

## UNIVERSITÀ DEGLI STUDI DI UDINE

Corso di Laurea in Ingegneria elettronica

Dipartimento di (DPIA) Dipartimento Politecnico di  
Ingegneria e Architettura

### Tesi di Laurea

### LORA E INTERNET OF THINGS: EVERYWARE SOFTWARE COME CASO DI STUDIO

Relatore:

Prof. Antonio Abramo

Laureando:

Enrico Tolotto

Correlatore:

Marco Carrer

---

Anno Accademico 2016/2017.



## **Sommario**

Grazie al progresso dell'elettronica, si prevede che la presenza di dispositivi connessi, secondo il paradigma dell'Internet delle Cose (IoT), aumenterà sostanzialmente nell'immediato futuro. Le dimensioni ridotte dei dispositivi in commercio, come sensori, attuatori, tag e tanto altro, sono particolarmente adatte a nuovi scenari applicativi. Internet of Things è la naturale evoluzione di Internet, ed è destinato a cambiare radicalmente la nostra vita futura. Diversi standard sono attualmente in competizione per aggiudicarsi la maggioranza del mercato e fornire la connettività su larga scala che è richiesta da questi dispositivi. Tra questi standard, le Low Power Wide Area Networks (LPWAN) sono in forte crescita, soprattutto grazie alla loro connettività a lungo raggio sfruttando bande di frequenza libere. Questa Tesi si focalizzerà su una delle tecnologie LPWAN predominanti: LoRa<sup>TM</sup> e l'integrazione di questa tecnologia con il framework ESF (Everyware Software Framework) sviluppato da Eurotech. Prima di tutto verrà introdotta la tecnologia LoRa ed il protocollo LoRaWAN. Successivamente, verranno presentati due nuovi bundle installabili nel framework ESF, tramite i quali sarà possibile controllare da remoto il comportamento dei dispositivi LoRa.



# Indice

<b>Indice</b>	<b>iii</b>
<b>Elenco delle figure</b>	<b>v</b>
<b>1 IoT</b>	<b>1</b>
1.1 Big Data . . . . .	2
1.2 La diffusione dell'Internet delle cose . . . . .	3
1.3 La tecnologia alla base dell'IoT . . . . .	4
<b>2 LPWAN</b>	<b>7</b>
2.1 Alla base delle reti LPWAN . . . . .	7
2.2 LPWAN . . . . .	9
2.3 NB-IoT . . . . .	10
2.4 LTE-M . . . . .	11
2.5 EC-GSM-IoT . . . . .	11
2.6 Sigfox . . . . .	12
2.7 LoRaWAN . . . . .	13
2.8 Osservazioni . . . . .	14
<b>3 LoRa</b>	<b>17</b>
3.1 Narrow Band e Spread Spectrum . . . . .	17
3.1.1 NarrowBand . . . . .	18
3.1.2 Spread spectrum . . . . .	18
3.2 Layer fisico LoRa . . . . .	19
3.2.1 Chirp Spread Spectrum . . . . .	19
3.2.2 Adattabilità . . . . .	22
3.2.3 LoRa packet . . . . .	24
3.3 LoRaWAN . . . . .	26
3.3.1 Tipologia di rete e classi di dispositivi . . . . .	27
3.3.2 Bande di frequenze . . . . .	29

3.3.3 Pacchetto LoRaWAN . . . . .	30
3.4 Adaptive Data Rate . . . . .	31
3.4.1 Sicurezza . . . . .	32
3.5 Limitazioni . . . . .	32
<b>4 Caso di studio</b>	<b>35</b>
4.1 OSGi . . . . .	36
4.1.1 Ciclo vitale dei bundle . . . . .	37
4.1.2 Module layer . . . . .	38
4.1.3 Registrazione del servizio . . . . .	38
4.2 Everyware Software . . . . .	39
4.3 Hardware utilizzato . . . . .	39
4.3.1 SX1301 . . . . .	39
4.3.2 ReliaGATE . . . . .	40
4.4 Architettura del software . . . . .	42
4.4.1 Semtech packet forwarder . . . . .	42
4.4.2 LoRa Gateway Bridge . . . . .	44
4.4.3 Bundle OSGi . . . . .	45
4.5 Misurazioni . . . . .	47
4.5.1 Osservazioni . . . . .	48
<b>5 Conclusioni</b>	<b>51</b>
<b>Bibliografia</b>	<b>53</b>

# Elenco delle figure

1.1	Catena del valore dell'IoT . . . . .	2
1.2	Numero di dispositivi per anno . . . . .	4
1.3	Layer del IoT . . . . .	5
2.1	Comparazione tipologia di reti . . . . .	9
2.2	Frequenze ISM . . . . .	10
2.3	Modalità di funzionamento NB-IoT . . . . .	11
2.4	Mappa copertura SigFox . . . . .	13
2.5	Capacità delle reti LPWA . . . . .	14
2.6	Comparazione feature reti LPWAN . . . . .	15
3.1	Comparazione tra UNB e SSP . . . . .	18
3.2	Esempio di segnale Chirp nel dominio del tempo . . . . .	20
3.3	Segnale Chirp nel dominio della frequenza . . . . .	22
3.4	Comparazione simbolica dei vari SF . . . . .	24
3.5	Struttura pacchetto Lora . . . . .	25
3.6	Pacchetto codificato dal layer fisico . . . . .	26
3.7	Comparazione del modello OSI con la struttura definita in LoRaWAN . . . . .	27
3.8	Struttura rete a stella LPWAN . . . . .	28
3.9	Stack del protocollo della rete LoRaWAN . . . . .	29
3.10	Stack del protocollo della rete LoRaWAN . . . . .	30
3.11	Simulazione prestazioni LoRa . . . . .	33
4.1	Layer OSGi . . . . .	37
4.2	Schema utilizzo servizi OSGi . . . . .	38
4.3	Interfaccia web ESF . . . . .	39
4.4	Struttura interna ricevitore SX1301 . . . . .	40
4.5	ReliaGATE 10-11 . . . . .	41
4.6	Architettura del software . . . . .	42

4.7 Installazione dell'antenna . . . . .	47
4.8 Dispositivo LoRa Mote . . . . .	48
4.9 Copertura Lora . . . . .	49

# Capitolo 1

## IoT

L'Internet delle cose (Internet of Things, IoT) è un paradigma riferito all'estensione di internet al mondo degli oggetti. Nel 1999 il ricercatore britannico Kevin Ashton, durante una presentazione, teorizzò per primo un mondo nel quale oggetti dotati di sensori interagiscono utilizzando la rete. La continua evoluzione delle tecnologie wireless e satellitari ha permesso l'ideazione di oggetti la cui maggiore connessione favoreggia il generarsi di una notevole mole di informazioni e dati. Oltre ai computer, agli smartphone e ai tablet, sempre più oggetti di uso quotidiano dispongono di una connessione ad internet. Smartwatch e smart band, lampadine e prese elettriche "intelligenti", sono già da tempo reperibili nel mercato con un prezzo accessibile alla stragrande maggioranza dei consumatori. Data la bassa complessità del hardware implementato in questi oggetti, essi necessitano di appoggiarsi ad un server esterno per l'elaborazione dei dati. Sfruttando la connessione ad internet gli oggetti riescono ad instaurare uno scambio di dati bidirezionale tra loro ed il server. Il dispositivo dopo aver convertito la grandezza fisica di suo interesse in un dato digitale, invia l'informazione al server, il quale, dopo un'elaborazione della stessa, formulerà dei comandi di risposta all'oggetto.

**In questo archetipo i dati sono considerati la materia prima**

I dati sono la materia prima del mondo dell'IoT, sono loro che giocano il ruolo principale sulla catena del valore (value chain), figura 1.1, dando una svolta importante, dal punto di vista economico, a nuovi settori quali il *data mining* e la *business analysis*. Attraverso lo studio delle informazioni raccolte si potrà aumentare l'efficienza di un servizio, oppure migliorare l'esperienza d'uso di una applicazione. Muovendoci verso un

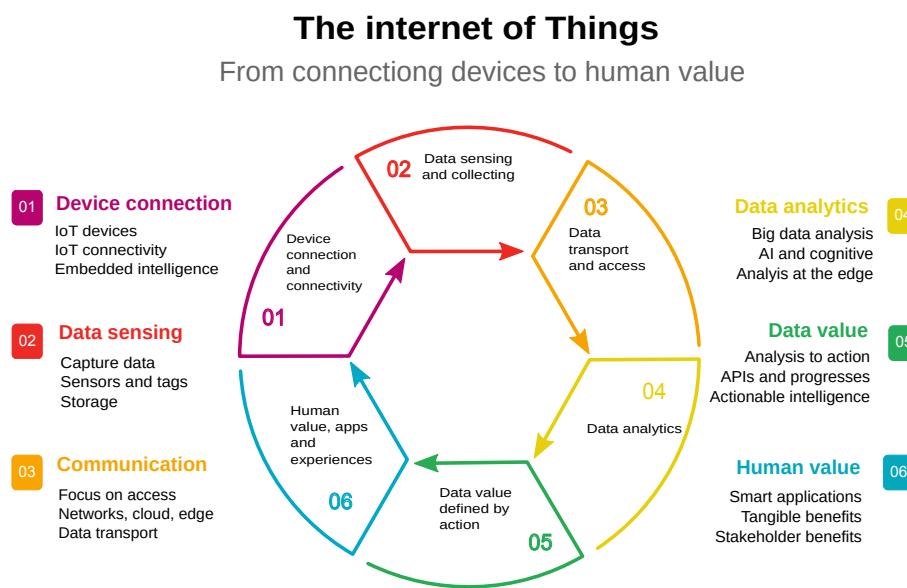


Figura 1.1: Catena del valore dell'IoT

modo progressivamente più connesso, è facile intuire il riscontrarsi di un maggior numero di problematiche riguardanti la privacy e la sicurezza. Molto spesso le aziende cercano di ridurre i costi di produzione dei gadget tecnologici, andando a limitare gli investimenti nella ricerca e sviluppo (R&D), producendo così dispositivi con un software non aggiornato o con componenti hardware di bassa qualità. Inoltre il ciclo vitale di un prodotto non sarà più determinato solo dalla rottura o dal mal funzionamento del dispositivo stesso ma, verrà ridotto dalla impossibilità di un aggiornamento del firmware. Le fal当地 di sicurezza presenti nel software ed il crescente numero di attacchi hacker rappresentano un grave pericolo per la sicurezza e la privacy dell'utente finale. È necessario quindi trovare un accordo tra case produttrici e consumatori al fine di garantire una durata minima, di alcuni anni, per quanto riguarda gli aggiornamenti software. Riducendo così quello che più comunemente viene chiamato il fenomeno della obsolescenza programmata.

## 1.1 Big Data

Oltre all'innumerabile quantità di dati che verrà prodotta da questi milioni di devices intelligenti, noi stessi, navigando il web, ne produciamo una grande quantità. Nel 2013 si è stimato che ogni secondo nel web ve-

nivano generati una quantità di dati pari a circa tremila gigabyte . Con il termine Big Data si vuole rappresentare l'insieme di tutti i dati eterogenei nei **virgola** che ogni giorno vengono prodotti e scambiati nella rete. Con il progredire della tecnologia il dataset (aggregazione di dati) a disposizione delle aziende è in continuo aumento. Secondo un articolo pubblicato da Verizon si stima che il 92% delle aziende usa meno del 25% dei dati raccolti [6]. Per di più solo la metà di esse prevede di riuscire a usufruire di una percentuale maggiore di quella attuale nei prossimi due anni. [6]. Con "Data mining" o "Data analytics" si identificano tutte le tecniche e le metodologie finalizzate all'estrazione di sapere e conoscenza partendo da una vasta mole di dati. L'enorme disponibilità, di ogni sorta di informazione, è una prospettiva che apre innumerevoli scenari di ricerca; ad esempio, predire con largo anticipo i trend del mercato, permetterebbe ad una azienda di investire in maniera più efficace le proprie risorse, andando a creare un maggiore profitto.

## 1.2 La diffusione dell'Internet delle cose

Nel 2008 il numero di oggetti con il quale è possibile connessi ad internet ha superato il numero di persone presenti nell'intero pianeta. Il continuo sviluppo tecnologico, la sempre maggior facilità d'uso e l'abbattimento dei costi, ha reso disponibile al mondo consumer tecnologie che fino a poco tempo fa erano destinate ad un uso aziendale ed universitario. Con l'avvento dell'IoT si prevede una crescita esponenziale di dispositivi connessi ad Internet. Secondo una stima da parte di Gartner il numero di smart device presenti nell'anno 2020 sarà superiore a 20 miliardi [1]. Per fronteggiare una crescita così esponenziale è d'obbligo cercare soluzioni in grado di prevenire il congestionamento della rete.

Categoria	2016	2017	2018	2019
Consumer	3963	5244,3	7063,3	12863
Business	1418	2135,4	4152,7	6171
<i>Totale</i>	6381	8380	11196	20415

Tabella 1.1: Stima di dispositivi IoT (Milioni di unità) Gartner[1]

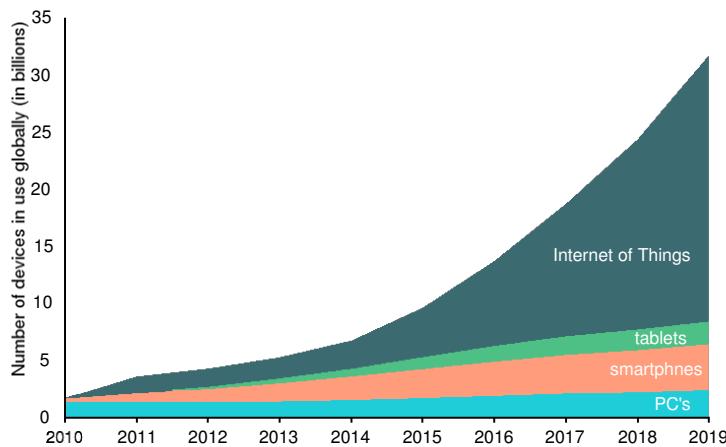


Figura 1.2: Numero di dispositivi per anno

Considerando le statistiche precedenti è comprensibile che l'IoT sia un mercato che agisce in maniera trasversale su tutti i settori. Se da tempo questi sensori "smart" sono impiegati in ambiti quali la domotica e l'automotive, con l'abbattimento del costo del singolo dispositivo e la facile implementazione di queste nuove tecnologie, nuovi mercati e nuove possibilità di investimento sono nate. Prendiamo come esempio l'agricoltura di precisione, che grazie a sensori in grado di rilevare dati riguardanti l'umidità del suolo, l'indice di piovosità oppure l'umidità fogliare, permette all'agricoltore di capire quando è il momento di intervenire per il trattamento delle colture. In questo modo è possibile dispiegare le risorse in maniera più efficace, andando ad agire solo nelle culture danneggiate. Sempre secondo quanto stimato da Gartner, ci si aspetta che entro il 2020, la cifra investita nell'industria del IoT, sarà pari a circa \$3,000,000 milioni di dollari con un investimento annuo di circa \$500,000 milioni di dollari. È utile osservare che la fetta maggiore è quella riservata al mercato consumer, dove smart TV, set-top box e smart cars saranno i prodotti maggiormente richiesti. [1].

### 1.3 La tecnologia alla base dell'IoT

Nell'industria il concetto di M2M (Machine to Machine) non è un concetto nuovo, già Kevin Ashton, durante la presentazione in cui introdusse il termine IoT, comprese le potenzialità della tecnologia RFID applicate alla

supply chain. Dal 1999 ad oggi molte cose sono cambiate, ma molte domande non hanno ancora trovato una risposta. Come accadde agli albori di Internet, quello che manca al modo dell'IoT è una standardizzazione dei protocolli e del linguaggio con cui questi oggetti dovranno comunicare. Astraendoci dal problema e portandoci ad una visione di più alto livello problema, è possibile individuare in questo paradigma tre livelli.

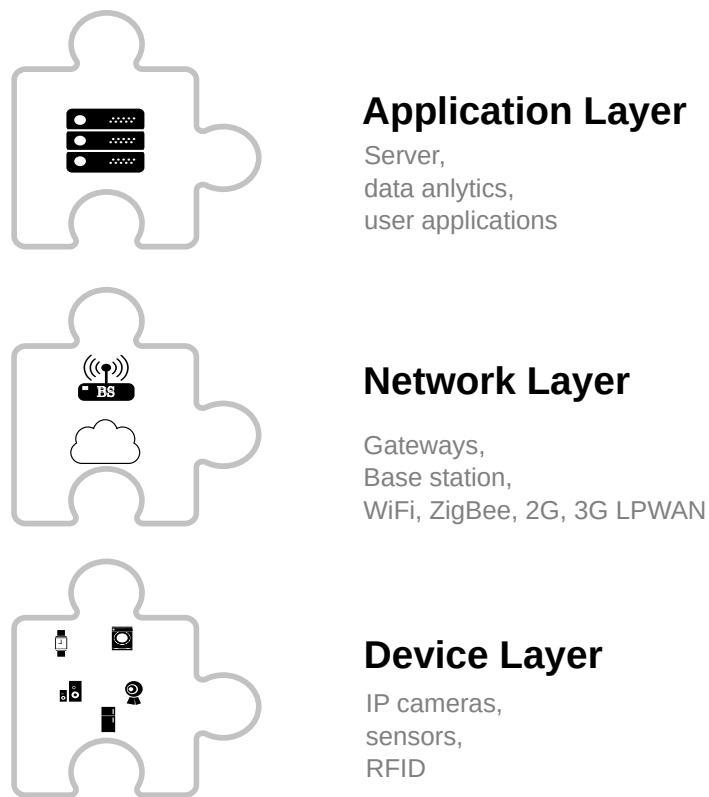


Figura 1.3: Layer del IoT

- **Device layer** o sensor layer. È il layer più basso. Esso raggruppa tutti i gli oggetti "smart". Questo layer è quello che mette in comunicazione il mondo reale con i livelli superiori. A loro spetta lo scopo di convertire una misura fisica in un dato in grado di essere interpretato da altri calcolatori. La maggior parte di questi sensori, utilizzerà una connessione Bluetooth, ZigBee, Wifi o una si baserà su una rete LPWAN per comunicare il dato al layer superiore.

- **Network layer** o mediation layer. In esso è raggruppa l'intera infrastruttura di rete e gateway che ricevono i dati da i vari sensori. Questo layer è semplicemente un livello di mediazione, dove l'informazione (dato) non viene altera o elaborata, ma semplicemente trasmesso all'Application layer.
- **Application layer** è il layer nel quale l'informazione viene immagazzinata ed elaborata. Questo layer è il più importante, è qui dove il dato viene trasformato da una semplice misurazione fisica ad una possibile revenue per l'azienda che lo gestisce.

Sono molte le aziende che stanno cercando di imporsi in questo mercato fiorente. Data la vastità dei campi di applicazione, non è semplice prevedere quale standard predominerà sugli altri. Aziende del calibro di Samsung con la piattaforma artik , Zigbee con DotDot e Google con Weave, hanno già proposto delle possibili soluzioni per il "linguaggio" universale utilizzabile dai vari dispositivi. Un dibattito ancora più acceso riguarda i protocolli e la topologia di rete da utilizzare. Dovendo superare i limiti delle tecnologie attuali, sono molteplici le problematiche che devono essere affrontate per poter offrire un'architettura adattabile ai vari use-case dell'IoT. Con questa tesi si cercherà di approfondire lo stato dell'arte del network layer andando a esporre le principali tecnologie ad oggi presenti sul mercato. In particolare verrà posta l'attenzione sulla tecnologia LoRaWAN e una sua possibile implementazione all'interno del framework Kura/ESF sviluppato da Eurotech.

# **Capitolo 2**

## **LPWAN**

Data la grande varietà dei possibili scenari applicativi dell'IoT , trovare uno standard capace di adattarsi, in modo dinamico, ad ognuno di essi, non è un compito facile. Le tecnologie wireless tradizionali, non sono in grado di soddisfare la dinamicità di questo paradigma. Già diverse soluzioni sono nate per cercare di fronteggiare questi problemi, optando per metodi risolutivi anche molto distanti l'uno dall'altro. Con questo capitolo si approfondiranno le problematiche, che i nuovi standard, riguardanti il network layer, dovranno essere in grado di risolvere.

### **2.1 Alla base delle reti LPWAN**

Le tecnologie wireless com ZigBee, WiFi, non sono ideate per connettere devices alimentati a batteria distribuiti in una vasta area geografica. Il range di queste tecnologie è limitato a poche centinaia di metri al massimo. Tutto ciò implica che i devices, non possono essere implementati in tutti gli ambienti, ma solo in uno spazio ristretto alla portata del segnale del gateway. Quindi scenari quali la sicurezza sanitaria, agricoltura di precisione, logistica non permettono l'utilizzo di questa tecnologia. La grande copertura offerta dalle tecnologia cellulare è la ragione per la quale è ad oggi la tecnologia più utilizzata nelle comunicazioni M2M. Tuttavia, il continuo progresso tecnologico, sta portando l'abbandono della rete GSM da parte degli operatori telefonici, per poter riutilizzare le bande da essa occupata con tecnologie più innovative. In generale, la tecnologia cellulare non garantisce una durata della batteria molto prolungata ed inoltre il costo complessivo dei moduli cellulari è molto elevato, data la complessità delle forme d'onda .

Dovendo ingegnerizzare il network layer, è importante capire quali sono le principali problematiche che le tecnologie attuali non sono in grado di colmare.

- *Indirizzabilità*: a causa dell'elevato numero di oggetti che entrano in gioco in IoT, la capacità di indirizzamento di IPv4 non è più sufficiente. IPv4 usa 32 bit per gli indirizzi, quindi ce ne possono essere massimo  $2^{32}$  diversi. Il passaggio a IPv6 risolverà il problema, in quanto si passerà da 32 bit a 128 bit per gli indirizzi.
- *Scalabilità*: Dato l'elevato numero di devices previsti nei scenari urbani ed industriali, la network technology alla base della rete dovrà essere adattabile, in modo dinamico, al carico di dispositivi connessi.
- *"Arrive and operate"*: dispositivi mobili eventualmente aggiunti dopo la formazione iniziale del sistema, non devono aver bisogno di configurazione, ma devono essere in grado di stabilire connessioni autonomamente con gli altri oggetti già presenti.
- *Costo unitario*: Il costo del end device, dovrà essere conveniente per garantire la più ampia fetta di mercato.
- *Fonti di energia*: Gli oggetti nella maggior parte dei casi sono mobili, quindi non hanno sempre la possibilità di essere collegati a una fonte di energia. Le batterie inoltre sono pesanti e grandi. È necessario quindi andare a ridurre la quantità di energia richiesta dagli oggetti, andando ad aumentare il tempo di deep-sleep .
- *Interoperabilità*: gli oggetti sono di natura diversa (per esempio possono avere requisiti di larghezza di banda diversi, o hardware diverso). Questo implica la necessità di standard, in modo che oggetti di tipo diverso possano comunicare tra loro
- *Costo computazionale*: La modulazione, alla base di queste nuove tipologie di rete, dovrà essere concepita in modo da non richiedere un costo computazionale elevato .
- *Raggio d'azione*: La necessità di utilizzare questi devices in ambienti difficili o rurali, rende necessario l'utilizzo di tecnologie wireless con un raggio di azione dell'ordine di una decina di chilometri.

- *Sicurezza:* Lo scambio dei dati dovrà avvenire in maniera sicura, implementando algoritmi di cifratura dei dati o procedure di handshaking.
- *Tolleranza ai guasti:* Il mal funzionamento o il guasto di un nodo della rete, non dovrà compromettere il funzionamento dell'intera rete a lui connessa.

## 2.2 LPWAN

Per colmare il gap tra tecnologie esistenti e la necessità di connettere milioni di devices diversi, sono nate le LPWAN *Low power wide area network*. Le reti LPWAN rappresentano un modello di comunicazione innovativo, che integra le tecnologie cellulari tradizionali e quelle a corto raggio per affrontare diverse esigenze delle applicazioni IoT. Le reti LPWA, andando a sacrificare il massimo throughput dei devices, ne sono in grado di gestire un gran numero contemporaneamente, cosa che con le tecnologie wireless tradizionali non è possibile realizzare.

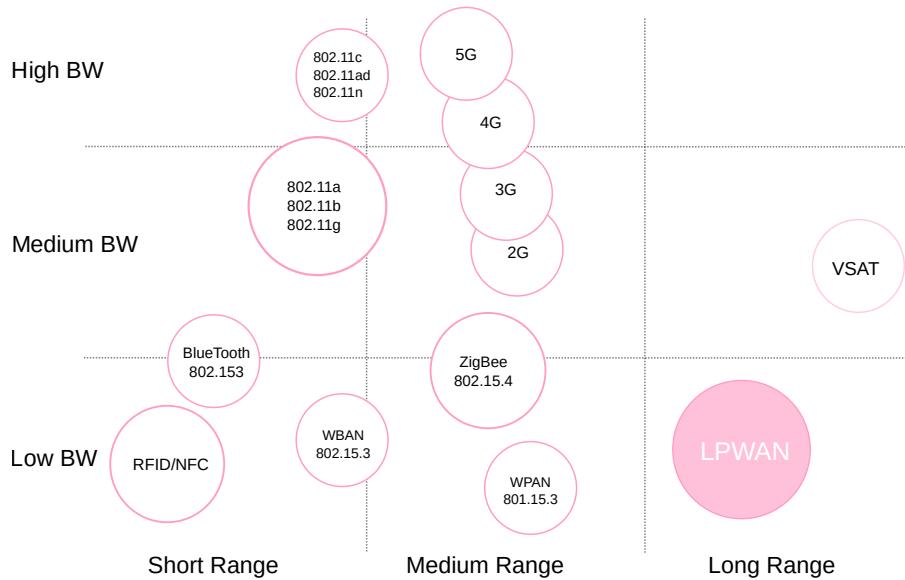


Figura 2.1: Comparazione tipologia di reti

In questo contesto, i maggiori competitor sono NB-IoT, EC-GSM-IoT LTE-M, SigFox e Lora. Le prime tre sono una evoluzione delle precedenti reti cellulari 2G, 3G e 4G. Operando sulle bande di frequenza licenziate.

ziate, e necessario che ognuna di queste tecnologie sia approvata dalla 3GPP ( 3rd Generation Partnership Project), la quale si occupa della standardizzazione dei sistemi di telecomunicazione a livello internazionale. All'opposto, Sigfox e Lora sono due tecnologie che operano nelle frequenze ISM (Industrial, Scientific and Medical). Le frequenze ISM sono uno spettro radio riservato alle applicazioni di radiocomunicazione non commerciali.

Frequency Spectrum Allocation			
Unlicensed ISM/SRD bands:			
<b>USA/Canada:</b>	260-470 MHz 902-928 MHz 2400- 2483.5 MHz	<b>Japan:</b>	315 MHz 426-430, 449, 469 MHz 2400-2483.5 MHz 2471-2497 MHz
<b>Europe:</b>	433.050-434.790 MHz 863.0-870.0 MHz 2400-2483.5 MHz		

Figura 2.2: Frequenze ISM

In particolare entrambe le tecnologie operano nella banda degli 868 [MHz] la quale permette una potenza del segnale inviato massima pari a 14 [dbm] ed un duty cycle che può variare dal 0.1% fino al 10% a seconda del canale scelto.

## 2.3 NB-IoT

Narrowband IoT (NB-IoT) o LTE Cat NB1 è uno standard certificato nella release 13 del 3GPP, la quale riutilizza le infrastrutture già presenti, quali 2G, 3G, 4G per la rapida realizzazione di una rete LPWA per l'IoT. Focalizzandosi sulla durata della batteria, i moduli NB-IoT risultano avere un costo all'unità minore del 75% rispetto ad un normale modulo LTE. Basato sulle frequenze licenziate, NB-IoT è in grado di offrire tre diversi scenari di sviluppo [2]

- *standalone*, utilizzando qualsiasi spettro disponibile dell' operatore.
- *guard band*, utilizzando lo spettro libero presente tra due bande radio, per prevenire interferenze.
- *in band*, utilizzando lo stesso spettro della banda LTE.

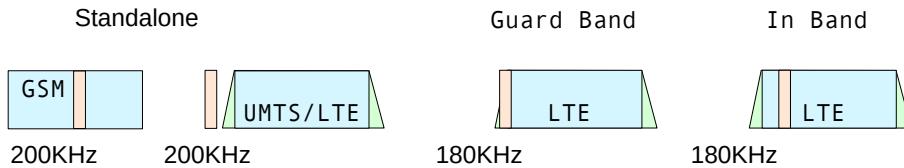


Figura 2.3: Modalità di funzionamento NB-IoT

L'obiettivo che NB-IoT si prefigge è quello di mettere a disposizione una tecnologia con una elevata copertura ed un basso data-rate. La possibilità di riutilizzare strutture già esistenti, ed il basso costo per device , rendo NB-IoT, una delle tecnologie che sta riscuotendo maggiore successo nel abito IoT.

## 2.4 LTE-M

Dalla realise 8 del 3GPP, diverse nuove tipologie di rete LTE sono disponibili. La categoria che offre le migliori performance batteria/data-rate è la categoria LTE Cat-M1 o LTE-M. Questa categoria ,a differenza del NB-IoT, rispecchia lo standard LTE in pieno, implementando la Frequency Division Multiplexing (FDM) e Time Division Multiplexing (TDM). Risultando adatta per applicazioni nelle quali è necessario l'invio di dati audio o video oppure per comunicazioni a bassa latenza . Il data-rate raggiungibile è pari a 5Mbps in uplink e 10 [Mbps] in downlink . Questo tipo di connessione sarà utile per tutte quelle applicazioni in cui è richiesta una elevata sicurezza del dato da trasmettere, come ad esempio applicazioni di video-sorveglianza o automotive .Questa tecnologia ,già disponibile negli Stati Uniti tramite la rete Verizon, è in fase di roll out per molti operatori europei.

## 2.5 EC-GSM-IoT

EC-GSM-IoT si basa su funzionalità aggiuntive a partire da EGPRS che consentono ad una rete GSM/EDGE di essere predisposta per fornire servizi IoT. Lo standard è stato pensato in particolare per quei Paesi, come quelli in via di sviluppo, dove una rete LTE non è ancora disponibile. L'occupazione spettrale di ogni canale corrisponde a 200 kHz. Tuttavia, al fine di dispiegare EC-GSM-IoT, si richiede una banda utile di 2.4 MHz per

permettere il frequency hopping , che, con l'aggiunta di 2 canali di guardia di 200 kHz ciascuno agli estremi della banda, porta l'occupazione di banda complessiva a 2.8 MHz. La potenza di trasmissione del Il data rate di picco raggiungibile sia in DL sia in UL è di 491 kbps, mentre il valore mediato nominale è di 98 kbps sia in DL sia in UL. Al fine di soddisfare i requisiti di capacità (più di 50.000 terminali in ogni singolo settore di una cella trisetoriale).

La figura 2.1 riassume in breve le varie caratteristiche delle reti cellulari facenti parte della categoria LPWA

	LTE-M	NB-IoT	EC-GSM-IoT
Dispiegamento	In-band LTE	In-band LTE, Guard-band LTE, Stand-alone	In-band GSM
Copertura radioelettrica	155.7 dB (20 dBm)	164 dB (23 dBm)	154 dB (23 dBm) 164 dB (33 dBm)
Canalizzazione	1.08 MHz	180 kHz	200 kHz
Banda minima richiesta	1.4 MHz	180 kHz (In-band LTE, Guard-band LTE) 200 kHz (Stand-alone)	2.4 MHz (con servizi voce GSM) 600kHz (senza servizi voce GSM)
Duplexing	HD-FDD, FD-FDD, TDD	HD-FDD	HD-FDD
Data rate di picco	HD-FDD e FD-FDD 1 Mbps DL HD-FDD e FD-FDD 1 Mbps UL	In-band LTE 170 kbps DL Stand-alone 226.7 kbps DL 250 kbps UL	491 kbps DL 491 kbps UL
Data rate mediati nominali	FD-FDD 800 kbps DL FD-FDD 1 Mbps UL HD-FDD 300 kbps DL HD-FDD 375 kbps UL	21.25 kbps DL 62.5 kbps UL	98 kbps DL 98 kbps UL
Classi di potenza dei terminali	20 dBm / 23 dBm	20 dBm / 23 dBm	23 dBm / 33 dBm

Tabella 2.1: Comparazione reti cellulari per l'IoT

## 2.6 Sigfox

SigFox, azienda francese, sta sviluppando in partnership con altri operatori di rete una soluzione LPWAN basata sulla sua tecnologia. Sigfox punta alla costruzione di una rete mondiale proprietaria basata su frequenze ISM. Correntemente SigFox è presente in Francia, Belgio, Olanda e Portogallo come illustrato nella figura 2.4.

Gli end-devices comunicano con le varie base stations usando una modulazione (BPSK) *Binary Phase Shift Keying* con una banda di soli 100 [Hz]. Per via delle regolazioni vigenti nello spettro ISM, è per garantire una durata della batteria pari ad una decina di anni, il numero massimo

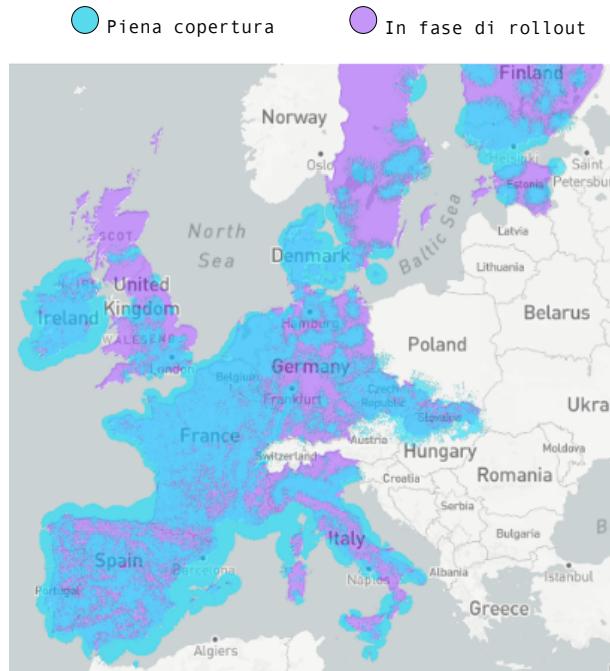


Figura 2.4: Mappa copertura SigFox

di messaggi inviabili in un giorno è 140, con lunghezza del payload pari a 12 [byte] e un throughput pari a 100 [bps]. SigFox si colloca come rete LPWAN con il minore throughput, limitando il numero di use-case possibili. Inizialmente SigFox supportava solo comunicazioni unidirezionali, successivamente, ha introdotto la possibilità di avere una comunicazione bidirezionale, limitando il numero di byte trasmissibili da gateway a devices a 4-8 bytes per giorno.

## 2.7 LoRaWAN

LoraWAN è una tecnologia di modulazione wireless semi-proprietaria sviluppata da Semtech. Essa è composta da un layer fisico ,proprietario, che prende il nome di *Lora*[3] , e una parte libera chiamata LoRaWAN[5] nella quale viene definito un protocollo di comunicazione, il quale usa LoRa come layer fisico. Basandosi su una tecnica di comunicazione a *spread spectrum*, LoRa è in grado di instaurare una comunicazione bidirezionale tra device e gateway. I punti chiave dei questa tecnologia sono il grande raggio di copertura , il basso consumo energetico e la capacità

di adattare in maniera dinamica il data rate, il quale può variare dai 0.3 ai 50 [Kbps] a seconda dell'utilizzo. Come per SigFox, la tecnologia sviluppata da Semtech, si basa sulle bande ISM, inoltre Essendo il protocollo LoRaWAN open source, si ha la possibilità di creare delle reti pubbliche o private senza disporre di alcuna licenza, riducendo così il time to market di questa tecnologia. Progetti come The Things Network mirano a creare una rete LoRa ,pubblica è privata, a livello globale.

## 2.8 Osservazioni

In questo mercato frammentato, non è semplice capire quale tecnologia sia adatta a ricoprire una data applicazione. Essendo questi standard molto giovani, è complicato comprendere le reali potenzialità di ognuna di queste soluzioni. Quello che è possibile prevedere, sarà un incremento esponenziale di device che stanno alla base della piramide in figura 2.5, devices i quali potranno essere utilizzati in innumerevoli settori, non ancora esplorati dalle tecnologie attuali, come per esempio i contatori della dell'acqua, applicazioni per l'agricoltura di precisione e così via.

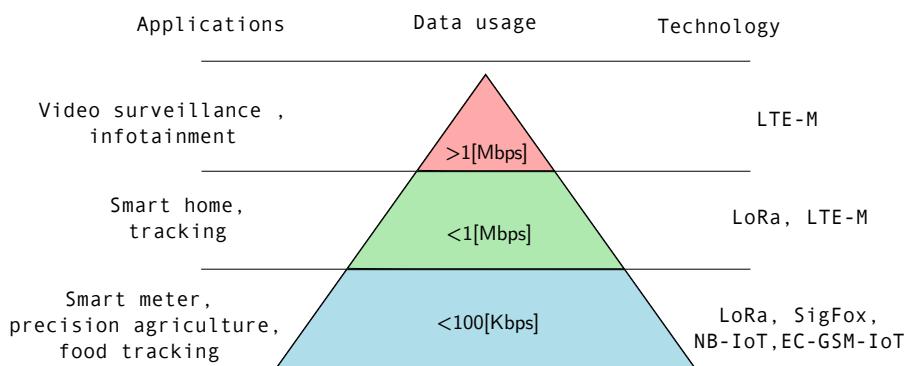


Figura 2.5: Capacità delle reti LPWA

Per le aziende, che si apprestano ad investire sul mondo dell'IoT, la scelta della corretta tecnologia su cui andare a sviluppare i loro servizi non risulta semplice, in quanto, fattori quali sicurezza, aggiornamenti software, affidabilità devono essere ancora testati a pieno. Con la figura 2.6 si vuole riassumere in breve i punti chiave delle tecnologie appena trattate.

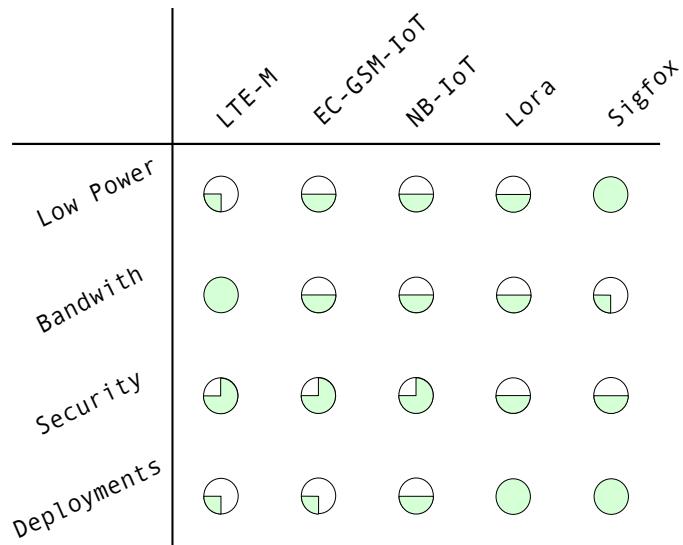


Figura 2.6: Comparazione feature reti LPWAN

Nel prossimo capitolo verrà analizzata in dettaglio la soluzione che Semtech propone, approfondendo il layer fisico *Lora* e la struttura del protocollo LoRaWAN.



# **Capitolo 3**

## **LoRa**

Il successo delle reti LPWAN risiede nella loro abilità di offrire una connessione, a bassa potenza, per connettere un gran numero di dispositivi distribuiti in una vasta area geografica. Una delle tecnologie che sta riscuotendo un grande successo in ambito europeo, è LoRa. Brevettata dalla francese Cycleo e successivamente acquistata da Semtech, LoRa offre un buon compromesso tra data-rate, durata della batteria e distanza di comunicazione. Grazie all'adozione di un protocollo open source, la facile implementazione della rete ed il costo contenuto dei moduli radio, LoRa, ha già una vasta community attiva alle spalle. Per comprendere il funzionamento di questa tecnologia, ed i vantaggi che ne derivano rispetto alle tecnologie concorrenti , è necessario studiare la modulazione alla base del layer fisico.

### **3.1 Narrow Band e Spread Spectrum**

Per garantire la copertura wireless di una vasta area geografica, le reti LPWAN sono state ideate basandosi su bilanci di collegamento (link budget) ideali dell'ordine dei  $150 \pm 10$ [dB]. Queste specifiche, permettono un range di operatività pari ad una decina di chilometri nelle zone rurali. In aggiunta, il basso data-rate richiesto dalle applicazioni IoT, permette ,a queste tecniche, di concentrare una elevata energia in ogni simbolo trasmesso; rendendo possibile ai gateways la decodifica di segnali con una attenuazione pari o superiore ai -110[dBm].

Per ottenere queste prestazioni la maggior parte delle LPWAN utilizza tecniche di comunicazioni *Narrow Band* o *Spread Spectrum*.

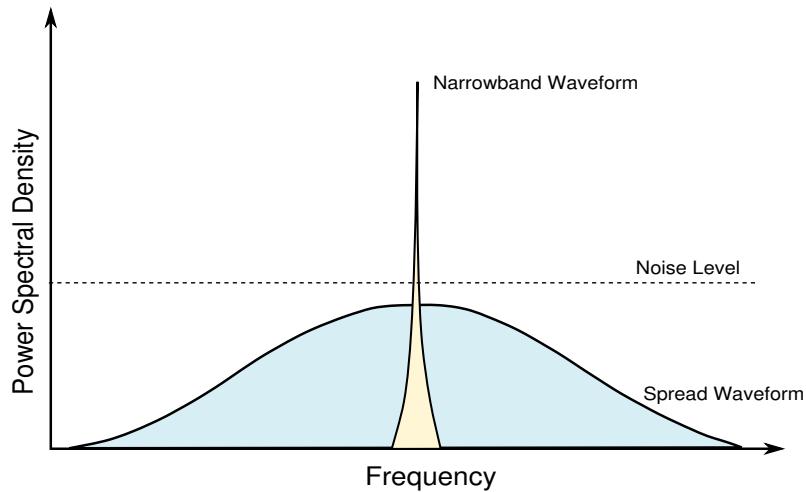


Figura 3.1: Comparazione tra UNB e SSP

### 3.1.1 NarrowBand

La modulazione narrowband è in grado di offrire il link budget desiderato, codificando il segnale in una banda molto ristretta ( $\approx 25[\text{KHz}]$ ). Assegnando ad ogni portante una banda siffatta, la modulazione NB è in grado di utilizzare l'intero spettro in maniera efficace e al tempo stesso ridurre il rumore al interno del singolo canale. Inoltre, quest'ultimo vantaggio, permette al gateway di demodulare il segnale più facilmente rendendo i moduli radio semplici da implementare a livello hardware e poco costosi.

Per aumentare la capacità di ricezione del singolo gateway e diminuire di molto la complessità dei moduli radio, SigFox e altre tecnologie hanno estremizzato il concetto di narrowband, andando ad implementare quella che viene chiamata *Ultra Narrow Band*. Questa modulazione assegna ad ogni portante una banda di appena  $100[\text{Hz}]$ , riducendo in maniera drastica il data-rate ed i vari scenari applicativi.

### 3.1.2 Spread spectrum

Per Spread spectrum si intende un tecnica di trasmissione in grado di diffondere il segnale informativo in una banda più ampia di quella originariamente occupata dal segnale. Come risultato otteniamo una trasmissione che incorpora un quantitativo di rumore maggiore all'interno del

singolo canale rispetto ad una trasmissione NB, rendendola però molto più resistente alle interferenze naturali, al rumore e agli attacchi basati sul jamming. Per ottimizzare l'uso dello spettro, è possibile inviare segnali composti da frequenze ortogonali tra loro, in questo modo è realizzabile la decodifica in maniera concorrente di segnali diversi, andando ad aumentare la capacità della rete. LoRa utilizza un sistema di modulazione basato su di una tecnica a spettro espanso che prende il nome di Chirp Spread Spectrum (CSS).

## 3.2 Layer fisico LoRa

### 3.2.1 Chirp Spread Spectrum

CSS o Chirp Spread Spectrum è la modulazione alla base del layer fisico LoRa. Con chirp (Compressed High Intensity Radar Pulse) si intende un segnale di ampiezza costante, il quale incrementa o decrementa la sua frequenza nel tempo. Parliamo quindi di *UpChirp* nel caso di un aumento di frequenza e di *DownChirp* nel caso di un decremento. Un generico segnale chirp può essere rappresentato da una sinusode che come argomento ha una funzione  $\theta(t)$  che varia nel tempo.

$$s(t) = A \cos(\theta(t)) \quad (3.1)$$

Andando ora a derivare la funzione  $\theta(t)$  nel tempo, possiamo definire due nuovi parametri:  $\gamma(t)$  3.2 che prende il nome di frequenza istantanea e  $c(t)$  3.3 .

$$\gamma(t) = \frac{1}{2\pi} \frac{d\theta(t)}{dt} \quad (3.2)$$

$$c(t) = \frac{1}{2\pi} \frac{d^2\theta(t)}{dt^2} = \frac{d\gamma(t)}{dt} \quad (3.3)$$

A seconda di come viene scelta la funzione  $\theta(t)$  il segnale avrà andamenti diversi nel dominio del tempo; per semplificare la modulazione e demodulazione di quest'ultimo, LoRa utilizza una variazione lineare della frequenza. Il modo più semplice per ottenere un segnale siffatto, è andando a scegliere  $\theta(t)$  come un argomento che dipende in modo quadratico dal tempo.

$$\theta(t) = 2\pi\mu t^2 + 2\pi f_i t + \varphi \quad (3.4)$$

In questo modo  $\gamma(t)$  avrà una dipendenza lineare da  $t$ .

$$\gamma(t) = \frac{1}{2\pi} \frac{d}{dt} \theta(t) = 2\mu t + f_i \quad (3.5)$$

In 3.5  $f_i$  rappresenta la frequenza iniziale del segnale e definiamo  $k = 2\mu$  come

$$k = \frac{f_e - f_i}{T} \quad (3.6)$$

il quale non dipende dal tempo ma è una costante. Dove  $f_e$  è la frequenza di finale,  $f_i$  è la frequenza iniziale e  $T$  è il tempo impiegato dal segnale per passare da  $f_i$  a  $f_e$ . Ricapitolando, il segnale  $s(t)$  sarà uguale a

$$\begin{aligned} s(t) &= A \cos(\theta(t)) = A \cos\left(2\pi t \left(\frac{k}{2}t + f_i\right) + \varphi\right) \\ &= A \cos\left(2\pi \frac{k}{2}t^2 + 2\pi f_0 t\right) \end{aligned} \quad (3.7)$$

Dall'ultima equazione è possibile trarre alcune osservazioni,

- Se la chirpizzazione è nulla la frequenza non varia in funzione del tempo, quindi  $s(t)$  rappresenta una normale sinusode.
- Se  $\theta(t)$  è lineare, la frequenza è costante.
- Se  $\theta(t)$  dipende in modo quadratico dal tempo, allora la frequenza varia in modo lineare.

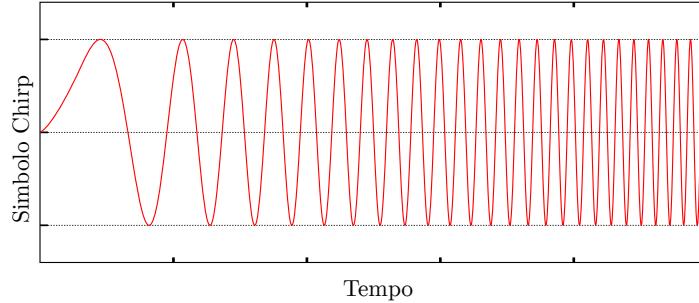


Figura 3.2: Esempio di segnale Chirp nel dominio del tempo

Per aumentare l'efficienza delle trasmissioni, LoRa utilizza i segnali chirp in combinazione ad una tecnica Spread Spectrum. Prima di procedere alla analisi della implementazione di questa tecnica, è utile introdurre un risultato derivante dal teorema di *Shannon-Hartley*. Nella

teoria delle telecomunicazioni il teorema di Shannon–Hartley stabilisce il data-rate massimo ottenibile, senza la perdita di dati, attraverso l'utilizzo di un canale di comunicazione rumoroso ad una larghezza di banda fissata.

$$C = B \cdot \log_2 \left( 1 + \frac{S}{N} \right) \quad (3.8)$$

Dove  $C$  è la capacità del canale in [bit/s],  $B$  la larghezza di banda ,  $S$  la potenza utile del segnale [mW] e  $N$  la potenza del rumore presente nel sistema [mW] e  $\frac{S}{N} = SNR$ . Riformulando l'equazione precedente e passando al logaritmo naturale otteniamo

$$\frac{C}{B} = 1,442 \cdot \frac{S}{N} \quad (3.9)$$

Sapendo ora che le trasmissioni Spread Spectrum il rapporto segnale rumore è molto basso non è sbagliato assumere  $\frac{S}{N} \ll 1$ . In questo caso l'equazione 3.9 diventa,

$$\frac{C}{B} \approx \frac{S}{N} \quad (3.10)$$

dalla quale si evince un importante risultato: Per non avere errori nella trasmissione dei dati in presenza di un canale di comunicazione rumoroso, è necessario aumentare la banda del canale. Basandosi su questo risultato, Semtech ha progettato LoRa in modo tale che i segnali Chirp vengano distribuiti in modo uniforme in tutta la banda. Con questo si intende che, data una banda  $B = [f_0, f_1]$ , il segnale Chirp inviato dai dispositivi LoRa sarà distribuito uniformemente all'interno di  $B$ . Nel caso in cui il segnale , aumentando linearmente la sua frequenza, arrivi all'istante  $t_c$  ad uno degli estremi della banda  $f_0, f_1$ , non potendo continuare ad aumentare o diminuire la sua frequenza, è costretto all'istante  $t_{c+1}$  a ripartire dalla frequenza opposta a quella dell'estremo raggiunto. È possibile osservare questo fenomeno in \*3.3. Utilizzando una modulazione Spread Spectrum in combinazione con i segnali di tipo Chirp, sono ottenibili numerosi vantaggi.

- L'utilizzo dello spettro in maniera efficace.
- La possibilità di evitare collisioni nelle trasmissioni utilizzando segnali ortogonali.
- Ottima immunità all'effetto Doppler, rendendo possibile l'utilizzo dei dispositivi in mobilità.

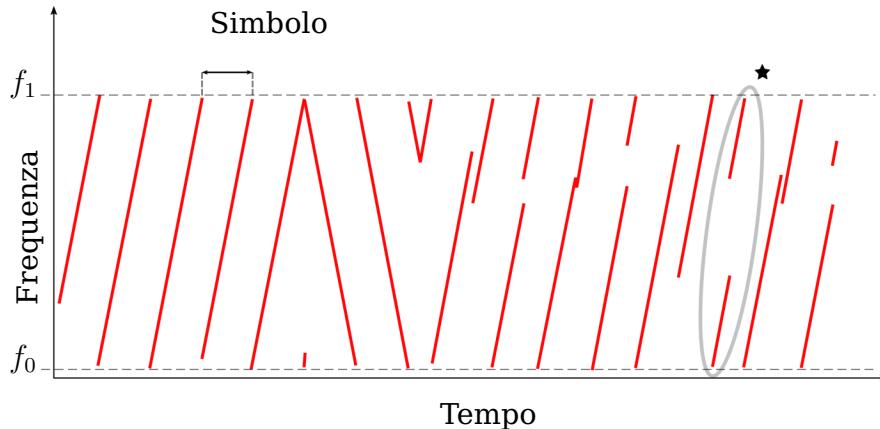


Figura 3.3: Segnale Chirp nel dominio della frequenza

- Ottima resistenza alle interferenze naturali.
- Ottima resistenza alle interferenze da parte di altri segnali.
- Per una fissata potenza di trasmissione e un data-rate fissato, LoRa permette di raggiungere distanze maggiori rispetto ad altre modulazioni.

### 3.2.2 Adattabilità

Uno degli aspetti peculiari del layer fisico è la possibilità di variare tre parametri, in modo dinamico, per ottenere la massima efficienza nella trasmissione. Il primo parametro, che prende il nome di Spread Factor (SF), è l'indice di quanti bit sono utilizzati, all'interno di un segnale Chirp, per rappresentare un simbolo. Questo vuol dire che, preso uno SF pari a  $X$ , il segnale utilizzerà  $2^X$  bit per la rappresentazione del simbolo a lui associato. Variando il SF, variano anche le possibili frequenze iniziali del segnale; infatti, ogni segnale avrà  $M = 2^X$  possibili frequenze iniziali. Nella documentazione tecnica fornita da Semtech [4] troviamo 6 possibili Spread Factor partendo dal 7SF fino ad arrivare al 12SF, ad ognuno di essi è associato un rapporto segnale rumore, che sarà più elevato per SF maggiori 3.1.

SF	SNR
7	-7.5[dB]
8	-10[dB]
9	-12.5[dB]
10	-15[dB]
11	-17.5[dB]
12	-20[dB]

Tabella 3.1: Rapporto segnale rumore dei diversi Spreading Factors

Il secondo dei parametri variabili è la larghezza di banda utilizzata. Questo parametro dipende dal dispositivo utilizzato, ed è quello che in combinazione allo Spreading Factor, determina il bit rate. Nel modello di trasmettitore SX1272 è possibile utilizzare tre larghezza di banda diverse, 125 [KHz], 250 [KHz] e 500 [KHz]. In alternativa, il modello più recente, SX1276, ha la possibilità di variare la banda partendo da 7.8 [KHz] fino a 500 [KHz]. Per capire come questi due parametri influenzino il data-rate, è necessario utilizzare la formula 3.11 presente nella documentazione ufficiale [4]. 3.4.

$$T_s = \frac{2^X}{B}. \quad (3.11)$$

Dove  $T_s$  rappresenta il tempo necessario per l'invio del simbolo,  $X$  lo Spreading Factor usato e  $B$  la banda. Ponendoci nella situazione in cui la banda utilizzata nella comunicazione sia fissata a priori; la variazione di un'unità dello SF comporterà una durata doppia del tempo di comunicazione. Analogamente un incremento della banda  $B$  comporterà un incremento della velocità con cui i segnali chirp vengono trasmessi ottenendo quindi un aumento del bit rate .

L'ultimo dei parametri variabili è la potenza impiegata nella trasmissione. Maggiore sarà la potenza impiegata, maggiore risulterà la distanza percorribile dal messaggio andando però a diminuire la durata della batteria.

Per ottenere la massima efficienza della rete, è necessario calibrare in modo opportuno questi tre parametri per ogni singolo devices.

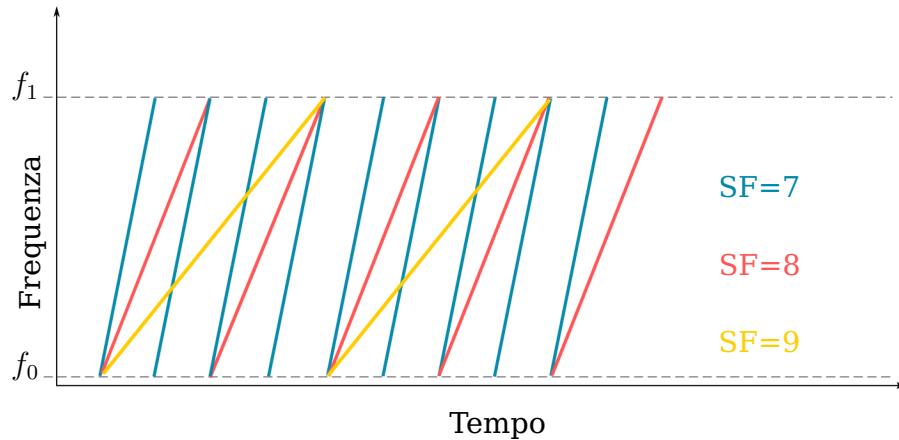


Figura 3.4: Comparazione simbolica dei vari SF

### 3.2.3 LoRa packet

Il consumo energetico dei dispositivi è un punto fondamentale sul quale la tecnologia LoRa è stata costruita. Per ottenere una durata della batteria pari ad una decina d'anni, è necessario che i devices spendano la maggior parte del tempo in modalità deep-sleep e comunichino con il server solo in presenza di input esterni o all'attivazione di un timer. Inoltre, per utilizzare le risorse in maniera ancora più efficace, i devices non implementano nessun tipo di sincronizzazione con i gateways. La scelta di una comunicazione asincrona, ha portato alla implementazione di un "pacchetto fisico" costruito in questo modo:

- *Preambolo.* Composto da soli UpChirp, il preambolo permette al gateway di capire quando un dispositivo inizia la comunicazione.
- *Header e CRC.* L'header contiene le informazioni riguardanti il payload quali , la sua lunghezza , il code rate utilizzato nel payload e la presenza o meno del CRC. Da specifica l'header ha sempre un code-rate pari a 4/8 che rappresenta la massima ridondanza possibile, nelle comunicazioni LoRa, con correlato CRC.
- *Payload.* In esso è contenuto il dato rilevato dal dispositivo, la sua è lunghezza variabile fino ad un massimo di 255 [byte].
- *Payload CRC.*

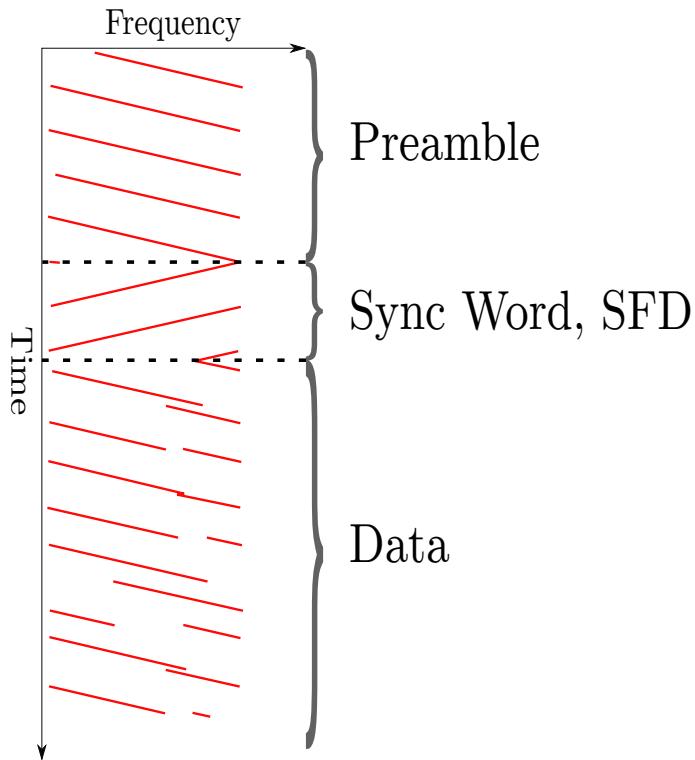


Figura 3.5: Struttura pacchetto Lora

In caso di specifiche stringenti, è possibile far operare i moduli LoRa in modalità *implicita*. In questa modalità, il pacchetto non conterrà l'header. Per garantire il corretto funzionamento è dunque necessario che il payload sia di lunghezza fissata e nota al gateway al quale il messaggio è indirizzato. L'immagine 3.6 rappresenta la struttura di un pacchetto "fisico" LoRa. La prima parte della trasmissione rappresentata dal preambolo, è codificata con una serie di *UpChirp*. In questo modo, il gateway è in grado di capire quando un dispositivo inizia la comunicazione, riuscendo ad allocare le risorse in maniera preventiva per la ricezione del dato. Concluso l'invio del preambolo, il dispositivo invia una serie di *DownChirp* i quali rappresentano lo *SFD Start Frame Delimiter* tramite il quale il gateway è in grado di sincronizzarsi in maniera esatta nella frequenza utilizzata dalla comunicazione. L'ultima parte è composta da cambiamenti istantanei della frequenza da parte del segnale, i quali indicano la presenza di "dati" nella trasmissione. Oltre alla modulazione, LoRa specifica delle operazioni di codifica che vengono fatte prima che il

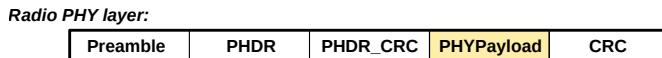


Figura 3.6: Pacchetto codificato dal layer fisico

segnale venga modulato.

- **Gray Indexing** procedura simile alla codifica Grey. È utilizzata per diminuire la probabilità di errore nel sistema.
- **Data whitening** è una tecnica utilizzata per ridurre la probabilità di avere lunghe sequenze di 1 e 0. Oltre a semplificare la decodifica, il data whitening aiuta a distribuire l'informazione uniformemente in tutta la banda.
- **Interliving** è una tecnica utilizzata per diminuire la possibilità di errori nelle comunicazioni. Con questa tecnica permette di scambiare i simboli all'interno del messaggio, andando ad aumentare le probabilità di correzione degli errori in caso di disturbi.
- **Forward Error Correction** è implementata tramite l'utilizzo dei codici di Hamming, la lunghezza della parola del codice è fissata e pari a 4, mentre la lunghezza della parola di controllo è un parametro che può variare da 5 a 8.

La lunghezza del payload come detto prima è un numero variabile il quale dipende da molti fattori

$$L_{payload} = 8 + \max \left( \left\lceil \frac{8PL - 4SF + 44 - 20H}{4(SF - 2DE)} \right\rceil + (CR + 4), 0 \right) \quad (3.12)$$

Dove PL è il numero di byte del payload iniziale, H può essere 1 o 0 a seconda se il device opera in modalità "implicita" oppure no, CR è il numero di bits di parità e DE può essere 0 o 1 a seconda dell'abilitazione o meno della funzione di *low data rate*. L'opzione di low data rate è attivabile in caso di trasmissioni lunghe e lente, attivandola si forzerà il dispositivo trasmittente ad aumentare la stabilità della frequenza scelta per la comunicazione .

### 3.3 LoRaWAN

Basato sul layer fisico LoRa, LoRaWAN è un protocollo MAC o *media access control* open source, utilizzabile nelle comunicazioni LPWAN. Stan-

dardizzato tramite la LoRa Alliance, si prefigge il compito di mettere in comunicazione il device layer con l'application layer. Volendo fare un confronto con il modello OSI, LoRaWAN può essere collocato tra il secondo e terzo layer del suddetto modello.

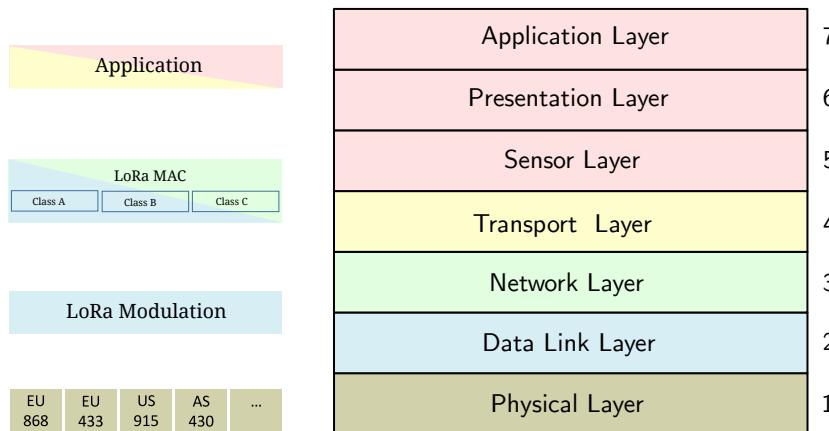


Figura 3.7: Comparazione del modello OSI con la struttura definita in LoRaWAN

### 3.3.1 Tipologia di rete e classi di dispositivi

Per garantire un elevato numero di devices contemporaneamente connessi, LoRaWAN si basa su una topologia di rete a stella. In questa topologia gerarchica, gli *end devices* comunicano con l'application layer, solo attraverso i gateways, i quali traducono i pacchetti LoRa in pacchetti UDP/TCP per poi inviarli al server. Data la struttura della rete, ogni messaggio inviato dagli *end devices* potrà essere ricevuto da uno o più gateways; starà quindi al server eliminare i duplicati e selezionare il gateway più adatto per rispondere al dispositivo.

Nella documentazione LoRaWAN sono definite tre classi di dispositivi ideate per diversi tipi di utilizzo. La classe principale è la classe A, questa classe è implementata in ogni dispositivo ed è quella focalizzata al risparmi energetico. Le classi B, e C sono invece delle estensione della classe A. Questo tipo di classi sono riservate a oggetti alimentati tramite la rete elettrica oppure tramite fonti di energia esterne.

- **Class A** è la modalità di funzionamento predefinita. In questa modalità il device comunica in modo asincrono con il gateways. In essa

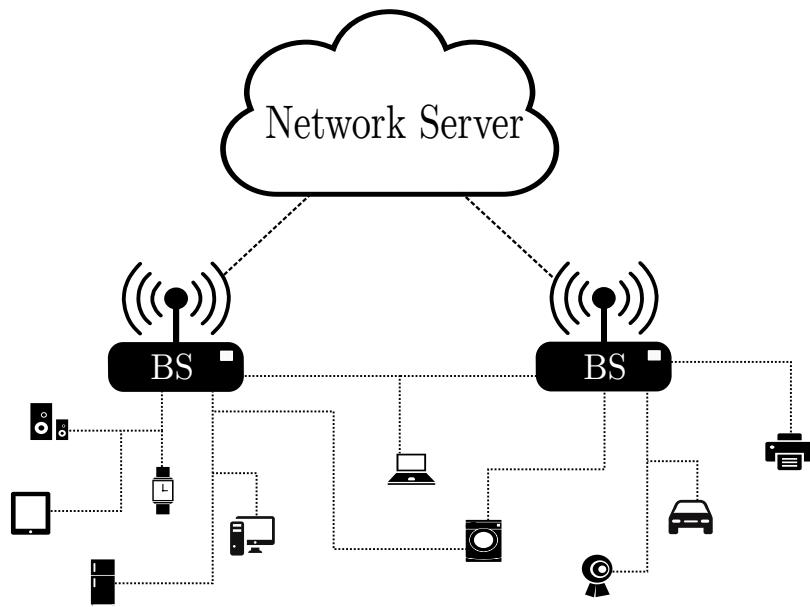


Figura 3.8: Struttura rete a stella LPWAN

sono implementate due finestre di ascolto da parte dei devices dopo 1[s] e 2[s] dalla fine della trasmissione. Se il gateway non risponde al messaggio ricevuto durante uno di questi intervalli, è necessario aspettare che l'invio di un nuovo messaggio da parte del device.

- **Class B** sono devices che estendono le funzionalità della classe A. Questi dispositivi sono sincronizzati con la Base Station attraverso messaggi *beacon* inviati dal gateway. Grazie a questa sincronizzazione, il gateway è in grado di comunicare con il dispositivo in intervalli di tempo prestabiliti.
- **Class C** è anch'essa una estensione della classe A. Questa classe permette il funzionamento quasi complementare del device; infatti un dispositivo di tipo C rimarrà continuamente in ascolto finché non necessita di comunicare lui stesso con il gateway. Ovviamente, rimanendo per la maggior parte del tempo in ascolto, i devices che operano con questa classe dovranno essere connessi ad una fonte di energia esterna.

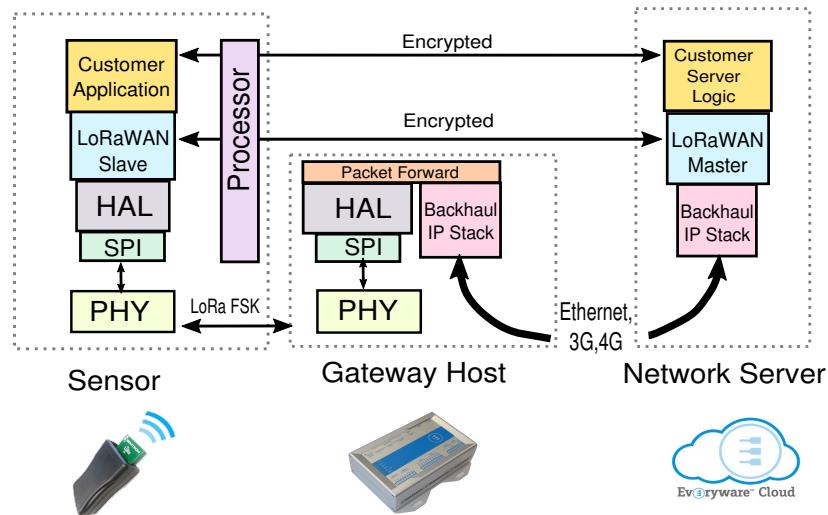


Figura 3.9: Stack del protocollo della rete LoRaWAN

### 3.3.2 Bande di frequenze

La tecnologia LoRa opera nelle bande non licenziate dello spettro radio. Come accade per le più comuni tecnologie wireless, WiFi, Bluetooth, ZigBee, anche LoRa può essere utilizzata dal consumatore senza la necessità di possedere una licenza o pagare un abbonamento. A discapito di ciò, la regolamentazione in vigore per l'utilizzo di queste bande impone limiti severi sulla potenza di trasmissione utilizzabile e l'occupazione del canale trasmittivo da parte dei dispositivi. Il protocollo LoRaWAN supporta sia le frequenze che vanno dagli 863-868[MHz] sia la banda dei 433[MHz]. Per la banda nella fascia degli 860[MHz], LoRaWAN specifica tre diversi canali (868.10, 868.30 and 868.50 MHz), con una bandwidth di 125[KHz] ciascuno, i quali dovranno essere supportati da ogni device. Inoltre, ogni gateway dovrà rimanere in ascolto su tutti e tre questi canali, in particolare, essi formano un set comune utilizzabile nella *join procedure* di un nuovo device. Per quanto riguarda la banda dei 433[MHz], si hanno a disposizione sempre tre tipi diversi di canali per la *join procedure* (433.175, 433.375 433.575 MHz). La regolamentazione Europea impone un duty-cycle molto ristretto per l'utilizzo delle frequenze ISM, in particolare nella banda degli 868 si è imposto l'utilizzo di duty cycle inferiori al 1% e per i 433 duty cycle inferiori al 0,1%.

Stato	Frequenza [MHz]
Europa	868-870
US	902-928
China	779-787

Tabella 3.2: Bande di frequenza per le varie regioni

### 3.3.3 Pacchetto LoRaWAN

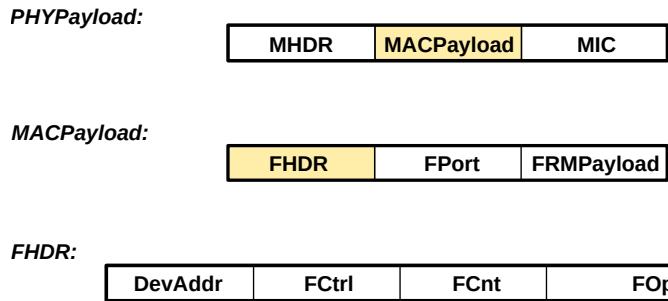


Figura 3.10: Stack del protocollo della rete LoRaWAN

All'interno del frame *PHYPayload* del pacchetto LoRa, è contenuto il messaggio LoRaWAN il quale è strutturato in questo modo:

- **MHDR:** Il MAC Header compone la prima parte del pacchetto, la quale si divide in tre sottocampi. Il primo specifica il tipo di messaggio nel campo MType che può essere di 6 tipi diversi, rappresentati da 3 bit ciascuno. Major specifica il formato dei messaggi scambiati nella procedura di join ed i primi 4 bytes del MAC Payload.
- **MACPayload:** è composto da *Frame header*(FHDR) il quale indica a quale nodo va inviato/arriva il pacchetto ed è utilizzato per l'algoritmo ADR. Il secondo campo è il *Frame counter*, nel quale sono presenti due contatori, uno per il numero di messaggi in uplink (FCntUp) inviati dal nodo, il secondo per quelli ricevuti (FCntDown). Oltre a questi due frame è presente il *FPoprt* il quale è opzionale e indica la presenza o meno di *FRMPayload* dove è contenuta l'informazione.
- **MIC:** acronimo di Message Integrity Check, è un codice che viene calcolato per ogni pacchetto e definisce l'integrità del messaggio.

Per gestire la rete esiste una lista di MAC command che possono essere scambiati esclusivamente fra il server e MAC layer di un end-device. Non possono essere visibili alle applicazioni nel server o nei nodi. I più importanti sono: *LinkCheckReq* viene utilizzato da un nodo per validare la connessione con la rete; *LinkADRReq* è una richiesta per il nodo in cui viene chiesto di cambiare data-rate, potenza in trasmissione, canale, o rate di ritrasmissione;

## 3.4 Adaptive Data Rate

Adaptive data rate (ADR), è un meccanismo utilizzato per ottimizzare il data rate dei dispositivi in modo dinamico. Questo meccanismo, implementato tramite l'applicatation layer, permette di modificare il Spread Factor a seconda delle condizioni della rete a cui i devices sono connessi. Dal momento in cui il nodo richiede di usufruire di ADR, l'application layer inizierà a collezionare le prestazioni delle ultime 20 trasmissioni effettuate dal dispositivo. In base ai dati collezionati, l'application layer sarà in grado di ottimizzare la connessione con il nodo in esame andandone a variare lo SF ,la bandwidth o la potenza utilizzata nella trasmissione I parametri utilizzati per l'ADR, sono il frame counter, il rapporto segnale rumore e il numero di gateways che hanno ricevuto i messaggi inviati. Basandosi su questi tre parametri, è evidente che ADR è applicabile solo ai nodi fissi della rete oppure a quei nodi che hanno periodi di mobilità limitata. Come esempio è possibile considerare un nodo fisso che comunica con la rete utilizzando uno SF pari a 12 , una bandwidth pari a 125[KHz] e ha un SNR pari a 2.0[dB]. Un rapporto segnale rumore positivo è indice che il nodo si trova ad una distanza ravvicinata dal gateway e non sono presenti elementi che possono interferire con la comunicazione; avendo un margine pari a 22[dB] è ragionevole andare ad abbassare il SF di 2-3 valori (10/9 SF) oppure andare a diminuire la potenza con cui il nodo trasmette. Determinare i parametri ottimi con cui questi devices devono operare non è una decisione semplice, essi variano dalla regione in cui i devices operano e dallo stato della rete. Un possibile algoritmo utilizzabile è quello consigliato da Semtech nella documentazione ufficiale.

### 3.4.1 Sicurezza

Un aspetto fondamentale che non viene sottovalutato nella specifica LoRaWAN è la sicurezza. Ogni nodo LoRa implementa al suo interno due chiavi di sicurezza uniche *AppSkey* e *NwkSkey*, le quali sono criptate secondo le specifiche AES a 128 bits. La network session key (*NwkSkey*) è la chiave che viene utilizzata per garantire l'affidabilità nella comunicazione tra dispositivo e la rete. Questa chiave inoltre, è utilizzata per verificare la validità del messaggio tramite la procedura di controllo MIC (Message Integrity Check). L'application session key (*AppSkey*) viene utilizzata per la criptazione e decriptazione del payload. Tramite questa chiave viene garantito lo scambio di informazioni in modo sicuro tra il device layer e l'application layer. Queste due chiavi (*AppSkey*, *NwkSkey*) sono uniche per ogni devices, e vengono rigenerate ad ogni volta che il dispositivo si spegne o cambia rete. Se il devices è attivato, in modo dinamico, tramite la procedura OTAA (*Over the air activation*) queste chiavi vengono generate utilizzando una terza chiave chiamata *AppKey* sempre lunga 128 bits. Contrariamente, i nodi che utilizzando la procedura APB (*Activation by personalization*), manterranno invariate le chiavi anche per sessioni diverse, rendendo necessario un intervento manuale nella eventualità di un aggiornamento delle stesse.

## 3.5 Limitazioni

Essendo una tecnologia molto giovane, non sono ancora chiari i limiti della rete LoRa e delle reti LPWAN in generale. Un punto cruciale che pone ancora molti interrogativi, è la scalabilità di queste reti. Per cercare di rispondere a questa domanda il ricercatore Maarten Wey e successivamente M.C.Bor e U.Roedig, hanno effettuato delle simulazioni sulla base dei dati forniti da Semtech. Il grafico 3.11 è tratto dalla simulazione di Maarten Wey. In esso sono riportate il numero di collisioni avvenute durante la simulazione di una rete LoRa, alla quale erano connessi 1000 nodi tramite un solo gateway. In questa simulazione viene messo in evidenza come all'aumentare del numero di messaggi inviati in un minuto, la percentuale di errore nello scambio dei dati aumenta in maniera lineare. In concordanza con questi risultati, la ricerca effettuata da M.C.Bor e U.Roedig, dimostra che per un corretto funzionamento delle reti LoRa, il numero massimo di devices contemporaneamente connessi per gateway

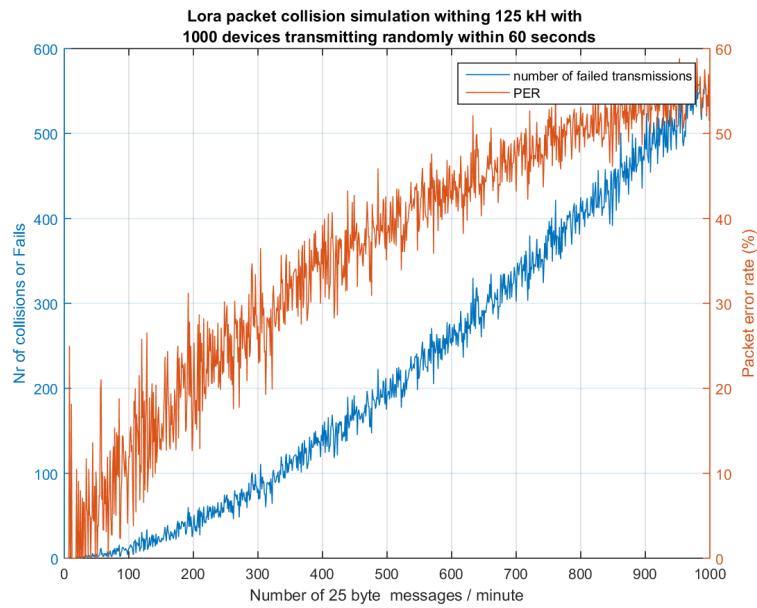


Figura 3.11: Simulazione prestazioni LoRa

è 120. È importante osservare che i risultati, ottenuti da queste simulazioni, sono basati su di una rete composta da un solo gateway e che lo SF di ogni messaggio era scelto in maniera casuale.



# **Capitolo 4**

## **Caso di studio**

Con l'evolversi della tecnologia, ed in particolare con l'avvento dell'IoT, l'aspettativa delle applicazioni basate sul cloud è cambiata. La necessità di elaborare un numero sempre maggiore di dati ha spinto, sempre più aziende, ad adottare strumenti cloud based forniti da terzi come ad esempio Microsoft Azure o Amazon AWS. Ed è utilizzando questi servizi che l'architettura classica del software monolitico ha mostrato i suoi limiti. Con architettura monolitica si intende una applicazione self-contained indipendente dalle altre applicazioni presenti nel sistema. Basandosi su questa modello, ogni software necessita di essere disinstallato e reinstallato nel momento in cui sia necessario apportare delle semplici modifiche. Per superare i limiti del modello monolitico, è stato introdotto il concetto di *microservizio*. La differenza tra una architettura monolitica ed una basata su microservizi è la modularità che quest'ultima garantisce. Prendiamo come esempio una applicazione formata da un database, un interfaccia web (client-side user interface) e una applicazione server (server-side application). L'applicazione server-side interpreterà le richieste fatte dall'utente andando ad eseguire operazioni interne al server, aggiornerà il database e fornirà il risultato all'utente finale tramite l'interfaccia web. Nel modello classico è necessario riscrivere o aggiornare l'intero software per apportare delle modifiche o aggiungere delle funzionalità.

Con il termine *microservices* si intende una architettura basata su oggetti chiamati *component* ognuno dei quali fornisce ed utilizza dei servizi. Con *component*, si intende una "parte" di software che può essere aggiornata e sostituita indipendente dalla applicazione principale. Ogni *component* si basa sull'utilizzo e la condivisione di servizi. I servizi so-

no l'equivalente delle librerie in una architettura monolitica, al contrario di queste ultime però, i servizi sono sviluppabili indipendentemente dalla applicazione principale. In questo capitolo verrà esposto lo sviluppo di due di applicativi per la gestione e l'interazione di sensori basati su LoRa, tramite l'utilizzo del framework EveryWare Software Framework ESF basato su OSGi e sviluppato da Eurotech.

## 4.1 OSGi

La tecnologia OSGi (Open Services Gateway initiative), è un insieme di specifiche che definiscono dei componenti dinamici in grado di estendere le funzionalità dell'ambiente Java. Basandosi su di una architettura a microservizi, OSGi permette la gestione in modo remoto di componenti chiamati Bundle o Deployment Package(dp), i quali possono essere installati, attivati, aggiornati e fermati senza la necessità di riavviare l'applicazione principale.

Ogni bundle è composto dai file contenenti le classi ed i metodi specifici per le funzionalità che implementa e da metadati utili al framework. Utilizzando questi ultimi OSGi è in grado di capire quali classi sono pubbliche e condivisibili con altri moduli, da quelle private e utilizzabili solo all'interno del bundle. Per rendere compatibile questa architettura con il linguaggio di programmazione Java e la Java Virtual Machine, OSGi ha strutturato il framework su vari livelli:

- *Bundles*. Bundles sono normali applicazioni JAR contenenti
- *Services*. Il service layer connette in maniera dinamica i bundles offrendo un modello publish-find-bind basato su l'interfaccia Java POJIs.
- *Services Registry*. Questo layer è composto da una serie API per la gestione dei servizi.
- *Life-Cycle*. Insieme di API per gestire il ciclo vitale dei bundle
- *Modules*. È il layer che definisce in che modo i bundle sono in grado di interagire importando ed esportando pezzi di codice.
- *Security* È un layer che opera su tutti i livelli e garantisce la sicurezza delle applicazioni OSGi andando a limitare le funzionalità del bundle.

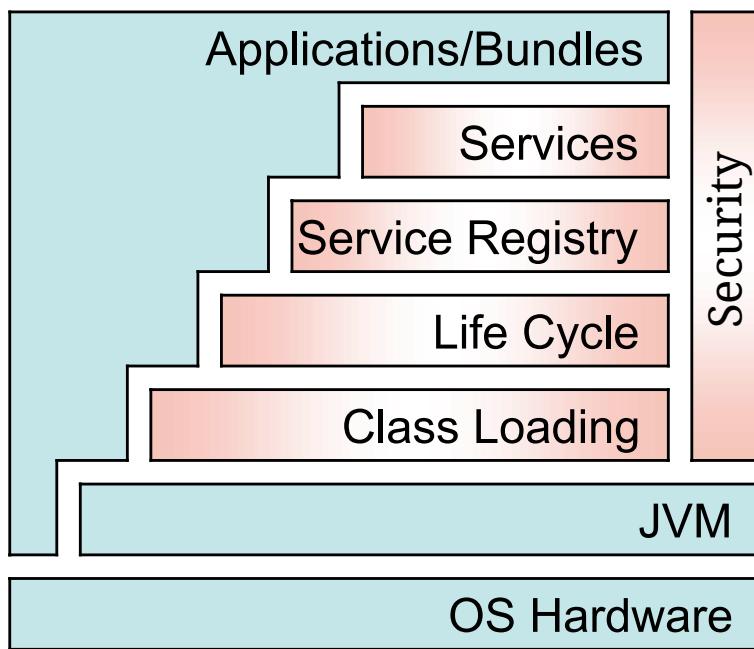


Figura 4.1: Layer OSGi

### 4.1.1 Ciclo vitale dei bundle

In un modello così dinamico, è necessario che il framework sia basato su di un software fail-safe in grado di gestire le eccezioni che si possono verificare durante l'installazione e disinstallazione di nuovi moduli. Non è raro infatti che nuovi bundle installati nel sistema, richiedano servizi non ancora disponibili o attualmente utilizzati da altri componenti. Per far fronte ai problemi riscontrabili, OSGi assegna ad ogni bundle uno *stato* tra:

- *Installed* Il bundle è stato installato nel sistema, ma non sono presenti alcune delle sue dipendenze.
- *Resolved* Il bundle è installato e le sue dipendenze presenti nel sistema.
- *Starting* Uno stadio temporaneo attraverso il quale il bundle passa prima di essere attivato.
- *Active* Il bundle è stato correttamente attivato e sta eseguendo le sue funzioni all'interno del sistema.

- *Stopping* Uno stadio temporaneo in cui il bundle passa prima di essere disattivato.
- *Uninstalled* Il bundle è stato rimosso dal container OSGi

### 4.1.2 Module layer

Il module layer è dove il framework OSGi gestisce la "modularità" di un bundle. È in questo layer che vengono processati i metadati contenuti all'interno del file MANIFEST.MF. Tramite questo file il framework OSGi è in grado di determinare le dipendenze del bundle e quali servizi è in grado di esportare.

### 4.1.3 Registrazione del servizio

Un servizio in OSGi è definito tramite una classe Java standard. Per implementare un nuovo servizio, è necessario definire quale classe o quale interfaccia si vuole fornire il servizio. La soluzione a questa problematica è l'utilizzo di registri di servizio. Ogni bundle può esporre dei metodi e registrarli attraverso il service registry. In questo modo, altri bundle possono accedere al service registry e utilizzare i metodi listati nel registro. Un bundle quindi può registrare un servizio, può utilizzare un servizio oppure può mettersi in attesa aspettando che un servizio venga registrato o eliminato. Inoltre i servizi sono dinamici, ogni bundle può mettersi in lista per richiedere un servizio mentre altri stanno ancora utilizzando il servizio.

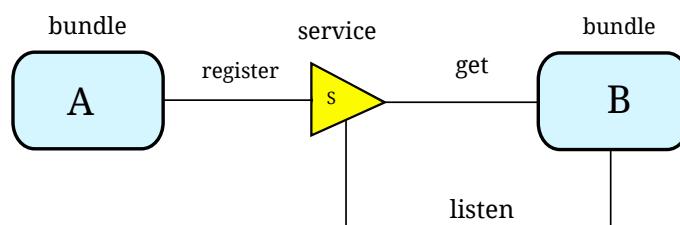


Figura 4.2: Schema utilizzo servizi OSGi

## 4.2 Everyware Software

ESF o Everyware Software è un framework che si interpone tra il sistema operativo e le applicazioni utente. Basato su Java/OSGi, ESF si pone l'obiettivo di offrire la possibilità di sviluppare applicativi, per il mercato M2M ,in maniera semplice e veloce . Ideato per operare all'interno di gateway industriali, fornisce al consumatore servizi e librerie per l'accesso delle più comuni porte di comunicazione RS232/485, GPIO, CAN. ESF tramite la sua interfaccia web, permette di controllare da remoto il comportamento dei gatway. 4.3.

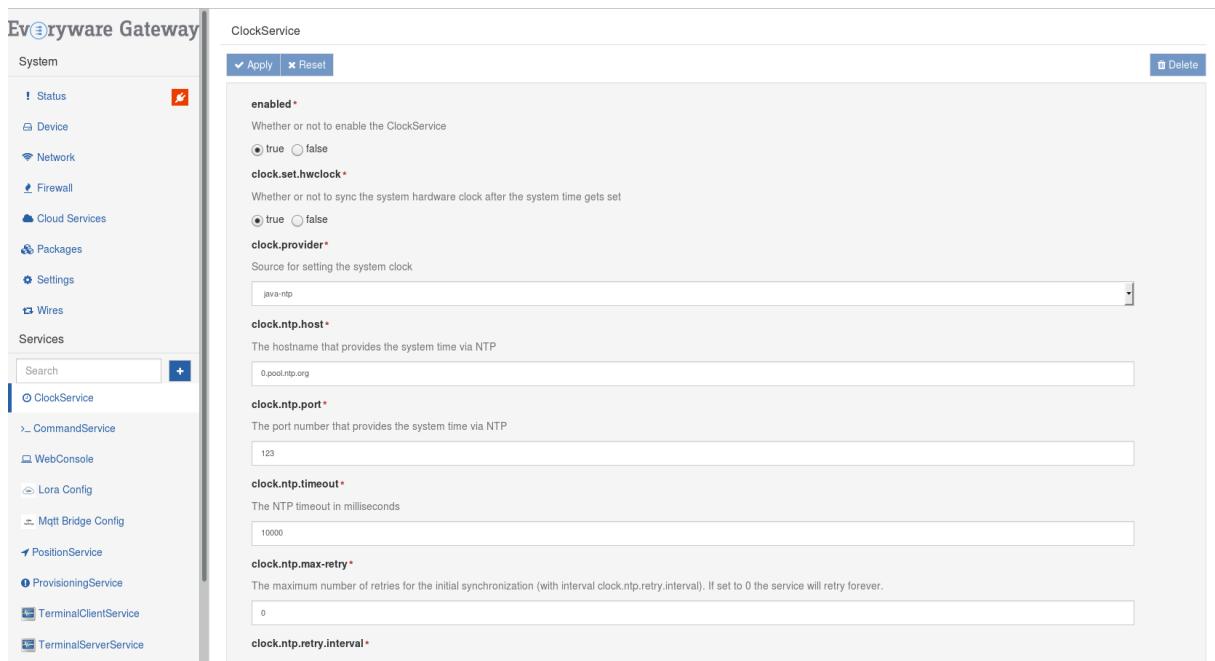


Figura 4.3: Interfaccia web ESF

## 4.3 Hardware utilizzato

### 4.3.1 SX1301

Per la ricezione dei pacchetti LoRa, è stato utilizzato il chip SX1301 prodotto da Semtech e collegato al gateway ReliaGate 10-11.

### Struttura interna

Per quanto riguarda la struttura interna del modulo radio, non si hanno molte informazioni dato che la tecnologia è proprietaria di Semtech. Nella documentazione ufficiale è presente una rappresentazione grafica dei vari blocchi interni al chip.

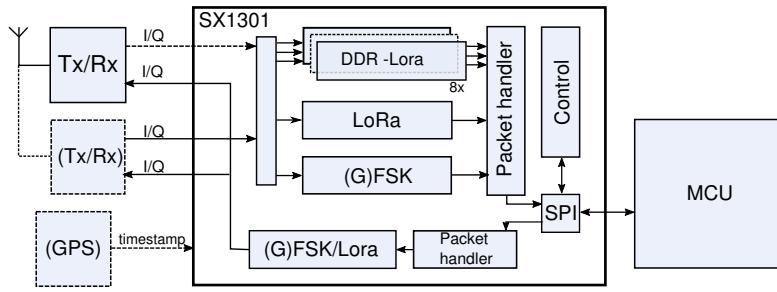


Figura 4.4: Struttura interna ricevitore SX1301

Come scritto nella documentazione [?] e intuibile dalla figura 4.4 il chip è in grado di scansionare contemporaneamente 8 canali diversi (IF0 a IF7) permettendoli di rimanere in ascolto per garantire la ricezione dei segnali con SF diversi. Inoltre data la quasi ortogonalità degli Spreading Factor , il chip è in grado di ricevere un pacchetto con uno Spreading Factor  $i$  anche nel caso in cui si sovrapponga ad un altro pacchetto con Spreading Factor pari a  $j$ , fintanto che  $i \neq j$ . Questa pseudo-ortogonalità utilizzata in LoRa, permette al ricevitore SX1301 di demodulare fino ad un massimo di 8 pacchetti contemporaneamente. Nella tabella 4.1 sono riportate le caratteristiche elettriche massime del chip SX1301. Il Chip, supporta tensioni di alimentazione fino a 4V e come è possibile osservare il range di temperatura in cui il chip può operare è molto ampio, rendendolo ideale per applicazioni esterne ed interne.

Utilizzando i valori nominali riportati nella tabella 4.2, si hanno valori di corrente pari a 1[uA] in idle e di 5[mA] in pieno funzionamento. Per la ricezione del segnale è stata utilizzata una antenna omnidirezionale, ideata per la banda degli 868[MHz] con un guadagno pari a 3[dB].

### 4.3.2 ReliaGATE

Il gateway a cui è collegato il modulo SX1301 è il ReliaGATE 10-11 prodotto da Eurotech. Al suo interno troviamo un processore Texas Instru-

Parameter	Symbol	Conditions	Value
IO power supply to VSS	$V_{DDIO,ABSMAX}$		-0.5 V to 4.0 V
Core power supply to VSS	$V_{DDCORE,ABSMAX}$		-0.5 V to 2.0 V
Storage temperature	$T_{J,STORE}$		-50 °C to 150 °C
Junction temperature	$T_{J,ABSMAX}$		-40 °C to 125 °C
Pin voltage on IO and Clock pins	$V_{DPIN,ABSMAX}$		-0.3 V to $VDDIO + 0.3$ V
Peak reflow temperature	$T_{PKG}$		260 °C
Latchup	$I_{LUP}$	JESD78D, class I	+/-100 mA
Humidity	$H_R$		0 – 95 %
ESD	HBM	Human Body Model JESD22-A114 CLASS 2	2 kV
	CDM	Charged Device Model JESD22-C101 CLASS III	300 V

Tabella 4.1: Caratteristiche elettriche SX1301

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Digital IO supply	$V_{DDIO}$	Operating Conditions for Electrical Specification	3.0		3.6	V
Digital core supply	$V_{DDCORE}$	Operating Conditions for Electrical Specification	1.75		1.85	V
Ambient operating temperature	$T_A$	With chip paddle soldered to PCB ground plane with minimum 100 cm <sup>2</sup> air exposed area and heat sink	-40		85	°C

Tabella 4.2: Caratteristiche elettriche SX1301

ments TI AM335X Cortex-A8 equipaggiato con 512MB di RAM e 4GB di storage eMMC. Il ReliaGATE offre una vasta gamma di porte tra cui 232/485, 2CAN bus, 2 porte USB e 2 porte Ethernet, inoltre ha connettività Bluetooth, WiFi e GPS. Al suo interno è installato Everyware™ Software Framework (ESF) versione 5.



Figura 4.5: ReliaGATE 10-11

## 4.4 Architettura del software

Per integrare ESF con il ricevitore lora SX1301, è stato necessario l'utilizzo di due software aggiuntivi. Il primo è LoRa packet forwarder fornito da Semtech. Questo applicativo permette di comunicare con le periferiche di basso livello presenti nel ReliaGATE 10-11 in modo da astrarre ad un più alto livello i dati ricevuti dal ricevitore . L'altra applicazione utilizzata, è LoRa Gateway Bridge. Tramite questa applicazione è stato possibile reindirizzare i dati forniti dal packet forwarder ad un server MQTT.

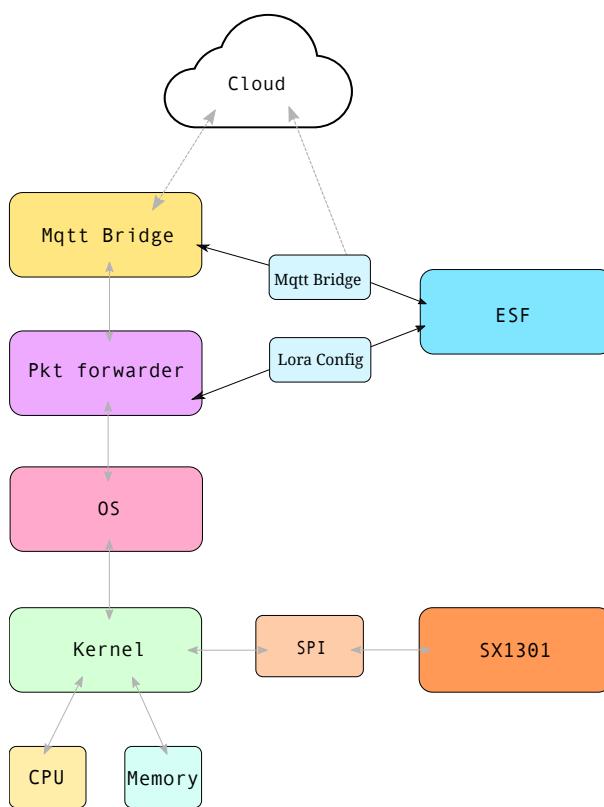


Figura 4.6: Architettura del software

### 4.4.1 Semtech packet forwarder

Il packet forwarder è un software che permette la ricezione e l'invio di pacchetti radio Lora , tramite una connessione SPI con il device SX1301.

Nel caso di ricezione di un pacchetto, l'applicativo incapsula i dati ricevuti in un formato UDP e li ritrasmette nella rete internet/intranet. Per la sua configurazione viene utilizzato un file Json nel quale troviamo tutte le varie opzioni di configurazione per i moduli radio presenti al interno del chip.

---

```
1 "SX1301_conf": {
2     "lorawan_public": true,
3     "clksrc": 1,
4     "radio_0": {
5         "enable": true,
6         "type": "SX1257",
7         "freq": 867500000,
8         "rss_i_offset": -166.0,
9         "tx_enable": true
10    },
11    "radio_1": {
12        "enable": true,
13        "type": "SX1257",
14        "freq": 868500000,
15        "rss_i_offset": -166.0,
16        "tx_enable": false
17    },
18    "chan_multiSF_0": {
19        "enable": true,
20        "radio": 1,
21        "if": -400000
22    },
23}
```

---

```
173 "gateway_conf": {
174     "gateway_ID": "AA555A0000000000",
175     "server_address": "0.0.0.0",
176     "serv_port_up": 1703,
177     "serv_port_down": 1702,
178     "keepalive_interval": 10,
179     "stat_interval": 30,
180     "push_timeout_ms": 100,
```

---

```

181     "forward_crc_disabled": false,
182     "forward_crc_valid": true,
183     "forward_crc_error": false
184 }
```

---

#### 4.4.2 LoRa Gateway Bridge

LoRa Gateway Bridge è un applicativo in grado di identificare i vari pacchetti UDP inviati dal packet forwarder e inoltrarli ad un Broker MQTT. Il software è scritto nel linguaggio GO e permette una configurazione tramite linea di comando. Per renderlo interfaciabile con ESF, sono state apportate delle modifiche al codice; in particolare è stata aggiunta la possibilità di specificare il publish e il subscribed topic, i quali erano hard-coded. Il codice sottostante, rappresenta un esempio di pacchetto LoRa inoltrato dal LoRa Gateway Bridge ad un broker MQTT

---

```

1
2     "rxInfo": {
3         "mac": "aa555a0000000101",
4         "time": "2017-08-26T17:01:56.870329Z",
5         "timestamp": 1122401132,
6         "frequency": 868500000,
7         "channel": 1,
8         "rfChain": 1,
9         "crcStatus": 1,
10        "codeRate": "4/5",
11        "rssi": -74,
12        "loRaSNR": 8.2,
13        "size": 29,
14        "dataRate": {
15            "modulation": "LORA",
16            "spreadFactor": 12,
17            "bandwidth": 125
18        }
19    },
20    "phyPayload": "QKMbMroAXQACF8XVkpmbTIoBHzV0iNtTa0cbkCo="
21
```

---

### 4.4.3 Bundle OSGi

Per gestire i due software preinstallati si è optato per la creazione di due applicativi osgi distinti Lora Config e Mqtt Bridge.

#### Lora Config

Il primo applicativo chiamato Lora Config si pone il compito di leggere ed interpretare il file di configurazione utilizzato dal programma *Packet Forwarder*, per poi andare ad esporre i parametri principali all'utente tramite l'interfaccia web di ESF. La libreria utilizzata per manipolare i file di tipo Json è

```
import com.eclipsesource.json
```

Tramite questa libreria vengono riempiti i campi della classe LoraSettings. Il file Json è composto da due parti. Nella prima parte troviamo tutte le impostazioni per la configurazione dei canali (IF0 a IF7) , nella seconda parte sono presenti le impostazioni per il forward dei pacchetti. Per semplificare la gestione del bundle, sono state create due classi distinte SX1301Configuration e GatewayConfiguration, accessibili tramite la classe LoraSettings.

---

```
1 public class LoraSettings {
2     private SX1301Configuration sx1301Conf;
3     private GatewayConfiguration gatewayConf;
4 }
```

---

Per applicare le modifiche apportate alla configurazione, è necessario che il packet forwarder venga riavviato. Per eseguire questa operazione si è scelto di utilizzare la libreria

```
import com.apache.commons.exec
```

la quale fornisce delle API per chiamare processi esterni, in particolare si è scelto di utilizzare l'utility di sistema *pkill* per terminare il processo del pkt forwarder.

---

```

1   public void startPktForwarder() {
2       DefaultExecutor pktExecutor = new DefaultExecutor();
3       CommandLine pktCmdLine = new CommandLine(KEY_PKT_BIN);
4       pktCmdLine.addArgument("start");
5       pktExecutor.setExitValue(0);
6       try {
7
8           pktExecutor.execute(pktCmdLine);
9           s_logger.info("start PKT");
10      } catch (Exception e) {
11          s_logger.warn("Coulden't start pkt forwarder");
12      }
13  }
14
15  public void stopPktForwarder() {
16      DefaultExecutor pktExecutor = new DefaultExecutor();
17      CommandLine pktCmdLine = new CommandLine("pkill");
18      pktCmdLine.addArgument("basic_pkt_fwd");
19      pktExecutor.setExitValue(0);
20      try {
21
22          pktExecutor.execute(pktCmdLine);
23          s_logger.info("Stop PKT");
24      } catch (Exception e) {
25          s_logger.warn("Coulden't stop pkt forwarder");
26      }
27  }

```

---

## Mqtt Bridge Config

Il secondo bundle prende il nome di MQTT Bridge Config e ha lo scopo di esporre , tramite l'interfaccia web di ESF, le varie opzioni a linea di comando dell'applicativo *Lora Gateway Bridge*. Anche in questo caso è necessario il riavvio del applicativo per fare in modo che le modifiche abbiano effetto. Come nel bundle precedente è stata usata la libreria

```
import com.apache.commons.exec
```

Tramite l'interfaccia web è possibile modificare i seguenti parametri:

- il topic sul quale pubblicare i messaggi ricevuti in formato UDP dal *pkt forwarder*.
- il topic al quale rimanere il gateway dovrà rimanere in ascolto .
- su quale indirizzo e porta mandare/ricevere i pacchetti UDP.
- a quale broker MQTT iscriversi.
- l'username e password per connettersi al broker.

## 4.5 Misurazioni

Uno degli aspetti più discussi riguardo alle tecnologie LPWAN è la massima copertura raggiungibile. E' un dettaglio cruciale per la stima corretta dei costi necessari alla creazione di una rete pubblica o privata. Per questo motivo terminato lo sviluppo dei bundle si è scelto di testare la copertura fornita dal ricevitore SX1301. In via sperimentale è stato installato il gateway ReliaGATE 10-11 ad una altezza di circa 11m in una zona suburbana pianeggiante. La prova di ricezione è stata condotta per

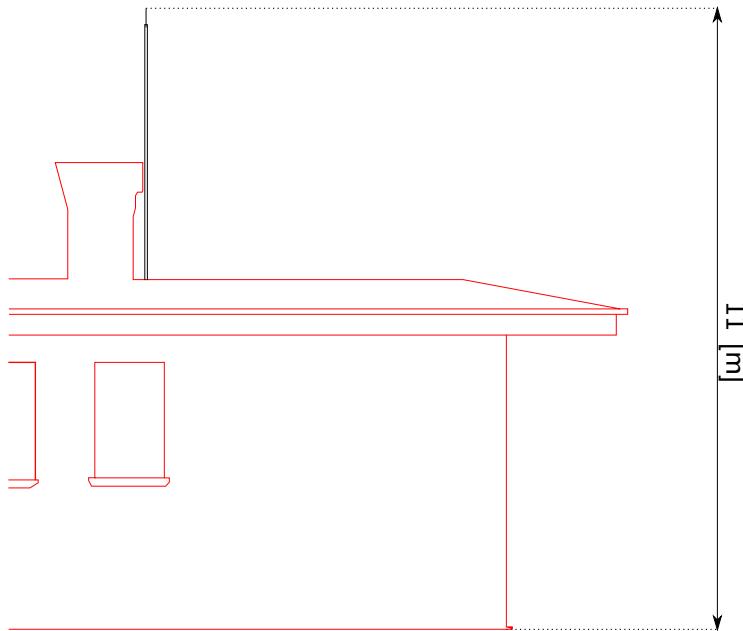


Figura 4.7: Installazione dell'antenna

tentativi cercando , per quanto possibile, di testare diversi punti distribuiti alla stessa distanza radiale. Il dispositivo utilizzato per l'invio dei pacchetti LoRa è prodotto dalla Semtech e prende il nome di LoRa Mote.



Figura 4.8: Dispositivo LoRa Mote

Per constatare l'avvenuta ricezione del messaggio è stata utilizzata l'applicazione Android gratuita My MQTT, tramite la quale è possibile iscriversi ad un topic predefinito ed rimanere in ascolto dei messaggi pubblicati in esso. Come broker MQTT si è scelto di utilizzare il broker open source "mosquitto.org".

#### 4.5.1 Osservazioni

Nella mappa ogni colore corrisponde ad un livello RSSI ( Received Signal Strength Indicator) con cui il quale il messaggio inviato in quel punto è stato ricevuto. Dovendo testare la massima distanza di comunicazione l'algoritmo ADR è stato disattivato, permettendo così al device di non cambiare configurazione durante i vari test. LoRa Mote è stato configurato per l'invio di messaggi con uno Spread Factor pari a 12 e una larghezza di banda pari a 125[Khz]. La distanza massima raggiunta varia di alcuni chilometri in base alla conformazione del territorio. In assenza di edifici in linea d'aria, tra gateway e il dispositivo LoRa Mote, è stato possibile ricevere correttamente un pacchetto alla distanza di 8,2[Km]. È utile ricordare che l'antenna utilizzata non apportava nessun guadagno al ricevitore, infatti è un'antenna adatta alla prototipazione e non ad un utilizzo esterno. Considerato questo ultimo fatto è possibile affermare che i risultati ottenuti sono in linea con quelli dichiarati da Semtech.

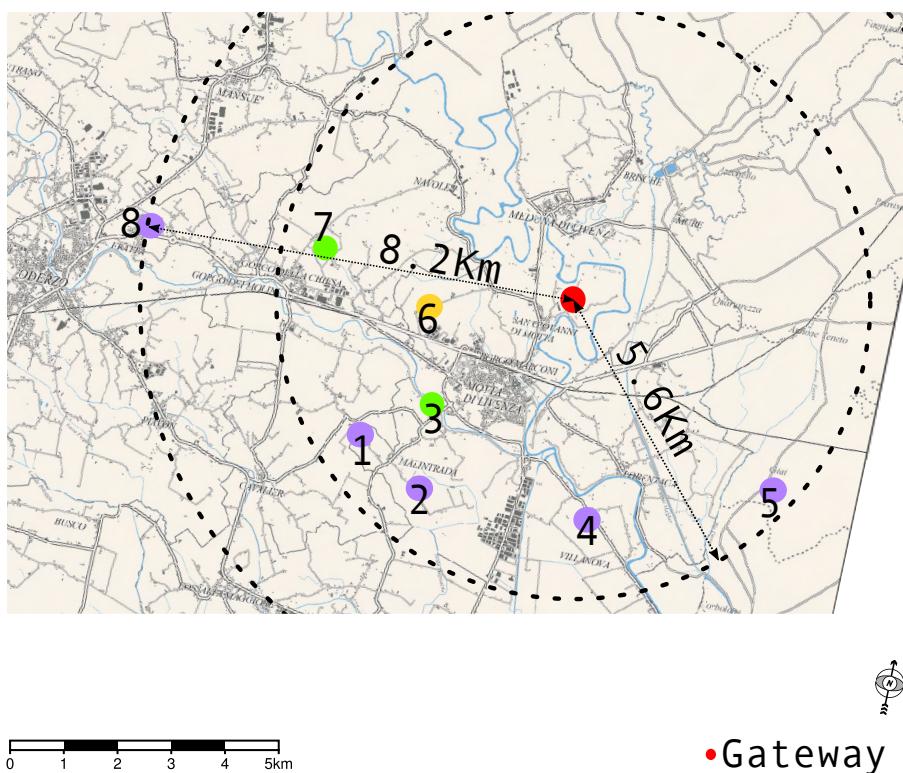


Figura 4.9: Copertura Lora



# **Capitolo 5**

## **Conclusioni**

Lo scopo dell’elaborato era di introdurre la visione di Internet of Things e testare un possibile sviluppo applicativo tramite l’utilizzo della tecnologia LoRa ed il framework ESF sviluppato da Eurotech.

Dopo un’introduzione relativa agli ambiti applicativi e alle problematiche da affrontare in questa visione, sono state trattate le relative tecnologie abilitanti per la realizzazione di questo paradigma mettendone in evidenza i vari pregi e difetti. Successivamente si è parlato della modulazione LoRa e lo standard LoRaWAN andandone a trattare gli aspetti principali che caratterizzano questa tecnologia. Infine si è approfondito il concetto di microservizi e di come questo concetto è stato implementato nella piattaforma Everyware Software Framework di Eurotech. Questa piattaforma si è dimostrata particolarmente adatta a questa visione, poiché permette di risolvere alcuni dei principali problemi legati allo sviluppo di applicazioni M2M, in particolare all’eterogeneità dell’hardware relativa ai dispositivi in gioco. L’approccio con il framework OSGi risulta molto efficace poiché permette la scrittura di software di ridotta complessità e dinamicamente aggiornabile, riuscendo così a ridurre il tempo di sviluppo e di conseguenza il time to market dei nuovi prodotti. Tramite questa piattaforma e l’utilizzo di due software aggiuntivi sono stati sviluppati due bundle , Mqtt Bridge Config e Lora Config, per la gestione di un ricevitore LoRa, connesso al gateway prodotto da Eurotech ReliaGATE 10-11 . A sviluppo completato è stata condotta una prova per testare la distanza massima raggiungibile dal modulo radio in dotazione, la quale è risultata coerente con i dati forniti dalla casa produttrice del ricevitore.



# Bibliografia

- [1] Gartner. Gartner says 8.4 billion connected 'things' will be in use in 2017, up 31 percent from 2016. *[Online]*, 2016. <http://www.gartner.com/newsroom/id/3598917>.
- [2] LTD Hauwei Technologies co. Nb-iot white paper. *[Online]*, 2015. [http://www.huawei.com/minisite/iot/img/nb\\_iot\\_whitepaper\\_en.pdf](http://www.huawei.com/minisite/iot/img/nb_iot_whitepaper_en.pdf).
- [3] Semtech. An1200.22 lora™ modulation basics. *[Online]*, 2015. <http://www.semtech.com/images/datasheet/an1200.22.pdf>.
- [4] Semtech. Lora™ modulation basics. *[Online]*, 2015. <http://www.semtech.com/images/datasheet/an1200.22.pdf>.
- [5] Semtech. Lorawan 101 a technical introduction. *[Online]*, 2017. [https://docs.wixstatic.com/ugd/eccc1a\\_20fe760334f84a9788c5b11820281bd0.pdf](https://docs.wixstatic.com/ugd/eccc1a_20fe760334f84a9788c5b11820281bd0.pdf).
- [6] Verizion. State of the market: Internet of things 2016. *[Online]*, 2016. <https://www.verizon.com/about/sites/default/files/state-of-the-internet-of-things-market-report-2016.pdf>.