

POLITECNICO DI TORINO

Collegio di Ingegneria Informatica, del Cinema e Meccatronica

**Corso di Laurea Magistrale
in Ingegneria Informatica (Computer Engineering)**

Tesi di Laurea Magistrale

Tecnologie per l'IoT urbano

*Studio delle prestazioni delle reti LoRaWAN™ nel contesto urbano di
Torino*



Relatore

prof. Giovanni Malnati

Candidato

Michele di Girolamo

Dicembre 2017

Indice

1. Introduzione.....	1
2. LPWANs per l’IoT.....	3
2.1. Casi d’uso.....	3
2.2. Tecnologie.....	4
2.2.1. Licensed Spectrum.....	5
2.2.1.1. EC-GSM-IoT.....	5
2.2.1.2. LTE MTC Cat-M1.....	5
2.2.1.3. NB-IoT.....	6
2.2.2. Unlicensed Spectrum.....	6
2.2.2.1. LoRaWAN.....	6
2.2.2.2. Sigfox.....	7
2.2.2.3. Weightless.....	7
2.2.2.4. Regolamentazioni delle bande “unlicensed”.....	8
2.2.3. Le possibilità a Torino.....	11
3. La rete LoRaWAN.....	12
3.1. Il layer PHY: LoRa.....	14
3.2. Il layer MAC: LoRa MAC.....	19
3.3. Le reti LoRaWAN attive.....	24
4. Scalabilità di LoRaWAN.....	26
4.1. Il modello.....	27
4.2. Il simulatore.....	32
4.2.1. Il progetto.....	33
4.2.1.1. Analisi dei profili applicativi.....	33
4.2.1.2. Distribuzione dei dispositivi.....	33
4.2.1.3. Pianificazione ed invio dei pacchetti.....	34
4.2.1.4. Analisi delle interferenze.....	35
4.2.1.5. Decodifica dei pacchetti.....	37
4.2.2. L’implementazione.....	37
4.3. Le prestazioni.....	39
4.3.1. Validazione del simulatore.....	39

4.3.2.	Analisi del capture effect.....	41
4.3.3.	Simulazioni della cella LoRaWAN.....	44
4.3.3.1.	I profili applicativi.....	44
4.3.3.2.	I risultati	47
5.	Copertura di LoRaWAN	54
5.1.	Realizzazione dei test	54
5.2.	Risultati	57
6.	Conclusioni e lavori futuri.....	59
7.	Riferimenti	61

1. Introduzione

L'IoT, "Internet of Things" ovvero "Internet delle cose", racchiude in sé un mondo vasto, complesso ed eterogeneo. In letteratura vi sono diversi studi e articoli che ne descrivono e spiegano il concetto [1, 2, 3]. Riassumendo i loro risultati possiamo affermare che l'IoT è un insieme di applicazioni, sistemi e servizi basati su oggetti "intelligenti". Tale intelligenza consiste nella capacità di connettersi ad internet tramite un qualsiasi mezzo di comunicazione per inviare e/o ricevere informazioni.

Poiché ad oggi qualsiasi oggetto può diventare smart, esistono svariati scenari e casi d'uso adatti all'IoT. Vi sono prodotti per la casa (domotica), applicazioni per il benessere e la salute delle persone, sistemi di monitoraggio ambientale, progetti per smart cities e molto altro.

Dalla diversità del panorama accennato si evince che i requisiti dei diversi sistemi possano variare sensibilmente l'uno dall'altro in termini di quantità di dati da trasmettere, intervalli di trasmissione, consumi energetici, collocazione dei dispositivi, etc. È necessario allora capire sino in fondo le tecnologie per scegliere correttamente quale impiegare nella realizzazione di nuovi progetti per l'IoT in modo da rispettarne le necessità.

Tra i parametri da considerare vi sono senza dubbio il mezzo di comunicazione e l'alimentazione dei dispositivi. Questi due aspetti cruciali dipendono fortemente dall'ambiente in cui si installerà l'applicazione IoT. Calandoci nello scenario urbano si necessita spesso di comunicazioni a lungo raggio poiché i punti di accesso ad Internet sono distribuiti su aree molto vaste; inoltre, per quanto riguarda l'energia, non è sempre possibile accedere alla rete elettrica, quindi può risultare necessario l'impiego di batterie da utilizzare con parsimonia da parte dei dispositivi.

Per rispondere a questa richiesta di connettività a basso consumo è nato un insieme di tecnologie dette LPWAN, "Low Power Wide Area Network" ovvero "rete vasta a bassi consumi", sulle quali gli operatori telefonici nazionali hanno effettuato investimenti più o meno rilevanti in previsione di una importante crescita del mercato dell'IoT, attesa per il 2020 [4]. Nonostante l'andamento positivo oggettivamente riscontrato nell'aumento dei dispositivi connessi ad internet, i numeri raggiunti ad oggi risultano significativamente inferiori rispetto alle stime presentate negli ultimi anni [5], e ciò ha portato ad un raffreddamento dell'interesse delle compagnie sulle LPWANs. In Italia solo alcuni operatori telefonici offrono servizi per l'IoT e sono poche le aziende locali nate per investire su tali tecnologie.

Dato il contesto incerto e la realtà sperimentale del settore smart italiano, si capisce come rendere una città intelligente connettendola alla rete non sia facile o scontato. Occorre dapprima determinare le possibilità concrete presenti sul territorio, analizzare le prestazioni dichiarate delle tecnologie e verificarne l'affidabilità.

Il presente lavoro ha come obiettivo proporre un'analisi delle capacità di una delle tecnologie emersa negli ultimi anni: LoRaWAN, "Long Range Wide Area Network". In particolare, partendo dagli studi e dagli articoli realizzati dalla comunità scientifica, tenta di quantificare la scalabilità della rete in termini di dispositivi connessi e di verificare la copertura del segnale radio raggiungibile nell'ambito urbano di Torino.

La tesi si sviluppa dunque con la seguente struttura:

- Il secondo capitolo offre una panoramica sui casi d'uso delle LPWANs per l'IoT soffermandosi sulle applicazioni urbane. Inoltre elenca le principali tecnologie LPWAN presenti ad oggi sul mercato descrivendo in particolare lo stato dell'arte in Italia e nel capoluogo piemontese.
- Il terzo capitolo descrive lo stack di rete LoRaWAN e ne analizza i due componenti principali: il livello fisico LoRa, "Long Range", e il sottolivello MAC, "Medium Access Control", LoRaMAC.
- Il quarto capitolo presenta il modello d'interferenza utilizzato per studiare la scalabilità di LoRaWAN, il simulatore sviluppato per implementarlo, le procedure seguite durante le simulazioni e i risultati ottenuti.
- Il quinto capitolo riporta i dettagli riguardanti i test di copertura svolti a Torino.
- Il sesto ed ultimo capitolo conclude il lavoro.

2. LPWANs per l'IoT

Le LPWANs sono reti che utilizzano un livello fisico wireless capace di offrire comunicazioni a lunga distanza e con consumi energetici contenuti a discapito di data rate piuttosto esigui. Esse sono particolarmente adatte al paradigma IoT in quanto le caratteristiche di trasmissione si sposano con i requisiti dei principali casi d'uso dell'“Internet delle Cose”. Inoltre vi è un vantaggio economico legato al costo contenuto dei chip radio, data la loro relativa semplicità [6], che li rende facilmente installabili in larga scala negli “oggetti intelligenti”.

2.1. Casi d'uso

Utenze energetiche	Monitoraggio dei consumi di gas e acqua
	Monitoraggio dei generatori di energie rinnovabili
	Smart Grid
Logistica	Monitoraggio di posizione e stato delle risorse (ad es. beni industriali, containers, veicoli)
Impianti industriali	Monitoraggio di cisterne per sostanze pericolose (ad es. petrolchimici, rifiuti, gas)
	Controllo dei macchinari
	Monitoraggio di distributori automatici
Città intelligenti	Sensori per i parcheggi
	Gestione dei rifiuti
	Controllo remoto dell'illuminazione
Ambiente & Agricoltura	Monitoraggio del bestiame e della fauna selvatica
	Controllo dei terreni
	Monitoraggio in tempo reale per allerte ambientali e raccolta dati (ad es. inquinamento dell'aria, temperatura)
Prodotti di consumo e per la salute	Dispositivi indossabili (ad es. smartwatch)
	Localizzatori di persone e animali
	Elettrodomestici
	Biciclette intelligenti (dispositivi per la localizzazione)
	Telemonitoraggio medico
Edifici intelligenti	Rilevatori di fumo
	Sistemi di allarme
	Domotica

Tabella 1. Casi d'uso per LPWANs. Tratto da [7].

Le LPWANs possono essere alla base di molti casi d'uso dell'IoT [7]. Possiamo suddividere gli scenari d'utilizzo in applicazioni per le "Città Intelligenti", impianti per "Edifici Intelligenti", sistemi per il monitoraggio dell'ambiente e dell'agricoltura, prodotti per la salute, strumenti utili alla logistica, apparecchi per il controllo delle utenze domestiche e sistemi di controllo per impianti industriali.

Tra tutte le applicazioni elencate ne individuiamo alcune che, poste in un contesto urbano, offrono strumenti per la gestione delle risorse e servizi utili per i cittadini. Ne sono un esempio:

- Gestione dei rifiuti: per ottimizzare lo svuotamento dei cassonetti;
- Controllo remoto dell'illuminazione: per accertarsi che gli impianti lungo le strade siano funzionanti e attivi solo quando serve;
- Monitoraggio ambientale: per raccogliere dati riguardanti l'inquinamento dell'aria, l'inquinamento acustico, la temperatura, etc.;
- Sensori per i parcheggi: per sapere quanto sia facile trovare un posto auto in una determinata zona della città e assicurarsi che tutti paghino le soste.

In aggiunta questa tesi considera anche un sistema di monitoraggio dei mezzi pubblici capace di trasmettere ad una centrale operativa il numero di passeggeri presenti su un particolare mezzo in tempo reale, si riporteranno maggiori dettagli a riguardo nel paragrafo 4.3.3.1.

In tutti questi casi d'uso abbiamo necessità di inviare e/o ricevere dati quindi è necessario dotare i dispositivi di una connessione. Ovviamente una possibilità per fornire connettività in aree molto vaste potrebbe essere l'utilizzo delle tecnologie impiegate dai moderni cellulari. Analizzando le precedenti applicazioni notiamo però come la tipologia di traffico prodotta e la disponibilità energetica dei dispositivi impiegati non trovino soluzione nelle classiche reti 2/3/4G. Infatti i dati da inviare sono pochi, payload applicativi solitamente nell'ordine delle decine di byte, e gli intervalli di trasmissione possono essere anche molto lunghi, pertanto una banda larga risulta essere uno spreco energetico. Le LPWANs sono nate proprio dalla mancanza di soluzioni capaci di soddisfare queste necessità.

2.2. Tecnologie

Il mercato offre diverse tecnologie per le LPWANs suddivise in due famiglie. Vi sono le LPWANs che utilizzano frequenze di trasmissioni appartenenti a bande a pagamento ("licensed spectrum") e altre invece che sfruttano le bande libere da contratti ("unlicensed spectrum") ma regolamentate da vincoli nazionali ed internazionali.

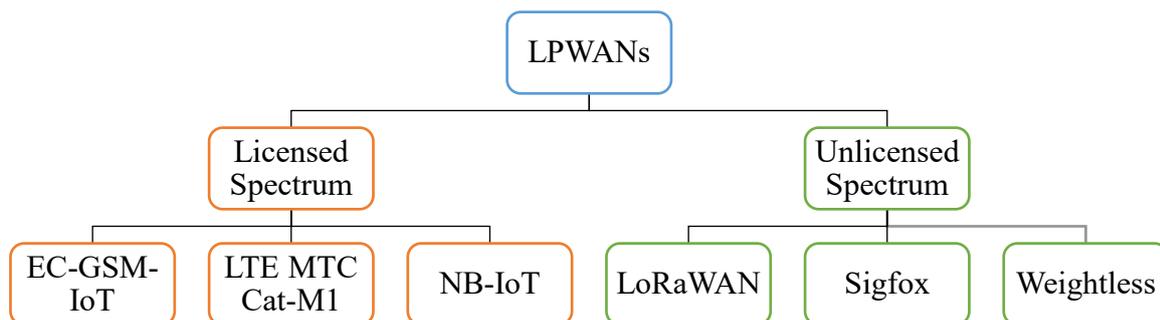


Figura 1. Gerarchia LPWANs.

2.2.1. Licensed Spectrum

Le tecnologie di trasmissione LPWANs “licensed spectrum” derivano dalle classiche reti di telecomunicazioni offerte dagli operatori telefonici e pertanto sono compatibili, almeno in parte, con le infrastrutture presenti ad oggi. Esse sfruttano lo spettro delle reti 2/3/4G ma ottimizzano il consumo energetico utilizzando protocolli di comunicazione che prevedono una minore attività da parte dei dispositivi ed una minore complessità degli stessi. Conseguentemente i costi di produzione e di vendita di tali prodotti risultano contenuti così da permetterne l’impiego in larga scala.

Il 3GPP, “3th Generation Partnership Project”, ente che negli anni si impegna a standardizzare tecnologie di telecomunicazione, ha definito in uno dei suoi ultimi documenti [6] le specifiche di tre soluzioni LPWANs in banda a pagamento: EC-GSM-IoT, “Extended Coverage - Global System for Mobile communications - Internet of Things”, LTE MTC Cat-M1, “Long Term Evolution – Machine Type Communication Category - M1” anche abbreviato “LTE-M”, e NB-IoT, “Narrow Band – Internet of Things”. Si presentano di seguito le loro principali caratteristiche.

2.2.1.1. EC-GSM-IoT

EC-GSM-IoT è uno standard avanzato dell’eGPRS, “enhanced General Packet Radio Service”, migliorato in termini di consumi, complessità dei dispositivi e sicurezza. Esso implementa diverse tecniche di risparmio energetico tra cui il PSM, “Power Saving Mode”, e l’eDRX, “extended idle-mode Discontinuous Reception”, che permettono alle batterie utilizzate per alimentare i dispositivi di durare sino a 10 anni [6, 7], requisito imposto dal 3GPP a questa tecnologia. Inoltre l’EC-GSM-IoT utilizza sistemi per la sicurezza analoghi a quelli presenti sulle reti LTE, più recenti rispetto a quelli del GSM. Punto di forza di questa tecnologia è la retro-compatibilità con le infrastrutture di rete 2G che, grazie ad alcuni aggiornamenti software, sono in grado di supportare le nuove comunicazioni offrendo copertura nelle zone del mondo in cui le reti 4G non sono ancora presenti.

2.2.1.2. LTE MTC Cat-M1

LTE MTC Cat-M1 è uno degli standard della famiglia LTE con performance più contenute in termini di trasmissione. Esso è dotato di una connessione half-duplex con throughput sia in uplink che in downlink inferiore ad 1Mbps. Anche questa tecnologia sfrutta il PSM e l’eDRX tra le altre tecniche di

risparmio energetico. LTE-M condivide diverse caratteristiche interessanti con alcune delle categorie LTE tra cui una banda non eccessivamente ridotta ed una bassa latenza nella trasmissione. Queste ultime due proprietà lo rendono adatto ad applicazioni IoT che richiedono anche traffico voce.

2.2.1.3. NB-IoT

NB-IoT, più precisamente LTE Cat-NB1, “Long Term Evolution Category - NB1”, è uno standard che riutilizza diversi componenti di LTE, sia a livello fisico che nei protocolli di livello più alto. È dotato di connessioni ancora meno performanti rispetto a LTE-M ma è in grado di supportare fino a 50000 dispositivi con una singola base station e garantisce una lunga durata delle batterie dei dispositivi che lo utilizzano: anche in questo caso infatti, come per EC-GSM-IoT, il 3GPP ha imposto dei vincoli sul risparmio energetico.

	EC-GSM-IoT	LTE MTC Cat-M1	NB-IoT
LTE user equipment category	N/A	Cat-M1	Cat-NB1
Spectrum	Licensed GSM bands	Licensed LTE bands in-band	Licensed LTE in-band guard-band stand-alone
Bandwidth	200 kHz	1.08 MHz (1.4 MHz carrier bandwidth)	180 kHz (200 kHz carrier bandwidth)
Max. downlink data rate	<140 kbps	<1 Mbps	<170 kbps
Max. uplink data rate	<140 kbps	<1 Mbps	<250 kbps

Tabella 2. Confronto tra le prestazioni offerte dalle 3 tecnologie LPWAN operanti in banda a pagamento. Tratto da [8].

2.2.2. Unlicensed Spectrum

Le tecnologie LPWANs “unlicensed spectrum” sono tutte soluzioni che sfruttano diverse bande prive di licenza per la trasmissione dei dati. Tali bande non richiedono il pagamento di tasse per il loro utilizzo ma sono regolamentate da organi nazionali ed internazionali per assicurarne una fruizione condivisa e imparziale a tutti gli utenti interessati. Si presentano di seguito alcune delle tecnologie “unlicensed spectrum” più diffuse e successivamente si forniscono i dettagli sulle regolamentazioni presenti in Europa per le bande utilizzate.

2.2.2.1. LoRaWAN

LoRaWAN è uno stack di rete prodotto dalla LoRa Alliance, un consorzio di diverse aziende delle telecomunicazioni. Il livello fisico della rete utilizza LoRa, una tecnologia di modulazione wireless proprietaria di Semtech basata sulla tecnica CSS, “Chirp Spread Spectrum”, in grado di garantire una

elevata resistenza alle interferenze e al rumore. La trasmissione è bidirezionale, uplink e downlink, ma i dispositivi sono suddivisi in classi (A, B e C) che prevedono capacità di invio e ricezione dei dati differenti. A livello MAC, LoRaWAN presenta un protocollo aperto che permette a chiunque di dotarsi di gateway e creare cloud privati per lo sviluppo di applicazioni IoT [9].

2.2.2.2. Sigfox

Sigfox è una startup nata nel 2009 ed è stata tra le prime ad investire sulle LPWANs. La sua tecnologia è basata su una modulazione UNB, “Ultra Narrow Band”, capace di trasmettere dati con un basso data rate in una banda molto stretta per evitare interferenze tra i diversi pacchetti. Il canale di trasmissione è half-duplex ma la capacità in downlink è quasi nulla mentre quella in uplink prevede data rate contenuti con payload applicativi di dimensioni ridotte (12 byte) [10].

Il piano di mercato di Sigfox segue una filosofia opposta rispetto alla LoRa Alliance: in questo caso infatti l’hardware può essere acquistato liberamente da qualsiasi venditore che decida di implementare il protocollo fisico di trasmissione; d’altra parte tutto il livello software della rete è proprietario, dunque per utilizzare una rete Sigfox è necessario pagare periodicamente il loro servizio [10].

2.2.2.3. Weightless

Weightless SIG, “Special Interest Group”, è un’organizzazione no profit che ha ideato tre diverse soluzioni: Weightless-W, la prima ad essere realizzata, Weightless-N e Weightless-P.

La prima ad essere realizzata è stata Weightless-W ma non ha avuto un grande successo. Il suo sistema di comunicazione prevede l’uso delle frequenze lasciate libere dai canali TV e non garantisce un risparmio energetico elevato [11].

La seconda, Weightless-N, è una tecnologia molto simile a Sigfox in quanto utilizza una modulazione UNB ma afferma di possedere un livello MAC migliore rispetto al concorrente poiché dotato di “advanced demodulation techniques”, un sistema capace di recuperare errori dovuti all’interferenza di con altre trasmissioni radio [11].

Weightless-P, la più recente, utilizza l’FDMA, “Frequency Division Multiple Access”, e il TDMA, “Time Division Multiple Access”, in una banda di 12.5kHz (più ampia di Sigfox ma minore di LoRa) [11].

Tutte le soluzioni di Weightless sono aperte sia a livello hardware, acquisto dei chip presso qualsiasi rivenditore, sia a livello software, dal momento che la piattaforma di gestione e acquisizione dei dati attraverso la rete può essere installata privatamente [11].

	Model	Frequency	Range	Data transfer rate	Packet size	Stage
Sigfox	Proprietary	868 / 902 MHz	Rural: 30-50 km Urban: 3-10 km	Upload: <300 bps Download: 8 byte per day	12 byte	Scale
LoRaWAN	Alliance	433 / 868 / 780 / 915 MHz	Rural: 15 km Urban: 2-5 km	Upload / download: 300 bps – 50 kbps	User-defined	Scale
Weightless-N	Alliance	<1 GHz	3 km	Upload / download: 100 bps	<20 byte	Introduction
Weightless-P	Alliance	<1 GHz	2 km	Upload / download: 200 bps – 100 kbps	>10 byte	Under development

Tabella 3. Confronto tra le prestazioni offerte dalle tecnologie LPWAN operanti in banda libera. Tratto da [10].

2.2.2.4. Regolamentazioni delle bande “unlicensed”

Le tecnologie proposte in questo paragrafo sfruttano bande per la trasmissione radio libere, ovvero non destinate dagli organi regolatori regionali ad utilizzi che richiedano una licenza. Tali frequenze possono variare tra le diverse regioni pertanto occorre considerare le direttive alle quali è sottoposta l’area di interesse per la realizzazione della rete. In Europa l’ETSI, “European Telecommunications Standards Institute”, e il CEPT, “European Conference of Postal and Telecommunications Administrations”, hanno definito quali frequenze possono essere utilizzate e quali parametri occorre rispettare.

A seconda della banda utilizzata è necessario rispettare un limite di potenza e di duty cycle. Quest’ultimo parametro si definisce come:

$$DC = \frac{T_{on}}{T_{on}+T_{off}} \quad (1)$$

Dove DC è il duty cycle, T_{on} è il tempo in cui si utilizza la banda per trasmettere e T_{off} è l’intervallo di tempo in cui il dispositivo non è attivo. Imporre vincoli stringenti su tale grandezza è molto importante poiché è l’unico modo per regolamentare le bande “unlicensed”. Infatti occorre impedire che i trasmettitori occupino troppo a lungo il mezzo comune, l’etere, rendendolo inutilizzabile a causa dell’insorgere di interferenze tra le diverse trasmissioni. Viceversa, l’effetto negativo di avere DC ridotti è la diminuzione del data rate complessivo delle comunicazioni.

Si riportano di seguito le bande di trasmissione stabilite da ETSI e i relativi duty cycle imposti per l’utilizzo delle frequenze impiegate più comunemente dalle tecnologie unlicensed.

Frequency Bands/frequencies	Applications
26,995 MHz, 27,045 MHz, 27,095 MHz, 27,145 MHz, 27,195 MHz, 34,995 MHz to 35,225 MHz, 40,665 MHz, 40,675 MHz, 40,685 MHz, 40,695 MHz	Model control
26,957 MHz to 27,283 MHz	Non-specific use
40,660 MHz to 40,700 MHz	Non-specific use
138,200 MHz to 138,450 MHz	Non-specific use
169,400 MHz to 169,475 MHz	Tracking, tracing and data acquisition and meter reading
169,475 MHz to 169,4875 MHz	Social alarms
169,5875 MHz to 169,6000 MHz	Social alarms
433,050 MHz to 434,790 MHz	Non-specific use
863,000 MHz to 870,000 MHz	Non-specific use
864,800 MHz to 865,000 MHz	Wireless audio applications
868,000 MHz to 868,600 MHz	Non-specific use
868,600 MHz to 868,700 MHz	Alarms
868,700 MHz to 869,200 MHz	Non-specific use
869,200 MHz to 869,250 MHz	Social alarms
869,250 MHz to 869,300 MHz	Alarms (0,1% duty cycle)
869,300 MHz to 869,400 MHz	Alarms (1% duty cycle)
869,400 MHz to 869,650 MHz	Non-specific use
869,650 MHz to 869,700 MHz	Alarms
869,700 MHz to 870,000 MHz	Non-specific use

Tabella 4. Bande di trasmissione e ricezione stabilite dall'ETSI in [12]. Evidenziate in rosso le bande utilizzate dalle tecnologie LPWANs in Europa. Tratto da [12].

Frequency Bands/frequencies	Applications	Maximum radiated power, e.r.p. / power spectral density	Channel spacing	Spectrum access and mitigation requirement (e.g. Duty cycle or LBT + AFA)
863,000 MHz to 870,000 MHz Modulation bandwidth up to 300 kHz is allowed	Non-specific use (Narrow/wideband modulation)	25 mW	≤100 kHz	0,1 % or LBT + AFA ¹
863,000 MHz to 870,000 MHz	Non-specific use (DSSS and other wideband modulation other than FHSS)	25 mW Power density is limited to -4,5 dBm/100 kHz	No requirement	0,1 % or LBT + AFA ¹
863,000 MHz to 870,000 MHz	Non-specific use (FHSS modulation)	25 mW	≤100 kHz	0,1 % or LBT ¹
868,000 MHz to 868,600 MHz	Non-specific use	25 mW	No requirement	1 % or LBT + AFA
868,700 MHz to 869,200 MHz	Non-specific use	25 mW	No requirement	0,1 % or LBT + AFA
869,400 MHz to 869,650 MHz	Non-specific use	500 mW	≤25 kHz The whole stated frequency band may be used as 1 wideband channel for high speed data transmission	10 % or LBT + AFA
869,700 MHz to 870,000 MHz	Non-specific use	25 mW	No requirement	1 % or LBT+AFA
869,700 MHz to 870,000 MHz	Non-specific use	5 mW	No requirement	No restriction

1 - Duty cycle may be increased to 1 % if the band is limited to 865 MHz to 868 MHz.

Tabella 5. Duty cycle imposti dall'ETSI alle bande di trasmissione maggiormente impiegate dalle LPWANs "unlicensed spectrum", tratto da [12].

2.2.3. Le possibilità a Torino

Durante le fasi iniziali del lavoro di tesi qui presentato è stato necessario scegliere la tecnologia di trasmissione da utilizzare per lo sviluppo dei casi d'uso individuati nel paragrafo 2.1.

Dopo aver valutato la disponibilità delle reti presenti sul territorio è emerso che le opzioni “licensed spectrum” non risultano ancora pienamente supportate dalle compagnie telefoniche italiane. Inoltre volendo perseguire l'obiettivo di fornire informazioni utili al progresso della città, l'ambiente “unlicensed” offre sicuramente maggior spazio alla sperimentazione, grazie alla natura “aperta” di parte delle tecnologie e alla presenza di aspetti non ancora indagati in ricerche e studi effettuati in merito.

Fra le diverse soluzioni libere da licenza la più adatta allo studio è risultata essere LoRaWAN. Essa infatti utilizza un protocollo di rete aperto che ne permette l'analisi, a differenza di Sigfox. Inoltre la community “The Things Network”, che si impegna a promuovere la diffusione di tale tecnologia fornendo strumenti utili alla messa a punto di applicazioni IoT, ha avviato una rete LoRaWAN a Torino. Per quanto riguarda Sigfox e Weightless invece nessun servizio simile è al momento disponibile nel capoluogo piemontese.

3. La rete LoRaWAN

LoRaWAN è uno stack di rete definito dalla LoRa Alliance, implementato a livello fisico dalla tecnologia di modulazione wireless LoRa e a livello medium access control dal protocollo LoRaMAC.

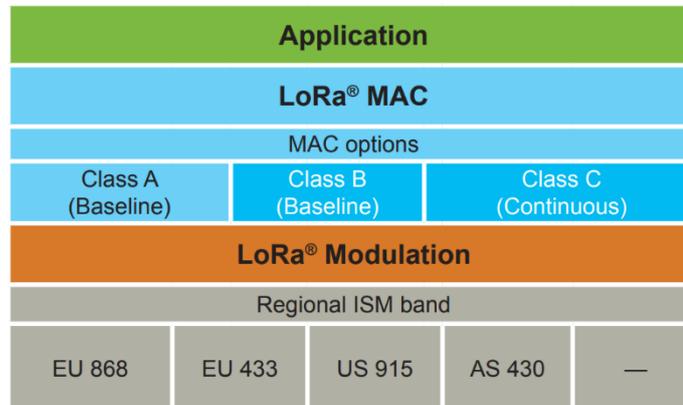


Figura 2. Stack tecnologico di LoRaWAN, tratto da [9].

Come mostrato in Figura 2, è presente una classificazione all'interno del livello MAC. Le classi "A", "B" e "C" identificano infatti capacità differenti di inviare pacchetti in uplink e riceverli in downlink. La prima prevede che i dispositivi inviino messaggi ai gateway quando lo ritengono necessario e che dopo ogni trasmissione aprano al più due brevi finestre temporali in cui possono ricevere dati dalla rete. Nella seconda invece, oltre alle due finestre di ricezione presenti nella classe A, sono programmate altre finestre in downlink ad intervalli fissi. Infine nella terza i dispositivi sono costantemente in ascolto ed interrompono la fase ricezione solo per l'invio di dati ai gateway. Come imposto da [13] è necessario che tutti i dispositivi implementino almeno le funzionalità della classe A. quest'ultima risulta essere spesso la più adatta agli scenari IoT e consente un maggior risparmio energetico rispetto alle altre due classi.

La struttura delle reti è generalmente organizzata in una "stars of stars topology" in cui più dispositivi comunicano con un "gateway", abbreviato GW e detto anche concentratore o "base station", e più gateway comunicano con un "network server", NS.

Compito dei gateway è fornire l'interfaccia di collegamento tra le trasmissioni LoRa e le tradizionali reti internet. Essi infatti sono dotati di antenne per ricevere messaggi in uplink dai dispositivi, indicati anche come "end nodes" o "end devices" (EDs), trasmessi su canali LoRa e inoltrarli al network server attraverso connessioni TCP/IP. Ovviamente è anche possibile percorrere il tragitto nella direzione opposta per consegnare messaggi in downlink agli EDs.

I network server sono la mente della rete. Essi sanno a quali applicazioni inoltrare i messaggi ricevuti dai dispositivi tramite i gateway, filtrano eventuali pacchetti ripetuti giunti dai concentratori che hanno captato la stessa trasmissione di un ED, determinano a quale gateway spedire un messaggio in downlink generato da una certa applicazione e destinato ad un particolare dispositivo, gestiscono gran parte degli aspetti di sicurezza di rete e molto altro.

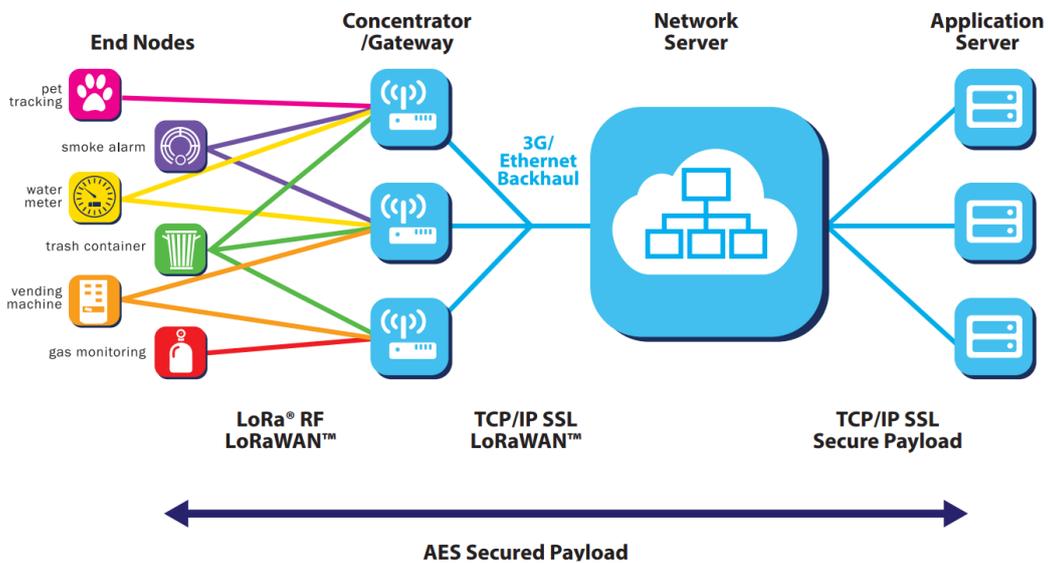


Figura 4. Topologia delle reti LoRaWAN, tratto da [9].

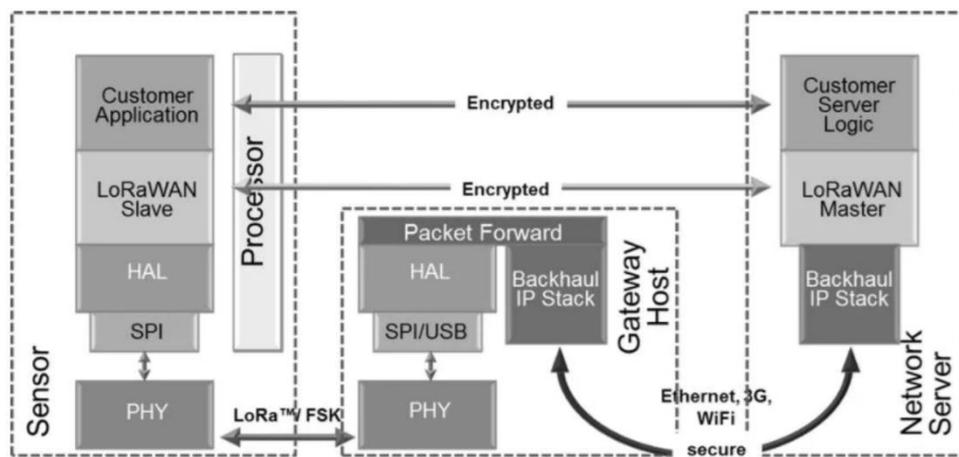


Figura 3. Struttura dei diversi componenti di una rete LoRaWAN e connessioni presenti tra di esse. Tratto da <https://www.lora-alliance.org/technology>.

3.1. Il layer PHY: LoRa

LoRa è una tecnologia di trasmissione wireless proprietaria di Semtech. Essa utilizza la tecnica CSS, “Chirp Spread Spectrum”, per ottenere una buona resilienza al rumore e raggiungere coperture di diversi chilometri sia in aree rurali che in zona urbane. Si presentano brevemente gli aspetti principali inerenti alla trasmissione LoRa che sono risultati utili al prosieguo del lavoro.

Come già accennato il segnale wireless trasmesso si basa sulla tecnica CSS. Esso è composto da simboli, contenenti le informazioni da trasportare, ciascuno ha le seguenti caratteristiche:

- Le frequenze che lo compongono sono tutte comprese tra una f_{\min} ed una f_{\max} che ne delimitano la banda, BW;
- Esso inizia all'istante $t=t_0$ con una frequenza $f=f_0$ appartenente all'intervallo $[f_{\min}; f_{\max}]$;
- Nel caso si tratti di un “upchirp”/“downchirp”, la frequenza f del segnale cresce/diminuisce linearmente nel tempo sino a quando non raggiunge f_{\max}/f_{\min} . A questo punto essa riparte dall'estremo opposto della banda, ovvero f_{\min}/f_{\max} , fino al raggiungimento della frequenza f_0 .

All'interno della banda del segnale esistono 2^{SF} livelli di frequenza dai quali ciascun simbolo può originarsi (SF è lo “Spreading Factor”). Pertanto per ciascun SF è possibile ottenere 2^{SF} simboli diversi e ogni simbolo codifica un numero di bit pari a SF.

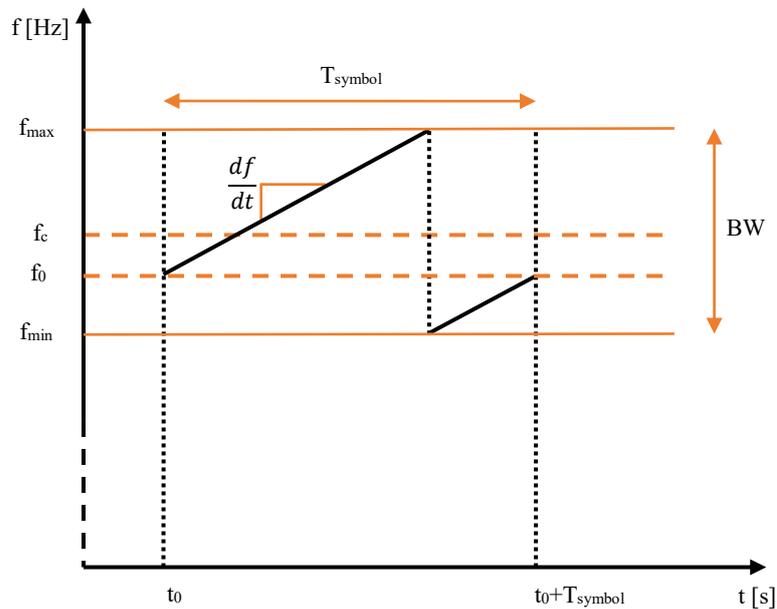


Figura 5. Rappresentazione di un "upchirp".

Si riportano alcune delle equazioni definite in [14]:

$$R_{chip} = BW \quad (2)$$

$$T_{chip} = R_{chip}^{-1} \quad (3)$$

Con R_{chip} frequenza di un singolo “chip” e T_{chip} durata di un singolo “chip”.

A partire dall’equazione (3) definiamo la durata di un simbolo:

$$T_{symbol} = 2^{SF} * T_{chip} = \frac{2^{SF}}{BW} \quad (4)$$

$$R_{symbol} = T_{symbol}^{-1} = \frac{BW}{2^{SF}} \quad (5)$$

Infatti ciascun simbolo è ottenuto moltiplicando i dati da trasmettere per uno “spreading code” composto da 2^{SF} “chips” [14].

Infine calcoliamo la rapidità di variazione delle frequenze di un simbolo come:

$$\frac{df}{dt} = \frac{BW}{T_{symbol}} = \frac{BW^2}{2^{SF}} \quad (6)$$

Come si può notare dall’equazione (6) è possibile modificare la “pendenza” di un simbolo agendo sulla BW e sullo SF. Una proprietà interessante che si ottiene da questo fenomeno è l’ortogonalità dei simboli. Due simboli si definiscono ortogonali quando configurati con opportune valori di BW e SF non interferiscono l’uno con l’altro pur sovrapponendosi temporalmente. Grazie a tale proprietà è possibile avere contemporaneamente trasmissioni multiple sullo stesso canale. In Figura 6. Spettrogramma di simboli non ortogonali. si possono osservare simboli non ortogonali tutti dotati della stessa rapidità di variazione di frequenza nel tempo. Invece in Figura 7. Spettrogramma di simboli ortogonali. i simboli rappresentati sono ortogonali tra di loro.

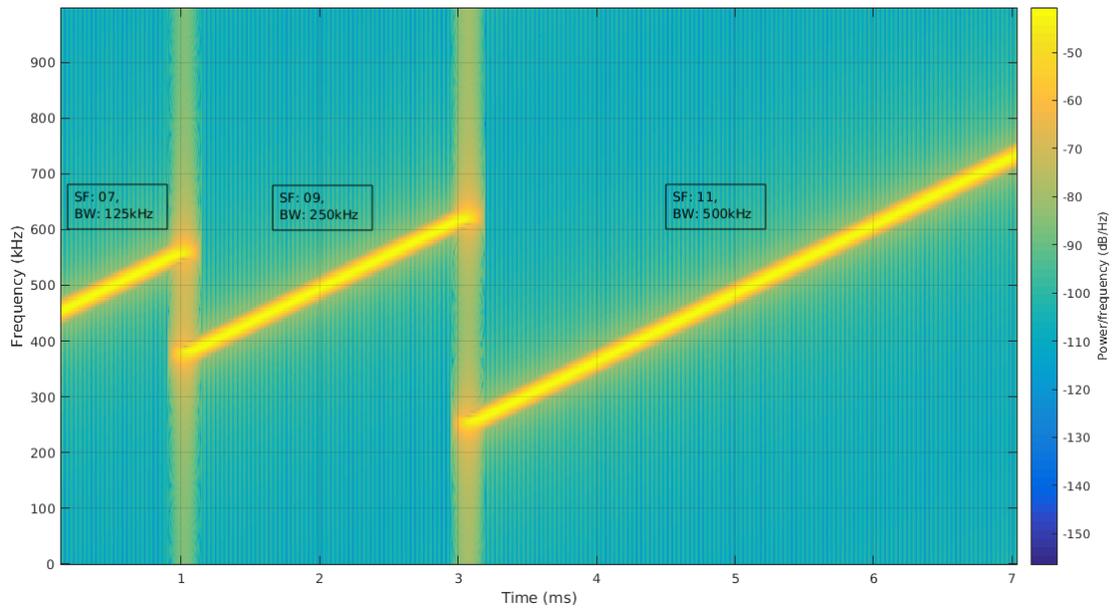


Figura 6. Spettrogramma di simboli non ortogonali.

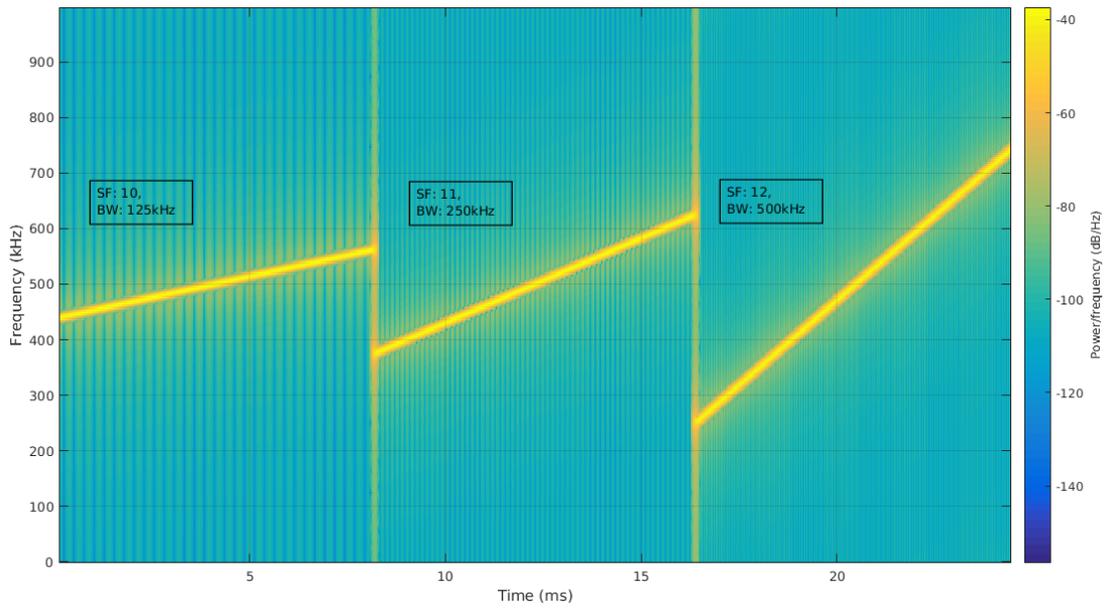


Figura 7. Spettrogramma di simboli ortogonali.

SF	07	08	09	10	11	12	07	08	09	10	11	12	07	08	09	10	11	12
BW	125	125	125	125	125	125	250	250	250	250	250	250	500	500	500	500	500	500
SF	BW																	
07	125	X						X									X	
08	125		X						X									X
09	125			X						X								
10	125				X						X							
11	125					X												
12	125						X											
07	250							X						X				
08	250								X						X			
09	250	X								X						X		
10	250		X								X						X	
11	250			X								X						X
12	250				X								X					
07	500												X					
08	500													X				
09	500						X								X			
10	500							X								X		
11	500	X							X								X	
12	500		X							X								X

Tabella 6. Configurazioni di Spreading Factor (SF) e Bandwidth [kHz] (BW) non ortogonali. Evidenziate in rosso le configurazioni disponibili per le reti LoRaWAN.

Un altro aspetto della modulazione LoRa è lo schema di correzione degli errori utilizzato per migliorare la robustezza dei segnali trasmessi. Esso garantisce prestazioni di decodifica migliori a spese di una certa ridondanza introdotta nei dati trasmessi.

Partendo da un bit rate di modulazione R_b pari a:

$$R_b = SF * R_{symbol} = SF * \frac{BW}{2^{SF}} \quad (7)$$

dove SF indica il numero di bit presenti in ogni simbolo, possiamo definire un bit rate nominale R_b' che tiene conto anche della ridondanza introdotta dalla tecnica di correzione:

$$R_b' = R_b * Rate\ Code = SF * \frac{BW}{2^{SF}} * \left[\frac{4}{4+CR} \right] \quad (8)$$

Nell'equazione (8) il *Rate Code*, sempre minore di uno, quantifica l'overhead in trasmissione dovuto allo schema di correzione e dipende dal parametro detto "code rate", abbreviato con CR [14].

Prima di proseguire mostrando le prestazioni dichiarate da Semtech sulla "sensitivity" offerta da LoRa nella decodifica dei messaggi, ovvero la minima intensità di un segnale correttamente rilevabile da un ricevitore, introduciamo il concetto di "link budget". Esso rappresenta la quantità di energia che una trasmissione radio può spendere per propagarsi nel mezzo di comunicazione e raggiungere un ricevitore con una potenza sufficiente alla decodifica delle informazioni trasportate.

La potenza di una trasmissione radio varia durante il tragitto tra il trasmettitore e il ricevitore. Si hanno solitamente perdite dovute alle connessioni tra i vari componenti dei dispositivi, guadagni di potenza forniti dalle antenne ma soprattutto si ha un grandissimo degrado delle prestazioni durante la propagazione del segnale in aria. In Figura 8 si propone una rappresentazione grafica della variazione di potenza di un segnale durante la sua propagazione.

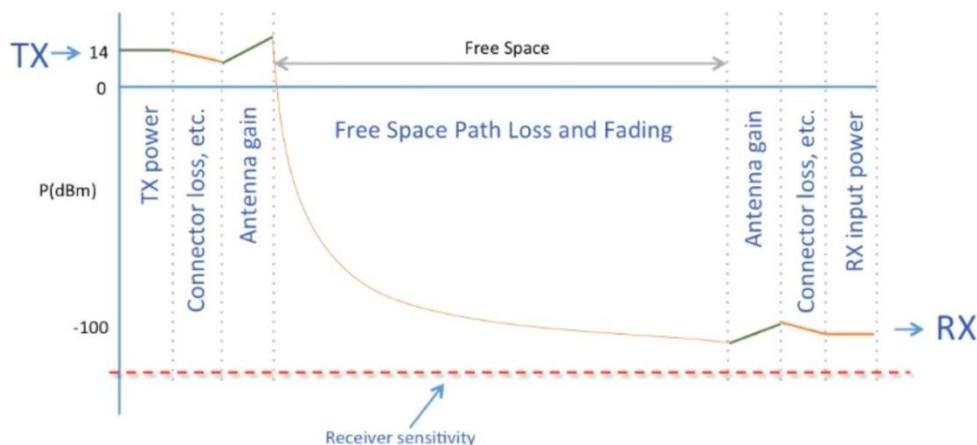


Figura 8. Potenza di un segnale nel tragitto dal trasmettitore al ricevitore.

Utilizzare modulazioni wireless che offrono una "sensitivity" molto bassa è vantaggioso perché permettono di coprire lunghe distanze in trasmissione.

Per quantificare le prestazioni dichiarate da Semtech [14] si sono calcolate le distanze teoriche massime di alcune trasmissioni LoRa supponendo l'utilizzo della massima potenza permessa ai dispositivi, ovvero 14dB [12], e il modello di propagazione denominato "Okumura-Hata", appositamente realizzato per lo studio di comunicazioni radio in scenari urbani [15]. In particolare la formula considerata per il calcolo del "path loss", ovvero la perdita di potenza del segnale durante la fase di propagazione in aria, è stata:

$$L_u = 69.55 + 26.16 * \log_{10} f - 13.82 * \log_{10} h_B - C_H + (44.9 - 6.55 * \log_{10} h_B) * \log_{10} d \quad (9)$$

Dove per città grandi si ha:

$$C_H = \begin{cases} 8.29 * (\log_{10}(1.54 * h_M))^2 - 1.1 ; \text{se } 150 \leq f \leq 200 \\ 3.2 * (\log_{10}(11.75 * h_M))^2 - 4.97 ; \text{se } 200 < f \leq 1500 \end{cases} \quad (10)$$

Con:

- L_u = “path loss” nelle aree urbane. [dB]
- h_B = altezza dell’antenna del gateway. [m]
- h_M = altezza dell’antenna del end-device. [m]
- f = frequenza della trasmissione. [MHz]
- C_H = fattore di correzione dell’altezza dell’antenna. [dB]
- d = distanza tra il gateway e il gateway. [km]

I valori ottenuti nei calcoli al variare delle configurazioni di rete considerate sono riassunti in Tabella 7.

Spreading Factor	Bit Rate ¹ [kb/s]	Sensitivity ¹ [dBm]	Covered Distance ² [km]
12	0.293	-137	< 4.68
11	0.537	-134.5	< 3.99
10	0.976	-132	< 3.49
09	1.757	-129	< 2.88
08	3.125	-126	< 2.38
07	5.468	-123	< 1.96
06	9.375	-118	< 1.42

¹ – si considera una bandwidth di 125kHz.

² – nell’utilizzo dell’equazione (9) è stato considerato $h_B = 25m$, $h_M = 1.5m$ e $f = 868MHz$.

Tabella 7. Bit rate e sensitivity di diverse configurazioni di trasmissioni LoRa, tratto da [14], e relative distanze di copertura derivate con l’utilizzo dell’equazione (9).

3.2. Il layer MAC: LoRa MAC

Il secondo componente dello stack LoRaWAN è il protocollo LoRaMAC. Esso è stato realizzato cercando di sfruttare al massimo la tecnologia di trasporto per minimizzare i consumi degli end devices.

In questa sezione si presentano: le classi di dispositivi previste dalle specifiche di LoRaWAN, in particolare la classe “A”; i formati dei pacchetti trasmessi; come avviene l’installazione di un nuovo end node su una rete; quali meccanismi di sicurezza sono previsti; alcuni aspetti del protocollo strettamente legati al livello fisico, come l’ADR, “adaptive data rate”, e le dimensioni massime dei payload applicativi.

Le informazioni riportate nel prosieguo del paragrafo sono tratte da [13], testo al quale si rimanda per maggiori dettagli.

Un dispositivo di una rete LoRaWAN può appartenere ad una delle seguenti classi:

- Classe “A”, dispositivi bi-direzionali: questi end-devices prevedono trasmissioni in uplink seguite da due finestre temporali in cui possono ricevere messaggi in downlink; è il dispositivo a decidere

quando inviare dati verso i gateways. Questa classe risulta essere quella a maggior risparmio energetico poiché tra le diverse trasmissioni il chip radio dell'end node è disattivo e non raggiungibile. Sarà compito del network server inoltrare messaggi al nodo quando quest'ultimo aprirà le prossime finestre di ricezione in seguito all'invio di un nuovo pacchetto in uplink.

- Classe "B", dispositivi bi-direzionali con intervalli di ricezione programmati: oltre alle finestre di ricezione disponibili dopo l'invio di un pacchetto, questi end-devices sono capaci di ricevere messaggi dalla rete ad intervalli prestabiliti noti al network server. La presente classe offre una maggiore capacità di comunicazione rispetto alla precedente a discapito di consumi energetici maggiori.
- Classe "C", dispositivi bi-direzionali con massime capacità di ricezione: a differenza delle due classi precedenti, questi dispositivi si trovano costantemente in modalità di ricezione, interrotta saltuariamente quando necessitano di inviare dati. Per questa ragione la terza ed ultima classe è la più dispendiosa dal punto di vista energetico e pensata per quei dispositivi che hanno la disponibilità

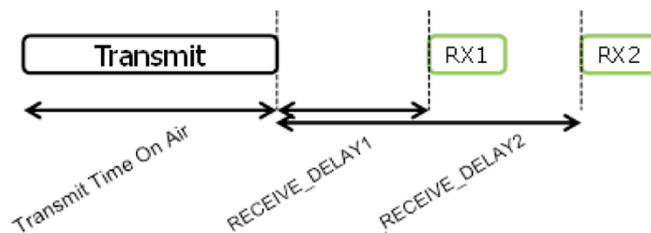


Figura 9. Dettaglio delle due finestre di ricezione disponibili per inviare messaggi in downlink successive alla trasmissione di un messaggio in uplink da parte di un dispositivo. Tratto da [13].

di un collegamento alla rete elettrica.

I messaggi in uplink, ovvero prodotti da un dispositivo e destinati ai gateway, utilizzano la modalità esplicita di LoRa [16] e sono costituiti in prima istanza da un header fisico (PHDR), dotato di "cyclic redundancy check" (PHDR_CRC), un payload (PHYPayload) e un controllo sull'integrità del payload stesso (CRC).

Uplink PHY:



Figura 10. Struttura di un messaggio in uplink a livello PHY. Tratto da [13].

I messaggi in downlink, cioè inviati dai gateway ad uno specifico end device, hanno una struttura analoga ai precedenti ma non possiedono il CRC calcolato sul payload. In tal modo si minimizza l'utilizzo della banda da parte delle base station.

Downlink PHY:



Figura 11. Struttura di un messaggio in downlink a livello PHY. Tratto da [13].

Scendendo maggiormente nel dettaglio, tutti i PHY payload contenuti sia nei messaggi in uplink che in quelli in downlink sono composti da MAC header (MHDR), formata da un ottetto¹, un MAC payload (MACPayload), con un numero massimo di ottetti variabile, e infine un message integrity code di 4 ottetti (MIC).

A sua volta il MACPayload contiene un frame header (FHDR) e due elementi opzionali che sono il campo FPort - un ottetto utile ad indicare la “porta” applicativa alla quale è destinato il messaggio - e il payload applicativo vero e proprio detto FRMPayload. Quest’ultimo è sempre cifrato con l’algoritmo generico specificato in [17] con una chiave AES, “Advanced Encryption System”, di lunghezza 128 bit. Si noti che il FRMPayload è cifrato prima del calcolo del MIC.

In particolare il FHDR trasporta informazioni relative al device che ha generato il messaggio (DevAddr), un ottetto (FCtrl) contenente bit di controllo del protocollo, due ottetti (FCnt) con il counter del messaggio e infine un massimo di 15 ottetti con eventuali opzioni.

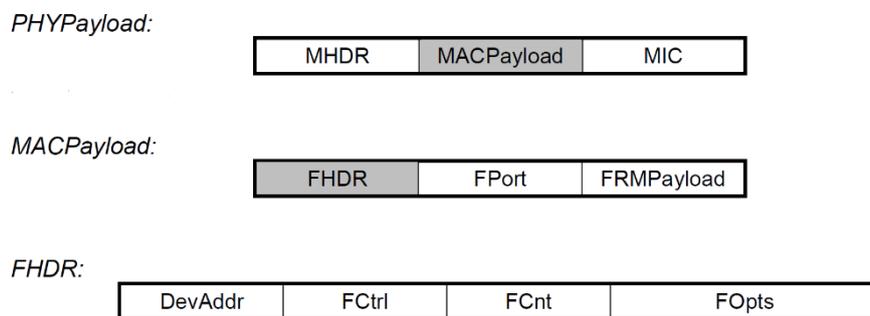


Figura 12. Strutture dei payload dei diversi livelli dello stack di rete. Tratto da [13].

LoRaWAN prevede sei tipologie di messaggi MAC differenti specificate all’interno del MHDR: join request, join accept, unconfirmed data up, unconfirmed data down, confirmed data up and confirmed data down. I pacchetti di join sono utilizzati nell’accesso di un nuovo dispositivo in una rete, gli altri trasportano dati applicativi. Questi ultimi si differenziano tra di loro sia per la sorgente, gli up sono inviati dai nodi mentre i down dai gateway, sia per l’utilizzo del meccanismo di acknowledgement, i confirmed richiedono conferma della ricezione e gli unconfirmed no.

Tutti i dispositivi che desiderano iniziare ad usufruire di una rete LoRaWAN per la trasmissione di dati applicativi devono per prima cosa eseguire la procedura di attivazione. Vi sono due modalità di attivazione: la OTAA, “Over-The-Air-Activation”, e la ABP, “Activation By Personalization”. In entrambi i casi al termine della procedura ciascun dispositivo sarà dotato di:

- DevAddr, Device Address: una stringa di 32 bits in cui i 7 bits più significativi [31-25] specificano la rete e i restanti 25 bits [24-0] identificano univocamente il dispositivo nella rete.
- AppEUI, Application Identifier: una stringa di 64 bits in formato little endian che identifica univocamente chi fornisce l’applicazione per gli end devices.
- NwkSKey, Network Session Key: è una chiave specifica per ciascun dispositivo ed è utilizzata sia dal network server sia dal nodo per calcolare e verificare il MIC di tutti i messaggi. Inoltre, nel caso

¹ Ottetto: insieme di 8 bits.

in cui il FRMPayload pacchetto contenga solo comandi MAC (FPort = 0), è utilizzata anche per la cifratura e decifratura di tale campo.

- AppSKey, Application Session Key: è una chiave specifica per ciascun dispositivo utilizzata per cifrare e decifrare il FRMPayload dei messaggi contenenti dati applicativi (FPort > 0). Inoltre può essere utilizzato per il calcolo e la verifica di un eventuale message integrity code inserito all'interno del campo FRMPayload.

Nella modalità OTAA i parametri sopra indicati sono concordati “al volo” tra il nuovo nodo e il NS grazie allo scambio dei diversi pacchetti di join definiti in precedenza. Nella ABP è cura dell'application provider fornire “out-of-band” la configurazione per i dispositivi.

Come accennato in precedenza il MACPayload ha una dimensione massima variabile che dipende dalla particolare configurazione di data rate utilizzato per la trasmissione. Il data rate è determinato fissando alcuni parametri introdotti nel paragrafo 3.1 tra cui lo spreading factor e la bandwidth. Le reti LoRaWAN definiscono coppie fisse di SF e BW riportate nella Tabella 8.

Data Rate	Configuration	
	Spreading Factor	Bandwidth
0	12	125 kHz
1	11	125 kHz
2	10	125 kHz
3	9	125 kHz
4	8	125 kHz
5	7	125 kHz
6	7	250 kHz

Tabella 8. Configurazioni data rate utilizzate da LoRaWAN. Tratto da [18].

Utilizzando l'equazione (4) si osserva come, al crescere dell'indice dei data rate, si ottenga una durata dei simboli minore e quindi un ToA, “Time on Air” ovvero tempo di trasmissione dell'intero messaggio, più breve con un conseguente bit rate più elevato. ToA brevi permettono di inviare messaggi più frequentemente rispetto a ToA più lunghi, infatti, a parità di duty cycle, il tempo di inattività sulla banda radio successivo all'invio di un messaggio, T_{off} , è minore tanto più è breve la durata della trasmissione.

$$T_{off} = \frac{(1-DC)}{DC} * TOA \quad (11)$$

Infine, come si evince dalla Tabella 7, avere simboli più rapidi comporta una minore capacità di trasmissione in termini di distanza coperta poiché il valore della sensitivity in ricezione aumenta. Si presenta dunque un problema di compromesso tra data rate e raggio di copertura di una comunicazione radio.

In questo scenario, i network server implementano un meccanismo di “fairness” attraverso la gestione automatica della scelta del data rate, detto ADR. Essi cercano di ottimizzare la rete in modo che tutti i nodi possano comunicare con almeno un gateway e allo stesso tempo occupare la banda per meno tempo possibile, limitando il rischio di interferenze. Il funzionamento si basa sull’analisi degli RSSI, “received signal strength indication”, dei diversi pacchetti inviati da un dispositivo. Nel caso in cui ci sia margine per aumentare il data rate e sfruttare sensitivity maggiori senza inficiare la corretta ricezione dei pacchetti, il NS invia i comandi MAC utili a richiedere al nodo di aumentare la rapidità di trasmissione. Purtroppo questa tecnica di ottimizzazione è disponibile solamente per i nodi fissi, ovvero che non modificano la loro posizione nel tempo: infatti uno spostamento potrebbe comportare una variazione repentina dell’ambiente e pertanto il data rate precedentemente accordato con la rete potrebbe non essere più sufficiente alla ricezione di un nuovo messaggio. Fortunatamente nei casi d’uso IoT si hanno spesso oggetti immobili per la maggior parte del tempo, ad esempio i cassonetti della spazzatura, lampioni, semafori, etc., che quindi possono usufruire di questo meccanismo.

Si riportano nella Tabella 9 le dimensioni massime, espresse in byte, permesse per il MACPayload e conseguentemente per il FRMPayload in funzione del data rate.

Data Rate	MACPayload max size [byte]	FRMPayload max size [byte]
0	59	51
1	59	51
2	59	51
3	123	115
4	230	222
5	230	222
6	230	222

Tabella 9. Dimensioni massime per il MACPayload e il FRMPayload in funzione del data rate. Tratto da [18].

Si elencano infine i canali obbligatori che ciascuna rete LoRaWAN deve supportare nel rispetto delle specifiche [18].

Bandwidth [kHz]	Channel Frequency [MHz]	Data Rate	Duty Cycle
125	868.10	0 to 5	<1%
125	868.30	0 to 5	<1%
125	868.50	0 to 5	<1%

Tabella 10. Canali obbligatori per le reti LoRaWAN. Tratto da [18].

3.3. Le reti LoRaWAN attive

LoRaWAN si sta diffondendo in tutto il mondo con la creazione di infrastrutture LoRa gestite da operatori delle telecomunicazioni e non solo.

Negli Stati Uniti, Senet e Comcast sono due tra le compagnie che stanno investendo maggiormente su questa tecnologia. Anche in Europa è presente un interesse a riguardo che ha coinvolto tra le altre KPN, compagnia olandese che ha annunciato reti LoRaWAN con copertura nazionale, Proximus, azienda belga che vanta una rete molto diffusa sul territorio, Orange, che in Francia ha iniziato a fornire collegamenti in diversi centri urbani, Unidata e Telmar, due realtà locali italiane con diffusione ancora limitata a poche città.

A parte troviamo la community “The Things Network” che offre liberamente la possibilità di collegare gateway privati alla loro piattaforme di gestione della rete in modo da permettere una rapida diffusione della connettività senza dover pagare bollette ad un particolare operatore per usufruirne. Ad Amsterdam è presente la loro più grande rete con una copertura quasi totale della città. The Things Network è presente anche a Torino con una rete piuttosto piccola ma che ha permesso la realizzazione dei test eseguiti nell’ambito del presente lavoro. Si elencano i canali disponibili nelle reti LoRaWAN di “The Things Network”.

Channel Frequency [MHz]	Data Rate	Duty Cycle
868.10	0 to 5	<1%
868.30	0 to 6	<1%
868.50	0 to 5	<1%
867.10	0 to 5	<1%
867.30	0 to 5	<1%
867.50	0 to 5	<1%
867.70	0 to 5	<1%
867.90	0 to 5	<1%

Tabella 11. Canali in uplink attualmente disponibili nelle reti LoRaWAN di The Things Network. Tratto da <https://www.thethingsnetwork.org/wiki/LoRaWAN/Frequencies/Frequency-Plans> (visitato 10/11/2017).

4. Scalabilità di LoRaWAN

Le applicazioni IoT che si stanno sviluppando in questi anni sono molteplici e la possibilità delle LPWANs di coprire una vasta area con una singola base station introduce un problema cruciale per lo sviluppo di queste reti: la scalabilità.

Nell'ambito del presente lavoro definiamo la scalabilità come la capacità di servire un numero crescente di nodi con un singolo gateway mantenendo un'affidabilità delle comunicazioni al di sopra di una certa soglia. Per misurare l'affidabilità ci serviamo di un parametro detto DER, "Data Extraction Rate", espresso come il rapporto tra il numero di pacchetti correttamente ricevuti e decodificati da un gateway e il numero totale di messaggi inviati dai dispositivi.

Considerando solamente i casi d'uso indicati nel paragrafo 2.1 per l'ambito urbano e applicando delle stime di densità dei dispositivi delle singole applicazioni per chilometro quadrato, possiamo ottenere a Torino un numero di end node per singola cella² anche superiore alle 10000 unità. In una prospettiva di creazione di una rete LoRaWAN duratura negli anni è dunque fondamentale capire le effettive prestazioni offerte da tale soluzione e verificare che esse rispettino la richiesta di connettività generata sul territorio.

In letteratura sono presenti diversi articoli che analizzano la scalabilità raggiungibile con LoRaWAN. Alcuni, più pessimisti, paragonano le prestazioni offerte a quelle di una rete ALOHA pura [19, 20], altri invece, utilizzando un modello che tiene maggiormente in considerazione le caratteristiche della tecnologia LoRa, presentano risultati più confortanti [21]. In tutti gli studi osservati però si è riscontrata una mancanza a nostro avviso grave: le simulazioni eseguite considerano un singolo profilo applicativo per test e pertanto si discostano significativamente da uno scenario futuro plausibile in cui una vasta gamma di applicazioni differenti condivide la stessa rete. Per spiegare meglio questo aspetto, una simulazione necessita di un'entità che esegua il lavoro dei gateway e di un altro elemento che fornisca le funzionalità degli end node. Tralasciando momentaneamente la prima, notiamo che ciascun dispositivo avrà una configurazione più strettamente legata alla rete - scelta dello spreading factor, del canale di trasmissione, del data rate - e un'altra relativa alla natura dei messaggi da inviare. Quest'ultima dipende dalla specifica applicazione IoT a cui ciascun nodo è destinato e ne determina la dimensione del payload e l'intervallo temporale tra un messaggio e l'altro. Negli articoli studiati si considera un solo set di parametri applicativi per simulazione fornendo risultati legati ad uno scenario piatto, omogeneo.

L'obiettivo dello studio svolto in questa tesi è di effettuare un ulteriore passo in avanti nella stima della scalabilità di una rete LoRaWAN sfruttando un modello simulativo capace di considerare contemporaneamente differenti profili applicativi.

² Cella: superficie di una rete wireless coperta da una singola base station. Generalmente nelle reti telefoniche essa ha forma esagonale e il gateway è posizionato al centro. Per la stima dei dispositivi appartenenti ad una singola cella fornita nel testo si è considerato un esagono inscritto in una circonferenza di raggio 1.5 km [26].

4.1. Il modello

La scalabilità di una rete LoRaWAN è limitata da due aspetti: le interferenze che possono avvenire durante le trasmissioni dei pacchetti e la capacità di ricevere e decodificare correttamente tutti i messaggi giunti ai gateway.

Si differenziano le interferenze in due categorie: interne, ovvero prodotte reciprocamente tra i diversi pacchetti LoRa, ed esterne, cioè causate dall'utilizzo della banda di trasmissione da parte di altre comunicazioni e segnali di vario genere. Nel lavoro qui presentato si considera solamente la prima tipologia, infatti per indagare la seconda sarebbero necessarie maggiori conoscenze nell'ambito delle telecomunicazioni e occorrerebbe una strumentazione più sofisticata di quella utilizzata nell'ambito della presente tesi. È comunque possibile trovare informazioni riguardo l'interferenza esterna in [22].

In prima battuta si può affermare che due pacchetti modulati secondo lo schema LoRa interferiscono tra di loro se le rispettive trasmissioni risultano almeno in parte sovrapposte temporalmente.

Riferendoci ai concetti esposti nel paragrafo 3.1 notiamo come grazie all'ortogonalità sia possibile avere più trasmissioni contemporanee senza che si produca interferenza distruttiva tra le comunicazioni. Poiché tutti i data rate impiegati da LoRaWAN (Tabella 8) sono ortogonali non vi è interferenza tra loro. Inoltre i messaggi che utilizzano canali diversi all'interno della banda (Tabella 10) non interferiscono vicendevolmente.

Si ottiene dunque un nuovo criterio per valutare l'interferenza tra due pacchetti ovvero che essi si disturbano solo se le rispettive trasmissioni risultano almeno in parte sovrapposte nel tempo, si trovano sullo stesso canale e possiedono la stessa configurazione di data rate.

Stabilito quando si produce dell'interferenza studiamo cosa accade nella collisione tra due pacchetti. LoRa può essere considerata una tecnologia con modulazione in frequenza e pertanto gode del capture effect [19]. Tale fenomeno si verifica quando due segnali, che interferiscono tra di loro, sono contemporaneamente presenti al ricevitore ma, a causa della diversità della potenza con la quale giungono a destinazione, il più intenso “sopprime” l'altro e può quindi essere decodificato a discapito del meno forte. In letteratura sono presenti esperimenti condotti per ottenere una quantificazione del fenomeno nelle reti LoRaWAN. In particolare in [19] i ricercatori allestiscono un test in cui sono trasmessi due pacchetti, X e Y: questi presentano dimensione identica, stessa configurazione di data rate e potenze in ricezione differenti, con $P_X > P_Y$. Nelle diverse prove T_Y , istante di inizio trasmissione del pacchetto Y, è costante mentre T_X è ottenuto con valori di offset, rispetto a T_Y , contenuti nell'intervallo $[-ToA; +ToA]$. Il passo di variazione dello sfasamento è pari ad $1 T_{symbol}$.

Dalla Figura 13 si può osservare come la trasmissione X sia decodificata correttamente quando arriva al ricevitore tra un ToA prima e $3 T_{symbol}$ dopo T_y sopprimendo completamente la trasmissione Y. Invece con un offset maggiore di $3 T_{symbol}$ e minore di ToA nessuno dei due pacchetti è ricevuto. Questo andamento si può spiegare con il funzionamento del chip radio utilizzato in ricezione. Esso infatti utilizza gli ultimi 5 simboli del preambolo, che nell'esperimento è configurato con una lunghezza di 8 simboli, per sincronizzarsi e agganciarsi ad una trasmissione, pertanto dopo $3 T_{symbol}$ di un primo segnale

esso si blocca su tale trasmissione e non è più in grado di allinearsi con un nuovo segnale, anche se quest'ultimo risulta più potente del primo [19].

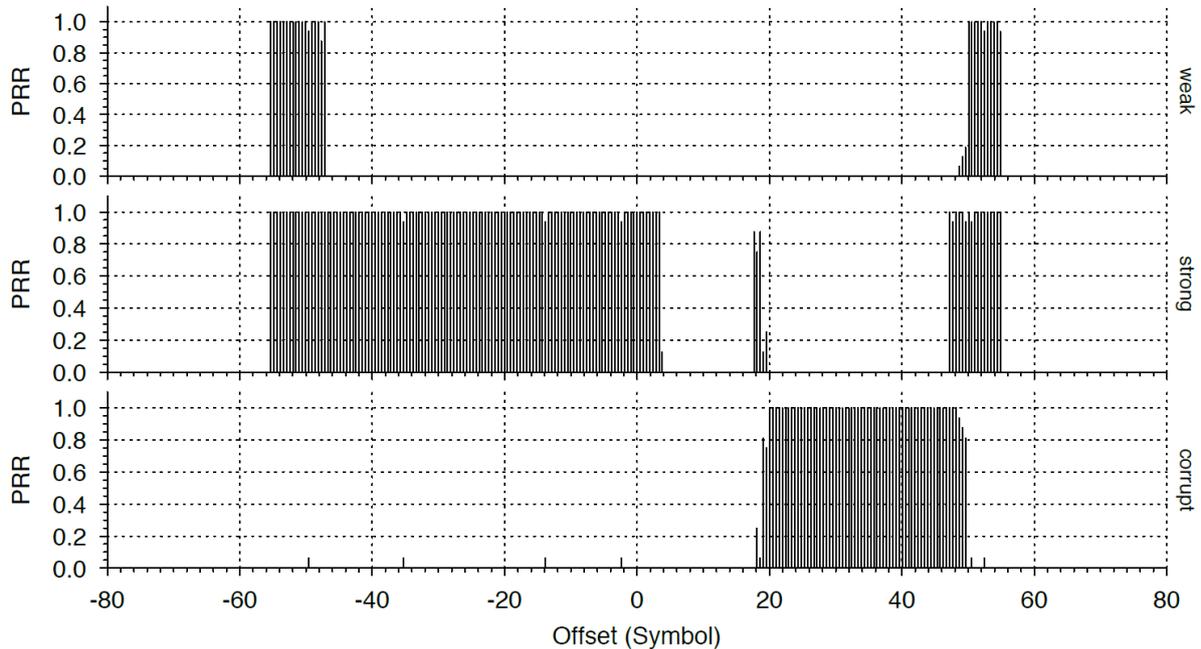


Figura 13. Risultati dell'esperimento per la verifica del capture effect. Lungo l'asse x abbiamo l'offset della trasmissione più potente rispetto alla più debole; sull'asse y il PRR, "packet reception rate". Tratto da [19].

Nell'altro studio [21] si svolgono test in due scenari differenti. In un caso la trasmissione interferita, Tx_1 , e quella interferente, Tx_2 , hanno potenze diverse - situazione simile all'esperimento del precedente articolo - nel secondo invece Tx_1 e Tx_2 hanno stesso RSSI. Anche in questa serie di esperimenti si studia il capture effect al variare dello sfasamento tra le trasmissioni. In particolare l'istante iniziale di Tx_2 è traslato rispetto a quello di Tx_1 di intervalli regolari partendo da 0 ms fino al ToA del pacchetto, identico per entrambe le comunicazioni. I risultati dei test del primo scenario si trovano in Tabella 14 e Tabella 13 mentre quelli del secondo in Tabella 16 e Tabella 15.

Si noti che per entrambe le situazioni i ricercatori hanno usato due configurazioni in modo da evidenziare eventuali fenomeni legati alla variazione dello spreading factor. I dettagli delle configurazioni sono riportati in Tabella 12.

Parametro	Configurazione #01	Configurazione #02
Spreading Factor	12	7
Bandwidth	125 kHz	125 kHz
Code Rate	4	4
Explicit Header	Sì	Sì
Channel	868.3 MHz	868.3 MHz
Payload CRC	Sì	Sì
Lunghezza sezione programmabile del preambolo	8	14
Ottimizzazione per basso data rate	Sì	No
Dimensione payload	17 byte	17 byte
Bit rate equivalente	183.11 bps	3417.97 bps
Time on Air	1712.13 ms	76.03 ms
Durata preambolo	401.41 ms	18.69 ms
Durata simbolo	32.77 ms	1.02 ms

Tabella 12. Configurazioni utilizzate nell'esecuzione dei test dello studio [21]. Tratto da [21].

Shift in ms	100		200		300		400		500		600	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	100	100	100	100	100	100	100	0	62	50	2	100
Packet received with BAD_CRC (%)	NA	NA	NA	NA	NA	NA	NA	0	38	0	98	NA
Average of SNR (dB)	NA	NA	NA	NA	NA	NA	NA	10.3	1.01	9.7	1.3	NA
Shift in ms	700		800		900		1000		1100		1200	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	0	100	0	100	0	100	0	100	0	100	0	100
Packet received with BAD_CRC (%)	100	NA	100	NA	100	NA	100	NA	100	NA	100	NA
Average of SNR (dB)	1.3	NA	1.1	NA	1.1	NA	1.7	NA	1.6	NA	1.7	NA
Shift in ms	1300		1400		1500		1600					
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2		
Packet lost due to collisions (%)	0	100	0	100	0	48	0	0	0	0		
Packet received with BAD_CRC (%)	100	NA	100	NA	99	2	73	0	1	0		
Average of SNR (dB)	5.1	NA	6	10.4	6.8	10.4	8.2	11.3				

Tabella 14. Risultati dei test in cui Tx1 ha RSSI maggiore di 12 dB rispetto a Tx2 ed entrambe utilizzano la configurazione 1 della Tabella 11. Tratto da [20].

Shift in ms	5		10		20		30	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	100	100	100	100	72	41	0	100
Packet received with BAD_CRC (%)	NA	NA	NA	NA	28	0	100	NA
Average of SNR (dB)	NA	NA	NA	NA	1.8	9.1	1.91	NA
Shift in ms	40		50		60		70	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	0	100	0	100	0	100	0	0
Packet received with BAD_CRC (%)	100	NA	100	NA	100	NA	0	0
Average of SNR (dB)	2.1	NA	2.08	NA	6.19	NA	8.99	9.1

Tabella 13. Risultati dei test in cui Tx1 ha RSSI maggiore di 12 dB rispetto a Tx2 ed entrambe utilizzano la configurazione 2 della Tabella 11. Tratto da [20].

Shift in ms	100		200		300		400		500		600	
Transmitters	Tx1	Tx2										
Packet lost due to collisions (%)	2.2	100	4	100	3.2	100	3	100	4.4	100	2	100
Packet received with BAD_CRC (%)	25	NA	9	NA	4.4	NA	1	NA	7	NA	4.3	NA
Average of SNR (dB)	2	NA	2	NA	2.1	NA	2.2	NA	1.7	NA	1.8	NA

Shift in ms	700		800		900		1000		1100		1200	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	3.4	100	2.4	100	0.8	100	0.8	100	2	100	2	100
Packet received with BAD_CRC (%)	3.5	NA	8.4	NA	4.3	NA	2.5	NA	5.4	NA	3.2	NA
Average of SNR (dB)	2.1	NA	2.4	NA	2.4	NA	2.5	NA	3	NA	3.4	NA

Shift in ms	1300		1400		1500		1600	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	1	100	2.2	84	1	24	2	5.2
Packet received with BAD_CRC (%)	2.1	NA	0	0	2.1	1	4.5	0
Average of SNR (dB)	4	NA	5	8.4	6.5	8.5	8.6	8.9

Tabella 16. Risultati dei test in cui Tx1 e Tx2 hanno stesso RSSI e utilizzano la configurazione 1 della Tabella 11. Tratto da [20].

Shift in ms	5		10		20		30	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	2.3	100	2.1	100	2.8	100	2.5	100
Packet received with BAD_CRC (%)	18	NA	7	NA	4	NA	5.2	NA
Average of SNR (dB)	3	NA	3.1	NA	3.2	NA	3.1	NA

Shift in ms	40		50		60		70	
Transmitters	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2	Tx1	Tx2
Packet lost due to collisions (%)	2.1	100	2	100	0	100	0	0
Packet received with BAD_CRC (%)	3.1	NA	2.1	NA	2	NA	0	0
Average of SNR (dB)	4	NA	5	NA	8	NA	9.1	9.1

Tabella 15. Risultati dei test in cui Tx1 e Tx2 hanno stesso RSSI e utilizzano la configurazione 2 della Tabella 11. Tratto da [20].

Confrontando i risultati degli studi dei due articoli si osserva che risultano essere in accordo. Entrambi infatti dimostrano la possibilità di ricevere correttamente almeno uno dei due messaggi in caso di interferenza dovuta all'utilizzo contemporaneo dello stesso canale da parte di due trasmissioni con la stessa configurazione di data rate.

Riassumendo dunque quanto ottenuto dai diversi esperimenti si può stabilire il seguente modello di interferenza:

- Due pacchetti con canali di trasmissione differenti e data rate differenti non interferiscono mai tra di loro.
- Due pacchetti con stesso canale di trasmissione ma con data rate differenti non interferiscono mai tra di loro.
- Due pacchetti con stesso canale di trasmissione e stesso data rate interferiscono solo se sono almeno in parte sovrapposti temporalmente. In tal caso se il messaggio giunto per primo al ricevitore ha RSSI maggiore dell'altro esso sarà decodificato correttamente mentre il secondo verrà perso. Se invece le due trasmissioni hanno RSSI simili si perderanno entrambe.

Si noti come il modello appena presentato sia più conservativo rispetto ai risultati degli esperimenti infatti considera sempre persi i messaggi sovrapposti nel tempo e giunti al ricevitore con potenze simili.

Conclusa la trattazione delle interferenze interne si procede all'analisi delle prestazioni offerte dai gateway in ricezione.

Le base station presenti ad oggi sul mercato sfruttano il chip SX1301, prodotto da Semtech, per ricevere le trasmissioni LoRa. Esso è in grado di supportare una rete con topologia a stella costituita da 10'000 end node [23]. Tralasciando gli aspetti pubblicitari del datasheet del chip [23] e concentrandoci sulle caratteristiche di ricezione osserviamo che esso può decodificare contemporaneamente fino ad 8 trasmissioni con qualsiasi data rate e su qualsiasi canale. Nel nostro modello si tiene conto di questo aspetto considerando 8 path di decodifica indipendenti. Nel caso giunga un pacchetto quando tutti i path sono già occupati quest'ultimo messaggio andrà perso.

A questo punto occorre determinare quanti dispositivi sono presenti in una cella, quanti messaggi inviano in un certo intervallo di tempo e quanto sono grandi i payload. Per farlo consideriamo un insieme di casi d'uso e per ciascuno di essi ne delineiamo un profilo applicativo.

Un profilo applicativo definisce: la densità al chilometro quadrato di end-device utilizzati per la specifica applicazione, la dimensione media dei payload inviati nei pacchetti e l'intervallo di trasmissione dei messaggi per ciascun dispositivo.

Si sono dunque forniti tutti gli elementi utili a descrivere la logica del modello utilizzato nello studio della scalabilità sostenuto in questa tesi. Si procede a presentare il simulatore sviluppato *ad hoc* per implementare tale modello.

4.2. Il simulatore

Il simulatore sviluppato fornisce un'indicazione sul DER offerto da una cella LoRaWAN popolata da un determinato numero di dispositivi impiegati da un particolare insieme di applicazioni.

4.2.1. Il progetto

Per svolgere l'analisi del DER di una cella LoRaWAN sono stati identificati i seguenti passaggi salienti:

- Analisi dei profili applicativi da simulare;
- Distribuzione dei dispositivi nella cella;
- Pianificazione ed invio dei pacchetti di ciascun dispositivo;
- Raccolta ed analisi dei pacchetti inviati da tutti gli end node per determinare eventuali collisioni dovute ad interferenze in trasmissione;
- Decodifica dei pacchetti integri giunti al gateway.

Si presentano quindi i dettagli di ciascuna fase nei paragrafi seguenti.

4.2.1.1. Analisi dei profili applicativi

Come accennato all'inizio del capitolo 4, negli studi trovati in letteratura riguardo la scalabilità delle reti LoRaWAN si presentano risultati ottenuti considerando tipologie di pacchetti distanti dagli scenari reali. Ad esempio in [24] si considerano molti casi d'uso ma sempre singolarmente; inoltre si fornisce come stima del numero massimo di end node per cella il valore ottimale per il modello Aloha puro. In [19] tutte le simulazioni svolte con il software sviluppato dai ricercatori, LoRaSim³, utilizzano una sola dimensione di pacchetto per ciascun dispositivo. Situazioni analoghe si riscontrano anche in [20] e [21].

Per superare questo limite, il simulatore presentato in questo testo permette di gestire più profili applicativi. Ciascuno di essi specifica una densità superficiale di end node, una dimensione del payload e un intervallo di trasmissione. Quest'ultimo può essere espresso sia direttamente con un singolo valore temporale, nel caso in cui l'applicazione invii periodicamente i propri messaggi, sia indirettamente fornendo il numero di messaggi da inviare durante un certo intervallo. Nel secondo caso il tempo trascorso tra le diverse trasmissioni non risulterà costante.

Analizzati tutti i profili applicativi presenti nella configurazione - contenente anche il numero totale di end node da generare, N , il raggio della cella, R , e il tempo della simulazione, T - il software procede nel determinare quanti dispositivi per ciascuna applicazione devono essere considerati, grazie alle diverse densità, e quanti pacchetti deve inviare ciascun dispositivo.

Al termine di questo passaggio si ottiene quindi una popolazione di end node variegata in termini di applicazione di appartenenza, conseguentemente associata a differenti tipologie di messaggi da inviare e diversi intervalli di trasmissione.

4.2.1.2. Distribuzione dei dispositivi

Il simulatore dispone tutti i dispositivi in modo casuale lungo il raggio R ; in funzione della distanza dal gateway, calcola l'RSSI da associare a ciascun end node: esso risulta costante per tutto l'arco della simulazione e lo si stabilisce con l'equazione (9). Fissata distanza e potenza, il software procede alla

³ LoRaSim: simulatore di rete LoRaWAN disponibile a <http://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>.

configurazione del canale e del data rate per ogni singolo dispositivo, scegliendoli casualmente tra quelli disponibili. Nel simulatore è possibile definire quali e quanti canali di trasmissione e data rate utilizzare.

Prima di procedere, si osservi che il concetto di cella impiegato in questo lavoro è di una superficie in cui tutte le configurazioni di data rate permettono di raggiungere la base station. L'obiettivo infatti è verificare la massima scalabilità in termini di end node serviti da un solo gateway. Risulta evidente che, per avere il massimo della capacità di trasmissione, convenga usufruire di tutti i data rate per massimizzare i pacchetti spediti in parallelo.

Pertanto il software procede all'attribuzione casuale del canale e del data rate per ciascun dispositivo solo dopo aver verificato che il raggio della cella sia minore della minima distanza coperta dalle diverse configurazioni di rete (Tabella 7).

Si sottolinea infine un aspetto critico delle simulazioni. Come affermato in precedenza, si considerano degli RSSI costanti per tutti i messaggi inviati da uno specifico dispositivo. Tali valori sono determinati con il modello di Okumura-Hata [15] e non per via sperimentale. A causa di questo procedimento non si considera che, in un'area - specialmente urbana - con raggio tale per cui tutti i data rate dovrebbero teoricamente fornire copertura, ci possano essere zone in cui l'RSSI assuma valori molto più bassi rispetto a quelli previsti. In uno scenario reale quindi si potrebbero avere perdite di pacchetti non considerate nelle simulazioni. Per valutare l'effettiva copertura raggiungibile in città sono stati effettuati dei test per le strade di Torino i cui risultati sono presentati nel capitolo 5.

4.2.1.3. Pianificazione ed invio dei pacchetti

Giunti a questo punto del flusso di esecuzione, si ha la lista di tutti i dispositivi con le relative configurazioni, si sa quanti messaggi deve inviare ciascun nodo e con quali RSSI giungono i messaggi al gateway. Il passo successivo è determinare gli istanti di inizio e fine trasmissione di ciascun messaggio inviato da ogni singolo nodo, per poter studiare l'interferenza tra le diverse comunicazioni.

Come già anticipato, occorre distinguere tra due modalità di invio dei messaggi ottenibili dai profili applicativi. In un caso si ha un intervallo di tempo fisso che intercorre tra due ricezioni, nell'altro invece un numero medio di messaggi inviati in un certo periodo. Nella prima categoria rientrano i casi d'uso che prevedono una comunicazione tra nodo e gateway periodica, ad esempio il monitoraggio del numero di passeggeri presenti su un pullman; nella seconda invece trovano spazio applicazioni in cui la comunicazione si basa su eventi, come nel caso dei sensori per i parcheggi. Questa differenza è considerata nel simulatore calcolando i tempi di trasmissione dei messaggi con due algoritmi differenti.

Per tutti i dispositivi di un'applicazione definita "periodica", l' i -esimo messaggio ha tempo di inizio trasmissione pari al tempo di fine trasmissione del messaggio $i-1$ più l'intervallo T_{off} previsto dallo specifico profilo applicativo. L'istante di fine trasmissione dell' i -esimo messaggio invece è pari al proprio tempo di inizio trasmissione più il ToA, calcolato con l'equazione (12) e i parametri presenti nel profilo applicativo. Infine, per ricreare una situazione verosimile in cui i messaggi di dispositivi differenti impiegati da una stessa applicazione non sono tutti sincronizzati tra di loro, il primo messaggio di ciascun nodo ha tempo di inizio trasmissione pari ad un valore casuale contenuto nell'intervallo $[0, T_{\text{off}}]$.

Per tutti i dispositivi di un'applicazione definita "ad evento", per prima cosa sono estratti k valori casuali nell'intervallo $[0, T]$, dove k è il numero di messaggi che ciascun dispositivo deve inviare e T è il tempo

di simulazione. In seguito si procede ordinatamente, dall'istante estratto più vicino allo 0 sino a quello più prossimo a T, a calcolare gli istanti di inizio e fine trasmissione dei diversi pacchetti. Per l'*i*-esimo messaggio, l'istante di inizio trasmissione è pari all'*i*-esimo istante estratto in precedenza mentre l'istante di fine trasmissione è pari al proprio istante di inizio trasmissione più il ToA, calcolato con l'equazione (12) e i parametri presenti nel profilo applicativo. Infine si verifica che tutti i messaggi di cui sono stati programmati sia l'istante di inizio trasmissione che di fine non si sovrappongano e rispettino il T_{off} previsto per la particolare applicazione. Nel caso in cui un pacchetto violi tali vincoli viene eliminato. Pertanto, concluso l'algoritmo, si ottengono al più *k* pacchetti pianificati per ciascun dispositivo.

Si noti che il T_{off} calcolato per ciascuna comunicazione corrisponde al massimo valore tra l'intervallo dei messaggi calcolato a partire dal profilo applicativo e il tempo di inattività imposto dalle regolamentazioni del duty cycle.

Il ToA è calcolato utilizzando le seguenti equazioni [16]:

$$ToA = T_{preamble} + T_{payload} \quad (12)$$

$$T_{preamble} = (n_{preamble} + 4.25) * T_{symbol} \quad (13)$$

$$T_{payload} = payloadSymNb * T_{symbol} \quad (14)$$

$$payloadSymNb = 8 + \max \left(\text{ceil} \left(\frac{8*PL - 4*SF + 28 + 16*CRC - 20*IH}{4*(SF - 2*DE)} \right) * (CR + 4), 0 \right) \quad (15)$$

Con:

- T_{preamble}: durata della prima sezione della trasmissione, detta preambolo;
- T_{payload}: durata della trasmissione del payload;
- n_{preamble}: numero di simboli contenuti nella porzione programmabile del preambolo;
- payloadSymNb: numero di simboli del PHYPayload;
- PL: numero di byte del PHYPayload;
- CRC: variabile booleana che assume valore 1 se è attivo il “cyclic redundancy check” su FRMPayload, 0 altrimenti;
- IH: variabile booleana pari a 1 se è attiva la modalità “implicit header”, 0 altrimenti;
- DE: variabile booleana pari a 1 se è attiva l'ottimizzazione del data rate, 0 altrimenti.
L'ottimizzazione è sempre attiva per SF11 e SF12.

4.2.1.4. Analisi delle interferenze

Dopo aver programmato i tempi di trasmissione di tutti i messaggi, si procede a verificare quali pacchetti hanno subito interferenza e quali invece sono giunti integri al gateway. Per farlo utilizziamo il modello proposto nel paragrafo 4.1 e riassunto con lo pseudo codice riportato in Figura 14. Pseudocodice del modello utilizzato per lo studio dell'interferenza tra due pacchetti trasmessi sulla rete LoRaWAN..

```

1 Packet interfered; //rappresenta la trasmissione interferita, contiene i parametri
2 Packet interferer; //rappresenta la trasmissione interferente, contiene i parametri
3 POWER_THRESHOLD = 10dB; //è la differenza minima che deve intercorrere tra gli RSSI
4 di due trasmissioni al fine di considerare una più potente dell'altra
5
6 if (interfered.getChannel() == interferer.getChannel())
7     AND
8     interfered.getDataRate() == interferer.getDataRate()){
9     //nel caso in cui i due pacchetti utilizzino stesso canale e stesso data rate
10    //le due trasmissioni potrebbero collidere
11
12    //estraiamo i parametri utili all'analisi dell'interferenza
13    interferedStartingTime = interfered.getStartingTime();
14    interferedEndTime = interfered.getEndTime();
15    interfererStartingTime = interferer.getStartingTime();
16    interfererEndTime = interferer.getEndTime();
17
18    //verifichiamo se i due pacchetti collidono
19    if(interfered.getRssi() >= interferer.getRssi() + POWER_THRESHOLD){
20        // nel caso in cui il pacchetto interferito sia più potente dell'interferente
21        if (interfererStartingTime <= interferedStartingTime
22            AND
23            interferedStartingTime <= interfererEndTime){
24            // nel caso in cui il pacchetto interferito abbia istante di inizio
25            // trasmissione
26            // compreso tra l'istante di inizio e fine trasmissione del pacchetto
27            // interferente
28            collision = true;
29        }
30    }
31    else{
32        //nel caso in cui il pacchetto interferito abbia potenza pari o inferiore a
33        //quella del pacchetto interferente
34        if (interfererStartingTime <= interferedStartingTime
35            AND
36            interferedStartingTime <= interfererEndTime){
37            // nel caso in cui il pacchetto interferito abbia istante di inizio
38            // trasmissione
39            // compreso tra l'istante di inizio e fine trasmissione del pacchetto
40            // interferente
41            collision = true;
42        }
43        else if(interfererStartingTime <= interferedEndTime
44            AND
45            interferedEndTime<= interfererEndTime){
46            // nel caso in cui il pacchetto interferito abbia istante di fine
47            // trasmissione
48            // compreso tra l'istante di inizio e fine trasmissione del pacchetto
49            // interferente
50            collision = true;
51        }
52        else if(interfererStartingTime >= interferedStartingTime AND
53            interferedEndTime >= interfererEndTime){
54            // nel caso in cui il pacchetto interferente abbia intervallo di
55            // trasmissione completamente contenuto
56            // tra l'istante di inizio e fine trasmissione del pacchetto interferito
57            collision = true;
58        }
59    }
60
61    //in tutti gli altri casi non si ha sovrapposizione temporale tra le due
62    //trasmissioni
63    collision = false;
64 }
65 else{
66     //i due pacchetti non collidono mai poichè utilizzano canali e/o data rate
67     //differenti
68     collision =false;
69 }
70
71 return collision;

```

Figura 14. Pseudocodice del modello utilizzato per lo studio dell'interferenza tra due pacchetti trasmessi sulla rete LoRaWAN.

Alla fine di questa fase otteniamo una lista con tutti e soli i pacchetti giunti correttamente alla base station.

4.2.1.5. Decodifica dei pacchetti

Ultimo passaggio da effettuare per stabilire il DER della cella simulata è calcolare quanti dei pacchetti giunti integri al gateway possano essere effettivamente decodificati. Per farlo sfruttiamo le caratteristiche del chip radio SX1301 descritte brevemente nel paragrafo 4.1. Da quanto riportato, una base station dispone di 8 cammini di decodifica paralleli, quindi possono essere decodificati al più 8 trasmissioni sovrapposte nel tempo. Questo funzionamento è simulato tenendo traccia di quanti degli 8 cammini sono occupati in ogni istante e scartando un eventuale messaggio giunto al gateway quando non vi è alcun cammino disponibile.

Terminata anche questa ultima fase, si può calcolare quanti messaggi sono stati ricevuti e decodificati correttamente a fronte dei messaggi inviati da tutti i dispositivi. Otteniamo quindi il DER come:

$$DER = \frac{N_{decodificati}}{N_{inviati}} \leq 1 \quad (16)$$

Con:

- $N_{decodificati}$: il numero totale di pacchetti decodificati dal gateway;
- $N_{inviati}$: il numero totale di pacchetti inviati da tutti i dispositivi.

4.2.2. L'implementazione

Seguendo lo schema progettuale proposto nei paragrafi precedenti, si è deciso di implementare il simulatore in un'applicazione realizzata con la release SE 1.8 del linguaggio di programmazione Java. Il codice sorgente è reperibile su GitHub⁴.

Il software realizzato presenta una struttura suddivisa in package e classi, riportata in Figura 15, basata sulle diverse fasi descritte nei precedenti paragrafi di progettazione.

⁴ Repository privata: <https://github.com/MicheledG/Tesi-LoRaWANSimulator>. Per richiederne l'accesso è necessario contattare l'autore all'indirizzo email: michele.digirolamo.to@gmail.com.

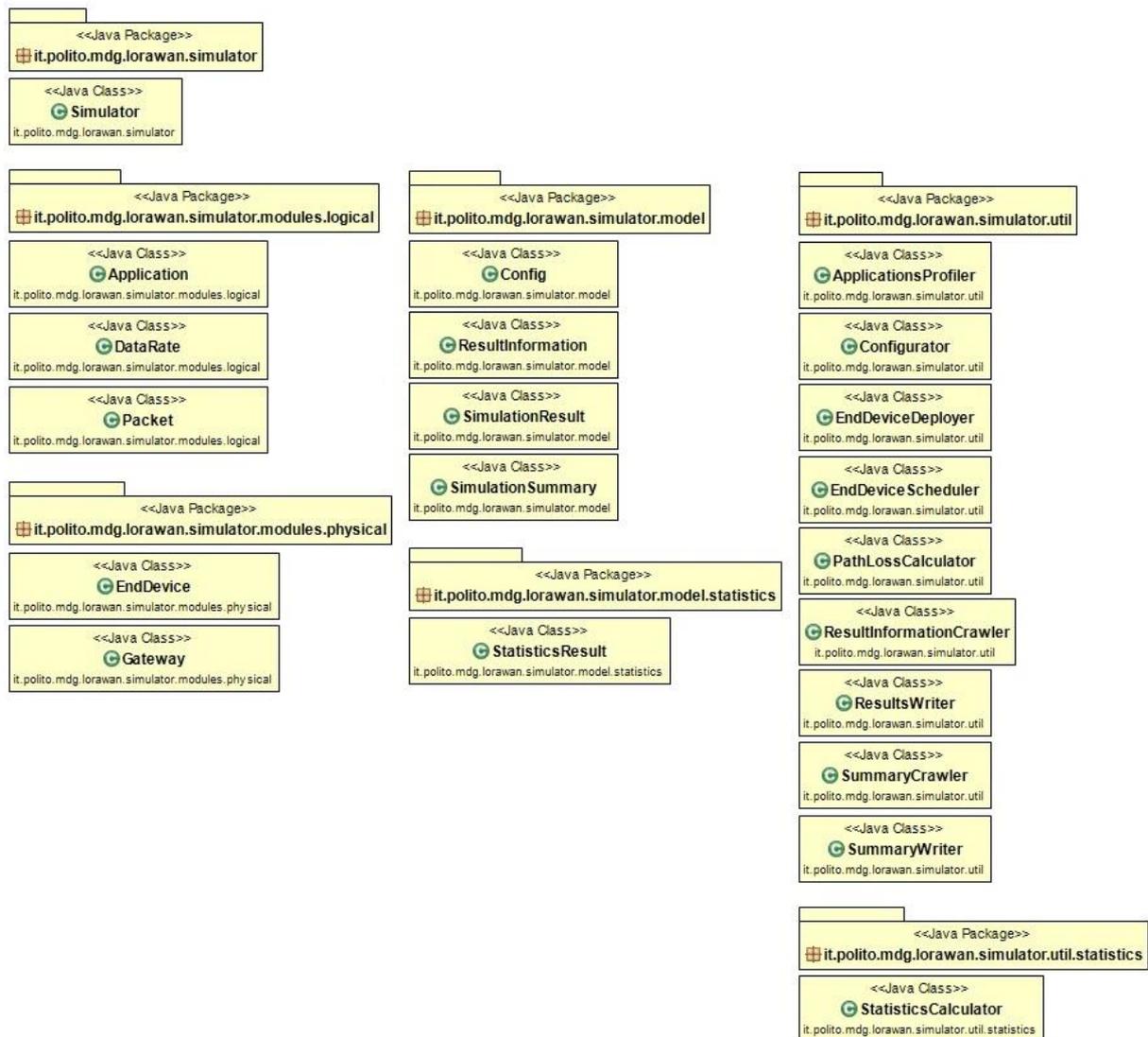


Figura 15. Elenco dei package e delle relative classi del simulatore implementato in Java.

La classe ‘Simulator’, appartenente al package principale, indica il punto di ingresso dell’applicazione. Essa contiene il main, in cui si svolgono tutti i passaggi delle simulazioni, e alcuni metodi di secondo ordine.

Le entità di una rete LoRaWAN sono rappresentate dalle classi appartenenti al package ‘modules’, in cui si differenziano quelle più propriamente hardware, ‘physical’, da quelle più astratte, ‘logical’. Nel primo gruppo troviamo le classi ‘EndDevice’ e ‘Gateway’. Un’istanza della classe ‘EndDevice’, dopo essere stata opportunamente configurata, è in grado di inviare dei messaggi, rappresentati dalla classe ‘Packet’, stabilendone l’istante di inizio e fine trasmissione. Quando tutti gli oggetti ‘EndDevice’ hanno inviato i loro messaggi entra in azione la classe ‘Gateway’ la quale offre due metodi che permettono rispettivamente di analizzare la lista di tutti i pacchetti inviati, per determinare le possibili collisioni avvenute in trasmissione, e decodificare i messaggi.

Per configurare l’ambiente di simulazione, orchestrare i moduli sia fisici sia logici ed estrarre i risultati sono state realizzate un insieme di classi, ospitate nel package ‘utility’, che offrono solo metodi statici. La prima ad entrare in scena è la classe ‘Configurator’ che estrae dal file di configurazione in formato JSON un oggetto della classe ‘Config’. Al suo interno si trovano tutti i dati relativi ai profili applicativi

(modellati con la classe 'Application'), i data rate da utilizzare (memorizzati in oggetti di tipo 'DataRate'), il raggio, il tempo di simulazione ed altre informazioni.

L'istanza della classe 'Config' ottenuta è in seguito processata dai metodi della classe 'ApplicationsProfiler' per determinare quanti oggetti della classe 'EndDevice' occorre istanziare per ciascun tipo di applicazione e quanti messaggi devono essere inviati, nell'arco del tempo di simulazione, da ciascun dispositivo.

A questo punto dell'esecuzione, la classe 'EndDeviceDeployer' procede all'inizializzazione e configurazione di tutti i dispositivi. Per farlo utilizza i dati prodotti in precedenza dal 'ApplicationsProfiler' e per ciascun oggetto di tipo 'EndDevice': fissa una distanza casuale dal gateway, mediante il metodo `Math.random()`, calcola il RSSI associato al dispositivo con l'aiuto della classe 'PathLossCalculator', e sceglie in modo aleatorio sia un canale di trasmissione sia un data rate.

Dopo che tutti gli end device sono pronti all'invio dei messaggi, la classe 'EndDeviceScheduler' richiede a ciascun nodo di inviare il numero di pacchetti stabilito dal 'ApplicationsProfiler' e, nel caso in cui un dispositivo appartenga ad un'applicazione "periodica", fornisce uno shift casuale per il tempo di inizio trasmissione del primo messaggio (cfr. paragrafo 4.2.1.3).

Quando tutti gli oggetti della classe 'Packet' sono stati istanziati dai diversi end device, sono pronti per essere processati dall'istanza di 'Gateway'.

Infine, dopo che il gateway ha svolto i suoi compiti di ricezione e decodifica, entrano in azione le utilities necessarie al calcolo dei risultati e alla produzione dei sommari salvati sia in formato JSON che in modalità grezza, utile per l'inserimento dei dati in un foglio di calcolo.

Si noti che la presenza del metodo `Math.random()` all'interno delle fasi di distribuzione e pianificazione (metodi delle classi 'EndDeviceDeployer' e 'EndDeviceScheduler') fornisce risultati differenti ad ogni simulazione. Per ottenere dei dati statistici accurati è possibile fornire al simulatore, tramite argomenti della linea di comando, alcuni parametri per impostare quante ripetizioni della stessa simulazione eseguire. I metodi della classe 'StatisticsCalculator' permettono di gestire i molteplici risultati ottenuti durante una singola esecuzione del programma calcolando medie aritmetiche e deviazioni standard.

4.3. Le prestazioni

In questa sezione descriviamo i risultati ottenuti sulla scalabilità di una cella LoRaWAN mediante l'utilizzo del simulatore presentato nei paragrafi precedenti. Inizialmente si riportano alcuni test di validazione del simulatore per verificarne la correttezza nell'analisi delle interferenze. Per farlo si configura il software in uno scenario di rete tipico del metodo di accesso Aloha puro e si confrontano i risultati ottenuti con quelli teorici calcolati in [25]. In seguito si presentano i profili applicativi utilizzati per stabilire la scalabilità della cella. Infine si analizzano i risultati forniti dalle simulazioni eseguite con scenari il più possibile simili ad una reale cella LoRaWAN.

4.3.1. Validazione del simulatore

Per la validazione del simulatore è stato considerato, ad ogni simulazione, uno scenario con un singolo profilo applicativo, un unico data rate ed un solo canale per tutti i dispositivi. In questo modo si hanno

messaggi tutti omogenei tra di loro ed un solo mezzo di trasmissione in cui tutti i pacchetti inviati non sono ortogonali e quindi collidono.

Per ricreare le condizioni di accesso Aloha puro è stata aggiunto un metodo alla classe ‘Gateway’ che non considera il capture effect, presente in LoRa: pertanto due trasmissioni sovrapposte temporalmente collidono sempre e sono entrambe scartate, indipendentemente dagli RSSI posseduti. Sappiamo da [25] che la tecnica di accesso casuale Aloha puro prevede una probabilità di successo nella trasmissione di un pacchetto definita come:

$$P_{succ} = e^{-2*G} \quad (17)$$

Dove:

$$G = \sum_{i=1}^N \frac{t_i}{\tau} \quad (18)$$

Con:

- N: numero di dispositivi;
- t_i : durata trasmissione pacchetto dell’ i -esimo dispositivo;
- τ : intervallo di trasmissione.

Pertanto per verificare la correttezza del software, è sufficiente confrontare l’andamento del DER al variare del numero di dispositivi ottenuto durante le simulazioni con la funzione descritta dall’equazione (17).

I parametri del profilo applicativo considerato sono riassunti in **Tabella 17**.

Dimensione MACPayload	9 byte
Npp	8
IH	1
CR	1
CRC	1
Intervallo di trasmissione	60 s

Tabella 17. Parametri del profilo applicativo utilizzato durante la validazione del simulatore.

Inoltre, per verificare l’impatto dello spreading factor sulla scalabilità, sono state eseguite due batterie di simulazioni le cui configurazioni sono riportate in Tabella 18.

	Configurazione #01	Configurazione #02
SF	07	11
BW	125 kHz	125 kHz
ToA¹	56.58 ms	659.46 ms

1: calcolato con le equazioni (12), (13), (14) e (15).

Tabella 18. Configurazioni utilizzate nel primo e nel secondo set di simulazioni per la validazione del simulatore.

Per ogni configurazione sono state effettuati numerosi set di prove. Esse si distinguono per il numero di dispositivi impiegati (N). Ciascun set è composto da 100 simulazioni, grazie alle quali è stato possibile ricavare il valor medio del DER, utilizzato per descriverne l'andamento. I valori di DER ottenuti dai test sono riportati, insieme all'andamento teorico calcolato in base all'equazione (17), in Figura 16. Come si può osservare la probabilità di successo delle trasmissioni decade esponenzialmente, in accordo con la teoria. Pertanto possiamo affermare che il software esegue correttamente l'operazione di analisi delle interferenze in condizioni di assenza di capture effect, singolo canale e singolo data rate.

Con le configurazioni utilizzate notiamo che il numero di dispositivi supportati da un gateway per avere un'affidabilità della rete sufficiente ($DER [\%] \geq 90\%$) è piuttosto esiguo: circa 50 end devices con la configurazione #01 e addirittura meno di 10 con la configurazione #02. Si attendono risultati più incoraggianti con l'introduzione di più data rate e diversi canali fisici da utilizzare simultaneamente per le trasmissioni.

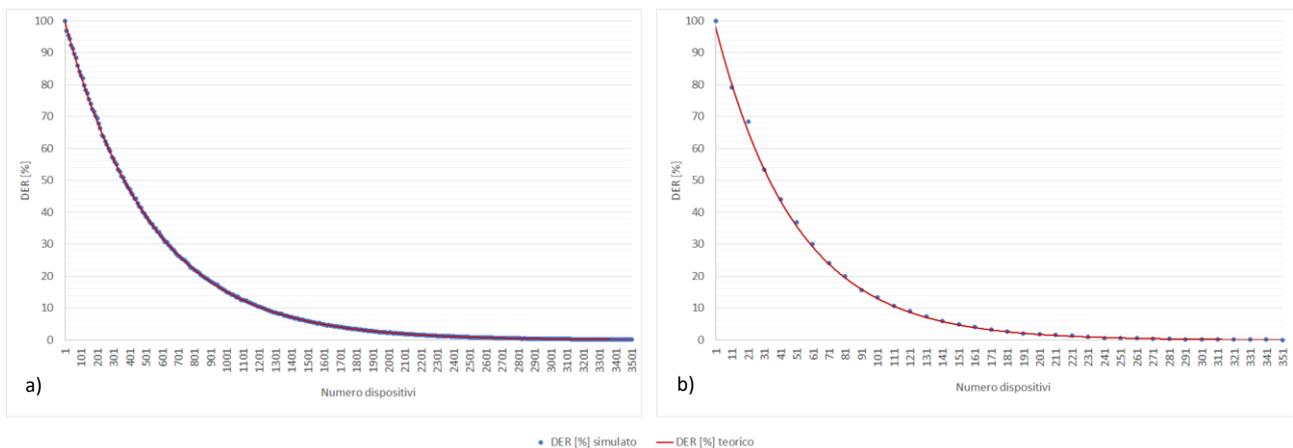


Figura 16. Confronto tra i DER [%] ottenuti con le simulazioni e calcolati con l'equazione (17). a) configurazione #01; b) configurazione #02. Per le configurazioni si fa riferimento alla Tabella 18.

4.3.2. Analisi del capture effect

I test successivi sono stati realizzati con una procedura analoga alla precedente: diversi set di simulazioni con differenti numeri di dispositivi (N), ciascun set composto da 100 simulazioni. La differenza rispetto

alle simulazioni precedenti è l'utilizzo contemporaneo di più data rate e l'aumento progressivo del numero di canali impiegati. Inoltre, per verificare le funzionalità del modello d'analisi d'interferenza con il capture effect, ciascuna simulazione è stata effettuata sia considerando il capture effect sia trascurandolo.

Il pacchetto utilizzato nei test mantiene la struttura riportata in Tabella 17 mentre le quattro configurazioni utilizzate sono descritte in Tabella 19.

	Configurazione #01	Configurazione #02	Configurazione #03	Configurazione #04
N_{SF}¹	1	5	5	5
SF	07	07, 08, 09, 10, 11	07, 08, 09, 10, 11	07, 08, 09, 10, 11
BW	125 kHz	125 kHz	125 kHz	125 kHz
N_{CH}²	1	1	3	8

1: numero di spreading factor utilizzati.

2: numero di canali utilizzati.

Tabella 19. Configurazioni utilizzate nei diversi set di simulazioni per l'analisi del capture effect.

I DER [%] ottenuti dalle simulazioni sono rappresentati in Figura 17. Tralasciando temporaneamente il capture effect, notiamo come l'introduzione di diversi spreading factor (configurazione #02) aumenti più del doppio il numero di dispositivi supportati per ciascun DER rispetto al singolo spreading factor (configurazione #01). Ad esempio, alla soglia del DER pari al 90% sono stati ottenuti circa 50 end device nella prima configurazione e 100 nella seconda. Anche l'aumento dei canali provoca un miglioramento sostanziale delle prestazioni. In particolare, osservando il numero di end device gestiti da un solo gateway per avere $DER [\%] \geq 90\%$, con le configurazioni #02, #03 e #04 si nota una crescita lineare all'inserimento di ogni canale. I dettagli dei numeri di dispositivi supportati con il DER [%] pari al valore di soglia sono mostrati in Figura 18.

Come era lecito ipotizzare, l'aggiunta anche del capture effect nel modello di analisi delle interferenze ha introdotto un ulteriore miglioramento delle prestazioni. L'incremento del numero di dispositivi supportati con DER prossimo al valore di soglia si è attestato a circa il 20%.

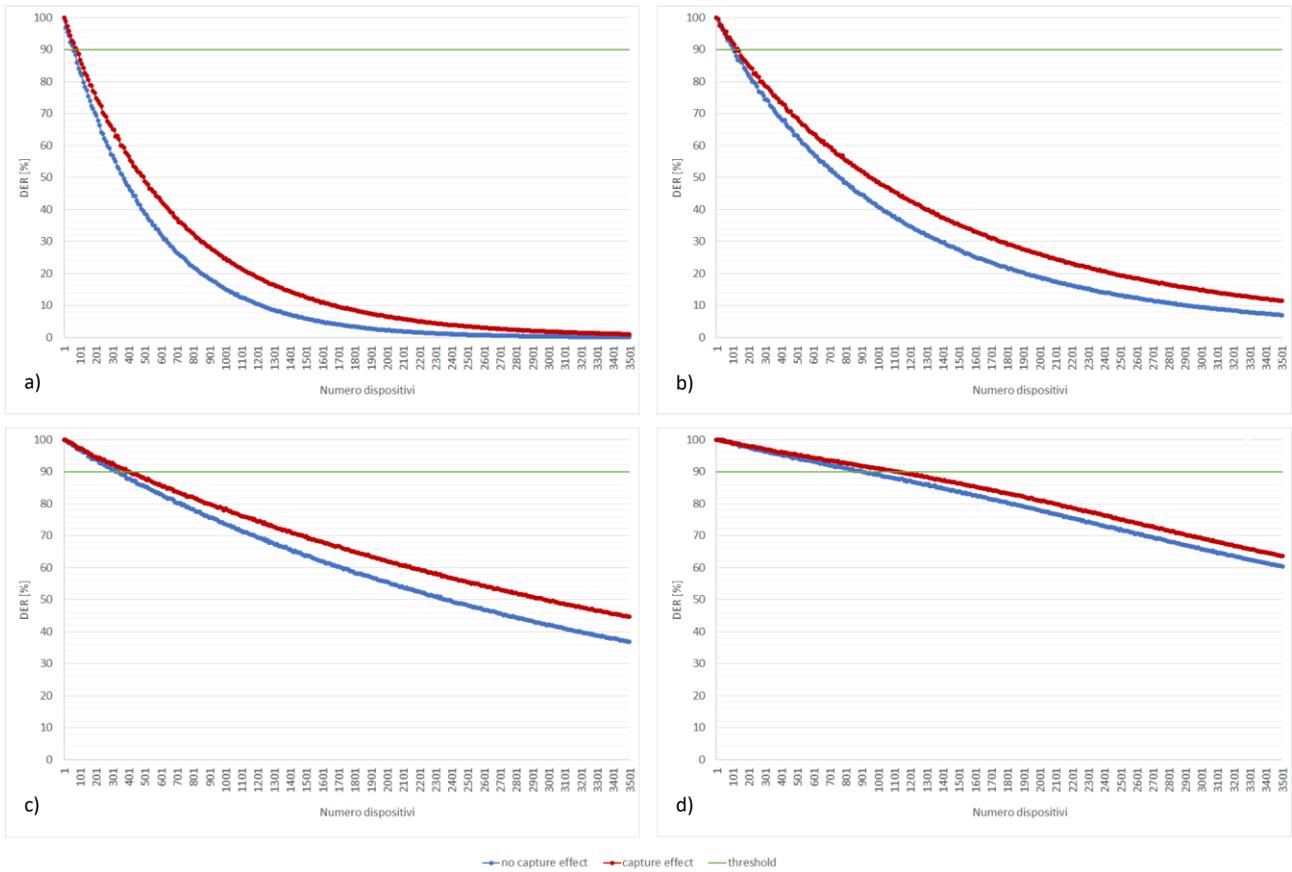


Figura 17. Confronto tra i DER [%] ottenuti senza e con capture effect. a) configurazione #01; b) configurazione #02; c) configurazione #03; d) configurazione #04. Per le configurazioni si fa riferimento alla Tabella 19.

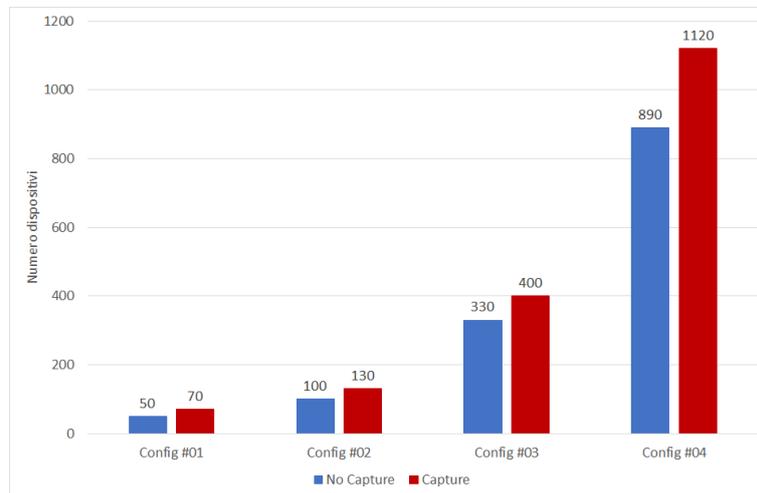


Figura 18. Numero massimo di dispositivi supportati da un singolo gateway per avere DER [%] $\geq 90\%$.

Riassumendo le indicazioni ottenute dalle analisi presentate in questo paragrafo, affermiamo che l'utilizzo di diversi spreading factor, coadiuvato dalla trasmissione su canali multipli, fornisce un considerevole aumento delle prestazioni; inoltre l'impatto prodotto dal capture effect sul DER[%] risulta

non trascurabile e pertanto tale fenomeno deve essere tenuto in considerazione nella stima delle performance delle reti LoRaWAN. Il dato non confortante tuttavia emerso dalle simulazioni svolte è il numero massimo di dispositivi supportati per avere un DER accettabile ($\geq 90\%$): circa 1100 dispositivi, un valore molto distante dai 10000 end node dichiarati da Semtech [23].

4.3.3. Simulazioni della cella LoRaWAN

Tutti i test presentati sino ad adesso hanno considerato una sola tipologia di pacchetto, ovvero un solo profilo applicativo. In diversi paragrafi del presente lavoro abbiamo sottolineato la sostanziale differenza tra tale scenario ed un contesto reale in cui molti casi d'uso utilizzano una stessa rete per trasmettere informazioni diverse. In questa sezione superiamo il vincolo dell'unica struttura dei messaggi e, dopo aver presentato le applicazioni considerate per le simulazioni di un'ipotetica cella LoRaWAN a Torino, riportiamo i risultati ottenuti sia simulando singolarmente alcuni dei profili applicativi descritti sia eseguendo test con tutti i profili contemporaneamente.

4.3.3.1. I profili applicativi

Come accennato nel paragrafo 2.1, i casi d'uso di maggiore interesse per l'ambito urbano, selezionati tra quelli proposti in [7], sono: la gestione dei rifiuti, il controllo remoto dell'illuminazione, il monitoraggio ambientale e i sensori di parcheggi. In aggiunta si considera anche un servizio per i mezzi di trasporto utile a fornire il numero di passeggeri presenti sui diversi veicoli.

Per tutte le applicazioni la stima iniziale sulla densità dei dispositivi per chilometro quadrato è stata calcolata considerando il numero complessivo di oggetti da connettere e dividendolo uniformemente per la superficie di Torino. Per i sensori di parcheggio e i mezzi di trasporto si sono considerati più valori di densità a partire dalla stima iniziale ottenuta come appena riportato.

Per le dimensioni dei payload ci siamo affidati ai parametri riportati in [7] tranne che per i parcheggi e i mezzi di trasporto, in cui il pacchetto è stato progettato *ad hoc*.

Infine anche per gli intervalli di trasmissione l'articolo [7] ha fornito indicatori molto utili, ma per il caso d'uso dei mezzi pubblici è stato necessario calcolare appositamente la frequenza d'invio dei messaggi.

Profilo applicativo	Densità dei dispositivi [unità/km²]	Dimensione FRMPayload [byte]	Intervallo di trasmissione
Gestione rifiuti	350 ¹	10 ⁵	1/ora ⁵
Controllo illuminazione	760 ²	50 ⁵	5/giorno ⁵
Monitoraggio ambientale	1	50 ⁵	1/ora ⁵
Sensori di parcheggio	500 ÷ 5000 ³	23	60/giorno ⁵
Mezzi di trasporto	9 ÷ 1000 ⁴	9	1/minuto

1: calcolato in base alla distribuzione dei cassonetti riportata dall'Amiat, <http://www.amiat.it/cms/servizi/53-raccolta-e-smaltimento-rifiuti/raccolta-stradale/124-mappa-dei-cassonetti> (visitato 10/10/2017).

2: calcolato in base al numero di punti luce riportata da IREN, http://www.irenenergia.it/ChiSiamo/Attivita/Servizi_energetici/III/Pubblica.html (visitato 10/10/2017).

3: calcolati in base al numero di posti auto a pagamento gestiti da GTT, [https://it.wikipedia.org/wiki/GTT_\(azienda\)](https://it.wikipedia.org/wiki/GTT_(azienda)) (visitato 10/10/2017).

4: calcolati in base al numero di pullman e tram che compongono la flotta GTT, [https://it.wikipedia.org/wiki/GTT_\(azienda\)](https://it.wikipedia.org/wiki/GTT_(azienda)) (visitato 10/10/2017).

5: valore tratto da [7].

Tabella 20. Parametri considerati nelle simulazioni di un'ipotetica cella LoRaWAN sul territorio urbano di Torino.

Per il sistema dei mezzi di trasporto, la dimensione del payload è basata sulle informazioni che ciascun messaggio deve trasportare: un timestamp che indichi l'orario in cui è stato effettuato il rilevamento (3 byte, per una precisione al secondo nell'arco di 24 ore), il numero della linea (2 byte), l'identificativo del mezzo (2 byte) e il numero dei passeggeri (2 byte), per un totale di 9 byte.

Nel dimensionamento dell'intervallo di trasmissione è stato calcolato il tempo minimo che intercorre tra due fermate di una stessa linea, in modo da determinare ogni quanto tempo possa cambiare il numero di passeggeri.

$$\bar{T}_{min} = \frac{d_{min}}{\bar{v}_{max}} = \frac{300 \text{ m}}{4,72 \text{ m/s}} = 63,56 \text{ s} > 60 \text{ s}$$

Dove:

- d_{min} : distanza minima obbligatoria presente tra due fermate successive qualsiasi delle linee urbane GTT⁵;

⁵ http://torino.repubblica.it/cronaca/2016/10/04/news/torino_gtt_taglia_le_fermate_per_aver_mezzi_piu_veloci_il_test_sulla_linea_4-149084850/ (visitato 10/10/2017)

- \bar{v}_{max} : massimo tra la velocità media registrata dai tram e dai pullman sulle linee urbane GTT⁶.

Si è ottenuto dunque un intervallo per il profilo applicativo di circa 1 minuto.

Noti l'intervallo di trasmissione e la dimensione del payload applicativo, si riportano in Tabella 21 i tempi di inattività dovuti ai principali duty cycle presenti sulle reti LoRaWAN. In particolare, con un DC pari all'1%, vincolo presente sulla rete The Things Network disponibile a Torino, si osserva un valore di T_{off} inferiore all'intervallo di trasmissione richiesto fino allo spreading factor 10; con SF11 si ha un ritardo di circa 10 secondi nell'invio dei messaggi, mentre con SF12 il tempo di inattività è superiore al doppio del periodo calcolato per l'applicazione.

Data rate	Time on air [ms]	Tempo di inattività		
		DC 10%	DC 1%	DC 0.1%
5 (SF = 7)	56.58	509.22 ms	5.6 s	56.52 s
4 (SF = 8)	102.91	926.19 ms	10.19 s	102.81 s
3 (SF = 9)	205.82	1.85 s	20.38 s	205.61 s
2 (SF = 10)	370.69	3.34 s	36.7 s	370.32 s
1 (SF = 11)	741.38	6.67 s	73.4 s	740.64 s
0 (SF = 12)	1482.75	13.34 s	146.79s	1481.27 s

Tabella 21. Time on air dei pacchetti progettati per l'applicazione per i mezzi di trasporto al variare del data rate e relativi tempi di inattività dei dispositivi. Si evidenziano (rosso) i tempi di inattività superiori all'intervallo di trasmissione fissato per l'applicazione specifica (pari a 60 s).

Nello studio del payload applicativo per il sistema di monitoraggio dei parcheggi è stato ipotizzato il seguente insieme di informazioni da trasmettere in ogni messaggio: il timestamp del rilevamento (3 byte), la latitudine e la longitudine del parcheggio (8 + 8 byte), il numero della piazzola di sosta (3 byte) e la tipologia dell'evento (1 byte per indicare arrivo o partenza di un veicolo), per un totale di 23 byte. In Tabella 22 sono presenti i diversi ToA per questo profilo applicativo e i relativi tempi di inattività imposti dai duty cycle. Per il caso di DC pari all'1% si ottiene un T_{off} obbligatorio di circa 3 minuti, che non crea problemi nel notificare il cambio di stato di un parcheggio poiché la sosta a pagamento prevista da GTT ha una durata minima nell'ordine dei 15-20 minuti.

⁶ <http://www.lastampa.it/2016/10/05/cronaca/a-torino-i-tram-pi-lenti-deuropa-ZEBXbZlwM2P7rGKvRy8ZRN/pagina.html> (visitato 10/10/2017)

Data rate	Time on air [ms]	Tempo di inattività		
		DC 10%	DC 1%	DC 0.1%
5 ($SF = 7$)	77.06	693.54 ms	7.63 s	76.98 s
4 ($SF = 8$)	143.87	1.29 s	14.24 s	143.73 s
3 ($SF = 9$)	267.26	2.41 s	26.46 s	266.99 s
2 ($SF = 10$)	493.57	4.44 s	48.86 s	493.08 s
1 ($SF = 11$)	987.14	8.88 s	97.73 s	986.15 s
0 ($SF = 12$)	1974.27	17.77 s	195.45 s	1972.3 s

Tabella 22. Time on air dei pacchetti progettati per l'applicazione per i parcheggi al variare del data rate e relativi tempi di inattività dei dispositivi.

I profili applicativi presentati nel paragrafo sono stati utilizzati nelle successive simulazioni per la stima definitiva sulla scalabilità di una cella LoRaWAN. Le configurazioni dei test e i relativi risultati sono descritti nel paragrafo seguente.

4.3.3.2. I risultati

Le simulazioni effettuate per quantificare la scalabilità di una cella LoRaWAN si sono sviluppate in più fasi. Inizialmente è stato analizzato il DER offerto da un gateway che gestisse un numero progressivamente crescente di end node, dediti alla sola applicazione per i mezzi di trasporto. In seguito sono stati ripetuti i test considerando unicamente i sensori di parcheggio. Successivamente sono state effettuate le prime simulazioni con profili applicativi misti, utilizzando dispositivi sia per i mezzi pubblici sia per i parcheggi. Infine sono stati considerati contemporaneamente tutti i profili applicativi presentati nel paragrafo precedente.

Si sottolinea come in tutti questi casi non sia stato utilizzato il data rate con SF pari a 12. La scelta è dovuta ai lunghi tempi di inattività previsti per tale data rate, non appropriati per l'applicazione dei mezzi di trasporto; di conseguenza il vincolo si è esteso a tutti i dispositivi poiché il simulatore non prevede una selezione specifica per ciascun profilo applicativo di tale parametro. L'aspetto negativo di perdere la configurazione a massimo raggio di copertura ha comunque un effetto limitato sulle performance: infatti la cella ipotizzata e descritta nel paragrafo 4.1 ha dimensioni tali per cui tutti gli altri data rate riescono ad inviare messaggi senza problemi di copertura. Inoltre con lo spreading factor massimo si avrebbero tempi di trasmissione molto lunghi e rischi di interferenze maggiori.

Iniziamo dunque con le simulazioni per il sistema dei mezzi di trasporto. In questo primo scenario i test sono stati effettuati impostando la dimensione del payload a 9 byte, l'intervallo di trasmissione a 60 secondi e la densità dei dispositivi con valori compresi tra 9 e 1000 unità/km². Il software ha eseguito le simulazioni più volte utilizzando 3 diverse configurazioni, tutte dotate di 5 data rate ma differenziate in termini di numero di canali: un solo canale per la prima configurazione, 3 canali per la seconda e 8 canali per la terza. La prima configurazione, utilizzata anche nelle prove di validazione, fornisce risultati confrontabili con quelli ottenuti nel paragrafo precedente. La seconda rispecchia i canali minimi

obbligatorie previste dalle specifiche di LoRaWAN [18]. Infine la terza descrive la situazione presente sulla rete di The Things Network attiva a Torino. Le configurazioni sono riassunte in Tabella 23.

	Configurazione #01	Configurazione #02	Configurazione #03
N_{SF}¹	5	5	5
SF	07, 08, 09, 10, 11	07, 08, 09, 10, 11	07, 08, 09, 10, 11
BW	125 kHz	125 kHz	125 kHz
N_{CH}²	1	3	8

1: numero di spreading factor utilizzati.

2: numero di canali utilizzati.

Tabella 23. Configurazioni utilizzate nei diversi set di simulazioni per l'analisi della scalabilità di una cella LoRaWAN.

Osservando i risultati delle simulazioni in Figura 19. DER [%] ottenuti simulando diverse densità di dispositivi con il profilo applicativo dei mezzi di trasporto e utilizzando le configurazioni riportate in Tabella 23. si nota che nel caso di soli 9 end node per km² - densità molto ridotta ottenuta dividendo i 1100 mezzi della flotta della GTT per la superficie urbana di Torino di circa 130 km² - tutte le configurazioni offrono un DER [%] superiore al 90%. Uno scenario simile è comunque poco probabile, poiché equivarrebbe ad avere solamente 64 pullman all'interno di una cella estesa per 7 km² (superficie maggiore dell'intero centro città torinese). Aumentando il numero di dispositivi simulati, le configurazioni #02 e #03 offrono buoni risultati, raggiungendo rispettivamente 50 end node per km² - equivalenti ad oltre 350 mezzi per cella - e 100 end node per km² - pari ad oltre 700 mezzi. Il dettaglio del numero massimo di dispositivi supportati da un singolo gateway per ciascuna configurazione è riportato in Figura 20. Numero massimo di dispositivi supportati nelle diverse configurazioni della Tabella 23 con il profilo applicativo dei mezzi di trasporto..

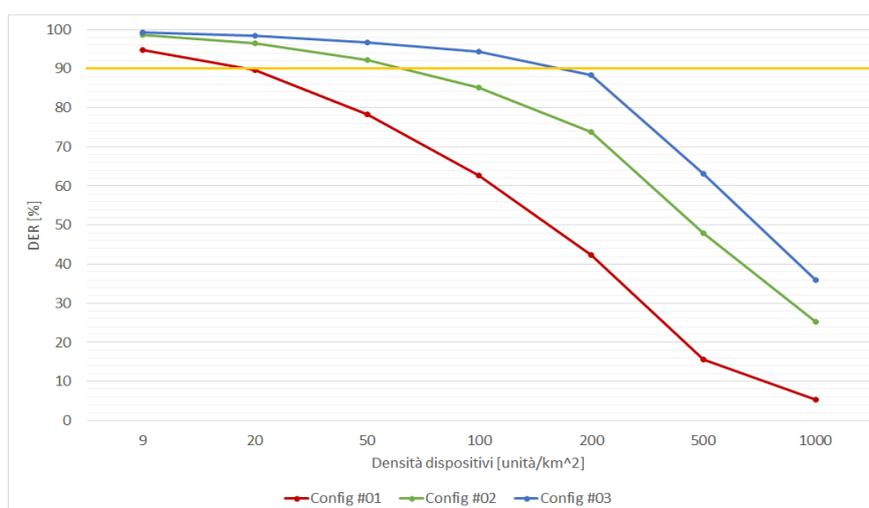


Figura 19. DER [%] ottenuti simulando diverse densità di dispositivi con il profilo applicativo dei mezzi di trasporto e utilizzando le configurazioni riportate in Tabella 23.

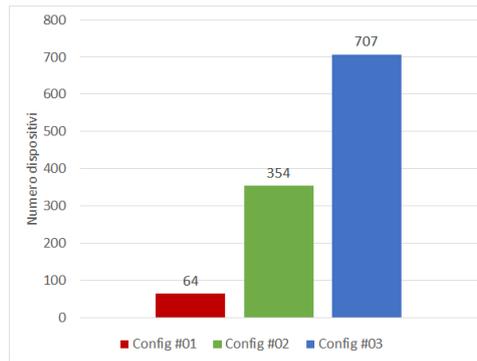


Figura 20. Numero massimo di dispositivi supportati nelle diverse configurazioni della Tabella 23 con il profilo applicativo dei mezzi di trasporto.

Procediamo con i risultati ottenuti simulando il profilo applicativo dei parcheggi. Per queste simulazioni sono stati utilizzati un payload di dimensione pari a 23 byte, un intervallo di trasmissione casuale in modo da inviare circa 60 messaggi ogni 24 ore e delle densità di dispositivi comprese tra i 500 e le 5000 unità al km². Anche per questi test si sono utilizzate le tre configurazioni di rete descritte in Tabella 23. I risultati presentati in Figura 21 mostrano che già alla densità più bassa di 500 end node per km², ovvero 3500 dispositivi per cella, la configurazione #01 non è in grado di garantire un DER [%] oltre la soglia del 90%. Con la configurazione #02 invece si supporta senza affanni tale carico, ma si scende di poco al di sotto della soglia del 90% quando si considerano 1000 end node al km². Infine la configurazione #03 permette di gestire sino a 2000 dispositivi per km² per un totale di oltre 14000 parcheggi per cella. Analogamente a quanto fatto per l'analisi del sistema dei mezzi di trasporto, si riporta in Figura 22 il numero massimo di end node supportati per cella con ciascuna configurazione.

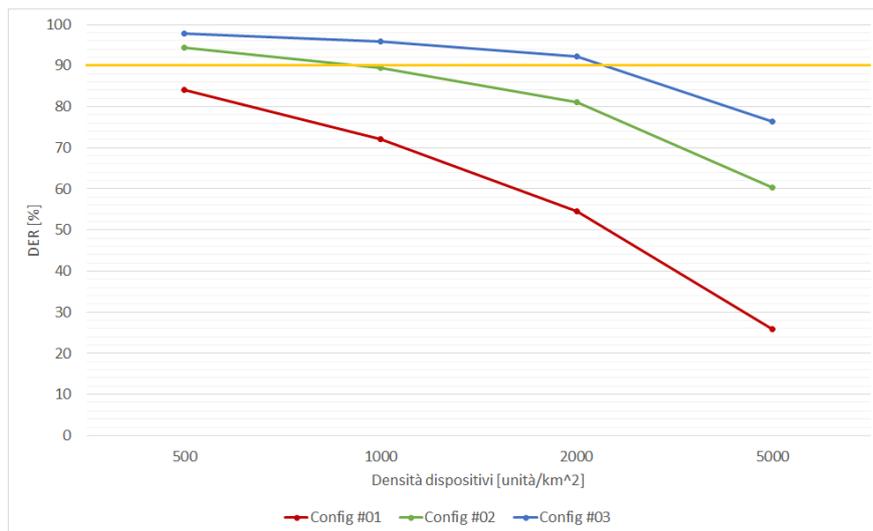


Figura 21. DER [%] ottenuti simulando diverse densità di dispositivi con il profilo applicativo dei mezzi di trasporto e utilizzando le configurazioni riportate in Tabella 23.

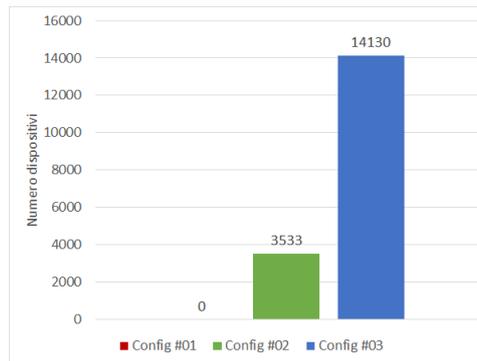


Figura 22. Numero massimo di dispositivi supportati nelle diverse configurazioni della Tabella 23 con il profilo applicativo dei mezzi di trasporto.

Ci apprestiamo adesso a descrivere i risultati delle simulazioni effettuate considerando contemporaneamente sia il profilo applicativo dei mezzi di trasporto sia quello dei sensori di parcheggio. La procedura adottata in questo caso è stata considerare un sottoinsieme delle densità di dispositivi per entrambi i casi d'uso. In particolare, sono stati scelti dei valori più vicini ad uno scenario verosimile per le due applicazioni con 20, 50 e 100 end node al km² per il sistema dei mezzi di trasporto e 500, 1000 e 2000 end node al km² per i parcheggi. Per ciascun valore di densità applicato ai dispositivi di un profilo applicativo sono state condotte più simulazioni con le diverse densità previste per l'altro caso d'uso, in modo da ottenere tutte le combinazioni possibili. Inoltre, viste le deludenti prestazioni ottenute con la configurazione #01, nei successivi test sono state utilizzate solamente la configurazione #02 e la #03.

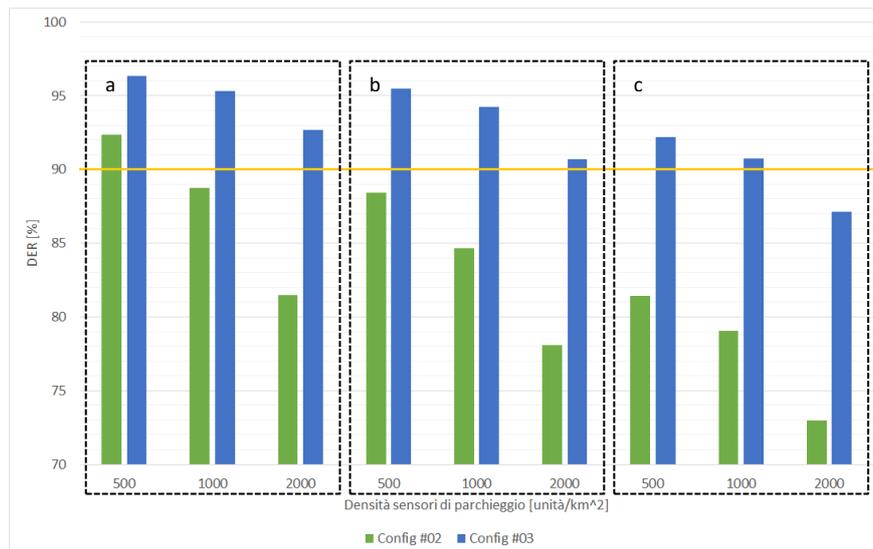


Figura 23. DER [%] ottenuti simulando diverse densità di dispositivi per il profilo applicativo dei mezzi di trasporto e dei sensori di parcheggio: a) densità mezzi di trasporto pari a 20 unità per km²; b) densità mezzi di trasporto pari a 50 unità per km²; c) densità mezzi di trasporto pari a 100 unità per km².

Analizzando gli esiti dei test presentati in Figura 23, osserviamo che la configurazione #02 è sufficiente a gestire 20 pullman e 500 parcheggi al km². In tutte le altre combinazioni di densità i 3 canali non garantiscono un DER adeguato. Utilizzando la configurazione #03 invece, il gateway riesce a supportare tutte le densità impostate, eccetto l'ultima con 100 dispositivi al km², per il sistema dei mezzi di trasporto, e 2000 sensori di parcheggio al km². Confrontando i DER [%] ottenuti in questa sessione di simulazione con quelli presentati in precedenza (Figura 21) notiamo un numero ridotto di densità supportate. In particolare la configurazione #02 non raggiunge più i 50 dispositivi al km² per i mezzi pubblici quando deve gestire 1000 o 2000 sensori di parcheggio al km². Per la configurazione #03 invece non sono più supportati 100 mezzi pubblici al km² se contemporaneamente si hanno 2000 parcheggi al km².

Un altro aspetto interessante che emerge dagli ultimi risultati è che un aumento del numero di dispositivi per i mezzi pubblici di entità minore rispetto ad un incremento di sensori di parcheggio comporta una riduzione del DER [%] equivalente. Considerando ad esempio la configurazione #02, si ha una diminuzione del DER [%] di circa 10 punti percentuali tra lo scenario con 20 mezzi pubblici più 500 parcheggi al km² e il caso con 20 mezzi pubblici più 2000 parcheggi al km². La stessa variazione di DER [%] si osserva, sempre per la configurazione #02, tra le simulazioni effettuate nello scenario con 20 mezzi pubblici più 500 parcheggi al km² e il caso con 100 mezzi pubblici più 500 parcheggi al km². Il grafico di Figura 23 riporta altre situazioni analoghe. Quando si dimensiona una rete LoRaWAN occorre dunque tenere in considerazione l'impatto che ciascun profilo applicativo può avere sulla congestione e l'affidabilità di una cella. Sicuramente l'intervallo di trasmissione dei messaggi - 60 secondi per i mezzi pubblici e 60 messaggi in 24h per i parcheggi - influisce su tale aspetto con un consumo maggiore di "banda" da parte di quei casi d'uso in cui l'invio dei pacchetti è molto frequente (intervalli di trasmissione nell'ordine di pochi minuti).

In Figura 24 si riportano anche per questo insieme di simulazioni il numero massimo di dispositivi supportati dal singolo gateway nelle due configurazioni.

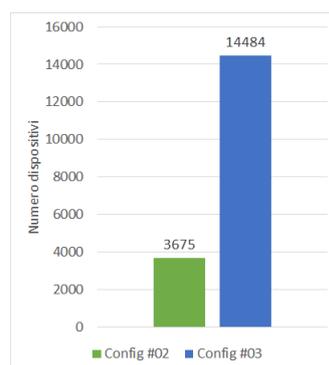


Figura 24. Numero massimo di dispositivi supportati utilizzando dispositivi sia con il profilo applicativo dei mezzi di trasporto che dei sensori di parcheggio.

Il quarto ed ultimo insieme di simulazioni procede nella direzione intrapresa dai test appena riportati, in quanto incrementa ulteriormente il numero di profili applicativi utilizzati simultaneamente nelle simulazioni. In questa fase sono stati considerati tutti i casi d'uso presentati nel paragrafo 4.3.3.1. Per il sistema di gestione dei rifiuti, di controllo dell'illuminazione urbana e di monitoraggio ambientale le densità dei dispositivi sono state mantenute fisse e pari ai valori riportati in Tabella 20. Per quanto riguarda i mezzi pubblici e i sensori di parcheggio invece le densità hanno variato seguendo la stessa procedura utilizzata in precedenza, per considerare tutte le combinazioni. Le dimensioni dei payload e gli intervalli di trasmissione di tutte le applicazioni sono descritti in Tabella 20. Le configurazioni di rete applicate a queste ultime simulazioni sono le stesse impiegate finora, con il solo utilizzo della #02 e #03.

I risultati delle simulazioni sono rappresentati in Figura 25. Considerando i valori di DER [%] ottenuti, si osserva un andamento simile a quello precedentemente riportato in Figura 23. Anche in questo scenario la configurazione #02 supporta tutte le applicazioni solamente se le densità di mezzi pubblici e parcheggi sono quelle minime considerate. La configurazione #03 invece supporta tutti i casi, tranne nello scenario con densità massime per mezzi pubblici e parcheggi. In generale i DER [%] mostrati in Figura 25 hanno una riduzione di circa uno o due punti percentuali rispetto a quelli presenti nei grafici di Figura 23.

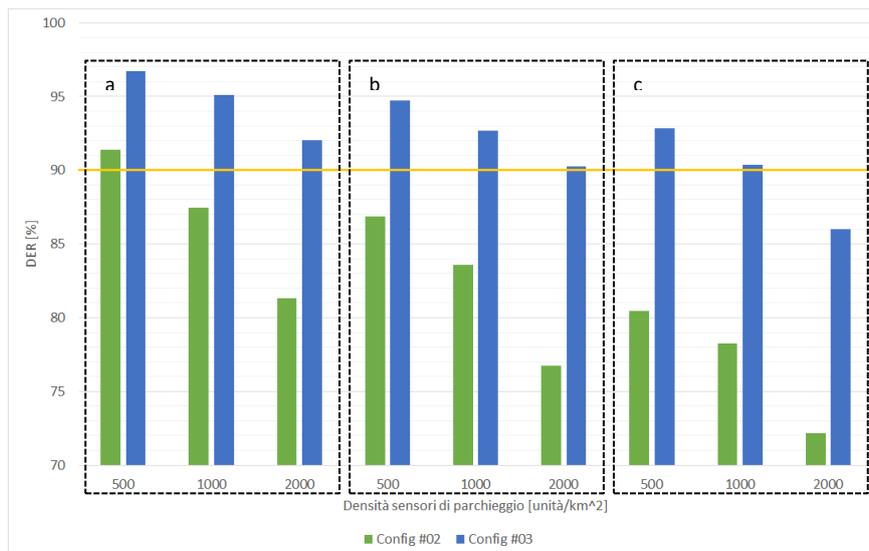


Figura 25. DER [%] ottenuti simulando tutti i casi d'uso contemporaneamente. Per il profilo applicativo dei mezzi di trasporto e dei sensori di parcheggio sono state utilizzate densità variabili: a) densità mezzi di trasporto pari a 20 unità per km²; b) densità mezzi di trasporto pari a 50 unità per km²; c) densità mezzi di trasporto pari a 100 unità per km².

Parlando invece in termini di numero massimo di dispositivi supportati, otteniamo un grande aumento delle prestazioni. Infatti, come si può osservare confrontando la Figura 24 con la Figura 26, si passa da 3675 a 11540 dispositivi per cella con la configurazione #02 e da 14484 a 22349 dispositivi per cella con la configurazione #03.

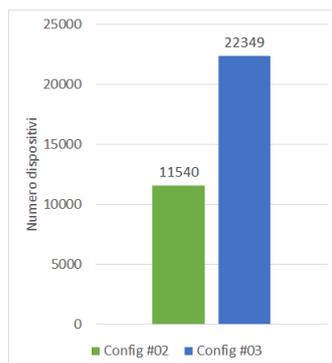


Figura 26. Numero massimo di dispositivi supportati utilizzando dispositivi con tutti i profili applicativi.

Poiché i tre profili applicativi aggiunti in questa ultima serie di simulazioni prevedono intervalli di trasmissione molto maggiori rispetto al caso d'uso dei mezzi di trasporto si avvalorano ulteriormente quanto esposto in precedenza, ovvero che le reti LoRaWAN possono garantire grandi scalabilità tanto più l'intervallo di trasmissione dei diversi messaggi è ampio. In ogni caso anche profili applicativi con invii di messaggi più frequenti possono essere supportati da tale tecnologia, a patto che si utilizzi un numero massimo di dispositivi congruo.

5. Copertura di LoRaWAN

In questo paragrafo si presenta lo studio effettuato a Torino in merito alle prestazioni di trasmissione e di copertura offerte dalla tecnologia LoRaWAN. L'obiettivo del lavoro è stato determinare le dimensioni che dovrebbe assumere una cella della rete nel contesto urbano specifico.

Riproponendo il concetto di cella descritto nel paragrafo 4.1, affermiamo che essa corrisponde ad un'area più o meno estesa nella quale tutti i messaggi generati dai dispositivi sono elaborati con successo da un'unica base station. Come affermato in [26], una cella ha generalmente una forma esagonale nel cui centro è posizionato un gateway. Un valore comune per il raggio della circonferenza che circonda la cella è 1.5 km [26].

Basandoci su tale definizione, occorre trovare una distanza massima al di sotto della quale tutti i messaggi generati dagli end node sono ricevuti dal gateway. Per stabilire tale distanza per la tecnologia LoRaWAN è possibile inviare pacchetti di prova impostati con il data rate che garantisce la minima copertura tra tutte quelle offerte dalle configurazioni di rete disponibili (Tabella 7), ovvero il numero 05 (SF = 7). Pertanto se un messaggio così configurato giunge incolume al gateway, allora anche tutte le altre configurazioni non incontrano difficoltà nelle trasmissioni.

5.1. Realizzazione dei test

Per la realizzazione dei test è stato sfruttato il gateway installato dall'azienda informatica 5T. Tale base station rientra nel progetto della community The Things Network di Torino⁷. Il modello del gateway è Wirnet Station 868, prodotto da Kerlink, e sfrutta il chip SX1301 descritto in precedenza.



Figura 27. Gateway installato sul tetto degli uffici dell'azienda informatica 5T.

⁷ Community "The Things Network" di Torino: <https://www.thethingsnetwork.org/community/turin/>.

La decodifica dei pacchetti avviene su tutti i data rate con indice compreso tra 0 e 5 (estremi inclusi). L'altezza dell'antenna è di circa 25 metri rispetto al manto stradale.

Al fine di comunicare con il gateway è stata realizzata un'applicazione con la piattaforma messa a disposizione da The Things Network. Per farlo è stato necessario accedere alla loro console, tramite l'interfaccia web, e definire la struttura del payload dei messaggi da inviare con i dispositivi. Ciascun pacchetto è composto da un solo byte contenente un counter del messaggio calcolato dal programma applicativo in esecuzione sull'end node.

Completata la configurazione sul cloud, è stato assemblato un end node con i seguenti componenti hardware:

- un Arduino UNO: microcontrollore general purpose;
- un LoRa Bee 868 v1.1: scheda prodotta da Dragino che monta il chip Hope RF RFM95. Quest'ultimo è basato sul modulo ricetrasmittitore LoRa SX1276, realizzato da Semtech.

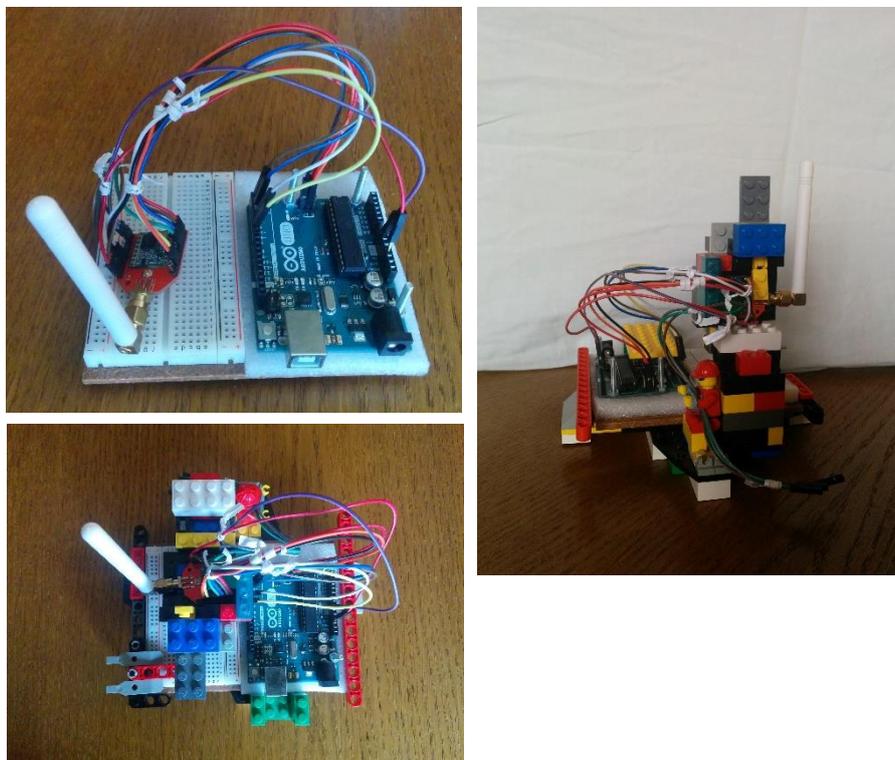


Figura 28. LoRa Bee 868 connesso ad Arduino UNO.

In generale, per costruire un dispositivo da connettere ad una rete LoRaWAN è sufficiente acquistare un solo componente “full stack”, ovvero capace di svolgere tutte le operazioni di rete: gestione del protocollo LoRa MAC e modulazione LoRa per la trasmissione. Tali prodotti sono costituiti da un microcontrollore dedicato e da un chip radio ricetrasmittitore, un esempio è l'RN2483 della Microchip.

Nel nostro caso, il LoRa Bee 868 dispone solamente del chip radio ricetrasmittitore che si occupa di trasmettere i dati via radio. Arduino invece esegue le operazioni previste dal protocollo LoRaMAC gestendo il modulo trasmettitore ed eseguendo il codice applicativo. L'implementazione del software è stata possibile grazie ad un porting della libreria LoRaMAC-in-C, sviluppato appositamente per

Arduino⁸. Esso espone tutte le funzionalità previste nel protocollo LoRaMAC fornendo un'interfaccia di basso livello adeguata ai nostri scopi di test [27].

L'end node è stato dunque programmato⁹ per utilizzare il data rate 05 ed inviare messaggi - conformi alla struttura riportata in precedenza - ogni 10 secondi. Si sottolinea come il ToA dei pacchetti da inviare sia pari a 41 millisecondi e pertanto il T_{off} imposto dal DC all'1% sia 4,08 secondi.

Oltre al software per l'end node, è stata anche sviluppata un'applicazione client¹⁰ in Java da connettere al message broker di The Things Network. Sfruttando le librerie¹¹ prodotte dalla community, quest'ultimo programma è in grado di ricevere tramite internet i messaggi spediti dall'end node corredati di metadati, allegati dal cloud di The Things Network. Tra i parametri aggiunti al messaggio troviamo il frame counter, l'RSSI e l'SNR; questi ultimi due sono calcolati dal gateway durante la ricezione del pacchetto. Alla ricezione di ogni messaggio, l'applicazione Java registra le informazioni inerenti al pacchetto in un file di log.

Completata la fase di programmazione, abbiamo allestito un'automobile assicurando il dispositivo descritto in precedenza alla capote del veicolo, a circa 1.5m da terra, e fornendo alimentazione ad Arduino tramite cavo USB. In seguito ci siamo recati per le strade del centro urbano ed abbiamo avviato i test.



Figura 29. LoRa Bee 868 e Arduino installati sul tetto dell'automobile per i test in città.

⁸ Disponibile su GitHub: <https://github.com/matthijskooijman/arduino-lmic>.

⁹ Lo sketch di Arduino è ospitato nella repository privata: <https://github.com/MicheledG/Tesi-TTNNode>. Per richiederne l'accesso è necessario contattare l'autore all'indirizzo email: michele.digirolamo.to@gmail.com.

¹⁰ Il codice dell'applicazione è ospitato nella repository privata: <https://github.com/MicheledG/Tesi-TTNApp>. Per richiederne l'accesso è necessario contattare l'autore all'indirizzo email: michele.digirolamo.to@gmail.com.

¹¹ Disponibili su GitHub: <https://github.com/TheThingsNetwork/java-app-sdk>.

Per poter tener traccia del tragitto percorso in auto è stata utilizzata l'applicazione *Automatic Mileage Log GPS Tracker for Businesses*, sviluppata da MyCarTracks¹². Essa registra le coordinate GPS dello smartphone e permette di consultarle in un file alla fine del viaggio. Inoltre si è usufruito anche dell'applicazione *TTNMapper*, prodotta da The Things Network per cellulari Android. Il suo compito è inserire su una mappa i punti in cui sono stati inviati i pacchetti LoRa decodificati correttamente da un gateway della community. Le due applicazioni sono state utilizzate contemporaneamente perché solo con *TTNMapper* non sarebbe stato possibile determinare dove si fossero originati gli eventuali pacchetti andati persi durante la trasmissione: tali dati sono stati recuperati conoscendo il percorso effettuato dall'auto, gli istanti di trasmissione dei pacchetti giunti al gateway e l'intervallo di trasmissione tra i messaggi.

Terminata la fase di trasmissione dei pacchetti per le strade della città, sono state raccolte le informazioni prodotte dalle due applicazioni in esecuzione sullo smartphone e dall'applicazione client Java attiva per tutta la durata dei test. Grazie ad un programma scritto appositamente, è stato possibile analizzare i risultati e produrre tre file riassuntivi. Nel primo sono state riportate le coordinate del tragitto ed un timestamp che indica l'orario in cui l'auto si è trovata in ciascuna posizione; nel secondo sono state memorizzate le coordinate dei messaggi ricevuti correttamente dal gateway, il contenuto dei payload e i metadati associati a ciascun pacchetto; nel terzo sono state raccolte solamente le coordinate dei messaggi andati persi.

I dati così organizzati sono stati infine rappresentati graficamente su delle mappe, riportate nel seguente paragrafo, per stabilire gli esiti dei test.

5.2. Risultati

Osservando la Figura 30 con le disposizioni dei pacchetti ricevuti dal gateway e persi durante le trasmissioni, notiamo immediatamente che una cella di raggio pari a 1.5km non è supportata dal set up utilizzato durante i test. Infatti all'interno dell'area evidenziata in figura, si trovano molti pacchetti persi che inficiano il DER [%] della rete. Per perseguire l'obiettivo di determinare la dimensione di una cella LoRaWAN a Torino, i dati ottenuti sono stati inseriti in un foglio di calcolo per ottenere una stima quantitativa del DER al variare della distanza del dispositivo dal gateway. In particolare vediamo in Figura 31 sia il dettaglio del numero di pacchetti ricevuti e persi in funzione della distanza sia l'andamento del DER [%].

Osservando i valori riportati in Figura 31 confermiamo l'inadeguatezza di un raggio pari a 1.5 km per una cella LoRaWAN a Torino, il suo DER [%] infatti è inferiore al 60%. Considerando anche in questo caso una soglia del 90% vediamo che la distanza massima di trasmissione raggiunta nei test è di circa 600 m. Il risultato ottenuto è molto al di sotto delle prestazioni descritte nel paragrafo 3.1.

¹² <https://www.mycartracks.com/>.

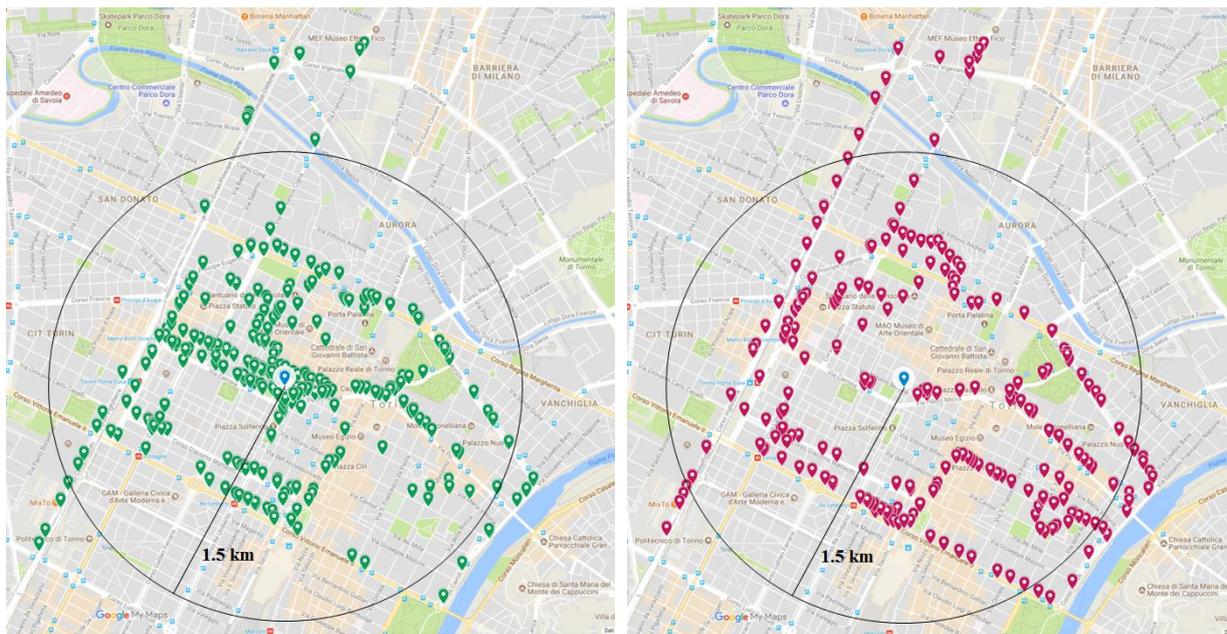


Figura 30. A sinistra sono riportate sulla mappa le posizioni dove sono stati inviati i pacchetti ricevuti correttamente dal gateway. A destra invece le posizioni dei pacchetti persi. Al centro di entrambe le mappe è evidenziato il gateway con un marcatore blu.

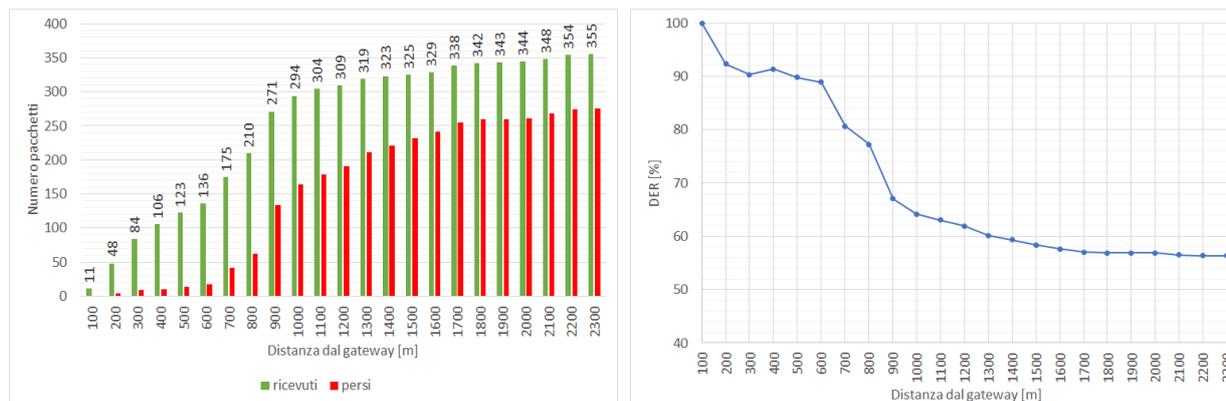


Figura 31. A sinistra sono riportati i numeri di pacchetti ricevuti e persi in funzione della distanza dell'end node dal gateway. A destra invece è mostrato il DER [%] calcolato con i valori presentati nel grafico accanto.

6. Conclusioni e lavori futuri

Nella prima sezione di questa tesi abbiamo introdotto le problematiche legate alla creazione di applicazioni IoT per l'ambito urbano. Le principali difficoltà emerse derivano dalle caratteristiche della connessione da garantire ai dispositivi intelligenti. Essa infatti deve essere a lungo raggio, poiché le base station della rete possono essere anche molto lontane dagli oggetti da connettere, ma allo stesso tempo a basso consumo energetico. Abbiamo dunque visto come le LPWANs siano tecnologie di telecomunicazione nate proprio per coprire questi scenari.

Nel presente studio si è optato per analizzare le prestazioni offerte dalla soluzione LoRaWAN. La scelta è stata effettuata considerando le LPWANs presenti ad oggi nel capoluogo piemontese: l'unica infrastruttura di rete disponibile è quella offerta liberamente dalla community "The Things Network" di Torino, che sfrutta proprio LoRaWAN.

Dopo aver descritto il funzionamento dello stack tecnologico di tale soluzione ci siamo concentrati su due aspetti cruciali per le reti votate all'IoT: la scalabilità in termini di numero di dispositivi supportati da un singolo gateway e la copertura offerta.

Per determinare il numero massimo di end device connessi ad una sola base station, abbiamo inizialmente individuato dei profili applicativi IoT plausibili per il contesto urbano di Torino. In seguito, basandoci sulle caratteristiche della città, abbiamo stimato quanti end device potrebbero essere installati sul territorio nei prossimi anni per ciascuna applicazione. Grazie a queste stime è stato possibile simulare l'attività di una cella LoRaWAN e analizzarne le prestazioni. I risultati ottenuti hanno mostrato che tutti i casi d'uso ipotizzati sono stati gestiti dalla rete. Il singolo gateway simulato è riuscito a supportare oltre 20000 end node. Abbiamo però notato che dispositivi impostati per inviare messaggi molto frequentemente inficiano la scalabilità: simulando solamente end node configurati con un intervallo di trasmissione pari ad 1 minuto, il gateway ha supportato meno di 1000 dispositivi. Occorre dunque considerare nella fase di progetto di un'applicazione se i suoi requisiti siano adatti alle caratteristiche di una rete LoRaWAN o se sia meglio optare per altre soluzioni.

Per quanto concerne la copertura, in prima battuta è stata ipotizzata una cella della rete con raggio pari a 1500 m: infatti le prestazioni dichiarate per LoRa prevedono una distanza massima di trasmissione prossima a tale valore, ottenibile utilizzando la configurazione con lo spreading factor più piccolo e quindi con capacità minima di propagazione del segnale. I test svolti in città hanno però smentito le previsioni, restituendo un raggio massimo di circa 600 m: oltre tale distanza l'affidabilità delle trasmissioni è risultata essere al di sotto del 90%. Pertanto, per creare una rete capace di coprire l'intera superficie di Torino, occorrerebbero più di 100 base station. Si noti che tale stima si basa su prove di trasmissione svolte nel centro città. Quest'ultimo rappresenta un ambiente che possiamo definire un worst case per la rete, infatti un'alta densità di edifici e di altre costruzioni imponenti non favorisce la propagazione dei segnali radio.

L'analisi delle prestazioni di una rete LoRaWAN non si può dichiarare completa con il lavoro qui presentato. Sarebbero presenti altri aspetti da considerare. In particolare si potrebbe indagare se la scalabilità di una rete migliori all'aumentare del numero dei gateway; il simulatore sviluppato per questa tesi è in grado di calcolare il DER di una sola cella gestita da un'unica base station. Inoltre si potrebbero effettuare nuove prove di trasmissione per verificare se sia possibile aumentare la massima distanza tra

il gateway e gli end node nell'invio dei messaggi mantenendo un'affidabilità in ricezione superiore al 90%, sfruttando chip LoRa full stack e considerando differenti altezze di antenne sia del gateway che dei dispositivi. Infine sarebbe opportuno verificare l'intensità dell'eventuale attività elettromagnetica presente sulla banda libera utilizzata da LoRaWAN e determinarne l'impatto sulle comunicazioni.

7. Riferimenti

- [1] O. Vermesan e P. Friess, *Internet of Things - From Research and Innovation to Market Deployment*, River Publishers, 2014.
- [2] F. Mattern e C. Floerkemeier, «From the Internet of Computers to the Internet of Things» 2010.
- [3] The Hammersith Group, «The Internet of things: Network objects and smart devices» 2010.
- [4] D. Evans, «The Internet of Things - How the Next Evolution of the Internet Is Changing Everything» Cisco, 2011.
- [5] A. Nordrum, «The Internet of Fewer Things» 2016.
- [6] 3GPP, «release 13» 2016.
- [7] GSMA, «3GPP Low Power Wide Area Technologies» 2016.
- [8] Nokia, «LTE Evolution for IoT connectivity» 2017.
- [9] Technical Marketing Workgroup 1.0, «LoRaWAN What is it?» LoRa Alliance, 2015.
- [10] I. Brown, «A Detailed Breakdown of LPWAN Technologies and Providers» 2016.
- [11] Leverage, «LPWAN White Paper» 2016.
- [12] ETSI, «Final Draft ETSI EN 300 220-1 v.2.4.1» 2012.
- [13] N. Sornin, M. Luis, T. Eirich, T. Kramp e O. Hersent, «LoRaWAN Specification v.1.0.2» LoRa Alliance, 2016.
- [14] Semtech, «AN1200.22 LoRa Modulation Basics, revisione 2» 2015.
- [15] Wikipedia, «Hata model» *Wikipedia*, vol. https://en.wikipedia.org/wiki/Hata_model.
- [16] Semtech, «SX1272/3/6/8: LoRa Modem - Designer's Guide - AN1200.3» Semtech, 2013.
- [17] IEEE, «IEEE Std 802.15.4™» IEEE, 2006.
- [18] LoRa Alliance, «LoRaWAN™ 1.0.2 Regional Parameters» 2017.
- [19] M. C. Bor e U. Roedig, «Do LoRa Low-Power Wide-Area Networks Scale?» Conference Paper, 2016.
- [20] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melià-Seguf e T. Watteyne, «Understanding the Limits of LoRaWAN» *IEEE Communications Magazine*, Gennaio 2017.
- [21] J. Haxhibeqiri, F. Van den Abeele, I. Moerman e J. Hoebeke, «LoRa Scalability: A Simulation Model Based on Interference Measurements» *Sensors - MDPI*, 2017.

- [22] B. Vejlgaard, M. Lauridsen e H. Nguyen, «Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks» IEEE, 2017.
- [23] Semtech, «SX1301 Datasheet» 2017.
- [24] K. Mikhaylov, J. Petäjärvi e T. Hänninen, «Analysis of the Capacity and Scalability of the LoRa Wide Area Network Technology» European Wireless (EW) conference paper, 2016.
- [25] A. S. Tanenbaum, Computer Networks, New Jersey: Person Education International, 2003.
- [26] R. Huang, H. Li, B. Hamzeh, Y.-S. Choi, S. Mohanty e C.-Y. Hsu, «Proposal for Evaluation Methodology for 802.16p» IEEE 802.16 Broadband Wireless Access Working Group, 2011.
- [27] IBM, «LoRaWAN in C (LMiC) Technical Specification» 2015.