



UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE DELL'INGEGNERIA
CURRICULUM IN INGEGNERIA BIOMEDICA, ELETTRONICA E DELLE
TELECOMUNICAZIONI

Studio, implementazione e analisi delle prestazioni di architetture Bluetooth Mesh per applicazioni IoT basate su protocollo Bluetooth 5 e successivi

Tesi di Dottorato di:
Marco Mercuri

Tutor:
Prof.ssa Paola Pierleoni

Coordinatore del Curriculum:
Prof. Franco Chiaraluce

XX ciclo - nuova serie



UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE DELL'INGEGNERIA
CURRICULUM IN INGEGNERIA BIOMEDICA, ELETTRONICA E DELLE
TELECOMUNICAZIONI

Studio, implementazione e analisi delle prestazioni di architetture Bluetooth Mesh per applicazioni IoT basate su protocollo Bluetooth 5 e successivi

Tesi di Dottorato di:
Marco Mercuri

Tutor:
Prof.ssa Paola Pierleoni

Coordinatore del Curriculum:
Prof. Franco Chiaraluce

XX ciclo - nuova serie

UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE DELL'INGEGNERIA
FACOLTÀ DI INGEGNERIA
Via Brezze Bianche – 60131 Ancona (AN), Italy

Sommario

La ricerca si è concentrata sull'analisi e l'implementazione delle reti Bluetooth mesh, un'importante tecnologia di comunicazione wireless per l'Internet of Things (IoT) che consente ai dispositivi di comunicare tra loro formando una rete di dispositivi interconnessi. Inizialmente, l'obiettivo è stato quello di migliorare l'affidabilità nella ricezione dei messaggi di Status all'interno di una rete Bluetooth Mesh (BM). Sono stati sviluppati codice e firmware per tale scopo e i risultati dei test sperimentali condotti in collaborazione con Nordic Semiconductor in ambiente reale hanno dimostrato un significativo miglioramento delle prestazioni. Successivamente è stato sviluppato un ambiente di simulazione per reti BM per analizzare l'impatto di diverse configurazioni dei parametri di rete sull'affidabilità e sulla latenza end-to-end. I risultati dei test sperimentali hanno evidenziato l'importanza dei parametri temporali e dell'implementazione dello standard. In collaborazione con il team di sviluppo del toolbox Bluetooth di MathWorks, è stata sviluppata una nuova implementazione dello standard, in particolare per la gestione dei messaggi nelle fasi di scansione e advertising nei nodi relay. Ciò ha permesso di ottenere importanti miglioramenti delle prestazioni, migliorando l'affidabilità della rete e riducendo la latenza di trasmissione dei messaggi. La ricerca dimostra l'utilità dell'utilizzo di ambienti di simulazione per testare e validare le prestazioni delle reti BM, particolarmente importante in presenza di una complessità dell'architettura e dei parametri di rete. La collaborazione tra la ricerca accademica e l'industria ha permesso di sviluppare tecnologie sempre più avanzate e performanti per applicazioni IoT sempre più complesse e richiedenti reti di comunicazione affidabili e performanti.

Abstract

The research focused on the analysis and implementation of Bluetooth mesh networks, an important wireless communication technology for the Internet of Things (IoT) that allows devices to communicate with each other by forming a network of interconnected devices. The initial goal was to improve the reliability of receiving Status messages within a Bluetooth Mesh (BM) network. To this end, code and firmware were developed, and the results of experimental tests conducted in collaboration with Nordic Semiconductor in a real environment showed a significant improvement in performance. A simulation environment for BM networks was then developed to analyze the impact of different network parameter configurations on reliability and end-to-end latency. The results of the experimental tests highlighted the importance of timing parameters and standard implementation. In collaboration with the MathWorks Bluetooth Toolbox development team, a new implementation of the standard was developed, specifically for message management in the scanning and advertising phases in relay nodes. This enabled significant performance improvements, improved network reliability, and reduced message transmission latency. The research demonstrates the usefulness of using simulation environments to test and validate the performance of Bluetooth mesh networks, which is especially important when the architecture and network parameters are complex. Collaboration between academic research and industry has enabled the development of increasingly advanced and powerful technologies for increasingly complex IoT applications that require reliable and high-performance communication networks.

Indice

Introduzione	1
1 Internet Of Things e Bluetooth	3
1.1 Internet Of Things e Bluetooth	3
1.2 Stato dell'arte	6
1.3 Obiettivi della ricerca e del lavoro svolto	8
2 Bluetooth Low Energy and Bluetooth Mesh	9
2.1 Bluetooth Low Energy	9
2.1.1 Caratteristiche chiave del BLE	10
2.1.2 Stack overview	11
2.1.3 Advertising	19
2.1.4 Scanning	23
2.2 Bluetooth Mesh	24
2.2.1 Overview	24
2.2.2 Tipi di Nodi	28
2.2.3 Architettura	29
2.2.4 Comunicazioni Broadcast	31
3 Materiali e Metodi	33
3.1 Kit di sviluppo nRF52	33
3.2 Bluetooth Toolbox di MathWorks	35
3.3 Metriche	35
3.3.1 Packet Delivery Ratio	35
3.3.2 Latenza end-to-end	36
4 Miglioramento della ricezione dei messaggi di conferma nel Bluetooth mesh	37
4.1 Messaggi di Status	37
4.2 Setup sperimentale	39
4.3 Evaluation tests design	43
4.4 Risultati	45
4.5 Conclusioni	51

5 Ambiente simulato: PDR e Latenza (TchPDU, scanInterval, SRC-DST)	53
5.1 TchPDU e scanInterval	53
5.2 Analisi proposta	54
5.3 Setup di simulazione	55
5.4 Risultati	57
5.4.1 Packey Delivery Ratio e Latenza per singola coppia Sorgente-Destinatario	57
5.4.2 Performance per topologie multisorgente	59
5.5 Implementazione secondo Mathworks	62
5.6 Conclusioni	64
6 Nuova implementazione dello standard	67
6.1 L'implementazione proposta	68
6.2 Setup di simulazione	70
6.3 Risultati	71
6.3.1 Test con R = 9 m	71
6.3.2 Test con R = 16m	75
6.4 Discussione	77
6.5 Conclusioni	79
7 Miglioramento dell'implementazione della gestione dei messaggi con prelazione	81
7.1 Schema ottimizzazione proposto	81
7.2 Risultati	84
7.2.1 Simulazioni con R = 9m	84
7.2.2 Simulazioni con R = 16m	89
7.3 Confronto complessivo	93
7.4 Conclusioni	95
Conclusione	97

Elenco delle figure

2.1	Stack Bluetooth Low Energy. (Fonte: Bluetooth SIG)	12
2.2	Macchina a stati del Link Layer.(Fonte: Bluetooth SIG)	14
2.3	BLE Physical Channel. (Fonte: [1])	15
2.4	Formato dei pacchetti del Link Layer per i <i>LE Uncoded PHY</i> . (Fonte: [2])	16
2.5	Formato dei pacchetti del Link Layer per i <i>LE Coded PHY</i> . (Fonte: [2])	16
2.6	Advertising Physical Channel PDU. (Fonte: [2])	21
2.7	Advertising Event perturbati nel tempo utilizzando l'advDelay. (Fonte: [2])	22
2.8	Advertising Event Non-connectable and non-scannable undirected che utilizza l'ADV_NONCONN_IND PDU	22
2.9	Parametri di scansione. (Fonte: [3])	24
2.10	Elementi del Bluetooth mesh. (Fonte: [4])	25
2.11	Esempio tologia di una rete mesh. (Fonte: [5])	29
2.12	Architettura del sistema mesh. (Fonte: [5])	31
3.1	nRF52 Development Kit	34
4.1	Esempio del problema di collisione durante il flusso di comunicazione tra tre dispositivi in una rete Bluetooth mesh. (Fonte: [4])	38
4.2	Schema di implementazione della rete Bluetooth mesh dello scenario controllato. (Fonte: [4])	42
4.3	Schema di implementazione della rete Bluetooth mesh dello scenario reale. (Fonte: [4])	44
4.4	Diagramma a blocchi dell' algoritmo di randomizzazione dei messaggi di Status. (Fonte: [4])	45
4.5	Confronto dei PSR degli Status per ciascun test dello scenario controllato, mediato per tutti e dieci i test, sia per i casi CO che CS. (Fonte: [4])	47
4.6	Confronto dei PSR degli Status per ciascun test dello scenario controllato, mediato per tutti e dieci i nodi, sia per i casi CO che CS. (Fonte: [4])	48

Elenco delle figure

4.7	Confronto dei PSR degli Status per ciascun test dello scenario reale, mediato per tutti e dieci i test, sia per i casi CO che CS.(Fonte: [4]) . . .	50
4.8	Confronto dei PSR degli Status per ciascun test dello scenario reale, mediato per tutti e dieci i nodi, sia per i casi CO che CS.(Fonte: [4]) . . .	50
5.1	Topologie utilizzate nelle simulazioni: (a) singola coppia SRC-DST, (b) 2 coppie SRC-DST, (c) 4 coppie SRC-DST, (d) 8 coppie SRC-DST. (Fonte: [6])	55
5.2	Packet Delivery Ratio e Latenza end-to-end per una singola coppia Sorgente-Destinatario al variare dei parametri temporali scanInterval e TchPDU	58
5.3	Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST: (a) intervallo TchPDU $1 \div 2$ ms, (b)intervallo TchPDU $1 \div 10$ ms . . .	60
5.4	Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST: (a) intervallo TchPDU $1 \div 2$ ms, (b)intervallo TchPDU $1 \div 10$ ms . . .	61
5.5	Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST: (a) intervallo TchPDU $1 \div 2$ ms, (b)intervallo TchPDU $1 \div 10$ ms . . .	62
5.6	Esempio di propagazione dei messaggi all'interno del simulatore della Mathworks tra un nodo sorgente (SRC) e un nodo destinatario (DST), separati da due nodi relay intermedi.	63
6.1	Confronto tra la fase di scansione e advertising dei nodi relay dell'implementazione senza prelazione e dell'implementazione con prelazione. (Fonte: [6])	69
6.2	Topologia con 2 coppie sorgente-destinazione ed esempio di copertura. (Fonte: [6])	70
6.3	Packet Delirvery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	72
6.4	Packet Delirvery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	73
6.5	Packet Delirvery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	74
6.6	Packet Delirvery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	75

6.7	Packet Delirvery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	76
6.8	Packet Delirvery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.	77
7.1	Confronto tra la fase di scansione e advertising dei nodi relay dell'implementazione con prelazione e dell'implementazione ottimizzata della prelazione.	82
7.2	Packet Delirvery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	85
7.3	Packet Delirvery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	86
7.4	Packet Delirvery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	87
7.5	Packet Delirvery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	89
7.6	Packet Delirvery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	90
7.7	Packet Delirvery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms	91
7.8	Comparazione complessiva tra l'implementazione senza prelazione e le diverse implementazioni con prelazione negli scenari con R = 9 m: (a) topologia con 2 SRC-DST; (b) topologia con 4 SRC-DST e (c) topologia con 8 SRC-DST	93

Elenco delle figure

7.9	Comparazione complessiva tra l'implementazione senza prelazione e	
	le diverse implementazioni con prelazione negli scenari con $R = 16$	
	m: (a) topologia con 2 SRC-DST; (b) topologia con 4 SRC-DST e (c)	
	topologia con 8 SRC-DST	94

Elenco delle tabelle

4.1 Parametri di configurazione	40
4.2 Risultati complessivi dei PSR degli Status per il test CS dello scenario controlato.	46
4.3 Risultati complessivi dei PSR degli Status per il test CO dello scenario controlato.	47
4.4 Confronto tra i test CO e CS dello scenario controllato.	48
4.5 Risultati complessivi dei PSR degli Status per il test CS dello scenario reale.	49
4.6 Risultati complessivi dei PSR degli Status per il test CO dello scenario reale.	49
4.7 Confronto tra i test CO e CS dello scenario reale.	50
5.1 Parametri di Setup	56

Acronimi

ACK Acknowledged

AdvA Advertising Address

AdvDelay Advertising Delay

AdvInterval Intervallo di Advertising

ARM Advanced RISC Machine

ATT Attribute

BIS Broadcast Isochronous Stream

BLE Bluetooth Low Energy

BM Bluetooth mesh

CIS Connected Isochronous Stream

CO Codice Ottimizzato

CRC Cyclic Redundancy Check

CS Codice Standard

DK Development Kit

DSP Digital Signal Processor

DST Destinazione

E2E End-to-End

FEC Forward Error Correction

GAP Generic Access Profile

GATT Generic ATtribute Profile

Elenco delle tabelle

GCC GNU Compiler Collection

GPIO General Purpose Input/Output

I2C Inter Integrated Circuit

IEEE Institute of Electrical and Electronic Engineers

IoT Internet of Things

ISM Industrial, Scientific and Medical

L2CAP Logical Link Control Adaptation Protocol

LE 1M PHY Low Energy 1 MegaSymbol Physical Layer

LED Light Emitting Diodes

LoRaWAN Long Range Wide Area Network

LPN Low Power Node

LPWAN Low Power Wide Area Network

LTE-M Long-Term Evolution Machine Type Communication

NB-IoT Narrowband Internet of Things

NFC Near Field Communication

PCB printed circuit board

PDR Packet Delivery Ratio

PDU Protocol Data Unit

PHY Physical Layer

PN Provisioner

prel.v1 Implementazione prelazione versione 1

prel.v2 Implementazione prelazione ottimizzata

prel.v2B Implementazione prelazione ottimizzata più tempo di Backoff

PSR Packet Success Rate

QoS Quality of Service

R Raggio di copertura

RF Radio Frequenza

RFID Radio Frequency Identification

RTT Round Trip Time

RX Ricezione

SDK Software Development Kit

SIG Special Interest Group

SMP Security Manager Protocol

SoC System on Chip

SPI Serial Peripheral Interface

SRC Sorgente

TadvEvent Tempo tra due Advertising Event successivi

TchPDU Tempo tra l'inizio della trasmissione di due PDU consecutivi

TTL Time-To-Live

TX Trasmissione

UART Universal Asynchronous Receiver-Transmitter

UNACK unacknowledged

UWB Ultra Wideband

Wi-Fi Wireless Fidelity

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

Introduzione

Negli ultimi anni, l'Internet of Things (IoT) si è diffuso ampiamente in vari settori, come la domotica, la produzione industriale, la logistica e la salute. Tuttavia, l'adozione crescente di dispositivi connessi e la necessità di una comunicazione sempre più efficiente e affidabile rappresentano nuove sfide nella progettazione delle reti IoT. In questo contesto, tra le diverse tecnologie di comunicazione a corto raggio, quali il Wifi, ZigBee, Thread e Z-Wave, il Bluetooth si è imposto come uno dei principali standard per la comunicazione wireless grazie alla sua ampia diffusione, semplicità d'uso, scalabilità e robustezza. In particolare, la versione 5 del Bluetooth introduce una serie di vantaggi come maggiore velocità, copertura e capacità trasmissiva, che lo rendono ancora più efficiente per le comunicazioni IoT e per soddisfare le nostre esigenze. Tuttavia, l'implementazione di reti Bluetooth presenta alcune problematiche che devono essere affrontate, come l'affidabilità della ricezione dei messaggi e la latenza end-to-end, che possono influire sulla qualità del servizio offerto dalle reti e sulla loro capacità di supportare applicazioni sempre più complesse. Uno dei principali problemi che gli sviluppatori di reti Bluetooth mesh devono affrontare è l'affidabilità della trasmissione dei messaggi all'interno della rete. Le reti Bluetooth mesh differiscono dalle reti wireless tradizionali perché sono composte da numerosi nodi interconnessi. Questo significa che la configurazione e il debug della rete richiedono una maggiore attenzione ai dettagli e sono più complessi rispetto alle reti wireless tradizionali. In altre parole, ogni nodo della rete deve essere configurato correttamente affinché l'intera rete funzioni in modo efficace e efficiente. La ricerca presentata si focalizza su questi aspetti, cercando di migliorare le prestazioni delle reti Bluetooth mesh attraverso lo sviluppo di codice e firmware dedicati e l'utilizzo di un ambiente simulato per l'analisi delle prestazioni. In particolare, si analizzano le specifiche delle versioni più recenti del protocollo Bluetooth per comprendere le caratteristiche e le opportunità offerte dal Bluetooth Mesh. Successivamente, si procede all'implementazione di un'architettura di rete basata su Bluetooth Mesh. Il progetto mira a migliorare l'affidabilità nella ricezione dei messaggi di Status in una rete Bluetooth Mesh attraverso lo sviluppo di codice per dispositivi embedded. In collaborazione con la Nordic Semiconductor, verranno realizzati dei setup sperimentali per confrontare la procedura proposta con quella basata sulla configurazione standard. Tuttavia, la complessità dell'architettura e l'elevata numerosità dei parametri di rete che influenzano il comportamento della rete

stessa, richiedono necessariamente l'utilizzo di strumenti di simulazione per garantire un elevato livello di affidabilità e ripetibilità dei test. Pertanto, la ricerca si concentrerà sullo sviluppo di un ambiente di simulazione per reti Bluetooth Mesh, utilizzando i toolbox messi a disposizione dalla Mathworks e conformi allo standard Bluetooth SIG. I test sperimentali saranno condotti per analizzare l'impatto che diverse configurazioni dei parametri di rete hanno sull'affidabilità, in termini di Packet Delivery Ratio (PDR) e latenza end-to-end. Infine, in collaborazione con il MathWorks Bluetooth Toolbox Development Team, verrà realizzata e ottimizzata una nuova implementazione dello standard che riguarda la gestione dei messaggi nella fase di scansione e advertising nei nodi relay.

Capitolo 1

Internet Of Things e Bluetooth

1.1 Internet Of Things e Bluetooth

Nel 1999 l'ingegnere inglese Kevin Ashton formulò per la prima volta l'espressione Internet of Things in stretta relazione con i dispositivi RFID [7]. Da allora le tecnologie IoT si sono moltiplicate e sviluppate, così come si sono profondamente evoluti i numerosi ambiti applicativi. Le applicazioni abbracciano ogni settore, da quello domestico e industriale con la creazione di Smart Home e Smart Building, a quello urbano (smart City) e sanitario, fino ad arrivare ai settori automotive, agricolo, logistico, ambientale ecc. Lo sviluppo di un paradigma che non conosce, potenzialmente, confini applicativi e che ha come obiettivo quello di creare una connettività pervasiva tra persone, dispositivi o persino cose [8], è stato possibile grazie anche alla progressiva e costante evoluzione delle tecnologie di comunicazione wireless. L'eterogeneità degli scenari applicativi IoT, e conseguentemente la diversità dei dispositivi in gioco, implica esigenze differenti nei requisiti di comunicazione. Caratteristiche come raggio di copertura, throughput, modalità di trasmissione connected-oriented o connectionless, consumo di potenza, capacità di rete, livelli di affidabilità, sicurezza, ecc [9] differiscono da un caso d'uso all'altro e devono essere opportunamente modellati. Per far fronte alle diverse necessità, nel corso degli anni sono stati sviluppati e migliorati numerosi standard di comunicazione, che possono essere suddivisi in due ampie categorie sulla base della copertura radio : tecnologie a lungo e corto raggio. Nel primo caso rientrano standard come SigFox [10] and LoRaWAN [11], NB-IoT [12] and LTE-M [13]; tecnologie appartenenti alle cosiddette Low Power Wide Area Network (LPWAN) e caratterizzate da una bassa potenza, basso bit rate, basso costo, lungo raggio (qualche chilometro) e bassa complessità sia di installazione che di utilizzo. Per le loro caratteristiche vengono ampiamente impiegate in ambiti come lo Smart Cities e applicazioni in aree remote. Tuttavia le carenze nella velocità, nel throughput, e la loro complessità di sviluppo non riescono a far fronte alle esigenze delle applicazioni a corto raggio, necessità che invece possono essere sopperite dall'eterogeneità di servizi e caratteristiche offerte da tecnologie come Thread [14], Wi-fi, ZigBee [15], Z-Wave [16], UWB [17].

Capitolo 1 Internet Of Things e Bluetooth

Tra queste tecnologie, si inserisce anche il Bluetooth, che grazie al costante sviluppo e alla sua capacità di saper rispondere prontamente alle esigenze del crescente sviluppo tecnologico, gli hanno permesso di ottenere un posto di prestigio nelle applicazioni a corto raggio.

Sviluppato per la prima volta negli anni novanta dalla multinazionale svedese Ericsson con l'intento di creare un'interfaccia radio a corto raggio e a basso costo per la connessione di periferiche, come auricolari e stampanti, ai telefoni cellulari e computer, il Bluetooth ha visto importanti evoluzioni nel corso degli ultimi 20 anni. Evoluzioni che gli hanno consentito di rendere le connessioni sempre più veloci e sicure, di migliorare il consumo di potenza, la copertura radio, accrescere la robustezza e la capacità di rete, ma anche di aumentare le velocità di trasferimento e introdurre nuove funzionalità. Il Bluetooth, giunto oggi alla versione 5.3 [2], deve la sua grande diffusione all'introduzione nel 2010 all'interno del core specification 4.0 [18] di una tipologia di Bluetooth progettata per essere uno standard radio con un consumo di energia il più basso possibile, un costo contenuto, una ristretta ampiezza di banda ed una ridotta complessità, il Bluetooth Low Energy (BLE).

Le sue caratteristiche, migliorate nel tempo, gli hanno permesso di affermarsi in settori applicativi fino ad allora inesplorati, come smart home e entertainment [19], healthcare [20], accessori per ufficio, automotive [21], dispositivi indossabili [22], sport e fitness[], ma soprattutto avvicinarsi all'Internet of Things [23, 24, 25, 26]. Tuttavia, nel corso dello sviluppo delle applicazioni di IoT, la propensione ad accrescere sempre più il numero di dispositivi nelle reti e l'esigenza di incrementare la copertura e l'interconnessione, hanno spostato l'attenzione su soluzioni non più basate sulle tradizionali topologie di rete centralizzate (stella) o ad albero, tipiche del Bluetooth, ma su tecnologie in grado di garantire una topologia mesh.

Prima di luglio 2017, la tecnologia Bluetooth era carente sotto questo punto di vista, soprattutto se comparata ad altre tecnologie come ZigBee, Z-Wave che hanno integrato fin dalla loro nascita la possibilità di sfruttare i vantaggi di una topologia mesh. Diverse sono state le proposte e gli studi effettuati a riguardo per sopperire a questa carenza. L'articolo [27] raccoglie e compara quanto proposto negli anni. Tuttavia nessuno degli schemi mesh proprietari ha avuto successo dal punto di vista commerciale, principalmente a causa dell'impossibilità di garantire quel principio fondamentale di interoperabilità che caratterizza una tecnologia di rete standardizzata [28]. Il punto di svolta per la Bluetooth Special Interest Group (SIG) si ha per l'appunto nel 2017, anno in cui viene presentata la specifica Bluetooth Mesh. Una tecnologia di rete basata sul BLE, di cui sfrutta alcuni concetti e funzionalità [5], e che consente finalmente la creazione di reti di dispositivi su larga scala con comunicazioni many-to-many.

Il Bluetooth mesh è stato sviluppato per superare le limitazioni delle reti precedenti di Bluetooth, come il numero limitato di dispositivi che potevano essere connessi contem-

poraneamente e la portata limitata dei segnali. Lo stato attuale dell'arte del Bluetooth mesh è molto promettente. La tecnologia sta rapidamente guadagnando terreno nel settore IoT, con un numero sempre maggiore di produttori che adottano il Bluetooth mesh per le loro soluzioni IoT. La crescente popolarità del Bluetooth mesh è dovuta alla sua semplicità d'uso, alla sua affidabilità e alla sua capacità di supportare grandi reti di dispositivi. Il Bluetooth mesh è molto versatile e può essere utilizzato in molte applicazioni diverse, tra cui la domotica, la sicurezza, in ambito illuminotecnico e nella gestione delle risorse [29, 30, 8, 31, 32]. La sua capacità di supportare grandi reti di dispositivi lo rende particolarmente adatto alle applicazioni che richiedono la gestione di molti sensori e dispositivi su vasta scala. Inoltre, il Bluetooth mesh è compatibile con molte piattaforme e sistemi operativi, tra cui iOS, Android e Windows, rendendo molto semplice la creazione di soluzioni IoT basate su questa tecnologia.

I sistemi di illuminazione commerciale e industriale sono ottimi esempi di scenari di implementazione in cui la rete Bluetooth mesh è una soluzione perfetta. Includendo le funzionalità specifiche per i sistemi di illuminazione e le reti per smart-home [33], il sistema di rete wireless Bluetooth mesh ha dimostrato di essere una soluzione flessibile e a basso costo adatta per le funzionalità di illuminazione integrate. Tutte le applicazioni mesh garantiscono l'interoperabilità tra prodotti di diversi fornitori e i comportamenti sono implementati utilizzando modelli standard [34], definiti dal Bluetooth SIG per soddisfare le esigenze specifiche di un prodotto.

Ulteriori esempi di applicazioni IoT in cui è possibile impiegare il Bluetooth mesh, e su cui sono stati pubblicati studi per valutarne la fattibilità e le eventuali carenze sono:

- *Smart Building.*
 - sistemi di localizzazione indoor [35, 36, 37]
 - sistemi di citofonia intelligenti [38]
 - integrazioni di funzionalità aggiuntive per lo sviluppo di smart-home [39, 40, 41, 42, 43]
- *Smart Lightning:*
 - in ambiente indoor (abitazione e ufficio) [44, 29]
 - in ambiente outdoor per lo sviluppo di semafori urbani intelligenti [45]
- *Sistemi di Monitoraggio* [36, 46, 47, 48]
 - osservazione del comportamento umano
 - controllo delle condizioni ambientali
 - controllo dello stato di salute di apparecchiature critiche tramite sensori
- *Comunicazioni di emergenza:*

- alternativa in caso di mancanza di connettività Internet quando l'infrastruttura di comunicazione si interrompe in scenari di disastro come proposto in [49].
- *Smart factory*
 - comunicazioni uomo-macchina e viceversa [50, 51]
- *Smart Parking*
 - implementare sistemi per l'emissione di biglietti senza infrastrutture e la gestione dei parcheggi [52, 53], con l'ausilio di beacon BLE.

Quelle elencate sopra sono solo alcune delle possibili applicazioni del Bluetooth mesh in ambito IoT.

1.2 Stato dell'arte

Negli ultimi anni sono stati condotti alcuni studi per analizzare e valutare le caratteristiche e le prestazioni delle reti Bluetooth mesh, per evidenziare alcuni limiti implementativi e per proporre ottimizzazioni dello standard.

Junji Yin et al. in [8] esaminano il Bluetooth mesh da una prospettiva globale, andando ad evidenziare come in futuro questa tecnologia sarà un forte concorrente per la fornitura di soluzioni complete capaci di soddisfare le esigenze di comunicazione senza soluzione di continuità nell'area dell'Internet of Things. Altri autori approfondiscono lo standard esaminando l'impatto di varie caratteristiche sull'affidabilità e l'efficienza della rete. Le funzionalità sono configurate e controllate a diversi livelli e viene valutato l'impatto delle diverse combinazioni di parametri scelte. Vengono inoltre evidenziate le sfide di configurazione e vengono delineati i possibili miglioramenti [54]. Il paper [55] valuta la tecnologia Bluetooth mesh in termini di efficienza energetica e di potenziale utilizzando un banco di prova sperimentale, un approccio statistico e un modello di simulazione basato su grafici. Lo studio conclude che la tecnologia ha un grande potenziale, ma non è ancora efficiente dal punto di vista energetico perché si basa su una dorsale con nodi senza vincoli energetici.

In [56], basandosi su misure eseguite su una piattaforma hardware reale, gli autori creano un modello analitico sul consumo di potenza, sulla durata e sul costo energetico per bit per i nodi Bluetooth mesh definiti Low Power Node. Gli articoli [57] e [58], presentano invece un approccio applicativo sul tema dell'ottimizzazione energetica dei nodi Bluetooth mesh. Senza modificare la specifica, gli autori propongono due nuove feature per ridurre il consumo energetico complessivo nelle reti Bluetooth mesh. Sebbene il tema del consumo energetico sia cruciale per consentire al Bluetooth mesh di

adattarsi e coprire settori applicativi in cui l'efficienza energetica è un requisito essenziale, altrettanto importanti sono le valutazioni delle prestazioni della rete in termini di affidabilità, scalabilità, sicurezza e performance. A riguardo, Adnan Aijaz et al. in [59] valutano la rete e le sue prestazioni in varie condizioni. Vengono esaminate le modalità di comunicazione unicast e di gruppo, le prestazioni in diversi modelli di traffico, gli effetti della segmentazione dei messaggi e la latenza. Inoltre, il documento esamina i possibili miglioramenti delle prestazioni per le reti Bluetooth mesh utilizzando varie tecniche. I risultati forniscono informazioni sulle prestazioni a livello di sistema del Bluetooth mesh. Altri autori hanno studiato l'uso di soluzioni Bluetooth mesh per il monitoraggio congiunto di spazi interni e persone. Hanno scoperto che le prestazioni di rete e localizzazione della rete mesh BLE sono più influenzate dalla distanza in termini di salti (hop) tra i diversi nodi della rete che dal carico di traffico generato dal nodo sorgente [36]. In [46], gli autori esaminano sperimentalmente le prestazioni in termini di Packet Delivery Ratio del Bluetooth Low Energy (BLE) in una rete mesh indagando, in particolare, i limiti della tecnologia in termini di capacità di consegna dei dati nelle applicazioni di monitoraggio. I risultati mostrano che i nodi relay impongono limitazioni critiche per la consegna dei messaggi nelle reti multi-hop, limitando l'utilizzo della tecnologia BLM per molte applicazioni di monitoraggio. Altri autori discutono dei modi per migliorare la rete mesh Bluetooth con nuove funzionalità di advertising avanzate. Una proposta si basa sul protocollo attuale, mentre l'altra apporta modifiche minime allo standard. Entrambe le proposte migliorano il throughput e il ritardo dell'attuale rete mesh Bluetooth [60].

La maggior parte degli articoli citati presentano risultati di studi sperimentali condotti in ambienti reali, utilizzando dispositivi reali su reti Bluetooth Mesh di piccole dimensioni, costituite da pochi nodi e topologie semplici. Tuttavia, tali test sperimentali presentano un alto grado di variabilità dovuto ad elementi esterni e un basso indice di ripetibilità, che rendono difficoltosa una valutazione accurata dell'impatto delle diverse configurazioni dei parametri sulla prestazione, in condizioni ambientali paritarie.

In contrasto, lo studio di Rondon et al. (2019), pubblicato nel 2020, utilizza uno strumento di simulazione per analizzare le reti Bluetooth Mesh. Gli autori hanno creato un simulatore personalizzato per analizzare le prestazioni del servizio Bluetooth Mesh, fornendo anche linee guida generali per la sua configurazione e utilizzo. Attraverso estese simulazioni, gli autori analizzano l'effetto della configurazione di diversi parametri del protocollo sulla qualità del servizio, il ritardo, la scalabilità end-to-end e le prestazioni in scenari eterogenei, anche in presenza di interferenze WLAN.

Per determinare con precisione le prestazioni della QoS di una rete mesh Bluetooth, è fondamentale considerare attentamente tutti i parametri di rete e la sua struttura complessiva. Tuttavia, quando si tratta di analizzare reti di grandi dimensioni, l'uso di uno strumento di simulazione diventa essenziale. Creare un ambiente virtuale presenta di-

versi vantaggi, come una maggiore affidabilità dei test, la riduzione dei costi evitando di acquistare nuove attrezzature e la riduzione dei tempi necessari per la configurazione della rete. Inoltre, l'utilizzo di un simulatore consente un migliore controllo della rete e dell'ambiente, escludendo disturbi esterni non controllabili e fornendo un elevato grado di affidabilità in termini di ripetibilità dei test. In questo modo, è possibile valutare solo gli effetti della configurazione dei diversi parametri sul comportamento della rete. Tuttavia, è importante scegliere con cautela uno strumento in grado di emulare il comportamento di un dispositivo BM il più fedelmente possibile.

1.3 Obiettivi della ricerca e del lavoro svolto

L'obiettivo della ricerca svolta è stato lo studio, l'implementazione e l'analisi di reti basate sul recente standard Bluetooth Mesh con particolare attenzione all'analisi ed all'ottimizzazione delle prestazioni sia in ambiente reale che simulato. Inizialmente la ricerca si è focalizzata sullo studio e sullo sviluppo di codici per dispositivi embedded atto ad implementare funzionalità che migliorassero l'affidabilità nella ricezione di messaggi di stato in una rete Bluetooth Mesh. Nella seconda fase della ricerca, l'attenzione si è concentrata sulla creazione di un ambiente di simulazione per le reti Bluetooth Mesh e sullo studio dell'impatto che differenti configurazioni dei diversi parametri di rete hanno sull'affidabilità. Sulla base dei risultati ottenuti è stata poi realizzata e, successivamente ottimizzata, una nuova implementazione dello standard che riguardava, in particolare, la gestione dei messaggi nella fase di scansione e di advertising nei nodi relay.

Capitolo 2

Bluetooth Low Energy and Bluetooth Mesh

Questo capitolo presenta una panoramica sul Bluetooth Low Energy (BLE) e Bluetooth mesh (BM). In esso vengono illustrate le peculiarità di queste tecnologie, come i loro stack protocollari, la topologia di rete, i meccanismi di comunicazione e le caratteristiche generali, con particolare attenzione a quegli aspetti che sono oggetto di ricerca in questa tesi. Viene esaminato inizialmente il BLE, poichè fornisce gli strumenti essenziali per la creazione di processi di comunicazione su cui si basa la tecnologia di rete Bluetooth mesh, e successivamente vengono presentate le caratteristiche salienti dello standard Bluetooth mesh. Lo scopo principale di questo capitolo è quello di fornire gli strumenti necessari per la comprensione dell'attività di ricerca svolta e presentata nei capitoli successivi.

2.1 Bluetooth Low Energy

La tecnologia Bluetooth esiste dagli anni 90. Il suo intento originale era quello di sostituire i cavi seriali utilizzati per la connessione di telefoni cellulari, computer e altri dispositivi elettronici, ma presto si è evoluto in uno standard globale per la comunicazione wireless tra dispositivi. Oggi, quest'idea risulta ancora essere il fulcro delle sue applicazioni, e il suo sviluppo gli ha permesso di diventare uno dei modi più comuni per connettere dispositivi come cuffie, altoparlanti, mouse, tastiere, sensori e molto altro ancora. È supportato da molte aziende di tecnologia leader e incorporato in molti dispositivi mobili, tra cui smartphone e tablet.

Il Bluetooth è una tecnologia di comunicazione wireless a corto raggio che opera nella banda di frequenza non licenziata dei 2,4GHz ISM (Industrial, Scientific and Medical) per trasmettere dati e costruire delle wireless personal area network (WPAN).

Sicuramente la maggior parte delle persone conosce cos'è il Bluetooth, o almeno ad alto livello, ma quello che c'è dietro, la complessità dello standard tecnologico è

gestito da un'organizzazione no-profit fondata nel 1998 e conosciuta come Bluetooth Special Interest Group (SIG). Composta da membri di tutto il mondo, tra cui produttori di dispositivi, sviluppatori, operatori di reti e altri rappresentanti dell'industria, quest'organizzazione ha come obiettivi principali quello di rendere sempre più conveniente e accessibile la tecnologia Bluetooth e supportare gli sviluppatori e i produttori nella creazione di prodotti che utilizzano questa tecnologia.

La Bluetooth SIG sviluppa continuamente il protocollo Bluetooth, adattandolo alle esigenze mutevoli del mercato e delle applicazioni e, nel corso degli anni ha rilasciato differenti versioni delle specifiche. Tra i protocolli più popolari abbiamo il Bluetooth Classic e il Bluetooth Low Energy (BLE).

In particolare il BLE, è stato introdotto a partire dal 2010 con il rilascio da parte della Bluetooth SIG della versione v4.0 del Bluetooth Core Specification. Questa nuova tipologia di protocollo, non va a sostituire il suo predecessore (Bluetooth Classic), ma lo affianca come alternativa con capacità e qualità che lo rendono perfetto per una nuova generazione di prodotti.

Nonostante il Bluetooth Classic e il BLE condividono diversi aspetti, sono due protocolli distinti e incompatibili. Bluetooth Classic è progettato per gestire discrete quantità di dati, ma a discapito di un notevole consumo di energia. Un tipico impiego è il trasferimento di musica dal proprio telefono ad una cassa audio Bluetooth. Contrariamente, il BLE è progettato per essere molto efficiente dal punto di vista energetico, permettendo ai dispositivi di funzionare per mesi o addirittura anni con una sola batteria, ma a scapito del trasferimento dati sia da un punto di vista della copertura radio che velocità di trasferimento. Il BLE è in grado di trasmettere dati a velocità inferiori rispetto al Bluetooth tradizionale, ma consuma molta meno energia e ha una portata più ridotta. Questo lo rende ideale per le applicazioni che richiedono una connessione affidabile e a basso consumo energetico a corto raggio, come la trasmissione di dati da sensori a un dispositivo mobile.

2.1.1 Caratteristiche chiave del BLE

Il Bluetooth Low Energy è stato progettato come tecnologia complementare al suo predecessore Bluetooth Classic. BLE mira al minor consumo energetico possibile lavorando su diversi aspetti per raggiungere le sue caratteristiche principali: robustezza, basso consumo e basso costo. Di seguito alcune delle caratteristiche principali e una breve descrizione su come vengono ottenute:

- **Basso consumo di potenza:** il BLE si concentra sulla riduzione al minimo della potenza Tx/Rx nelle comunicazioni attraverso diversi meccanismi come la riduzione della struttura del pacchetto, tempo di trasmissione e riconessioni bre-

vi, gestione ottimizzata dei tempi di inattività del transiever, riduzione del quantitativo di dati trasmessi e throughput.

- **Robustezza:** la banda a 2,4 GHz è già sovraffollata quindi, per coesistere con altre tecnologie radio, sono stati implementati diversi meccanismi per ottenere robustezza: la verifica dell'intergrità dei pacchetti attraverso l'utilizzo di un controllo degli errori di ridondanza ciclica (CRC); tecniche di trasmissione di tipo 'Adaptive Frequency-hopping Spread Spectrum' (AFH) per cercare di minimizzare l'interferenza causata da altri segnali radio operanti nella stessa banda ISM; meccanismi di acknowledgment e ritrasmissione dei pacchetti; nelle versioni più recenti del BLE anche nell'introduzione di meccanismi per la correzione degli errori lato ricevente (Forward Error Correction).
- **Basso Costo :** l'utilizzo di una banda di frequenze senza licenza come le ISM e la politica di accesso alle specifiche della Bluetooth SIG facilità le aziende e le startup nella creazione e produzione di propri prototipi Bluetooth. A differenza di altri protocolli e tecnologie wireless, in cui le documentazioni sono disponibili solo tramite il pagamento di ingenti royalty, le specifiche Bluetooth sono tutte accessibili gratuitamente dal sito Bluetooth.com. Inoltre, la consueta riduzione delle dimensioni, l'architettura semplice, la bassa potenza trasmissiva, la bassa manutenzione e la lunga durata dell'hardware che derivano dal design della tecnologia consentono una produzione massiva riducendo i costi al dettaglio.
- **Enorme diffusione negli smartphone:** uno dei più grandi vantaggi del BLE rispetto a ZigBee, Z-Wave e Thread è che la maggior parte delle persone possiede uno smartphone che dispone di hardware BLE, offrendo agli sviluppatori e produttori di accessori una base di potenziali utenti molto più ampia rispetto ai competitor.

2.1.2 Stack overview

Lo stack Bluetooth Low Energy (BLE) è costituito da una serie di livelli e moduli funzionali, alcuni dei quali sono obbligatori e altri facoltativi. Queste componenti sono poi organizzate in due blocchi distinti noti come *Host* e *Controller* e un'interfaccia logica standard, l'*Host Controller Interface*, che definisce i modi con il quale questi due contenitori logici possono comunicare. La Fig. 2.1 mostra lo stack BLE, i diversi layer e la loro distribuzione tra Host e Controller.

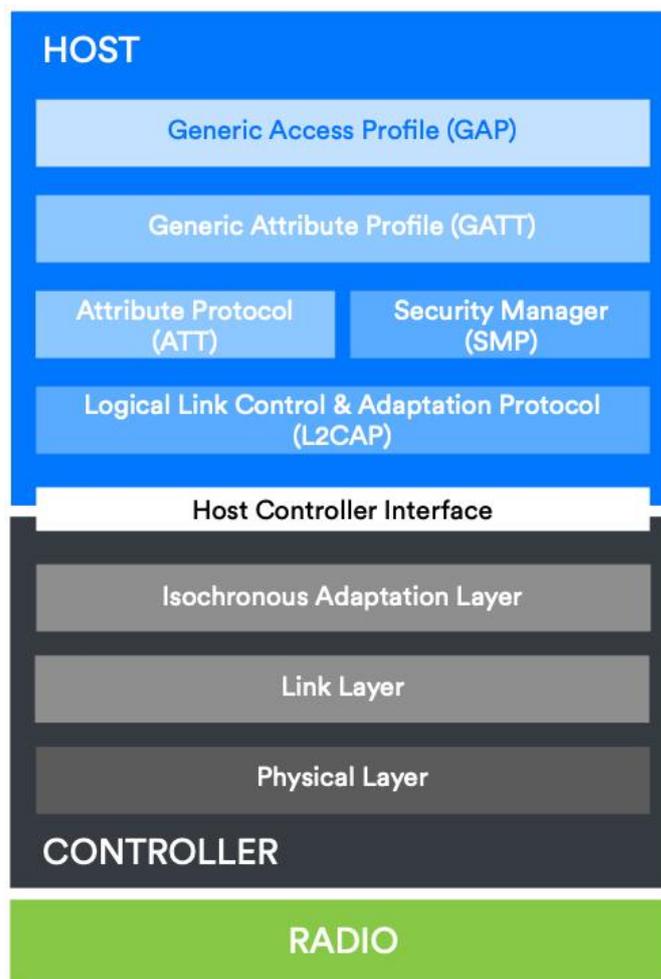


Figura 2.1: Stack Bluetooth Low Energy. (Fonte: Bluetooth SIG)

Controller

Il controller è il componente hardware che si occupa di gestire le funzioni di basso livello del Bluetooth, come la modulazione e la decodifica del segnale radio, è responsabile della gestione della connessione con altri dispositivi Bluetooth e del mantenimento della sicurezza della connessione.

1. **Physical Layer** è il livello più basso dello stack BLE ed è responsabile della comunicazione radio vera e propria. Definisce tutti quegli aspetti della tecnologia legati all'uso del transceiver (RF), inclusi schemi di modulazione, bande di frequenza, uso del canale, caratteristiche del trasmettitore e ricevitore e potenza trasmessa.

Il BLE opera nella banda di frequenza dei 2,4 GHz ISM nel range tra 2400 MHz e 2483,5 MHz. Lo standard prevede una suddivisione di tale intervallo in 40

canali, ognuno con una larghezza di 2 MHz. Il modo in cui questi canali vengono utilizzati è definito dal Link Layer.

Per codificare i dati digitali provenienti dai livelli superiori dello stack prima della trasmissione e per decodificare i segnali radio ricevuti, il BLE utilizza uno schema di modulazione chiamato *Gaussian Frequency Shift Keying (GFSK)* differenziato sulla base della velocità di trasmissione del Physical Layer espressa in *simboli per secondo*:

- *LE 1M PHY*, con una frequenza di simbolo di 1 Msym/s. L'implementazione di questa variante è obbligatorio in ogni dispositivo BLE.
- *LE 2M PHY*, è simile all'LE 1M ma usa una frequenza di simbolo di 2 Msym/s. Il supporto per LE 2M PHY è opzionale.
- *LE Coded PHY*, utilizza una frequenza di simbolo di 1 Msym/s, ma i pacchetti sono soggetti ad una codifica definita *Forward Error Correction (FEC)* che è definita nel Link Layer. La FEC aumenta la portata effettiva delle trasmissioni ma riduce la velocità dei dati dell'applicazione. Il supporto per LE Coded PHY è facoltativo.

2. **Link Layer** Il Link Layer è una delle più importanti e complesse sezioni della Bluetooth Core Specification. E' responsabile di molte funzioni, come la definizione dei diversi tipi di pacchetti trasmessi via radio e il protocollo di interfaccia aerea associato, ma anche della selezione e classificazione del canale radio per la trasmissione e ricezione. Funziona seguendo una macchina a stati ben definita e, a seconda dello stato, può operare in modi diversi guidato da eventi di vario tipo. Il Link Layer supporta comunicazioni di tipo *connection-oriented* e *connection-less*. Può gestire sia la comunicazione tra due dispositivi (*point-to-point*) che la comunicazione da un dispositivo a molti *one-to-many*.

Il suo funzionamento può essere rappresentato sotto forma di macchina a stati mostrata in Fig. 2.2. Il Link Layer può mantenere attivo un solo stato per volta.

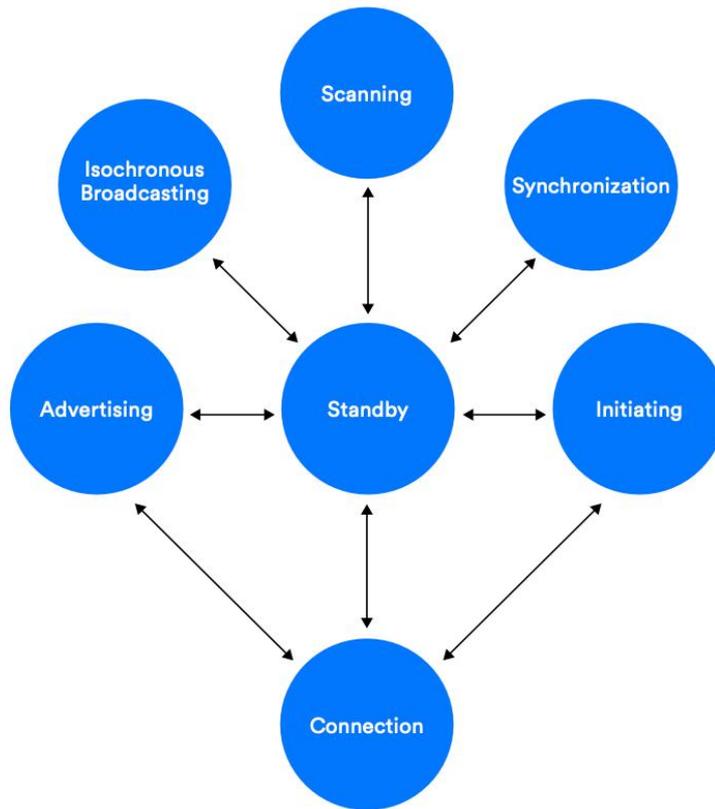


Figura 2.2: Macchina a stati del Link Layer.(Fonte: Bluetooth SIG)

Di seguito una breve descrizione degli stati mostrati in Fig. 2.2

- *Stand-by*: il dispositivo non trasmette né riceve pacchetti.
- *Initiating*: Risponde a pacchetti di advertising da un particolare dispositivo per richiedere di instaurare una connessione.
- *Advertising*: consente al Link Layer di trasmettere pacchetti sui canali di advertising ed eventualmente ascoltare e rispondere a richieste di informazioni provenienti da altri dispositivi. Un dispositivo nello stato di advertising viene definito *advertiser*
- *Connection*: lo stato in cui un dispositivo ha stabilito un collegamento con un altro dispositivo e scambia regolarmente dati con quest'ultimo.
- *Scanning*: stato in cui un dispositivo BLE ascolta pacchetti di advertising provenienti da altri dispositivi BLE al fine di scoprire e stabilire connessioni o semplicemente rilevare le informazioni inviate. Un dispositivo in questo stato viene definito *scanner*.
- *Isochronous Broadcast*: Trasmette pacchetti di dati isocronous.

2.1 Bluetooth Low Energy

- *Synchronization*: Inizia ad ascoltare un "treno" di pacchetti di advertising periodici provenienti da un dispositivo specifico.

Come descritto nel paragrafo del Physical Layer, l'intervallo di frequenze utilizzato dal BLE è suddiviso in 40 canali a radio frequenza (RF) (Fig. 2.3). Il Link Layer si occupa di definire come vengono indicizzati e il loro utilizzo in base alla tipologia di comunicazione BLE. I canali RF sono suddivisi in 3 *Primary Advertising Channel* utilizzati per tutte le attività di advertising, per stabilire connessioni tra due dispositivi e per comunicare informazioni in broadcast tra dispositivi non connessi; sono indicizzati con i numeri 37, 38 e 39. I restanti 37 canali sono definiti *general-purpose channel* e sono impiegati per un uso generale e per la maggior parte delle comunicazioni.

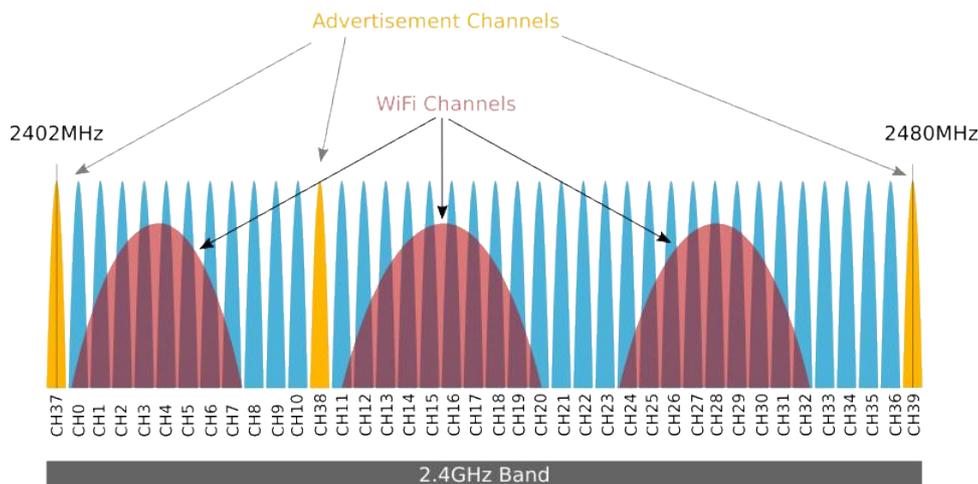


Figura 2.3: BLE Physical Channel. (Fonte: [1])

Per quanto riguarda i Primary Advertising Channel, la scelta della loro posizione all'interno della banda considerata tiene conto della coesistenza del BLE con altre tecnologie (es. Wi-Fi, IEEE 802.11). A questa tipologia di canali sono state assegnate frequenze che riducono al minimo le potenziali sovrapposizioni. Nella Fig. 2.3 sono indicati in giallo.

La specifica del Link Layer si occupa di definire anche il formato dei pacchetti e per facilità d'uso, il Bluetooth ne definisce due tipologie. Il primo è utilizzato nelle trasmissioni che impiegano gli Uncoded PHY (LE 1M PHY e LE 2M PHY), mostrato in Fig. 2.4. Mentre il secondo tipo è associato all'utilizzo del Coded PHY, Fig. 2.5

Entrambe le tipologie includono quattro campi obbligatori: un *Preamble*, un *Access Address*, un *Protocol Data Unit (PDU)* e un *Cyclic Redundancy Check*

(CRC). Mentre il *Constant Tone Extension* è presente solo quando è in uso la feature per il Direction Finding (Angolo di arrivo o Angolo di partenza, Bluetooth v5.1).

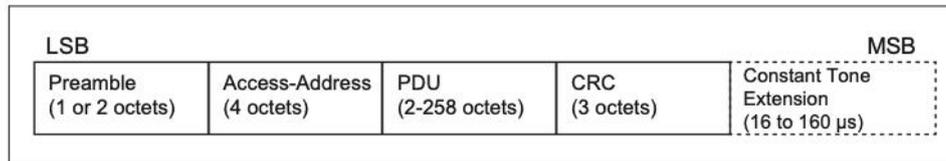


Figura 2.4: Formato dei pacchetti del Link Layer per i *LE Uncoded PHY*. (Fonte: [2])

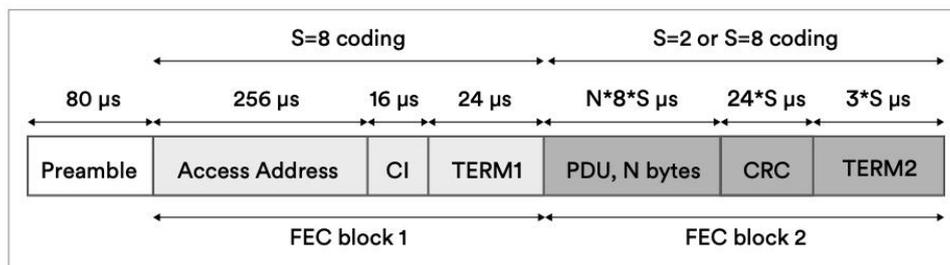


Figura 2.5: Formato dei pacchetti del Link Layer per i *LE Coded PHY*. (Fonte: [2])

- *Preamble*: consente al ricevitore di sincronizzarsi con precisione sulla frequenza del segnale, eseguire il controllo automatico del guadagno e stimare la temporizzazione del simbolo.
- *Access Address*: utilizzato dai ricevitori per differenziare i segnali dal rumore di fondo e per determinare la rilevanza o meno di un pacchetto per il dispositivo ricevente.
- *PDU*: è un campo variabile che cambia a seconda del tipo di pacchetto che si sta trasmettendo e del particolare dato che il livello applicativo vuole trasmettere.
- *CRC* utilizzato per il rilevamento degli errori.

Le tipologie di PDU presenti nella specifica BLE sono differenziate in base alla loro funzione e struttura e possono essere classificate in tre categorie principali:

- *Advertising physical channel PDU*: Gli Advertising Physical Channel Protocol Data Unit (PDU) sono utilizzati per trasmettere dati in applicazioni che non richiedono una connessione completa o durante le fasi preliminari di instaurazione di una connessione. Nonostante il nome suggerisca il contrario, questa categoria di PDU contiene diverse sottocategorie, tra cui l'Advertising PDU, lo Scanning PDU e l'Initiating PDU.

- *Data physical channel PDU*: questi pacchetti sono utilizzati per la trasmissione di dati effettivi tra i dispositivi connessi. Essi costituiscono il contenuto principale delle comunicazioni BLE e sono utilizzati per trasmettere informazioni come i dati dell'applicazione, le richieste di servizio e le risposte ai comandi.
- *Isochronous physical channel PDU*: questi pacchetti sono utilizzati nella trasmissione di flussi di dati che richiedono una elevata precisione temporale, come ad esempio l'audio in streaming (LE Audio).

In generale, la specifica BLE prevede una varietà di PDU con funzioni e caratteristiche specifiche, progettate per soddisfare le esigenze di diverse applicazioni e scenari di utilizzo. La comprensione di queste tipologie di PDU è importante per poter utilizzare la tecnologia BLE in modo efficace ed adattarla alle esigenze specifiche.

In riferimento all'attività di ricerca, nel seguito sarà illustrata la sottocategoria *Advertising PDUs*, con particolare attenzione al PDU di tipo

ADV_NONCONN_IND in quanto utilizzato dallo standard Bluetooth mesh.

3. **Isochronous Adaptation Layer** : Lo scopo di questo layer è principalmente quello di affrontare un potenziale problema che può influenzare sia le comunicazioni isocrone connesse che quelle trasmesse attraverso dispositivi audio. Consente l'utilizzo di frame di differente durata da parte dei dispositivi che utilizzano la comunicazione isocrona. In questo modo la temporizzazione usata per generare o ricevere dati isocroni nello strato superiore può essere indipendente dalla temporizzazione usata nel trasporto logico Connected Isochronous Stream (CIS) o Broadcast Isochronous Stream (BIS) usato per portare i dati isocroni.

Host

1. **Logical Link Control and Adaptation Protocol (L2CAP)** funge da livello di moltiplicazione di protocollo. È preso in prestito dallo standard Bluetooth Classic e nel caso di BLE svolge le seguenti attività:
 - Prende più protocolli dai livelli superiori e li inserisce in pacchetti BLE standard che vengono trasmessi ai livelli inferiori sottostanti.
 - I pacchetti provenienti dai livelli superiori dello stack non sempre rientrano all'interno delle dimensioni massime del payload BLE supportato per la trasmissione e, per questo motivo necessitano di essere frazionati. Il L2CAP gestisce la frammentazione lato trasmettitore e la ricombinazione lato ricevitore di questi pacchetti.

2. **Attribute Protocol (ATT)** funziona come un protocollo client/server ed è utilizzato nel BLE per la gestione dei dati. Definisce come i dati sono rappresentati in un database del *server* BLE e i metodi con cui tali dati possono essere letti o scritti da un *client*. Gli attributi possiedono le seguenti proprietà:
 - *Attribute type*: identificato da un Universally Unique Identifier or UUID.
 - *Attribute handle*: un valore di 16 bit che il server assegna ad ognuno dei suoi attributi. Il client utilizza questo valore per identificare univocamente l'attributo durante l'intera durata della connessione tra due dispositivi.
 - *Attribute Permission*: una combinazione di permessi di accesso (lettura, scrittura o entrambi), crittografia, autenticazione e autorizzazione per mostrare l'attributo e il dato contenuto in esso.

3. **Security Manager Protocol (SMP)**: è un protocollo di sicurezza utilizzato per proteggere la comunicazione tra dispositivi. Fornisce una serie di funzioni di sicurezza per garantire la protezione dei dati sensibili scambiati tra dispositivi BLE, come ad esempio le informazioni sulle credenziali di accesso. Il SMP supporta diversi livelli di sicurezza, tra cui la crittografia a chiave pubblica, l'autenticazione basata su password e la sicurezza basata su certificati. Inoltre, il SMP supporta la generazione automatica di chiavi crittografiche e la gestione delle chiavi condivise tra i dispositivi.

4. **Generic Access Profile (GAP)** fornisce un framework che definisce i modi, le procedure e i ruoli necessari per l'interoperabilità tra dispositivi BLE. In particolare, oltre ai diversi ruoli che verranno illustrati tra poco, il GAP definisce come i dispositivi rilevano gli altri dispositivi e come essi stessi vengono rilevati, le procedure di connessione, l'utilizzo dei diversi livelli di sicurezza e la definizione dei parametri nelle fasi di advertising, scanning e connection. L'implementazione di questo framework all'interno di ciascun dispositivo BLE è obbligatorio.

Esistono quattro ruoli GAP principali in cui opera un dispositivo BLE:

- *Central*: dispositivo in grado di stabilire e mantenere connessioni con dispositivi peer. È un ruolo spesso assunto da un tablet o uno smartphone, in grado di connettersi a più periferiche contemporaneamente.
- *Peripheral*: dispositivo che annuncia la propria presenza inviando pacchetti di advertising e accetta una connessione da un altro dispositivo BLE.
- *Broadcaster*: un dispositivo che trasmette pacchetti di advertising senza accettare connessioni in ingresso. Un dispositivo che implementa questo ruolo utilizza l'*advertising state* del Link Layer.

- *Observer*: un dispositivo che scansiona passivamente i canali e processa i pacchetti ricevuti da dispositivi periferici e broadcaster, ma senza la capacità di instaurare una connessione o inviare richieste per la trasmissione di informazioni aggiuntive ai dispositivi trasmettenti. Un observer utilizza lo *scanning state* del Link Layer.

Una Peripheral o un Broadcaster iniziano le comunicazioni sempre in modalità advertising. In effetti, i pacchetti di advertising sono l'unico modo che consente a un Observer o un Central di rilevare una Peripheral o un Broadcaster.

5. **Generic Attribute Profile (GATT)**: è un importante componente del Bluetooth Low Energy (BLE), che funge da interfaccia tra i dispositivi BLE e le applicazioni che devono scambiare dati. Il GATT è uno standard che descrive come i dati devono essere strutturati e scambiati tra i dispositivi BLE. Ad esempio, se si desidera che un dispositivo BLE invii la sua temperatura ad un altro dispositivo, il GATT definisce come questa informazione deve essere strutturata e inviata come attributo. Questo rende possibile lo scambio affidabile e sicuro di dati tra i dispositivi BLE, indipendentemente dal fatto che siano prodotti da diversi produttori. Il Generic Attribute Profile utilizza tre elementi fondamentali per la rappresentazione e lo scambio dei dati: *services*, *characteristics* e *descriptor*.

- *Services*: un servizio è un insieme di caratteristiche che rappresentano una particolare funzione o un particolare tipo di dati che un dispositivo BLE è in grado di fornire
- *Characteristics*: una caratteristica rappresenta un singolo valore o un insieme di valori associati a una particolare funzione o a un particolare tipo di dati.
- *Descriptor*: un descrittore è un valore che fornisce ulteriori informazioni su una caratteristica, come ad esempio il formato o l'unità di misura del valore associato alla caratteristica.

Il GATT definisce due ruoli:

- *client GATT*: invia comandi e richieste ATT al server GATT.
- *server GATT*: accetta ed elabora i comandi e le richieste ricevute da un client GATT e invia al client GATT notifiche, indicazioni e risposte ATT.

2.1.3 Advertising

Che si tratti di un beacon (che trasmette posizione, dati meteo o altri dati) o di un dispositivo indossabile che stabilisce una connessione a lungo termine con un host (tablet

Capitolo 2 Bluetooth Low Energy and Bluetooth Mesh

o telefono), tutti i dispositivi periferici che sfruttano la tecnologia Bluetooth, almeno inizialmente, iniziano in modalità *advertising*.

L'*advertising* fornisce una modalità di comunicazione senza connessione e come accennato in precedenza, può essere utilizzato per:

- indicare la disponibilità di un dispositivo ad instaurare una connessione con comunicazione bidirezionale (ad esempio tra uno smartwatch e uno smartphone)
- inviare informazioni senza mai connettersi ad un altro dispositivo (ad esempio un beacon che trasmette informazioni in un museo, o in un negozio)

In generale, i pacchetti di *advertising* possono essere ricevuti da qualsiasi dispositivo Bluetooth in scansione (*scanner*) all'interno del raggio d'azione del dispositivo trasmettente. Tuttavia, esistono tipi specifici di *advertising*, chiamati *Directed Advertising*, che consentono di limitare la ricezione dei dati solo a dispositivi specifici, sulla base del loro indirizzo.

Gli *advertising* possono essere utilizzati non solo per una comunicazione *one-to-one*, ma anche per trasferire contemporaneamente dati a più dispositivi di scansione all'interno di una topologia *one-to-many*.

Questa modalità di comunicazione rappresenta uno degli elementi fondamentali del protocollo Bluetooth mesh, in quanto consente a ogni dispositivo di effettuare trasmissioni *one-to-many* e di inoltrare pacchetti ricevuti da altri dispositivi. Grazie a queste funzionalità aggiuntive, è possibile creare una topologia *many-to-many*, ovvero una rete Bluetooth *mesh*, in cui tutti i dispositivi partecipano alla comunicazione.

Advertising PDU

Gli Advertising PDUs comprendono diverse tipologie di PDU che possono essere suddivise in due gruppi:

- i *Legacy Advertising PDUs*, composti da ADV_IND, ADV_DIRECT_IND, ADV_SCAN_IND, ADV_IND, e ADV_NONCONN_IND. Questi PDU sono supportati da tutte le versioni del protocollo Bluetooth.
- gli *Extended Advertising PDUs*, costituiti da ADV_EXT_IND, AUX_ADV_IND, AUX_SYNC_IND e AUX_CHAIN_IND. Richiedono l'utilizzo di una versione del protocollo Bluetooth 5.0 o superiore.

Nel seguito della nostra trattazione, ci concentreremo solamente sui Legacy Advertising PDU.

Il pacchetto per le operazioni di advertising nel protocollo Bluetooth assume la struttura descritta nella Fig. 2.6 e costituito da un *Header* a 16 bit e un *Payload* variabile a seconda del tipo di PDU e lunghezza specificati rispettivamente, nel campo *PDU type* e *Length* dell'Header.

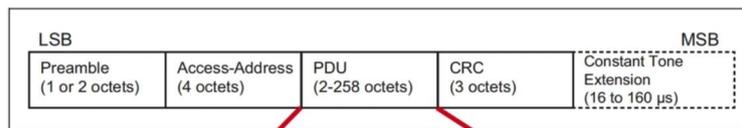


Figure 2.1: Link Layer packet format for the LE Uncoded PHYs

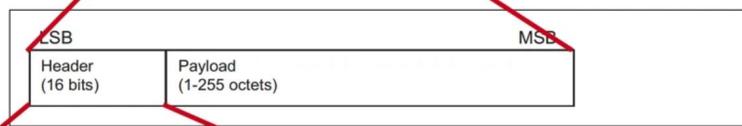


Figure 2.4: Advertising physical channel PDU

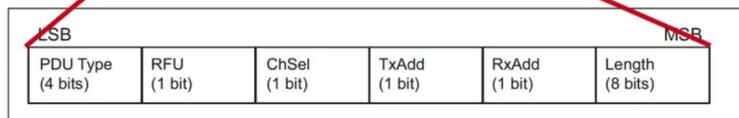


Figure 2.5: Advertising physical channel PDU header

Figura 2.6: Advertising Physical Channel PDU. (Fonte: [2])

Nel caso specifico dei Legacy Advertising la lunghezza del Payload è limitato a 37 ottetti, di cui 6 contengono l'Advertiser's Address(AdvA) e i restanti 31 sono riservati all'Advertising Data.

Tutte le trasmissioni dei pacchetti Legacy Advertising PDU possono avvenire solo attraverso l'impiego dei Primary Advertising Channel (37, 38, 39) e del solo LE 1M PHY. La scelta dell'indice del canale e del loro ordine è a discrezione dello sviluppatore. Tuttavia, all'interno di ciascun canale è possibile trasmettere un solo PDU alla volta.

Indipendentemente dal tipo di Legacy Advertising PDU, la trasmissione di uno o più pacchetti di advertising deve avvenire all'interno di una finestra temporale definita *Advertisign Event* e attraverso i soli Primary Advertising Channel (37, 38, 39). Su ogni indice di canale di advertising utilizzato deve essere inviata al massimo un PDU. La pianificazione degli Advertising Event Fig. 2.7 è gestita attraverso parametri temporali, che vengono regolati in modo tale da ridurre le collisioni con altri dispositivi di advertising. In particolare sono presenti: l'*Advertising Interval (advInterval)* e l'*Advertising Delay (advDelay)*.

Il valore dell'*advInterval* deve essere un multiplo intero di 0.625 ms nel range tra 20 ms e 10,485.759375 s, mentre l'*advDelay* è un valore pseudo-random compreso tra

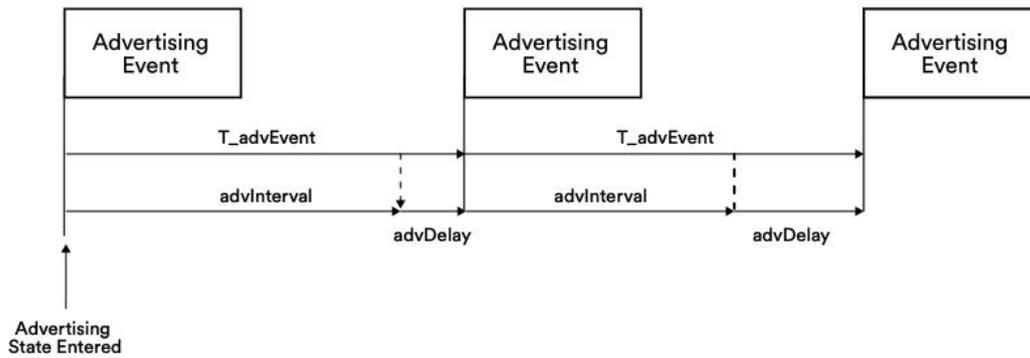


Figura 2.7: Advertising Event perturbati nel tempo utilizzando l'advDelay. (Fonte: [2])

0 e 10 ms generato dal Link Layer per ogni Advertising Event. La somma dei due parametri definisce il $T_{advEvent}$ cioè il tempo tra l'inizio di due Advertising Event consecutivi appartenenti allo stesso advertising set (v5.3 | Vol 6, Part B sez. 4.4.2.10)

ADV_NONCONN_IND

L'ADV_NONCONN_IND è un Legacy Advertising PDU di tipo "non-connectable and non-scannable undirected", il che significa che non accetta richieste di connessione (non-connectable), nè richieste di invio di informazioni aggiuntive (non-scannable) e può essere rilevato da qualsiasi scanner (undirected).

Tipicamente, l'ADV_NONCONN_IND viene utilizzato in applicazioni in cui si vuole trasmettere informazioni in broadcast senza la necessità di stabilire una connessione tra i dispositivi. Un esempio di questo utilizzo è rappresentato dai beacon che diffondono informazioni sul loro contenuto dati e sulla loro presenza e posizione.

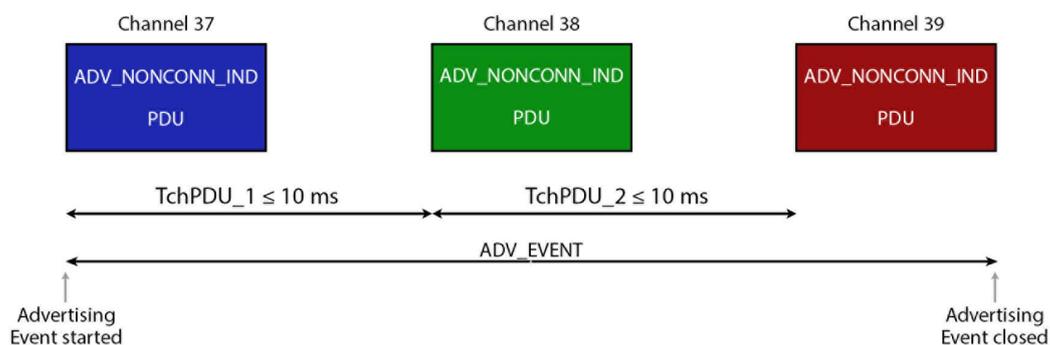


Figura 2.8: Advertising Event Non-connectable and non-scannable undirected che utilizza l'ADV_NONCONN_IND PDU

Nell'ambito di un Advertising Event di tipo Non-connectable and non-scannable undirected, i PDU devono essere trasmessi in modo sequenziale solo nei 3 Primary Ad-

vertising Channel o in un sottoinsieme di essi. Al termine della trasmissione, l'advertiser deve chiudere la finestra temporale dell'Advertising Event entro il tempo definito dall'advertising interval.

Inoltre, il tempo che intercorre tra l'inizio della trasmissione di due ADV_NONCONN_IND consecutivi (TchPDU) deve essere inferiore o uguale a 10 ms. Le scelte riguardanti la precisione, casualità, intervallo di valori e altri aspetti relativi a questo parametro sono a discrezione dello sviluppatore e saranno oggetto di analisi nel proseguo della tesi.

2.1.4 Scanning

Lo scanning è il processo attraverso il quale un dispositivo BLE è in grado di ricevere informazioni da parte di un dispositivo che effettua operazioni di advertising.

Nel Bluetooth Low Energy sono supportate due modalità di scansione:

- *attiva*: il dispositivo che ascolta, una volta ricevuta l'advertisement, può inviare degli specifici PDU per richiedere ulteriori informazioni o la volontà di instaurare una connessione.
- *passiva* il dispositivo ascolta passivamente i pacchetti di advertising e non invia alcuna richiesta all'advertiser.

I parametri principali che regolano il processo di scanning sono:

- *scanWindow* : indica il tempo durante il quale il dispositivo BLE è in ascolto su un determinato canale di advertising per cercare eventuali dispositivi advertiser che trasmettono informazioni sulla loro presenza. La durata di scanWindow è tipicamente breve, nell'ordine di pochi millisecondi.
- *scanInterval* rappresenta l'intervallo di tempo tra il termine di una finestra di scansione (scanWindow) e l'inizio della successiva. In altre parole, indica la frequenza con cui il dispositivo BLE ripete il processo di scansione per cercare dispositivi vicini. La durata di scanInterval è tipicamente più lunga rispetto a quella di scanWindow.

In ogni scanWindow il Link Layer dovrebbe effettuare la scansione su un diverso indice dei Primary Advertising Channel rispetto a quello utilizzato nella finestra di scansione precedente. Il Link Layer deve utilizzare tutti i Primary Advertising Channel

I parametri di scanWindow e scanInterval devono essere inferiori a 40,96 s. scanWindow deve essere inferiore o uguale a scanInterval. Se i parametri di scanWindow e scanInterval sono impostati allo stesso valore dall'Host, il Link Layer dovrebbe effettuare la scansione in modo continuo.

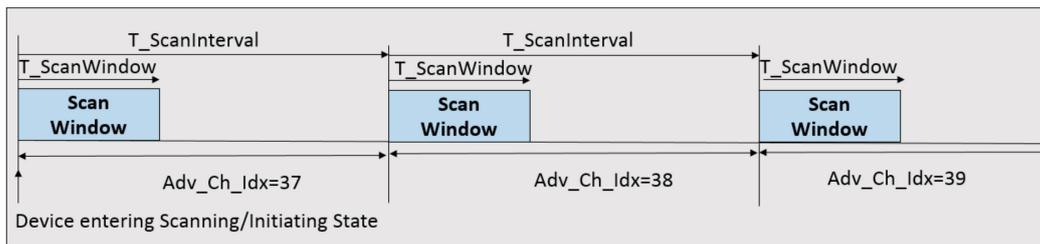


Figura 2.9: Parametri di scansione. (Fonte: [3])

2.2 Bluetooth Mesh

Il Bluetooth Mesh è una tecnologia di rete wireless basata sul Bluetooth Low Energy che consente ai dispositivi Bluetooth di comunicare tra loro in una rete a maglia, in cui ogni elemento della rete può trasmettere e ricevere dati da tutti gli altri dispositivi della rete.

In una rete mesh, i dispositivi non sono collegati direttamente tra loro, ma formano una rete ad hoc in cui i messaggi sono inoltrati tra i vari dispositivi fino a destinazione, creando un percorso dinamico per la trasmissione dei dati. Questo permette la comunicazione tra i dispositivi anche quando non si trovano nelle immediate vicinanze l'uno dell'altro, aumentando la flessibilità e l'affidabilità della rete.

Introdotta per la prima volta nel 2017 dalla Bluetooth Special Interest Group (SIG), il Bluetooth mesh è stato sviluppato per rispondere alle esigenze di una vasta gamma di applicazioni IoT (Internet of Things), in cui è necessario mettere in comunicazione un grande numero di dispositivi, come ad esempio in sistemi di smart-lighting, nella gestione di smart-building, nei sistemi di sicurezza e nelle wireless sensor network.

Attualmente, il Bluetooth Mesh è una tecnologia in evoluzione, con molte aziende che stanno sviluppando prodotti e soluzioni basati su questa tecnologia. La Bluetooth SIG continua a sviluppare la specifica Bluetooth Mesh, introducendo nuove funzionalità e migliorando le prestazioni della tecnologia.

2.2.1 Overview

La comprensione della topologia di rete Bluetooth mesh richiede di conoscere una serie di nuovi termini tecnici e nuovi concetti al mondo BLE.

In primo luogo, è importante distinguere i dispositivi che appartengono alla rete mesh Bluetooth, chiamati *node*, da quelli che hanno le caratteristiche necessarie per implementare lo standard ma non sono ancora stati associati ad una rete mesh Bluetooth, chiamati *unprovisioned node*. Un nodo non provisioned diventa un nodo attraverso un processo chiamato *provisioning* (vedi Fig. 2.10). Questa procedura viene eseguita da nodi specifici o dispositivi esterni.

Il termine "nodo" si riferisce generalmente ad un dispositivo che fa parte della mesh, senza entrare nel merito delle sue caratteristiche interne e funzionalità. Ogni nodo è costituito da almeno una o più entità indipendenti chiamate *elements*. Per esempio, un lampadario (node) può contenere più lampadine (element) che possono essere accese o spente indipendentemente l'una dall'altra. Ogni elemento può avere uno stato che indica la sua condizione (ad esempio, acceso o spento). Ogni elemento necessita di un indicatore della propria condizione, ad esempio nel caso della lampadina se questa è accesa o spenta, quest'informazione è espressa dallo *state*.

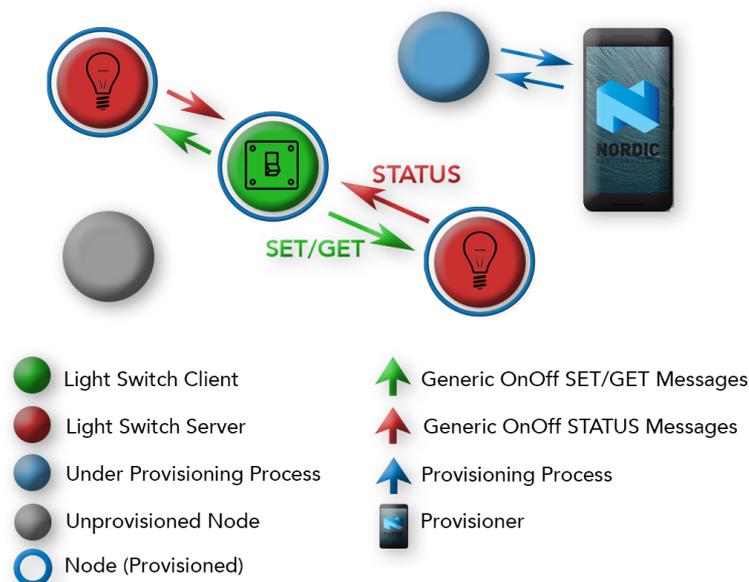


Figura 2.10: Elementi del Bluetooth mesh. (Fonte: [4])

Ogni elemento è indirizzabile tramite un indirizzo *unicast* per identificare in modo univoco chi invia o riceve un messaggio all'interno del nodo. Lo standard definisce anche un'altra categoria di indirizzi, denominata *multicast*, suddivisa in *group virtual*. Entrambi possono etichettare uno o più elementi all'interno di uno o più nodi. La differenza tra le due tipologie sta nel numero di indirizzi che possono essere assegnati. Nel primo caso possono esistere 16384 indirizzi di gruppo differenti divisi in 256 indirizzi di gruppo fissi e 16128 assegnati dinamicamente in base alle esigenze applicative; mentre nel caso degli indirizzi virtuali è possibile indirizzare molti più nodi utilizzando una rappresentazione con un UUID a 128 bit.

La comunicazione tra gli elementi è possibile grazie alla combinazione di questi indirizzi e al paradigma di publish-subscribe. In questo schema, il mittente (*Publisher*) si limita a pubblicare il proprio messaggio sulla rete senza essere a conoscenza della natura del destinatario (*Subscriber*). Il Publisher può dedurre solo la tipologia di trasmissione, se unicast o multicast, a causa della definizione dell'intervallo di valori degli

indirizzi per entrambi i casi. I subscribers, invece, attraverso il meccanismo di sottoscrizione, possono definire a quali indirizzi sottoscrivere e da quali publisher ricevere un dato messaggio. Tuttavia, anche in questo caso, non si conosce la natura dei Publisher o il numero di nodi che pubblicano su un determinato indirizzo.

Le strutture di dati utilizzate all'interno del paradigma di publish-subscribe per richiamare le operazioni tra i nodi sono chiamate *message*. I message sono una delle caratteristiche chiave del Bluetooth mesh, infatti tale tecnologia di rete viene definita message-oriented. I nodi inviano messaggi per controllare o trasmettere informazioni l'un l'altro. All'interno dello standard possiamo identificare tre tipologie di messaggi (Fig. 2.10), ognuno dei quali è definito da un *opcode* (operation code) unico.

- *Get message*: un messaggio per richiedere lo Stato da uno o più nodi.
- *Set message*: un messaggio per modificare il valore di stato di un nodo o di un gruppo di nodi. I messaggi di Set possono essere di tipo acknowledged (ACK) o unacknowledged (UNACK). Un UNACK Set viene trasmesso ad un nodo quando non è richiesta alcuna risposta al comando inviato, contrariamente con un ACK Set viene richiesto al destinatario di rispondere a quanto ricevuto attraverso un messaggio di Status.
- *Status message*: un messaggio di Status può essere usato in differenti scenari:
 - per rispondere ad una richiesta Get
 - per rispondere ad un messaggio Set di tipo unacknowledged
 - per indicare autonomamente il proprio Stato; un chiaro esempio è dato da un sensore che comunica periodicamente il suo valore di temperatura ad un altro nodo.

Un concetto importante definito nella specifica mesh Bluetooth è il paradigma *model*. Un model porta con sé tutti i concetti precedenti ed è responsabile della definizione e dell'implementazione delle caratteristiche di un dato element[26]. Ogni elemento all'interno di un nodo deve supportare uno o più models che possono essere raggruppati in tre macrocategorie:

- *Server model*: definisce i messaggi che il modello può trasmettere e ricevere, gli stati in cui un elemento può trovarsi e i comportamenti (cambiamenti di stato) che vengono innescati dalla ricezione degli opportuni messaggi.
- *Client model*: definisce i messaggi necessari al nodo per richiedere, modificare o utilizzare i corrispondenti stati del server. Il modello client non ha stati.

- *Control model*: ha diverse funzionalità, coesistono anche uno o più dei modelli sopra descritti come logica di controllo per coordinare le iterazioni tra i modelli, per i quali è composto.

Per caratterizzare ciascuna delle tre macro-categorie, il Bluetooth SIG ha definito delle classi generali di model ben definite: *fundamental*, *generic*, *sensor*, *lighting* e *vendor-specific model*; ulteriormente suddivise in base alla loro destinazione d'uso.

In considerazione del fatto che per il lavoro di questa tesi vengono utilizzati dei messaggi appartenenti alla classe *Generic*, cerchiamo di chiarire questo concetto con un esempio. Qualsiasi dispositivo alimentato ha uno stato on-off, ad esempio un ventilatore, un condizionatore d'aria, una luce o una presa di corrente, ecc.; lo standard per evitare di caratterizzare ogni dispositivo con un proprio modello on-off, ha generalizzato il concetto creando un *Generic On/Off model* con stati e messaggi adattabili ad ogni situazione. All'interno della classe *Generic* ci sono anche altri modelli basati sullo stesso principio come *livello*, *transizione*, *localizzazione*, *potenza*, ecc. [61] e il termine *Generic* accanto a questi nomi indica proprio la generalità d'uso.

La tecnica di instradamento attraverso la quale i messaggi si propagano nella rete è il *managed flooding*. Si tratta di un meccanismo completamente asincrono basato sul *broadcasting*, è di semplice implementazione e richiede poche risorse, sia dal punto di vista computazionale, sia di spazio di archiviazione. La versione attuale della specifica non prevede alcuna funzionalità di *routing*. La tecnica del *flooding*, come è noto, potrebbe portare a *loop* e *ritrasmissioni illimitate* delle informazioni all'interno della rete. Per limitare questi problemi vengono messi in atto una serie di peculiarità volte ad ottimizzare la rete dal punto di vista dell'affidabilità e scalabilità. In particolare:

- *Time-To-Live (TTL)* : Il valore esprime il numero massimo di hops che un messaggio può compiere per raggiungere il destinatario, cioè il numero massimo di volte che può essere inoltrato dai nodi relay. Ogni volta che un messaggio viene ricevuto e poi inoltrato, il TTL viene decrementato di 1. Una volta raggiunto un valore di TTL pari a 1, il messaggio non verrà più inoltrato e verrà eliminato dalla coda di trasmissione.
- *Message cache* : una cache che memorizza i messaggi visualizzati di recente e si trova nel nodo relay. Quando un messaggio viene ricevuto, il nodo verifica se il messaggio è già presente nella cache. Se lo è, il messaggio viene ignorato. Se il messaggio non è presente nella cache, il nodo lo inoltra e lo aggiunge alla cache dei messaggi per ulteriori controlli futuri. La dimensione della cache dei messaggi è determinata dall'implementazione e dipende dalla topologia e dal numero di nodi della rete.

- *Heartbeat*: sono particolari messaggi utilizzati per monitorare i nodi e determinare la distanza in termini di hops tra mittente e destinatario al fine di ottimizzare la scelta del TTL più opportuno.
- *Subnets*: vengono utilizzate per consentire la partizione delle reti. Questo consente di evitare che i messaggi vengano inoltrati al di fuori di una data sottorete, ottenendo benefici dal punto di vista della sicurezza che del risparmio energetico.

2.2.2 Tipi di Nodi

In una rete mesh Bluetooth tutti i nodi possono ricevere e inviare messaggi, tuttavia ci sono però alcune caratteristiche particolari che un nodo può possedere e che gli permettono di avere delle capacità particolari:

- *Relay*: i nodi che supportano la funzionalità Relay, chiamati *nodì relay*, sono in grado di inoltrare i messaggi che ricevono da altri dispositivi. Il meccanismo di relaying, permette ad un messaggio di attraversare l'intera rete mesh effettuando una serie di hops (salti) tra diversi dispositivi che si trovano tra destinatario/i e mittente/i. Questo è un requisito fondamentale della rete mesh, poiché consente di estendere virtualmente il raggio di copertura del dispositivo che originariamente ha pubblicato il messaggio e permette di aumentare l'affidabilità della rete stessa.
- *Proxy*: i nodi della rete che supportano la funzionalità Proxy, noti come *Nodi Proxy*, sono in grado di inviare e ricevere messaggi tra il *Generic Attribute Profile (GATT)* e l'*Advertising Bearer*. In questo modo qualsiasi dispositivo dotato di tecnologia Bluetooth LE che non supporta l'*Advertising Bearer*, ad es. smartphone, tablet e PC possono interfacciarsi con la rete e comunicare con essa.
- *Friend*: di solito abilitato su un dispositivo senza vincoli di potenza per agire come buffer per i messaggi destinati ai *Low Power Node*. In questo modo, l'LPN può rimanere inattivo la maggior parte del tempo per risparmiare energia e comunque essere in grado di recuperare i pacchetti inviati a loro.
- *Low Power*: tale funzionalità consente ai dispositivi con limitazioni energetiche di far parte in modo efficiente della rete tramite il meccanismo di *friendship*. Un *Friend Node* si occupa della corrispondenza dell'LPN ricevendo e inoltrando i messaggi dell'LPN su richiesta. Ciò consente all'LPN di ridurre il tempo che deve dedicare all'ascolto della rete.
- *Provisioner*: questa funzionalità è prerogativa di tutti i dispositivi che, attraverso il processo di Provisioning, possono aggiungere un dispositivo alla rete mesh.

Durante il provisioning, il nodo di provisioning (PN) fornisce al nodo unprovisioned le informazioni e i parametri necessari per associarsi alla rete. Il ruolo di PN può essere svolto o da un dispositivo all'interno della rete Bluetooth, che quindi implementa l'intero stack di protocollo dello standard [29] o da un dispositivo esterno Bluetooth, ad esempio smartphone, tablet, attraverso l'uso della funzione Proxy. In quest'ultimo caso, la mobilità e le risorse messe a disposizione da un dispositivo mobile rendono molto più versatile la configurazione della rete e la sua gestione.

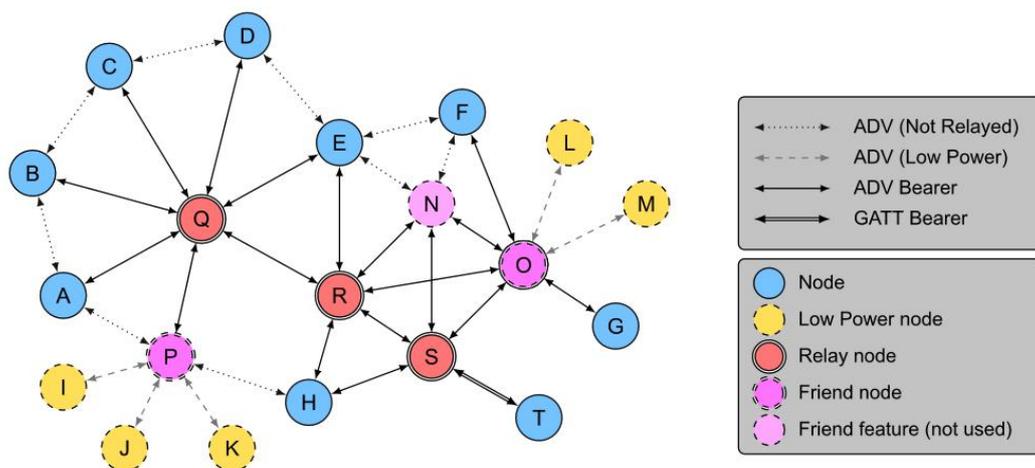


Figura 2.11: Esempio topologia di una rete mesh. (Fonte: [5])

La rete mesh Bluetooth utilizza il controller Bluetooth LE con il livello di collegamento e il livello fisico, mentre la parte host contiene una raccolta di livelli alternativi che implementano i protocolli e le procedure della rete mesh Bluetooth.

2.2.3 Architettura

Il Bluetooth mesh ha un'architettura a strati basata sul Bluetooth Low Energy, di cui utilizza i livelli più bassi, Link Layer e Physical Layer per trasmettere e ricevere le informazioni. Durante la trasmissione, la rappresentazione fisica del Bluetooth mesh è compatibile con i dispositivi Bluetooth LE esistenti, poiché i messaggi mesh sono contenuti all'interno del payload dei pacchetti di Advertising Bluetooth LE. Tuttavia, il Bluetooth mesh specifica uno stack host completamente nuovo e, sebbene alcuni concetti siano condivisi, il Bluetooth Mesh è incompatibile con l'host Bluetooth. Ciascun livello dello stack svolge una funzione specifica all'interno del nodo della rete. Di seguito vengono brevemente illustrati i diversi livelli di cui è composta l'architettura:

Capitolo 2 Bluetooth Low Energy and Bluetooth Mesh

1. Model layer: implementazione dei Model come comportamenti standardizzati per casi d'uso comuni. La definizione dei modelli include i loro messaggi, comportamenti, associazioni e stati.
2. Foundation Model layer: definisce gli stati, i messaggi e i modelli necessari per configurare e gestire una rete mesh.
3. Access layer: funge da mediatore tra i livelli applicativi superiori e l'upper transport layer definendo il formato dei dati dell'applicazione, il processo di crittografia e decrittografia che viene eseguito nell'Upper Transport Layer e la verifica dei dati. Altri meccanismi come Publish-Subscribe e Acknowledgments sono specificati a questo livello in base alla funzionalità del modello nei livelli superiori.
4. Upper transport layer: in questo livello i dati che attraversano il livello di accesso vengono crittografati/decrittografati e autenticati con Application-Keys. Inoltre, attraverso questo livello vengono trasportati i messaggi di controllo della rete (*Heartbeat e Friendship*)
5. Lower transport layer: si occupa della segmentazione e riassettaggio dei pacchetti rispettivamente per i nodi in trasmissione e ricezione.
6. Network layer: definisce funzionalità come la crittografia, la decrittografia, l'autenticazione dei messaggi del Lower Transport layer e il modo in cui i messaggi di trasporto vengono indirizzati verso uno o più elementi. Inoltre, il network layer decide se inoltrare i messaggi, accettarli per un'ulteriore elaborazione o rifiutarli.
7. Bearer layer: definisce l'interfaccia tra lo stack della rete Bluetooth mesh e lo stack del protocollo BLE e le modalità di trasporto dei pacchetti tra i nodi della rete. La specifica del Bluetooth mesh prevede due mesh bearer differenti:
 - Advertising Bearer: sfrutta le funzionalità di advertising e scanning del Generic Access Protocol (GAP) del Bluetooth LE per trasmettere e ricevere PDU di tipo *Mesh Message AD* tra i nodi della rete. Qualsiasi advertisement che usi il Mesh Message AD, deve essere un non-connectable e non-scannable undirected advertising events (ADV_NONCONN_IND). Affinchè questo bearer funzioni, a ciascun nodo è richiesto di supportare sia il ruolo di *Observer GAP* che quello di *Broadcaster GAP*.
 - GATT Bearer: utilizzato per fornire un'interfaccia di comunicazione tra i nodi delle rete e tutti quei dispositivi che non supportano l'advertising bearer (smartphone, pc, tablet). Attraverso il protocollo Proxy, questa tipologia

di livello bearer consente di trasmettere e ricevere i Proxy PDU tra due dispositivi mediante una connessione GATT.

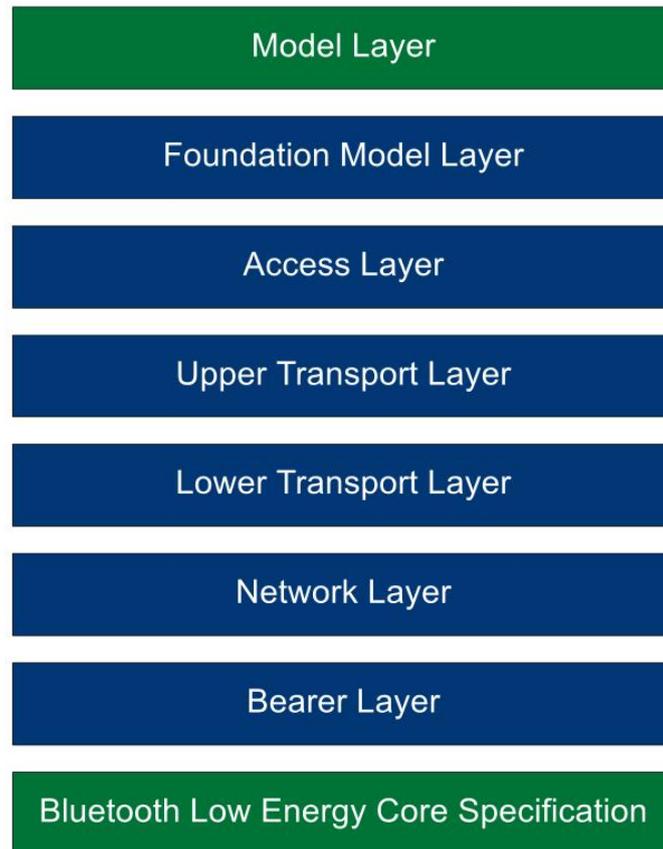


Figura 2.12: Architettura del sistema mesh. (Fonte: [5])

2.2.4 Comunicazioni Broadcast

Le reti Bluetooth mesh utilizzano un meccanismo di flooding senza connessione, in cui ogni dispositivo trasmette sia i propri dati che quelli degli altri dispositivi. A differenza di BLE, che funziona sia in modalità advertising che in modalità di connessione, i dispositivi Bluetooth mesh utilizzano solo la modalità advertising e i tre Primary Advertising Channel (37, 38 e 39).

I nodi non sono connessi ma comunicano tramite pacchetti di Advertising di tipo ADV_NONCONN_IND, utilizzando le regole di temporizzazione imposte dal protocollo BLE. Lo stesso Protocol Data Unit (PDU) viene trasmesso sequenzialmente sui tre Primary Advertising Channel all'interno di un Advertising Event e ricevuto da ogni dispositivo nel range radio sul corrispondente Primary Advertising Channel durante lo Scanning Event.

Capitolo 2 *Bluetooth Low Energy and Bluetooth Mesh*

La durata di ogni scansione su un Primary Advertising Channel è determinata dal parametro di scanWindow, che deve essere impostato con un duty-cycle il più vicino possibile al 100% per evitare di perdere eventuali PDU in ingresso. L'inizio di due eventi pubblicitari consecutivi è definito dall'AdvInterval (≥ 20 ms) più il valore del ritardo casuale (advDelay) compreso tra 0 e 10 ms, come specificato dalle specifiche BLE.

Per ridurre la probabilità di collisioni di pacchetti, lo standard raccomanda che il tempo che intercorre tra l'inizio della trasmissione di due ADV_NONCONN_IND consecutivi (TchPDU) all'interno dell'Advertisign Event deve essere inferiore o uguale a 10 ms e randomizzato (Fig. 2.8).

Capitolo 3

Materiali e Metodi

In questo capitolo verranno descritti gli strumenti utilizzati per lo sviluppo della ricerca. In particolare, verrà illustrata la piattaforma hardware utilizzata per lo sviluppo delle applicazioni reali, analizzeremo il simulatore e le metriche utilizzate per confrontare le prestazioni della rete Bluetooth mesh nei diversi scenari e implementazioni proposti.

3.1 Kit di sviluppo nRF52

L'nRF52 DK è una scheda di sviluppo prodotta da Nordic Semiconductor, progettata per aiutare gli sviluppatori a creare prototipi e prodotti finali basati sui microcontrollori nRF52.

Il nRF52 DK dispone di un microcontrollore a 32 bit ARM Cortex-M4F, che funziona a 64 MHz, ed è dotato di Bluetooth Low Energy (BLE), NFC, ANT, e una serie di interfacce per la comunicazione, come UART, SPI, I2C e GPIO.

La scheda di sviluppo è compatibile con molte piattaforme di sviluppo software, come il Nordic nRF5 SDK, il toolchain di GNU Arm Embedded, il Nordic nRF Connect SDK, e altri. È compatibile con lo standard Arduino Uno Revision 3, il che consente di montare facilmente shield di terze parti. È fornito inoltre completo di un debugger SEGGER J-Link integrato che permette la programmazione e il debug sia del SoC sulla scheda che dei SoC esterni via basetta esterna. È inoltre interfacciabile direttamente con il Power Profiler Kit e Power Profiler Kit versione 2.

L'nRF52 DK è anche compatibile con altri strumenti e librerie, come ad esempio il software di sviluppo ARM Keil MDK, il compilatore GCC, e diverse librerie open source. Ciò significa che gli sviluppatori possono utilizzare gli strumenti che preferiscono per sviluppare le loro applicazioni.

Nordic Semiconductor fornisce una vasta gamma di esempi di progetti per la scheda di sviluppo nRF52 DK, che coprono diverse applicazioni, tra cui Bluetooth a bassa energia, mesh networking, sensori e monitoraggio della salute.

L'nRF52 DK è una scelta popolare per gli sviluppatori che lavorano su dispositivi IoT, wearable e smart home, grazie alla sua flessibilità e alla sua vasta gamma di funzionalità integrate. Inoltre, la scheda è relativamente economica e facile da usare.

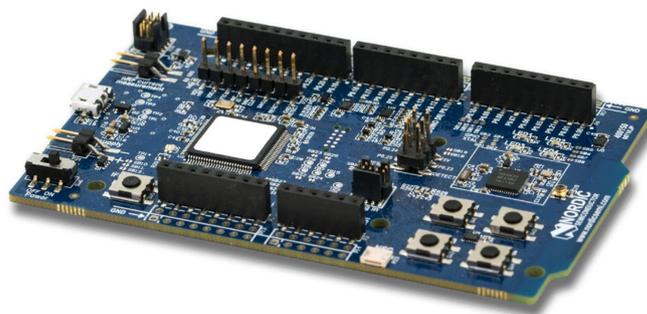


Figura 3.1: nRF52 Development Kit

Caratteristiche principali:

- Economico, soluzione rapida per la prototipazione e lo sviluppo per i SoC nRF52810 e nRF52832.
- Supporta lo sviluppo di Bluetooth LE, Bluetooth mesh, ANT e sviluppo proprietario a 2,4 GHz
- Supporto per le funzionalità Bluetooth 5: 2 Mbps, Advertising Extensions e Channel Selection Algorithm #2 (CSA #2)
- Arm® Cortex™ M4 con supporto floating point.
- Tutti i pin di I/O e le interfacce sono disponibili tramite connettori.
- Set di istruzioni DSP.
- Compatibile con Arduino Uno Rev. 3 per l'uso con schede di terze parti.
- Supporto alla programmazione/debugging tramite Segger J-Link OB.
- Supporto alla programmazione/debugging di schede esterne.
- Antenna PCB integrata a 2,4 GHz
- Antenna NFC.
- Connettore per le misurazioni RF
- Pin per la misurazione del consumo energetico.

- Porta per batteria a bottone.
- Alimentazione da 1,7-3,6 V fornita dalla batteria, da USB o esterna.

3.2 Bluetooth Toolbox di MathWorks

Il Bluetooth Toolbox di MathWorks è uno strumento fondamentale per gli ingegneri e gli sviluppatori che lavorano con dispositivi Bluetooth, in quanto semplifica notevolmente il processo di sviluppo e test dei dispositivi Bluetooth. Il Toolbox offre un set completo di strumenti e funzioni per progettare, simulare e verificare i sistemi di comunicazione Bluetooth, compresi Bluetooth Low Energy (BLE) e Bluetooth Mesh.

Grazie al Toolbox, è possibile configurare, simulare e analizzare collegamenti di comunicazione Bluetooth end-to-end, creare e riutilizzare test bench per verificare la conformità delle progettazioni, dei prototipi e delle implementazioni allo standard Bluetooth. Gli utenti possono utilizzare questo strumento per creare una vasta gamma di dispositivi Bluetooth, dal controllo remoto a basso consumo energetico a dispositivi IoT complessi.

Oltre alla creazione di dispositivi, il Toolbox permette di valutare gli scenari di coesistenza, interferenza, localizzazione e modellare i canali Bluetooth per simulare il comportamento del segnale durante la trasmissione. Ciò significa che gli utenti possono testare la loro implementazione Bluetooth in un ambiente simulato prima di implementarla sul campo.

In particolare, il Bluetooth Toolbox fornisce la capacità di configurare, simulare e visualizzare la comunicazione multi-nodo in una rete Bluetooth Mesh, fornendo in output diverse statistiche riferite ai differenti livelli del protocollo implementati (Application Layer, Transport Layer, Network Layer, Link Layer, Physical Layer). Il simulatore per reti Bluetooth Mesh incluso nel Toolbox è compliant con lo standard Bluetooth Mesh, oltre che con lo standard BLE.

3.3 Metriche

In questa sezione verranno definite le metriche di valutazione che saranno utilizzate successivamente nella trattazione.

3.3.1 Packet Delivery Ratio

Il packet delivery ratio (PDR) è una metrica utilizzata per valutare la qualità del servizio fornito da una rete di comunicazione, in particolare in relazione alla trasmissione di dati.

In generale, il PDR rappresenta la frazione di pacchetti trasmessi correttamente dalla sorgente alla destinazione rispetto al numero totale di pacchetti inviati dalla sorgente. In altre parole, il PDR indica la percentuale di pacchetti che raggiungono la destinazione con successo rispetto a quelli che vengono persi o danneggiati durante la trasmissione.

Ad esempio, se una sorgente invia 100 pacchetti e solo 90 di essi arrivano a destinazione correttamente, il PDR sarà del 90

Il PDR è una misura importante per valutare l'affidabilità e la robustezza di una rete di comunicazione,

Nel caso specifico della ricerca, si è scelto di valutare il Packet Delivery Ratio a livello applicativo, che corrisponde al rapporto tra i pacchetti generati dal nodo sorgente (TX_packet_APP_SRC) e quelli ricevuti dal nodo destinatario finale (RX_packet_APP_DST), entrambi a livello applicativo.

$$\text{PDR} = \frac{\Sigma(\text{RX_packet_APP_DST})}{\Sigma(\text{TX_packet_APP_SRC})} * 100$$

I dati a livello applicativo possono essere prelevati direttamente dalle statistiche fornite dal simulatore Bluetooth mesh del Bluetooth Toolbox.

3.3.2 Latenza end-to-end

La latenza end-to-end (E2E latency in inglese) è una metrica che indica il tempo che trascorre tra l'invio di un pacchetto di dati dalla sorgente e la sua ricezione corretta da parte della destinazione, considerando tutti i ritardi presenti nella rete di comunicazione.

La latenza E2E può avere un impatto significativo sulla qualità del servizio fornito da una rete di comunicazione, soprattutto in applicazioni in tempo reale come la telefonia, il controllo di processi industriali o i giochi online. In tali applicazioni, latenze elevate possono causare ritardi e scatti che possono compromettere l'esperienza dell'utente finale. Pertanto, la latenza E2E è una metrica critica che deve essere monitorata e minimizzata nelle reti di comunicazione.

Per gli scopi della ricerca, risulta interessante valutare la latenza media end-to-end utilizzando la seguente formula:

$$\text{Latenza} = \frac{\Sigma(\text{Latenza_msg_n_RX})}{(\text{Num_msg_Tot_RX})}$$

Dove Latenza_msg_n_RX indica la latenza dell'n-esimo messaggio ricevuto; mentre Num_msg_Tot_RX è il numero totale (n) di messaggi ricevuti.

Capitolo 4

Miglioramento della ricezione dei messaggi di conferma nel Bluetooth mesh

La tecnologia Bluetooth mesh rappresenta una soluzione eccellente per le applicazioni di illuminazione commerciale e industriale, anche se è ancora in evoluzione. Tuttavia, come evidenziato dalle specifiche del Bluetooth mesh [14], uno dei maggiori problemi di questa tecnologia di rete riguarda il miglioramento della ricezione dei messaggi di conferma, soprattutto quando più dispositivi rispondono contemporaneamente, come può accadere in alcune applicazioni di illuminazione. Questo comportamento può generare problemi imprevisti e ridurre la probabilità di consegna dei messaggi a causa di collisioni, soprattutto quando il numero di dispositivi nella rete aumenta. Pertanto, determinare la ricezione dei messaggi di Status in una rete Bluetooth mesh è un aspetto critico che richiede ulteriori sviluppi per garantire una comunicazione affidabile e senza interruzioni tra i dispositivi.

4.1 Messaggi di Status

La possibilità di avere un feedback da parte di dispositivi controllati è importante in molte applicazioni perché consente ai dispositivi di comunicare lo stato delle loro attività e di fornire informazioni sulle loro condizioni.

Ad esempio, nei sistemi di domotica, il feedback dei dispositivi come luci, termostati o sensori di movimento consente all'utente di controllare la casa in modo più efficiente e sicuro. Quando l'utente interagisce con il sistema di domotica tramite un'applicazione mobile o una console centrale, riceve un feedback che indica lo stato attuale di ogni dispositivo. Ciò consente all'utente di sapere se le luci sono accese o spente, se il termostato sta regolando la temperatura in modo corretto o se i sensori di movimento stanno rilevando eventuali movimenti all'interno della casa. Nella specifica Bluetooth

mesh, la funzionalità di conferma delle operazioni è garantita attraverso l'utilizzo dei messaggi ACK Set e Status (descritti nel paragrafo Background). In breve, i messaggi ACK Set sono utilizzati per modificare il valore di uno o più elementi e richiedono l'invio del messaggio di Status per confermare che l'operazione sia stata eseguita con successo.

La presenza di messaggi di conferma può causare un incremento del traffico nella rete, aumentando la probabilità di collisioni tra i pacchetti in ricezione. Questo problema può verificarsi più frequentemente in situazioni in cui diversi nodi inviano messaggi di conferma a un singolo nodo, ad esempio in applicazioni di illuminazione in cui molteplici corpi illuminanti sono controllati da un singolo interruttore. In Fig 4.1 viene presentato un possibile scenario di comunicazione tra 3 nodi, assimilabili ad un interruttore (nodo A) e due lampade (nodi B e C), solo a scopo esemplificativo. Il nodo A invia un comando di ACK Set che viene ricevuto da entrambi i nodi B e C attraverso il canale 38. Successivamente, i due nodi inviano due messaggi di Status, che vengono ricevuti quasi contemporaneamente dal dispositivo A sul canale 39 causando una collisione tra i messaggi e la perdita di informazione.

In questo contesto, i parametri temporali definiti nella specifica BLE non sempre sono in grado di garantire un'adeguata spaziatura temporale delle trasmissioni e quindi sopperire al problema delle loro sovrapposizioni.

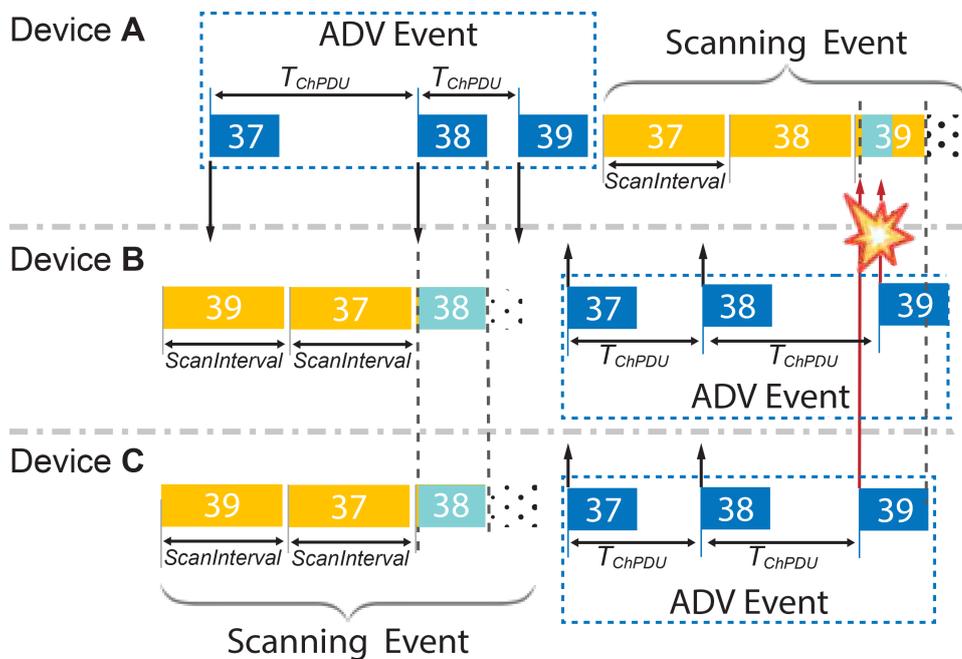


Figura 4.1: Esempio del problema di collisione durante il flusso di comunicazione tra tre dispositivi in una rete Bluetooth mesh. (Fonte: [4])

Più nodi ricevono un messaggio ACK Set all'interno di un gruppo, maggiore sarà il numero di messaggi di Status che il nodo mittente dovrà gestire.

Per evitare il problema delle collisioni tra pacchetti, la specifica Bluetooth mesh raccomanda di applicare un ritardo nella trasmissione del messaggio di conferma. Tuttavia, l'implementazione di queste soluzioni non è obbligatoria e rimane a discrezione dello sviluppatore.

In particolare, lo Standard differenzia il ritardo da applicare in base ai destinatari del messaggio di conferma. Se il messaggio di stato viene inviato come risposta a un messaggio destinato a un indirizzo unicast, il nodo dovrebbe trasmettere la risposta con un ritardo casuale tra i 20 e i 50 millisecondi. Al contrario, se il messaggio è inviato come risposta a un messaggio destinato a un indirizzo di gruppo, il nodo dovrebbe utilizzare un ritardo casuale compreso tra i 20 e i 500 millisecondi.

Questa soluzione riduce la probabilità che più nodi rispondano contemporaneamente, evitando collisioni di messaggi e aumentando le possibilità di consegna corretta degli stessi.

4.2 Setup sperimentale

Nella configurazione proposta sono stati impiegati dei kit di sviluppo (nRF52 DK) della Nordic Semiconductor per la realizzazione dei nodi della rete Bluetooth mesh. A partire dal Software Development Kit (SDK) di proprietà della stessa nordic, sono stati implementati nelle nRF52 DK, dieci *Light Switch Server* e un *Light Switch Client*. Nel seguito saranno indicati rispettivamente come *Server* e *Client*. Il Server implementa un'istanza del Generic OnOff Server Model e identifica le lampade da accendere e spegnere. A livello visivo la lampada è stata rappresentata da uno dei LED integrati nelle schede nRF52 DK. Il Client implementa un'istanza di un Generic OnOff Client Model e rappresenta un interruttore che aziona tutte le lampade. Nella configurazione proposta, il client è stato connesso a un PC e attraverso l'uso del software J-link Real-Time Transfer (RTT) Logger sono stati prelevati i dati dalla scheda nRF52 DK e memorizzati all'interno di un file di log. Il software J-link RTT Logger fa parte del pacchetto software J-Link.

Per la creazione della rete Bluetooth mesh, come menzionato nel capitolo [2.2.1](#) è necessario un Provisioner che sia in grado di configurare ciascun nodo permettendogli di operare e scambiare messaggi. Nel nostro caso, sia il Server che il Client adottati sono stati integrati con funzionalità proxy, ciò ha consentito di utilizzare uno smartphone come provisioner. Attraverso l'app nRF mesh è stato eseguito il processo di provisioning. Una volta effettuato il provisioning, ogni nodo mostra le proprie capacità, attraverso l'adozione di un insieme di stati di configurazione, che ne descrivono il comportamento all'interno della rete mesh. Oltre a fornire le informazioni essenziali per associare

Tabella 4.1: Parametri di configurazione

Parametro	Valore
Numero di dispositivi	11
Servers $S\Delta$ (m)	0.40
Client-Servers $CS\Delta$ (m)	3-3.6
Tx Power	0 dBm
Dimensione PDU	37 bytes
TTL	6
Network retransmit interval steps	5
Network retransmit count	3
Relay retransmit interval steps	5
Relay retransmit count	3
AdvInterval (ms)	20

i dispositivi alla rete, attraverso l'app è stato possibile assegnare gli indirizzi di pubblicazione e sottoscrizione. Per ottenere una comunicazione one-to-many tra Client e Server, al Generic OnOff Client Model è stato assegnato un indirizzo di pubblicazione di tipo Group. Lo stesso valore è stato poi impostato come indirizzo di sottoscrizione nel Generic OnOff Server Model implementato in ciascun Server. In questo modo, ad ogni messaggio di tipo Set, tutti i Server sono interessati contemporaneamente dalla stessa richiesta. Per quanto riguarda i messaggi di conferma diretti dai Server al Client, sebbene possa essere schematizzata come una comunicazione many-to-one, la risposta di ciascun Server avviene sempre attraverso l'utilizzo di un indirizzo Unicast e in modalità one-to-one.

La topologia di configurazione e i parametri di temporizzazione scelti per la rete mesh del setup proposto sono riassunti nella Tab. 4.1.

Per tutti i nodi è stata impostata una potenza trasmittiva a 0 dBm, più che sufficiente per il tipo di test effettuato, e un valore di TTL pari a 6. Tenendo presenti alcune fondamentali considerazioni emerse in [25], sono stati adottati sia per il Client che per il server, un Network Retransmit Interval Steps = 3 e un Network Retransmit Count = 5. Questi valori garantiscono la ricezione del comando On-Off da parte dei Server. Per quanto riguarda la funzionalità relay, questa è stata attivata sia per i Client che per i Server con un Network Retransmit Interval Steps = 3 e un Network Retransmit Count = 5. La funzionalità relay è particolarmente utile quando il numero di nodi che ritrasmettono simultaneamente i messaggi di Status aumenta nella rete, poiché incrementa i possibili percorsi che un pacchetto può intraprendere migliorando il successo della trasmissione all'interno della rete e, conseguentemente, le prestazioni di affidabilità complessive.

4.2 Setup sperimentale

Si noti che la scelta dei valori dei parametri fin qui descritti sono il risultato di decisioni prese attraverso test preliminari effettuati al fine di ottimizzare le prestazioni della rete.

Infine, è fondamentale considerare la relazione tra la durata di un Advertising Event e lo Scanning Event per garantire il successo nella ricezione dei messaggi. In particolare, è necessario valutare il corretto allineamento tra il canale di trasmissione e il rispettivo canale di scansione. Nel setup sperimentale proposto, il comportamento dei dispositivi può essere riassunto come segue:

1. il Client invia un comando Generic OnOff Set con acknowledgment, ad esempio richiede che i LED di ciascun server vengano accesi. Tale comando può assumere solo due valori: 1 per On e 0 per Off;
2. i Server ricevono il comando, lo elaborano modificando lo stato del LED e infine inviano il loro messaggio di Generic OnOff Status al client. Allo stesso tempo, dato che la funzionalità relay è attiva in ogni server, i nodi si occupano anche di inoltrare i messaggi ricevuti dagli altri nodi e dallo stesso Client;
3. i molteplici messaggi di Status raggiungono il client;
4. il software J-link RTT Logger prende i dati dal client e li memorizza nel file designato.

La sequenza di comandi generati dal Client consiste in un'alternanza di Set On e Set Off spazati di 10s uno dall'altro. Il comportamento del Client consente di ripetere ciclicamente quanto descritto dal punto 1 al punto 4 per l'intera durata, circa 15 ore, di ogni prova. Il totale dei messaggi di tipo Set con Acknowledgment inviati dal Client ai 10 nodi server è pari a 5470. Al termine della sequenza di comandi, ogni file di log generato è stato sottoposto ad un'attenta revisione per trovare eventuali errori che potessero invalidare il processo. Confrontando una singola prova con un possibile utilizzo quotidiano, ad esempio all'interno di un locale commerciale, e ipotizzando una media di sei comandi al giorno tra accensioni e spegnimenti, il valore simulato di 5470 comandi corrisponde a circa 911 giorni o equivalentemente a 2,5 anni di utilizzo.

L'analisi dei pacchetti persi nella rete Bluetooth mesh è stata effettuata mediante l'analisi dei risultati di test composti da più prove, come descritto in precedenza. Nel setup sperimentale proposto, ogni prova è stata ripetuta dieci volte per garantire una quantità di dati statisticamente significativa. I test sono stati eseguiti in scenari diversi, sia in un ambiente controllato che in uno reale.

Lo scenario controllato è stato allestito in un laboratorio con pochi elementi interferenti, come mostrato in Fig. [4.2](#). Il Client e il PC sono stati posizionati su un banco

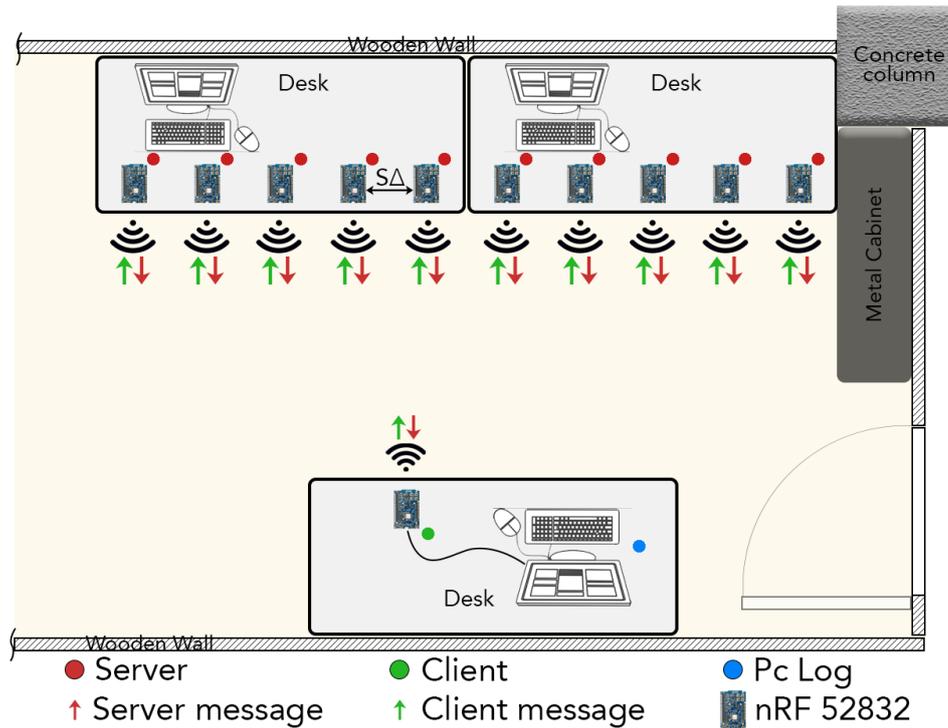


Figura 4.2: Schema di implementazione della rete Bluetooth mesh dello scenario controllato. (Fonte: [4])

del laboratorio, mentre i dieci nodi server sono stati posizionati su un altro banco a una distanza di circa 3-3,6 m l'uno dall'altro, con una distanza di circa 40 cm tra loro.

In base alla potenza di trasmissione impostata e alla disposizione, tutti i nodi della rete sono in completa visibilità radio. Questa configurazione non rappresenta le condizioni più favorevoli per lo scambio di messaggi all'interno della rete. Infatti, a causa della funzionalità di relay attiva su ogni nodo e alla completa visibilità di tutti i nodi, il traffico di messaggi che la rete deve gestire è elevato, generando possibili collisioni e perdite di pacchetti. Tuttavia, questo tipo di scenario è comune in ambienti commerciali, case, industrie, sale espositive, ecc. e rappresenta uno dei peggiori scenari di smart-lighting in termini di comunicazione tra dispositivi. Il Client deve gestire un elevato numero di messaggi di conferma in ricezione, intervallati temporalmente solo da piccoli ritardi e, nella maggior parte dei casi, insufficienti.

Le prove dello scenario controllato sono state effettuate durante le ore di chiusura del laboratorio, in un ambiente privo di ostacoli e dispositivi per ridurre al minimo le interferenze non prevedibili e garantire una condizione sperimentale simile per ogni test. Ciò garantisce la riproducibilità del test e impedisce che le misurazioni PLR siano influenzate dall'interferenza co-channel dovuta a ostacoli e dispositivi nell'ambiente.

Per lo scenario reale, è stato utilizzato un grande ufficio (Fig. 4.3) con due blocchi di 12 scrivanie ciascuno. Ogni postazione di lavoro è separata da pareti in legno che

si alzano di circa 70 cm sopra il piano della scrivania. I nodi Server sono stati disposti in modo da avere una distribuzione omogenea in tutta la stanza e una ridondanza dei percorsi per lo scambio dei messaggi. Le distanze tra i nodi variano tra i 2 e i 6 metri. Le prove dello scenario reale sono state effettuate a ridosso dell'orario d'ufficio, considerando sempre un arco temporale di 15 ore.

Contrariamente alla situazione controllata, questo scenario presenta numerose fonti interferenti e attenuanti per i segnali trasmessi dai dispositivi Bluetooth mesh. Oltre all'attenuazione e alle riflessioni dovute agli elementi strutturali e di arredo, sono presenti molteplici disturbi RF a cui si aggiungono la presenza di persone che occupano l'ambiente. Come mostrato in Fig. 4.3, troviamo un Access Point Wi-Fi posto a soffitto in posizione centrale e una rete di otto Beacon Bluetooth distribuiti in tutta la stanza. Questi ultimi trasmettono le proprie informazioni sugli stessi canali di advertising (canali 37, 38, 39) sfruttati dalla rete Bluetooth mesh. Ai disturbi elencati, che possiamo considerare fissi all'interno della stanza, si aggiungono tutti quelli casuali dovuti alle persone e ai loro dispositivi personali, come smartphone, tablet e PC che accedono a questo spazio.

4.3 Evaluation tests design

Per valutare le prestazioni della ricezione dei messaggi di conferma nella rete Bluetooth mesh sono stati condotti due test in ciascuno degli scenari descritti nella sezione precedente. I test differiscono per l'introduzione di una novità sostanziale nella gestione dei messaggi di status, a livello del model layer dei Server. In particolare, è stato effettuato un test con una configurazione standard (CS) della rete Bluetooth mesh e un altro con una configurazione ottimizzata (CO) in termini di temporizzazione nell'invio dei messaggi di status.

Nel test CO, la modifica apportata al model layer dei Server consiste nell'introduzione di un ritardo casuale compreso tra 20 e 500 ms, prima di inoltrare i messaggi di Status attraverso i livelli sottostanti. Nel test CS, invece, i server non implementano alcun ritardo. Tutti gli altri parametri di configurazione della rete Bluetooth mesh sono comuni a entrambe le tipologie di test, il che consente di confrontare direttamente le prestazioni di CO con quelle di CS, partendo dalle stesse condizioni iniziali per tutte le altre variabili di configurazione. Pertanto, CO si differenzia da CS nell'implementazione di una tecnica di trasmissione dello Status nel tempo, ma l'impostazione e il comportamento generale dell'intera rete sono identici, come descritto nella sezione sopra.

La tecnica implementata, grazie alla diffusione dei messaggi di Status nel tempo, riduce il problema delle collisioni permettendo di aumentare le possibilità della loro ricezione. Il test CO è stato implementato in ogni Server e i principali passaggi coinvolti,

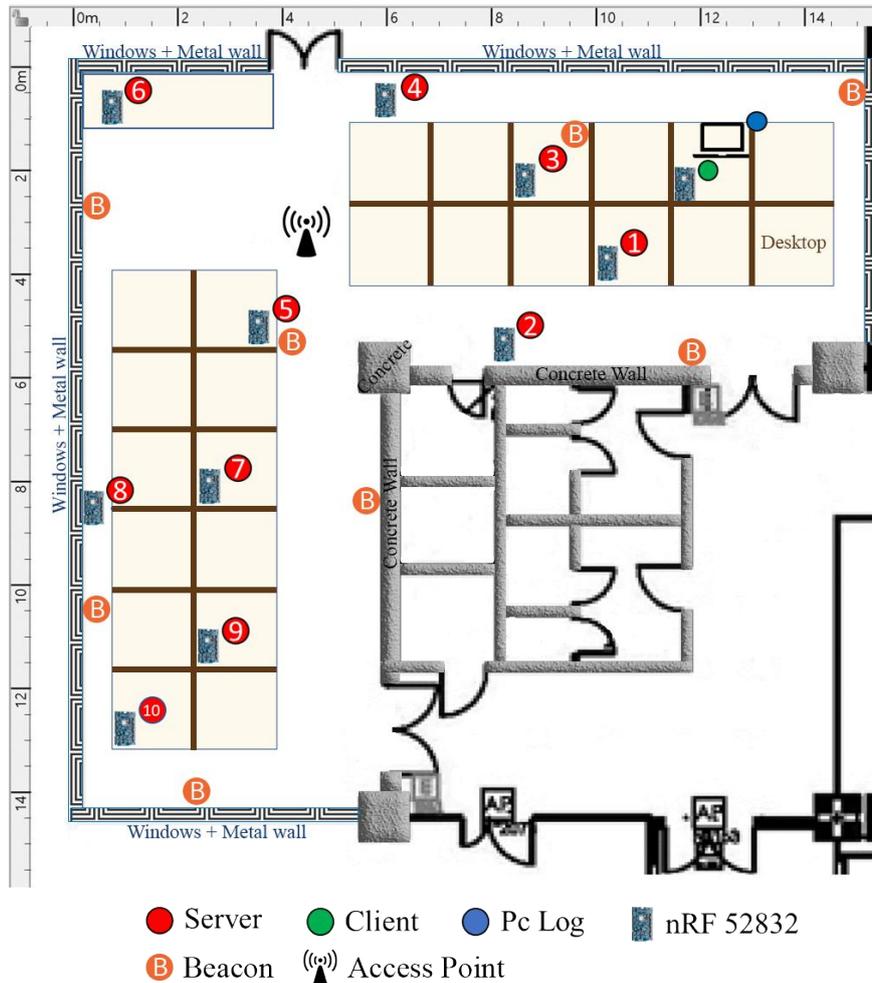


Figura 4.3: Schema di implementazione della rete Bluetooth mesh dello scenario reale. (Fonte: [4])

dalla ricezione di un messaggio Set alla risposta con la pubblicazione di uno Status, sono mostrati in Fig. 4.4. Inizialmente, per ogni comando On/Off inviato dal Client, ogni Server riceve, a livello di applicazione, un nuovo valore Set On/Off derivato dall'Access Layer PDU trasmesso dai livelli mesh sottostanti. Successivamente, attraverso una funzione richiamata a livello Model, viene verificato se un messaggio di stato deve essere inviato al Client, e pubblicato verso i livelli sottostanti dopo un ritardo casuale imposto.

Per generare un valore di ritardo casuale per ogni Status inviato, viene utilizzato un algoritmo di generazione di numeri pseudo-casuali. Il seme dell'algoritmo dipende dall'indirizzo del dispositivo Bluetooth, cioè un identificativo univoco a 48 bit assegnato a ciascun dispositivo dal produttore. L'indirizzo generalmente è visualizzato tramite sei byte scritti in formato esadecimale.

Per generare il seme, vengono concatenate le prime quattro cifre dell'indirizzo del

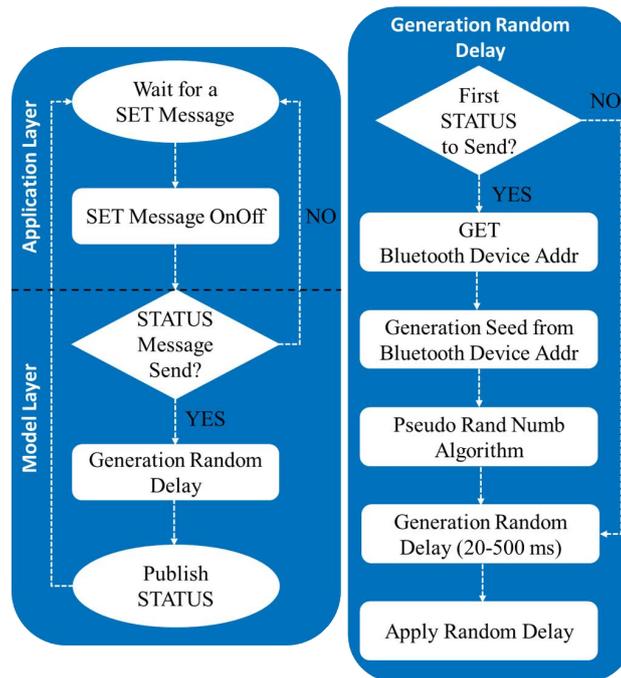


Figura 4.4: Diagramma a blocchi dell’algoritmo di randomizzazione dei messaggi di Status. (Fonte: [4])

dispositivo, partendo dal byte meno significativo fino a quello più significativo. La stringa risultante viene convertita in un valore decimale e utilizzata come seme per l’algoritmo generatore di numeri pseudocasuali. Questo processo viene eseguito la prima volta che viene ricevuto un messaggio di ACK Set. Successivamente, ogni volta che deve essere inviato un nuovo Status, viene generato un valore di ritardo casuale compreso tra 20 e 500 millisecondi, come specificato dal protocollo Bluetooth mesh, utilizzando il seme iniziale come punto di partenza.

4.4 Risultati

Sia per lo scenario controllato che per quello reale le prestazioni della rete sono state valutate in termini di PSR degli Status. Per analizzare le prestazioni delle reti nell’esecuzione dei comandi On/Off, sono stati considerati un totale di 54.700 comandi inviati dal Client per ciascun test (10 prove), ovvero 5.470 comandi per ciascuno dei dieci Server nella rete.

Successivamente, è stato calcolato il totale dei messaggi di Status persi per ciascun test e il numero di messaggi di Status ricevuti. Infine, è stata calcolata la media dei risultati ottenuti in ciascun test per tutte le dieci prove eseguite in ciascuno scenario. Per valutare la significatività della differenza tra i due test condotti, è stato adottato il *test t di Student* [31] tra le medie lungo i 10 testSe CO.

Tabella 4.2: Risultati complessivi dei PSR degli Status per il test CS dello scenario controllato.

ID Test	NODO										Avg. Single Test
	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	
CS_1	93.03	97.51	96.33	95.90	96.25	94.75	94.73	95.37	95.98	96.16	95.60
CS_2	96.07	97.88	96.63	97.68	97.07	96.73	96.42	96.36	96.45	98.17	96.95
CS_3	96.56	98.45	97.11	97.48	97.92	97.04	98.01	97.44	97.53	97.57	97.51
CS_4	95.70	97.90	97.22	97.48	97.61	96.65	97.61	96.89	96.78	97.09	97.09
CS_5	96.01	98.76	96.87	97.75	97.95	97.53	97.86	97.77	97.33	97.39	97.52
CS_6	95.78	98.68	96.91	95.72	96.86	96.98	96.91	96.18	97.39	96.20	96.76
CS_7	96.75	98.61	96.98	97.31	97.26	97.53	97.44	97.68	97.90	98.08	97.55
CS_8	97.35	98.57	97.62	97.15	96.76	97.57	97.57	96.97	97.71	98.03	97.53
CS_9	96.67	98.48	97.09	97.42	96.78	97.46	97.70	97.15	97.17	97.70	97.36
CS_10	94.46	97.77	96.12	96.07	96.80	95.01	95.85	95.72	95.61	95.72	95.91
Node Avg.	95.84	98.26	96.89	97.00	97.13	96.73	97.01	96.75	96.99	97.21	

Per quanto riguarda lo scenario controllato eseguito in laboratorio, i risultati dei test CS e CO sono riportati rispettivamente nelle Tabelle 4.2 e 4.3. Considerando i risultati complessivi per ogni nodo (Tabella 4.2) e poi mediati per tutti e dieci i test, nel caso peggiore del test CS è stata ottenuta una media del 95,84% di messaggi di Status ricevuti dal Client dal nodo n1, mentre il miglior risultato è stato raggiunto dal nodo n2, con il 98,26% di messaggi di Status consegnati correttamente. Nel test CO, invece, la media nel caso peggiore è pari al 98,44% per i messaggi di Status ricevuti dal client dal nodo uno e il miglior risultato è stato ottenuto dal nodo due, con il 99,50% di messaggi di Status correttamente recapitati.

Considerando i risultati complessivi ottenuti per ogni test Tab. 4.2, nel calcolo della media tra tutti i nodi, il test CS mostra il risultato peggiore, con una media del 95,60% per i messaggi di Status ricevuti dal Client nel test CS_1, e il miglior risultato raggiunto nel test CS_7, con il 97,55% dei messaggi recapitati correttamente. Mentre, per il test CO, nella situazione peggiore, abbiamo ottenuto una media del 98,77% nel caso del test CO_7, e il miglior risultato ottenuto nel test CO_6, dove il 98,91% dei messaggi di Status sono stati consegnati correttamente.

In generale, le prestazioni di ciascun nodo sono state sempre superiori per il test CO rispetto alla sua controparte CS per tutte le medie dei dieci test (vedi Fig. 4.5 e Fig. 4.6). Risultati simili possono essere ottenuti considerando la media tra tutti i nodi per ogni test.

Come mostrato nella Tab. 4.4, il PSR degli Status risultante, nel caso del test CO, è stato significativamente diverso rispetto a quello CS, con una differenza di 1.86 punti percentuali. Per quanto riguarda la CO, sono stati ricevuti il 98.84% degli Status,

Tabella 4.3: Risultati complessivi dei PSR degli Status per il test CO dello scenario controllato.

ID Test	NODO										Avg. Single Test
	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	
CO_1	98.56	99.45	99.07	98.85	98.52	99.40	98.81	98.48	99.21	98.50	98.88
CO_2	98.32	99.47	98.96	98.56	98.48	99.20	98.78	98.39	99.16	98.57	98.79
CO_3	98.45	99.52	99.38	98.81	98.59	99.18	98.68	98.19	99.32	98.83	98.90
CO_4	98.63	99.31	99.36	98.63	98.70	99.10	98.48	98.03	99.12	98.78	98.81
CO_5	98.45	99.52	99.36	98.63	98.56	99.07	98.59	98.12	99.31	98.76	98.84
CO_6	98.23	99.63	99.52	98.76	98.78	98.98	98.63	98.46	99.29	98.79	98.91
CO_7	98.28	99.65	99.34	98.39	98.81	99.16	98.37	98.19	98.88	98.61	98.77
CO_8	98.56	99.65	99.51	98.72	98.68	98.90	98.74	98.34	98.94	98.52	98.86
CO_9	98.46	99.31	99.29	98.68	98.50	99.03	98.67	98.37	98.87	98.79	98.80
CO_10	98.50	99.49	99.21	98.63	98.45	99.21	98.72	98.52	99.25	98.57	98.86
Node Avg.	98.44	99.50	99.30	98.67	98.61	99.12	98.65	98.31	99.14	98.67	

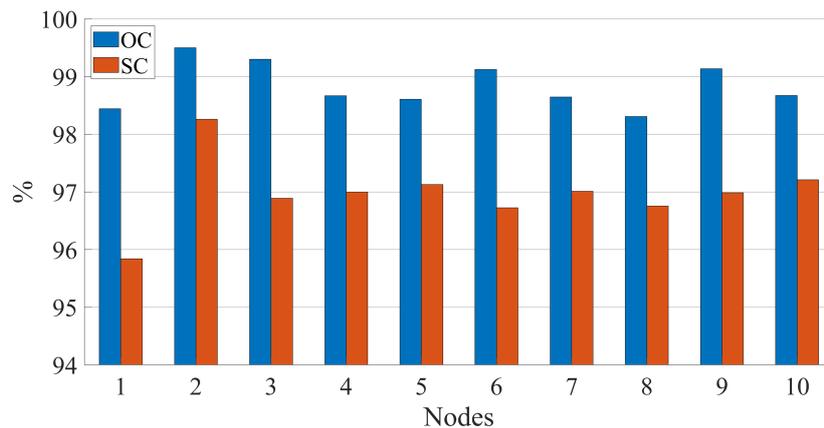


Figura 4.5: Confronto dei PSR degli Status per ciascun test dello scenario controllato, mediato per tutti e dieci i test, sia per i casi CO che CS. (Fonte: [4])

mentre per l'SC il 96.98%. Inoltre, la relativa varianza media calcolata su entrambi i test CS e CO è risultata particolarmente significativa, rispettivamente dello 0,44% per CS e dello 0,002% per CO, come presentato nella Tab. 4.4. Ciò risulta sia per quanto riguarda la varianza tra le medie sui singoli nodi che per i singoli test effettuati.

Per quanto riguarda lo scenario reale svolto in ufficio, i risultati dei test CS e CO sono riportati nelle Tab. 4.5 e Tab. 4.6.

Analizzando i risultati dei dieci test effettuati, si può notare che nel caso del test CS i valori migliori e peggiori sono stati rispettivamente ottenuti dal nodo n3 e dal nodo n9. Il nodo n3 ha raggiunto l'80,67% di messaggi di stato consegnati correttamente, mentre il nodo n9 ha ottenuto il valore più basso, pari al 54,52%. Per quanto riguarda

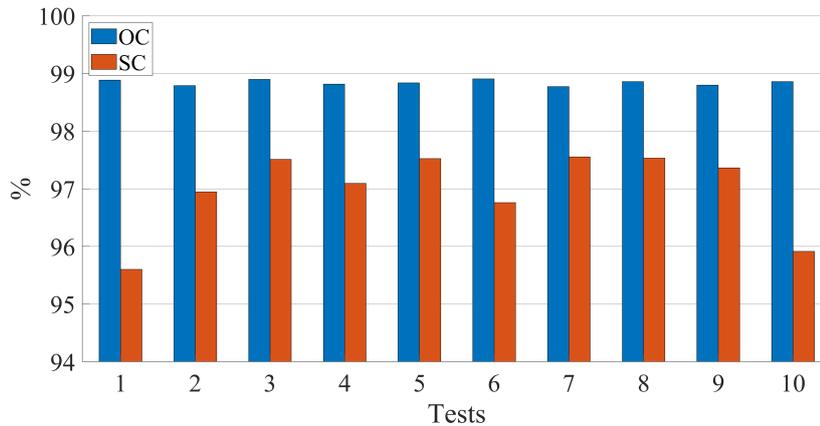


Figura 4.6: Confronto dei PSR degli Status per ciascun test dello scenario controllato, mediato per tutti e dieci i nodi, sia per i casi CO che CS. (Fonte: [4])

Tabella 4.4: Confronto tra i test CO e CS dello scenario controllato.

	Media Complessiva (%)	Varianza (%)
CS	96.98	0.445
CO	98.84	0.002

il test CO, i risultati migliori sono stati ottenuti dal nodo n3, che ha conseguito un tasso di consegna del 83,25% dei messaggi di stato. Al contrario, il nodo dieci ha ottenuto il risultato peggiore, con solo il 60,41% di messaggi di stato recapitati correttamente.

Se si considerano i risultati complessivi ottenuti per il test CS, si può notare che la media tra tutti i nodi ha registrato i risultati migliori per il test CS_4, con il 70,47% di messaggi di stato consegnati correttamente. Il risultato peggiore invece è stato ottenuto dal test CS_6, con il 68,36% di messaggi recapitati correttamente. Per quanto riguarda il test CO, il test CO_7 ha registrato la prestazione migliore, con il 73,65% di messaggi di stato consegnati correttamente, mentre il test CO_8 ha ottenuto il risultato peggiore, con solo il 71,19% dei messaggi recapitati correttamente.

Nello scenario reale, è stato osservato che le prestazioni dei nodi sono state generalmente migliori per il test CO rispetto al suo corrispettivo in CS, come evidenziato nella Fig. 4.7. L'unico nodo che ha mostrato un risultato peggiore nel test CO rispetto a quello CS è stato il nodo due. Tuttavia, con lo scenario controllato, si è osservato che le prestazioni medie di ogni test sono state sempre superiori per il test CO, come evidenziato nella Figura 4.8.

La Tabella 4.7 mostra che il PSR degli Status risultante dal test CO è stato significativamente diverso rispetto a quello del test CS, con una differenza di 2,88 punti. Nel test CO, è stato registrata una media complessiva del 72,14% di ricezione dei messaggi

Tabella 4.5: Risultati complessivi dei PSR degli Status per il test CS dello scenario reale.

ID Test	NODO										Avg. Single Test
	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	
CS_1	67.41	77.59	80.65	76.94	72.96	72.41	58.52	69.54	54.17	55.65	68.58
CS_2	69.00	79.23	80.95	79.53	73.86	70.62	56.94	69.60	51.57	55.83	68.71
CS_3	68.98	77.04	79.54	78.33	73.70	70.37	55.93	67.31	60.74	62.78	69.47
CS_4	64.17	77.31	80.56	78.61	77.04	70.00	59.07	70.19	63.33	64.44	70.47
CS_5	70.37	77.50	78.61	78.15	69.17	74.26	57.69	70.56	52.69	56.11	68.51
CS_6	70.37	78.52	81.30	77.22	69.91	74.17	56.48	70.46	50.28	54.91	68.36
CS_7	70.19	81.67	83.52	79.35	72.69	74.35	58.80	68.15	54.17	59.89	69.68
CS_8	74.17	77.50	80.37	80.19	72.69	73.70	57.69	73.15	52.59	56.20	69.82
CS_9	71.02	77.59	81.39	77.96	76.48	76.11	57.31	70.93	52.22	58.06	69.91
CS_10	71.51	78.45	79.83	78.91	75.39	73.91	54.58	66.05	53.47	58.65	69.07
Node Avg.	69.72	78.24	80.67	78.52	73.39	72.99	57.30	69.59	54.52	57.65	

Tabella 4.6: Risultati complessivi dei PSR degli Status per il test CO dello scenario reale.

ID Test	NODO										Avg. Single Test
	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	
CO_1	74.17	83.15	79.07	81.76	77.04	74.00	69.35	70.74	56.57	62.78	72.86
CO_2	74.26	81.57	78.89	80.19	76.48	72.04	65.37	66.67	56.30	63.15	71.49
CO_3	71.57	79.63	82.78	77.31	75.37	75.46	58.33	72.69	63.24	58.15	71.45
CO_4	73.70	77.22	83.61	81.39	76.20	74.17	60.74	68.98	64.17	58.98	71.92
CO_5	77.96	70.65	82.96	81.20	75.00	71.02	67.31	72.41	61.85	60.09	72.05
CO_6	77.69	72.41	84.35	81.67	75.09	70.56	70.09	73.70	63.24	60.37	72.92
CO_7	77.22	73.43	84.72	81.20	77.13	73.70	70.00	74.26	65.37	59.44	73.65
CO_8	73.80	67.96	84.44	82.31	72.78	73.33	64.07	68.43	64.54	60.28	71.19
CO_9	75.83	67.41	84.07	80.56	72.78	74.35	63.33	67.96	64.91	61.30	71.25
CO_10	78.33	70.74	87.50	83.98	73.15	72.50	64.81	68.89	66.67	59.54	72.61
Node Avg.	75.45	74.42	83.25	81.16	75.11	73.11	65.34	70.47	62.69	60.41	

di Status, mentre nel test CS il valore è stato del 69,26%. Inoltre, la varianza media correlata calcolata per entrambi i test CS e CO è stata rispettivamente dello 0,62% per l'CO e dello 0,46% per l'CS, come indicato nella Tabella [4.7](#).

Per lo scenario controllato e lo scenario reale, è stato eseguito un *t-test di Student* per campioni indipendenti per determinare se ci fossero differenze statisticamente significative tra le medie osservate per CS e CO. Sia per lo scenario controllato che per quello reale le analisi preliminari non hanno mostrato la presenza di valori anomali né per CS

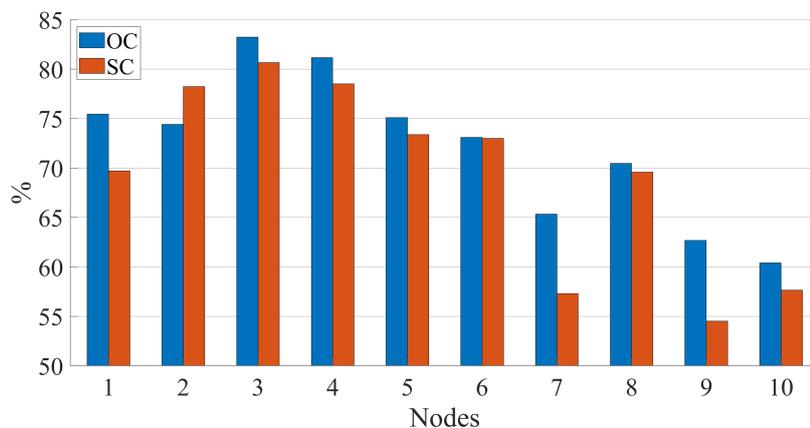


Figura 4.7: Confronto dei PSR degli Status per ciascun test dello scenario reale, mediato per tutti e dieci i test, sia per i casi CO che CS.(Fonte: [4])

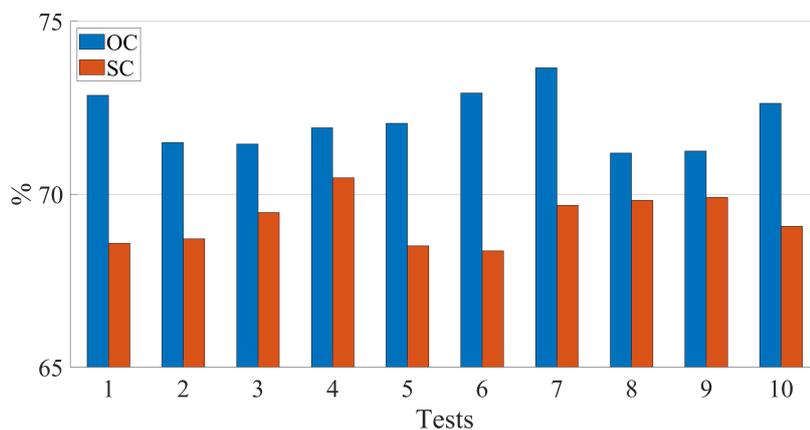


Figura 4.8: Confronto dei PSR degli Status per ciascun test dello scenario reale, mediato per tutti e dieci i nodi, sia per i casi CO che CS.(Fonte: [4])

Tabella 4.7: Confronto tra i test CO e CS dello scenario reale.

	Media Complessiva (%)	Varianza (%)
CS	69.26	0.46
CO	72.14	0.62

né per CO e i dati costituiti dalle medie dei singoli test (Avg Single Test) sono risultati distribuiti secondo una distribuzione normale in entrambi i test, come verificato dal test di Shapiro-Wilk ($p > 0,05$).

Per lo scenario controllato è stata assunta l'ipotesi nulla (la differenza nelle medie di gruppo è pari a zero) . Cioè, supponendo che le medie dei due test siano le stesse, è stato osservato che la differenza tra le medie del test CS e CO risulta $p = 1.48E-07$ ($p <$

0.5), quindi statisticamente significativa, con deviazione standard $\sigma = 0,59$ e $\sigma = 0,40$ rispettivamente. Mentre, per lo scenario reale, è stato osservato che la differenza tra le medie del test CS e CO era $p = 1.46E-07$ ($p < 0.5$), quindi statisticamente significativa con deviazione standard $\sigma = 0,71$ e $\sigma = 0,83$ rispettivamente.

4.5 Conclusioni

In questo capitolo, viene proposta una tecnica di spaziatura dei messaggi di Status nel tempo per superare i problemi della ricezione dei messaggi di conferma all'interno della rete. Per valutare le prestazioni della tecnica proposta in termini di PSR degli Status, è stata allestita una configurazione sperimentale composta da una rete Bluetooth mesh che opera in due diversi scenari: uno controllato con elementi interferenti limitati e uno reale con numerose sorgenti interferenti e attenuanti.

Per evidenziare la differenza tra la procedura standard della rete Bluetooth mesh e la tecnica proposta, sono stati eseguiti due tipologie di test specifici utilizzando diverse configurazioni sperimentali. Il primo test ha impiegato la procedura standard, denominata CS, mentre nel secondo test abbiamo utilizzato la tecnica proposta, denominata CO.

Nel test in ambiente controllato, i risultati complessivi in termini di packet success rate dei comandi di accensione/spengimento mostrano che quanto proposto e implementato nella CO, ha un effetto positivo sul PSR degli Status, con un miglioramento medio del 1,86% rispetto al test CS. A conferma di quanto ottenuto, è possibile osservare una differenza nell'ordine di grandezza della varianza calcolata per entrambi i test: la varianza media calcolata è stata dello 0,44% per il test CS e dello 0,002% per il CO.

Nel contesto reale, è stato ottenuto un miglioramento medio del 2,88% nel test CO rispetto a quello CS. In entrambi i casi, l'aumento delle prestazioni ottenuto con il test CO rispetto a quello CS è significativo come dimostrato dai *test t* condotti. In generale, i PSR complessivi dei test CS e CO nel contesto reale sono inferiori rispetto ai test svolti nel contesto controllato. Ciò è dovuto ai numerosi elementi di interferenza nel contesto reale, come l'interferenza co-channel da parte di un punto di accesso WIFI e altri dispositivi Bluetooth, attenuazione e riflessioni dovuti ad elementi strutturali e di arredamento e la presenza variabile di persone che occupano l'ambiente durante le prove. Queste variabili non sono chiaramente controllabili e rappresentano una fonte di interferenza casuale.

Capitolo 5

Ambiente simulato: PDR e Latenza (TchPDU, scanInterval, SRC-DST)

Anche se i test in ambito reale hanno prodotto risultati positivi, la difficoltà nel controllo delle variabili ambientali interferenti, la complessità dell'architettura e l'elevato numero di parametri di rete che influenzano il comportamento della rete stessa richiedono l'utilizzo di strumenti che garantiscano un elevato livello di affidabilità e ripetibilità dei test. Per questo motivo la ricerca si è concentrata sull'analisi e lo sviluppo di un ambiente di simulazione per le reti Bluetooth Mesh.

La prima attività di ricerca in ambiente simulato ha riguardato l'analisi delle prestazioni di rete in termini di Packet Delivery Ratio (PDR) e Latenza end-to-end, al variare dei parametri temporali coinvolti sia nella fase di scansione che in quella di advertising nelle comunicazioni Bluetooth Mesh.

5.1 TchPDU e scanInterval

Il managed flooding è una tecnica che consente ai messaggi di raggiungere la loro destinazione attraverso percorsi multipli nella rete, migliorando potenzialmente le prestazioni di affidabilità. Tuttavia, questa tecnica può anche aumentare la probabilità di collisione dei pacchetti trasmessi. Per mitigare questo problema e migliorare le prestazioni di affidabilità, il protocollo Bluetooth mesh offre diversi strumenti, tra cui l'introduzione di casualità nei parametri temporali e la trasmissione di repliche del PDU di interesse. I parametri temporali includono quelli specifici dello standard, come quello utilizzato per la temporizzazione dei messaggi di Status, nonché i parametri dei processi di advertising e scanning del BLE su cui si basano le comunicazioni del Bluetooth mesh. Tra questi rientrano il TchPDU utilizzato all'interno degli Advertising Event e lo scanInterval e scanWindow specifici dello Scanning Event.

Il TchPDU è un valore che consente di distribuire le trasmissioni di PDU all'interno dell'Advertising Event. Per rispettare i limiti temporali degli Advertising Event di tipo Non-connectable e Non-scannable Undirect (ADV_NONCONN_IND), il valore massimo del TchPDU è fissato a 10 ms secondo le specifiche della Bluetooth SIG. Tuttavia, aspetti come il range di valori da cui selezionare il TchPDU e la precisione (valori interi nell'ordine dei millisecondi o con precisione microsecondo) sono lasciati a discrezione dello sviluppatore. L'unica raccomandazione sul TchPDU proviene dalla specifica Bluetooth mesh, che suggerisce di randomizzare questo valore per ridurre la probabilità di collisione tra pacchetti.

Per quanto riguarda i parametri della fase di scansione, non ci sono particolari limitazioni nelle specifiche BLE. Tuttavia, secondo le indicazioni contenute nel Bluetooth Mesh Profile v1.0.1 - sez 3.3.1, un dispositivo che utilizza solo l'Advertising Bearer dovrebbe eseguire una scansione passiva con un duty cycle il più vicino possibile al 100% per evitare di perdere eventuali messaggi di rete in arrivo o PDU di provisioning. A livello dei parametri di scansione del BLE, questo si traduce nella coincidenza tra il valore di scanInterval e di scanWindow.

5.2 Analisi proposta

Poiché i parametri menzionati offrono flessibilità di configurazione, è importante studiare l'impatto che differenti configurazioni possono avere sull'affidabilità e sui ritardi end-to-end. L'analisi proposta si basa sulla valutazione delle prestazioni in termini di Packet Delivery Ratio end-to-end e sulla latenza media end-to-end tra coppie sorgente-destinazione (SRC-DST). In particolare, lo studio si è concentrata sull'effetto dell'uso della randomizzazione del parametro TchPDU e l'interdipendenza tra la configurazione degli eventi di scansione e quelli di advertising. Nello specifico è stata analizzata l'interazione tra i TchPDU e scanInterval in differenti scenari.

Per raggiungere tale obiettivo, è stato utilizzato il simulatore per reti Bluetooth mesh della Mathworks, come descritto nel paragrafo di riferimento. Tuttavia, prima di avviare le fasi di test, si è reso necessario apportare alcune modifiche alla librerie riferite al Link Layer. È stato notato che il parametro TchPDU era settato su valori costanti per ogni Advertising Event e per ogni nodo della rete, e che i valori di TchPDU_1 e TchPDU_2 all'interno dell'Advertising Event (Fig. 2.8) erano impostati sullo stesso valore. Per risolvere questa problematica, è stata integrata una funzionalità aggiuntiva che permette di generare valori di TchPDU casuali all'interno di intervalli prefissati, sia tra TchPDU all'interno dello stesso Advertising Event, sia tra Advertising Event differenti, e anche diversi da nodo a nodo. Dall'analisi delle librerie del simulatore, è emerso che quanto suggerito in merito alla scansione passiva è stato implementato con-

siderando il 100% del duty cycle ($\text{scanInterval} = \text{scanWindows}$). Pertanto, nel seguito utilizzeremo il termine scanInterval per riferirci ad entrambi i parametri.

5.3 Setup di simulazione

Senza perdere di generalità, sono state considerate diverse topologie di rete, che differiscono principalmente per il numero di coppie di nodi sorgente-destinazione e per i parametri di rete considerati, piuttosto che per la loro struttura. In particolare, sono state testate reti con 1 (Fig. 5.1a), 2 (Fig. 5.1b), 4 (Fig. 5.1c) e 8 (Fig. 5.1d) coppie di nodi Sorgente-Destinazione (SRC-DST) e copertura radio di 9 m.

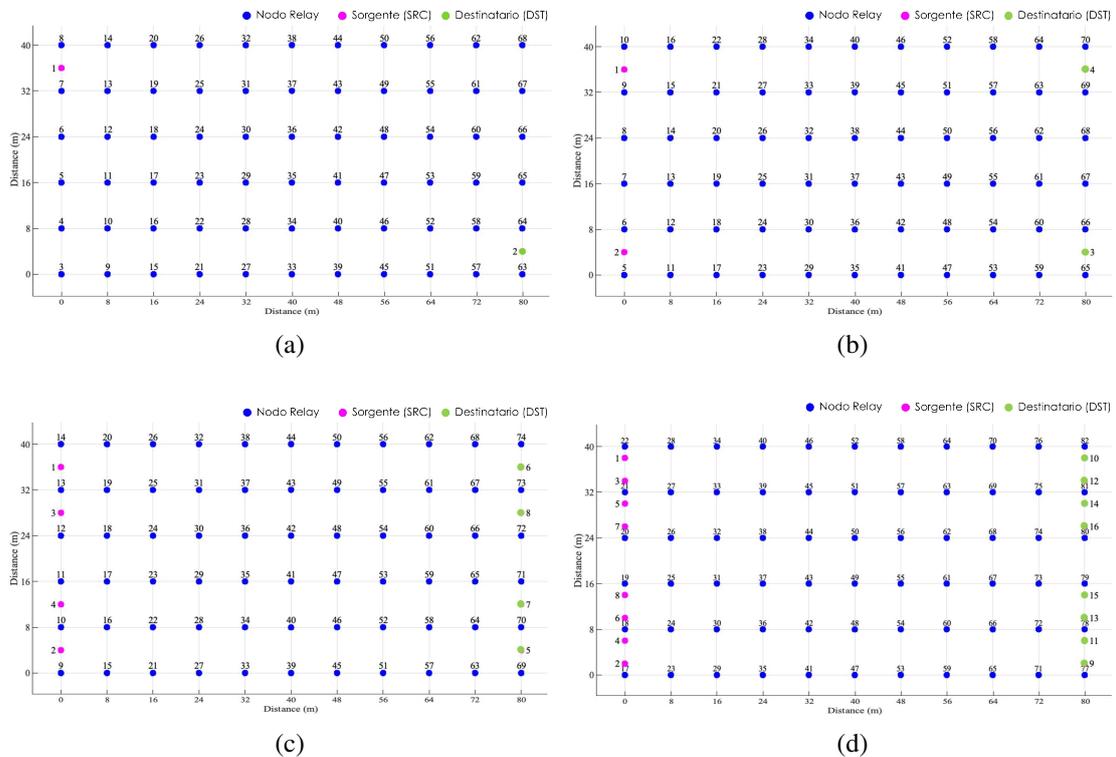


Figura 5.1: Topologie utilizzate nelle simulazioni: (a) singola coppia SRC-DST, (b) 2 coppie SRC-DST, (c) 4 coppie SRC-DST, (d) 8 coppie SRC-DST. (Fonte: [6])

In ogni topologia, le sorgenti sono state posizionate sul lato sinistro della griglia, mentre i nodi riceventi sul lato destro. Le combinazioni tra SRC e DST sono elencate nella Tabella 5.1 e sono state scelte per distribuire maggiormente il carico sui nodi relay.

La distanza tra i nodi, sia orizzontalmente che verticalmente, è di 8 metri, mentre la loro copertura radio di 9 metri. Grazie a questi parametri, ogni nodo relay può trasmet-

tere i propri messaggi ai nodi relay adiacenti nella direzione orizzontale e verticale, ma non in quella obliqua

Tabella 5.1: Parametri di Setup

Parametro	1 SRC	2 SRC	4 SRC	8 SRC
Nodi totali	68	70	74	82
Coppie Sorgente-Destinatario	1-2	1-3; 2-4	1-5; 2-6; 3-7; 4-8	1-9; 2-10; 3-11; 4-12; 5-13; 6-14; 7-15; 8-16
Distanza dei Noi (m)	8			
Raggio di copertura (m)	9			
No. Nodi Relay	66			
TTL	127			
Frequenza messaggi Sorgente	1 p/s			
No. messaggi generati da una SRC	400			
Tempo di ripetizione (ms)	30			
No. Repliche messaggi SRC	1			
Adv_delay	0 ÷ 10			
scanInterval (ms)	10 ÷ 200			
Intervallo TchPDU (ms)	1÷2; 2÷4; 4÷6; 6÷8; 8÷10, 1÷10	1÷2; 1÷10		

Ogni nodo sorgente è configurato per generare 400 messaggi a livello applicativo con una cadenza di 1 pacchetto al secondo. Inoltre, solo i nodi sorgente sono programmati per trasmettere una replica di ogni messaggio generato. L'intervallo di tempo tra l'invio del messaggio originale e la sua ripetizione è impostato a 30 ms. I nodi relay sono responsabili di ritrasmettere ogni messaggio ricevuto una volta, a condizione che superi i controlli di livello superiore dello stack. Non è previsto che venga generata una replica da parte dei nodi relay.

In modo semplice, si può descrivere il comportamento dei nodi in ciascuna simulazione in questo modo: il nodo sorgente genera un messaggio ogni secondo e lo invia ai livelli inferiori per la trasmissione. Dopo 30 millisecondi, viene inviata una replica generata a livello Network. I nodi relay ricevono il messaggio, effettuano i controlli nei diversi livelli dello stack e lo inoltrano ai nodi adiacenti. I nodi relay non creano nuove repliche del messaggio ricevuto, ma si limitano a ricevere e ritrasmettere il messaggio originale. Il nodo destinatario rimane in modalità di scansione per ricevere i messaggi.

Per quanto riguarda il parametro TchPDU ai fini di questa ricerca, si è scelto di utilizzare valori interi dell'ordine dei millisecondi. Il valore minimo ammissibile si è scelto di impostarlo a 1ms per motivi di praticità e coerenza logica. Non sarebbero

stati plausibili intervalli con estremo inferiore pari a zero come 0-10ms o 0-2ms, in quanto il TchPDU rappresenta il tempo tra l'inizio della trasmissione di due PDU consecutivi all'interno dell'Advertising Event, e un valore pari a zero avrebbe comportato la trasmissione di due PDU nello stesso istante su due canali differenti. Per valutare l'impatto del parametro TchPDU sulle prestazioni della rete, è stato deciso di esaminare un intervallo di valori compreso tra 1 e 10 ms, suddividendolo successivamente in sotto-intervalli più specifici: $1 \div 2\text{ms}$, $2 \div 4\text{ms}$, $4 \div 6\text{ms}$, $6 \div 8\text{ms}$ e $8 \div 10\text{ms}$, al fine di condurre una valutazione più approfondita e selettiva.

Le performance della rete nelle diverse topologie è stata valutata non solo in termini del numero di coppie SRC-DST e del TchDPU, ma anche in termini di variazione del parametro che definisce l'intervallo di scansione (scanInterval). I suoi valori sono fatti variare tra 10ms e 200ms. Gli altri parametri utilizzati sono elencati nella Tabella [5.1](#) e vengono mantenuti invariati per tutti i test eseguiti.

5.4 Risultati

In questa sezione, verranno presentati e discussi i risultati delle varie simulazioni. Ogni test è associato a un grafico che mostra il valore del Packet Delivery Ratio in percentuale e la Latenza in secondi sulle ordinate e sull'ascissa il valore dell'intervallo di scansione considerato per ogni ciclo di simulazione. Per facilitare la presentazione dei dati raccolti, la sezione è stata divisa in due sottosezioni: la prima inerente la topologia con singola coppia di SRC-DST e la seconda ai risultati delle simulazioni multisorgente.

5.4.1 Packey Delivery Ratio e Latenza per singola coppia Sorgente-Destinatario

Le simulazioni condotte per valutare il Packet Delivery Rate (PDR) e la latenza in relazione a diverse combinazioni di TchPDU e scanInterval nel caso di topologia con singola coppia SRC-DST hanno prodotto i risultati illustrati nella Fig. [5.2](#). Le curve a tratto continuo rappresentano i valori di PDR per ogni intervallo di TchPDU al variare dello scanInterval, mentre le linee tratteggiate rappresentano le latenze medie end-to-end ottenute nelle stesse simulazioni per gli stessi intervalli di TchPDU.

Analizzando il PDR, si nota un comportamento comune a tutte le curve: una fase di salita più o meno rapida seguita da una stabilizzazione in cui i valori si attestano tra il 98% e il 100%. Nella parte sinistra del grafico, tra 0 e i 50 ms di ascissa, si può osservare come i valori di PDR e l'inizio dei plateau dipendano dalla scelta degli intervalli di TchPDU.

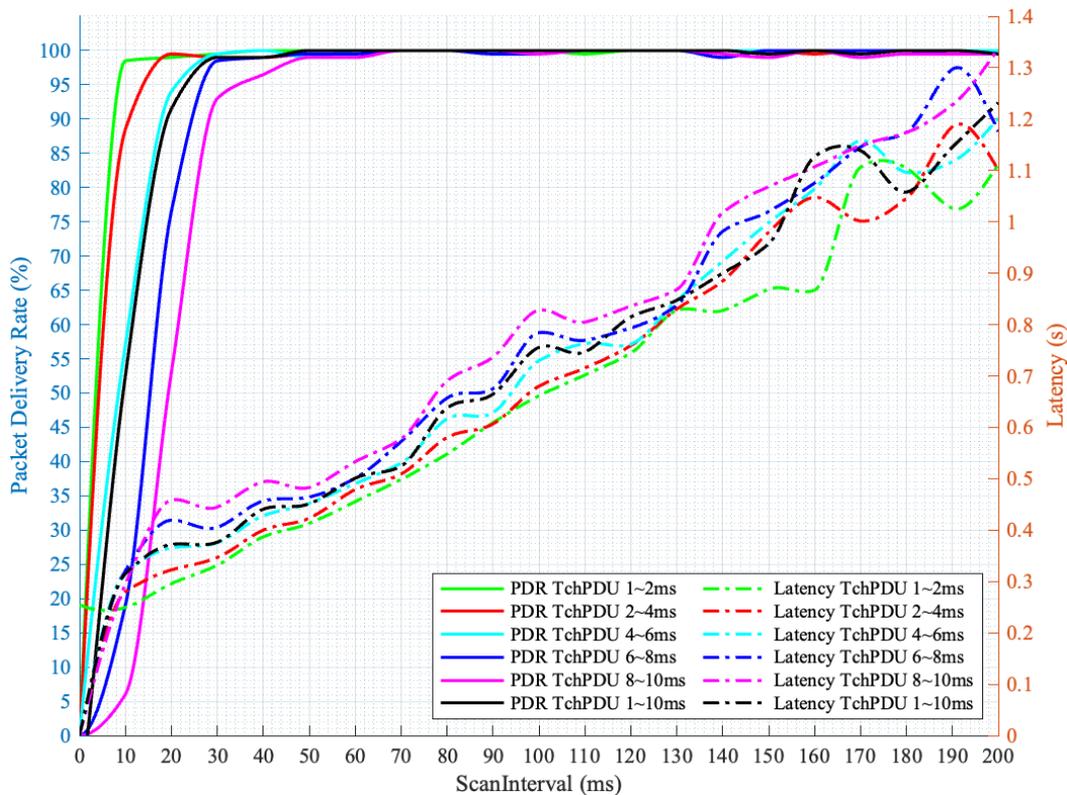


Figura 5.2: Packet Delivery Ratio e Latenza end-to-end per una singola coppia Sorgente-Destinatario al variare dei parametri temporali scanInterval e TchPDU

Per l'intervallo 1÷2 ms (curva verde), si ottengono PDR del 98% per valori di scanInterval molto bassi (10 ms), mantenendosi sopra il 98% per il resto della curva. Per valori tra 2 e 4 ms, il PDR ha un andamento molto simile, ma si osserva una diminuzione della percentuale al valore di 10 ms di scanInterval, attestandosi all'87-88%, mentre raggiunge il 99% per scanInterval di 20 ms.

Per l'intervallo 4-6 ms, la pendenza del tratto ascendente della curva del PDR è minore rispetto alle due precedenti, con valori percentuali del 55% e del 94%, rispettivamente per scanInterval di 10 e 20 ms. Solo con un'ascissa di 30 ms, il PDR supera il 99% e rimane sopra questo valore per scanInterval successivi.

Dall'analisi effettuata, emerge che all'aumentare degli estremi dell'intervallo in cui si prelevano i valori di TchPDU, aumenta anche il valore di scanInterval necessario per raggiungere un tasso di successo pari o superiore al 98% nella consegna dei pacchetti. Ad esempio, per l'intervallo 8-10 ms (rappresentato in viola), si ottiene un valore pari o superiore al 98% solo per valori di scanInterval maggiori di 50 ms.

Infine, resta da analizzare la curva nera, relativa all'intervallo di TchPDU compreso tra 1 e 10 ms. Osservando attentamente il grafico della Fig. 5.2, possiamo notare che questa curva segue l'andamento della curva corrispondente all'intervallo centrale di

4-6 ms. Poiché è stata utilizzata una funzione di generazione dei valori di TchPDU random uniformemente distribuita, gli intervalli 4÷6 ms e 1÷10 ms presentano un valore medio molto simile, rispettivamente 4,5 ms e 5 ms. Ciò dimostra perché le due curve presentano un PDR quasi identico.

Se estendiamo questo ragionamento anche ad altri intervalli di TchPDU, calcolando la media dei valori medi di ogni intervallo, otteniamo una quantità di circa 5 ms. Per questo motivo, l'andamento della curva nera può essere considerato una media degli andamenti di tutte le altre curve.

Il grafico presente in Fig. 5.2 riporta anche l'andamento delle latenze medie end-to-end tra la sorgente e il destinatario per diversi intervalli di TchPDU, al variare dello scanInterval. Si può notare un andamento comune: la latenza aumenta linearmente all'aumentare dello scanInterval per ogni intervallo di TchPDU. Questo risultato è la conseguenza dell'implementazione della gestione di ricezione/trasmissione dei messaggi da parte dei nodi relay del simulatore, come spiegato nel paragrafo 5.5.

Se consideriamo valori fissi di scanInterval e prendiamo i campioni di latenza corrispondenti, possiamo individuare dei fattori di crescita della latenza legati allo scanInterval. Ad esempio, confrontando le latenze di scanInterval 100ms e 200ms con quelle ottenute a 10ms, otteniamo fattori di crescita media del 143% e del 292%, rispettivamente.

È importante sottolineare che la scelta dello scanInterval deve essere ponderata non solo in base al valore di PDR, ma anche in base alla latenza, poiché valori di scanInterval crescenti generano scostamenti sostanziali sulla latenza end-to-end. Inoltre, notiamo che all'aumentare degli estremi degli intervalli di TchPDU aumenta anche la latenza a parità di scanInterval. Ad esempio, con un'ascissa di 20ms, il valore di latenza per l'intervallo 1÷2ms è 0,2956 s, mentre per l'intervallo 8÷10ms è 0,4584 ms. Tale scostamento corrisponde ad un aumento del 55%.

Infine, per quanto riguarda i TchPDU dell'intervallo 1÷10 ms, la curva nera tratteggiata approssima mediamente l'andamento di tutti gli altri intervalli, posizionandosi centralmente rispetto alle altre e ricalcando la curva dell'intervallo 4÷6ms.

5.4.2 Performance per topologie multisorgente

Sulla base dei risultati ottenuti dalle simulazioni della topologia a singola sorgente, si è deciso di condurre un'analisi e una comparazione con soli due intervalli di TchPDU per le topologie multi-sorgenti: 1÷2 ms e 1÷10 ms. La scelta del primo intervallo è stata motivata dal fatto che, a parità di scanInterval, ha mostrato il PDR più favorevole. Il secondo intervallo, invece, è stato scelto sia per ottenere un'analisi generale dei comportamenti delle reti multi-sorgente, sia perché la presenza di più sorgenti che

immettono messaggi all'interno della rete aumenta il rischio di collisioni. Ampliare l'intervallo di TchPDU e renderlo più variabile può mitigare questo problema.

Ogni figura è composta da due grafici. Il primo nella parte superiore si riferisce ai test effettuati con l'intervallo $1 \div 2$ ms, mentre il secondo rappresenta i test con quello $1 \div 10$ ms. Le Fig. 5.3, 5.4 e 5.5 rappresentano rispettivamente i risultati delle topologie con 2, 4 e 8 coppie di SRC-DST.

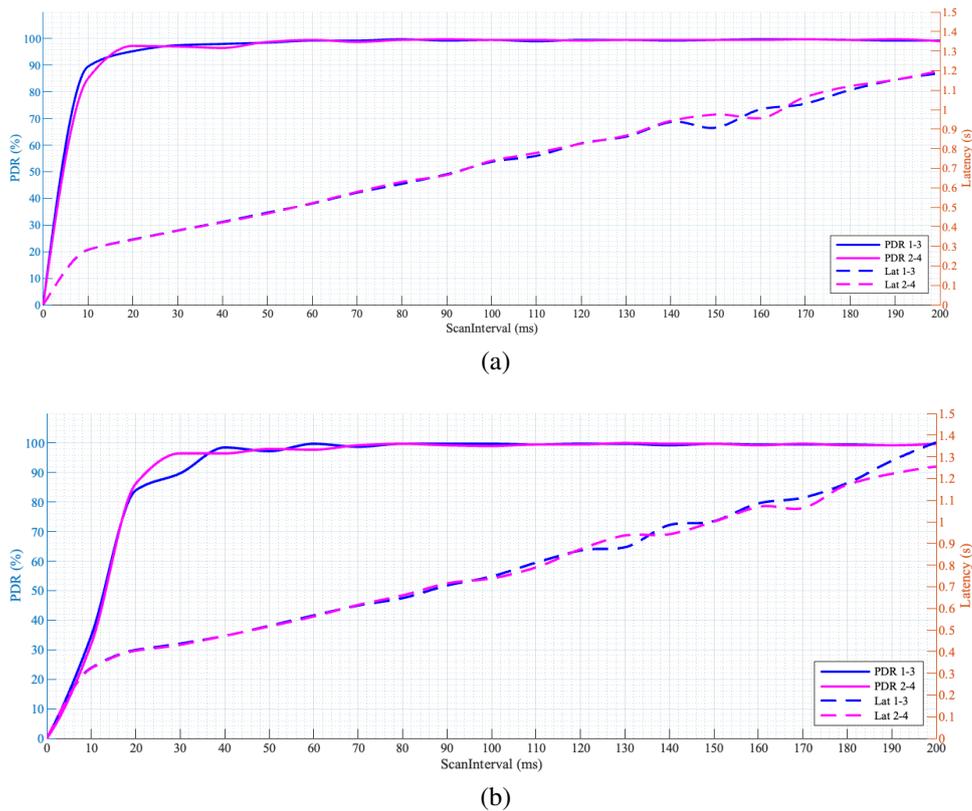
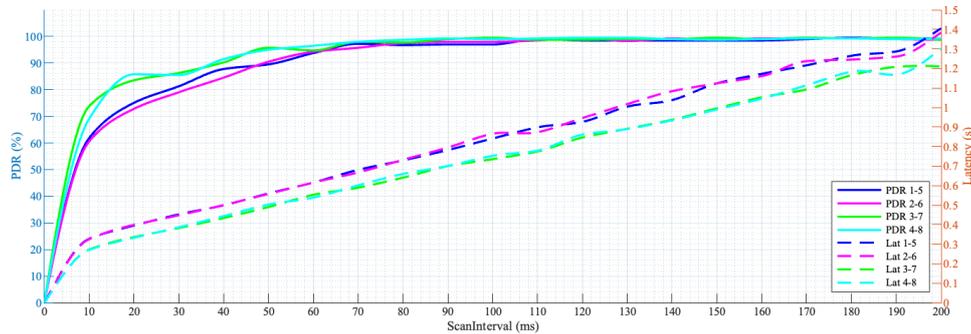


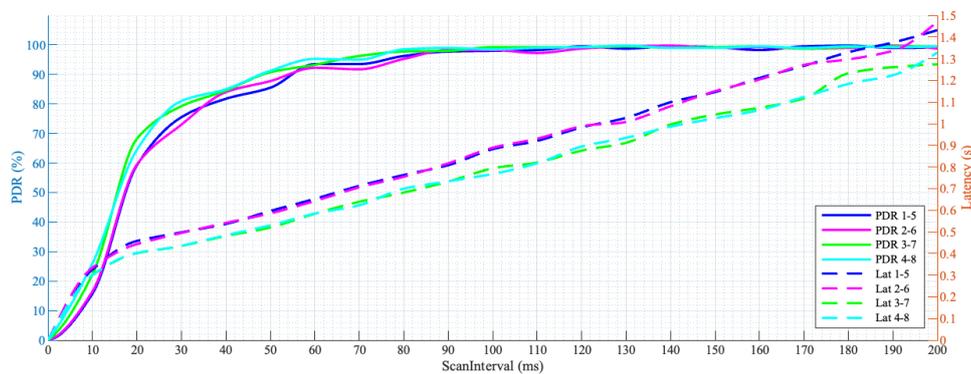
Figura 5.3: Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST: (a) intervallo TchPDU $1 \div 2$ ms, (b) intervallo TchPDU $1 \div 10$ ms

In generale, le simulazioni effettuate mostrano che all'aumentare del numero di sorgenti nella rete, i dispositivi relay richiedono un intervallo di scansione sempre maggiore per garantire un PDR elevato, compreso tra il 97% e il 100%, indipendentemente dal parametro TchPDU. Tuttavia, un valore di scanInterval più elevato comporta un aumento della latenza. Ad esempio, nel caso con 4 coppie SRC-DST, per ottenere un PDR prossimo al 100% è necessario scegliere un valore di scanInterval pari a 90 ms, ma questo implica una latenza tra 720 ms e 820 ms, a seconda dell'intervallo di TchPDU scelto.

Nel caso di 8 sorgenti (Fig. 5.5), la situazione peggiora significativamente: i valori di PDR più elevati, compresi tra il 94% e il 98%, si ottengono con uno scanInterval di 170



(a)



(b)

Figura 5.4: Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST:
 (a) intervallo TchPDU 1÷2 ms, (b) intervallo TchPDU 1÷10 ms

ms, comportando una latenza compresa tra 1.35 s e 1.75 s, a seconda dell'intervallo di TchPDU scelto.

Per quanto riguarda il confronto tra i due intervalli di TchPDU, si nota che l'utilizzo di valori compresi tra 1 ms e 2 ms non garantisce un'ottimizzazione del PDR come nel caso di una singola sorgente. Inoltre, all'aumentare del numero di coppie SRC-DST, questa differenza diventa sempre meno significativa. L'unica eccezione risulta nel caso con 2 sorgenti, in cui l'intervallo 1÷2 ms (Fig. 5.5a) riesce a garantire prestazioni migliori a valori di scanInterval più bassi rispetto all'intervallo 1÷10 ms (Fig. 5.5b), ma con valori di PDR ben al di sotto del 90%.

In merito alle latenze, dai grafici è evidente che il valore dello scanInterval ha un forte impatto, poiché la latenza cresce linearmente all'aumentare di questo parametro di scansione. Inoltre, si osserva una correlazione anche con il numero di sorgenti presenti nella rete. In particolare, si nota che passando da una topologia con 2 coppie SRC-DST ad una con 8 coppie, i valori massimi di latenza aumentano dal valore di 1,35 secondi a quello di 2,1 secondi, corrispondente ad un incremento percentuale del 55%.

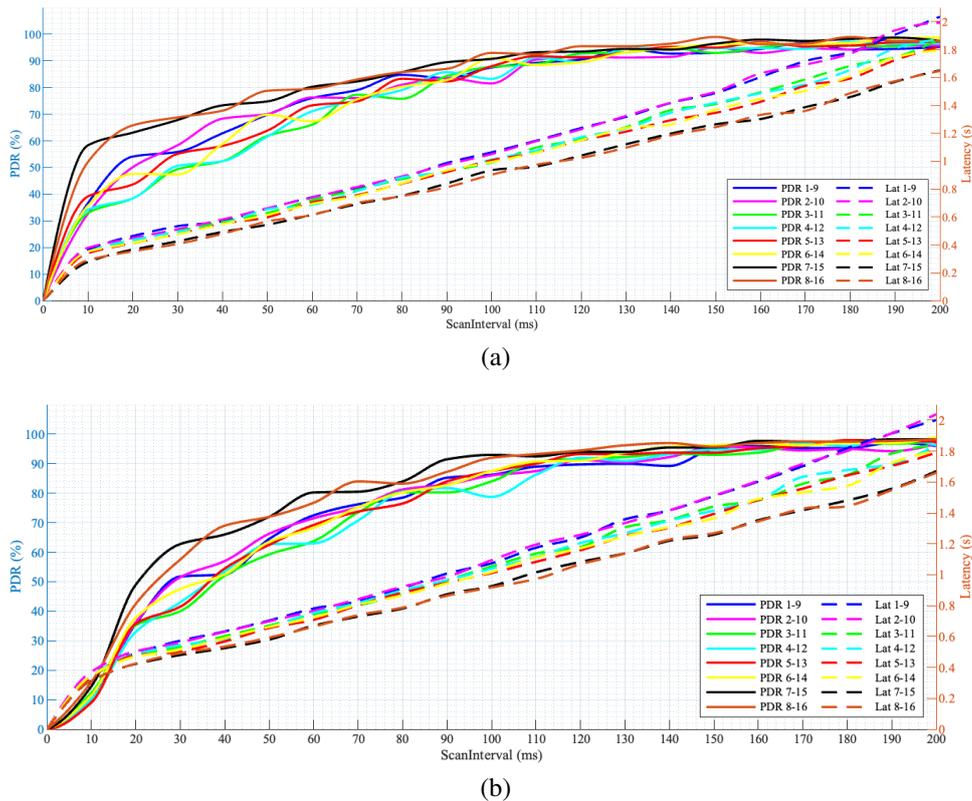


Figura 5.5: Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST:
 (a) intervallo TchPDU 1÷2 ms, (b) intervallo TchPDU 1÷10 ms

5.5 Implementazione secondo Mathworks

Per capire la natura dei valori delle latenze e il legame con il parametro scanInterval, è necessario approfondire l'analisi della struttura e le scelte implementative adottate dalla Mathworks.

All'interno del simulatore, ogni nodo relay è stato progettato per eseguire l'attività di advertising solo al termine di ogni scanInterval, qualora fosse necessario. In altre parole, i nodi relay che si trovano in fase di scansione registrano nella coda di advertising i messaggi idonei da inoltrare, senza inviarli immediatamente. Essi continuano la scansione sullo specifico canale fino al termine dell'intervallo di scansione. Una volta raggiunto il termine dell'intervallo, i nodi passano alla modalità di advertising e inviano tutti i messaggi memorizzati.

Nella Fig. 5.6 è stata riportata a titolo di esempio una rappresentazione semplificata di una possibile trasmissione di messaggi tra un nodo sorgente (SRC) e un nodo destinatario (DST) attraverso due nodi relay (Relay_1 e Relay_2).

Possiamo dividere la propagazione del messaggio in quattro step, una per ogni nodo:

1. Il nodo SRC genera un messaggio e lo incapsula in un PDU di tipo

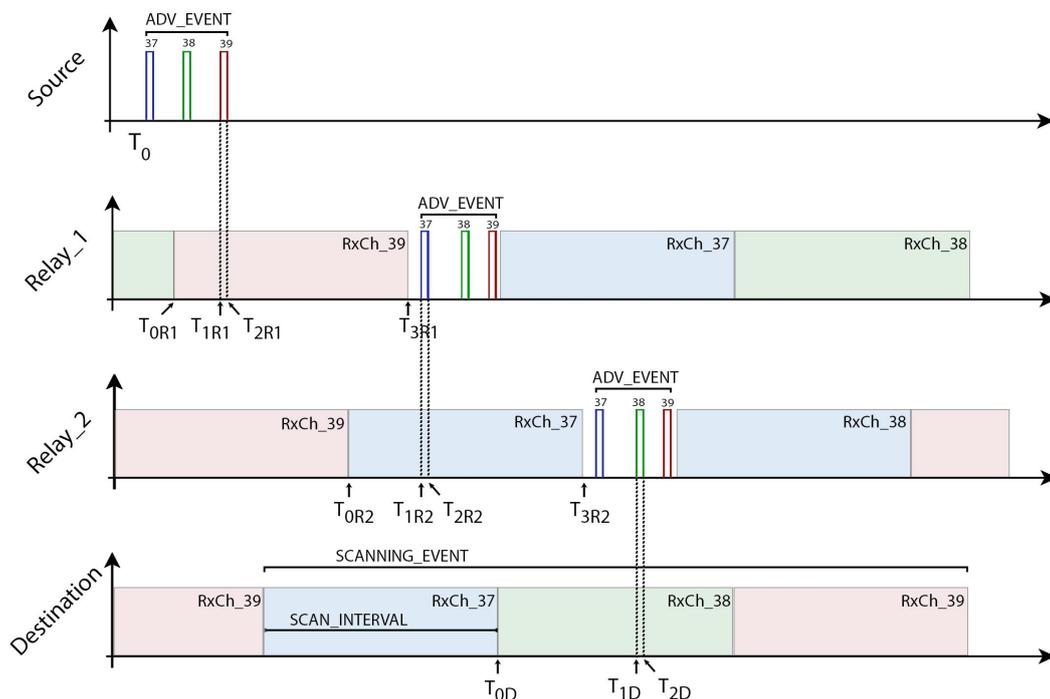


Figura 5.6: Esempio di propagazione dei messaggi all'interno del simulatore della Mathworks tra un nodo sorgente (SRC) e un nodo destinatario (DST), separati da due nodi relay intermedi.

ADV_NONCONN_IND. Questo PDU viene trasmesso attraverso un Advertising Event nei tre Primary Advertising Channel (37, 38, 39) a partire dall'istante T_0 . Tra un canale e l'altro vengono applicati i ritardi T_{chPDU_1} e T_{chPDU_2} .

2. Al tempo T_{0R1} , Relay_1 completa la scansione del canale 38 e si mette in ascolto del canale 39 (RxCh 39). Successivamente, al tempo T_{1R1} , riceve il messaggio inviato dal nodo SRC. Secondo l'implementazione riscontrata nel simulatore MathWorks, Relay_1 non interrompe la scansione per trasmettere il messaggio, ma la porta a termine fino al tempo T_{3R1} , tempo in cui entra in modalità advertising. Terminato l'Advertising Event, Relay_1 torna in modalità di scansione.
3. Relay_2 riceve tra T_{1R2} e T_{2R2} il messaggio proveniente da Relay_1 attraverso la scansione del canale 37. Anche in questo caso, Relay_2 passerà in modalità di trasmissione solo al termine della scansione del canale (T_{3R2}).
4. L'ultimo passaggio della trasmissione del messaggio è la ricezione da parte del nodo DST, che avviene all'istante T_{1D} durante l'intervallo di scansione del canale 38.

Questa implementazione introduce dei ritardi nella trasmissione del messaggio dal nodo di origine (SRC) al nodo di destinazione (DST), che dipendono non solo dai

T_{chPDU} , eventuali tempi di elaborazione, switch di canale e dal tempo di utilizzo del canale, ma anche dal tempo necessario per completare la fase di scansione dopo la ricezione del messaggio da parte del nodo.

Analizzando la figura, possiamo calcolare il tempo totale di latenza, ovvero il tempo che intercorre tra l'inizio della trasmissione del messaggio dalla sorgente e l'istante in cui il messaggio viene ricevuto dal nodo destinatario come segue

$$Latenza = T_{chPDU1_SRC} + T_{chPDU2_SRC} + (T_{3R1} - T_{2R1}) + (T_{3R2} - T_{2R2}) + T_{chPDU1_R2} \quad (5.1)$$

Si tratta di un calcolo di massima, che ha solo scopo indicativo ed esplicativo. Esula da questa trattazione fornire un calcolo dettagliato considerando ogni singolo elemento temporale in gioco nella trasmissione.

I valori di interesse sono $(T_{3R1} - T_{2R1})$ e $(T_{3R2} - T_{2R2})$, i quali dipendono dall'istante di ricezione del messaggio e dalle dimensioni dello $scanInterval$. La latenza sarà maggiormente influenzata se il messaggio viene ricevuto all'inizio dello $scanInterval$. Invece, se il messaggio viene ricevuto alla fine dell'intervallo di scansione, il ritardo da aggiungere sarà minore.

La posizione dell'istante di ricezione non è l'unico fattore che influenza l'ampiezza dei ritardi. Anche la dimensione dello $scanInterval$ svolge un ruolo importante. Infatti, se lo $scanInterval$ è più ampio, ci sono maggiori probabilità di avere intervalli più ampi tra l'istante di ricezione e la fine della scansione, il che ha un impatto maggiore sulla latenza.

Questo effetto è amplificato all'aumentare del numero di nodi relay tra la sorgente e la destinazione. Più nodi ci sono, maggiori sono gli intervalli tra l'istante di ricezione e l'inizio della fase di advertising da considerare. Inoltre, se un nodo relay riceve più pacchetti nello stesso $scanInterval$, durante la fase di trasmissione deve effettuare Advertising Event consecutivi, introducendo un ulteriore ritardo dovuto all' $advDelay$ tra i diversi Advertising Event [2.1.3](#).

L'implementazione della gestione della fase di scansione e advertising dei nodi relay, come quella realizzata da Mathworks, consente di ottenere prestazioni molto elevate in termini di PDR. Tuttavia, ciò ha forti ripercussioni sulla latenza.

5.6 Conclusioni

In questo capitolo viene presentata un'analisi delle prestazioni di rete nel contesto delle comunicazioni Bluetooth Mesh, concentrandosi sui parametri temporali coinvolti sia nella fase di scansione che in quella di advertising. In particolare, l'analisi si

focalizza sull'effetto dell'uso della randomizzazione del parametro TchPDU e sull'interdipendenza tra la configurazione degli eventi di scansione e quelli di advertising. Utilizzando il simulatore per reti Bluetooth Mesh della Mathworks, lo studio esamina l'interazione tra i TchPDU e l'intervallo di scansione in differenti scenari. I risultati dei test mostrano che i parametri temporali analizzati hanno un impatto significativo sulle prestazioni di rete, soprattutto in relazione alla numerosità delle sorgenti considerate. Inoltre, l'analisi del simulatore e dei test condotti dimostra che le scelte implementative adottate dalla Mathworks hanno un forte impatto sui tempi di risposta della rete, traducendosi in elevati valori di latenza end-to-end.

Ciò che emerge dall'analisi evidenzia come un'implementazione dello standard di questo tipo possa limitarne l'utilizzo in applicazioni in cui bassi valori di latenza sono un requisito fondamentale.

Capitolo 6

Nuova implementazione dello standard

La specifica del Bluetooth mesh definisce le regole e le specifiche tecniche che un dispositivo Bluetooth Low Energy deve seguire per integrare il profilo mesh e conformarsi al protocollo. Tuttavia, l'analisi del contenuto dello standard mostra che c'è un certo grado di libertà interpretativa, il che può portare a una diversificazione delle possibili implementazioni sotto alcuni aspetti. Pur essendo tutte valide, queste implementazioni possono avere caratteristiche e prestazioni diverse.

Sfruttando questi gradi di libertà interpretativa e basandosi su quanto emerso dai risultati dei test presentati nel capitolo precedente, è stata realizzata una nuova implementazione dello standard. In particolare, si è lavorato sulla gestione dei messaggi nella fase di scansione e advertising nei nodi relay, al fine di ottenere una rete più reattiva e soddisfare le esigenze di settori applicativi in cui i bassi valori di latenza sono un requisito fondamentale.

Al fine di valutare come le differenti implementazioni dello standard Bluetooth mesh possono influire sull'affidabilità della rete, viene presentato un confronto delle prestazioni in termini di Packet Delivery Ratio e latenza end-to-end tra l'implementazione proposta e quella già presente nel Bluetooth Toolbox di Mathworks. Vengono utilizzate le stesse metriche del capitolo precedente, ovvero il Packet Delivery Ratio (PDR) e la latenza end-to-end tra le coppie sorgente-destinazione. Il confronto viene effettuato su diverse topologie di rete, prendendo in considerazione il numero di sorgenti di trasmissione e il raggio di copertura. Inoltre, si tiene sempre in considerazione l'intervallo di scansione dei nodi relay, il quale è strettamente correlato alle implementazioni esaminate.

6.1 L'implementazione proposta

Grazie al supporto del reparto di sviluppo del Mathworks Bluetooth Toolbox, è stata realizzata un'implementazione dello standard alternativa a quella presente nel Bluetooth Toolbox. Questa nuova implementazione prevede il principio di prelazione nella gestione dei messaggi nelle fasi di scansione e advertising nei nodi relay.

Nell'implementazione senza prelazione della gestione dei messaggi nella fase di scannig e advertising nei nodi relay, analizzata nel capitolo precedente, la scansione dei nodi relay è prioritaria rispetto all'evento di advertising. I nodi eseguono la scansione di un singolo canale di trasmissione per un intervallo di tempo stabilito dal parametro di scanInterval. Qualsiasi messaggio ricevuto durante questo intervallo viene inviato ai livelli superiori dello stack, analizzato e, se ritenuto idoneo per essere inoltrato, viene inserito nella coda di ritrasmissione. Alla scadenza dell'intervallo di scansione, se la coda di ritrasmissione contiene dei messaggi, il nodo passa alla modalità di advertising e li trasmette. In caso contrario, il nodo inizia la scansione del canale di trasmissione successivo.

La proposta di gestire la scansione e l'advertising attraverso un principio di prelazione si basa sull'idea di privilegiare la fase di ritrasmissione a discapito della finestra di scansione. In altre parole, l'obiettivo è di effettuare la ritrasmissione nel minor tempo possibile. I nodi relay effettuano la scansione dei canali trasmissivi, commutando da uno all'altro sulla base dello scanInterval. Tuttavia, quando ricevono un messaggio adatto per essere inoltrato, passano immediatamente alla modalità advertising e inviano il messaggio ai nodi all'interno del loro raggio di copertura. Dopo l'invio dei dati i nodi riprendono la scansione dei canali. Questo meccanismo di switch repentino si ripete per ogni messaggio idoneo all'inoltro.

Si noti che la ripresa della fase di scansione, dopo un evento di advertising, sia nell'implementazione con prelazione che senza, riparte sempre dal canale 37 e per un intervallo di scansione pari a quello indicato da scanInterval. Nel caso in cui non si verificano Advertising Event durante la scansione, il simulatore rispetta l'ordine dei canali definito in un parametro specifico. Nel caso in esame, l'ordine è 37-38-39, ma è possibile modificarlo a piacere.

La Fig. 6.1, riportata a titolo esemplificativo, permette di comprendere più facilmente la logica delle due implementazioni. Viene mostrata la trasmissione di tre messaggi attraverso tre differenti nodi generici (Nodo 1, Nodo 2, Nodo 3) e la presenza di due nodi relay su cui sono implementati la gestione senza prelazione e quella con prelazione, rispettivamente nodo A e nodo B. L'intervallo effettivo di scansione, scanInterval (TscnInter), è identico per entrambi i nodi relay. Per semplificare la trattazione, i nodi A e B non sono nel reciproco raggio di copertura.

Il nodo A si trova inizialmente nella fase di scansione sul canale 37 e all'istante T0

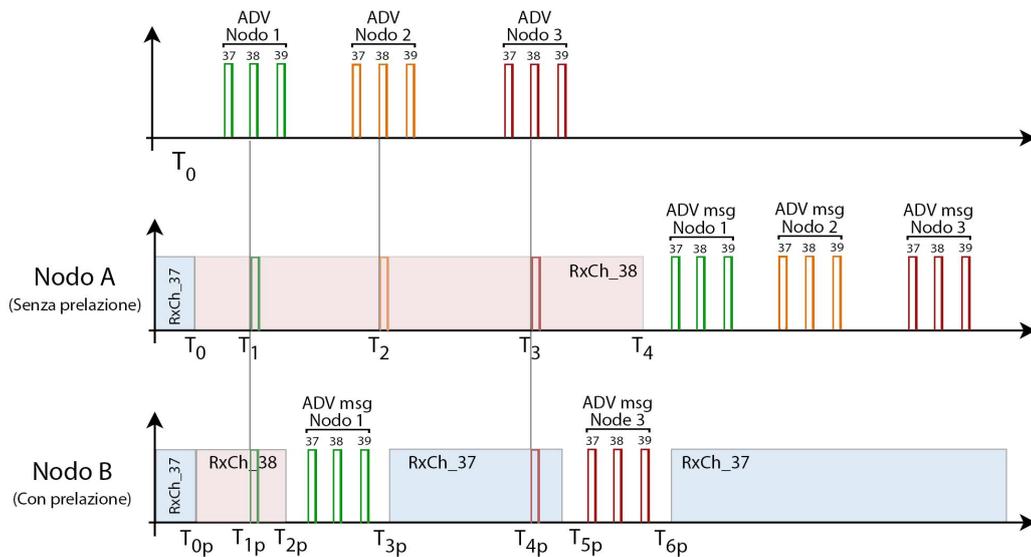


Figura 6.1: Confronto tra la fase di scansione e advertising dei nodi relay dell'implementazione senza prelazione e dell'implementazione con prelazione. (Fonte: [6])

commuta sul canale 38. Trattandosi di un nodo senza prelazione, deve mantenere l'ascolto per un intero tempo di $T_{scnInter}$, cioè fino alla fine dell'intervallo di scansione. I messaggi ricevuti all'istante T_1 , T_2 e T_3 non interferiscono con la modalità operativa del nodo, ma vengono semplicemente elaborati dai livelli superiori dell'architettura ed eventualmente inseriti nella coda di ritrasmissione. All'istante T_4 , la fase di scansione si interrompe e il nodo avvia gli eventi di advertising per i messaggi contenuti nella coda di ritrasmissione. Nell'esempio, tutti e tre i messaggi necessitano di essere inoltrati. Al termine della trasmissione dei messaggi, il nodo A torna in modalità scanning ripartendo dal canale 37.

Il nodo B ha un comportamento diverso. Come nel caso precedente, il nodo B inizia l'ascolto sul canale 38 all'istante T_{0p} . Sebbene la sua finestra di scansione debba terminare dopo un tempo $T_{scnInter}$, la ricezione del messaggio proveniente dal Nodo 1 e, adatto ad essere inoltrato, provoca la permutazione della modalità operativa da Scanning ad advertising. Successivamente, all'istante T_{3p} , il nodo riprende la scansione sul canale 37. Anche in questo caso, la ricezione del messaggio del Nodo 2 (T_{4p}), interrompe la scansione e porta in modalità trasmissiva il Nodo B all'istante T_{5p} . Dopo l'evento di advertising il nodo relay B riprende la scansione sul canale 37.

La Fig. 6.1 e la spiegazione fornita sopra costituiscono solo un esempio della logica alla base delle due implementazioni. Non è possibile coprire in modo esaustivo ogni singolo evento che potrebbe verificarsi a causa della casualità di alcuni parametri e del loro effetto sulla frequenza e sulla durata degli eventi. Ciò aumenta notevolmen-

te il numero di combinazioni possibili che possono influenzare il comportamento del sistema.

6.2 Setup di simulazione

Il confronto tra l'implementazione con prelazione e quella senza prelazione è stato effettuato utilizzando le stesse topologie di rete multi-sorgente delle figure 5.1b, 5.1c e 5.1d. Tuttavia, sono state apportate alcune variazioni ai parametri di rete della Tabella 5.1, tra cui il raggio di copertura trasmissiva (R), che è stato considerato sia di 9m che di 16m, e l'intervallo TchPDU scelto solamente nell'intervallo tra 1 e 10 ms.

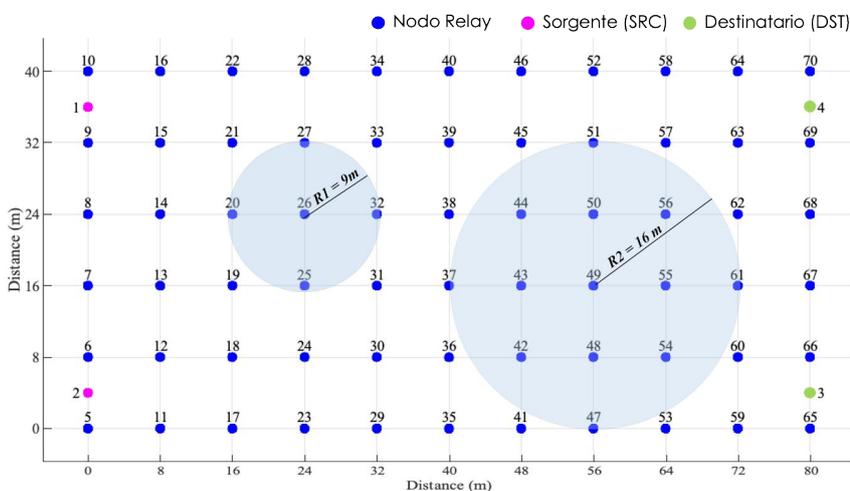


Figura 6.2: Topologia con 2 coppie sorgente-destinazione ed esempio di copertura. (Fonte: [6])

Il test multi-sorgente descritto nel capitolo 5.4.2 ha evidenziato che all'aumentare del numero di coppie sorgente-destinatario, i relay della rete richiedono uno scanInterval sempre più ampio per garantire un PDR prossimo al 100% e ciò comporta un aumento della latenza fino a circa 2 secondi. Inoltre, è stato osservato che l'utilizzo di un intervallo di TchPDU tra 1 e 2ms non fornisce miglioramenti sostanziali rispetto a quello tra 1 e 10ms. Pertanto, al fine di ridurre il rischio di collisioni tra le trasmissioni dei nodi della rete, nei test eseguiti per esaminare le due implementazioni è stato deciso di considerare i TchPDU solo nell'intervallo compreso tra 1 e 10ms.

La Fig. 6.2 mostra un esempio delle coperture radio in esame. Con un raggio di 9 m, le trasmissioni di un nodo relay, come ad esempio il 26, possono raggiungere solo i nodi che si trovano a un solo hop sugli assi longitudinali e trasversali. Invece, con un raggio di 16 m, il numero di nodi raggiunti dai messaggi trasmessi da un qualsiasi nodo relay, come il 46, aumenta notevolmente. Una singola trasmissione può raggiungere nodi che altrimenti richiederebbero due ritrasmissioni da differenti nodi relay. Come

vedremo in seguito, la scelta del raggio di copertura radio influisce significativamente sulle prestazioni della rete in ogni topologia considerata. Va precisato che l'utilizzo della Fig. 6.2 è puramente indicativo e che nelle diverse simulazioni di questa topologia, le coperture radio sono mantenute uguali per ogni nodo.

6.3 Risultati

In questa sezione vengono presentati e discussi i risultati delle varie simulazioni. Ad ogni test è associato un grafico che mostra il Packet Delivery Ratio e la latenza in ordinata, e in ascissa il valore dell'intervallo di scansione considerato per ogni ciclo di simulazione. Ogni figura è composta da due grafici. Il primo, nella parte superiore di ogni figura, si riferisce ai test eseguiti in reti con nodi relay senza prelazione, mentre il secondo rappresenta i test con gestione della ritrasmissione con prelazione.

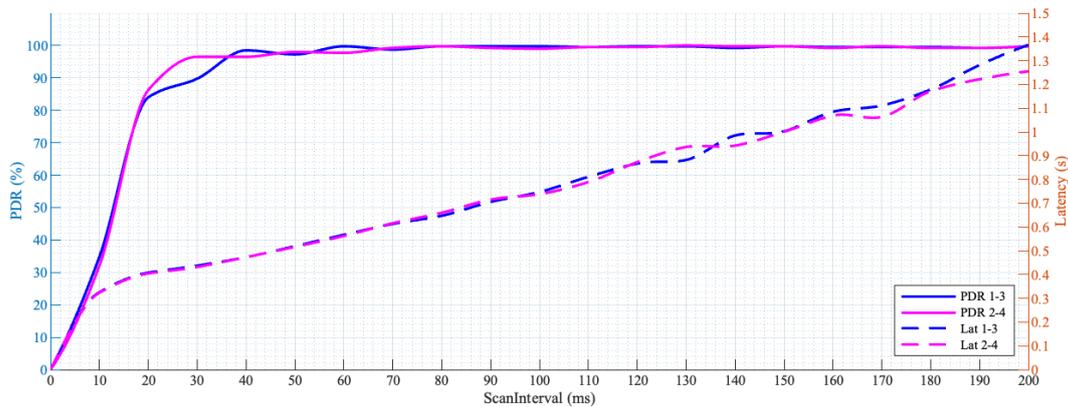
Per facilitare la presentazione dei dati raccolti, la sezione è stata suddivisa in due sottosezioni in base al raggio di copertura considerato nelle simulazioni.

6.3.1 Test con $R = 9$ m

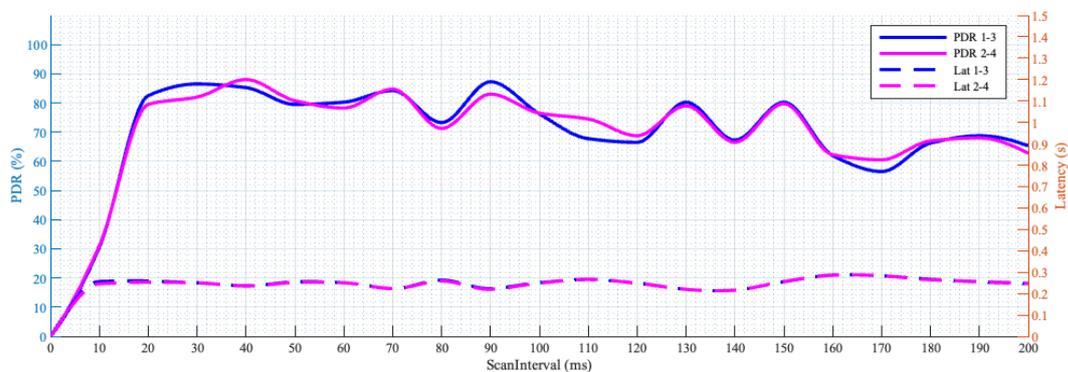
Sulla base dei risultati dei test eseguiti per la topologia senza prelazione (Fig. 5.1b), riportati in Fig. 6.3a, si osserva un aumento lineare ma piuttosto repentino del PDR nei primi valori di scanInterval (10 ÷ 40 ms). Successivamente, si verifica una stabilizzazione delle prestazioni a valori di scanInterval > 40 ms. I PDR si attestano tra il 99-100% e rimangono costanti per il resto dei valori di scanInterval. Come conseguenza di quanto analizzato nel capitolo 5, la latenza presenta una crescita lineare all'aumentare del valore di scanInterval, passando da valori di 300 ms a 1.35s, rispettivamente per valori di scanInterval di 10 ms e 200 ms.

Analizzando il diagramma di Fig. 6.3b, riferito alla gestione con prelazione, possiamo notare che i valori di PDR non superano mai l'88% e mostrano una certa variabilità per tutti i valori dell'ascissa. Inoltre, troviamo un'analogia nell'andamento del PDR nei primi valori dell'intervallo di scansione dei grafici considerati. Anche in questo caso il Packet Delivery Ratio mostra un aumento lineare e piuttosto repentino. Tuttavia quello che risulta interessante è il valore della latenza. A meno di leggere fluttuazione, si attesta su valori dell'intorno dei 250 ms per l'intero asse delle ascisse.

I risultati dei test della topologia con 4 coppie SRC-DST presentano un andamento simile a quello già descritto. In particolare, il grafico della Figura 6.4a mostra un aumento graduale del PDR che si estende fino a valori di scanInterval di circa 90 ms per tutte le coppie SRC-DST considerate. Tuttavia, si osservano lievi differenze nelle prestazioni delle coppie 1-5 e 2-6 rispetto a quelle delle coppie 3-7 e 4-8, soprattutto per i primi valori di scanInterval. Tale differenza può essere quantificata mediamente



(a)



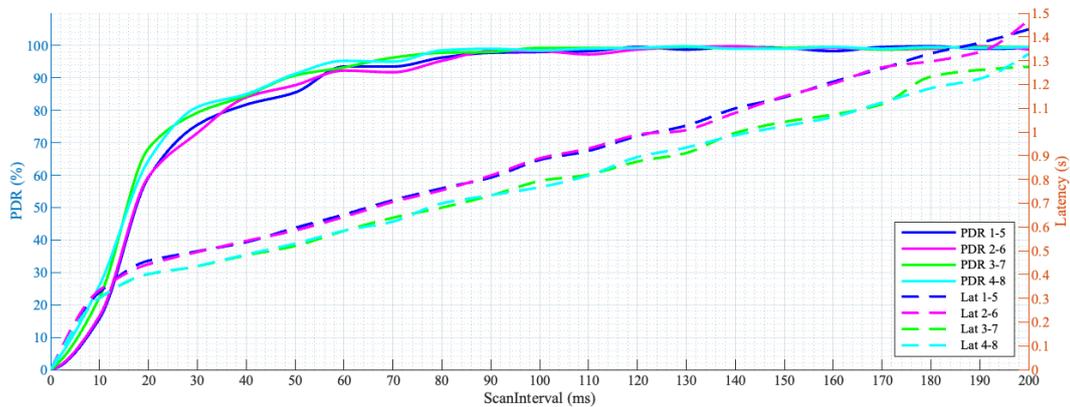
(b)

Figura 6.3: Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

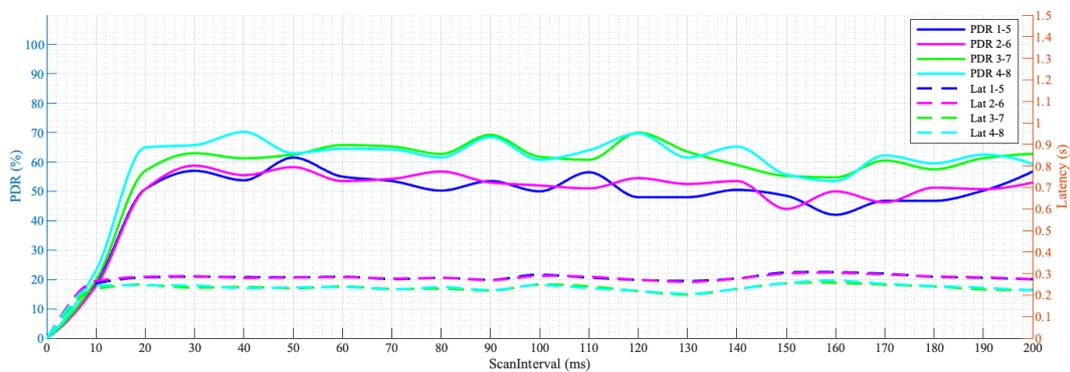
intorno agli 8 punti percentuali e si attenua man mano che ci si avvicina alla fase di stabilizzazione. Nei rimanenti valori di scanInterval ($> 90\text{ms}$), le prestazioni raggiungono circa il 100%. Per quanto riguarda le latenze, esse mostrano lo stesso andamento con una crescita lineare e valori compresi tra i 340 ms e 1.45 s. Come avviene per il PDR, si osservano delle differenze tra le latenze delle coppie 1-5 e 2-6 rispetto a quelle delle coppie 3-7 e 4-8. In particolare, tra di loro emerge un divario nelle latenze già dai primi valori di scanInterval e tale divario aumenta all'aumentare del parametro di scansione.

Osservando il grafico relativo ai test con l'implementazione della prelazione, si nota una netta differenza di prestazioni tra le coppie 1-5 e 2-6 rispetto a 3-7 e 4-8. Escludendo i dati ottenuti nel range 0-20ms, le curve blu e verde in Fig. 6.4b mostrano valori di PDR compresi tra 54% e 70%, mentre le curve delle coppie SRC-DST 1-5 e 2-6 mostrano un range di variabilità compreso tra 42% e 62%. Invece, le latenze, ad eccezione di un lieve scostamento tra le stesse coppie considerate per il PDR, mostrano andamenti lineari per tutti i valori di scanInterval che si attestano sui 230 ms e 270 ms.

Gli ultimi risultati sul PDR presentati in questa sottosezione riguardano i dati ottenuti



(a)



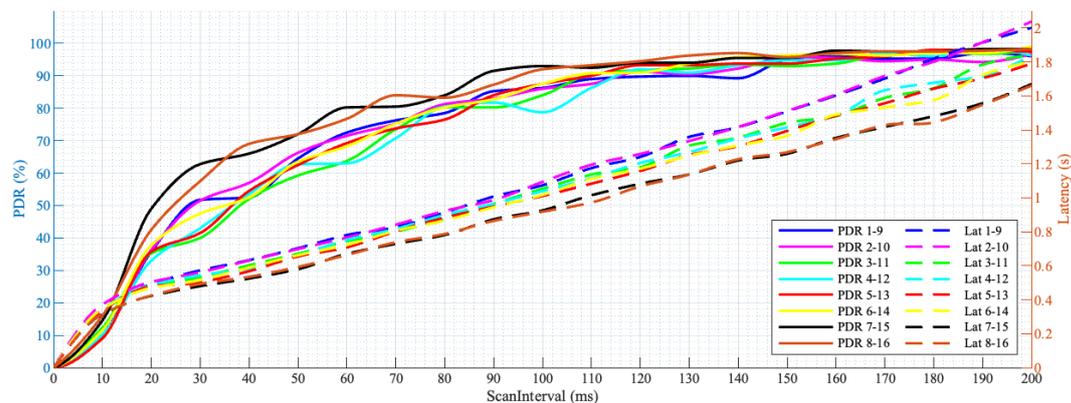
(b)

Figura 6.4: Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

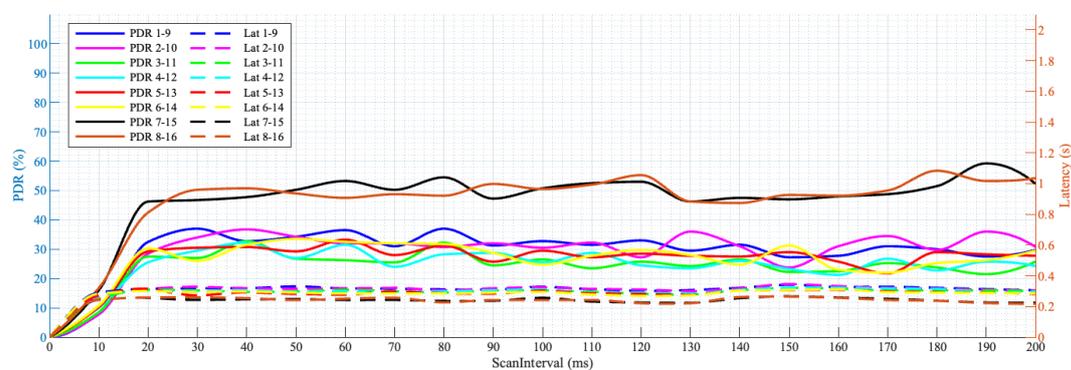
dalla topologia con 8 SRC-DST. Le curve della Fig. [6.5a](#) mostrano una crescita molto graduale del PDR all'aumentare dello scanInterval. Questo vale per tutte le coppie considerate; solo per valori di scanInterval maggiori di 160 ms l'affidabilità del recapito del messaggio aumenta a valori compresi tra 95% e 98%. Tuttavia, PDR del 100% non vengono registrati per nessuna sorgente nell'intervallo scanInterval considerato. Per quanto riguarda le latenze, in questo caso la crescita lineare in base allo scanInterval presenta una pendenza maggiore (anche se non visibile dalla figura). Infatti, i valori passano dai 360 ms per uno scanInterval pari a 10 ms fino ad arrivare a oltre 2 s nel caso di un intervallo di scansione di 200 ms. Inoltre, è possibile notare come le diverse coppie presentino latenze differenti a parità di valori di scanInterval.

I dati delle simulazioni per l'implementazione con prelazione e 8 sorgenti (Fig. [6.5b](#)) mostrano un andamento molto diverso da quello appena analizzato. Le percentuali massime riscontrate sono molto inferiori, con valori non superiori al 55 ÷ 60% per le coppie 7-15 e 8-16 e al 26 ÷ 36% per le restanti coppie. Al contrario, le latenze, sebbene siano leggermente più alte rispetto ai casi con 2 e 4 coppie sorgenti per la stessa

Capitolo 6 Nuova implementazione dello standard



(a)



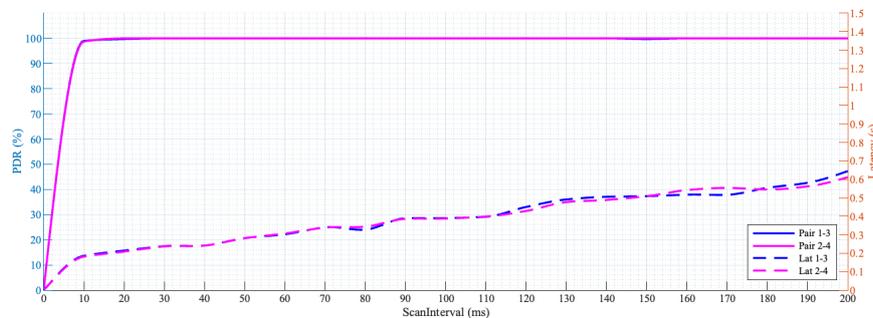
(b)

Figura 6.5: Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

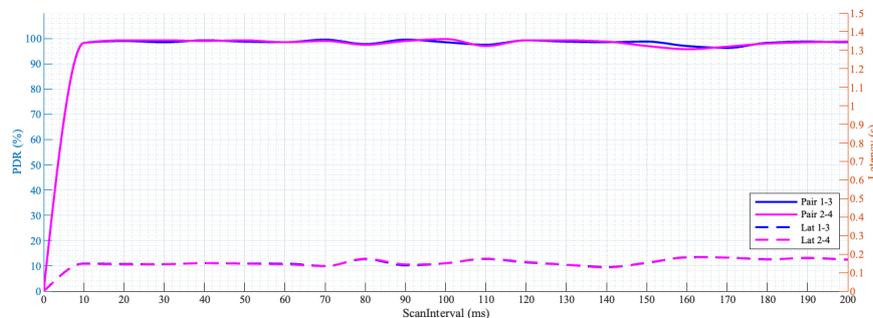
implementazione, presentano valori molto inferiori e costanti rispetto all'implementazione senza prelazione. Dal grafico i valori variano dai 240 e 310 ms a seconda della coppia considerata.

6.3.2 Test con $R = 16m$

Partendo dalla topologia con 2 SRC-DST, i dati raccolti e riportati in Fig. 6.6a mostrano un andamento vicino al caso "ideale". Per ogni valore di scanInterval considerato, il PDR dei pacchetti è del 100%, anche per quei valori di scanInterval di Fig. 6.3a dove si sono verificate percentuali significativamente inferiori. Anche le latenze hanno subito un miglioramento rispetto al caso con $R=9m$, ottenendo valori compresi tra i 180 ms e i 640 ms. L'effetto positivo derivante dall'aumento del raggio di copertura è più evidente in Fig. 6.6b, dove i dati PDR mostrano valori compresi tra 96% e 100% per ogni valore dell'asse delle ascisse. In merito alla latenza, il valore medio si attesta intorno ai 170 ms, sempre per l'intero asse dei valori di scanInterval.



(a)



(b)

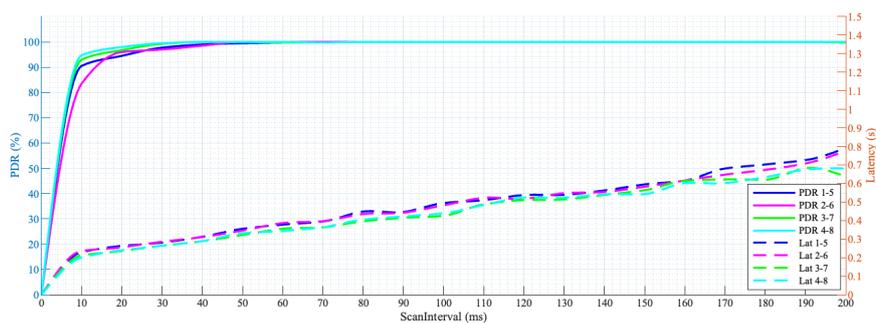
Figura 6.6: Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

Troviamo situazioni simili nei diagrammi di Fig. 6.7a e Fig. 6.7b. Nel primo caso il PDR diminuisce leggermente nei primi valori di scanInterval, ma oltre i 40 ms i valori sono pressoché costanti e pari al 100%. Nel secondo diagramma, invece, il PDR è mediamente più alto che nel caso con 4 SRC-DST, prelazione e raggio di 9 m, ma non raggiunge valori superiori al 94% in nessuna simulazione. Nel caso specifico le coppie 1-5 e 2-6 hanno un PDR che assume valori compresi tra 72% e 87%, mentre le altre variano tra 78% e 94%.

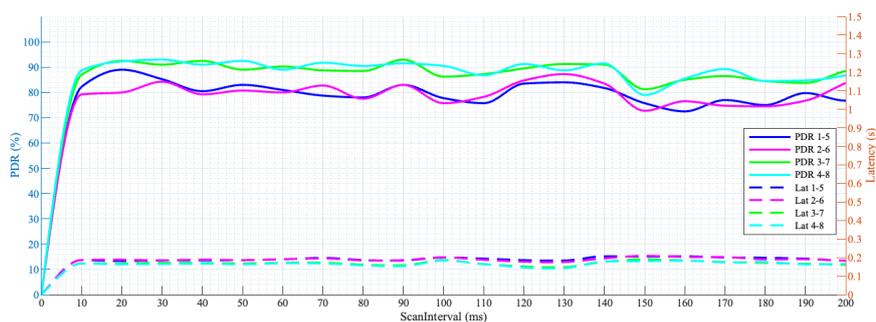
Capitolo 6 Nuova implementazione dello standard

A differenza del caso in cui $R=9$ m, le latenze mostrate nella Fig. 6.7a sono uniformi per tutte le coppie SRC-DST, ma mantengono comunque un andamento di crescita lineare. Il maggiore raggio di copertura garantisce una diminuzione delle latenze, che ora oscillano tra 240 ms e 800 ms.

Nella Fig. 6.7b, si può notare che i valori delle latenze si sovrappongono, ma con un andamento costante che si aggira intorno ai 200ms.



(a)

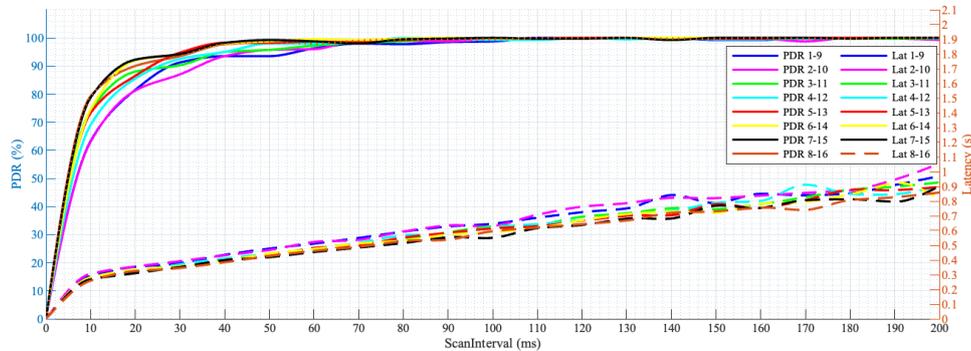


(b)

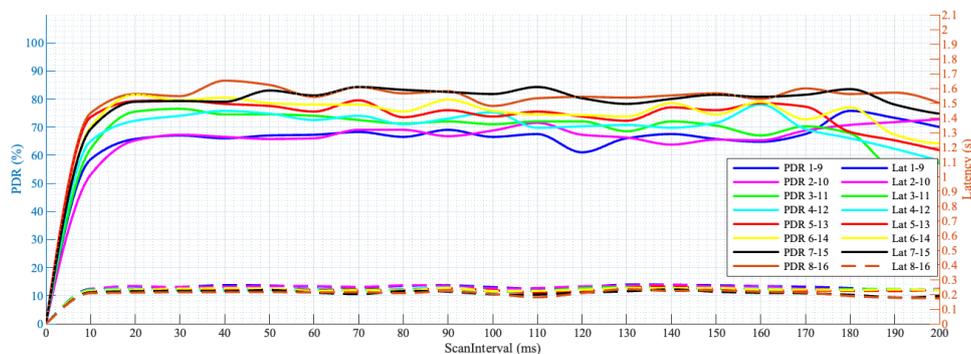
Figura 6.7: Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

Gli ultimi dati da presentare sono quelli della Fig. 6.8 e riguardano le simulazioni della topologia con 8 coppie di nodi sorgente-destinazione. In Fig. 6.8a, si verifica un comportamento simile a quello dei risultati presentati in precedenza. Il PDR ha una crescita progressiva nei primi valori di ScanInterval e una fase di stabilizzazione in cui tutte le coppie raggiungono valori intorno al 98-100%. Il passaggio tra le due fasi avviene a uno scanInterval di 70 ms. Le latenze, invece, sono comprese tra 300 ms e 1 s, rappresentando una diminuzione del 50% rispetto al caso di Fig. 6.5a. Anche in questo caso, la variabilità tra le diverse coppie si è ridotta, come accaduto nel caso di 4 coppie. Per quanto riguarda il grafico di Fig. 6.8a, i valori di PDR variano tra 58% e 86%, corrispondenti ad un incremento tra 20 e 30 punti percentuali rispetto al caso di

Fig. 6.5a, mentre le latenze presentano valori molto positivi, compresi tra i 180 ms e i 240 ms.



(a)



(b)

Figura 6.8: Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione senza prelazione; (b) implementazione con prelazione.

6.4 Discussione

I risultati ottenuti e presentati nella sezione precedente mostrano differenze non solo tra le due implementazioni, ma anche all'interno delle stesse a causa di variazioni nel numero di coppie sorgente-destinatario, del raggio di copertura radio e dell'intervallo di scansione. Dai grafici "a" e "b" di ogni figura emerge che la gestione senza prelazione dal punto di vista del PDR risulta migliore rispetto a quella con prelazione, sia in termini relativi che assoluti. In quasi tutti i casi, la percentuale di pacchetti consegnati raggiunge il 100%, tranne in un caso dove raggiunge il 98% (Fig. 6.5a). Al contrario, le prestazioni nella gestione con prelazione sono sempre inferiori. L'unica eccezione si verifica nei test con 2 coppie SRC-DST e un intervallo di copertura di 16 m (Fig. 6.6b), dove i PDR sono quasi identici al caso senza prelazione. Le prestazioni inferiori nella soluzione con prelazione sono dovute alla diversa implementazione dello standard.

L'avvio di un Advertising Event subito dopo la ricezione di un pacchetto da parte di un nodo relay accelera la ritrasmissione, ma preclude la possibilità che il nodo riceva pacchetti aggiuntivi durante la fase di ritrasmissione. Inoltre, la trasmissione immediata di più nodi relay può aumentare la possibilità di interferenza in ricezione, poiché possono sovrapporsi più segnali contemporaneamente.

La latenza presenta una situazione completamente diversa. L'uso della gestione con prelazione offre prestazioni migliori rispetto a quella senza. In tutti i casi esaminati, la latenza presenta un andamento lineare per tutti i valori di scanInterval considerati, stabilizzandosi tra i 150 ms e i 330 ms, a seconda della topologia di rete e del raggio di copertura considerati. Al contrario, nel caso di implementazione senza prelazione, la latenza aumenta in modo lineare in relazione allo scanInterval, a causa dell'implementazione realizzata dalla Mathworks, come analizzato nel capitolo precedente. In particolare, la latenza può raggiungere valori anche dell'ordine dei 2 secondi.

I risultati delle simulazioni eseguite su reti con 4 e 8 coppie sorgente-destinazione (riportati nelle Fig. 6.4, Fig. 6.5, Fig. 6.7 e Fig. 6.8) suggeriscono una certa dipendenza del PDR e della latenza dalla posizione dei nodi nella rete, indipendentemente dal raggio e dall'implementazione. In particolare, i grafici della Fig. 6.4 e della Fig. 6.7 indicano che le coppie di nodi con le migliori prestazioni si trovano più vicine all'asse di simmetria trasversale della topologia, ovvero le coppie 3-7 e 4-8. La Fig. 6.8 conferma questa tendenza e la estende a tutte le 8 coppie prese in considerazione. Si nota che, allontanandosi dall'asse di simmetria trasversale, i PDR delle diverse coppie SRC-DST diminuiscono passando da un massimo dell'86% ad un minimo del 58%. Sebbene nella Fig. 6.8b sia poco visibile, questo divario è presente anche per le latenze di entrambe le implementazioni.

In generale, è più evidente nei risultati dei test per la gestione con prelazione rispetto a quelli senza prelazione. È importante notare che le coppie che si trovano in posizioni opposte rispetto all'asse trasversale hanno PDR e latenze molto simili. Inoltre, per quanto riguarda la dipendenza del PDR dalla posizione dei nodi, la Fig. 6.5b mostra un comportamento insolito rispetto agli altri grafici trattati. Le due coppie più centrali continuano ad avere le migliori prestazioni, mentre per le coppie rimanenti, la differenziazione dei PDR e latenze in base alla loro posizione non è più evidente. D'altra parte, se osserviamo le curve delle coppie di sorgenti da 1 a 6, notiamo un'organizzazione casuale e un raggruppamento in un intervallo di valori compreso tra il 20% e il 36% per il PDR e tra il 300 ms e 340 ms per le latenze.

Confrontando i grafici della stessa implementazione, si possono ottenere ulteriori informazioni. In particolare, dalle simulazioni relative all'implementazione senza prelazione, emerge che all'aumentare del numero di sorgenti nella rete, i dispositivi richiedono un intervallo di scansione maggiore per garantire un PDR tra il 97% e il 100%. Questo fenomeno è più evidente nelle simulazioni con un raggio di 9 m rispetto a quel-

le con una maggiore copertura radio. Dall'analisi dei risultati risulta che il raddoppio del numero di coppie corrisponde approssimativamente al raddoppio dell'intervallo di scansione. Ad esempio, nel passaggio da 2 coppie SRC-DST a 4 SRC-DST, l'intervallo di scansione necessario per garantire un PDR superiore al 98% è di 40 ms e 90 ms, rispettivamente. Al contrario, se si passa dalla topologia con 4 coppie SRC-DST a quella con 8, l'intervallo di scansione richiesto sembra essere maggiore di 180-190 ms. Tuttavia, il costo in termini di latenza per questa situazione è molto elevato. A causa degli intervalli di scansione necessari per garantire un PDR superiore al 98%, le trasmissioni saranno affette da ritardi elevati, pari a 480 ms per 2 SRC-DST, tra 750 ms e 820 ms per la topologia con 4 SRC-DST, e infine tra 1,35 s e 1,8 s per 8 SRC-DST.

La situazione è completamente diversa quando si confrontano tutti i grafici di implementazione con prelazione. Oltre un certo valore di scanInterval, ovvero 20 ms per i test con $R1 = 9m$ e 10 ms per i test con $R2 = 16m$, il PDR delle diverse coppie non sembra essere influenzato dal suo aumento. Anche per quanto riguarda le latenze, si può fare la stessa osservazione. Dalle simulazioni emerge che le latenze non dipendono dall'intervallo di scansione specifico. Se i pacchetti ricevuti vengono trasmessi immediatamente, i ritardi causati dal tempo trascorso tra la ricezione del pacchetto e la fine della finestra di scansione dell'implementazione Mathworks sono azzerati. La latenza dipende invece dal percorso del pacchetto e quindi dal numero di salti, tempi di elaborazione e tempi di advertising.

6.5 Conclusioni

In questo capitolo è stata presentata un'analisi di come diverse interpretazioni dello standard Bluetooth mesh e le relative implementazioni possono influire sul comportamento della rete in termini di affidabilità e latenza end-to-end. In particolare, viene presentato un confronto di due diverse implementazioni della gestione dei messaggi nelle fasi di scanning e advertising nei nodi relay, differenziate dall'introduzione di un principio di prelazione di gestione. Dai risultati delle diverse simulazioni effettuate, è emerso che, al momento, le prestazioni migliori in termini di Packet Delivery Ratio sono ottenute mediante una gestione senza prelazione dei messaggi. In tutte le prove svolte, l'implementazione senza prelazione ha costantemente raggiunto percentuali tra il 98% e il 100%, mentre quella con prelazione ha ottenuto percentuali inferiori. D'altro canto, se consideriamo i risultati dal punto di vista della latenza, le migliori prestazioni si ottengono con l'implementazione con prelazione in ogni scenario simulato, con valori di latenza compresi tra i 150 ms e i 350 ms. Questi dati confermano il raggiungimento dell'obiettivo prefissato, cioè di ottenere una rete più reattiva con bassi valori di latenza. Tuttavia, il passo successivo sarà quello di migliorare le prestazioni anche dal punto di vista del Packet Delivery Ratio, attraverso l'introduzione di migliorie che

Capitolo 6 Nuova implementazione dello standard

ottimizzino l'implementazione con prelazione della gestione dei messaggi nelle fasi di scanning e advertising nei nodi relay. L'obiettivo finale sarà quello di garantire un'implementazione performante sia dal punto di vista dell'affidabilità in termini di Packet Delivery Ratio che della latenza end-to-end.

Capitolo 7

Miglioramento dell'implementazione della gestione dei messaggi con prelazione

Dopo aver valutato i risultati dei test sulla riduzione della latenza ottenuti dalla nuova implementazione presentata nel capitolo precedente e aver consultato sia gli sviluppatori di Mathwork che quelli della NordicSemiconductor, si è deciso di concentrare l'attività di ricerca sul miglioramento della gestione dei messaggi nella fase di advertising e scanning dei nodi relay presentata nel capitolo precedente, al fine di aumentare le prestazioni, in particolar modo per quanto riguarda il Packet Delivery Rate. In linea con le specifiche del Bluetooth mesh e apportando modifiche compliant, sono state introdotte alcune variazioni nella gestione della fase di scansione, mantenendo comunque la prelazione dell'advertising sulla scansione.

7.1 Schema ottimizzazione proposto

La prima versione dell'implementazione della gestione dei messaggi nella fase di advertising e scanning dei nodi relay con prelazione prevedeva tre fasi:

1. Interrompere la fase di scansione al momento della ricezione di un messaggio adatto per l'inoltro.
2. Effettuare l'operazione di advertising del messaggio ricevuto.
3. Riprendere la fase di scansione a partire dal canale 37 e reimpostare il timer dello scanInterval al valore iniziale.

Tuttavia, come mostrato nel capitolo precedente, queste caratteristiche hanno portato a ottimi risultati in termini di riduzione della latenza ma hanno generato dei PDR troppo bassi rispetto al caso senza prelazione.

Di fronte a queste lacune, è stata sviluppata una seconda versione dell'implementazione con priorità al fine di migliorare le performance, inclusa l'affidabilità del Packet Delivery Ratio (PDR). In linea di principio, la modalità di prelazione costituita dai punti 1 e 2 menzionati in precedenza, è rimasta invariata poiché è essa stessa a permettere di ottenere bassi valori di latenza inoltrando i pacchetti nel più breve tempo possibile. Tuttavia, si è deciso di intervenire sulla fase di ripresa della scansione, modificando la sua logica al fine di migliorare il PDR.

Due sono le regole su cui si basa la nuova fase di scansione:

- Dopo ogni Advertising Event, la fase di scansione riprende dallo stesso canale in cui era stata interrotta.
- Il nodo relay valuta il tempo residuo per completare l'intervallo di scansione e lo assegna all'evento successivo come nuovo intervallo di scansione.

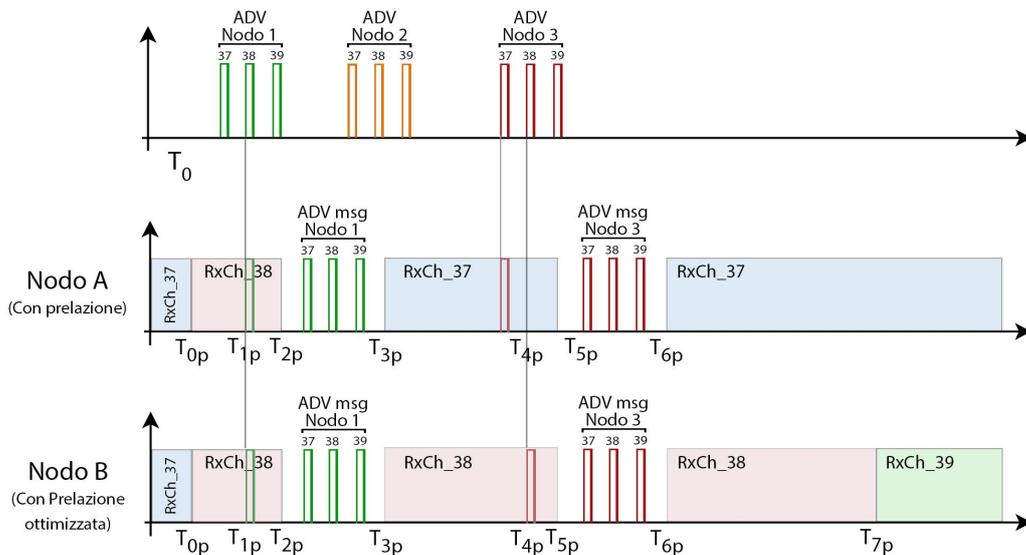


Figura 7.1: Confronto tra la fase di scansione e advertising dei nodi relay dell'implementazione con prelazione e dell'implementazione ottimizzata della prelazione.

I nodi relay effettuano una scansione continua dei canali trasmissivi, commutando da uno all'altro in base allo scanInterval. Tuttavia, quando ricevono un messaggio conforme per l'inoltro, memorizzano il canale e il tempo trascorso in scansione, quindi passano immediatamente alla modalità advertising per inviare il messaggio ai nodi entro il loro raggio di copertura radio. Dopo l'invio dei dati, i nodi calcolano il tempo

rimanente della finestra di scansione e lo impostano come nuovo intervallo di scanning. Successivamente, i nodi impostano lo stesso canale in cui sono stati interrotti e riprendono la scansione del canale fino alla fine del nuovo intervallo di scansione o fino alla ricezione di un nuovo messaggio.

La Fig. 7.1 è un esempio che aiuta a comprendere più facilmente la logica della nuova implementazione. Tralasciando il comportamento del nodo A descritto nel capitolo precedente, concentriamoci sul nodo B che utilizza la nuova versione della prelazione. Il nodo B inizia l'ascolto sul canale 38 all'istante $T0p$. Al tempo $T1p$, riceve un messaggio dal Nodo 1 adatto ad essere inoltrato e al tempo $T2p$ passa immediatamente dalla modalità di scansione a quella di advertising. Successivamente, all'istante $T3p$, il nodo riprende la scansione sullo stesso canale della fase interrotta, ovvero il canale 38. Il tempo per la nuova fase di ascolto viene calcolato e settato dal nodo sottraendo al valore iniziale di `scanInterval` il tempo impiegato nella prima scansione parziale ($T2p - T0p$). Al tempo $T4p$, il nodo B riceve un nuovo messaggio dal nodo 3, interrompe la scansione e al tempo $T5p$ si mette in modalità advertising. Dopo aver inviato il messaggio, ricalcola la nuova finestra di scansione, che questa volta sarà calcolata sottraendo da `scanInterval` il tempo della prima e della seconda scansione parziali, rispettivamente ($T2p - T0p$) e ($T4p - T3p$). Quindi, al tempo $T6p$, il nodo riprende la scansione per un tempo pari al nuovo intervallo, sempre sullo stesso canale in cui è stato interrotto. Quando il tempo di scansione termina, ovvero all'istante $T7p$, il nodo passa alla scansione del canale successivo, nel caso in figura il canale 39.

L'idea principale consiste nella suddivisione dell'intervallo di scansione in sottointervalli di durata variabile, in modo che la loro somma equivalga al tempo specificato dal parametro `scanInterval`. Ad esempio, nel caso illustrato in Figura 7.1, al tempo $T7p$, la somma delle fasi di scansione deve corrispondere al valore di `scanInterval`.

$$(T2p - T0p) + (T4p - T3p) + (T7p - T6p) = scanInterval$$

In aggiunta all'ottimizzazione della prelazione, è stata decisa l'introduzione di un accorgimento raccomandato dalle specifiche Bluetooth mesh per ridurre le collisioni tra le trasmissioni di più nodi relay non presenti nelle librerie del simulatore della Mathworks.

Nello specifico, nella sezione 3.4.6.3 del Bluetooth mesh v.1.0.1 viene consigliato di introdurre un breve ritardo casuale tra la ricezione e l'inoltro di un Network PDU da parte di un nodo relay in modo da evitare collisioni tra i relay che ricevono il PDU contemporaneamente. Per questo motivo, è stato deciso di applicare un tempo di attesa, definito "tempo di backoff", prima dell'inizio di ogni Advertising Event di ogni nodo relay.

Per generare i tempi di backoff, è stato utilizzata una funzione pseudo-random con

distribuzione uniforme. Tuttavia, poiché il tempo di backoff è un parametro temporale applicato alle trasmissioni, avrà un impatto negativo sui tempi di latenza. Per minimizzare questo impatto si è scelto di selezionare un intervallo di valori non troppo ampio, ma allo stesso tempo che consenta una separazione temporale adeguata tra le trasmissioni dei diversi nodi relay. Si è stabilito che un intervallo compreso tra 0 e 20 ms fosse la scelta più appropriata.

7.2 Risultati

I risultati riportati di seguito sono stati ottenuti utilizzando le stesse topologie e parametri descritti nei capitoli precedenti. Per facilitare l'analisi, i dati sono stati suddivisi in due gruppi in base al raggio di copertura utilizzato durante i test. Inoltre, sono state considerate diverse coppie SRC-DST e sono state prodotte rappresentazioni grafiche per la versione di prelazione iniziale (prel.v1), quella ottimizzata (prel.v2) e per l'implementazione della prelazione ottimizzata con l'aggiunta del tempo di backoff (prel.v2_B).

Come già spiegato nel capitolo precedente, valori di scanInterval inferiori a 20 ms per il caso con $R = 9$ m e 10 ms per quello con $R = 16$ m, non garantiscono un PDR sufficiente per la rete. Di conseguenza, quando si farà riferimento all'intero scanInterval tali valori non verranno considerati.

7.2.1 Simulazioni con $R = 9$ m

Dopo una prima analisi visiva e generale dei grafici delle Fig. 7.2, 7.3 e 7.4, si può notare che l'introduzione di miglorie nella fase di scansione, combinata con l'utilizzo del backoff, hanno portato ad un notevole aumento delle prestazioni in termini di Packet Delivery Ratio rispetto alla prima versione di prelazione, con incrementi anche di 26 punti percentuali. Tuttavia, è importante sottolineare che l'utilizzo del backoff comporta un costo in termini di latenza. Infatti, confrontando i dati relativi alla prel.v1 con quelli della prel.v2_B si osserva un aumento tra il 40% e il 70% a seconda dello scenario simulato.

Analizziamo in dettaglio alcuni grafici. Nella Fig. 7.2 è possibile confrontare i risultati delle simulazione della topologia con 2 coppie SRC-DST per le tre implementazioni. L'uso della prel.v2 consente di ottenere un PDR di oltre il 92% su tutto l'intervallo di scanInterval, a differenza della prel.v1 che oscilla tra il 58% e l'88%. Per quanto riguarda la latenza, non vi sono differenze significative nei valori tra le due implementazioni, attestandosi intorno ai 250 ms.

Con la prel.v2_B, invece, si ottiene un ulteriore incremento medio di circa 4 punti percentuali nel PDR con picchi del 98% e 99%. Tuttavia, questo miglioramento ha un

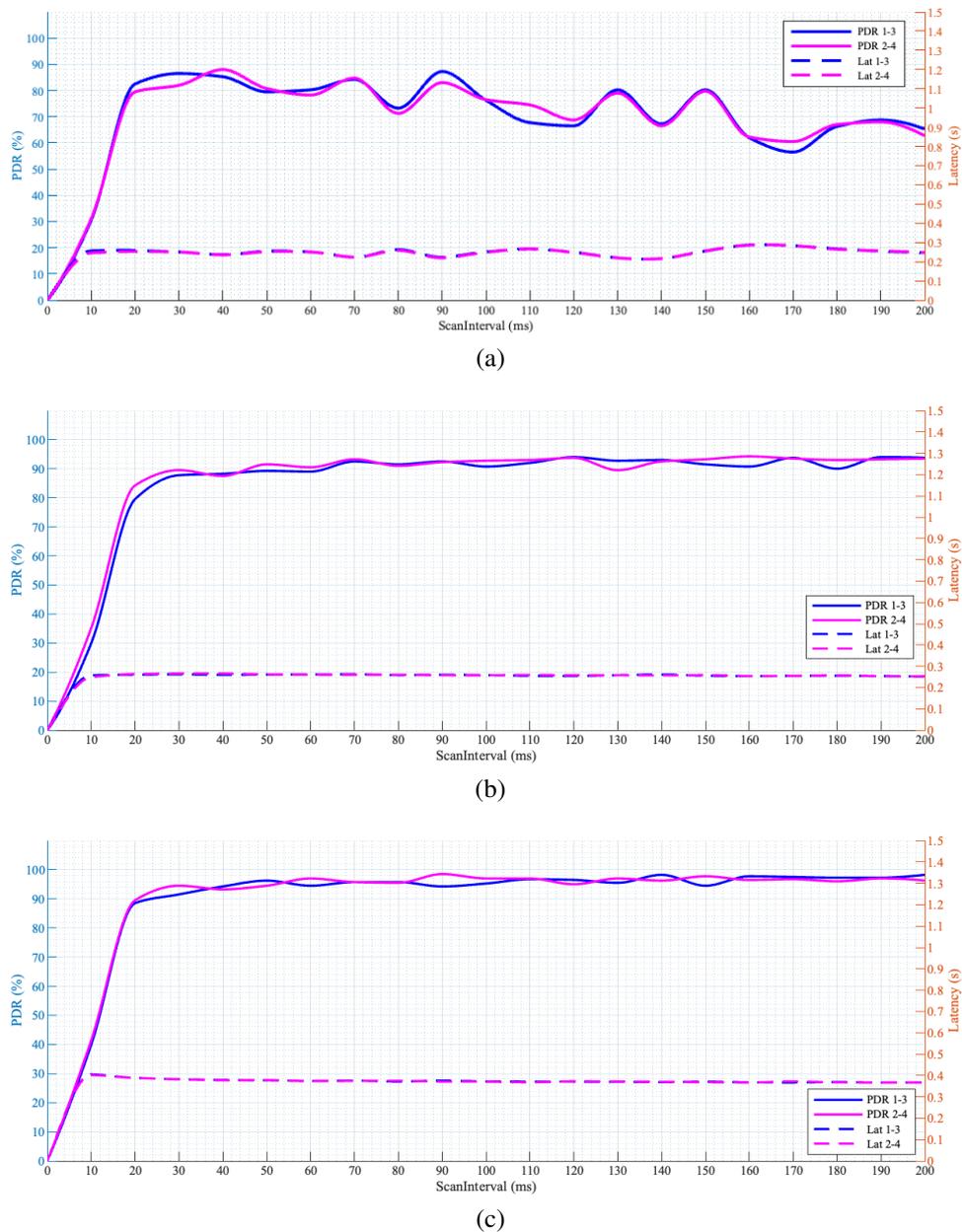


Figura 7.2: Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

costo molto elevato in termini di latenza end-to-end. Contrariamente alla prel.v2, la prel.v2_B genera un aumento del 42% del valore di latenza, passando da 260 ms a 370 ms.

Nel caso di topologie con 4 coppie SRC-DST, i risultati mostrati in Fig. [7.3](#) evidenziano un miglioramento generale delle prestazioni come nel caso precedente, ma non

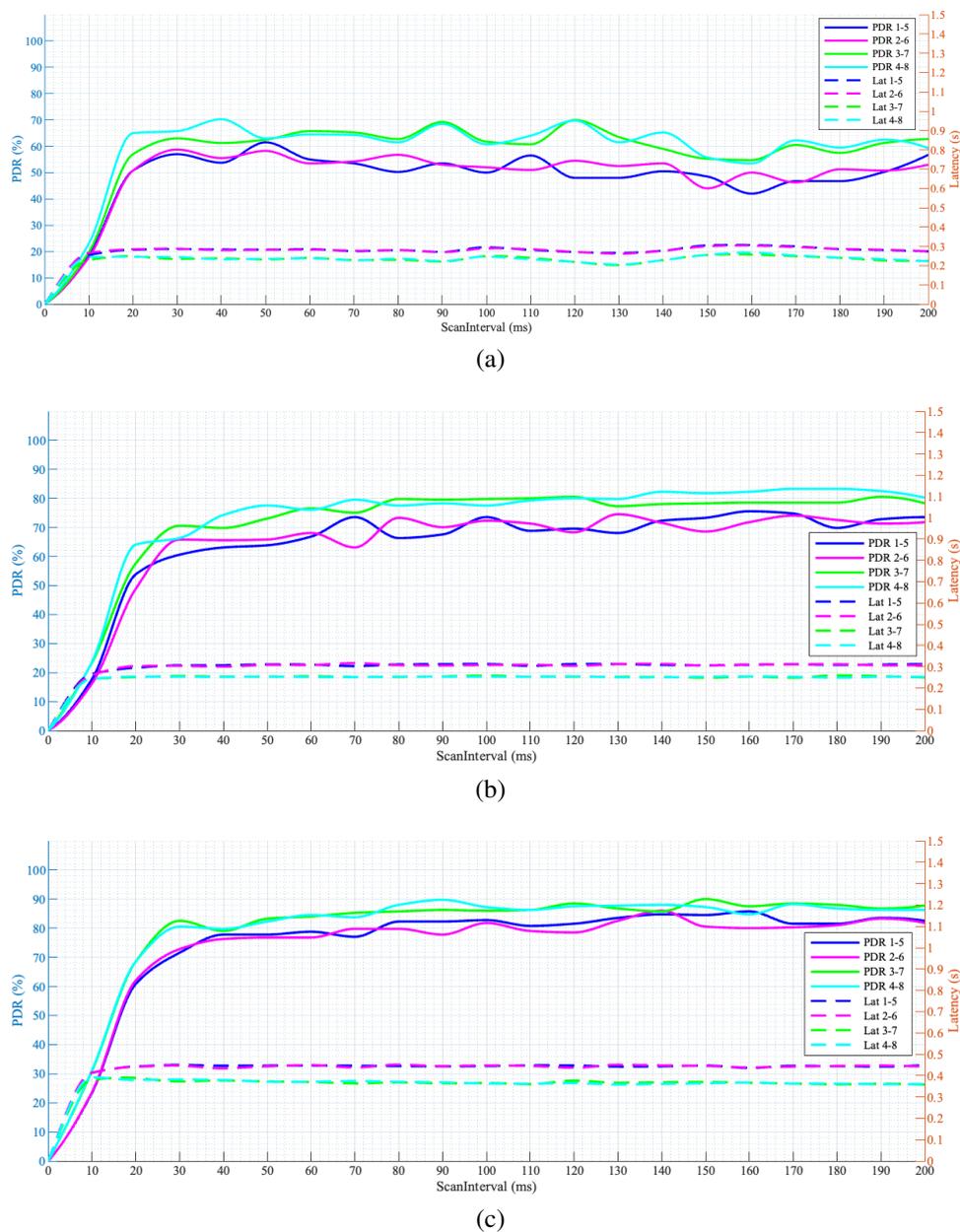


Figura 7.3: Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

con gli stessi valori di PDR prossimi al 100%. Nelle simulazioni dell'implementazione della prima versione della prelazione, i valori di PDR oscillavano tra il 42% e il 64% a seconda della coppia SRC-DST considerata, mentre nel caso della prel.v2 e di quella ottimizzata con il tempo di backoff, i valori sono passati rispettivamente tra il 64% e l'82% e tra il 78% e il 90%.

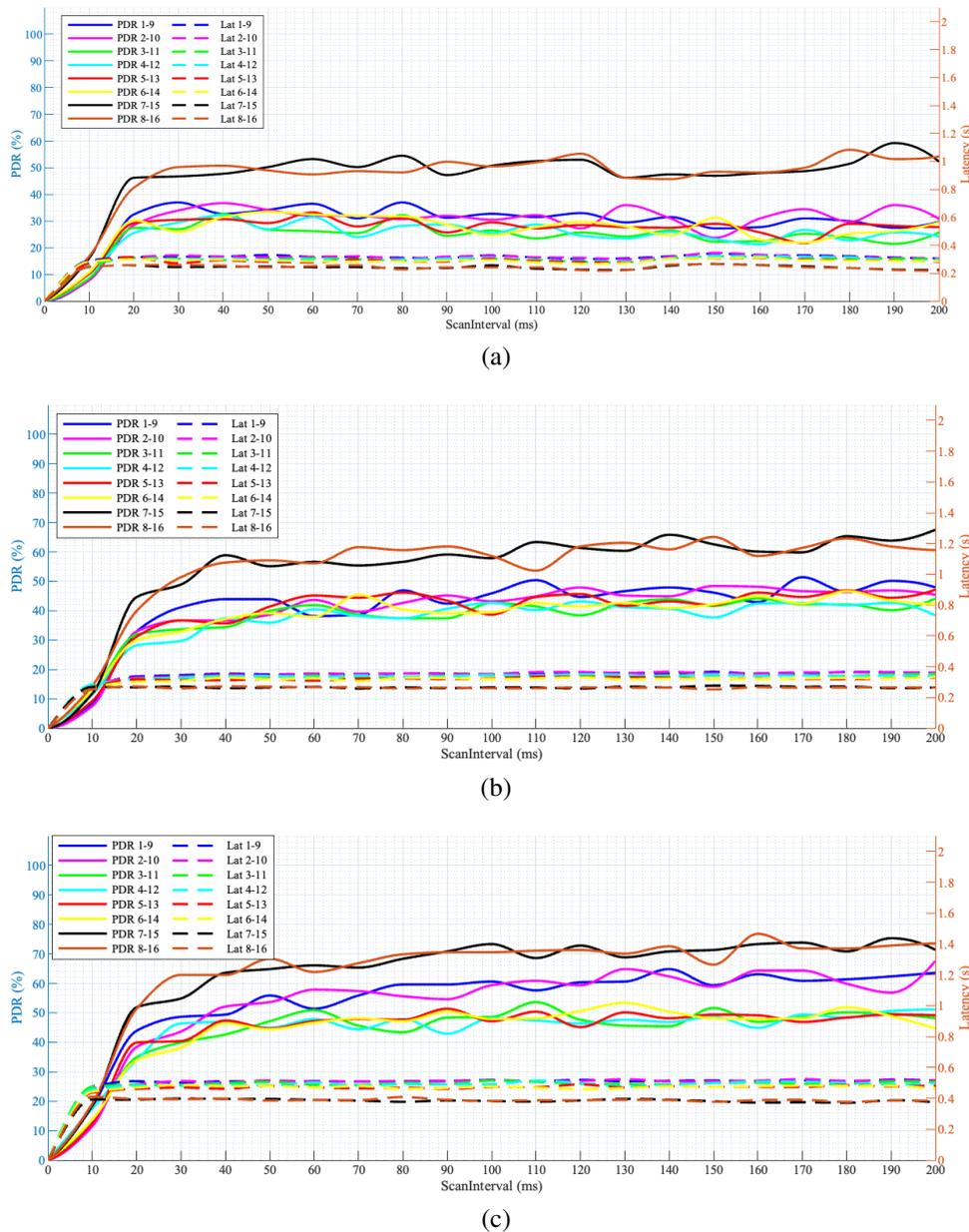


Figura 7.4: Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 9 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

Dagli intervalli numerici e dai grafici di Fig. 7.3c, è possibile osservare come i PDR delle diverse coppie SRC-DST tendano ad allinearsi e a raggiungere un valore di PDR simile. In sostanza, la prel.v2_B consente di ridurre le differenze nel PDR delle diverse coppie SRC-DST che possono essere causate dalla specifica posizione all'interno della topologia considerata. Per quanto riguarda le latenze, i valori riscontrati nella prel.v1 risultavano di 230 ms e 280 ms rispettivamente per le coppie 3-7, 4-8 e quelle 1-5,

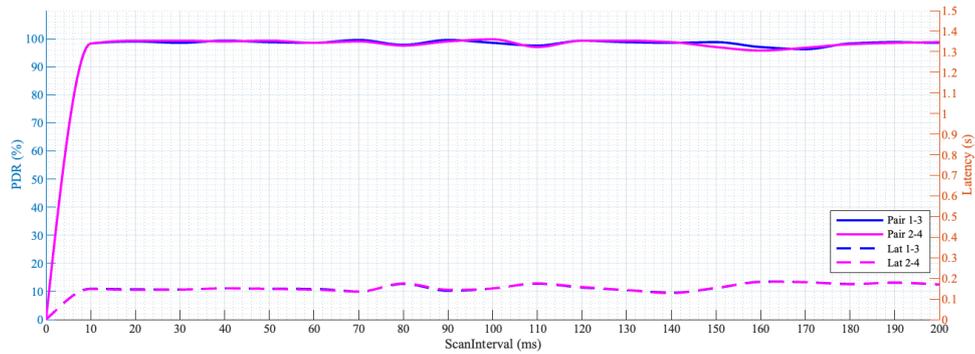
2-6. Nel caso della prel.v2 si attestano a valori di 250ms e 310ms, mentre a causa del backoff, anche in questo caso le ripercussioni sulla latenza nella prel.v2_B sono evidenti e portano ad incrementi di circa il 60% alzando i valori a 365ms e 450ms.

Infine, resta di analizzare il confronto della topologia con 8 coppie di SRC-DST Fig. 7.4. I comportamenti e le considerazioni sono simili a quelli riscontrati in precedenza. Il passaggio dalla prel.v1 alla prel.v2_B ha portato ad un incremento medio di circa 20 punti percentuali per quanto riguarda il PDR e del 70% per quanto riguarda le latenze.

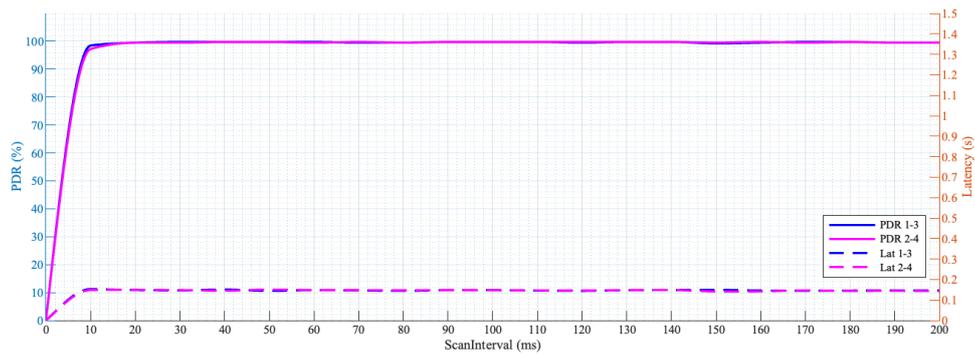
Poiché i valori iniziali del PDR per la prel.v1 (Fig. 7.4a) erano molto bassi, nonostante l'utilizzo della prel.v2_B abbia portato ad un miglioramento, i valori non superano il 71%. Per quanto riguarda le latenze i valori registrati nel grafico di Fig. 7.4c si attestano tra 385 ms e 510 ms.

7.2.2 Simulazioni con $R = 16m$

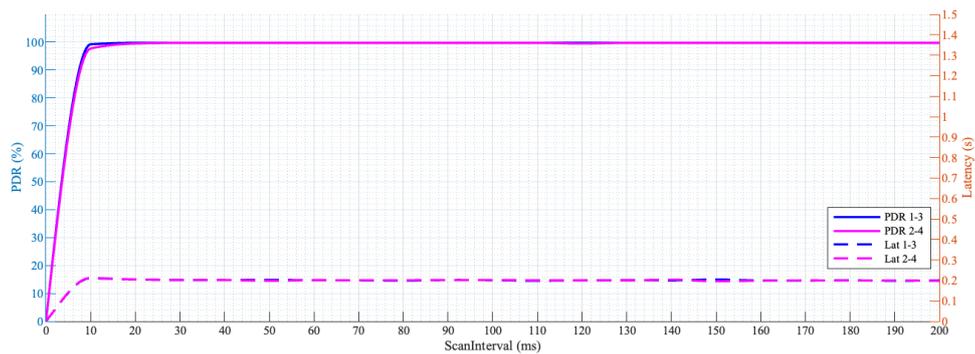
I risultati delle simulazioni con tre diverse topologie, un raggio di copertura di 16 metri e le tre diverse implementazioni con prelazione sono presentati nelle Fig. 7.5, 7.6 e 7.7.



(a)



(b)

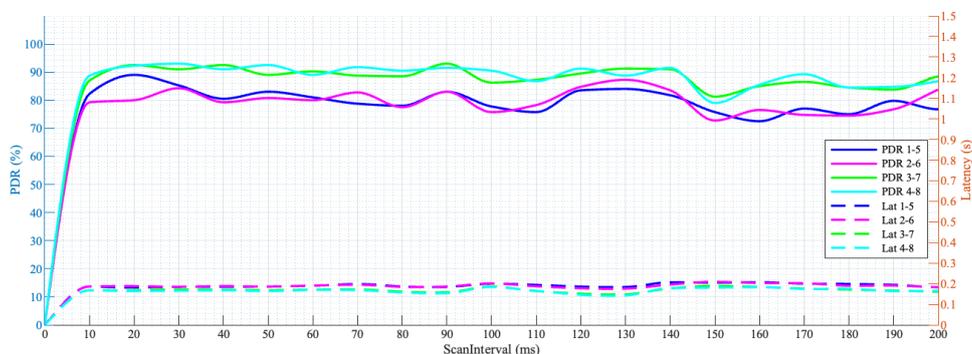


(c)

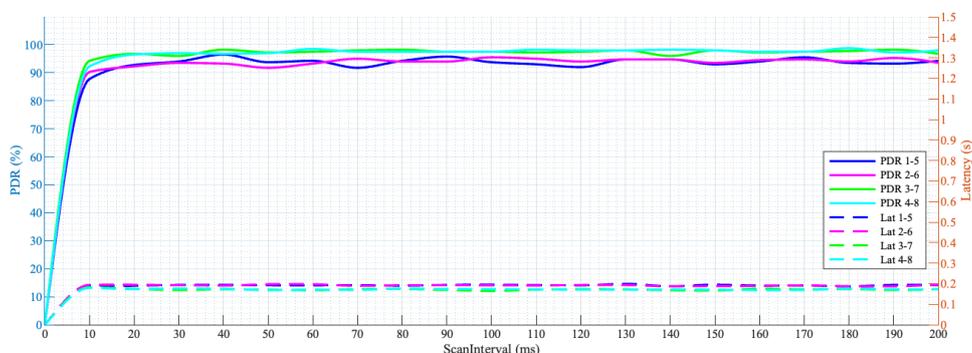
Figura 7.5: Packet Delivery Ratio e Latenza per la topologia con 2 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

I grafici delle Fig. 7.5a, 7.6a e 7.7a mostrano condizioni iniziali molto più favorevoli

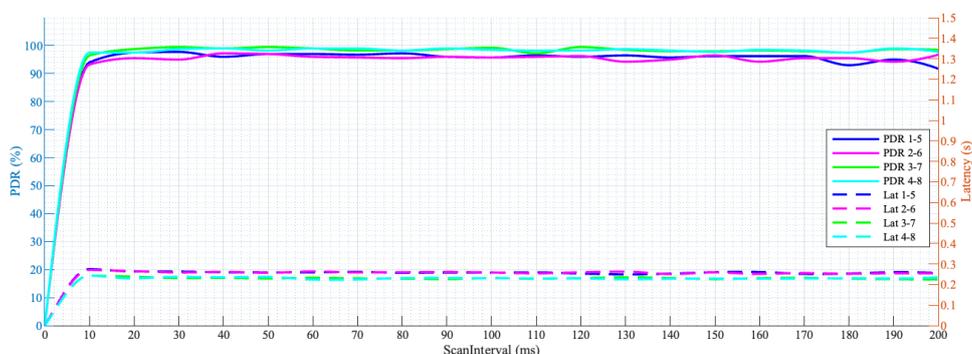
rispetto al caso con un raggio di copertura di 9 metri per ogni topologia considerata.



(a)



(b)



(c)

Figura 7.6: Packet Delivery Ratio e Latenza per la topologia con 4 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

Nel caso dei test per 2 coppie SRC-DST, i risultati delle Fig. 7.5 evidenziano che è possibile ottenere un PDR del 100% per tutti i valori di scanInterval utilizzando solo la versione prel.v2, senza alcun aumento di latenza. In questo caso, l'utilizzo della versione prel.v2_B risulta controproducente perchè pur ottenendo lo stesso risultato in

termini di PDR, porta ad un aumento della latenza del 40%, passando da 145 ms a valori intorno ai 200 ms (Fig. 7.5c).

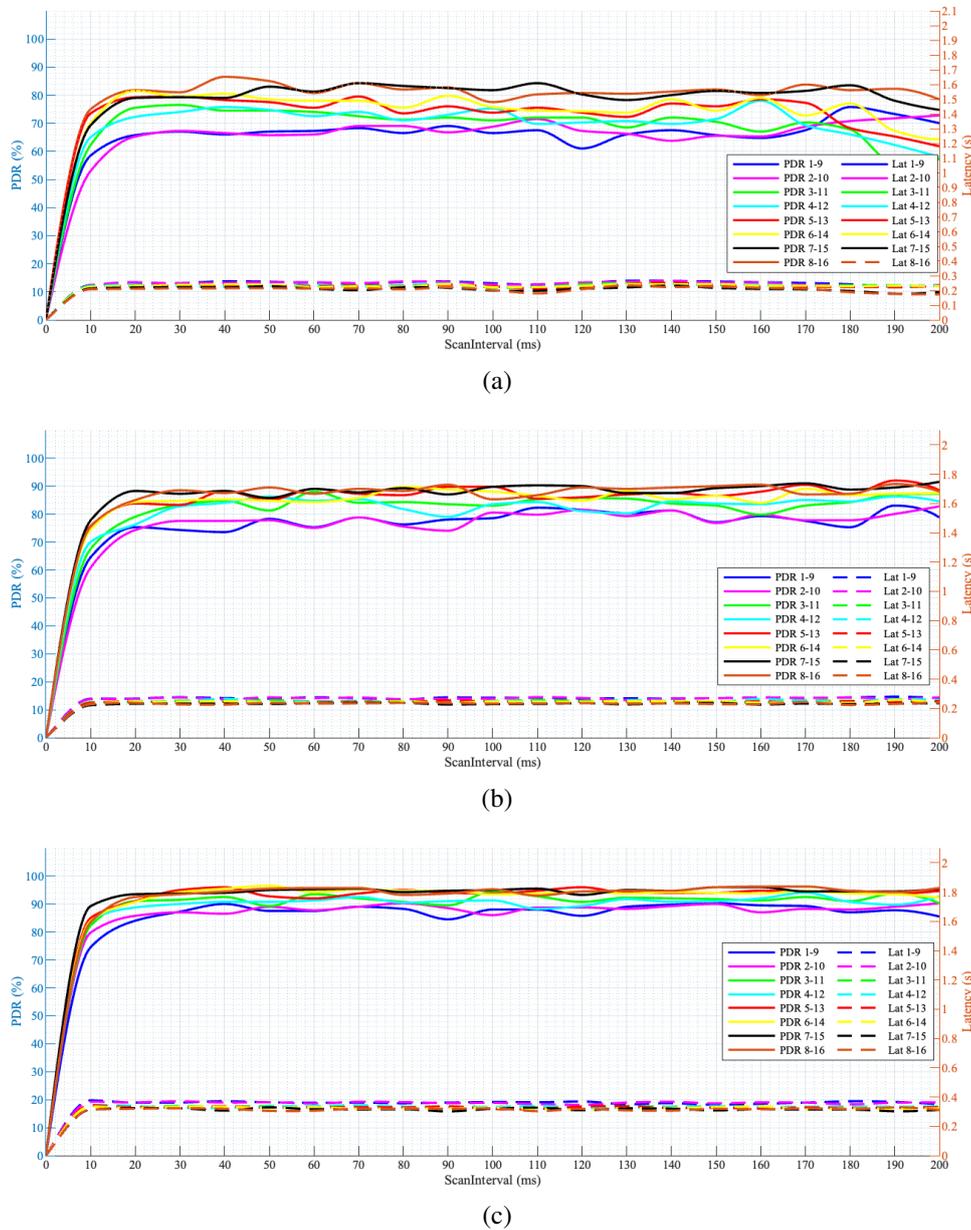


Figura 7.7: Packet Delivery Ratio e Latenza per la topologia con 8 coppie SRC-DST e raggio di copertura di 16 m: (a) implementazione con prelazione; (b) implementazione con prelazione ottimizzata; (c) implementazione con prelazione ottimizzata + backoff 0÷20ms

Per quanto riguarda la topologia con 4 coppie SRC-DST, i valori di PDR passano da valori compresi tra il 72% e il 92% nel caso della prima implementazione (Fig. 7.6a) a valori tra il 97% e il 100% per la prel.v2_B (Fig. 7.6c). La prel.v2 si pone nel mezzo con un intervallo di PDR compresi tra il 93% e il 98% (Fig. 7.6b). Le latenze mostrano

un comportamento simile al resto degli scenari esposti. Anche nel caso della topologia con 4 coppie SRC-DST e $R = 16m$ (Fig. 7.6), l'introduzione delle modifiche nella gestione della prelazione ha portato a una riduzione delle differenze tra i PDR delle singole coppie SRC-DST all'interno della stessa simulazione. Il gap dei PDR è passato da 20 a soli 5 punti percentuali nel caso della prel.v2 e a soli 3 punti percentuali nel caso dell'implementazione prel.v2_B.

Questo comportamento è stato riscontrato anche nei test con 8 coppie di SRC-DST, anche se con riduzioni minori degli intervalli. Con la prel.v1, le 8 coppie presentavano andamenti del PDR compresi tra il 62% e l'86%, traducendosi in un gap di 24 punti percentuali. Con la prel.v2, lo scarto si è ridotto a 16 punti percentuali e i valori dei PDR si sono attestati tra il 74% e il 90%, con valori di latenza compresi nel range di valori 250 ms÷275 ms. Infine, l'implementazione della prel.v2_B ha permesso di aumentare i PDR fino al 96%, a fronte di un aumento medio della latenza del 40% rispetto alla prel.v1, con valori compresi tra i 300 ms e i 360 ms.

7.3 Confronto complessivo

Le Figure 7.8 e 7.9 rappresentano i grafici di sintesi dei risultati ottenuti fino a questo momento nei diversi test condotti in un ambiente simulato. Le figure si riferiscono alle due diverse coperture di raggio testate, ovvero $R=9$ m e $R=16$ m, e ciascuna figura è composta da tre grafici relativi alle implementazioni con 2, 4 e 8 coppie di sorgente-destinatario. In ogni grafico, le linee continue rappresentano i PDR, mentre le linee tratteggiate indicano i valori delle latenze ottenute. Ogni curva rappresentata, sia per i PDR che per le latenze, è il risultato della media della metrica corrispondente calcolata per ogni implementazione testata fino ad ora (senza prelazione, con prelazione, prelazione ottimizzata e prelazione ottimizzata con tempo di backoff).

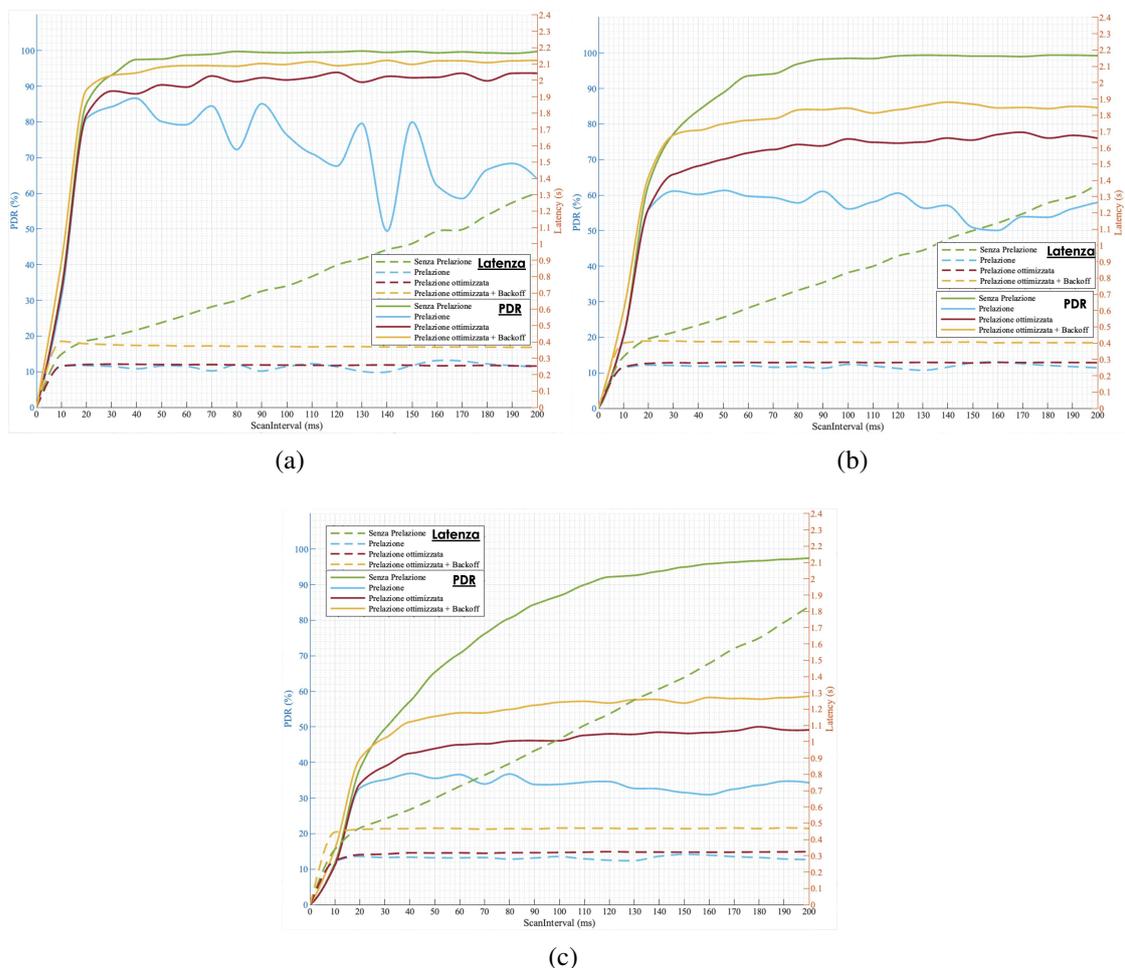


Figura 7.8: Comparazione complessiva tra l'implementazione senza prelazione e le diverse implementazioni con prelazione negli scenari con $R = 9$ m: (a) topologia con 2 SRC-DST; (b) topologia con 4 SRC-DST e (c) topologia con 8 SRC-DST

Ad esempio, nella Figura 7.8a, la curva gialla (linea continua) rappresenta il valore medio dei PDR ottenuti durante il test dell'implementazione con prelazione ottimizzata e tempo di backoff (Figura 7.2c); mentre la corrispondente linea tratteggiata alla latenza per le stesse condizioni.

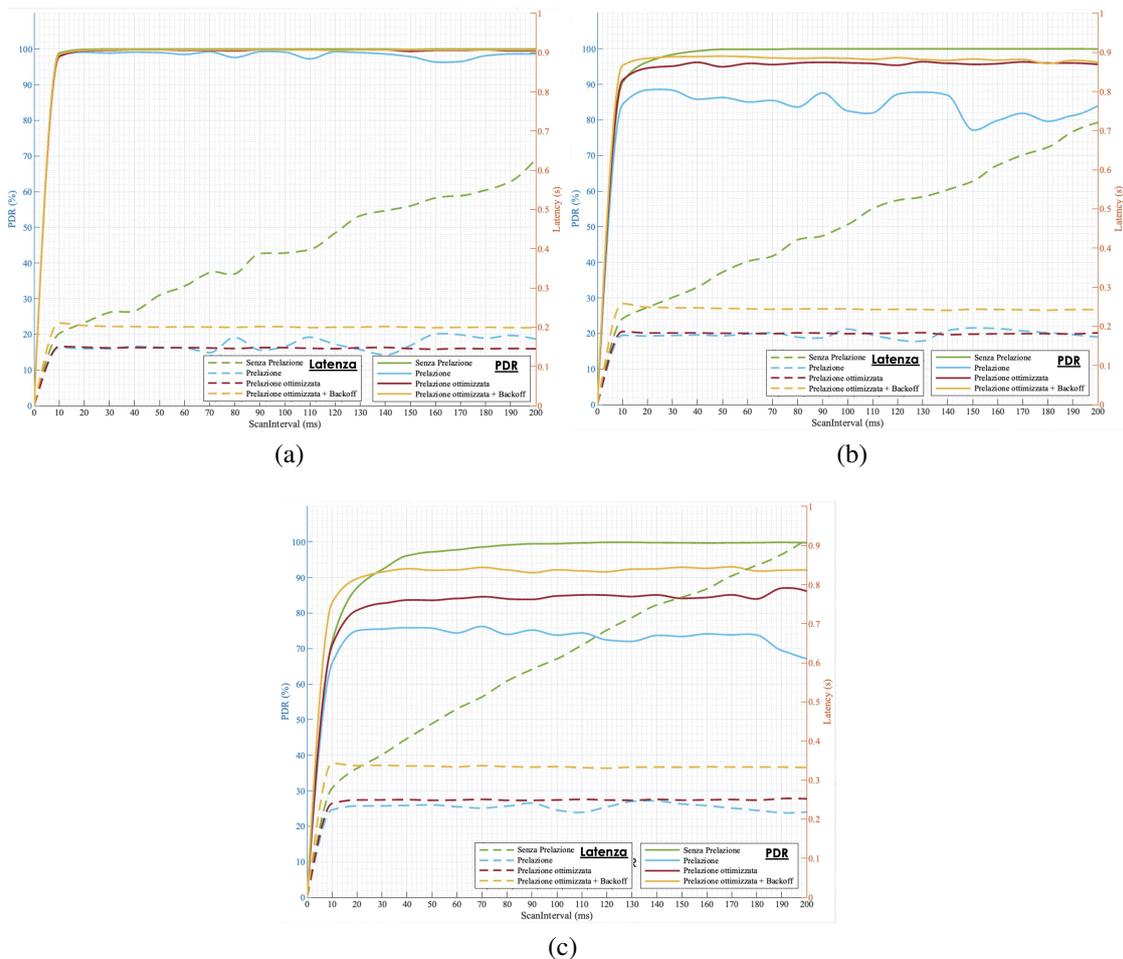


Figura 7.9: Comparazione complessiva tra l'implementazione senza prelazione e le diverse implementazioni con prelazione negli scenari con $R = 16$ m: (a) topologia con 2 SRC-DST; (b) topologia con 4 SRC-DST e (c) topologia con 8 SRC-DST

Dai grafici presenti nella Fig. 7.8, si può notare come l'implementazione ottimizzata, sia con che senza tempo di backoff, abbia permesso di ottenere un notevole miglioramento in termini di PDR in ogni scenario simulato, in relazione alla numerosità delle sorgenti. Tuttavia, se si confrontano i dati delle implementazioni con prelazione rispetto a quelle senza prelazione, non si ottengono ancora PDR comparabili. L'unica eccezione è rappresentata dal caso con 2 coppie di sorgenti (Figura 7.8a), dove la curva riferita all'implementazione con prelazione e tempo di backoff presenta un valore me-

dio intorno al 97%, unito ad un valore di latenza notevolmente inferiore e costante per ogni valore di scanInterval considerato rispetto al caso senza prelazione.

Per quanto riguarda i grafici presenti nella Figura 7.9, che riguardano i test con un raggio di copertura pari a 16 m, si può notare che l'implementazione con prelazione e tempo di backoff ha permesso di ottenere PDR superiori al 90% in ogni scenario simulato. In particolare, nella Figura 7.9a, l'implementazione con prelazione, sia con che senza backoff, ha prestazioni identiche al caso senza prelazione (100%), ma con una latenza significativamente inferiore e costante. In questa situazione, l'uso del backoff non fornisce alcun vantaggio rispetto all'implementazione solo con prelazione ottimizzata.

Effettuando un confronto sulla base del raggio di copertura, si può dedurre che questo gioca un ruolo chiave sulle prestazioni delle diverse tipologie di implementazioni proposte, anche rispetto al caso senza prelazione. La possibilità di raggiungere più nodi relay contemporaneamente, oltre a ridurre il numero di hop necessari ad un messaggio per passare dal nodo sorgente al destinatario, permette di aumentare il numero di percorsi alternativi che il messaggio può fare per raggiungere la destinazione e di conseguenza aumentare la probabilità di consegna.

7.4 Conclusioni

In questo capitolo viene presentata una nuova implementazione per la gestione dei messaggi con prelazione nei nodi relay, con l'obiettivo di migliorare le prestazioni del sistema, in particolare il Packet Delivery Rate. La nuova implementazione è stata sviluppata in linea con le specifiche del Bluetooth mesh e apporta modifiche compliant, basandosi su alcune variazioni nella gestione della fase di scansione. Tuttavia, viene mantenuta la prelazione dell'advertising sulla scansione.

Inoltre, viene introdotto un ritardo random all'inizio di ogni Advertising Event, come raccomandato dalle specifiche, per spaziare nel tempo l'inoltro dei pacchetti e ridurre la probabilità di collisione.

Dai risultati dei test effettuati e confrontati con quanto ottenuto nel capitolo precedente per la prima versione dell'implementazione con prelazione, emerge che quanto proposto permette di ottenere un significativo miglioramento delle prestazioni in termini di Packet Delivery Rate in ogni scenario simulato. In alcuni casi, i valori raggiunti si avvicinano al 100

In diversi scenari, le prestazioni ottenute sono comparabili a quelle dell'implementazione senza prelazione, ma con un notevole risparmio in termini di latenza end-to-end. Tuttavia, in situazioni in cui aumenta il numero di nodi e si ha un ridotto raggio di copertura, l'implementazione senza prelazione continua ad avere PDR più alti rispetto al caso con prelazione.

Conclusione

La ricerca condotta si è concentrata sull'implementazione e l'analisi delle prestazioni delle reti Bluetooth Mesh (BM). Bluetooth Mesh è uno standard di comunicazione wireless che consente ai dispositivi di comunicare tra loro formando una rete di dispositivi interconnessi. Questo standard ha ricevuto molta attenzione nell'ambito dell'Internet of Things (IoT), dove l'interconnessione dei dispositivi è fondamentale per la realizzazione di scenari di automazione domestica, industriale e commerciale. Nella prima parte della ricerca, l'obiettivo è stato quello di migliorare l'affidabilità nella ricezione dei messaggi di Status all'interno di una rete BM. I messaggi di Status sono messaggi di controllo che vengono inviati dai nodi della rete per informare gli altri nodi sul loro stato, sulla loro presenza o sulla loro disponibilità. L'ottimizzazione della ricezione di tali messaggi è fondamentale per garantire un funzionamento stabile e affidabile della rete. Per questo scopo, sono stati sviluppati codice e firmware per migliorare l'affidabilità della ricezione dei messaggi di Status. I test sperimentali sono stati condotti in collaborazione con Nordic Semiconductor, utilizzando schede di sviluppo proprietarie, per confrontare le prestazioni della procedura proposta con quella basata sulla configurazione standard. I risultati dei test in ambiente reale hanno dimostrato un significativo miglioramento delle prestazioni. Tuttavia, la complessità dell'architettura e la numerosità dei parametri di rete che influenzano il comportamento della rete hanno reso necessario l'utilizzo di strumenti in grado di garantire un elevato livello di affidabilità e ripetibilità dei test. Per questo motivo, nella seconda parte della ricerca è stato sviluppato un ambiente di simulazione per reti BM, utilizzando i toolbox messi a disposizione dalla Mathworks, compatibili con lo standard Bluetooth SIG. Lo sviluppo dell'ambiente di simulazione ha permesso di analizzare l'impatto di diverse configurazioni dei parametri di rete sull'affidabilità, in termini di Packet Delivery Ratio (PDR) e latenza end-to-end. I risultati dei test sperimentali hanno evidenziato come la latenza sia fortemente influenzata dai parametri temporali in gioco e dall'implementazione dello standard adottata. Per queste ragioni, grazie ai gradi di libertà interpretativa emersi nella specifica del BM e in collaborazione con il team di sviluppo del toolbox Bluetooth di MathWorks, è stata sviluppata una nuova implementazione dello standard, in particolare per quanto riguarda la gestione dei messaggi nelle fasi di scansione e advertising nei nodi relay. Questa nuova implementazione ha permesso di ottenere importanti miglioramenti delle prestazioni. In particolare, l'ottimizzazione della gestione dei mes-

saggi durante le fasi di scansione e advertising dei nodi relay ha consentito di ridurre la latenza di trasmissione dei messaggi di Status. Ciò ha migliorato l'affidabilità della rete e ha reso possibile l'utilizzo in applicazioni in cui bassi livelli di latenza sono un requisito fondamentale. Inoltre, l'utilizzo di ambienti di simulazione per testare le prestazioni delle reti BM ha dimostrato di essere un approccio molto utile per individuare le configurazioni ottimali dei parametri di rete. Questo è particolarmente importante perché, come evidenziato nella prima parte della ricerca, la complessità dell'architettura e la numerosità dei parametri di rete possono rendere difficile la realizzazione di test in ambiente reale. Infine, la ricerca condotta su Bluetooth Mesh dimostra l'importanza della collaborazione tra la ricerca accademica e l'industria per lo sviluppo di tecnologie sempre più avanzate e performanti. In particolare, la collaborazione con Nordic Semiconductor ha permesso di realizzare i setup sperimentali necessari per la validazione delle prestazioni della rete BM, mentre la collaborazione con il team di sviluppo dei toolbox Bluetooth di MathWorks ha permesso di sviluppare una nuova implementazione dello standard. In conclusione, la ricerca condotta su Bluetooth Mesh ha contribuito a migliorare l'affidabilità e le prestazioni delle reti basate su questo standard, dimostrando l'utilità dell'utilizzo di ambienti di simulazione per testare e validare le prestazioni delle reti BM. Questi risultati hanno importanti implicazioni per lo sviluppo di applicazioni IoT sempre più avanzate e complesse che richiedono reti di comunicazione affidabili e performanti.

Bibliografia

- [1] R. A. Gheorghiu, V. Iordache, and A. C. Cormos, “Analysis of handshake time for bluetooth communications to be implemented in vehicular environments,” in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2017, pp. 144–147.
- [2] *Bluetooth SIG*, “*Bluetooth specification version 5.3, Std.*, 2021. [Online]. Available: <https://www.bluetooth.com/specifications/specs/core-specification-5-3>
- [3] S. W. Song, Y. S. Lee, F. Imdad, M. T. Niaz, and H. S. Kim, “Efficient advertiser discovery in bluetooth low energy devices,” *Energies*, vol. 12, no. 9, p. 1707, 2019.
- [4] P. Pierleoni, A. Gentili, M. Mercuri, A. Belli, R. Garello, and L. Palma, “Performance improvement on reception confirmation messages in bluetooth mesh networks,” *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2056–2070, 2021.
- [5] *Bluetooth SIG*, “*Mesh Profile Version 1.0.1, Std.*, 2019. [Online]. Available: <https://www.bluetooth.com/specifications/mesh-specifications>
- [6] M. Mercuri, L. Palma, A. Belli, S. Raggiunto, and P. Pierleoni, “Performance comparison between bluetooth mesh network with and without preemption in relay nodes,” in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*. IEEE, 2022, pp. 1–8.
- [7] K. Ashton *et al.*, “That ‘internet of things’ thing,” *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.
- [8] J. Yin, Z. Yang, H. Cao, T. Liu, Z. Zhou, and C. Wu, “A survey on bluetooth 5.0 and mesh: New milestones of iot,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–29, 2019.
- [9] J. D. Gotz, O. K. Rayel, and G. L. Moritz, “Improving bluetooth mesh energy efficiency using clustering,” *Journal of Communication and Information Systems*, vol. 36, no. 1, pp. 156–165, 2021.

Bibliografia

- [10] A. Lavric, A. I. Petrariu, and V. Popa, "Sigfox communication protocol: The new era of iot?" in *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, 2019, pp. 1–4.
- [11] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of lorawan for iot: From technology to application," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/11/3995>
- [12] B. Martinez, F. Adelantado, A. Bartoli, and X. Vilajosana, "Exploring the performance boundaries of nb-iot," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5702–5712, 2019.
- [13] R. Ratasuk, N. Mangalvedhe, A. Ghosh, and B. Vejlgaard, "Narrowband Item system for m2m communication," in *2014 IEEE 80th vehicular technology conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.
- [14] W. Rzepecki and P. Ryba, "Iotsp: Thread mesh vs other widely used wireless protocols—comparison and use cases study," in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2019, pp. 291–295.
- [15] C. A. G. da Silva, E. L. dos Santos, A. C. K. Ferrari, and H. T. dos Santos Filho, "A study of the mesh topology in a zigbee network for home automation applications," *IEEE Latin America Transactions*, vol. 15, no. 5, pp. 935–942, 2017.
- [16] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "The z-wave routing protocol and its security implications," *Computers & Security*, vol. 68, pp. 112–129, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404817300792>
- [17] J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, "Uwb systems for wireless sensor networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 313–331, 2009.
- [18] *Bluetooth SIG, Bluetooth specification version 4.0, Std.*, 2016. [Online]. Available: <https://www.bluetooth.com/specifications/bluetooth-core-specification>
- [19] M. Porjazoski, P. Latkoski, and B. Popovski, "Bluetooth low energy-based smart home android solution," in *IEEE EUROCON 2019-18th International Conference on Smart Technologies*. IEEE, 2019, pp. 1–5.

- [20] P. Di Marco, P. Park, M. Pratesi, and F. Santucci, “A bluetooth-based architecture for contact tracing in healthcare facilities,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, p. 2, 2020.
- [21] R. Nusser and R. M. Pelz, “Bluetooth-based wireless connectivity in an automotive environment,” in *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No. 00CH37152)*, vol. 4. IEEE, 2000, pp. 1935–1942.
- [22] K. Mandanka, S. Mistry, and B. Tank, “Design and development of wearable device using bluetooth low energy,” in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 762–766.
- [23] K.-H. Chang, “Bluetooth: a viable solution for iot?[industry perspectives],” *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, 2014.
- [24] S. Raza, P. Misra, Z. He, and T. Voigt, “Bluetooth smart: An enabling technology for the internet of things,” in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2015, pp. 155–162.
- [25] K. Sornalatha and V. R. Kavitha, “Iot based smart museum using bluetooth low energy,” in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2017, pp. 520–523.
- [26] D. Hortelano, T. Olivares, M. C. Ruiz, C. Garrido-Hidalgo, and V. López, “From sensor networks to internet of things. bluetooth low energy, a standard for this evolution,” *Sensors*, vol. 17, no. 2, p. 372, 2017.
- [27] S. M. Darroudi and C. Gomez, “Bluetooth low energy mesh networks: A survey,” *Sensors*, vol. 17, no. 7, p. 1467, 2017.
- [28] R. Rondón, A. Mahmood, S. Grimaldi, and M. Gidlund, “Understanding the performance of bluetooth mesh: Reliability, delay, and scalability analysis,” *IEEE Internet of things journal*, vol. 7, no. 3, pp. 2089–2101, 2019.
- [29] K. Abboud, Y. Li, and S. Bermudez, “esnap: Enabling sensor network automatic positioning in iot lighting systems,” *IEEE Internet of Things Journal*, 2020.
- [30] S. M. Darroudi, C. Gomez, and J. Crowcroft, “Bluetooth low energy mesh networks: A standards perspective,” *IEEE Communications Magazine*, vol. 58, no. 4, pp. 95–101, 2020.

Bibliografia

- [31] P. Di Marco, P. Park, M. Pratesi, and F. Santucci, “A bluetooth-based architecture for contact tracing in healthcare facilities,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, 2021. [Online]. Available: <https://www.mdpi.com/2224-2708/10/1/2>
- [32] M. Jürgens, D. Meis, D. Möllers, F. Nolte, E. Stork, G. Vossen, C. Werner, and H. Winkelmann, “Bluetooth mesh networks for indoor localization,” in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, 2019, pp. 397–402.
- [33] Q. Wan and J. Liu, “Smart-home architecture based on bluetooth mesh technology,” in *IOP Conference Series: Materials Science and Engineering*, vol. 322, no. 7. IOP Publishing, 2018, p. 072004.
- [34] *Bluetooth SIG*, “*Mesh Model Version 1.0.1*, Std., 2019. [Online]. Available: <https://www.bluetooth.com/specifications/mesh-specifications>
- [35] M. Jürgens, D. Meis, D. Möllers, F. Nolte, E. Stork, G. Vossen, C. Werner, and H. Winkelmann, “Bluetooth mesh networks for indoor localization,” in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2019, pp. 397–402.
- [36] L. Montecchiari, A. Trotta, L. Bononi, and M. Di Felice, “Bluetooth mesh technology for the joint monitoring of indoor environments and mobile device localization: A performance study,” in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 193–199.
- [37] X. Tong, L. Wang, and Y. Cui, “Research on indoor positioning based on smart home bluetooth networking,” in *2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*. IEEE, 2022, pp. 1–4.
- [38] C. Martínez, L. Eras, and F. Domínguez, “The smart doorbell: A proof-of-concept implementation of a bluetooth mesh network,” in *2018 IEEE third Ecuador technical chapters meeting (ETCM)*. IEEE, 2018, pp. 1–5.
- [39] S. İ. Taştan and G. Dalkılıç, “Smart home system using internet of things devices and mesh topology,” in *2021 6th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2021, pp. 407–412.
- [40] Q. T. Tran, D. Doan, M. S. Nguyen *et al.*, “An approach of ble mesh network for smart home application,” in *2020 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2020, pp. 170–174.

- [41] V. M. Dvoynikov, V. A. Smirnov, and D. A. Buriylov, "Comparative analysis of mesh and thread networks and their application possibility in the " smart home" systems," in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE, 2021, pp. 8–11.
- [42] I. Sergi, T. Montanaro, M. C. Gammariello, and L. Patrono, "The use of bluetooth mesh networking in iot-aware applications," in *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, 2021, pp. 01–06.
- [43] X. Zheng, S. Xue, H. Cao, F. Wang, and M. Zhang, "A cost-efficient smart iot device controlling system based on bluetooth mesh and cloud computing," in *2020 Chinese Automation Congress (CAC)*. IEEE, 2020, pp. 3374–3379.
- [44] T.-Y. Lee, P. H. Truong, C.-K. Lee, and G.-M. Jeong, "Range extension of led control systems using a bluetooth mesh network," in *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2017, pp. 166–167.
- [45] A. A. Veiga and C. J. Abbas, "Proposal and application of bluetooth mesh profile for smart cities' services," *Smart Cities*, vol. 2, no. 1, pp. 1–19, 2018.
- [46] E. De Leon and M. Nabi, "An experimental performance evaluation of bluetooth mesh technology for monitoring applications," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.
- [47] V. M. Dvoynikov, V. A. Smirnov, and D. A. Burylov, "Implementation of a monitoring system for an electrical network based on a contactless temperature sensor and a hall effect current sensor," in *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE, 2022, pp. 604–607.
- [48] O. Witham, L. N. Johnston, M. Xiao, J. Feng, N. Zhou, and G. Shaker, "Battery-less wireless water leak detection system," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2019, pp. 1–4.
- [49] F. Álvarez, L. Almon, H. Radtki, and M. Hollick, "Bluemergency: Mediating post-disaster communication systems using the internet of things and bluetooth mesh," in *2019 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 2019, pp. 1–8.
- [50] T. C. Y. Lam, S. S. L. Yew, and S. L. Keoh, "Bluetooth mesh networking: An enabler of smart factory connectivity and management," in *2019 20th Asia-Pacific*

Bibliografía

- Network Operations and Management Symposium (APNOMS)*. IEEE, 2019, pp. 1–6.
- [51] C. Garrido-Hidalgo, D. Hortelano, L. Roda-Sanchez, T. Olivares, M. C. Ruiz, and V. Lopez, “Iot heterogeneous mesh network deployment for human-in-the-loop challenges towards a social and sustainable industry 4.0,” *Ieee Access*, vol. 6, pp. 28 417–28 437, 2018.
- [52] P. Seymer, D. Wijesekera, and C.-D. Kan, “Secure outdoor smart parking using dual mode bluetooth mesh networks,” in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–7.
- [53] ———, “Smart parking zones using meshed bluetooth sensor networks,” in *Smart Cities, Green Technologies and Intelligent Transport Systems: 8th International Conference, SMARTGREENS 2019, and 5th International Conference, VE-HITS 2019, Heraklion, Crete, Greece, May 3–5, 2019, Revised Selected Papers 8*. Springer, 2021, pp. 245–269.
- [54] D. Pérez-Díaz-De-Cerio, M. García-Lozano, A. V. Bardají, J.-L. Valenzuela *et al.*, “Bluetooth mesh analysis, issues, and challenges,” *IEEE Access*, vol. 8, pp. 53 784–53 800, 2020.
- [55] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, “The bluetooth mesh standard: An overview and experimental evaluation,” *Sensors*, vol. 18, no. 8, p. 2409, 2018.
- [56] S. M. Darroudi, R. Caldera-Sánchez, and C. Gomez, “Bluetooth mesh energy consumption: A model,” *Sensors*, vol. 19, no. 5, p. 1238, 2019.
- [57] D. Perez-Diaz-de Cerio, J. Valenzuela, M. Garcia-Lozano, Á. Hernández-Solana, and A. Valdovinos, “Bmads: Ble mesh asynchronous dynamic scanning,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2558–2573, 2020.
- [58] Á. Hernández-Solana, D. Perez-Diaz-de Cerio, M. García-Lozano, A. Valdovinos, and J. Valenzuela, “Psm-dmo: power save mode and discontinuous ble mesh operation,” *Computer Networks*, vol. 213, p. 109114, 2022.
- [59] A. Aijaz, A. Stanoev, D. London, and V. Marot, “Demystifying the performance of bluetooth mesh: Experimental evaluation and optimization,” in *2021 Wireless Days (WD)*. IEEE, 2021, pp. 1–6.
- [60] D. Perez-Diaz-de Cerio, A. Hernandez-Solana, M. Garcia-Lozano, A. V. Bardají, and J.-L. Valenzuela, “Speeding up bluetooth mesh,” *IEEE access*, vol. 9, pp. 93 267–93 284, 2021.

- [61] *Bluetooth SIG*, “*Mesh Profile Version 1.0.1*”, Std., 2019. [Online]. Available: <https://www.bluetooth.com/specifications/mesh-specifications>