato per il routing dei pacchetti è di tipo route-over come spiegato nel capitolo precedente e viene gestito da RPL

in base alla struttura DODAG e alle tabelle di routing contenenti gli indirizzi IPv6 dei nodi raggiungibili.

3.6.1 Routing verso l'alto e routing verso il basso

Il routing verso l'alto è immediato poiché ogni nodo possiede un solo collegamento verso un determinato "preferred parent". Di conseguenza, per inviare un dato verso il Border Router basta inoltrarlo al nodo genitore presente nel campo PREFERRED_PARENT.

```
addr = rpl_get_parent_ipaddr(dag->preferred_parent);
if (addr != NULL) simple_udp_sendto(&unicast_connection, &message, sizeof(message), addr);
else printf("Indirizzo non valido\n");
printf("Invio del messaggio al padre: "); uip_debug_ipaddr_print(addr); printf("\n");
```

Figura 3.11: Invio di un messaggio al nodo genitore

In questo modo il genitore che riceverà il messaggio si limiterà ad inoltrarlo al proprio padre e così, il messaggio arriverà a destinazione finale e sarà consegnato e ricevuto dal Border Router.

Il routing verso il basso invece è gestito da RPL tramite il point-to-multipoint. Il Border Router può richiedere che tutti i nodi forniscano le informazioni sullo stato dei loro sensori in un determinato istante. In questo modo invia ai figli che stanno ad un passo da lui un messaggio dove richiede i valori dei sensori. I nodi che riceveranno questo messaggio si limiteranno a rispondere al nodo da cui l'hanno ricevuto con un SetValue(ID,umidità,temperatura,pressione) inviando le informazioni sui sensori e inoltrando il messaggio ricevuto ai propri figli in accordo con i percorsi forniti nelle tabelle di routing.

```
static void children_multicast(){
    int j = 0;
    our_message_t m;
    uip_ds6_nbr_t *nbr;
    for(nbr = nbr_table_head(ds6_neighbors);
        nbr != NULL;
        nbr = nbr_table_next(ds6_neighbors,nbr)){

            printf("Vicino %d: ", j);
            uip_debug_ipaddr_print(&nbr->ipaddr);
            simple_udp_sendto(&unicast_connection, &m, mSize, &nbr->ipaddr);
            j += 1;
            printf("\n");
    }

    printf("Messaggi inviati a %d figli\n", j);
}
```

Figura 3.12: Invio di un messaggio al nodo genitore

3.6.2 Messaggi

Per gestire le comunicazioni si è scelto di tenere in considerazione due tipologie di messaggi:

1. GetALL(): questo messaggio viene generato dal Border Router per richiedere i dati sui sensori. I nodi che ricevono questo messaggio svolgono due operazioni. Lo inoltrano ai nodi nel proprio sotto albero e rispondono al nodo da cui l'hanno ricevuto con un messaggio di tipo SetValue(ID, sensorValue1, sensorValue2, sensorValue3) dove l'id rappresenta l'indirizzo IPv6 del dispositivo, e i restanti valori rappresentano i valori di temperatura, umidità e pressione della board dei sensori (IKS01A1)

2. Update(): questo messaggio viene generato dai nodi ogni T tempo, dove T rappresenta un timer e può assumere vari valori. Il valore scelto di dafault è 30 minuti. Questo messaggio viene generato da ogni nodo allo scadere del tempo T ed inoltrato verso il Border Router il quale, una volta acquisiti i dati sui sensori li memorizzerà in un apposita tabella.

Capitolo 4

Valutazioni

In questo capitolo verranno presentati i risultati relativi ai test effettuati sulla rete. Come primo test abbiamo voluto analizzare la tecnologia radio Spirit, per poi studiare la comunicazione tra i dispositivi andando a capire quanto incide il multi-hop sulle metriche calcolate.

4.1 Analisi Spirit Sub 1GHz RF

In questa analisi abbiamo voluto testare la tecnologia di comunicazione radio per capirne i vantaggi e i limiti. Per effettuare i test abbiamo utilizzato due schede NucleoF401 dotate di modulo radio, disposte ad una distanza fissa di circa 3 metri. Le metriche scelte per l'analisi sono due, il Goodput cioè la quantità di dati trasferita nell'unità di tempo (KB/s) senza tenere in considerazione i possibili overhead e il Delay cioè il ritardo nella consegna dei pacchetti. Queste metriche sono state calcolate andando a considerare la variazione del carico di lavoro, ovvero il numero di messaggi inviati nell'unità di tempo (pkt/s) e il datarate espresso in KB/s. Poiché con un payload superiore a 70 byte si va incontro ad una frammentazione del pacchetto, abbiamo scelto di fissare la dimensione del messaggio a 70 byte per il payload.

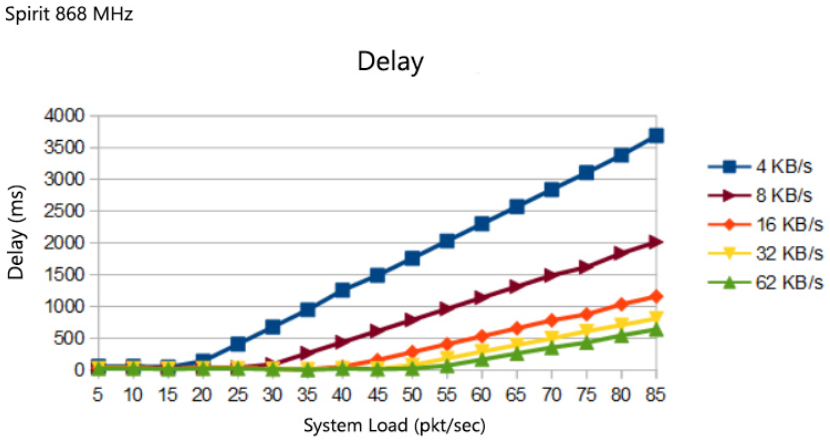

Figura 4.1: Delay SPIRIT1

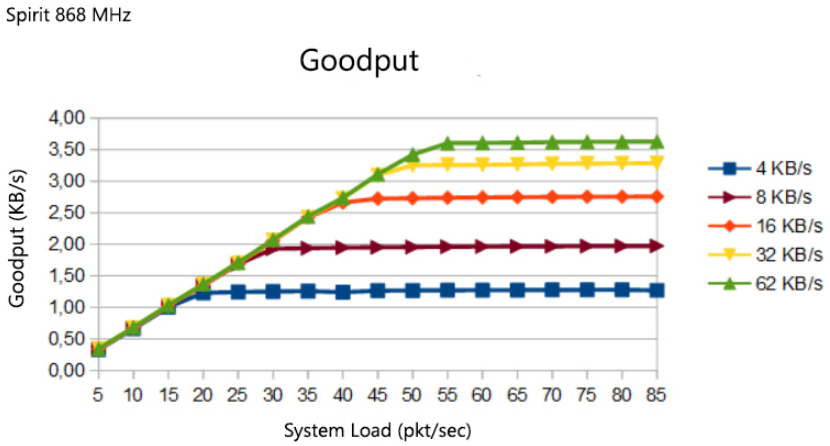

Figura 4.2: Goodput SPIRIT1

Si nota che il Goodput sale fino a stabilizzarsi ad un valore massimo di 3.6 Kbps con un massimo di 55 messaggi al secondo. Per quanto riguarda il delay, vediamo che aumenta proporzionalmente all'aumentare sia del carico di lavoro che del datarate. Il valore di Goodput che abbiamo misurato è nettamente inferiore a quello indicato nelle specifiche del modulo radio (62.5 KBps) per cui il problema potrebbe essere relativo alla dimensione del buffer che Contiki utilizza per la ricezione e trasmissione dei dati. Abbiamo quindi

effettuato lo stesso test andando ad aumentare la dimensione del buffer ma i risultati non sono migliorati, sia il Delay che il Goodput rimangono invariati. Abbiamo notato che il buffer gestito da Contiki, che noi pensavamo fosse simile ad una coda, riesce a processare un solo pacchetto alla volta. Di conseguenza anche aumentandone la dimensione non si riesce a processare più pacchetti contemporaneamente. Alla luce di ciò sembra che i dati ottenuti rispecchino le effettive limitazioni di questa tecnologia.

4.2 Analisi multi-hop

In questa analisi abbiamo voluto verificare quanto incide la comunicazione multi-hop nelle prestazioni. I nodi usati per il test sono quattro, dotati del modulo radio, indispensabile per la comunicazione. Le metriche scelte per questa valutazione sono le stesse dell'analisi precedente ovvero il Goodput cioè la quantità di dati trasferita nell'unità di tempo (KB/s) senza tenere in considerazione i possibili overhead e il Delay cioè il ritardo nella consegna dei pacchetti. I quattro nodi sono stati disposti ad una distanza fissa di 5 metri in modo da permettere un cammino multiplo di tre hop.

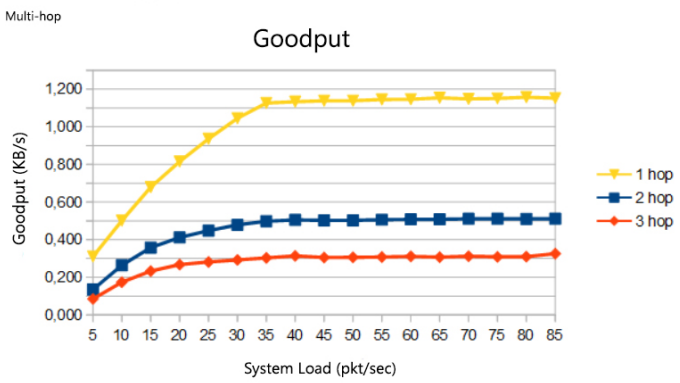

Figura 4.3: Goodput della rete

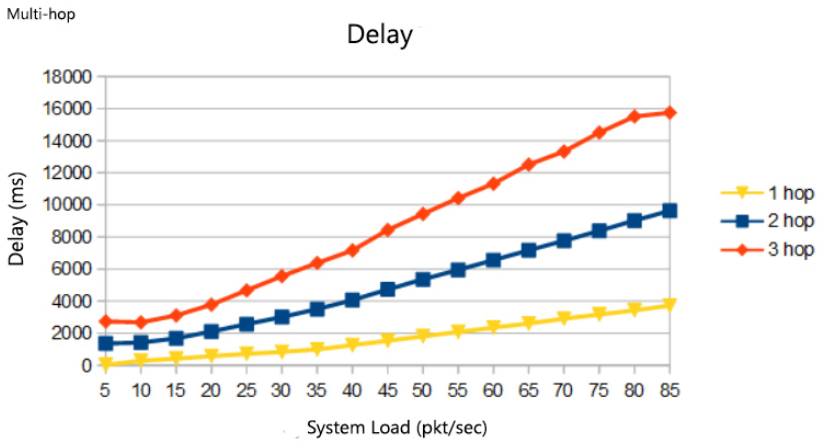

Figura 4.4: Delay della rete

Si nota che le prestazioni hanno lo stesso andamento per tutte le tre configurazioni e principalmente aumentano al diminuire del numero di hop nella catena per quanto riguarda il Goodput, il quale ha valore più alto con numero di hop uguale a uno. Questi grafici ci fanno capire che la scelta del numero di hop dipende da una serie di fattori come ad esempio le interferenze ambientali che limitano molto la comunicazione. I grafici mostrano il giusto trade-off tra le prestazioni della rete e il numero di hop. Ad esempio, con basse interferenze si preferisce comunicare in singol-hop avendo un Goodput maggiore e Delay minore,viceversa con alte interferenze si preferisce, per sicurezza, avvicinarsi più vicino possibile verso a destinazione inviando l'informazione al nodo più prossimo nella catena finale che porta a destinazione.

Nella rete presa in esame, la scelta del multi-hop sembra essere la soluzione migliore poiché trattandosi di una rete che si limita a monitorare l'ambiente, in media il numero di messaggi che circola risulta essere molto basso, riducendo al minimo la congestione, il delay e la percentuale di pacchetti persi.

Conclusioni

Lo studio effettuato in questo lavoro di tesi si è incentrato sulla creazione di una rete di sensori low-energy per il monitoraggio di un ambiente indoor. Lo studio inizia con una panoramica su quello che è lo stato dell'arte, per renderci conto del progresso tecnologico che sta caratterizzando le reti di sensori. Dopo aver illustrato i vari standard che le regolamentano sono state descritte le tecnologie utilizzate per lo sviluppo della tesi. In particolare tramite il protocollo 6LoWPAN è stato fornito uno stack protocollare progettato ad-hoc per questo tipo di reti che ha permesso di assegnare ai dispositivi della rete un identificativo, il quale, tramite il protocollo di routing RPL ha permesso di creare una struttura logica tramite la quale i dispositivi possono comunicare tra loro. Il terzo e il quarto capitolo hanno concluso il lavoro entrando più nel dettaglio, descrivendo le problematiche da affrontare e le soluzioni adottate per lo sviluppo della rete. È stato descritto il meccanismo con cui i dispositivi si auto configurano per poter comunicare. Il sistema infatti è composto da un nodo particolare (Edge Router) che interroga periodicamente i nodi-sensori i quali monitorano l'ambiente analizzando grandezze fisiche come temperatura, umidità e pressione. Infine, nel capitolo valutativo è stata analizzata la componente radio utilizzata per poi passare ad un analisi comparativa sulla comunicazione andando a capire qual'è il giusto trade-off tra una comunicazione di tipo singol-hop con una di tipo multi-hop tramite l'uso di metriche come il Goodput e la Latenza.

Questo lavoro di tesi vuole essere un punto di inizio per una serie di possibili sviluppi futuri che andrebbero sicuramente a migliorare molti aspetti,

che per motivi di tempo non sono stati trattati.

Il primo aspetto è il fattore energy saving. I nodi della rete infatti sono tutti alimentati tramite presa elettrica poiché non è stato possibile implementare alcun meccanismo di duty cycling, ovvero la percentuale di tempo in cui un dispositivo passa da una fase di sleep ad una fase in cui svolge la propria attività, permettendo cosi un risparmio energetico. In questo modo i dispositivi potranno essere alimentati a batterie permettendo una disposizione più comoda della rete, soprattutto in condizioni dove le prese o meglio la corrente elettrica scarseggiano.

Il secondo aspetto riguarda il modo in cui vengono gestite le operazioni di ricezione e trasmissione dei dati. Nelle reti di sensori vengono utilizzati particolari tipologie di thread chiamati protothread, creati per dispositivi con stretti vincoli di memoria. Nella mia implementazione l'operazione di trasmissione e ricezione è affidata a due thread differenti i quali si alternano per svolgere il proprio lavoro fornendo una sorta di pseudo parallelismo. Uno sviluppo futuro potrebbe consistere nell'implementazione di un parallelismo puro che indubbiamente limiterebbe i ritardi nella rete.

Bibliografia

[1] Tareq Alhmiedat, Anas Abu Taleb, and Mohammad Bsoul. A study on threats detection and tracking systems for military applications using wsns. International Journal of Computer Applications, 2012.

[2] Sang Hyuk Lee, Soobin Lee, Heecheol Song, and Hwang Soo Lee. Wireless sensor network design for tactical military applications: remote large-scale environments. In Military Communications Conference, 2009. MILCOM 2009. IEEE, pages 1-7. IEEE, 2009.

[3] Shaharil Mad Saad, Abdul Rahman Mohd Saad, Azman Muhamad Yusof Kamarudin, Ammar Zakaria, Ali Yeon Md Shakaff. Indoor Air Quality Monitoring System using Wireless Sensor Network (WSN) with Web Interface. International Conference on Electrical, Electronics and System Engineering, 2013

[4] G. Werner-Allen, K. Lorinez, M. Ruiz: Deploying a Wireless Sensor Network on an Active Volcano, IEEE Internet Computing March-April 2006

[5] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson. ANALYSIS OF WIRELESS SENSOR NETWORKS FOR HABITATMONITORING. University of California at Berkeley Computer Science Department.

[6] Dr. Kanta D Devangavi. MEDICAL MONITORING APPLICATION USING WIRELESS SENSOR NETWORK. VTU PG Center Bengaluru region ,Chikkaballapur-572101

[7] David Malan, Thaddeus FulfordJones, Matt Welsh, and Steve Moulton. CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. Division of Engineering and Applied Sciences Harvard University, School of Medicine Boston University

[8] Shelby, Zach, and Carsten Bormann. 6LoWPAN: The wireless embedded Internet. Vol. 43. John Wiley e Sons, 2011.

[9] Dunkels, Adam, Bjorn Gronvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors." Local Computer Networks, 2004. 29th Annual IEEE International Conference on. IEEE, 2004.

[10] Weyn, Maarten, et al. "Survey of the DASH7 alliance protocol for 433 MHz wireless sensor communication." International Journal of Distributed Sensor Networks 2013 (2013).

[11] Augustin, Aloÿs, et al. "A study of LoRa: Long range e low power networks for the internet of things." Sensors 16.9 (2016): 1466.

[12] S2-LP, Ultra-low power, high performance, sub-1GHz transceiver

[13] IEEE 802.11 Working Group. "Part11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications." ANSI/IEEE Std. 802.11 (1999).

[14] The Institute of Electrical and Electronics Engineers, 802.15.1, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs), 2002.

[15] The Institute of Electrical and Electronics Engineers, 802.15.4, Standard for Local and metropolitan area networks, 2011

[16] Bragg, G. M., et al. "868MHz 6LoWPAN with ContikiMAC for an Internet of Things environmental sensor network." SAI Computing Conference (SAI), 2016. IEEE, 2016.

[17] Cheng, Fu, et al. "A mesh-under routing protocol for 6LoWPAN based on Imote2 platform." Communication Technology (ICCT), 2011 IEEE 13th International Conference on. IEEE, 2011.

[18] Callaway, Ed, et al. "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks." IEEE Communications magazine 40.8 (2002): 70-77.

[19] Accettura, Nicola, et al. "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things." World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a. IEEE, 2013.

[20] Vasseur, Jean-Philippe, et al. Routing metrics used for path calculation in low-power and lossy networks. No. RFC 6551. 2012.