

Scuola di Scienze
Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea in Informatica

**Un sistema di monitoraggio
per reti BLE Mesh:
progettazione, implementazione e analisi per
l'identificazione della topologia e
la valutazione del traffico di rete**

**Relatore:
Dott.
ANGELO TROTTA**

**Presentata da:
GABRIELE APRILE**

*We have to remember these days, because
there's no guarantee that they will last forever.*

Enjoy them as long as they last.

Sebastian Vettel

2013 Formula 1 United States Grand Prix

Abstract

Lo sviluppo di reti IoT estese ha evidenziato la necessità di strumenti per il monitoraggio della loro struttura e delle loro performance. Questo lavoro di tesi propone la progettazione e l'implementazione di un sistema di monitoraggio per reti IoT basate sulla tecnologia wireless Bluetooth Low Energy Mesh. Il sistema consente di analizzare la topologia della rete, visualizzandone una rappresentazione grafica che include i nodi, le distanze stimate e i relativi link tra essi. Permette inoltre di estrarre un'ampia gamma di parametri dai nodi, comprendenti sia informazioni sulle caratteristiche del dispositivo, sia dati relativi all'identità del nodo all'interno della rete, sia le chiavi utilizzate per la comunicazione sicura all'interno del sistema. Proprio la funzione di estrazione dei parametri, evoluzione di un classico servizio di messaggistica, è stata sfruttata per condurre esperimenti volti a valutare le performance della rete. In particolare, l'analisi si è focalizzata su caratteristiche quali i tempi di risposta e l'affidabilità complessiva del sistema.

Parole chiave: *BLE Mesh, Bluetooth, IoT, monitoraggio di rete, topologia di rete*

Indice

1	Introduzione	1
2	Stato dell'arte	3
2.1	Internet of Things	3
2.2	Bluetooth	5
2.3	Bluetooth Low Energy	5
2.4	Bluetooth Low Energy Mesh	6
2.4.1	Panoramica generale	6
2.4.2	Come BLE Mesh soddisfa i requisiti dell' IoT	14
2.4.3	Confronto con altre tecnologie wireless	16
2.5	Il BLE Mesh negli articoli scientifici	17
3	Definizione del Framework: Progettazione	19
3.1	Struttura del framework	19
3.2	Descrizione ad alto livello dei servizi	20
3.3	Funzionalità secondarie	22
4	Definizione del Framework: Implementazione	23
4.1	Tecnologie utilizzate	23
4.1.1	Hardware	23
4.1.2	Software	23
4.2	Architettura della rete	25
4.3	Servizi	26
4.3.1	<i>discoverTopology</i> Service	26
4.3.2	<i>parameterRequest</i> Service	30
4.4	Dashboard	31
4.4.1	Funzionamento della Dashboard	32
4.5	Logging e salvataggio dati	33
4.5.1	Logging	34

4.5.2	Salvataggio dati	35
4.5.3	Salvataggio immagini della topologia di rete	37
5	Validazione	39
5.1	Setup sperimentale	39
5.2	Esperimenti	40
5.2.1	Esperimento 1 - <i>discoverTopology</i>	40
5.2.2	Esperimento 2 - <i>parameterRequest</i>	43
5.3	Analisi della performance di rete	45
5.3.1	Metriche	45
5.3.2	Procedimento	46
5.3.3	Risultati	47
6	Conclusione e sviluppi futuri	49
6.1	Conclusione	49
6.2	Sviluppi futuri	50
A	Dati di supporto alla validazione: Esperimento 1	53
A.1	File di log manuale	54
A.2	File di log	54
A.3	File <i>discoverTopology.csv</i>	71
B	Dati di supporto alla validazione: Esperimento 2	79
B.1	File di log manuale	79
B.2	File di log	80
B.3	File <i>parameterRequest.csv</i>	84
	Abbreviazioni e acronimi	89
	Bibliografia	91
	Sitografia	95

Elenco delle figure

2.1	Differenze tra Bluetooth Classic e Bluetooth Low Energy	7
2.2	Lo stack di BLE Mesh	8
2.3	Un esempio di meccanismo pub/sub: le prime tre luci attendono messaggi inviati all'indirizzo "Reception" (ne sono iscritti), mentre il primo interruttore invia (pubblica) i messaggi in quel gruppo	11
2.4	Confronto di BLE Mesh con altre tecnologie wireless	17
3.1	Architettura logica del sistema	19
3.2	Dashboard	20
3.3	Diagramma Use Case	21
4.1	Il microcontroller ESP32	24
4.2	Architettura del sistema nel dettaglio	25
4.3	Schema grafico di Implementazione 1 - <i>Heartbeat Publishing</i>	27
4.4	Schema grafico di Implementazione 3 - <i>Messaggistica On-Demand</i>	29
4.5	Diagramma di sequenza UML di Implementazione 3 - <i>Messaggistica On-Demand</i>	30
4.6	Alcuni esempi di output testuali su shell in seguito alla chiamata dei Servizi	34
5.1	Spazio fisico di deploy della rete	40
5.2	Posizionamento finale dei nodi di <i>Esperimento1</i>	41
5.3	Spostamento di 0x000d riflesso sul grafico	43
5.4	Spostamento di 0x000d mostrato sulla planimetria	43
5.5	Posizionamento dei nodi di <i>Esperimento 2</i>	44
5.6	Grafico con metriche per l'analisi dei risultati di <i>Esperimento 2</i>	47

Elenco delle tabelle

4.1	Valutazione del segnale RSSI	33
A.1	File di log scritto a mano durante l'esperimento per discoverTopology	54
B.1	File di log scritto a mano durante l'esperimento per parameterRequest	79

Capitolo 1

Introduzione

L'*Internet of Things (IoT)* è un tema di grande importanza tecnica, sociale ed economica. Da oltre un decennio il settore è in costante espansione anche a livello economico: solo in Italia, nel 2023, il valore del mercato IoT ha raggiunto gli 8,9 miliardi di euro, mentre a livello globale alcune previsioni stimano che entro il 2025 i dati generati da reti IoT supereranno i 73 zettabyte, equivalenti a 73 mila miliardi (73.000.000.000.000) di gigabyte.

L'IoT ha trasformato profondamente numerosi settori: dalla gestione di edifici, città, industrie, case connesse, logistica, fino all'agricoltura e alla medicina. Questo ecosistema consente un significativo miglioramento in termini di affidabilità, sostenibilità ed efficienza operativa: ciò è reso possibile dall'accesso ottimizzato alle informazioni e dall'interconnessione tra sistemi, che, tramite lo scambio continuo di dati, possono influenzarsi reciprocamente e agire in modo coordinato per ottimizzare processi e risorse.

Una delle tecnologie wireless per IoT più efficaci e in più rapida espansione, il *Bluetooth Low Energy (BLE)*, si distingue per efficienza e completezza. Basata sulla celeberrima e già ampiamente diffusa tecnologia Bluetooth, con la sua variante *Mesh* sta guadagnando popolarità grazie alla capacità di supportare reti con decine di migliaia di dispositivi, garantendo comunicazioni efficienti e un basso consumo energetico.

Tuttavia può essere difficile tenere monitorato questo tipo di reti: pensate per essere dinamiche e supportare un alto numero di dispositivi e sensori che comunicano continuamente tra di loro, guasti tecnici e rallentamenti sono sempre rischi concreti. Modifiche alla configurazione possono risultare onerose in termini di tempo e competenze richieste, rappresentando un ostacolo in particolar modo per utenti me-

no esperti, che desiderano adottare questa tecnologia per migliorare la produttività riducendo al minimo i problemi operativi.

Interviene in aiuto questo lavoro di tesi, che si propone di *progettare e implementare un sistema di monitoraggio per reti BLE Mesh* che consenta all'utente di monitorare costantemente la rete, acquisendo informazioni sulla sua struttura, sui collegamenti tra i dispositivi e monitorandone le performance, tramite la semplice interazione con una dashboard da PC.

Capitolo 2

Stato dell'arte

2.1 Internet of Things

L'espressione "*Internet of Things*" (IoT), è stata formulata per la prima volta nel 1999 dall'ingegnere inglese Kevin Ashton, cofondatore dell'Auto-ID Center del Massachusetts Institute of Technology. Pur essendo un concetto noto da più di due decenni, non esiste un consenso universale sulla sua definizione nella letteratura accademica o tecnica [1]. Numerose definizioni sono state proposte nel corso degli anni, spesso differenti a seconda dello specifico ambito trattato.

Una definizione generale, accettata anche dall'*IEEE IoT Initiative* descrive l'IoT come: "...un'infrastruttura di rete dinamica e globale con capacità di autoconfigurazione basate su protocolli di comunicazione standard e interoperabili in cui le cose fisiche e virtuali hanno identità, attributi fisici e personalità virtuali, utilizzano interfacce intelligenti e sono perfettamente integrate nella rete di informazione" [2]

Nell'ambito delle Wireless Sensor Networks (WSN), IEEE evidenzia una differenza tra una semplice rete di sensori (appunto, una WSN) e un ambiente IoT, definendo la prima come un insieme di nodi, dove ogni nodo è collegato a uno o più sensori, mentre lo scopo di un sistema IoT è quello di "*aggiungere intelligenza agli oggetti*" in modo tale che possano svolgere i task assegnati, anche scambiando dati tra di loro, per raggiungere un determinato obiettivo senza necessità di intervento umano[3].

Alla realizzazione dell'IoT possono partecipare diverse *tecnologie* [4]:

- *Sensori e attuatori*: elementi al centro dell'IoT, sono agenti che, tramite rilevatori d'ambiente, interagiscono con il mondo fisico senza stretta necessità di intervento umano
- *Tecnologie di connettività*: per lo scambio dei dati (Bluetooth, Wi-Fi, ...)
- *Cloud computing*: per archivio, elaborazione e analisi dei dati raccolti
- *Analisi dei big data*: per estrarre informazioni e identificare modelli, anche grazie a tecniche di apprendimento automatico, visualizzazione dei dati e modelli di analisi predittiva
- *Tecnologie per la sicurezza e la privacy*: crittografia, controlli di accesso e sistemi di rilevamento delle intrusioni per rendere i sistemi il più sicuro possibile.

L'IoT può essere utilizzato in una vasta gamma di *ambiti e applicazioni*, tra cui:

- *Assistenza sanitaria*: monitoraggio dei pazienti da remoto, rilevazione dei parametri vitali e loro analisi per costruire modelli e identificare problemi di salute tempestivamente
- *Produzione industriale*: monitoraggio delle prestazioni di macchine, rilevazione di guasti e ottimizzazione dei processi di produzione anche tramite controllo delle condizioni ambientali, oltre a gestione di inventario e controllo qualità dei prodotti
- *Retail*: monitoraggio del comportamento dei clienti per ottimizzare il layout del negozio, tracciamento delle catene di fornitura e controllo spedizioni
- *Agricoltura*: controllo condizioni del terreno e dell'ambiente, per un'ottimizzazione delle risorse impiegate, oltre al monitoraggio del bestiame; grazie al basso intervento umano richiesto da alcuni dispositivi, l'IoT è fondamentale per una supervisione in località remote e di difficile accesso
- *Trasporti*: monitoraggio prestazioni dei veicoli e ottimizzazione dei percorsi (ad esempio per la rete di trasporti pubblici), oltre a controllo delle condizioni del carico per il mantenimento delle condizioni ottimali.

L'Internet of Things è in continua evoluzione e crescita, e punta al miglioramento della qualità della vita, della produttività e della resa economica. Il futuro si prospetta ricco di sviluppi, con applicazioni che sfrutteranno sempre di più tecnologie come l'edge computing, l'IA, la blockchain e molto altro.

2.2 Bluetooth

Bluetooth [5] è uno standard di comunicazione wireless a corto raggio usato per lo scambio di dati sulle brevi distanze. Sviluppato dal *Bluetooth Special Interest Group (Bluetooth SIG)*, è stato introdotto nel 1998 e rappresenta tutt'oggi una delle tecnologie wireless di maggiore successo grazie alla grande flessibilità offerta agli sviluppatori, continuo sviluppo di nuove features, con rinnovamento e supporto di quelle attualmente presenti sul mercato per rimanere al passo delle esigenze in continua espansione della connettività wireless.

Esistono due tipi di soluzioni radio BT: il **Bluetooth Classic** e il **Bluetooth Low Energy (BLE)**.

Il **Bluetooth Classic** è una tecnologia radio a basso consumo che trasmette dati in streaming su 79 canali, nella banda di frequenza a 2,4GHz usata in ambito industriale, scientifico e medico senza necessità di licenza. Tecnologia che supporta la comunicazione punto-a-punto, è usata principalmente in due campi:

- *Streaming audio*: è diventato il protocollo radio standard per altoparlanti wireless, cuffie e sistemi di intrattenimento per auto
- *Trasferimento dati*: è impiegato da diversi dispositivi quali ad esempio periferiche PC, elettrodomestici, fitness tracker e sensori per la salute

Il Bluetooth ha subito numerose evoluzioni nel tempo, con il rilascio di nuove versioni che hanno apportato significativi miglioramenti nelle prestazioni e introdotto funzionalità sempre più avanzate. Ad oggi, la versione più recente è il Bluetooth 6.0, annunciato a settembre 2024. Questa versione si distingue per affrontare problematiche legate alla sicurezza, migliorare l'accuratezza nel calcolo delle distanze tra dispositivi e consentire la trasmissione di quantità maggiori di dati in pacchetti di minore dimensione, ottimizzando così l'efficienza complessiva del sistema.

2.3 Bluetooth Low Energy

Il secondo tipo di Bluetooth descritto dal BSIG è il **Bluetooth Low Energy (BLE)**. Introdotto con la versione 4.0 del *Bluetooth Core Specification*, è progettato per un funzionamento a bassissima potenza. Trasmette dati su 40 canali in banda 2.4GHz e offre grande flessibilità per progettare reti in base alle necessità del mercato. Supporta infatti diverse possibili topologie di rete: oltre al punto-a-punto, supporta broadcast e mesh, con quest'ultima aprendo le porte alla creazione di reti affidabili di larga scala.

Oltre agli ambiti già citati per il Bluetooth Classic, Bluetooth LE è ampiamente impiegato anche in:

- *Reti di dispositivi*: la topologia Mesh è ideale per creare sistemi di controllo, monitoraggio e automazione in cui decine, centinaia o migliaia di dispositivi devono comunicare tra loro in modo affidabile e sicuro
- *Servizi di localizzazione*: per informazioni su punti di interesse e ricerca di oggetti, nonché sistemi di posizionamento come ad esempio localizzazione in tempo reale, con features che consentono ai dispositivi di determinare presenza (*Advertising*), distanza (*RSSI*) e direzione di altri dispositivi (*Direction Finding*)

Il Bluetooth Low Energy è in continua evoluzione, offrendo nuove soluzioni e migliorando quelle esistenti, sia per comunicazioni a corto che a lungo raggio, soddisfacendo esigenze diverse in diversi settori. Tra le innovazioni più recenti spiccano *Auracast* [6], che rivoluziona la condivisione audio, *Bluetooth Direction Finding* [7], che permette di individuare l'esatto posizionamento del dispositivo e **Bluetooth Mesh Networking**, la tecnologia al centro di questo lavoro di tesi.

Grazie ai suoi nuovi modelli di connessione, alla gestione efficiente della larghezza di banda e alle tecnologie avanzate per il risparmio energetico, il Bluetooth Low Energy segna un significativo progresso rispetto al Bluetooth tradizionale, fornendo una soluzione wireless ideale per dispositivi IoT e indossabili.

2.4 Bluetooth Low Energy Mesh

Bluetooth Mesh è un protocollo wireless rilasciato ufficialmente nel 2017 e basato su Bluetooth Low Energy. Sviluppato per fornire una copertura estesa e una maggiore robustezza, nonché costi ridotti e bassi consumi energetici, consente la **comunicazione multi-a-molti**, aprendo la possibilità alla creazione di reti Bluetooth di tipo mesh. Le sue caratteristiche lo rendono ideale per molte applicazioni basate su IoT come l'automazione domestica e industriale. Le specifiche di Bluetooth Mesh sono state definite nel *Mesh Profile* [8] e nel *Mesh Model* [9] dal Bluetooth SIG.

2.4.1 Panoramica generale

Architettura stack del BLE Mesh

L'**architettura** del *BLE Mesh networking stack* è costruita sull'architettura dell'intero stack BLE, che fornisce le primitive fondamentali per la comunicazione wireless.

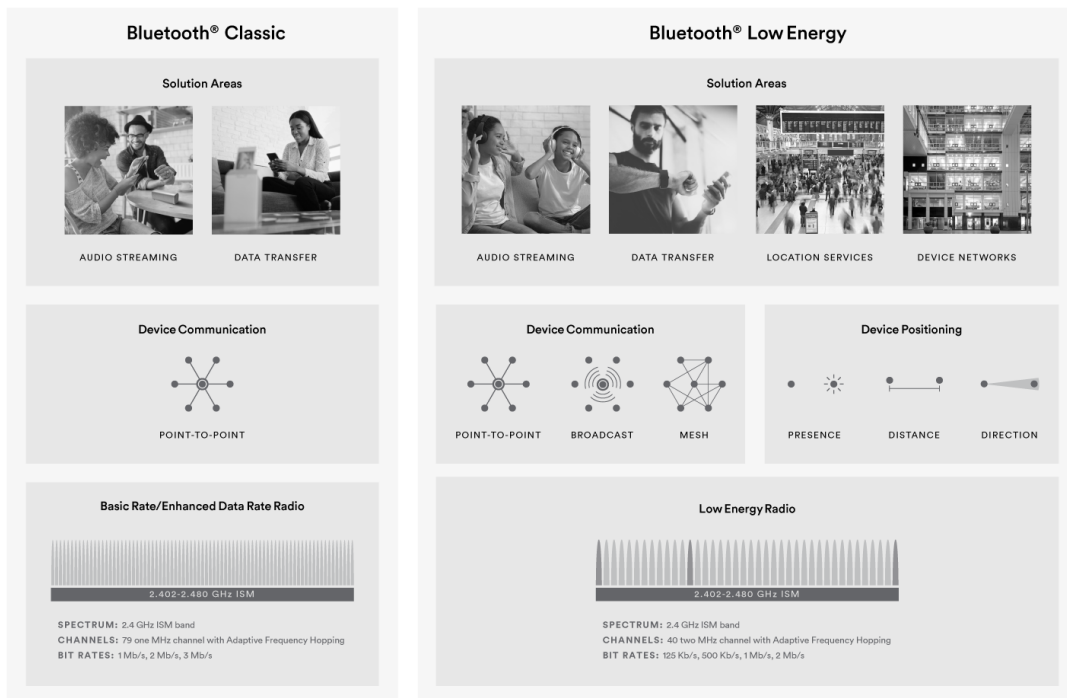


Figura 2.1: Differenze tra Bluetooth Classic e Bluetooth Low Energy

Come da Figura 2.2, osserviamo la presenza di 7 layer:

- **Bearer Layer:** definisce i meccanismi di trasmissione delle Protocol Data Unit (PDU); due sono i bearer principali usati per la comunicazione, l'*Advertising bearer* (che sfrutta advertising e scanning di BLE) e il *GATT bearer* (che usa il Protocollo Proxy)
- **Network Layer:** gestisce l'indirizzamento, la formattazione, la crittografia e l'autenticazione per le trasmissioni di dati, nonché le decisioni di inoltro ed eliminazione dei messaggi
- **Lower Transport Layer:** definisce la segmentazione e il riassettaggio dei messaggi dell'Upper Transport Layer in PDU
- **Upper Transport Layer:** è responsabile della gestione di crittografia, della decifrazione e dell'autenticazione dell'application data e offre riservatezza dei messaggi di accesso; trasmette messaggi di Heartbeat
- **Access Layer:** definisce come layer superiori possono utilizzare l'Upper Transport Layer, è responsabile della specifica del formato dell'application data e

dell'implementazione delle funzioni di crittografia/decrittografia, verifica che i dati in arrivo contengano le AppKey e NetKey corrette prima di inoltrare i dati agli strati successivi

- **Foundation Model Layer:** definisce gli stati, i messaggi e i modelli necessari per configurare e gestire mesh; sono contemplati due set di modelli, il modello *Configuration Client / Configuration Server* e il modello *Health Client / Health Server*
- **Model Layer:** modelli operazionali definiti (opzionalmente) dal Bluetooth SIG e dagli sviluppatori (*vendor models*) per definire comportamenti specifici nei loro prodotti

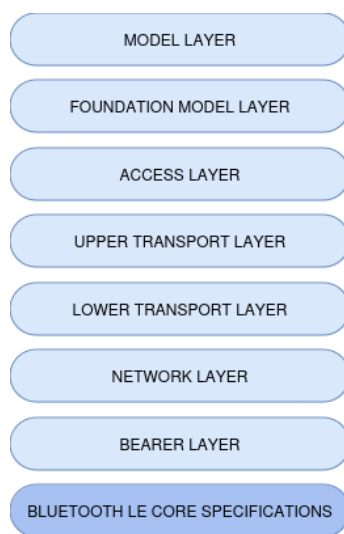


Figura 2.2: Lo stack di BLE Mesh

Tipi di nodi

Sono definiti **nodi** i dispositivi che fanno parte di una rete mesh. Dispositivi che non fanno parte di una rete sono definiti “dispositivi unprovisioned”. Il processo che trasforma un dispositivo in un nodo è chiamato “**provisioning**”. In generale tutti i nodi possono trasmettere e ricevere i messaggi.

Possono essere raggruppati in 4 tipologie:

- **Relay Node:** possono ritrasmettere i messaggi ricevuti, permettendo così alla rete di estendere la sua area di copertura e consentendo ai messaggi di compiere più “hops”; per prevenire la congestione della rete, ogni PDU è dotata di un *Time to Live* (TTL), che indica il numero massimo di hop che può effettuare nella rete prima di essere scartata

- **Low Power Node (LPN)**: spesso nodi funzionanti con uso di batteria, e che per questo mirano al minimo consumo energetico possibile, rimanendo in stato quiescente e risvegliandosi solo per comunicare con un nodo “amico” che supporti la funzionalità; solitamente sono nodi che trattano messaggi solo occasionalmente
- **Friend Node**: supporta gli LPN inoltrando, su richiesta, i messaggi a loro destinati che hanno precedentemente conservato in cache
- **Proxy node**: nodo “ponte” che permette la comunicazione tra dispositivi che supportano BLE ma non lo stack di comunicazione BLE Mesh

Messaggi

Tutta la comunicazione in una rete mesh è **message oriented**. I messaggi possono essere definiti anche da utente assegnando loro un **OPCODE** che ne identifica univocamente la categoria.

Possono essere:

- **Acknowledged**, se un messaggio richiede una risposta dai nodi che lo ricevono. Questo meccanismo è usato sia per avere conferma di ricezione che per mandare dei dati al nodo iniziale richiedente; per definizione di “ack”, se un nodo non riceve una risposta, manderà una nuova richiesta: per questo i messaggi ack devono essere idempotenti
- **Unacknowledged**, se non necessitano di risposta

Diversi casi d’uso di BLE Mesh usano messaggi *unack*, i cui possibili risvolti negativi sono mitigati da un meccanismo di ritrasmissione di più copie dello stesso messaggio in rapida successione. L’utilizzo di messaggi *ack* potrebbe risultare inefficace su reti di grandi dimensioni, poiché l’elevato volume di messaggi generato potrebbe causare congestione e compromettere le prestazioni complessive della rete.

I messaggi si suddividono in tre categorie principali, ciascuna corrispondente a un tipo di operazione supportata da Bluetooth Mesh:

- **GET**: per richiedere lo stato di uno o più elementi
- **SET**: per cambiare il valore di un dato stato
- **STATUS**: usato come
 - risposta ad una GET, contiene i valori richiesti
 - come *ack* ad una SET
 - mandata indipendentemente al verificarsi di determinati eventi

Indirizzamento

Ci sono tre tipi di **indirizzi** possibili:

- **Indirizzo Unicast:** assegnato ad ogni nodo al momento del provisioning, esso non può essere cambiato durante il suo periodo di vita; gli indirizzi unicast sono assegnati sequenzialmente a partire da un valore arbitrario
- **Indirizzo di Gruppo:** sono indirizzi multicast per comunicare con più nodi contemporaneamente; alcuni indirizzi sono fissati (denominati *All-proxies*, *All-friends*, *All-relays* and *All-nodes*, per raggiungere tutti i nodi di un determinato tipo) mentre altri sono assegnabili dinamicamente
- **Indirizzo Virtuale:** può essere assegnato a uno o più elementi distribuiti su uno o più nodi ed è rappresentato come un valore UUID a 128 bit. Simile a un'etichetta, può essere associato a qualsiasi elemento. Gli indirizzi virtuali, spesso preconfigurati durante la fabbricazione del dispositivo, sono ideali ad esempio per scenari quali l'indirizzamento unificato di tutti i proiettori per sale riunioni realizzati da un determinato produttore

Meccanismo di Publish / Subscribe

In una rete Bluetooth mesh i messaggi vengono scambiati sfruttando il meccanismo **publish-subscribe**. Viene definito *pubblicazione* l'atto di inviare un messaggio, mentre la *sottoscrizione* è una configurazione utilizzata per consentire ai nodi di ricevere solo i messaggi inviati a indirizzi specifici.

Di norma, i messaggi vengono inviati a indirizzi di gruppo o virtuali; questi indirizzi sono progettati con nomi chiari e significativi, in modo da risultare facilmente comprensibili per l'utente finale e intuitivi da utilizzare durante la configurazione dei dispositivi (si veda esempio in figura 2.3).

Modelli

Sono genericamente classificati in *Server model* e *Client model*, con funzionalità simili alle architetture di rete client/server conosciute.

- **Server model:** definisce una raccolta di stati che consentono la configurazione di vari aspetti del dispositivo, oltre alle transizioni di stato e messaggi che il modello può inviare o ricevere. Definisce inoltre i comportamenti relativi ai messaggi, agli stati e alle transizioni di stato supportati
- **Client model:** non definisce alcuno stato, ma definisce i messaggi che può inviare o ricevere per compiere operazioni GET, SET, e acquisire lo STATUS degli stati definiti nel modello server corrispondente

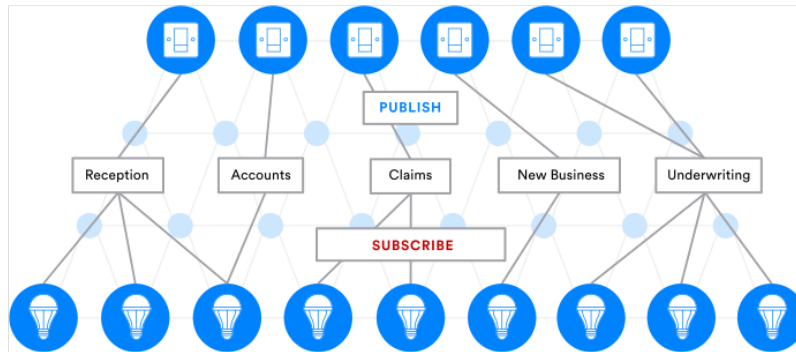


Figura 2.3: Un esempio di meccanismo pub/sub: le prime tre luci attendono messaggi inviati all’indirizzo “Reception” (ne sono iscritti), mentre il primo interruttore invia (pubblica) i messaggi in quel gruppo

I *modelli personalizzati* possono essere creati partendo dai modelli di base, i quali non sono modificabili direttamente ma possono essere ampliati integrando nuove funzionalità.

Molti dispositivi diversi condividono stati che, pur appartenendo a contesti differenti, hanno un significato semantico simile (e.g. lo stato ON / OFF può essere applicato a dispositivi come luci, ventilatori o prese di corrente). Per uniformità e riutilizzabilità del codice, la specifica del modello Bluetooth Mesh introduce il concetto di *stati generici*, come ad esempio *Generic OnOff*, definendone anche modelli (Generic OnOff Server/Client) e relativi messaggi generici (Generic OnOff Get/Set).

Relay di messaggi

La ritrasmissione dei messaggi in una rete può avvenire secondo diverse metodologie [10]. Come già accennato, in una rete BLE Mesh i dispositivi possono comunicare anche se non si trovano direttamente nel range radio l’uno dell’altro. Questo è possibile grazie a un processo chiamato **relaying**, in cui i messaggi vengono trasmessi attraverso la rete passando da un dispositivo all’altro, fino a raggiungere la destinazione finale. Copie dello stesso messaggio possono viaggiare simultaneamente attraverso percorsi diversi. La prima copia che raggiunge la destinazione viene elaborata, mentre le copie successive vengono scartate. Questa capacità *multi-hop multi-path* fornisce alle reti ridondanza in modo da aumentare le probabilità che i messaggi possano comunque essere recapitati, anche quando uno o più nodi relay non sono disponibili, migliorando notevolmente l’affidabilità della rete.

Esistono due tecniche principali di relaying:

- **Managed Flooding:** una versione più efficiente del flooding non controllato (in cui i messaggi vengono sempre ritrasmessi dai nodi). Nel Managed Flooding i nodi Relay ritrasmettono i messaggi solo in determinate circostanze. Ad esempio verificano il valore del TTL (Time-To-Live) del messaggio: se il TTL è 0, la ritrasmissione non avviene. In alternativa, controllano la propria cache, dove sono memorizzati i dettagli dei messaggi già inoltrati. Se il messaggio è presente nella cache, viene considerato come già ricevuto e ritrasmesso in precedenza, ignorando la nuova copia del messaggio ricevuto.
- **Directed Forwarding:** il Relay ritrasmette un messaggio solo se sa di essere un membro di una sequenza di nodi attraverso i quali il messaggio può viaggiare per raggiungere la sua destinazione. Se la destinazione non può essere raggiunta tramite questo nodo, il messaggio viene scartato. I path possibili vengono creati manualmente durante la configurazione o sono creati e gestiti automaticamente da varie procedure di sistema. Questo metodo più sofisticato permette di ridurre trasmissioni radio, lasciando più banda al resto della messaggistica, riducendo anche la probabilità di collisioni.

Privacy e sicurezza

Per proteggere la **privacy** dei dati scambiati, tutti i messaggi in BT Mesh sono criptati e autenticati. La **sicurezza** è gestita su più livelli: rete, applicazione e dispositivo, garantendo così protezione, autenticità e integrità dei messaggi. A differenza di BLE dove i vari livelli di sicurezza dei dispositivi sono decisi del designer a seconda della valutazione dei rischi, nel BLE Mesh la sicurezza delle reti è *imposta*, poiché il focus si sposta sulla sicurezza dell'intera rete piuttosto che su quella dei singoli dispositivi.

Di seguito alcuni degli elementi chiave riguardanti questo tema:

- **Separation of concerns:** l'idea di fondo è che solo i nodi che necessitano di una particolare informazione possano accedervi, mentre i nodi relay possono inoltrare il messaggio senza conoscerne il contenuto. Rientra sotto questo aspetto anche l'uso delle chiavi di sicurezza:
 - *DevKey*, chiave univoca di ogni nodo, conosciuta solo dal dispositivo e dal Provisioner. Serve a rendere sicuro il processo di configurazione
 - *NetKey*, protegge la comunicazione al Network Layer; un dispositivo in possesso di questa chiave viene identificato come facente parte della rete/sottorete a cui quella chiave fa riferimento

- *AppKey*, condivisa da un sottoinsieme di nodi che partecipano a un'applicazione comune (e.g. una AppKey del sistema di illuminazione verrebbe condivisa tra interruttori della luce e lampadine, ma non con un termostato o un sensore di movimento). Protegge la comunicazione dell'Upper Transport Layer.
- **Area isolation:** ogni sottorete è identificata usando la sua NetKey; per questo principio l'accesso ad una sottorete può essere gestito dalla sottorete stessa senza interventi esterni
- **Secure Device Provisioning:** l'aggiunta di ogni nuovo nodo alla rete può rappresentare un rischio alla sicurezza e per questo deve avvenire secondo certi criteri.
Distinguiamo 3 tipi di provisioning:
 - *Direct provisioning:* avviene attraverso 5 step.
 - I Beaconing - un dispositivo unprovisioned X lancia un broadcast (advertising packet) indicando la disponibilità ad essere incluso nella rete
 - II Invitation - il Provisioner manda un invito al device X, che risponde con le sue informazioni
 - III Scambio Public Key - i due attori si scambiano le chiavi pubbliche
 - IV Autenticazione - avviene in diversi modi anche seconda delle capacità dei dispositivi e della richiesta o meno di intervento umano
 - V Distribuzione dei dati di Provisioning - a partire dalle loro chiavi pubbliche e private viene creata una chiave di sessione, usata per proteggere la successiva distribuzione dei dati richiesti per completare il processo di provisioning (la NetKey e la DevKey, anche se quest'ultima è calcolata indipendentemente e non viene mai scambiata per messaggio). A fine procedimento il nodo è in possesso di NetKey, DevKey, IV Index e dell'indirizzo di Unicast
 - *Remote provisioning:* dove Provisioner e dispositivo unprovisioned non devono necessariamente essere a portata di segnale, ma possono effettuare la procedura di provisioning se esiste un percorso di comunicazione sicura attraverso la rete tra i due dispositivi
 - *Certificate based provisioning:* lo step di autenticazione avviene attraverso l'uso di certificati

- **Key refresh procedure:** è possibile cambiare la chiave della rete su richiesta dell'utente
- **Message obfuscation:** per migliorare la privacy dei dati inviati è necessario prevenire il tracciamento dei pacchetti offuscando gli header delle PDU mediante l'utilizzo della NetKey

2.4.2 Come BLE Mesh soddisfa i requisiti dell' IoT

Analizziamo ora quanto il BLE Mesh sia adatto alla costruzione di una rete IoT, considerando 7 caratteristiche e analizzandone, in ordine, requisiti richiesti, punti di forza e punti di debolezza.

1. **Scalabilità:** l'abilità di un sistema di adattarsi ad un crescente carico di lavoro
 - la rete deve funzionare anche con un alto numero di elementi in comunicazione, sia sulle piccole che sulle grandi distanze
 - utilizzabili fino a circa 32mila nodi per rete, con 127 hops per pacchetto e raggio d'azione esteso grazie alla topologia mesh
 - possibili problemi con metodologia *controlled-flooding*, che può causare uno storm di messaggi broadcast e conseguente perdita di pacchetti su reti estese; il problema è stato mitigato con l'introduzione del *managed-flooding*
2. **Flessibilità:** la capacità di adattare le risorse di rete disponibili ai cambiamenti dei requisiti di progettazione
 - essenziale supportare aggiunte, rimozioni, movimenti di nodi senza dover riconfigurare gli altri nodi
 - BT Mesh permette un aggiornamento dinamico grazie a caratteristiche di *self-form* e *self-heal* (dato dall'assegnamento dinamico di indirizzi di gruppo e virtuali), che migliorano anche la fault-tolerance della rete
 - problematiche sorgono quando un gran numero di acknowledgments devono essere mandati ad un singolo group owner, creando storm di messaggi e possibile perdita di pacchetti; una soluzione suggerita dalla stesso Bluetooth SIG è l'inserimento di delay randomici per l'invio dei messaggi
3. **Robustezza:** il grado in cui un sistema può funzionare correttamente in presenza di input non validi o condizioni ambientali stressanti
 - molti dispositivi comportano molti punti di possibile fallimento, possibilmente anche critici

- per la natura del mesh, il sistema di managed flooding e il re-invio di messaggi mitigano eventuali fallimenti di uno o più nodi
 - il managed-flooding è un metodo che richiede più risorse rispetto ad un classico single-path routing
4. **Reattività:** la velocità di risposta ad una richiesta o evento
- solitamente si considera responsiva un'applicazione con ritardi $< 500ms$
 - BT Mesh è in genere molto rapido nelle trasmissioni ma ciò dipende dal numero di hops e di LPN incontrati nel percorso; è inoltre possibile modificare alcuni parametri riguardanti velocità di ritrasmissione dei pacchetti
 - da considerare trade-off con la robustezza
5. **Funzionalità integrabili**
- per una rete IoT, è importante poter integrare vari tipi di dispositivi (sensori, processori, attuatori,...)
 - i *Models* supportano l'aggiunta di feature personalizzate, senza necessità di ulteriori elementi esterni
 - possibili limitazioni nel payload dei messaggi, mitigate dalla possibilità di usare "puntatori" ad altri pacchetti per creare catene di pacchetti
6. **Sicurezza**
- gli attacchi a dispositivi IoT spesso sono sottovalutati e la sicurezza in questi sistemi non è sempre una priorità
 - come spiegato in 2.4.1 BT Mesh fornisce protezione da diverse minacce, anche grazie all'uso di svariate chiavi che criptano e autenticano i messaggi
 - come per ogni sistema, non è possibile avere una sicurezza completa: è possibile il furto di chiavi tramite hardware exploits, attacchi tramite power analysis, fault attacks, attacchi DoS...
7. **Efficienza energetica**
- i sistemi IoT spesso impiegano dispositivi con fonti di energia limitate, come batterie, e coinvolgono numerosi dispositivi, fattore che potrebbe comportare un elevato consumo energetico

- problema che viene affrontato grazie all'uso degli LPN e tramite la configurazione dei TTL, per evitare che i messaggi vengano inoltrati più volte del necessario
- è importante pesare il numero dei nodi relay nella rete dato che per loro natura sono fortemente energivori

2.4.3 Confronto con altre tecnologie wireless

Abbiamo visto come BLE Mesh risponda efficacemente alle esigenze dell'Internet of Things. Analizziamo ora i risultati di diverse ricerche che hanno confrontato BLE Mesh con altre tecnologie wireless analoghe in questo ambito.

Iniziamo con una rapida panoramica delle tecnologie analizzate, evidenziandone i principali punti di forza e obiettivi. Successivamente, presentiamo una tabella comparativa basata su parametri chiave: raggio d'azione, throughput, consumo energetico, costi ricorrenti e topologie di rete supportate.

- **Wi-Fi:** basato sullo standard IEEE 802.15.4, è forse la tecnologia più conosciuta anche da non esperti del settore. Grazie all'ampia larghezza di banda, è la tecnologia ideale per il trasferimento di dati di grande dimensione; trova uno dei suoi più grandi limiti nell'alto consumo energetico, che non lo rende adatto all'uso su dispositivi alimentati da batterie
- **Zigbee:** anch'esso basato sullo standard IEEE 802.15.4, è adatto per il trasferimento di dati di dimensione ridotta sulle corte distanze
- **Z-Wave:** sfruttando la frequenza 908.42MHz, anziché la blasonata 2.4GHz delle due precedenti, questa tecnologia permette ridotti livelli di interferenza. Pensata per la domotica, supporta un buon numero di dispositivi; avendo problematiche di *single point of failure*, non è adatta a sistemi safety-critical
- **NB-IoT** e **LTE-M:** sono parti vitali dello sviluppo dell'IoT basato su 5G. La prima è adatta a sistemi che richiedono bassa energia e bassa larghezza di banda, mentre LTE-M ha una frequenza maggiore che si traduce in risparmio di tempo; entrambe sono soggette a SPOF data la loro topologia a stella
- **LoRaWAN** e **Sigfox:** wide area network a bassa potenza, comunemente utilizzate in applicazioni di smart city (contatori, parcheggi, ...). Anch'esse con topologia a stella soggetta a SPOF
- **BT Piconet:** rete ad-hoc che collega un gruppo di utenti con dispositivi wireless utilizzando protocolli di tecnologia Bluetooth; composta da due o più dispositivi che occupano lo stesso canale fisico (sincronizzati a un clock comu-

ne), consente a un dispositivo master di interconnettersi con un massimo di sette dispositivi slave attivi. Possono essere inattivi fino a 255 ulteriori dispositivi slave, che il dispositivo master può portare in stato attivo in qualsiasi momento[11]. Data la sua struttura, le dimensioni della rete limitano fortemente la quantità di dati scambiati e lo portano ad essere sfruttabile solo in ambito di Personal Area Network (PAN)

Attributes	Technologies								
	BT Mesh	Wi-Fi	Z-Wave	Zigbee	LTE-M	NB-IoT	Sigfox	LoRaWAN	BT Piconet
Range	100 m-1 km	15 m-100 m	30 m-50 m	30 m-100 m	1 km-10 km	1 km-10 km	3 km-50 km	2 km-20 km	<10 m
Through-put	125 kbps-2 Mbps	54 Mbps-1.3 Gbps	10 kbps-100 kbps	20 kbps-250 kbps	≤1 Mbps	≤200 kbps	≤100 bps	10 kbps-50 kbps	200-2100 kbps
Power consumption	Low	Medium	Low	Low	Medium	Low	Low	Low	Medium
Ongoing cost	One-time	One-time	One-time	One-time	Recurring	Recurring	Recurring	One-time	One-time
Topology	Mesh	Star, Mesh	Mesh	Mesh	Star	Star	Star	Star	Star

Figura 2.4: Confronto di BLE Mesh con altre tecnologie wireless

È necessario notare che la tecnologia BLE Mesh ha anche **punti di debolezza**, in particolare in applicazioni che richiedono un’elevata velocità di trasmissione dati (es. streaming di video ad alta qualità) o applicazioni a latenza ridotta (es. controller e visori per AR/VR/MR). Il SIG si impegna a colmare questo vuoto, in particolare sviluppando soluzioni come *High Data Throughput*, che promette di arrivare a fornire frequenza dati fino a 8 Mbps e *Ultra Low Latency HID*, che mira a raggiungere polling rate fino a 1000Hz [12]

2.5 Il BLE Mesh negli articoli scientifici

Analizziamo ora quanto la tecnologia del Bluetooth Mesh sia stata analizzata e approfondita, ricercando ed esaminando i principali studi scientifici pubblicati. Lo studio bibliografico è stato condotto ricercando su *Google Scholar* [13] query come: “ble mesh”, “ble mesh security”, “ble mesh performance”, “ble mesh applications”.

Focalizzandoci su articoli che trattano **analisi dello stato dell’arte**, è bene citare [14], paper scritto nel 2017 agli albori del BLE Mesh che offre una revisione completa sullo stato dell’arte di queste reti, proponendo una tassonomia delle soluzioni esistenti e analizzandone approcci, vantaggi e limiti, individuandone le principali problematiche; fondamentale per la stesura del capitolo 2.4 è stato [15], paper del 2023 che esplora il protocollo BLE Mesh, evidenziandone le potenzialità per sistemi IoT affidabili e scalabili, offrendone una panoramica sulle direzioni di ricerca attuali, i casi d’uso esistenti e le best practice per integrare il Bluetooth Mesh in scenari IoT. Alcuni paper osservano la questione da un punto di vista degli standard, come

[16], che mira a confrontare diversi standard principali tra cui Bluetooth Mesh e 6BLEMesh, versione basata su indirizzamento IPv6.

Per quanto riguarda articoli sull'**analisi della sicurezza** troviamo [17], che analizza i protocolli di rete BLE Mesh e le relative problematiche di sicurezza, evidenziando vulnerabilità e limiti nelle funzionalità. Esplorando un problema non sempre considerato, [18] fornisce una panoramica sulle funzionalità di sicurezza del Bluetooth Mesh e discute attacchi hardware rilevanti, proponendo contromisure per una implementazione più sicura.

Sono stati anche effettuate alcune **analisi sulle performance**: [19] analizza configurazioni, affidabilità, latenza e vulnerabilità. Attraverso simulazioni e dati reali, evidenzia criticità come la sensibilità alle interferenze e i problemi di scalabilità legati al meccanismo di flooding nei nodi densi e propone soluzioni come la randomizzazione dei parametri temporali per migliorare le performance. Interessante anche l'analisi della performance su reti Mesh con storm di messaggi broadcast effettuata da [20]; l'articolo [21] tratta progettazione e implementazione di una rete Bluetooth Mesh con dieci nodi basati su ESP-32, con esame di valori di throughput e latenza in relazione a diversi carichi di messaggi, evidenziando una significativa riduzione delle prestazioni con l'aumentare dei salti nella rete.

Diversi sono invece quelli che esplorano le **applicazioni pratiche** del BLE Mesh: *smart buildings* e *smart lightning* [22][23], *disaster communication* [24], *smart factory* [25], *smart parking* [26] e *monitoring* con [27], ricerca che trova tra gli autori anche il Professor Trotta, relatore di questa tesi. Il paper esplora l'uso delle soluzioni Bluetooth Mesh per il monitoraggio congiunto di spazi interni e persone: vengono valutate le prestazioni delle Wireless Sensor Networks in condizioni di carichi di traffico variabili e diverse dimensioni della rete; viene inoltre proposta una procedura per la localizzazione di dispositivi mobili indoor, valutandone la precisione.

Altri paper riguardano **analisi algoritmiche**, sia pratiche che teoriche: [28] con riflessioni su flooding e connection oriented networks in BLE Mesh; [29] che punta a creare un flooding bilanciato per il risparmio energetico dei nodi; [30] che propone un algoritmo di routing multipath basato su catena di Markov per risolvere i problemi di instabilità nella trasmissione dei dati in un contesto di sistemi di monitoraggio della salute.

Questo campo, ancora relativamente giovane ma in forte espansione, offre terreno fertile per la nascita di nuova letteratura e future ricerche.

Capitolo 3

Definizione del Framework: Progettazione

Il presente lavoro di tesi si pone un duplice obiettivo: da un lato, sviluppare un servizio per la scoperta della topologia di una rete Bluetooth Mesh in modo tale da *rappresentare visivamente la struttura* attraverso cui i nodi della rete sono interconnessi, e dall'altro implementare un servizio per la richiesta di parametri dei nodi al fine di *valutare le performance di rete*.

3.1 Struttura del framework

L'architettura del sistema, considerata ad alto livello, ha due componenti principali: la dashboard e la rete dei nodi.

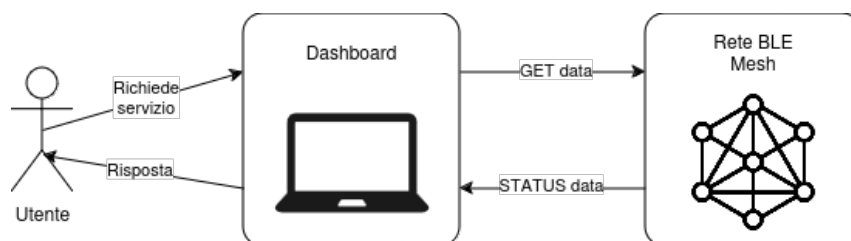


Figura 3.1: Architettura logica del sistema

- **Dashboard:** considerabile l'interfaccia tramite cui l'utente sceglie il tipo di servizio da richiedere e su cui visualizza i risultati testuali dell'esecuzione delle operazioni sulla rete

```

gabe@sansiro: ~/Documents/tesi-BLE-mesh
gabe@sansiro:~/Documents/tesi-BLE-mesh$ python3 dashboard.py

Service Type:
0 - Discover Network Topology - single request
1 - Discover Network Topology - continuous request
2 - Parameter Request
q - Press 'q' to exit dashboard
Select an option (0, 1 or 2): 

```

Figura 3.2: Dashboard

- **Rete di nodi:** passando all'elemento chiave del progetto, abbiamo una rete wireless di n nodi che comunicano tra loro tramite Bluetooth, in particolare sfruttando le potenzialità del BLE Mesh.

Possiamo identificare due tipi di nodi presenti nella rete:

- ◊ 1 nodo *proxy*, che funge anche da *client* della rete, che costituisce il “ponte” tra la dashboard e gli altri dispositivi; la comunicazione con la prima è cablata, mentre con i secondi è wireless
- ◊ $n - 1$ nodi *server*, che all'occorrenza forniscono le informazioni richieste

3.2 Descrizione ad alto livello dei servizi

Un *diagramma dei casi d'uso* è una rappresentazione grafica (spesso scritta in UML, Unified Modeling Language) delle possibili interazioni di un utente (*attore*) con un sistema. È frequentemente usato come strumento di analisi e progettazione software per rappresentare le funzionalità che un sistema può fornire.

Come brevemente accennato in precedenza, e come mostrato dal diagramma 3.3, il sistema fornisce all'utente due principali servizi, descritti più approfonditamente di seguito.

- **discoverTopology**, un servizio di “scoperta” della **topologia di rete**, grazie al quale è possibile vedere quali nodi sono connessi tra loro, capendo quindi anche quali sono i possibili path che un messaggio inviato può seguire per raggiungere una certa destinazione. Per avere una stima sulla **distanza** tra i

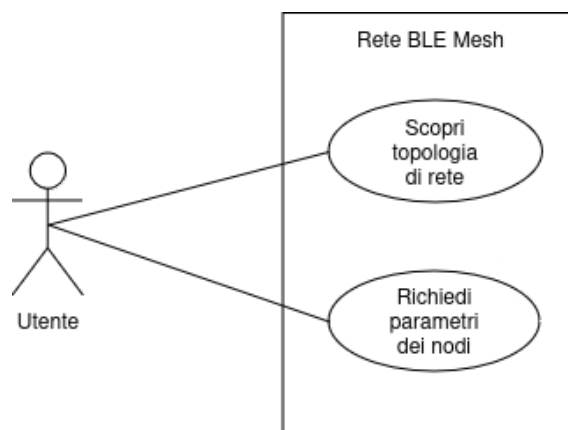


Figura 3.3: Diagramma Use Case

vari nodi, viene anche estratto l'RSSI, un indicatore della qualità del segnale tra due dispositivi; indicato come numero intero negativo, più il suo valore è vicino a 0, più il segnale è ritenuto di qualità.

Oltre ad un output testuale, l'utente può *visualizzare graficamente* la topologia di rete comprensiva dei relativi nodi e archi tra essi. Questo servizio ha due modalità di funzionamento, sempre selezionabili dall'utente:

1. *single request*, con cui è possibile effettuare una singola richiesta, visualizzando lo stato di rete in quel momento
2. *continuous request*, con la quale è possibile fare una richiesta continuata, visualizzando la topologia ad intervalli regolari di tempo; questa modalità è particolarmente utile nel caso si volesse seguire l'evoluzione della rete, assistendo ad aggiunte, rimozioni e spostamento di nodi nello spazio.

Da qui in avanti si considerino alias: *discoverTopology* e *Servizio1*, con le conseguenti sottocategorie *Servizio1.1*, *Servizio1.2*

- **parameterRequest** , che si occupa della **raccolta di** alcuni **parametri** dei nodi della rete, mostrandoli nella dashboard sotto forma di testo una volta che l'operazione è conclusa. Esplicitando i parametri raccolti, abbiamo:
 - Indirizzo unicast del nodo nella rete
 - CID (Company ID)[31]
 - PID (Product ID)
 - NetKey
 - AppKey

- Flag Relay (per indicare se un nodo è un Relay o meno)
- Numero di ritrasmissioni di un messaggio
- Intervallo di ritrasmissione di un messaggio
- TTL
- Potenza di trasmissione, segnata come indice;
ogni indice corrisponde a uno specifico livello di potenza in dBm

È necessario notare come i valori di NetKey e AppKey siano mostrati in chiaro esclusivamente per fini didattici. Data la loro funzione, discussa in 2.4.1, esporre queste chiavi in uno scenario reale sarebbe una grave compromissione della sicurezza dell'intera rete.

Si tenga inoltre a mente che questo servizio rappresenta banalmente solo un'applicazione arbitraria di un più generico servizio per lo scambio di messaggi.

Da qui in avanti si considerino alias: *parameterRequest* e *Servizio2*.

3.3 Funzionalità secondarie

A supporto dei servizi descritti sono state progettate ulteriori funzionalità del sistema, ispirate alle procedure comunemente utilizzate nella *data science*. Queste sono state pensate per garantire la riproducibilità degli esperimenti, per finalità statistiche e per consentire un efficace controllo sui dati e sui processi. Troviamo:

- **file di log** in cui il sistema registra diverse operazioni chiave effettuate da e verso la rete, ciascuna contrassegnata da un timestamp che ne indica il momento di esecuzione
- **file CSV** per il salvataggio in forma tabellare di tutti i dati recuperati in seguito all'esecuzione dei servizi, anch'essi corredati di marca temporale

Capitolo 4

Definizione del Framework: Implementazione

4.1 Tecnologie utilizzate

4.1.1 Hardware

Come anticipato in precedenza, l'architettura è stata realizzata impiegando diversi **ESP32** [32], microcontrollori avanzati sviluppati dall'azienda cinese Espressif Systems, specializzata nella produzione di semiconduttori. Questi dispositivi, dotati di connettività WiFi e Bluetooth, rappresentano una soluzione ideale per applicazioni IoT grazie al loro basso costo, al consumo energetico ridotto e alla possibilità di essere alimentati da fonti diverse, come batterie o energia solare. La loro progettazione robusta e versatile consente la programmazione in linguaggi come C, C++, Python, con il supporto di numerose librerie, anche open source e diversi framework e ambienti di sviluppo, come Visual Studio Code o Arduino IDE, rendendoli quindi estremamente flessibili e adatti a un'ampia gamma di progetti.

Sono stati utilizzati inoltre diversi cavi Micro-USB per l'alimentazione e il trasferimento dei dati da e verso i controllori e varie powerbank indispensabili per simulare gli spostamenti dei nodi nello spazio all'interno della rete.

4.1.2 Software

Diversi sono gli strumenti software usati per la realizzazione del progetto.

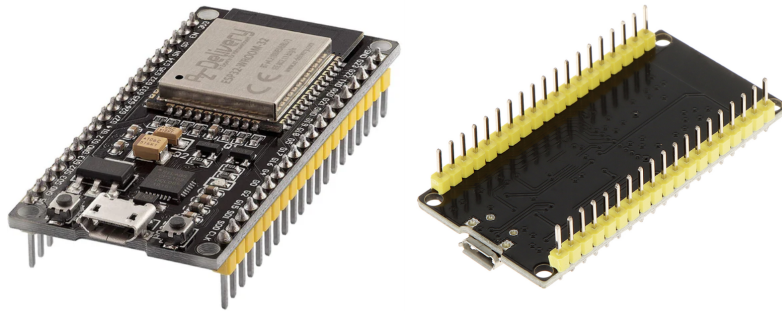


Figura 4.1: Il microcontroller ESP32

- **Framework, librerie e API**, in particolar modo **ESP-IDF** [33][34], il framework ufficiale di Espressif per lo sviluppo su ESP32. L'*Espressif IoT Development Framework* offre strumenti completi per la creazione di applicazioni ESP32 su sistemi operativi Windows, Linux e macOS, fornendo toolchain, API, componenti e flussi di lavoro completi. Espressif offre inoltre l'*ESP-IDF Visual Studio Code Extension* [35], un ambiente di sviluppo integrato (IDE) che rende l'esperienza di programmazione intuitiva e confortevole, grazie a strumenti avanzati per il debug e la gestione dei progetti.
- **C**, linguaggio di programmazione general-purpose usato per la programmazione degli ESP32. Grazie alla sua malleabilità e alle prestazioni elevate, è ampiamente utilizzato nello sviluppo di **codice per microcontroller**. Efficienza, portabilità, ampio supporto e controllo diretto dell'hardware lo rendono un linguaggio estremamente adatto per gestire le periferiche IoT
- **Python**, linguaggio di programmazione versatile e intuitivo, particolarmente apprezzato per la sua semplicità sintattica e la vasta gamma di librerie disponibili. Grazie a queste caratteristiche, è una scelta eccellente per lo **sviluppo della dashboard**, integrando in modo efficace l'elaborazione dei dati, la visualizzazione e il controllo in un'unica applicazione user-friendly

Nell'implementazione del progetto non è stata utilizzata un'app dedicata al provisioning dei dispositivi (come ad esempio *nRF Mesh* [36]): questo incarico è affidato al nodo proxy/client, che avvia la procedura di provisioning non appena rileva a portata di segnale un dispositivo che chiede di entrare nella rete. Si è optato per questa scelta per semplificare e velocizzare il processo di sviluppo, sottolineando però che una simile implementazione non è consigliabile in scenari reali, poiché espone la rete a potenziali rischi di sicurezza, come l'accesso non autorizzato da parte di nodi malevoli.

4.2 Architettura della rete

Presentiamo nella figura 4.2 l'architettura pensata per lo sviluppo del progetto, approfondendo quanto accennato in figura 3.1.

La **dashboard** comunica con la rete BLE Mesh tramite il nodo proxy/client, che è anche il provisioner della rete. In particolare la connessione è di tipo cablato, utilizzando una connessione seriale con l'ESP32 tramite un ponte USB-to-UART, accompagnato dai driver necessari per garantirne il funzionamento. [37]

La dashboard è anche un'unità di "assemblaggio" delle informazioni: dopo averle richieste alla rete, arrivano ad essa sotto forma di struttura di cui deve fare unpack e analisi, riorganizzandone i campi e producendo l'output (grafico o meno) desiderato.

La **rete** utilizzata per lo sviluppo è composta da **tre nodi**: un nodo *proxy/client* e due nodi *server*, connessi tramite Bluetooth in modalità wireless.

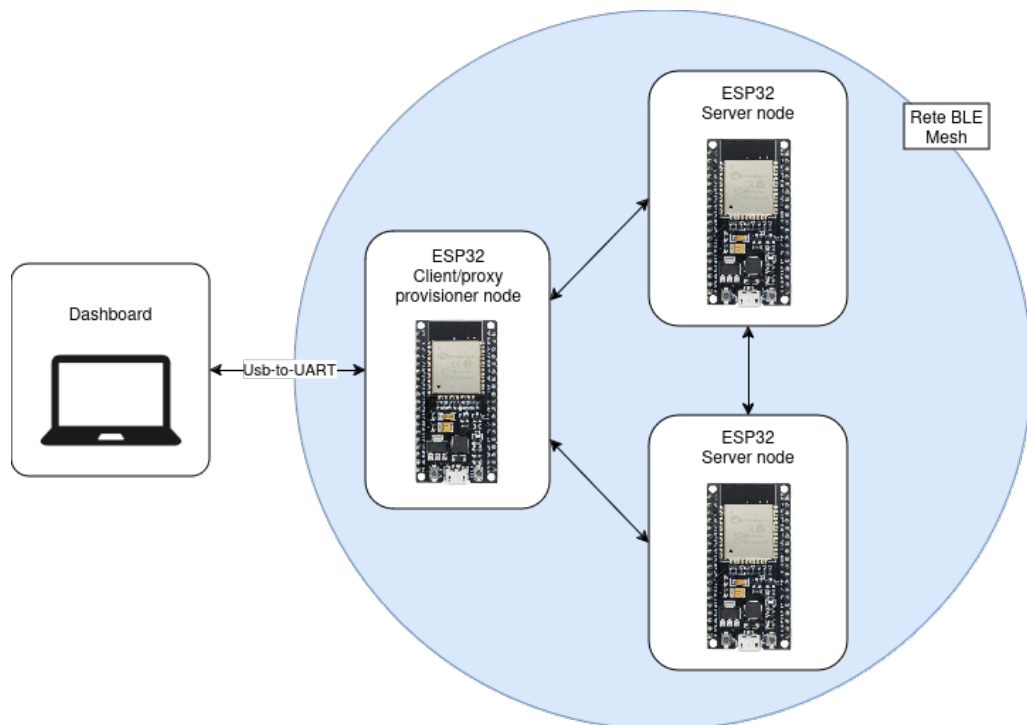


Figura 4.2: Architettura del sistema nel dettaglio

Il codice prodotto rappresenta una evoluzione dei modelli *ESP BLE Mesh Vendor Client Model* [38] e *ESP BLE Mesh Vendor Server Model* [39] forniti come esempio da Espressif sulla repository Github.

4.3 Servizi

Spieghiamo ora più nel dettaglio il funzionamento e le scelte implementative adottate nello sviluppo dei due servizi principali offerti dal sistema.

4.3.1 *discoverTopology* Service

Sono stati pensati due tipi diversi di implementazione.

1. Meccanismo basato su Heartbeat Publishing (figura 4.3)

- ogni X secondi, un nodo manda in broadcast ai nodi a distanza di 1 hop (nodi di immediata prossimità) un segnale di presenza (Heartbeat). Il sistema deve essere pensato in modo tale da sfalsare temporalmente il più possibile i momenti di invio degli Heartbeat per ridurre la congestione della rete ed evitare la perdita di pacchetti
- ogni nodo ha una tabella con N righe, ognuna delle quali contiene l'indirizzo del nodo da cui ha ricevuto l'Heartbeat, un timestamp che segni il momento di ricezione e l'RSSI per stimare la distanza dei due nodi
- ogni nodo aggiorna l'entry identificata dall'indirizzo del nodo vicino, garantendo che il timestamp e l'RSSI si riferiscano all'ultima interazione avvenuta (ovvero l'ultimo messaggio ricevuto)
- per ogni nodo, verifico periodicamente che i timestamp delle entries siano più recenti di un intervallo di tempo Y . Se una entry risulta più vecchia, viene rimossa dalla tabella, poiché il nodo non è più considerato un vicino.
- per visualizzare la topologia di rete, l'utente, tramite la dashboard, richiede con un broadcast tutte le tabelle dei nodi della rete, che vengono opportunamente analizzate e riorganizzate nella dashboard stessa

2. Meccanismo di messaggistica On-Demand

- su richiesta della dashboard, il client manda un messaggio broadcast "trigger" (nel codice identificato da `OPCODE TOPOLOGY_TRIGGER`) di tipo acknowledged. Contestualmente invia un messaggio IAM in broadcast a tutti i nodi a 1 hop di distanza per segnalare ai suoi vicini la sua presenza
- similmente i nodi che ricevono il trigger inviano a loro volta un IAM ai nodi a 1 hop di distanza. Alla prima ricezione di un messaggio IAM, nel nodo parte un timer di X secondi

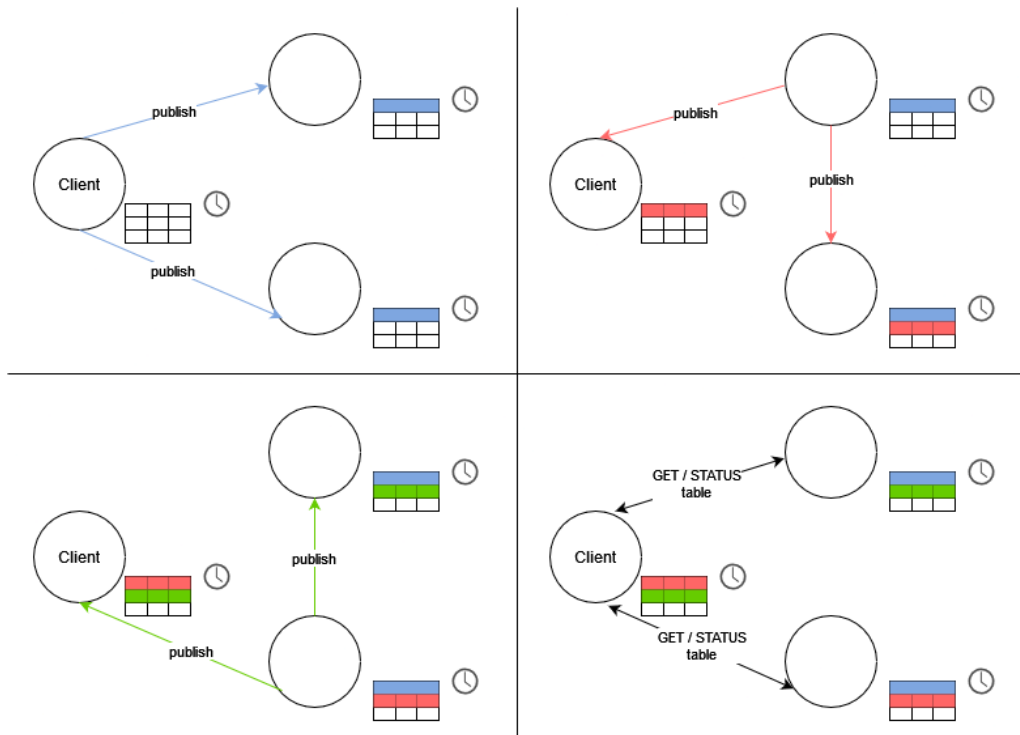


Figura 4.3: Schema grafico di Implementazione 1 - *Heartbeat Publishing*

- le informazioni ricevute vengono salvate in un array `iam_messages` composto da strutture `iam_data_t`; viene arbitrariamente fissata la dimensione dell'array a 20, ovvero vengono salvati al più dati di 20 nodi adiacenti (misura più che sufficiente ai fini didattici dell'esperimento). Ogni struttura ha come campi `src_addr` (indirizzo del nodo da cui ho ricevuto il messaggio) e `rsi` (per stimarne la distanza)
- alla scadenza del timer, viene mandato al client la risposta `TOPOLOGY_RESPONSE` contenente l'array con le informazioni raccolte fino a quel momento
- il client rielabora questi N array provenienti da N nodi, contando il numero di elementi per ogni array (`num_nearby_nodes`). Crea N strutture `topology_response_t` con campi: `src_addr` (l'indirizzo del nodo da cui proviene l'array), `num_nearby_nodes` e `nearby_nodes` (array di tipo `iam_data_t` contenente i nodi in prossimità del nodo mittente)

```

1 typedef struct {
2     uint16_t src_addr;
3     int8_t rssi;
4 } __attribute__((packed)) iam_data_t;
5
6 typedef struct {
7     uint16_t src_addr;
8     uint8_t num_nearby_nodes;
9     iam_data_t nearby_nodes[MAX_MESSAGES];
10 } __attribute__((packed)) topology_response_t;
11

```

Listing 4.1: Le strutture usate nell’implementazione On-Demand; Per facilitare il parsing della struttura rimuovendo l’eventuale padding inserito automaticamente tra i campi si usa `__attribute__((packed))`

3. Meccanismo di messaggistica On-Demand con invio randomizzato delle risposte (figure 4.4 e 4.5)

- considerabile un’evoluzione del metodo 2, ha in aggiunta un delay temporale per ridurre le collisioni tra pacchetti e quindi evitare di congestionare la rete
- una volta ricevuta la `TOPOLOGY_TRIGGER`, prima di chiamare la funzione per l’invio del messaggio `IAM`, il nodo applica un ritardo casuale compreso tra 1 e 5 secondi, con granularità al millesimo

La *prima implementazione* sfrutta meglio le funzionalità aggiunte di BLE Mesh, con lo svantaggio di avere nella rete un costante traffico base di messaggi mandati in automatico. La *seconda implementazione* si basa su un classico scambio di messaggi modificato, lasciando libera la rete quando non è richiesto il servizio. Ha come svantaggio uno scambio di un numero di messaggi superiore per effettuare la stessa operazione, con un rischio di storm di messaggi e perdita di pacchetti nel caso di reti con una quantità consistente di nodi. Questa problematica è infatti stata riscontrata durante l’esecuzione degli esperimenti pratici di cui si parlerà nel capitolo 5, ed è quindi stato necessario porvi rimedio: la **soluzione 3** è quella attualmente **adottata**, e non ha mostrato particolari problemi di congestione della rete durante gli esperimenti.

È stato condotto anche un breve studio sulla **complessità computazionale** in termini di quantità di messaggi generati, da cui emerge che i diversi metodi presentano lo stesso ordine di complessità.

Considerando n il numero di nodi della rete:

1. ogni nodo manda messaggi per segnalare la propria presenza, al più a tutti i

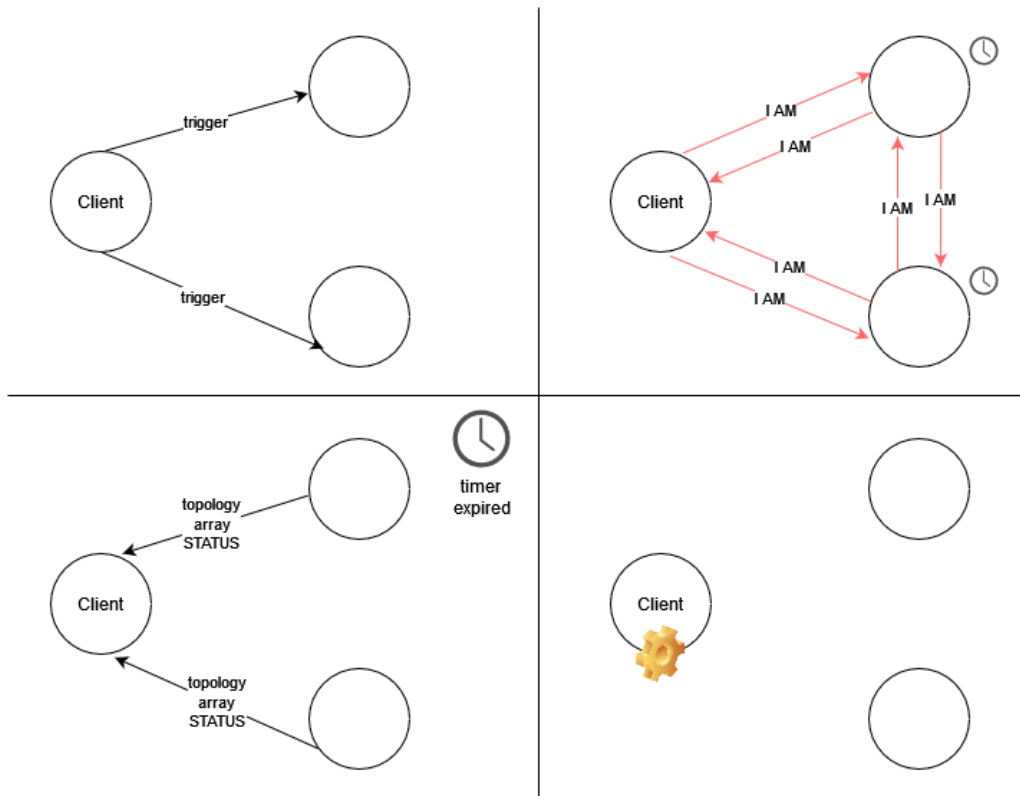


Figura 4.4: Schema grafico di Implementazione 3 - *Messaggistica On-Demand*

nodi tranne se stesso, per un numero arbitrario di volte (gli heartbeat mandati a intervalli regolari), a cui si aggiunge la richiesta delle tabelle da parte della dashboard, a cui consegue la risposta. Il processo genera al più una quantità polinomiale di messaggi, in particolare una quantità quadratica:

$$c \times (n \times (n - 1)) + 2n = c \times n^2 + 2n \rightarrow O(n^2)$$

- ogni nodo manda **IAM** al più a tutti i nodi tranne se stesso, il sistema manda il trigger iniziale per ogni nodo (da cui aspetterà risposta). Il processo genera al più una quantità polinomiale di messaggi, in particolare quadratica:

$$n + (n \times (n - 1)) + n = n^2 + 2n \rightarrow O(n^2)$$

- l'evoluzione del metodo 3 non influisce sulla complessità

Si segnala inoltre che i metodi 2 e 3 rappresentano un'evoluzione rispetto ad una prima implementazione rudimentale. In questa, anziché attendere l'arrivo di una serie di messaggi IAM per poi inviare al client un unico messaggio contenente i dati elaborati, ogni singolo messaggio IAM ricevuto veniva immediatamente inoltrato al client. Questo approccio generava un traffico di rete aggiuntivo, comportando un costo computazionale cubico anziché quadratico.

$$(n \times (n - 1)) \times n + n = n^3 + n \rightarrow O(n^3)$$

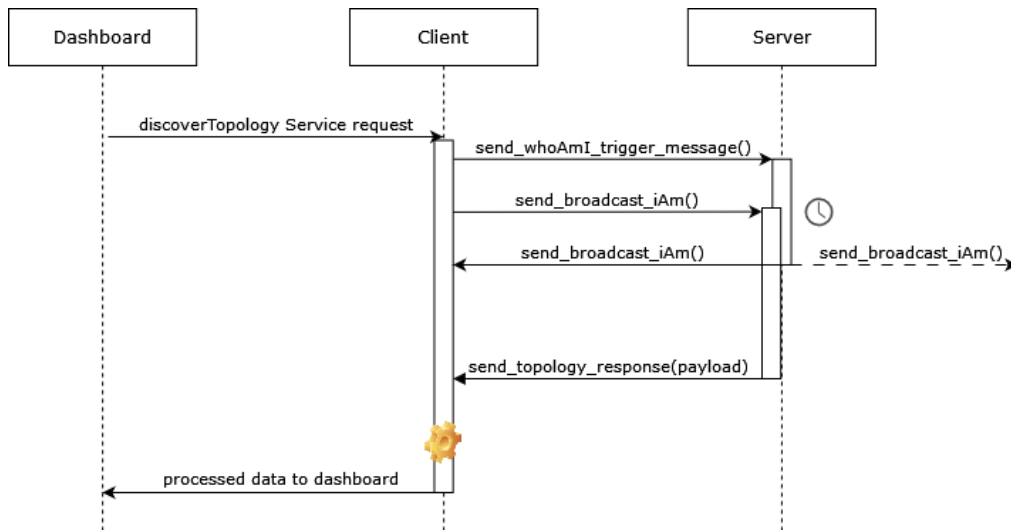


Figura 4.5: Diagramma di sequenza UML di Implementazione 3 - Messaggistica On-Demand

4.3.2 *parameterRequest* Service

Il servizio `parameterRequest` rappresenta una semplice implementazione pratica di un banale scambio di messaggi di tipo *acknowledged*. È utilizzato per estrarre alcune **informazioni dai nodi** della rete ed è usato per raccogliere dati a fini statistici sulle **performance di rete** (approfondimento al capitolo 5).

La sua implementazione, di cui riportiamo a grandi linee il funzionamento, non ha subito particolari modifiche nel corso del progetto.

- in seguito alla richiesta dell'utente per mezzo della dashboard, il client manda un broadcast con la richiesta dei parametri. Questo messaggio è di tipo *ack*, per garantire le risposte di ogni nodo nella rete
- contestualmente al broadcast, il client estrae i suoi parametri e crea la struttura `ble_parameters_t`, immediatamente inoltrata alla dashboard

- ogni nodo server estrae i parametri, popola `ble_parameters_t` e la manda in risposta al client, che provvederà ad inoltrarla alla dashboard

Presentiamo la struttura atta a contenere i parametri, mostrando i tipi di ogni campo:

```

1 typedef struct {
2     uint16_t cid;
3     uint16_t pid;
4     uint16_t this_node_addr;
5     uint8_t net_key;
6     uint8_t app_key;
7     uint8_t net_transmit;
8     bool relay_on;
9     int ttl;
10    esp_power_level_t tx_power;
11 }ble_parameters_t;
12

```

La classe di complessità del servizio è lineare:

$$n + n = 2n \rightarrow O(n)$$

4.4 Dashboard

La **Dashboard** è l'elemento della rete che si configura come **intermediario tra l'utente e la rete** stessa. Il programma è scritto con Python, e può essere tranquillamente lanciato da shell. Python è un linguaggio di programmazione estremamente versatile, ideale per affrontare una vasta gamma di task, anche molto specifici. La sua forza risiede nella ricca disponibilità di librerie e pacchetti, che estendono le sue funzionalità e permettono di adattarlo facilmente a contesti diversi e altamente specializzati.

Di seguito sono presentate le librerie utilizzate nello script.

- *serial*: permette la comunicazione con dispositivi tramite porte seriali (verso il proxy)
- *struct*: consente di gestire dati binari e formattarli in strutture definite
- *time*: utilizzato per operazioni legate alla gestione temporale (usato per timer)
- *os*: facilita l'interazione con il file system
- *datetime*: fornisce strumenti per lavorare con date e orari

- *networkx*: utilizzato per creare, analizzare e visualizzare grafi e reti complesse
- *matplotlib.pyplot*: consente di creare grafici e visualizzazioni
- *logging*: permette di registrare eventi e diagnosticare errori nel programma (creazione del file di log)
- *pandas*: fondamentale per la gestione e l'analisi di dati strutturati (creazione dei file CSV)
- *random*: serve per generare numeri casuali (creazione id)

4.4.1 Funzionamento della Dashboard

Illustriamo in modo discorsivo i passi principali del funzionamento della Dashboard.

1. L'utente, tramite input da tastiera, può selezionare il servizio desiderato o scegliere di uscire dalla Dashboard, terminando l'esecuzione del programma:

```
0 - Discover Network Topology - single request
1 - Discover Network Topology - continuous request
2 - Parameter Request
q - Press 'q' to exit dashboard
```

Sia 0 che 1 chiamano `discoverTopology_Service`, con la differenza di un flag `continuous` che indica se la richiesta è singola oppure continuata nel tempo fino a interruzione da parte dell'utente.

L'opzione 2 chiama `parameterRequest_Service`.

2. La relativa funzione viene chiamata; viene aperta la connessione seriale verso il nodo proxy secondo i parametri impostati (*baud rate* e *numero di porta seriale*)
3. Invia al proxy un byte (*0, 1 o 2*) che identifica il servizio richiesto dall'utente, consentendo al nodo di eseguire la funzione appropriata corrispondente. Nel proxy, un ciclo continuo attende questo byte e lo confronta con delle costanti definite a priori.

```
while (true) {
    uart_read_commands();
    vTaskDelay(pdMS_TO_TICKS(1000));
};
```

4. La funzione acquisisce le risposte inoltrate dal client per un massimo di X secondi, trascorsi i quali interrompe la lettura
5. Se riceve dei bit dal collegamento USB e la dimensione dei dati ricevuti corrisponde a quella attesa per quel tipo di dato, procede con l'*unpacking* della struttura, estraendo i campi e salvandoli in una struttura di tipo dizionario.
6. I dati sono mostrati come output testuale sulla shell e sono salvati in file CSV, come si analizzerà più nel dettaglio in sezione 4.5

6.1 SOLO PER *discoverTopology*

Viene chiamata la funzione `draw_topology_graph`, che ha il compito di generare un output grafico della rete, elaborando il dizionario dei dati salvati e visualizzando nodi e archi pesati con l'RSSI.

6.2 SOLO PER *discoverTopology*

Il grafico tiene conto della distanza tra i nodi, disegnando archi più lunghi se due nodi hanno un RSSI più basso (o un valore assoluto dell'RSSI più alto, dato che è un valore negativo): in questo modo i nodi con una comunicazione peggiore appaiono graficamente più distanti, suggerendo una maggiore distanza fisica tra di essi. Va sottolineato che questa relazione non è sempre valida, in quanto altre condizioni potrebbero influenzare negativamente il segnale (ad esempio congestione della frequenza radio, ostacoli, maggiore umidità ambientale...). Una comunicazione migliore è rappresentata da archi verdi e di maggiore spessore, mentre una connessione scarsa è mostrata con archi rossi più sottili. Sono state definite tre soglie per classificare la qualità del segnale in base all' RSSI:

Livello RSSI	Valutazione qualità di connessione
$\text{RSSI} > -50$	ottima
$-80 < \text{RSSI} \leq -50$	media
$\text{RSSI} \leq -80$	scarsa

Tabella 4.1: Valutazione del segnale RSSI

7. Ritorno al mainMenu della Dashboard

4.5 Logging e salvataggio dati

Durante l'esecuzione del codice della dashboard, vengono raccolte informazioni sullo stato di avanzamento dei processi e vengono salvati i dati restituiti dall'esecuzione dei servizi. Queste informazioni vengono gestite attraverso funzioni dedicate al logging e al salvataggio dei dati.

```

gabe@sansiro: ~/Documents/tesi-BLE-mesh
Service Type:
0 - Discover Network Topology - single request
1 - Discover Network Topology - continuous request
2 - Parameter Request
q - Press 'q' to exit dashboard
Select an option (0, 1 or 2): 0

Connected to /dev/ttyUSB0 at 115200 baud.
Press Ctrl-C to abort

Starting topology discovery.

Node 0x0006:
- Neighbour 0x0001, RSSI: -61
- Neighbour 0x0005, RSSI: -43

Node 0x0005:
- Neighbour 0x0001, RSSI: -60
- Neighbour 0x0006, RSSI: -45

gabe@sansiro: ~/Documents/tesi-BLE-mesh
Node 0x0005 Parameters:
  this_node_addr: 0x0005
  cid: 0x02e5
  pid: 0x0000
  net_key: 0x00f8
  app_key: 0x0056
  net_retransmit_count: 2
  net_retransmit_interval_ms: 20
  relay_on: True
  ttl: 7
  tx_def_power: 7

Node 0x0006 Parameters:
  this_node_addr: 0x0006
  cid: 0x02e5
  pid: 0x0000
  net_key: 0x00f8
  app_key: 0x0056
  net_retransmit_count: 2
  net_retransmit_interval_ms: 20
  relay_on: True
  ttl: 7
  tx_def_power: 7

```

(a) *discoverTopology*

(b) *parameterRequest*

Figura 4.6: Alcuni esempi di output testuali su shell in seguito alla chiamata dei Servizi

Il *logging* è essenziale per monitorare il corretto funzionamento del sistema, individuare eventuali errori o anomalie e garantire la tracciabilità delle operazioni. Il *salvataggio dei dati* permette invece di archiviare i risultati per analisi successive, confronti o integrazioni con altre fonti. Nella data science, queste pratiche sono fondamentali per assicurare la *replicabilità degli esperimenti* e facilitare l'*interpretazione dei risultati*.

4.5.1 Logging

Implementato in Python grazie al modulo `logging`, è stato pensato per avere una **timeline** di tutte le operazioni chiave avvenute durante l'esecuzione del codice.

Un *log* è un file testuale costituito da una serie di righe, ognuna per la registrazione di un dato evento. Ciascuna riga ha una struttura precisa:

- un **timestamp**, composto da data e ora nel formato *YYYY-MM-DD HH:MM:SS*
- un **livello di logging**, che indica a quale “categoria” (*debug, info, warning, error, critical*) appartiene l'evento registrato
- il nome della **funzione** in cui si verifica l'evento
- la **descrizione dell'evento**

Diverse sono le operazioni di cui si è scelto di fare logging, in particolare:

- inizio e fine delle operazioni della dashboard, corrispondente ad inizio e fine del programma Python

- selezione del servizio, con relativo indice, e uscita dal servizio al termine delle operazioni
- ricezioni di pacchetti di dati dalla rete, con relativo nodo di invio
- salvataggio dei dati nei file CSV, con *id* di riferimento (vedi in sezione 4.5.2)
- varie informazioni aggiuntive, come ad esempio inizio e fine di timer, invio richiesta parametri, nessun dato ricevuto dai nodi, plotting eseguito...
- segnalazione di eccezioni catturate o terminazioni di programma non previste

È possibile trovare un esempio di file di logging in appendice A.2.

4.5.2 Salvataggio dati

I dati delle sperimentazioni possono essere salvati in diversi formati, ciascuno con i propri vantaggi. Formati come XML o JSON sono particolarmente utili per gestire dati molto strutturati, grazie alla loro capacità di rappresentare relazioni complesse in modo gerarchico e leggibile. Tuttavia, CSV offre il vantaggio di essere semplice, leggibile da qualsiasi editor di testo e facilmente integrabile con strumenti comuni di analisi e visualizzazione dati (ad esempio posso visualizzare in modo semplice i dati in forma tabellare in software per fogli di calcolo).

Per questo progetto è stato scelto il **formato CSV**, al fine di rimanere il più vicino possibile alle pratiche della data science, dove la semplicità e la portabilità dei dati sono spesso preferite per la condivisione e l'analisi veloce.

Un file CSV (*Comma-Separated Values*) è un formato di testo semplice utilizzato per rappresentare dati strutturati. Ogni riga del file corrisponde a un record, e i campi all'interno di ciascun record sono separati da virgole. È comune includere un'intestazione all'inizio del file, che specifica i nomi dei campi per ogni colonna. Questo rende i dati facilmente comprensibili e consente di identificarne rapidamente il contenuto.

Per i due servizi offerti dalla rete, sono stati pensati due diversi file in cui salvare i dati, *discoverTopology.csv* e *parameterRequest.csv*, ciascuno con una propria struttura.

Per entrambi i servizi vengono generati gli **identificatori** *id* e *iterationId*, numeri interi scelti casualmente all'interno di un intervallo compreso tra 0 e 999. Questi indici, presenti sia nei dati salvati nei file CSV che nei corrispondenti record del file di log, facilitano l'associazione tra i due insiemi di informazioni.

I campi di **discoverTopology.csv** sono i dati estratti dal servizio con aggiunta di qualche metadato:

- **id**, che identifica tutti i dati raccolti in una singola iterazione del servizio
- **iterationId**, che identifica tutti i dati raccolti dal servizio una volta che è stato lanciato dalla dashboard. Questo significa che se viene scelto *discoverTopology* nella sua modalità *continuous*, i dati raccolti durante un ciclo di esecuzione del servizio avranno stesso *id*, dati in cicli di esecuzione differenti avranno diverso *id*, ma stesso *iterationId*
- **continuous**, booleano che indica se è stata scelta la ricerca continuativa o meno
- **timestamp**, marca temporale nel formato *YYYY-MM-DD HH:MM:SS*
- **service**, il nome della funzione che estrae i dati dalla rete
- **node**, nodo da cui ricevo il pacchetto di dati
- **num_nearby_nodes**, numero di nodi vicini
- **near_node_address**, indirizzo di uno dei nodi vicini
- **rssis**, potenza di segnale

Ogni record corrisponde ad un singolo arco del grafo, quindi se un nodo K ha ad esempio 2 nodi vicini, le entries con $node=K$ saranno 2, aventi lo stesso **id**, con differente **near_node_address**.

È possibile trovare un esempio di *discoverTopology.csv* in appendice A.3.

I campi di **parameterRequest.csv** sono i parametri estratti dai nodi, anche in questo caso con aggiunta di qualche metadato:

- **id**, similmente a prima, identifica tutti i dati raccolti in una singola iterazione del servizio.
- **timestamp**, marca temporale nel formato *YYYY-MM-DD HH:MM:SS*
- **service**
- **this_node_addr**
- **cid**
- **pid**
- **net_key**

- `app_key`
- `net_retransmit_count`
- `net_retransmit_interval_ms`
- `relay_on`
- `ttnl`
- `tx_def_power`

È possibile trovare un esempio di *parameterRequest.csv* in appendice B.3.

4.5.3 Salvataggio immagini della topologia di rete

È possibile salvare immagini (in formato .png) rappresentanti i plot della rete generati a partire dalle informazioni ottenute tramite il servizio `discoverTopology`.

Il salvataggio avviene all'interno della funzione `draw_topology_graph`, utilizzando il modulo *matplotlib.pyplot* di Python. I file vengono archiviati in una directory dedicata chiamata `network_plots`.

Il nome di ciascun file segue un pattern specifico, ovvero *YYYYMMDD_HHMMSS*, per facilitare il recupero delle immagini leggendo logfile e dati salvati nel CSV.

Capitolo 5

Validazione

5.1 Setup sperimentale

Questo capitolo è dedicato alla presentazione di metodi adottati e risultati ottenuti dal sistema durante i test effettuati su una configurazione della rete più ampia rispetto a quella utilizzata durante la fase di sviluppo.

La **struttura della rete** ha un'idea simile a quella raffigurata in 4.2): è composta da una Dashboard, che funge da interfaccia tra l'utente e la rete stessa, quest'ultima costituita da un nodo client/proxy con connessione cablata e da altri nodi che comunicano tramite connessione Bluetooth. Per la sperimentazione sono stati utilizzati diversi ESP32, alimentati sia tramite corrente a parete (*nodi statici*) che tramite powerbank (*nodi dinamici*) in modo tale da poter osservare il comportamento della rete in condizioni di mobilità dei dispositivi all'interno dello spazio.

L'esperimento è stato condotto in un **ambiente reale** costituito da un insieme di uffici, rappresentati dalla planimetria di figura 5.1 (il nodo rosso corrisponde al client/proxy, la cui posizione non è mutata nel corso degli esperimenti).

La configurazione della rete e la sua disposizione nello spazio sono state progettate per simulare un possibile *scenario d'uso reale*, come un deploy in ufficio, in un'aula universitaria o in una sala conferenze. Questa scelta mira a garantire una maggiore autenticità dei dati raccolti e a evidenziare eventuali criticità che potrebbero emergere durante l'utilizzo del sistema.

Specifichiamo che tutti i nodi della rete sono stati usati con la seguente **configurazione**:

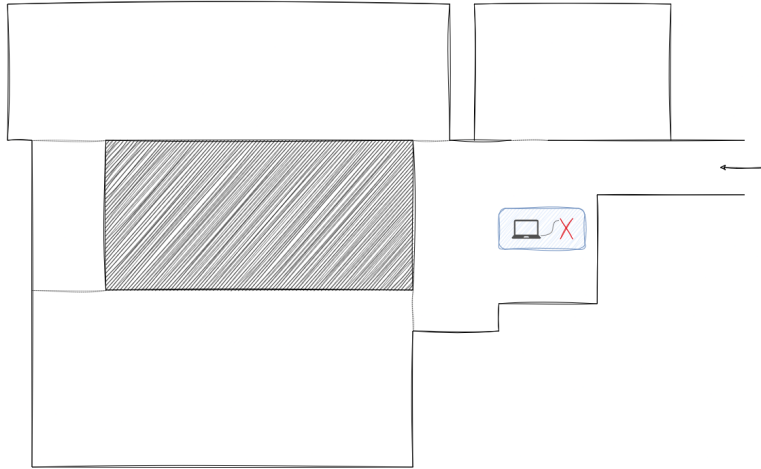


Figura 5.1: Spazio fisico di deploy della rete

- *potenza radio di trasmissione* = 7, ovvero ESP_PWR_LVL_P9, corrispondente a +9 dBm
- la trasmissione di un messaggio avviene 3 volte
- il *TTL* di default dei messaggi è impostato a 7

5.2 Esperimenti

Per ogni esperimento è stato tenuto un log manuale sulle operazioni “fisiche” intervenute sulla rete (aggiunta, spostamento, rimozione nodi, richiesta dei servizi...)

5.2.1 Esperimento 1 - *discoverTopology*

Scopo

Validazione del servizio `discoverTopology` per la *scoperta della topologia di rete* e analisi della sua risposta all’aggiunta, spostamento e rimozione di nodi.

Selezione nella Dashboard

1 - Discover Network Topology - continuous request

Numero di nodi della rete

9 nodi (1 proxy/client + 8 server)

Configurazione di rete

Eccezion fatta per il nodo proxy (indirizzo 0x0001), sempre presente per configurazione della rete, si è proceduto all'*aggiunta graduale dei dispositivi* alla rete. Dopo essere stati provisionati dal client, questi dispositivi diventano visibili come nodi. L'analisi della topologia è stata effettuata anche *spostando* alcuni *nodi dinamici* nello spazio, per osservare i cambiamenti negli archi della rete e le variazioni nella qualità del segnale sui collegamenti interessati. Nella fase finale dell'esperimento, si è proceduto con la *rimozione graduale* dei nodi, fino a ridurre nuovamente la rete alla sola presenza del nodo proxy.

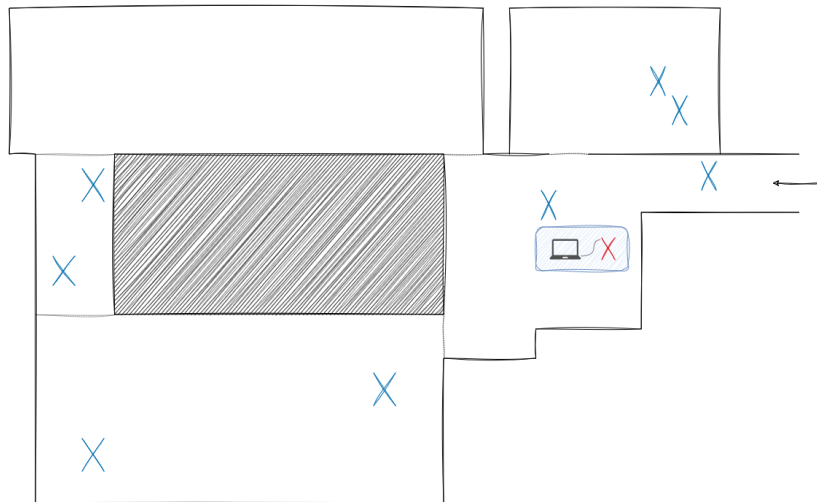


Figura 5.2: Posizionamento finale dei nodi di *Esperimento1*

Tabella del log manuale

Disponibile in appendice A.1

Log di sistema

Disponibile in appendice A.2

Dati raccolti nel file *discoverTopology.csv*

Disponibile in appendice A.3

Analisi dei risultati dell'esperimento

- Aggiunta dei primi nodi avvenuta regolarmente
- Notiamo che l'aggiunta di nodi più lontani (0x0009) dall'origine (identificata come 0x0001) si evidenzia con un RSSI basso verso l'origine e molto basso verso i nodi situati fisicamente sul lato opposto rispetto all'origine (0x0005 e 0x0006).
- Notiamo evoluzione del *nodo dinamico* 0x000c
 - 12:27:37 - 5 nodi vicini
 - 12:27:58 - 6 nodi vicini
 - 12:28:40 - 4 nodi vicini, esso perde infatti collegamento da 0x0005 e 0x0006 in seguito all'allontanamento da essi
- Oltre alla rimozione dei nodi 0x0005 e 0x0009, si osserva l'effetto del riorientamento dell'antenna del nodo 0x0001, che non è più diretta verso il cluster centrale della rete, ma verso l'esterno
 - 12:31:51 - RSSI da/verso 0x0001 medio
 - 12:32:12 - la connessione da/verso 0x0001 cala di qualità
- Riaggiunto regolarmente nodo 0x0009
- Notiamo evoluzione del *nodo dinamico* 0x000d in figura 5.3 e 5.4
 - 12:36:05 - 4 nodi vicini
 - 12:36:26 - 1 nodo vicino
 - 12:36:47 - 2 nodi vicini
- Si procede successivamente con la rimozione graduale dei nodi, evidenziata dal progressivo calo del numero di pacchetti di dati ricevuti dal proxy.

Commenti

I risultati dell'esperimento non si sono discostati dalle previsioni iniziali, confermando la corretta rilevazione dello spostamento dei nodi all'interno della rete. La rete ha dimostrato di scalare sufficientemente bene con l'aumentare del numero di nodi,

garantendo una buona stabilità complessiva. Nonostante si siano verificati alcuni episodi di perdita di pacchetti, possibilità prevista e non impattante sul risultato finale, il sistema ha ripristinato autonomamente la comunicazione all'iterazione successiva del servizio. Questi risultati confermano l'affidabilità e la robustezza del sistema implementato.

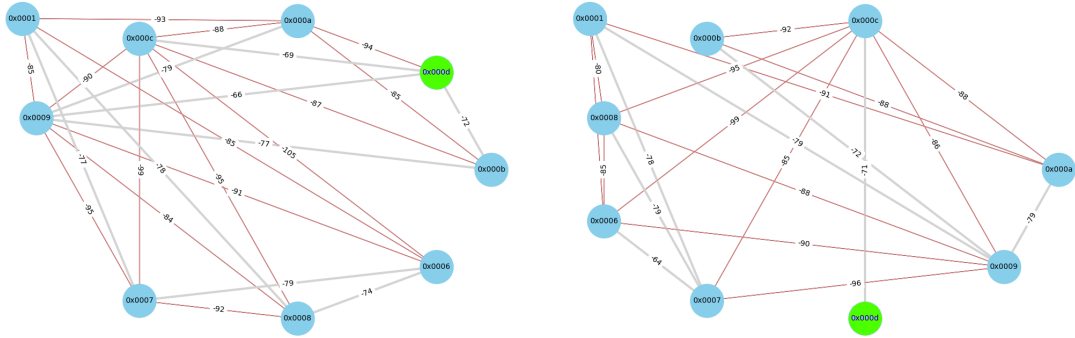


Figura 5.3: Spostamento di 0x000d riflesso sul grafico

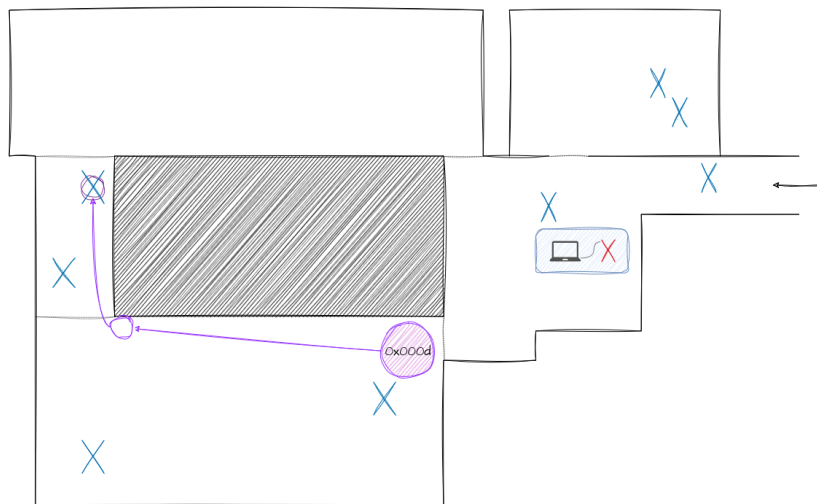


Figura 5.4: Spostamento di 0x000d mostrato sulla planimetria

5.2.2 Esperimento 2 - *parameterRequest*

Scopo

Validazione del servizio `parameterRequest` come servizio di messaggistica a fini di *analisi delle performance di rete*. In particolar modo si vuole verificare il tempo

medio di risposta dei nodi e l'eventuale perdita di pacchetti tramite PDR (Packet Delivery Ratio).

Selezione nella Dashboard

2 - Parameter Request

Numero di nodi della rete

11 nodi (1 proxy/client + 10 server)

Configurazione di rete

L'esperimento è stato condotto *aggiungendo progressivamente due nodi server* alla rete in ciascuna iterazione; per ogni aggiunta sono stati effettuati *tre esperimenti* (corrispondenti a tre chiamate al servizio). Questo procedimento è stato ripetuto cinque volte, portando a un totale di dieci nodi server aggiunti (per un totale complessivo di undici nodi, includendo il proxy). I nodi sono stati distribuiti arbitrariamente nello spazio come da figura 5.5.

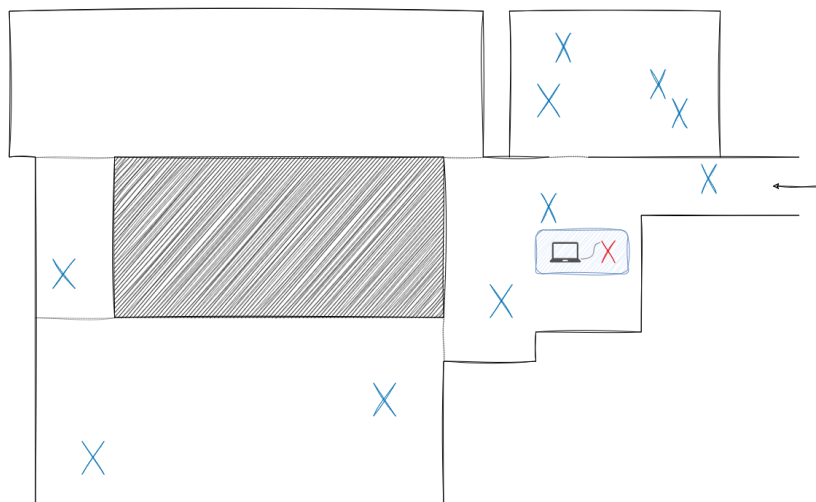


Figura 5.5: Posizionamento dei nodi di *Esperimento 2*

Tabella del log manuale

Disponibile in appendice B.1

Log di sistema

Disponibile in appendice B.2

Dati raccolti nel file *discoverTopology.csv*

Disponibile in appendice B.3

Analisi dei risultati dell'esperimento e commenti

La rete si è comportata bene in generale, mantenendo al minimo la perdita di pacchetti. Unica eccezione si è verificata durante la seconda iterazione dell'esperimento condotto sulla rete composta da 11 nodi. Per un'analisi più dettagliata dei dati raccolti, si rimanda alla sezione successiva.

5.3 Analisi della performance di rete

Analizziamo ora più a fondo i dati risultanti dall'*Esperimento 2* per poter effettuare un'**analisi delle performance della rete**. L'Esperimento 2 è particolarmente adatto a questi test poiché non prevede timer interni ai nodi che possano alterare le tempistiche di comunicazione misurate. Inoltre, il tipo di servizio utilizzato non comporta tempi di computazione significativi all'interno del nodo, rendendo trascurabile il tempo necessario per il recupero dei parametri.

Analizzeremo i tempi di risposta della rete tramite **Round Trip Time (RTT)** e la quantità di pacchetti consegnati e persi durante il procedimento mediante **Packet Delivery Ratio (PDR)**.

5.3.1 Metriche

Round Trip Time (RTT)

Il Round Trip Time (RTT) è la durata, qua considerata in secondi, impiegata da un segnale per andare da un punto di partenza (nodo client) a una destinazione (nodo server) e tornare di nuovo al punto di partenza. È una buona metrica per dare una stima della quantità di tempo richiesta per trasferire i dati all'interno della rete.

In questo senso indica quindi il tempo impiegato da una `parameterRequest` per ricevere i dati dai nodi della rete.

La formula dell'RTT è rappresentabile come segue:

$$RTT = endTime - startTime$$

Il concetto di RTT include intrinsecamente il concetto di *latenza di rete*, anche chiamata *delay* o *end-to-end delay*. Questa è definita come “il tempo impiegato da un pacchetto per essere trasmesso attraverso una rete dalla sorgente alla destinazione” [40], ed è approssimabile a metà dell’RTT.

Definiamo inoltre *intervallo di confidenza* (o per meglio dire *intervallo di fiducia*, dall’inglese *confidence interval*) come “l’intervallo di valori probabili per un dato parametro”. In questo contesto il termine viene utilizzato per indicare la fascia di valori RTT ottenuti, espressi in secondi, calcolati sulla base della serie di esperimenti condotti.

Packet Delivery Ratio (PDR)

Il Packet Delivery Ratio rappresenta il rapporto tra il numero di pacchetti ricevuti correttamente da un nodo e il numero di pacchetti totali generati alla sorgente. È una metrica per valutare l’affidabilità della rete e segnalare eventuali problemi di congestione di rete. Più il PDR si avvicina a 1, più la connessione è di qualità.

Nel nostro caso consideriamo “pacchetti ricevuti” quelli salvati nel CSV e visualizzabili testualmente, e “pacchetti generati” come quelli che ci aspettiamo di ricevere a seconda della fase dell’esperimento in cui siamo: se il passo di esperimento coinvolge 3 nodi, ci aspettiamo di ricevere 3 messaggi, se coinvolge 5 nodi, 5 messaggi, e così via.

Indichiamo la formula per il PDR:

$$PDR = \frac{\sum receivedPacket}{\sum sentPacket}$$

5.3.2 Procedimento

Come descritto nel log manuale B.1, l’*Esperimento 2* è stato strutturato in una serie di *sotto-esperimenti*, ciascuno caratterizzato da un diverso numero di nodi nella rete. Si è partiti con una rete di 3 nodi, incrementando gradualmente di 2 nodi per ogni sotto-esperimento, fino a raggiungere un massimo di 11 nodi. Per garantire la robustezza dei risultati, ogni sotto-esperimento è stato ripetuto 3 volte, aggregandone i dati per favorire la visualizzazione tramite grafico.

L’**estrazione dei dati, l’analisi e la creazione del grafico** sono state realizzate utilizzando lo script Python `analysis.py`, sviluppato ad-hoc. In particolare, è stato effettuato il parsing del file di log generato dall’esperimento, identificando per ogni pacchetto:

- *startTime*, dato dal timestamp presente nella riga contenente la descrizione “Sent parameters request”, indice dell’invio della richiesta dei parametri
- *endTime*, dato dal timestamp presente nella riga contenente la descrizione “Data packet received from 0xYYYY”, indice dell’arrivo del pacchetto richiesto per quel dato nodo

Lo script calcola anche il **numero totale di pacchetti ricevuti** per ogni esperimento, confrontandolo con il numero di nodi attesi mediante funzioni dedicate all’interno del file. Ad esempio, per un esperimento condotto su una rete con 3 nodi, ci si aspetta di ricevere 3 messaggi di risposta, mentre per un esperimento con 5 nodi si attendono 5 messaggi, e così via.

5.3.3 Risultati

Sebbene il numero di dati raccolti non sia sufficiente per delineare un comportamento generale della rete, è comunque possibile trarre alcune osservazioni significative, anche grazie al supporto grafico dato dalla figura 5.6.

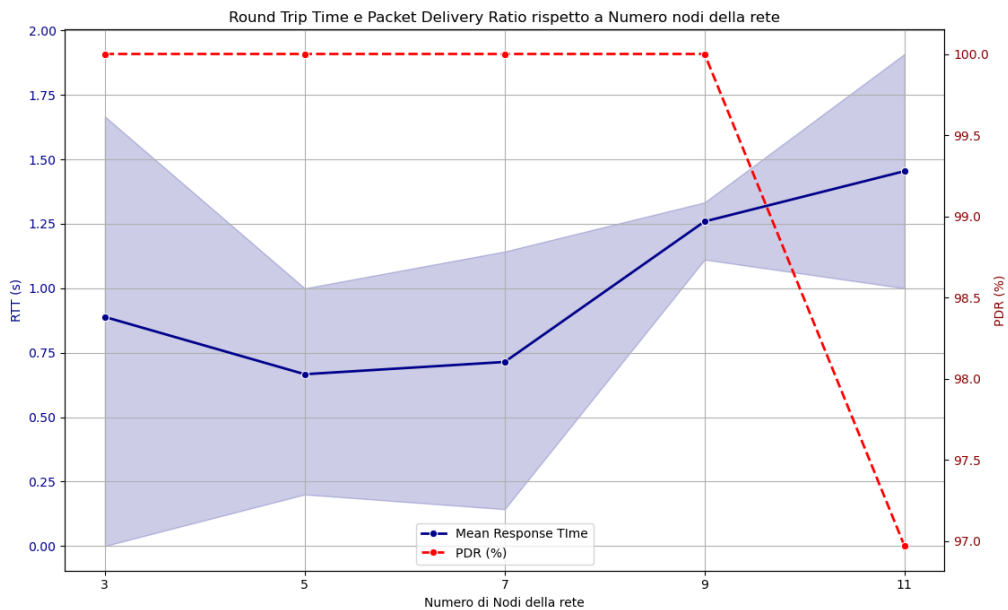


Figura 5.6: Grafico con metriche per l’analisi dei risultati di *Esperimento 2*

Parlando di **affidabilità della rete**, notiamo che il *Packet Delivery Ratio* si mantiene su livelli elevati, evidenziando un leggero calo al 97% solo nelle reti con 11

nodi. Questo suggerisce una buona scalabilità della rete, con una perdita minima di informazioni anche al crescere della complessità.

Risulta interessante anche l'analisi dei **tempi di risposta** tramite studio dell'RTT. Il *tempo medio di risposta*, come prevedibile, aumenta leggermente con l'aumentare del numero di nodi, passando da meno di 1 secondo a quasi 1,5 secondi per le reti con 11 nodi.

Un'analisi dell'*intervallo di confidenza* rivela alcune peculiarità. Nei sotto-esperimenti con 3 nodi, l'intervallo è piuttosto ampio, variando da 0 secondi a circa 1,6 secondi. Questo dato anomalo potrebbe essere attribuito a disturbi ambientali o eventi imprevisti durante l'esperimento, che hanno causato ritardi nel segnale. È improbabile infatti che i tempi di risposta nei sotto-esperimenti con 3 nodi siano superiori a quelli con 5 nodi, i quali seguono meglio il trend generale. Globalmente la larghezza dell'intervallo di confidenza tende a ridursi all'aumentare del numero di nodi, diventando più stabile nei sotto-esperimenti con 9 e 11 nodi. Questo rende i valori medi più rappresentativi e affidabili.

L'analisi dell'RTT evidenzia un **potenziale rischio di perdita di performance** in termini di velocità di rete a seguito dell'aumento del numero di nodi. Questo fenomeno, se confermato da ulteriori esperimenti e analisi, potrebbe **suggerire la necessità di modifiche all'architettura di rete**.

Tra le soluzioni possibili si potrebbe considerare un approccio di *decentralizzazione* nelle operazioni di richiesta e raccolta delle risposte, aumentando il numero di nodi atti a svolgere questi incarichi, garantendone una distribuzione più omogenea all'interno della rete. Inoltre, potrebbe essere utile *suddividere la rete in sotto-reti* per ottimizzare l'efficienza complessiva, riducendo i colli di bottiglia e migliorando le prestazioni generali.

Capitolo 6

Conclusione e sviluppi futuri

6.1 Conclusione

La tesi si concentra sulla progettazione, implementazione e analisi di una rete Bluetooth Low Energy di tipo Mesh, con attenzione particolare allo sviluppo di soluzioni per la scoperta della topologia della rete e per il recupero di alcuni parametri dei nodi nella rete.

Il sistema offre due servizi principali.

Il primo, chiamato *discoverTopology*, raccoglie dai nodi informazioni riguardanti i loro vicini immediati. I dati raccolti vengono elaborati per creare uno schema grafico che rappresenti la rete e i collegamenti tra i nodi, oltre a fornire una stima approssimativa delle distanze tra essi attraverso la misura dell'intensità di segnale.

Il secondo servizio, *parameterRequest*, consente di raccogliere informazioni dettagliate sui parametri di configurazione di ciascun nodo e della rete nel suo insieme. È possibile acquisire dati come indirizzo Unicast del nodo, TTL dei messaggi, potenza di trasmissione, indicando anche se il dispositivo è configurato come relay. A scopo didattico vengono raccolte anche informazioni sulle chiavi di rete (*NetKey*) e applicative (*AppKey*), fondamentali per garantire comunicazioni sicure tra i nodi.

L'utente interagisce con il sistema tramite una Dashboard intuitiva, che consente di selezionare il servizio desiderato. Tutti i dati raccolti e le operazioni effettuate sono salvati in file di log e CSV, corredati da timestamp, per facilitare analisi e studi statistici.

A questo proposito sono state condotte analisi sperimentali sulle performance della rete, esaminando in particolare metriche quali Round Trip Time (RTT) e Packet

Delivery Ratio (PDR). Gli esperimenti hanno valutato il comportamento della rete al variare del numero di nodi, misurandone l'efficienza e l'affidabilità. Sulla base dei risultati, sono state ipotizzate possibili soluzioni per migliorare le prestazioni complessive.

6.2 Sviluppi futuri

Il lavoro svolto in questa tesi apre la strada a numerose possibilità di ampliamento e miglioramento, sia a livello di architettura che di funzionalità. Di seguito, presentiamo alcune idee per sviluppi futuri.

Un primo ambito di miglioramento riguarda l'**architettura della rete**, con l'obiettivo di *distribuire in maniera più efficiente il carico di lavoro attualmente gestito dal client*, che si occupa interamente di gestione delle richieste, ricezione delle risposte ed elaborazione di queste ultime. Come già suggerito in 5.3.3, un approccio potrebbe essere quello di introdurre più nodi client, distribuendoli equamente nella rete: questa soluzione aiuterebbe a mitigare i colli di bottiglia che potrebbero verificarsi con l'aumento delle dimensioni della stessa. Un'altra proposta atta a ottimizzare la rete potrebbe essere la suddivisione di questa in più sotto-reti, rendendo la comunicazione più rapida ed efficiente.

Un'altra evoluzione interessante sarebbe quella di *dotare il client di capacità di comunicazione wireless con la Dashboard*, eliminando la necessità di una connessione cablata. Si potrebbero sfruttare le funzionalità WiFi degli ESP32 (che supportano in contemporanea sia Bluetooth che WiFi), collegando il client ad un access point e consentendo il controllo della rete a distanza tramite una Dashboard accessibile da PC, oppure tramite una web app o un'app dedicata per smartphone.

Sono possibili anche diversi miglioramenti per ottimizzare le **funzionalità della rete**. Nel servizio di scoperta della topologia *discoverTopology*, si potrebbe *implementare* il meccanismo basato sul *Publishing di messaggi Heartbeat*, come suggerito in 4.3.1. Effettuando confronti di prestazioni e di consumo energetico rispetto alla soluzione attuale, ci sarebbe la possibilità di identificare quale approccio sia più adatto a scenari specifici.

Inoltre, per *migliorare la precisione della rappresentazione grafica* dei nodi nella rete, sarebbe saggio sfruttare metodi avanzati per stimare le distanze. Si potrebbe impiegare in maniera più approfondita l'uso dell'RSSI per calcolare la distanza reale in metri tra i nodi, oltre a sfruttare tecnologie tutt'ora in sviluppo come il *Bluetooth Direction Finding*. Quest'ultima tecnica, attraverso l'analisi di parametri come l'*Angle of Arrival (AoA)* e l'*Angle of Departure (AoD)*, consente di determinare con elevata accuratezza la posizione dei nodi nella rete.

Un ulteriore passo avanti nella rappresentazione grafica della rete consiste nel mantenere fisse le posizioni dei nodi nel grafico della topologia, evitando che queste cambino a ogni aggiornamento: in questo modo si renderebbe il grafico più leggibile e intuitivo per gli utenti.

Un'evoluzione da cui trarrebbe particolare vantaggio l'utente finale, anche non esperto o privo di competenze tecniche nel settore, potrebbe consistere in una *maggiore integrazione della componente di analisi* direttamente nel sistema: esso stesso analizza i dati, calcola i valori delle metriche che vogliamo considerare, evidenzia i problemi e consiglia come intervenire per risolverli, eventualmente anche supportato da meccanismi di intelligenza artificiale.

A tale proposito, anche visto il forte sviluppo degli ultimi anni, si potrebbero implementare *meccanismi di Machine Learning*, in particolar modo tramite Reinforcement Learning, per poter consentire al sistema di diventare autonomo, adattandosi dinamicamente alle condizioni operative e ottimizzando i parametri per migliorare le prestazioni. Recenti studi e lavori di tesi, come [41], stanno già esplorando questo campo, dimostrando il potenziale delle reti BLE Mesh auto-configuranti basate su tecnologie di apprendimento automatico.

Percorrendo questa via, *l'interazione con il sistema* diventerebbe più *simile ad un supporto consulenziale*, in cui l'utente riceve indicazioni chiare, senza dover necessariamente comprendere i dettagli tecnici sottostanti, liberandosi inoltre dalla necessità di essere costantemente affiancato da un esperto, figura che rimarrebbe comunque fondamentale nella progettazione della rete e nel supporto a lungo termine.

Appendice A

Dati di supporto alla validazione: Esperimento 1

Nota: alcuni termini presenti nel logfile sono stati abbreviati per favorire una più piacevole esperienza di lettura. In particolare:

- dT_Srv = discoverTopology_Service
- dT.csv = discoverTopology.csv
- drawTG = draw_topology_graph
- NetTopGraph = Network Topology Graph

A.1 File di log manuale

Timestamp (HH.MM)	Operazione	Nodo
12.15	start	-
12.17	add	0x0005
12.17	add	0x0006
12.18	add	0x0007
12.18	add	0x0008
12.20	add	0x0009
12.21	add	0x000a
12.25	add	0x000b
12.27	add dynamic node	0x000c
12.28	moved	0x000c
12.31	remove	0x0005
12.32	remove	0x0009
12.32	moved antenna	0x0001
12.34	re-add	0x0009
12.35	add	0x000d
12.36	moved	0x000d
12.37	removing all nodes	-

Tabella A.1: File di log scritto a mano durante l'esperimento per discoverTopology

A.2 File di log

```

1 2024-11-19 12:15:18 INFO : <module> : Selected [1] - dT_Srv (continuous req)
2 2024-11-19 12:15:18 INFO : dT_Srv : Sent parameters request
3 2024-11-19 12:15:18 INFO : dT_Srv : Started 10 second timer
4 2024-11-19 12:15:38 WARNING : dT_Srv : No data received from network nodes
5 2024-11-19 12:15:38 INFO : dT_Srv : Data id=850, iterationId=899 in data/dT.csv
6 2024-11-19 12:15:38 INFO : dT_Srv : Calling drawTG
7 2024-11-19 12:15:38 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0001
8 2024-11-19 12:15:39 INFO : dT_Srv : Returned from drawTG
9 2024-11-19 12:16:43 INFO : dT_Srv : Sent parameters request
10 2024-11-19 12:16:43 INFO : dT_Srv : Started 10 second timer
11 2024-11-19 12:16:55 INFO : dT_Srv : Data packet received from 0x0005
12 2024-11-19 12:17:03 INFO : dT_Srv : Data id=413, iterationId=899 in data/dT.csv
13 2024-11-19 12:17:03 INFO : dT_Srv : Calling drawTG

```

```
14 2024-11-19 12:17:03 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001
15 2024-11-19 12:17:04 INFO : dT_Srv : Returned from drawTG
16 2024-11-19 12:17:04 INFO : dT_Srv : Sent parameters request
17 2024-11-19 12:17:04 INFO : dT_Srv : Started 10 second timer
18 2024-11-19 12:17:14 INFO : dT_Srv : Data packet received from 0x0005
19 2024-11-19 12:17:24 INFO : dT_Srv : Data id=131, iterationId=899 in data/dT.csv
20 2024-11-19 12:17:24 INFO : dT_Srv : Calling drawTG
21 2024-11-19 12:17:24 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001
22 2024-11-19 12:17:25 INFO : dT_Srv : Returned from drawTG
23 2024-11-19 12:17:25 INFO : dT_Srv : Sent parameters request
24 2024-11-19 12:17:25 INFO : dT_Srv : Started 10 second timer
25 2024-11-19 12:17:37 INFO : dT_Srv : Data packet received from 0x0005
26 2024-11-19 12:17:37 INFO : dT_Srv : Data packet received from 0x0006
27 2024-11-19 12:17:45 INFO : dT_Srv : Data id=556, iterationId=899 in data/dT.csv
28 2024-11-19 12:17:45 INFO : dT_Srv : Calling drawTG
29 2024-11-19 12:17:45 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001,
    0x0006
30 2024-11-19 12:17:46 INFO : dT_Srv : Returned from drawTG
31 2024-11-19 12:17:46 INFO : dT_Srv : Sent parameters request
32 2024-11-19 12:17:46 INFO : dT_Srv : Started 10 second timer
33 2024-11-19 12:17:55 INFO : dT_Srv : Data packet received from 0x0005
34 2024-11-19 12:17:56 INFO : dT_Srv : Data packet received from 0x0006
35 2024-11-19 12:18:06 INFO : dT_Srv : Data id=66, iterationId=899 in data/dT.csv
36 2024-11-19 12:18:06 INFO : dT_Srv : Calling drawTG
37 2024-11-19 12:18:06 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001,
    0x0006
38 2024-11-19 12:18:07 INFO : dT_Srv : Returned from drawTG
39 2024-11-19 12:18:07 INFO : dT_Srv : Sent parameters request
40 2024-11-19 12:18:07 INFO : dT_Srv : Started 10 second timer
41 2024-11-19 12:18:17 INFO : dT_Srv : Data packet received from 0x0007
42 2024-11-19 12:18:18 INFO : dT_Srv : Data packet received from 0x0005
43 2024-11-19 12:18:19 INFO : dT_Srv : Data packet received from 0x0006
44 2024-11-19 12:18:27 INFO : dT_Srv : Data id=266, iterationId=899 in data/dT.csv
45 2024-11-19 12:18:27 INFO : dT_Srv : Calling drawTG
46 2024-11-19 12:18:27 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0005, 0x0006
47 2024-11-19 12:18:28 INFO : dT_Srv : Returned from drawTG
48 2024-11-19 12:18:28 INFO : dT_Srv : Sent parameters request
49 2024-11-19 12:18:28 INFO : dT_Srv : Started 10 second timer
50 2024-11-19 12:18:37 INFO : dT_Srv : Data packet received from 0x0005
51 2024-11-19 12:18:38 INFO : dT_Srv : Data packet received from 0x0006
52 2024-11-19 12:18:40 INFO : dT_Srv : Data packet received from 0x0007
53 2024-11-19 12:18:48 INFO : dT_Srv : Data id=229, iterationId=899 in data/dT.csv
54 2024-11-19 12:18:48 INFO : dT_Srv : Calling drawTG
55 2024-11-19 12:18:48 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001,
    0x0006, 0x0007
56 2024-11-19 12:18:49 INFO : dT_Srv : Returned from drawTG
57 2024-11-19 12:18:49 INFO : dT_Srv : Sent parameters request
58 2024-11-19 12:18:49 INFO : dT_Srv : Started 10 second timer
59 2024-11-19 12:18:59 INFO : dT_Srv : Data packet received from 0x0007
```

```

60 2024-11-19 12:18:59 INFO : dT_Srv : Data packet received from 0x0008
61 2024-11-19 12:19:00 INFO : dT_Srv : Data packet received from 0x0005
62 2024-11-19 12:19:01 INFO : dT_Srv : Data packet received from 0x0006
63 2024-11-19 12:19:09 INFO : dT_Srv : Data id=780, iterationId=899 in data/dT.csv
64 2024-11-19 12:19:09 INFO : dT_Srv : Calling drawTG
65 2024-11-19 12:19:09 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0008, 0x0005, 0x0006
66 2024-11-19 12:19:10 INFO : dT_Srv : Returned from drawTG
67 2024-11-19 12:19:10 INFO : dT_Srv : Sent parameters request
68 2024-11-19 12:19:10 INFO : dT_Srv : Started 10 second timer
69 2024-11-19 12:19:19 INFO : dT_Srv : Data packet received from 0x0006
70 2024-11-19 12:19:22 INFO : dT_Srv : Data packet received from 0x0007
71 2024-11-19 12:19:22 INFO : dT_Srv : Data packet received from 0x0008
72 2024-11-19 12:19:23 INFO : dT_Srv : Data packet received from 0x0005
73 2024-11-19 12:19:30 INFO : dT_Srv : Data id=413, iterationId=899 in data/dT.csv
74 2024-11-19 12:19:30 INFO : dT_Srv : Calling drawTG
75 2024-11-19 12:19:30 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0007, 0x0008, 0x0005
76 2024-11-19 12:19:31 INFO : dT_Srv : Returned from drawTG
77 2024-11-19 12:19:31 INFO : dT_Srv : Sent parameters request
78 2024-11-19 12:19:31 INFO : dT_Srv : Started 10 second timer
79 2024-11-19 12:19:41 INFO : dT_Srv : Data packet received from 0x0007
80 2024-11-19 12:19:41 INFO : dT_Srv : Data packet received from 0x0008
81 2024-11-19 12:19:42 INFO : dT_Srv : Data packet received from 0x0005
82 2024-11-19 12:19:42 INFO : dT_Srv : Data packet received from 0x0006
83 2024-11-19 12:19:51 INFO : dT_Srv : Data id=495, iterationId=899 in data/dT.csv
84 2024-11-19 12:19:51 INFO : dT_Srv : Calling drawTG
85 2024-11-19 12:19:52 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0008, 0x0005, 0x0006
86 2024-11-19 12:19:53 INFO : dT_Srv : Returned from drawTG
87 2024-11-19 12:19:53 INFO : dT_Srv : Sent parameters request
88 2024-11-19 12:19:53 INFO : dT_Srv : Started 10 second timer
89 2024-11-19 12:20:01 INFO : dT_Srv : Data packet received from 0x0006
90 2024-11-19 12:20:01 INFO : dT_Srv : Data packet received from 0x0008
91 2024-11-19 12:20:03 INFO : dT_Srv : Data packet received from 0x0007
92 2024-11-19 12:20:05 INFO : dT_Srv : Data packet received from 0x0005
93 2024-11-19 12:20:13 INFO : dT_Srv : Data id=894, iterationId=899 in data/dT.csv
94 2024-11-19 12:20:13 INFO : dT_Srv : Calling drawTG
95 2024-11-19 12:20:13 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0008, 0x0007, 0x0005
96 2024-11-19 12:20:14 INFO : dT_Srv : Returned from drawTG
97 2024-11-19 12:20:14 INFO : dT_Srv : Sent parameters request
98 2024-11-19 12:20:14 INFO : dT_Srv : Started 10 second timer
99 2024-11-19 12:20:22 INFO : dT_Srv : Data packet received from 0x0007
100 2024-11-19 12:20:24 INFO : dT_Srv : Data packet received from 0x0006
101 2024-11-19 12:20:24 INFO : dT_Srv : Data packet received from 0x0008
102 2024-11-19 12:20:25 INFO : dT_Srv : Data packet received from 0x0005
103 2024-11-19 12:20:34 INFO : dT_Srv : Data id=150, iterationId=899 in data/dT.csv
104 2024-11-19 12:20:34 INFO : dT_Srv : Calling drawTG
105 2024-11-19 12:20:34 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,

```

```

0x0006, 0x0008, 0x0005
106 2024-11-19 12:20:35 INFO : dT_Srv : Returned from drawTG
107 2024-11-19 12:20:35 INFO : dT_Srv : Sent parameters request
108 2024-11-19 12:20:35 INFO : dT_Srv : Started 10 second timer
109 2024-11-19 12:20:45 INFO : dT_Srv : Data packet received from 0x0007
110 2024-11-19 12:20:46 INFO : dT_Srv : Data packet received from 0x0009
111 2024-11-19 12:20:46 INFO : dT_Srv : Data packet received from 0x0005
112 2024-11-19 12:20:47 INFO : dT_Srv : Data packet received from 0x0006
113 2024-11-19 12:20:47 INFO : dT_Srv : Data packet received from 0x0008
114 2024-11-19 12:20:55 INFO : dT_Srv : Data id=598, iterationId=899 in data/dT.csv
115 2024-11-19 12:20:55 INFO : dT_Srv : Calling drawTG
116 2024-11-19 12:20:55 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
0x0005, 0x0006, 0x0008, 0x0009
117 2024-11-19 12:20:56 INFO : dT_Srv : Returned from drawTG
118 2024-11-19 12:20:56 INFO : dT_Srv : Sent parameters request
119 2024-11-19 12:20:56 INFO : dT_Srv : Started 10 second timer
120 2024-11-19 12:21:06 INFO : dT_Srv : Data packet received from 0x0009
121 2024-11-19 12:21:06 INFO : dT_Srv : Data packet received from 0x0008
122 2024-11-19 12:21:06 INFO : dT_Srv : Data packet received from 0x0005
123 2024-11-19 12:21:07 INFO : dT_Srv : Data packet received from 0x0006
124 2024-11-19 12:21:08 INFO : dT_Srv : Data packet received from 0x0007
125 2024-11-19 12:21:16 INFO : dT_Srv : Data id=967, iterationId=899 in data/dT.csv
126 2024-11-19 12:21:16 INFO : dT_Srv : Calling drawTG
127 2024-11-19 12:21:16 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0009, 0x0001,
0x0008, 0x0005, 0x0006, 0x0007
128 2024-11-19 12:21:17 INFO : dT_Srv : Returned from drawTG
129 2024-11-19 12:21:17 INFO : dT_Srv : Sent parameters request
130 2024-11-19 12:21:17 INFO : dT_Srv : Started 10 second timer
131 2024-11-19 12:21:28 INFO : dT_Srv : Data packet received from 0x0007
132 2024-11-19 12:21:29 INFO : dT_Srv : Data packet received from 0x0008
133 2024-11-19 12:21:29 INFO : dT_Srv : Data packet received from 0x0009
134 2024-11-19 12:21:29 INFO : dT_Srv : Data packet received from 0x0005
135 2024-11-19 12:21:30 INFO : dT_Srv : Data packet received from 0x0006
136 2024-11-19 12:21:37 INFO : dT_Srv : Data id=145, iterationId=899 in data/dT.csv
137 2024-11-19 12:21:37 INFO : dT_Srv : Calling drawTG
138 2024-11-19 12:21:37 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
0x0008, 0x0009, 0x0005, 0x0006
139 2024-11-19 12:21:38 INFO : dT_Srv : Returned from drawTG
140 2024-11-19 12:21:38 INFO : dT_Srv : Sent parameters request
141 2024-11-19 12:21:38 INFO : dT_Srv : Started 10 second timer
142 2024-11-19 12:21:47 INFO : dT_Srv : Data packet received from 0x0007
143 2024-11-19 12:21:48 INFO : dT_Srv : Data packet received from 0x0009
144 2024-11-19 12:21:48 INFO : dT_Srv : Data packet received from 0x0008
145 2024-11-19 12:21:48 INFO : dT_Srv : Data packet received from 0x0005
146 2024-11-19 12:21:49 INFO : dT_Srv : Data packet received from 0x0006
147 2024-11-19 12:21:50 INFO : dT_Srv : Data packet received from 0x000a
148 2024-11-19 12:21:58 INFO : dT_Srv : Data id=273, iterationId=899 in data/dT.csv
149 2024-11-19 12:21:58 INFO : dT_Srv : Calling drawTG
150 2024-11-19 12:21:58 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
0x0009, 0x0008, 0x0005, 0x0006, 0x000a

```

151 2024-11-19 12:21:59 INFO : dT_Srv : Returned from drawTG
152 2024-11-19 12:21:59 INFO : dT_Srv : Sent parameters request
153 2024-11-19 12:21:59 INFO : dT_Srv : Started 10 second timer
154 2024-11-19 12:22:10 INFO : dT_Srv : Data packet received from 0x0007
155 2024-11-19 12:22:10 INFO : dT_Srv : Data packet received from 0x000a
156 2024-11-19 12:22:10 INFO : dT_Srv : Data packet received from 0x0008
157 2024-11-19 12:22:11 INFO : dT_Srv : Data packet received from 0x0009
158 2024-11-19 12:22:11 INFO : dT_Srv : Data packet received from 0x0005
159 2024-11-19 12:22:12 INFO : dT_Srv : Data packet received from 0x0006
160 2024-11-19 12:22:19 INFO : dT_Srv : Data id=876, iterationId=899 in data/dT.csv
161 2024-11-19 12:22:19 INFO : dT_Srv : Calling drawTG
162 2024-11-19 12:22:20 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001, 0x0008, 0x0009, 0x0005, 0x0006, 0x000a
163 2024-11-19 12:22:21 INFO : dT_Srv : Returned from drawTG
164 2024-11-19 12:22:21 INFO : dT_Srv : Sent parameters request
165 2024-11-19 12:22:21 INFO : dT_Srv : Started 10 second timer
166 2024-11-19 12:22:30 INFO : dT_Srv : Data packet received from 0x0005
167 2024-11-19 12:22:30 INFO : dT_Srv : Data packet received from 0x0006
168 2024-11-19 12:22:32 INFO : dT_Srv : Data packet received from 0x000a
169 2024-11-19 12:22:33 INFO : dT_Srv : Data packet received from 0x0007
170 2024-11-19 12:22:33 INFO : dT_Srv : Data packet received from 0x0008
171 2024-11-19 12:22:33 INFO : dT_Srv : Data packet received from 0x0009
172 2024-11-19 12:22:41 INFO : dT_Srv : Data id=498, iterationId=899 in data/dT.csv
173 2024-11-19 12:22:41 INFO : dT_Srv : Calling drawTG
174 2024-11-19 12:22:41 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001, 0x0006, 0x0007, 0x0008, 0x0009, 0x000a
175 2024-11-19 12:22:42 INFO : dT_Srv : Returned from drawTG
176 2024-11-19 12:22:42 INFO : dT_Srv : Sent parameters request
177 2024-11-19 12:22:42 INFO : dT_Srv : Started 10 second timer
178 2024-11-19 12:22:51 INFO : dT_Srv : Data packet received from 0x0007
179 2024-11-19 12:22:52 INFO : dT_Srv : Data packet received from 0x0008
180 2024-11-19 12:22:52 INFO : dT_Srv : Data packet received from 0x0009
181 2024-11-19 12:22:53 INFO : dT_Srv : Data packet received from 0x0005
182 2024-11-19 12:22:53 INFO : dT_Srv : Data packet received from 0x0006
183 2024-11-19 12:22:54 INFO : dT_Srv : Data packet received from 0x000a
184 2024-11-19 12:23:02 INFO : dT_Srv : Data id=80, iterationId=899 in data/dT.csv
185 2024-11-19 12:23:02 INFO : dT_Srv : Calling drawTG
186 2024-11-19 12:23:02 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001, 0x0008, 0x0005, 0x0006, 0x0009, 0x000a
187 2024-11-19 12:23:03 INFO : dT_Srv : Returned from drawTG
188 2024-11-19 12:23:03 INFO : dT_Srv : Sent parameters request
189 2024-11-19 12:23:03 INFO : dT_Srv : Started 10 second timer
190 2024-11-19 12:23:12 INFO : dT_Srv : Data packet received from 0x0005
191 2024-11-19 12:23:12 INFO : dT_Srv : Data packet received from 0x0006
192 2024-11-19 12:23:13 INFO : dT_Srv : Data packet received from 0x000a
193 2024-11-19 12:23:14 INFO : dT_Srv : Data packet received from 0x0007
194 2024-11-19 12:23:15 INFO : dT_Srv : Data packet received from 0x0009
195 2024-11-19 12:23:15 INFO : dT_Srv : Data packet received from 0x0008
196 2024-11-19 12:23:23 INFO : dT_Srv : Data id=467, iterationId=899 in data/dT.csv
197 2024-11-19 12:23:23 INFO : dT_Srv : Calling drawTG

198 2024-11-19 12:23:23 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001, 0x0006, 0x0007, 0x0008, 0x0009, 0x000a
199 2024-11-19 12:23:24 INFO : dT_Srv : Returned from drawTG
200 2024-11-19 12:23:24 INFO : dT_Srv : Sent parameters request
201 2024-11-19 12:23:24 INFO : dT_Srv : Started 10 second timer
202 2024-11-19 12:23:33 INFO : dT_Srv : Data packet received from 0x0007
203 2024-11-19 12:23:34 INFO : dT_Srv : Data packet received from 0x0009
204 2024-11-19 12:23:34 INFO : dT_Srv : Data packet received from 0x0008
205 2024-11-19 12:23:34 INFO : dT_Srv : Data packet received from 0x0005
206 2024-11-19 12:23:35 INFO : dT_Srv : Data packet received from 0x0006
207 2024-11-19 12:23:36 INFO : dT_Srv : Data packet received from 0x000a
208 2024-11-19 12:23:44 INFO : dT_Srv : Data id=835, iterationId=899 in data/dT.csv
209 2024-11-19 12:23:44 INFO : dT_Srv : Calling drawTG
210 2024-11-19 12:23:44 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001, 0x0008, 0x0009, 0x0005, 0x0006, 0x000a
211 2024-11-19 12:23:45 INFO : dT_Srv : Returned from drawTG
212 2024-11-19 12:23:45 INFO : dT_Srv : Sent parameters request
213 2024-11-19 12:23:45 INFO : dT_Srv : Started 10 second timer
214 2024-11-19 12:23:54 INFO : dT_Srv : Data packet received from 0x0006
215 2024-11-19 12:23:56 INFO : dT_Srv : Data packet received from 0x0007
216 2024-11-19 12:23:57 INFO : dT_Srv : Data packet received from 0x000a
217 2024-11-19 12:23:57 INFO : dT_Srv : Data packet received from 0x0008
218 2024-11-19 12:23:57 INFO : dT_Srv : Data packet received from 0x0009
219 2024-11-19 12:23:58 INFO : dT_Srv : Data packet received from 0x0005
220 2024-11-19 12:24:05 INFO : dT_Srv : Data id=339, iterationId=899 in data/dT.csv
221 2024-11-19 12:24:05 INFO : dT_Srv : Calling drawTG
222 2024-11-19 12:24:05 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0007, 0x0008, 0x0005, 0x0009, 0x000a
223 2024-11-19 12:24:06 INFO : dT_Srv : Returned from drawTG
224 2024-11-19 12:24:06 INFO : dT_Srv : Sent parameters request
225 2024-11-19 12:24:06 INFO : dT_Srv : Started 10 second timer
226 2024-11-19 12:24:15 INFO : dT_Srv : Data packet received from 0x0008
227 2024-11-19 12:24:16 INFO : dT_Srv : Data packet received from 0x0009
228 2024-11-19 12:24:16 INFO : dT_Srv : Data packet received from 0x0005
229 2024-11-19 12:24:17 INFO : dT_Srv : Data packet received from 0x0006
230 2024-11-19 12:24:18 INFO : dT_Srv : Data packet received from 0x000a
231 2024-11-19 12:24:19 INFO : dT_Srv : Data packet received from 0x0007
232 2024-11-19 12:24:26 INFO : dT_Srv : Data id=557, iterationId=899 in data/dT.csv
233 2024-11-19 12:24:26 INFO : dT_Srv : Calling drawTG
234 2024-11-19 12:24:27 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001, 0x0009, 0x0005, 0x0006, 0x000a, 0x0007
235 2024-11-19 12:24:28 INFO : dT_Srv : Returned from drawTG
236 2024-11-19 12:24:28 INFO : dT_Srv : Sent parameters request
237 2024-11-19 12:24:28 INFO : dT_Srv : Started 10 second timer
238 2024-11-19 12:24:38 INFO : dT_Srv : Data packet received from 0x0007
239 2024-11-19 12:24:38 INFO : dT_Srv : Data packet received from 0x0008
240 2024-11-19 12:24:38 INFO : dT_Srv : Data packet received from 0x0009
241 2024-11-19 12:24:39 INFO : dT_Srv : Data packet received from 0x000a
242 2024-11-19 12:24:39 INFO : dT_Srv : Data packet received from 0x0005
243 2024-11-19 12:24:39 INFO : dT_Srv : Data packet received from 0x0006

244 2024-11-19 12:24:48 INFO : dT_Srv : Data id=739, iterationId=899 in data/dT.csv
245 2024-11-19 12:24:48 INFO : dT_Srv : Calling drawTG
246 2024-11-19 12:24:48 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
0x0008, 0x0009, 0x0005, 0x0006, 0x000a
247 2024-11-19 12:24:49 INFO : dT_Srv : Returned from drawTG
248 2024-11-19 12:24:49 INFO : dT_Srv : Sent parameters request
249 2024-11-19 12:24:49 INFO : dT_Srv : Started 10 second timer
250 2024-11-19 12:24:58 INFO : dT_Srv : Data packet received from 0x0005
251 2024-11-19 12:24:58 INFO : dT_Srv : Data packet received from 0x0006
252 2024-11-19 12:25:01 INFO : dT_Srv : Data packet received from 0x0007
253 2024-11-19 12:25:01 INFO : dT_Srv : Data packet received from 0x0008
254 2024-11-19 12:25:01 INFO : dT_Srv : Data packet received from 0x0009
255 2024-11-19 12:25:01 INFO : dT_Srv : Data packet received from 0x000a
256 2024-11-19 12:25:09 INFO : dT_Srv : Data id=72, iterationId=899 in data/dT.csv
257 2024-11-19 12:25:09 INFO : dT_Srv : Calling drawTG
258 2024-11-19 12:25:09 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001,
0x0006, 0x0007, 0x0008, 0x0009, 0x000a
259 2024-11-19 12:25:10 INFO : dT_Srv : Returned from drawTG
260 2024-11-19 12:25:10 INFO : dT_Srv : Sent parameters request
261 2024-11-19 12:25:10 INFO : dT_Srv : Started 10 second timer
262 2024-11-19 12:25:19 INFO : dT_Srv : Data packet received from 0x0007
263 2024-11-19 12:25:20 INFO : dT_Srv : Data packet received from 0x0008
264 2024-11-19 12:25:20 INFO : dT_Srv : Data packet received from 0x0009
265 2024-11-19 12:25:20 INFO : dT_Srv : Data packet received from 0x000a
266 2024-11-19 12:25:21 INFO : dT_Srv : Data packet received from 0x0005
267 2024-11-19 12:25:21 INFO : dT_Srv : Data packet received from 0x0006
268 2024-11-19 12:25:30 INFO : dT_Srv : Data id=698, iterationId=899 in data/dT.csv
269 2024-11-19 12:25:30 INFO : dT_Srv : Calling drawTG
270 2024-11-19 12:25:30 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
0x0009, 0x0008, 0x0005, 0x0006, 0x000a
271 2024-11-19 12:25:31 INFO : dT_Srv : Returned from drawTG
272 2024-11-19 12:25:31 INFO : dT_Srv : Sent parameters request
273 2024-11-19 12:25:31 INFO : dT_Srv : Started 10 second timer
274 2024-11-19 12:25:42 INFO : dT_Srv : Data packet received from 0x000b
275 2024-11-19 12:25:42 INFO : dT_Srv : Data packet received from 0x0007
276 2024-11-19 12:25:42 INFO : dT_Srv : Data packet received from 0x000a
277 2024-11-19 12:25:43 INFO : dT_Srv : Data packet received from 0x0008
278 2024-11-19 12:25:43 INFO : dT_Srv : Data packet received from 0x0009
279 2024-11-19 12:25:44 INFO : dT_Srv : Data packet received from 0x0005
280 2024-11-19 12:25:44 INFO : dT_Srv : Data packet received from 0x0006
281 2024-11-19 12:25:51 INFO : dT_Srv : Data id=517, iterationId=899 in data/dT.csv
282 2024-11-19 12:25:51 INFO : dT_Srv : Calling drawTG
283 2024-11-19 12:25:51 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000b, 0x0001,
0x0009, 0x0007, 0x0008, 0x0005, 0x0006, 0x000a
284 2024-11-19 12:25:52 INFO : dT_Srv : Returned from drawTG
285 2024-11-19 12:25:52 INFO : dT_Srv : Sent parameters request
286 2024-11-19 12:25:52 INFO : dT_Srv : Started 10 second timer
287 2024-11-19 12:26:02 INFO : dT_Srv : Data packet received from 0x000a
288 2024-11-19 12:26:02 INFO : dT_Srv : Data packet received from 0x0009
289 2024-11-19 12:26:02 INFO : dT_Srv : Data packet received from 0x0008


```

290 2024-11-19 12:26:02 INFO : dT_Srv : Data packet received from 0x0005
291 2024-11-19 12:26:03 INFO : dT_Srv : Data packet received from 0x0006
292 2024-11-19 12:26:05 INFO : dT_Srv : Data packet received from 0x000b
293 2024-11-19 12:26:05 INFO : dT_Srv : Data packet received from 0x0007
294 2024-11-19 12:26:12 INFO : dT_Srv : Data id=900, iterationId=899 in data/dT.csv
295 2024-11-19 12:26:12 INFO : dT_Srv : Calling drawTG
296 2024-11-19 12:26:12 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0009,
    0x000b, 0x0008, 0x0005, 0x0006, 0x0007, 0x0001
297 2024-11-19 12:26:13 INFO : dT_Srv : Returned from drawTG
298 2024-11-19 12:26:13 INFO : dT_Srv : Sent parameters request
299 2024-11-19 12:26:13 INFO : dT_Srv : Started 10 second timer
300 2024-11-19 12:26:24 INFO : dT_Srv : Data packet received from 0x000b
301 2024-11-19 12:26:25 INFO : dT_Srv : Data packet received from 0x000a
302 2024-11-19 12:26:25 INFO : dT_Srv : Data packet received from 0x0007
303 2024-11-19 12:26:25 INFO : dT_Srv : Data packet received from 0x0009
304 2024-11-19 12:26:25 INFO : dT_Srv : Data packet received from 0x0005
305 2024-11-19 12:26:26 INFO : dT_Srv : Data packet received from 0x0008
306 2024-11-19 12:26:26 INFO : dT_Srv : Data packet received from 0x0006
307 2024-11-19 12:26:33 INFO : dT_Srv : Data id=125, iterationId=899 in data/dT.csv
308 2024-11-19 12:26:33 INFO : dT_Srv : Calling drawTG
309 2024-11-19 12:26:34 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000b, 0x000a,
    0x0009, 0x0001, 0x0007, 0x0005, 0x0006, 0x0008
310 2024-11-19 12:26:35 INFO : dT_Srv : Returned from drawTG
311 2024-11-19 12:26:35 INFO : dT_Srv : Sent parameters request
312 2024-11-19 12:26:35 INFO : dT_Srv : Started 10 second timer
313 2024-11-19 12:26:44 INFO : dT_Srv : Data packet received from 0x000c
314 2024-11-19 12:26:44 INFO : dT_Srv : Data packet received from 0x0006
315 2024-11-19 12:26:45 INFO : dT_Srv : Data packet received from 0x0008
316 2024-11-19 12:26:45 INFO : dT_Srv : Data packet received from 0x000a
317 2024-11-19 12:26:45 INFO : dT_Srv : Data packet received from 0x0009
318 2024-11-19 12:26:45 INFO : dT_Srv : Data packet received from 0x0005
319 2024-11-19 12:26:47 INFO : dT_Srv : Data packet received from 0x000b
320 2024-11-19 12:26:47 INFO : dT_Srv : Data packet received from 0x0007
321 2024-11-19 12:26:55 INFO : dT_Srv : Data id=358, iterationId=899 in data/dT.csv
322 2024-11-19 12:26:55 INFO : dT_Srv : Calling drawTG
323 2024-11-19 12:26:55 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000c, 0x0001,
    0x000a, 0x0006, 0x0008, 0x0009, 0x0005, 0x000b, 0x0007
324 2024-11-19 12:26:56 INFO : dT_Srv : Returned from drawTG
325 2024-11-19 12:26:56 INFO : dT_Srv : Sent parameters request
326 2024-11-19 12:26:56 INFO : dT_Srv : Started 10 second timer
327 2024-11-19 12:27:06 INFO : dT_Srv : Data packet received from 0x0007
328 2024-11-19 12:27:06 INFO : dT_Srv : Data packet received from 0x000a
329 2024-11-19 12:27:06 INFO : dT_Srv : Data packet received from 0x000b
330 2024-11-19 12:27:07 INFO : dT_Srv : Data packet received from 0x000c
331 2024-11-19 12:27:07 INFO : dT_Srv : Data packet received from 0x0006
332 2024-11-19 12:27:08 INFO : dT_Srv : Data packet received from 0x0008
333 2024-11-19 12:27:08 INFO : dT_Srv : Data packet received from 0x0009
334 2024-11-19 12:27:08 INFO : dT_Srv : Data packet received from 0x0005
335 2024-11-19 12:27:16 INFO : dT_Srv : Data id=0, iterationId=899 in data/dT.csv
336 2024-11-19 12:27:16 INFO : dT_Srv : Calling drawTG

```

337 2024-11-19 12:27:16 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001, 0x000c, 0x0006, 0x0008, 0x0005, 0x000a, 0x000b, 0x0009
338 2024-11-19 12:27:17 INFO : dT_Srv : Returned from drawTG
339 2024-11-19 12:27:17 INFO : dT_Srv : Sent parameters request
340 2024-11-19 12:27:17 INFO : dT_Srv : Started 10 second timer
341 2024-11-19 12:27:26 INFO : dT_Srv : Data packet received from 0x0008
342 2024-11-19 12:27:26 INFO : dT_Srv : Data packet received from 0x0009
343 2024-11-19 12:27:27 INFO : dT_Srv : Data packet received from 0x0005
344 2024-11-19 12:27:27 INFO : dT_Srv : Data packet received from 0x0006
345 2024-11-19 12:27:28 INFO : dT_Srv : Data packet received from 0x000b
346 2024-11-19 12:27:29 INFO : dT_Srv : Data packet received from 0x000a
347 2024-11-19 12:27:30 INFO : dT_Srv : Data packet received from 0x0007
348 2024-11-19 12:27:30 INFO : dT_Srv : Data packet received from 0x000c
349 2024-11-19 12:27:37 INFO : dT_Srv : Data id=487, iterationId=899 in data/dT.csv
350 2024-11-19 12:27:37 INFO : dT_Srv : Calling drawTG
351 2024-11-19 12:27:37 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001, 0x0009, 0x0005, 0x0007, 0x0006, 0x000b, 0x000a, 0x000c
352 2024-11-19 12:27:38 INFO : dT_Srv : Returned from drawTG
353 2024-11-19 12:27:38 INFO : dT_Srv : Sent parameters request
354 2024-11-19 12:27:38 INFO : dT_Srv : Started 10 second timer
355 2024-11-19 12:27:48 INFO : dT_Srv : Data packet received from 0x000a
356 2024-11-19 12:27:48 INFO : dT_Srv : Data packet received from 0x000b
357 2024-11-19 12:27:48 INFO : dT_Srv : Data packet received from 0x0007
358 2024-11-19 12:27:49 INFO : dT_Srv : Data packet received from 0x000c
359 2024-11-19 12:27:49 INFO : dT_Srv : Data packet received from 0x0008
360 2024-11-19 12:27:50 INFO : dT_Srv : Data packet received from 0x0009
361 2024-11-19 12:27:50 INFO : dT_Srv : Data packet received from 0x0005
362 2024-11-19 12:27:50 INFO : dT_Srv : Data packet received from 0x0006
363 2024-11-19 12:27:58 INFO : dT_Srv : Data id=424, iterationId=899 in data/dT.csv
364 2024-11-19 12:27:58 INFO : dT_Srv : Calling drawTG
365 2024-11-19 12:27:58 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x000b, 0x000c, 0x0009, 0x0007, 0x0001, 0x0008, 0x0005, 0x0006
366 2024-11-19 12:27:59 INFO : dT_Srv : Returned from drawTG
367 2024-11-19 12:27:59 INFO : dT_Srv : Sent parameters request
368 2024-11-19 12:27:59 INFO : dT_Srv : Started 10 second timer
369 2024-11-19 12:28:08 INFO : dT_Srv : Data packet received from 0x0008
370 2024-11-19 12:28:08 INFO : dT_Srv : Data packet received from 0x0009
371 2024-11-19 12:28:09 INFO : dT_Srv : Data packet received from 0x0005
372 2024-11-19 12:28:09 INFO : dT_Srv : Data packet received from 0x0006
373 2024-11-19 12:28:10 INFO : dT_Srv : Data packet received from 0x000b
374 2024-11-19 12:28:11 INFO : dT_Srv : Data packet received from 0x000a
375 2024-11-19 12:28:11 INFO : dT_Srv : Data packet received from 0x0007
376 2024-11-19 12:28:12 INFO : dT_Srv : Data packet received from 0x000c
377 2024-11-19 12:28:19 INFO : dT_Srv : Data id=478, iterationId=899 in data/dT.csv
378 2024-11-19 12:28:19 INFO : dT_Srv : Calling drawTG
379 2024-11-19 12:28:19 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001, 0x0009, 0x0005, 0x0006, 0x0007, 0x000b, 0x000a, 0x000c
380 2024-11-19 12:28:20 INFO : dT_Srv : Returned from drawTG
381 2024-11-19 12:28:20 INFO : dT_Srv : Sent parameters request
382 2024-11-19 12:28:20 INFO : dT_Srv : Started 10 second timer

```
383 2024-11-19 12:28:30 INFO : dT_Srv : Data packet received from 0x000c
384 2024-11-19 12:28:30 INFO : dT_Srv : Data packet received from 0x0007
385 2024-11-19 12:28:31 INFO : dT_Srv : Data packet received from 0x0008
386 2024-11-19 12:28:31 INFO : dT_Srv : Data packet received from 0x0009
387 2024-11-19 12:28:31 INFO : dT_Srv : Data packet received from 0x0005
388 2024-11-19 12:28:32 INFO : dT_Srv : Data packet received from 0x0006
389 2024-11-19 12:28:33 INFO : dT_Srv : Data packet received from 0x000b
390 2024-11-19 12:28:33 INFO : dT_Srv : Data packet received from 0x000a
391 2024-11-19 12:28:40 INFO : dT_Srv : Data id=741, iterationId=899 in data/dT.csv
392 2024-11-19 12:28:40 INFO : dT_Srv : Calling drawTG
393 2024-11-19 12:28:41 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000c, 0x0001,
    0x0009, 0x000b, 0x000a, 0x0007, 0x0008, 0x0005, 0x0006
394 2024-11-19 12:28:42 INFO : dT_Srv : Returned from drawTG
395 2024-11-19 12:28:42 INFO : dT_Srv : Sent parameters request
396 2024-11-19 12:28:42 INFO : dT_Srv : Started 10 second timer
397 2024-11-19 12:28:51 INFO : dT_Srv : Data packet received from 0x0006
398 2024-11-19 12:28:51 INFO : dT_Srv : Data packet received from 0x000b
399 2024-11-19 12:28:52 INFO : dT_Srv : Data packet received from 0x000a
400 2024-11-19 12:28:53 INFO : dT_Srv : Data packet received from 0x0007
401 2024-11-19 12:28:53 INFO : dT_Srv : Data packet received from 0x000c
402 2024-11-19 12:28:54 INFO : dT_Srv : Data packet received from 0x0008
403 2024-11-19 12:28:54 INFO : dT_Srv : Data packet received from 0x0009
404 2024-11-19 12:28:54 INFO : dT_Srv : Data packet received from 0x0005
405 2024-11-19 12:29:02 INFO : dT_Srv : Data id=382, iterationId=899 in data/dT.csv
406 2024-11-19 12:29:02 INFO : dT_Srv : Calling drawTG
407 2024-11-19 12:29:02 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0007, 0x0008, 0x000b, 0x000a, 0x000c, 0x0009, 0x0005
408 2024-11-19 12:29:03 INFO : dT_Srv : Returned from drawTG
409 2024-11-19 12:29:03 INFO : dT_Srv : Sent parameters request
410 2024-11-19 12:29:03 INFO : dT_Srv : Started 10 second timer
411 2024-11-19 12:29:12 INFO : dT_Srv : Data packet received from 0x0007
412 2024-11-19 12:29:12 INFO : dT_Srv : Data packet received from 0x0008
413 2024-11-19 12:29:13 INFO : dT_Srv : Data packet received from 0x0009
414 2024-11-19 12:29:13 INFO : dT_Srv : Data packet received from 0x0005
415 2024-11-19 12:29:14 INFO : dT_Srv : Data packet received from 0x000c
416 2024-11-19 12:29:14 INFO : dT_Srv : Data packet received from 0x0006
417 2024-11-19 12:29:14 INFO : dT_Srv : Data packet received from 0x000b
418 2024-11-19 12:29:15 INFO : dT_Srv : Data packet received from 0x000a
419 2024-11-19 12:29:23 INFO : dT_Srv : Data id=966, iterationId=899 in data/dT.csv
420 2024-11-19 12:29:23 INFO : dT_Srv : Calling drawTG
421 2024-11-19 12:29:23 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0008, 0x0005, 0x000c, 0x0006, 0x0009, 0x000b, 0x000a
422 2024-11-19 12:29:24 INFO : dT_Srv : Returned from drawTG
423 2024-11-19 12:29:24 INFO : dT_Srv : Sent parameters request
424 2024-11-19 12:29:24 INFO : dT_Srv : Started 10 second timer
425 2024-11-19 12:29:32 INFO : dT_Srv : Data packet received from 0x0006
426 2024-11-19 12:29:34 INFO : dT_Srv : Data packet received from 0x000a
427 2024-11-19 12:29:34 INFO : dT_Srv : Data packet received from 0x000b
428 2024-11-19 12:29:35 INFO : dT_Srv : Data packet received from 0x0007
429 2024-11-19 12:29:35 INFO : dT_Srv : Data packet received from 0x0008
```

```
430 2024-11-19 12:29:36 INFO : dT_Srv : Data packet received from 0x0009
431 2024-11-19 12:29:36 INFO : dT_Srv : Data packet received from 0x0005
432 2024-11-19 12:29:36 INFO : dT_Srv : Data packet received from 0x000c
433 2024-11-19 12:29:44 INFO : dT_Srv : Data id=574, iterationId=899 in data/dT.csv
434 2024-11-19 12:29:44 INFO : dT_Srv : Calling drawTG
435 2024-11-19 12:29:44 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0007, 0x0008, 0x0005, 0x000a, 0x000b, 0x0009, 0x000c
436 2024-11-19 12:29:45 INFO : dT_Srv : Returned from drawTG
437 2024-11-19 12:29:45 INFO : dT_Srv : Sent parameters request
438 2024-11-19 12:29:45 INFO : dT_Srv : Started 10 second timer
439 2024-11-19 12:29:54 INFO : dT_Srv : Data packet received from 0x0007
440 2024-11-19 12:29:54 INFO : dT_Srv : Data packet received from 0x0008
441 2024-11-19 12:29:54 INFO : dT_Srv : Data packet received from 0x0009
442 2024-11-19 12:29:54 INFO : dT_Srv : Data packet received from 0x000a
443 2024-11-19 12:29:55 INFO : dT_Srv : Data packet received from 0x000b
444 2024-11-19 12:29:55 INFO : dT_Srv : Data packet received from 0x0005
445 2024-11-19 12:29:55 INFO : dT_Srv : Data packet received from 0x000c
446 2024-11-19 12:29:55 INFO : dT_Srv : Data packet received from 0x0006
447 2024-11-19 12:30:05 INFO : dT_Srv : Data id=891, iterationId=899 in data/dT.csv
448 2024-11-19 12:30:05 INFO : dT_Srv : Calling drawTG
449 2024-11-19 12:30:05 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0008, 0x0009, 0x0005, 0x0006, 0x000c, 0x000a, 0x000b
450 2024-11-19 12:30:06 INFO : dT_Srv : Returned from drawTG
451 2024-11-19 12:30:06 INFO : dT_Srv : Sent parameters request
452 2024-11-19 12:30:06 INFO : dT_Srv : Started 10 second timer
453 2024-11-19 12:30:16 INFO : dT_Srv : Data packet received from 0x000a
454 2024-11-19 12:30:16 INFO : dT_Srv : Data packet received from 0x0007
455 2024-11-19 12:30:17 INFO : dT_Srv : Data packet received from 0x0008
456 2024-11-19 12:30:17 INFO : dT_Srv : Data packet received from 0x0009
457 2024-11-19 12:30:18 INFO : dT_Srv : Data packet received from 0x000c
458 2024-11-19 12:30:18 INFO : dT_Srv : Data packet received from 0x0005
459 2024-11-19 12:30:18 INFO : dT_Srv : Data packet received from 0x0006
460 2024-11-19 12:30:19 INFO : dT_Srv : Data packet received from 0x000b
461 2024-11-19 12:30:26 INFO : dT_Srv : Data id=770, iterationId=899 in data/dT.csv
462 2024-11-19 12:30:26 INFO : dT_Srv : Calling drawTG
463 2024-11-19 12:30:26 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0001,
    0x0009, 0x000c, 0x000b, 0x0007, 0x0008, 0x0005, 0x0006
464 2024-11-19 12:30:27 INFO : dT_Srv : Returned from drawTG
465 2024-11-19 12:30:27 INFO : dT_Srv : Sent parameters request
466 2024-11-19 12:30:27 INFO : dT_Srv : Started 10 second timer
467 2024-11-19 12:30:36 INFO : dT_Srv : Data packet received from 0x0005
468 2024-11-19 12:30:37 INFO : dT_Srv : Data packet received from 0x000c
469 2024-11-19 12:30:37 INFO : dT_Srv : Data packet received from 0x0006
470 2024-11-19 12:30:39 INFO : dT_Srv : Data packet received from 0x000b
471 2024-11-19 12:30:39 INFO : dT_Srv : Data packet received from 0x0007
472 2024-11-19 12:30:40 INFO : dT_Srv : Data packet received from 0x000a
473 2024-11-19 12:30:40 INFO : dT_Srv : Data packet received from 0x0008
474 2024-11-19 12:30:40 INFO : dT_Srv : Data packet received from 0x0009
475 2024-11-19 12:30:47 INFO : dT_Srv : Data id=993, iterationId=899 in data/dT.csv
476 2024-11-19 12:30:47 INFO : dT_Srv : Calling drawTG
```

477 2024-11-19 12:30:48 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0005, 0x0001, 0x0006, 0x0007, 0x0009, 0x000c, 0x000b, 0x000a, 0x0008
478 2024-11-19 12:30:49 INFO : dT_Srv : Returned from drawTG
479 2024-11-19 12:30:49 INFO : dT_Srv : Sent parameters request
480 2024-11-19 12:30:49 INFO : dT_Srv : Started 10 second timer
481 2024-11-19 12:30:59 INFO : dT_Srv : Data packet received from 0x000a
482 2024-11-19 12:30:59 INFO : dT_Srv : Data packet received from 0x000b
483 2024-11-19 12:30:59 INFO : dT_Srv : Data packet received from 0x0007
484 2024-11-19 12:30:59 INFO : dT_Srv : Data packet received from 0x0008
485 2024-11-19 12:30:59 INFO : dT_Srv : Data packet received from 0x0009
486 2024-11-19 12:31:00 INFO : dT_Srv : Data packet received from 0x000c
487 2024-11-19 12:31:00 INFO : dT_Srv : Data packet received from 0x0006
488 2024-11-19 12:31:09 INFO : dT_Srv : Data id=971, iterationId=899 in data/dT.csv
489 2024-11-19 12:31:09 INFO : dT_Srv : Calling drawTG
490 2024-11-19 12:31:09 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x000b, 0x0009, 0x000c, 0x0007, 0x0008, 0x0005, 0x0006, 0x0001
491 2024-11-19 12:31:10 INFO : dT_Srv : Returned from drawTG
492 2024-11-19 12:31:10 INFO : dT_Srv : Sent parameters request
493 2024-11-19 12:31:10 INFO : dT_Srv : Started 10 second timer
494 2024-11-19 12:31:19 INFO : dT_Srv : Data packet received from 0x0006
495 2024-11-19 12:31:21 INFO : dT_Srv : Data packet received from 0x000b
496 2024-11-19 12:31:21 INFO : dT_Srv : Data packet received from 0x000c
497 2024-11-19 12:31:21 INFO : dT_Srv : Data packet received from 0x0007
498 2024-11-19 12:31:22 INFO : dT_Srv : Data packet received from 0x000a
499 2024-11-19 12:31:22 INFO : dT_Srv : Data packet received from 0x0008
500 2024-11-19 12:31:22 INFO : dT_Srv : Data packet received from 0x0009
501 2024-11-19 12:31:30 INFO : dT_Srv : Data id=800, iterationId=899 in data/dT.csv
502 2024-11-19 12:31:30 INFO : dT_Srv : Calling drawTG
503 2024-11-19 12:31:30 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0007, 0x0008, 0x000b, 0x000c, 0x000a, 0x0009
504 2024-11-19 12:31:31 INFO : dT_Srv : Returned from drawTG
505 2024-11-19 12:31:31 INFO : dT_Srv : Sent parameters request
506 2024-11-19 12:31:31 INFO : dT_Srv : Started 10 second timer
507 2024-11-19 12:31:40 INFO : dT_Srv : Data packet received from 0x000a
508 2024-11-19 12:31:40 INFO : dT_Srv : Data packet received from 0x0008
509 2024-11-19 12:31:40 INFO : dT_Srv : Data packet received from 0x0009
510 2024-11-19 12:31:41 INFO : dT_Srv : Data packet received from 0x000c
511 2024-11-19 12:31:42 INFO : dT_Srv : Data packet received from 0x0006
512 2024-11-19 12:31:43 INFO : dT_Srv : Data packet received from 0x000b
513 2024-11-19 12:31:44 INFO : dT_Srv : Data packet received from 0x0007
514 2024-11-19 12:31:51 INFO : dT_Srv : Data id=209, iterationId=899 in data/dT.csv
515 2024-11-19 12:31:51 INFO : dT_Srv : Calling drawTG
516 2024-11-19 12:31:51 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0001, 0x0009, 0x000c, 0x000b, 0x0008, 0x0006, 0x0007
517 2024-11-19 12:31:52 INFO : dT_Srv : Returned from drawTG
518 2024-11-19 12:31:52 INFO : dT_Srv : Sent parameters request
519 2024-11-19 12:31:52 INFO : dT_Srv : Started 10 second timer
520 2024-11-19 12:32:03 INFO : dT_Srv : Data packet received from 0x0007
521 2024-11-19 12:32:03 INFO : dT_Srv : Data packet received from 0x000a
522 2024-11-19 12:32:03 INFO : dT_Srv : Data packet received from 0x000b

```
523 2024-11-19 12:32:04 INFO : dT_Srv : Data packet received from 0x000c
524 2024-11-19 12:32:04 INFO : dT_Srv : Data packet received from 0x0006
525 2024-11-19 12:32:05 INFO : dT_Srv : Data packet received from 0x0008
526 2024-11-19 12:32:12 INFO : dT_Srv : Data id=336, iterationId=899 in data/dT.csv
527 2024-11-19 12:32:12 INFO : dT_Srv : Calling drawTG
528 2024-11-19 12:32:12 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x000c, 0x0006, 0x0008, 0x000a, 0x000b
529 2024-11-19 12:32:13 INFO : dT_Srv : Returned from drawTG
530 2024-11-19 12:32:13 INFO : dT_Srv : Sent parameters request
531 2024-11-19 12:32:13 INFO : dT_Srv : Started 10 second timer
532 2024-11-19 12:32:23 INFO : dT_Srv : Data packet received from 0x000c
533 2024-11-19 12:32:23 INFO : dT_Srv : Data packet received from 0x0006
534 2024-11-19 12:32:23 INFO : dT_Srv : Data packet received from 0x0008
535 2024-11-19 12:32:25 INFO : dT_Srv : Data packet received from 0x000b
536 2024-11-19 12:32:25 INFO : dT_Srv : Data packet received from 0x0007
537 2024-11-19 12:32:26 INFO : dT_Srv : Data packet received from 0x000a
538 2024-11-19 12:32:33 INFO : dT_Srv : Data id=123, iterationId=899 in data/dT.csv
539 2024-11-19 12:32:33 INFO : dT_Srv : Calling drawTG
540 2024-11-19 12:32:33 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000c, 0x0001,
    0x000b, 0x0007, 0x000a, 0x0006, 0x0008
541 2024-11-19 12:32:34 INFO : dT_Srv : Returned from drawTG
542 2024-11-19 12:32:34 INFO : dT_Srv : Sent parameters request
543 2024-11-19 12:32:34 INFO : dT_Srv : Started 10 second timer
544 2024-11-19 12:32:43 INFO : dT_Srv : Data packet received from 0x0006
545 2024-11-19 12:32:45 INFO : dT_Srv : Data packet received from 0x000b
546 2024-11-19 12:32:45 INFO : dT_Srv : Data packet received from 0x000a
547 2024-11-19 12:32:46 INFO : dT_Srv : Data packet received from 0x0007
548 2024-11-19 12:32:46 INFO : dT_Srv : Data packet received from 0x000c
549 2024-11-19 12:32:46 INFO : dT_Srv : Data packet received from 0x0008
550 2024-11-19 12:32:54 INFO : dT_Srv : Data id=4, iterationId=899 in data/dT.csv
551 2024-11-19 12:32:54 INFO : dT_Srv : Calling drawTG
552 2024-11-19 12:32:55 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0007, 0x000b, 0x000a, 0x000c, 0x0008
553 2024-11-19 12:32:56 INFO : dT_Srv : Returned from drawTG
554 2024-11-19 12:32:56 INFO : dT_Srv : Sent parameters request
555 2024-11-19 12:32:56 INFO : dT_Srv : Started 10 second timer
556 2024-11-19 12:33:05 INFO : dT_Srv : Data packet received from 0x0008
557 2024-11-19 12:33:06 INFO : dT_Srv : Data packet received from 0x0006
558 2024-11-19 12:33:08 INFO : dT_Srv : Data packet received from 0x000a
559 2024-11-19 12:33:08 INFO : dT_Srv : Data packet received from 0x000b
560 2024-11-19 12:33:08 INFO : dT_Srv : Data packet received from 0x0007
561 2024-11-19 12:33:08 INFO : dT_Srv : Data packet received from 0x000c
562 2024-11-19 12:33:09 INFO : dT_Srv : Data packet received from 0x000a
563 2024-11-19 12:33:16 INFO : dT_Srv : Data id=626, iterationId=899 in data/dT.csv
564 2024-11-19 12:33:16 INFO : dT_Srv : Calling drawTG
565 2024-11-19 12:33:16 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001,
    0x0006, 0x0007, 0x000a, 0x000b, 0x000c
566 2024-11-19 12:33:17 INFO : dT_Srv : Returned from drawTG
567 2024-11-19 12:33:17 INFO : dT_Srv : Sent parameters request
568 2024-11-19 12:33:17 INFO : dT_Srv : Started 10 second timer
```

```
569 2024-11-19 12:33:26 INFO : dT_Srv : Data packet received from 0x000a
570 2024-11-19 12:33:27 INFO : dT_Srv : Data packet received from 0x0007
571 2024-11-19 12:33:27 INFO : dT_Srv : Data packet received from 0x000c
572 2024-11-19 12:33:28 INFO : dT_Srv : Data packet received from 0x0008
573 2024-11-19 12:33:29 INFO : dT_Srv : Data packet received from 0x0006
574 2024-11-19 12:33:30 INFO : dT_Srv : Data packet received from 0x000b
575 2024-11-19 12:33:37 INFO : dT_Srv : Data id=981, iterationId=899 in data/dT.csv
576 2024-11-19 12:33:37 INFO : dT_Srv : Calling drawTG
577 2024-11-19 12:33:37 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0001,
    0x000c, 0x000b, 0x0007, 0x0008, 0x0006
578 2024-11-19 12:33:38 INFO : dT_Srv : Returned from drawTG
579 2024-11-19 12:33:38 INFO : dT_Srv : Sent parameters request
580 2024-11-19 12:33:38 INFO : dT_Srv : Started 10 second timer
581 2024-11-19 12:33:47 INFO : dT_Srv : Data packet received from 0x0008
582 2024-11-19 12:33:48 INFO : dT_Srv : Data packet received from 0x0006
583 2024-11-19 12:33:49 INFO : dT_Srv : Data packet received from 0x000b
584 2024-11-19 12:33:49 INFO : dT_Srv : Data packet received from 0x000a
585 2024-11-19 12:33:50 INFO : dT_Srv : Data packet received from 0x000c
586 2024-11-19 12:33:50 INFO : dT_Srv : Data packet received from 0x0007
587 2024-11-19 12:33:58 INFO : dT_Srv : Data id=762, iterationId=899 in data/dT.csv
588 2024-11-19 12:33:58 INFO : dT_Srv : Calling drawTG
589 2024-11-19 12:33:58 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001,
    0x0006, 0x0007, 0x000b, 0x000a, 0x000c
590 2024-11-19 12:33:59 INFO : dT_Srv : Returned from drawTG
591 2024-11-19 12:33:59 INFO : dT_Srv : Sent parameters request
592 2024-11-19 12:33:59 INFO : dT_Srv : Started 10 second timer
593 2024-11-19 12:34:09 INFO : dT_Srv : Data packet received from 0x0007
594 2024-11-19 12:34:09 INFO : dT_Srv : Data packet received from 0x000a
595 2024-11-19 12:34:09 INFO : dT_Srv : Data packet received from 0x000c
596 2024-11-19 12:34:09 INFO : dT_Srv : Data packet received from 0x0008
597 2024-11-19 12:34:11 INFO : dT_Srv : Data packet received from 0x0006
598 2024-11-19 12:34:11 INFO : dT_Srv : Data packet received from 0x000b
599 2024-11-19 12:34:19 INFO : dT_Srv : Data id=859, iterationId=899 in data/dT.csv
600 2024-11-19 12:34:19 INFO : dT_Srv : Calling drawTG
601 2024-11-19 12:34:19 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x000c, 0x0008, 0x0006, 0x000a, 0x000b
602 2024-11-19 12:34:20 INFO : dT_Srv : Returned from drawTG
603 2024-11-19 12:34:20 INFO : dT_Srv : Sent parameters request
604 2024-11-19 12:34:20 INFO : dT_Srv : Started 10 second timer
605 2024-11-19 12:34:29 INFO : dT_Srv : Data packet received from 0x0006
606 2024-11-19 12:34:30 INFO : dT_Srv : Data packet received from 0x000c
607 2024-11-19 12:34:30 INFO : dT_Srv : Data packet received from 0x000b
608 2024-11-19 12:34:31 INFO : dT_Srv : Data packet received from 0x000a
609 2024-11-19 12:34:32 INFO : dT_Srv : Data packet received from 0x0009
610 2024-11-19 12:34:32 INFO : dT_Srv : Data packet received from 0x0007
611 2024-11-19 12:34:32 INFO : dT_Srv : Data packet received from 0x0008
612 2024-11-19 12:34:40 INFO : dT_Srv : Data id=60, iterationId=899 in data/dT.csv
613 2024-11-19 12:34:40 INFO : dT_Srv : Calling drawTG
614 2024-11-19 12:34:40 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001,
    0x0007, 0x0008, 0x000c, 0x000b, 0x000a, 0x0009
```

```
615 2024-11-19 12:34:41 INFO : dT_Srv : Returned from drawTG
616 2024-11-19 12:34:41 INFO : dT_Srv : Sent parameters request
617 2024-11-19 12:34:41 INFO : dT_Srv : Started 10 second timer
618 2024-11-19 12:34:50 INFO : dT_Srv : Data packet received from 0x0007
619 2024-11-19 12:34:51 INFO : dT_Srv : Data packet received from 0x0008
620 2024-11-19 12:34:52 INFO : dT_Srv : Data packet received from 0x000c
621 2024-11-19 12:34:52 INFO : dT_Srv : Data packet received from 0x0006
622 2024-11-19 12:34:53 INFO : dT_Srv : Data packet received from 0x000b
623 2024-11-19 12:34:54 INFO : dT_Srv : Data packet received from 0x000a
624 2024-11-19 12:34:54 INFO : dT_Srv : Data packet received from 0x0009
625 2024-11-19 12:35:01 INFO : dT_Srv : Data id=48, iterationId=899 in data/dT.csv
626 2024-11-19 12:35:01 INFO : dT_Srv : Calling drawTG
627 2024-11-19 12:35:01 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001,
    0x0008, 0x000c, 0x0006, 0x000b, 0x000a, 0x0009
628 2024-11-19 12:35:02 INFO : dT_Srv : Returned from drawTG
629 2024-11-19 12:35:02 INFO : dT_Srv : Sent parameters request
630 2024-11-19 12:35:02 INFO : dT_Srv : Started 10 second timer
631 2024-11-19 12:35:13 INFO : dT_Srv : Data packet received from 0x000a
632 2024-11-19 12:35:13 INFO : dT_Srv : Data packet received from 0x000d
633 2024-11-19 12:35:13 INFO : dT_Srv : Data packet received from 0x000b
634 2024-11-19 12:35:13 INFO : dT_Srv : Data packet received from 0x0009
635 2024-11-19 12:35:14 INFO : dT_Srv : Data packet received from 0x0007
636 2024-11-19 12:35:14 INFO : dT_Srv : Data packet received from 0x0008
637 2024-11-19 12:35:15 INFO : dT_Srv : Data packet received from 0x000c
638 2024-11-19 12:35:15 INFO : dT_Srv : Data packet received from 0x0006
639 2024-11-19 12:35:22 INFO : dT_Srv : Data id=767, iterationId=899 in data/dT.csv
640 2024-11-19 12:35:22 INFO : dT_Srv : Calling drawTG
641 2024-11-19 12:35:23 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0001,
    0x000b, 0x000d, 0x000c, 0x0009, 0x0008, 0x0006, 0x0007
642 2024-11-19 12:35:24 INFO : dT_Srv : Returned from drawTG
643 2024-11-19 12:35:24 INFO : dT_Srv : Sent parameters request
644 2024-11-19 12:35:24 INFO : dT_Srv : Started 10 second timer
645 2024-11-19 12:35:33 INFO : dT_Srv : Data packet received from 0x0008
646 2024-11-19 12:35:34 INFO : dT_Srv : Data packet received from 0x000c
647 2024-11-19 12:35:34 INFO : dT_Srv : Data packet received from 0x0006
648 2024-11-19 12:35:35 INFO : dT_Srv : Data packet received from 0x000b
649 2024-11-19 12:35:35 INFO : dT_Srv : Data packet received from 0x000a
650 2024-11-19 12:35:36 INFO : dT_Srv : Data packet received from 0x000d
651 2024-11-19 12:35:36 INFO : dT_Srv : Data packet received from 0x0009
652 2024-11-19 12:35:36 INFO : dT_Srv : Data packet received from 0x0007
653 2024-11-19 12:35:44 INFO : dT_Srv : Data id=938, iterationId=899 in data/dT.csv
654 2024-11-19 12:35:44 INFO : dT_Srv : Calling drawTG
655 2024-11-19 12:35:44 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001,
    0x0006, 0x0007, 0x000c, 0x000b, 0x000a, 0x000d, 0x0009
656 2024-11-19 12:35:45 INFO : dT_Srv : Returned from drawTG
657 2024-11-19 12:35:45 INFO : dT_Srv : Sent parameters request
658 2024-11-19 12:35:45 INFO : dT_Srv : Started 10 second timer
659 2024-11-19 12:35:54 INFO : dT_Srv : Data packet received from 0x000b
660 2024-11-19 12:35:54 INFO : dT_Srv : Data packet received from 0x000d
661 2024-11-19 12:35:55 INFO : dT_Srv : Data packet received from 0x0009
```



```
662 2024-11-19 12:35:55 INFO : dT_Srv : Data packet received from 0x0007
663 2024-11-19 12:35:56 INFO : dT_Srv : Data packet received from 0x000a
664 2024-11-19 12:35:56 INFO : dT_Srv : Data packet received from 0x0008
665 2024-11-19 12:35:57 INFO : dT_Srv : Data packet received from 0x000c
666 2024-11-19 12:35:57 INFO : dT_Srv : Data packet received from 0x0006
667 2024-11-19 12:36:05 INFO : dT_Srv : Data id=888, iterationId=899 in data/dT.csv
668 2024-11-19 12:36:05 INFO : dT_Srv : Calling drawTG
669 2024-11-19 12:36:05 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000b, 0x000d,
    0x000a, 0x000c, 0x0009, 0x0001, 0x0007, 0x0008, 0x0006
670 2024-11-19 12:36:06 INFO : dT_Srv : Returned from drawTG
671 2024-11-19 12:36:06 INFO : dT_Srv : Sent parameters request
672 2024-11-19 12:36:06 INFO : dT_Srv : Started 10 second timer
673 2024-11-19 12:36:15 INFO : dT_Srv : Data packet received from 0x000a
674 2024-11-19 12:36:15 INFO : dT_Srv : Data packet received from 0x0008
675 2024-11-19 12:36:15 INFO : dT_Srv : Data packet received from 0x000c
676 2024-11-19 12:36:16 INFO : dT_Srv : Data packet received from 0x0006
677 2024-11-19 12:36:16 INFO : dT_Srv : Data packet received from 0x000b
678 2024-11-19 12:36:17 INFO : dT_Srv : Data packet received from 0x000d
679 2024-11-19 12:36:18 INFO : dT_Srv : Data packet received from 0x0009
680 2024-11-19 12:36:18 INFO : dT_Srv : Data packet received from 0x0007
681 2024-11-19 12:36:26 INFO : dT_Srv : Data id=358, iterationId=899 in data/dT.csv
682 2024-11-19 12:36:26 INFO : dT_Srv : Calling drawTG
683 2024-11-19 12:36:26 INFO : drawTG : Plotted NetTopGraph with nodes: 0x000a, 0x0001,
    0x000c, 0x000b, 0x0008, 0x0006, 0x0007, 0x000d, 0x0009
684 2024-11-19 12:36:27 INFO : dT_Srv : Returned from drawTG
685 2024-11-19 12:36:27 INFO : dT_Srv : Sent parameters request
686 2024-11-19 12:36:27 INFO : dT_Srv : Started 10 second timer
687 2024-11-19 12:36:37 INFO : dT_Srv : Data packet received from 0x0009
688 2024-11-19 12:36:37 INFO : dT_Srv : Data packet received from 0x0007
689 2024-11-19 12:36:37 INFO : dT_Srv : Data packet received from 0x0008
690 2024-11-19 12:36:38 INFO : dT_Srv : Data packet received from 0x000c
691 2024-11-19 12:36:39 INFO : dT_Srv : Data packet received from 0x0006
692 2024-11-19 12:36:40 INFO : dT_Srv : Data packet received from 0x000a
693 2024-11-19 12:36:40 INFO : dT_Srv : Data packet received from 0x000d
694 2024-11-19 12:36:40 INFO : dT_Srv : Data packet received from 0x000b
695 2024-11-19 12:36:47 INFO : dT_Srv : Data id=837, iterationId=899 in data/dT.csv
696 2024-11-19 12:36:47 INFO : dT_Srv : Calling drawTG
697 2024-11-19 12:36:47 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0009, 0x0001,
    0x0007, 0x0008, 0x000c, 0x0006, 0x000b, 0x000a, 0x000d
698 2024-11-19 12:36:48 INFO : dT_Srv : Returned from drawTG
699 2024-11-19 12:36:48 INFO : dT_Srv : Sent parameters request
700 2024-11-19 12:36:48 INFO : dT_Srv : Started 10 second timer
701 2024-11-19 12:36:57 INFO : dT_Srv : Data packet received from 0x0006
702 2024-11-19 12:37:00 INFO : dT_Srv : Data packet received from 0x0007
703 2024-11-19 12:37:00 INFO : dT_Srv : Data packet received from 0x000a
704 2024-11-19 12:37:00 INFO : dT_Srv : Data packet received from 0x000b
705 2024-11-19 12:37:00 INFO : dT_Srv : Data packet received from 0x0008
706 2024-11-19 12:37:01 INFO : dT_Srv : Data packet received from 0x000c
707 2024-11-19 12:37:08 INFO : dT_Srv : Data id=315, iterationId=899 in data/dT.csv
708 2024-11-19 12:37:08 INFO : dT_Srv : Calling drawTG
```

709 2024-11-19 12:37:08 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0007, 0x0008, 0x000c, 0x000a, 0x000b, 0x000d
710 2024-11-19 12:37:09 INFO : dT_Srv : Returned from drawTG
711 2024-11-19 12:37:09 INFO : dT_Srv : Sent parameters request
712 2024-11-19 12:37:09 INFO : dT_Srv : Started 10 second timer
713 2024-11-19 12:37:18 INFO : dT_Srv : Data packet received from 0x0007
714 2024-11-19 12:37:19 INFO : dT_Srv : Data packet received from 0x000b
715 2024-11-19 12:37:19 INFO : dT_Srv : Data packet received from 0x0008
716 2024-11-19 12:37:20 INFO : dT_Srv : Data packet received from 0x000c
717 2024-11-19 12:37:20 INFO : dT_Srv : Data packet received from 0x000d
718 2024-11-19 12:37:20 INFO : dT_Srv : Data packet received from 0x0006
719 2024-11-19 12:37:29 INFO : dT_Srv : Data id=255, iterationId=899 in data/dT.csv
720 2024-11-19 12:37:29 INFO : dT_Srv : Calling drawTG
721 2024-11-19 12:37:30 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0007, 0x0001, 0x0008, 0x000c, 0x0006, 0x000b, 0x000a, 0x000d
722 2024-11-19 12:37:31 INFO : dT_Srv : Returned from drawTG
723 2024-11-19 12:37:31 INFO : dT_Srv : Sent parameters request
724 2024-11-19 12:37:31 INFO : dT_Srv : Started 10 second timer
725 2024-11-19 12:37:39 INFO : dT_Srv : Data packet received from 0x0006
726 2024-11-19 12:37:39 INFO : dT_Srv : Data packet received from 0x0008
727 2024-11-19 12:37:41 INFO : dT_Srv : Data packet received from 0x0007
728 2024-11-19 12:37:43 INFO : dT_Srv : Data packet received from 0x000c
729 2024-11-19 12:37:43 INFO : dT_Srv : Data packet received from 0x000d
730 2024-11-19 12:37:51 INFO : dT_Srv : Data id=729, iterationId=899 in data/dT.csv
731 2024-11-19 12:37:51 INFO : dT_Srv : Calling drawTG
732 2024-11-19 12:37:51 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0008, 0x0007, 0x000c, 0x000d
733 2024-11-19 12:37:52 INFO : dT_Srv : Returned from drawTG
734 2024-11-19 12:37:52 INFO : dT_Srv : Sent parameters request
735 2024-11-19 12:37:52 INFO : dT_Srv : Started 10 second timer
736 2024-11-19 12:38:02 INFO : dT_Srv : Data packet received from 0x0006
737 2024-11-19 12:38:02 INFO : dT_Srv : Data packet received from 0x0008
738 2024-11-19 12:38:04 INFO : dT_Srv : Data packet received from 0x0007
739 2024-11-19 12:38:12 INFO : dT_Srv : Data id=705, iterationId=899 in data/dT.csv
740 2024-11-19 12:38:12 INFO : dT_Srv : Calling drawTG
741 2024-11-19 12:38:12 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0008, 0x0007
742 2024-11-19 12:38:13 INFO : dT_Srv : Returned from drawTG
743 2024-11-19 12:38:13 INFO : dT_Srv : Sent parameters request
744 2024-11-19 12:38:13 INFO : dT_Srv : Started 10 second timer
745 2024-11-19 12:38:22 INFO : dT_Srv : Data packet received from 0x0006
746 2024-11-19 12:38:23 INFO : dT_Srv : Data packet received from 0x0007
747 2024-11-19 12:38:25 INFO : dT_Srv : Data packet received from 0x0008
748 2024-11-19 12:38:33 INFO : dT_Srv : Data id=435, iterationId=899 in data/dT.csv
749 2024-11-19 12:38:33 INFO : dT_Srv : Calling drawTG
750 2024-11-19 12:38:33 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0006, 0x0001, 0x0007, 0x0008
751 2024-11-19 12:38:34 INFO : dT_Srv : Returned from drawTG
752 2024-11-19 12:38:34 INFO : dT_Srv : Sent parameters request
753 2024-11-19 12:38:34 INFO : dT_Srv : Started 10 second timer

```

754 2024-11-19 12:38:44 INFO : dT_Srv : Data packet received from 0x0008
755 2024-11-19 12:38:54 INFO : dT_Srv : Data id=404, iterationId=899 in data/dT.csv
756 2024-11-19 12:38:54 INFO : dT_Srv : Calling drawTG
757 2024-11-19 12:38:54 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0008, 0x0001,
    0x0007
758 2024-11-19 12:38:55 INFO : dT_Srv : Returned from drawTG
759 2024-11-19 12:38:55 INFO : dT_Srv : Sent parameters request
760 2024-11-19 12:38:55 INFO : dT_Srv : Started 10 second timer
761 2024-11-19 12:39:15 WARNING : dT_Srv : No data received from network nodes
762 2024-11-19 12:39:15 INFO : dT_Srv : Data id=434, iterationId=899 in data/dT.csv
763 2024-11-19 12:39:15 INFO : dT_Srv : Calling drawTG
764 2024-11-19 12:39:15 INFO : drawTG : Plotted NetTopGraph with nodes: 0x0001
765 2024-11-19 12:39:16 INFO : dT_Srv : Returned from drawTG
766 2024-11-19 12:39:16 INFO : dT_Srv : Sent parameters request
767 2024-11-19 12:39:16 INFO : dT_Srv : Started 10 second timer
768 2024-11-19 12:39:18 WARNING : dT_Srv : Execution interrupted by user
769 2024-11-19 12:39:18 INFO : <module> : Returned from [1] - dT_Srv (continuous req)

```

Listing A.1: Sezioni del logfile interessanti per l'analisi dell'esperimento 1

A.3 File *discoverTopology.csv*

Nota: sono qui inseriti solo i dati utili per l'analisi dell'esperimento

```

1 id,iterationId,continuous,timestamp,service,node,n_near_nodes,near_node_address,
  rssi
2 556,899,True,2024-11-19 12:17:45,discoverTopology_Service,0x0005,2,0x0001,-85
3 556,899,True,2024-11-19 12:17:45,discoverTopology_Service,0x0005,2,0x0006,-93
4 556,899,True,2024-11-19 12:17:45,discoverTopology_Service,0x0006,2,0x0001,-83
5 556,899,True,2024-11-19 12:17:45,discoverTopology_Service,0x0006,2,0x0005,-86
6 66,899,True,2024-11-19 12:18:06,discoverTopology_Service,0x0005,2,0x0001,-86
7 66,899,True,2024-11-19 12:18:06,discoverTopology_Service,0x0005,2,0x0006,-79
8 66,899,True,2024-11-19 12:18:06,discoverTopology_Service,0x0006,2,0x0001,-82
9 66,899,True,2024-11-19 12:18:06,discoverTopology_Service,0x0006,2,0x0005,-84
10 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0007,3,0x0001,-72
11 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0007,3,0x0005,-70
12 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0007,3,0x0006,-69
13 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0005,3,0x0001,-84
14 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0005,3,0x0007,-78
15 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0005,3,0x0006,-81
16 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0006,3,0x0001,-79
17 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0006,3,0x0007,-76
18 266,899,True,2024-11-19 12:18:27,discoverTopology_Service,0x0006,3,0x0005,-84
19 -,,-,-,-,-,-,-,-
20 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0008,4,0x0001,-77
21 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0008,4,0x0009,-92
22 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0008,4,0x0005,-89
23 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0008,4,0x0007,-95
24 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x0001,-85

```

25 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x0008,-83
26 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x0005,-95
27 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x0006,-94
28 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x000b,-77
29 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x000a,-75
30 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x0007,-97
31 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0009,8,0x000c,-54
32 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0005,5,0x0001,-84
33 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0005,5,0x0009,-97
34 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0005,5,0x0008,-92
35 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0005,5,0x0006,-77
36 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0005,5,0x0007,-81
37 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0006,5,0x0001,-90
38 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0006,5,0x0009,-103
39 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0006,5,0x0008,-86
40 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0006,5,0x0005,-81
41 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0006,5,0x0007,-77
42 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000b,4,0x0001,-95
43 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000b,4,0x0009,-76
44 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000b,4,0x000a,-83
45 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000b,4,0x000c,-71
46 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000a,3,0x0009,-72
47 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000a,3,0x000b,-92
48 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000a,3,0x000c,-72
49 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0007,4,0x0001,-77
50 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0007,4,0x0008,-81
51 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0007,4,0x0005,-70
52 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x0007,4,0x0006,-73
53 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000c,5,0x0001,-94
54 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000c,5,0x0009,-63
55 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000c,5,0x0008,-93
56 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000c,5,0x000b,-71
57 487,899,True,2024-11-19 12:27:37,discoverTopology_Service,0x000c,5,0x000a,-68
58 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000a,3,0x000b,-91
59 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000a,3,0x000c,-75
60 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000a,3,0x0009,-87
61 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000b,3,0x000a,-84
62 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000b,3,0x000c,-71
63 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000b,3,0x0009,-77
64 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x0007,5,0x0001,-73
65 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x0007,5,0x0009,-92
66 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x0007,5,0x0008,-80
67 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x0007,5,0x0005,-65
68 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x0007,5,0x0006,-72
69 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x0001,-80
70 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x000b,-66
71 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x000a,-69
72 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x0009,-62
73 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x0008,-89
74 424,899,True,2024-11-19 12:27:58,discoverTopology_Service,0x000c,6,0x0006,-96

75 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x0001, -84
76 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x0007, -87
77 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x000c, -98
78 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x0009, -90
79 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x0005, -91
80 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0008,6,0x0006, -74
81 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x0001, -76
82 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x000b, -70
83 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x000a, -62
84 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x000c, -49
85 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x0008, -85
86 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x0005, -95
87 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0009,7,0x0006, -95
88 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0005,4,0x0001, -86
89 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0005,4,0x0007, -79
90 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0005,4,0x0008, -92
91 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0005,4,0x0006, -77
92 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0006,5,0x0001, -78
93 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0006,5,0x0007, -77
94 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0006,5,0x0008, -74
95 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0006,5,0x0009, -92
96 424,899, True ,2024-11-19 12:27:58, discoverTopology_Service ,0x0006,5,0x0005, -81
97 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x000c,4,0x0001, -85
98 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x000c,4,0x0009, -64
99 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x000c,4,0x000b, -82
100 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x000c,4,0x000a, -59
101 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0007,5,0x0001, -78
102 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0007,5,0x0008, -80
103 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0007,5,0x0009, -91
104 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0007,5,0x0005, -71
105 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0007,5,0x0006, -65
106 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0008,5,0x0001, -84
107 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0008,5,0x0007, -93
108 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0008,5,0x0009, -85
109 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0008,5,0x0005, -92
110 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0008,5,0x0006, -73
111 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x0001, -83
112 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x000c, -70
113 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x0008, -91
114 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x0005, -92
115 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x0006, -93
116 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x000b, -77
117 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0009,7,0x000a, -73
118 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0005,5,0x0001, -83
119 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0005,5,0x0007, -81
120 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0005,5,0x0008, -89
121 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0005,5,0x0009, -97
122 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0005,5,0x0006, -78
123 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0006,5,0x0001, -86
124 741,899, True ,2024-11-19 12:28:40, discoverTopology_Service ,0x0006,5,0x0007, -76

125 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x0006,5,0x0008,-74
126 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x0006,5,0x0009,-97
127 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x0006,5,0x0005,-81
128 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000b,3,0x000c,-83
129 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000b,3,0x0009,-80
130 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000b,3,0x000a,-92
131 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000a,3,0x000c,-58
132 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000a,3,0x0009,-73
133 741,899,True,2024-11-19 12:28:40,discoverTopology_Service,0x000a,3,0x000b,-83
134 -,,-,-,-,-,-,-,-
135 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000a,4,0x0001,-94
136 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000a,4,0x0009,-83
137 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000a,4,0x000c,-92
138 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000a,4,0x000b,-84
139 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0008,4,0x0001,-79
140 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0008,4,0x0009,-89
141 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0008,4,0x0006,-91
142 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0008,4,0x0007,-95
143 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x0001,-76
144 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x000a,-72
145 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x0008,-86
146 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x000c,-85
147 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x0006,-94
148 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x000b,-72
149 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0009,7,0x0007,-92
150 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x000a,-88
151 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x0009,-87
152 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x0008,-94
153 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x0006,-95
154 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x000b,-87
155 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000c,6,0x0007,-91
156 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0006,5,0x0001,-88
157 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0006,5,0x0009,-98
158 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0006,5,0x0008,-88
159 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0006,5,0x000c,-99
160 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0006,5,0x0007,-77
161 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000b,3,0x000a,-86
162 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000b,3,0x0009,-79
163 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x000b,3,0x000c,-93
164 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0007,5,0x0001,-77
165 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0007,5,0x0008,-79
166 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0007,5,0x0009,-88
167 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0007,5,0x000c,-85
168 209,899,True,2024-11-19 12:31:51,discoverTopology_Service,0x0007,5,0x0006,-64
169 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0007,4,0x0001,-71
170 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0007,4,0x000c,-89
171 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0007,4,0x0006,-65
172 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0007,4,0x0008,-80
173 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000a,2,0x000b,-83
174 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000a,2,0x000c,-94

175 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000b,2,0x000a,-83
176 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000b,2,0x000c,-91
177 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000c,4,0x0001,-84
178 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000c,4,0x0007,-91
179 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000c,4,0x000a,-95
180 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x000c,4,0x0006,-93
181 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0006,3,0x0001,-82
182 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0006,3,0x0007,-80
183 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0006,3,0x0008,-83
184 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0008,3,0x0001,-82
185 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0008,3,0x0007,-92
186 336,899,True,2024-11-19 12:32:12,discoverTopology_Service,0x0008,3,0x0006,-90
187 -,,-,-,-,-,-,-,-
188 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000b,3,0x000d,-72
189 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000b,3,0x000a,-91
190 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000b,3,0x000c,-88
191 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000d,4,0x000b,-72
192 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000d,4,0x0009,-70
193 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000d,4,0x000a,-82
194 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000d,4,0x000c,-74
195 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x0001,-85
196 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x000b,-77
197 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x000d,-66
198 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x0007,-95
199 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x000a,-79
200 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x0008,-84
201 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x000c,-90
202 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0009,8,0x0006,-91
203 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0007,4,0x0001,-77
204 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0007,4,0x0008,-79
205 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0007,4,0x000c,-85
206 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0007,4,0x0006,-64
207 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000a,4,0x0001,-93
208 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000a,4,0x000b,-85
209 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000a,4,0x000d,-94
210 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000a,4,0x000c,-95
211 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0008,3,0x0001,-78
212 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0008,3,0x0007,-92
213 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0008,3,0x0006,-85
214 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x000b,-87
215 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x000d,-69
216 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x0007,-99
217 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x000a,-88
218 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x0008,-95
219 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x000c,6,0x0006,-105
220 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0006,3,0x0001,-85
221 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0006,3,0x0007,-79
222 888,899,True,2024-11-19 12:36:05,discoverTopology_Service,0x0006,3,0x0008,-74
223 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000a,3,0x0001,-91
224 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000a,3,0x000c,-92

225 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000a,3,0x000b,-89
226 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0008,3,0x0001,-80
227 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0008,3,0x0006,-73
228 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0008,3,0x0007,-92
229 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000c,5,0x000a,-88
230 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000c,5,0x0008,-95
231 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000c,5,0x0006,-99
232 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000c,5,0x000d,-86
233 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000c,5,0x0007,-91
234 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0006,3,0x0001,-86
235 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0006,3,0x0008,-85
236 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0006,3,0x0007,-77
237 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000b,2,0x000a,-88
238 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000b,2,0x000c,-92
239 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x000d,1,0x000c,-71
240 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x0001,-79
241 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x000a,-79
242 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x0008,-88
243 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x000c,-86
244 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x0006,-90
245 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x000b,-72
246 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0009,7,0x0007,-96
247 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0007,4,0x0001,-78
248 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0007,4,0x0008,-79
249 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0007,4,0x000c,-85
250 358,899,True,2024-11-19 12:36:26,discoverTopology_Service,0x0007,4,0x0006,-64
251 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x0001,-96
252 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x0007,-95
253 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x0008,-86
254 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x000c,-86
255 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x0006,-92
256 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x000b,-79
257 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x000a,-63
258 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0009,8,0x000d,-96
259 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0007,4,0x0001,-89
260 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0007,4,0x0008,-81
261 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0007,4,0x000c,-87
262 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0007,4,0x0006,-64
263 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0008,3,0x0001,-75
264 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0008,3,0x0007,-90
265 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0008,3,0x0006,-73
266 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x0001,-102
267 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x0007,-92
268 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x0008,-93
269 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x000b,-86
270 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x000a,-89
271 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000c,6,0x000d,-85
272 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0006,3,0x0001,-84
273 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0006,3,0x0007,-77
274 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x0006,3,0x0008,-82


```
275 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000a,3,0x0001,-95
276 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000a,3,0x000c,-95
277 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000a,3,0x000b,-88
278 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000d,1,0x000c,-81
279 837,899,True,2024-11-19 12:36:47,discoverTopology_Service,0x000b,1,0x000a,-84
```

Listing A.2: Log parziali dati raccolti per analisi dell'esperimento 1

Appendice B

Dati di supporto alla validazione: Esperimento 2

Nota: alcuni termini presenti nel logfile sono stati abbreviati per favorire una più piacevole esperienza di lettura. In particolare:

- pR.Srv = parameterRequest_Service
- pR.csv = parameterRequest.csv

B.1 File di log manuale

Time (HH.MM)	# nodi aggiunti	# nodi tot	# exp	Note
12.45	1 proxy + 2	3	3	-
12.48	2	5	3	-
12.51	2	7	3	-
12.55	2	9	3	-
12.58	2	11	3	pkt loss exp2 12:58:47

Tabella B.1: File di log scritto a mano durante l'esperimento per parameterRequest

B.2 File di log

```
1 2024-11-19 12:45:36 INFO : <module> : Selected [2] - pR_Srv
2 2024-11-19 12:45:36 INFO : pR_Srv : Sent parameters request
3 2024-11-19 12:45:36 INFO : pR_Srv : Started 5 second timer
4 2024-11-19 12:45:36 INFO : pR_Srv : Data packet received from 0x0001
5 2024-11-19 12:45:36 INFO : pR_Srv : Data packet received from 0x0005
6 2024-11-19 12:45:36 INFO : pR_Srv : Data packet received from 0x0007
7 2024-11-19 12:45:41 INFO : pR_Srv : Expired 5 second timer
8 2024-11-19 12:45:41 INFO : pR_Srv : Data id=950 in data/pR.csv
9 2024-11-19 12:45:41 INFO : <module> : Returned from [2] - pR_Srv
10 2024-11-19 12:46:00 INFO : <module> : Selected [2] - pR_Srv
11 2024-11-19 12:46:00 INFO : pR_Srv : Sent parameters request
12 2024-11-19 12:46:00 INFO : pR_Srv : Started 5 second timer
13 2024-11-19 12:46:01 INFO : pR_Srv : Data packet received from 0x0001
14 2024-11-19 12:46:02 INFO : pR_Srv : Data packet received from 0x0005
15 2024-11-19 12:46:02 INFO : pR_Srv : Data packet received from 0x0007
16 2024-11-19 12:46:05 INFO : pR_Srv : Expired 5 second timer
17 2024-11-19 12:46:05 INFO : pR_Srv : Data id=408 in data/pR.csv
18 2024-11-19 12:46:05 INFO : <module> : Returned from [2] - pR_Srv
19 2024-11-19 12:46:30 INFO : <module> : Selected [2] - pR_Srv
20 2024-11-19 12:46:30 INFO : pR_Srv : Sent parameters request
21 2024-11-19 12:46:30 INFO : pR_Srv : Started 5 second timer
22 2024-11-19 12:46:31 INFO : pR_Srv : Data packet received from 0x0001
23 2024-11-19 12:46:31 INFO : pR_Srv : Data packet received from 0x0007
24 2024-11-19 12:46:31 INFO : pR_Srv : Data packet received from 0x0005
25 2024-11-19 12:46:35 INFO : pR_Srv : Expired 5 second timer
26 2024-11-19 12:46:35 INFO : pR_Srv : Data id=701 in data/pR.csv
27 2024-11-19 12:46:35 INFO : <module> : Returned from [2] - pR_Srv
28 2024-11-19 12:48:07 INFO : <module> : Selected [2] - pR_Srv
29 2024-11-19 12:48:07 INFO : pR_Srv : Sent parameters request
30 2024-11-19 12:48:07 INFO : pR_Srv : Started 5 second timer
31 2024-11-19 12:48:07 INFO : pR_Srv : Data packet received from 0x0001
32 2024-11-19 12:48:07 INFO : pR_Srv : Data packet received from 0x000a
33 2024-11-19 12:48:07 INFO : pR_Srv : Data packet received from 0x0008
34 2024-11-19 12:48:07 INFO : pR_Srv : Data packet received from 0x0007
35 2024-11-19 12:48:08 INFO : pR_Srv : Data packet received from 0x0005
36 2024-11-19 12:48:12 INFO : pR_Srv : Expired 5 second timer
37 2024-11-19 12:48:12 INFO : pR_Srv : Data id=892 in data/pR.csv
38 2024-11-19 12:48:12 INFO : <module> : Returned from [2] - pR_Srv
39 2024-11-19 12:48:25 INFO : <module> : Selected [2] - pR_Srv
40 2024-11-19 12:48:25 INFO : pR_Srv : Sent parameters request
41 2024-11-19 12:48:25 INFO : pR_Srv : Started 5 second timer
42 2024-11-19 12:48:25 INFO : pR_Srv : Data packet received from 0x0001
43 2024-11-19 12:48:26 INFO : pR_Srv : Data packet received from 0x0005
44 2024-11-19 12:48:26 INFO : pR_Srv : Data packet received from 0x0007
45 2024-11-19 12:48:26 INFO : pR_Srv : Data packet received from 0x000a
46 2024-11-19 12:48:26 INFO : pR_Srv : Data packet received from 0x0008
47 2024-11-19 12:48:30 INFO : pR_Srv : Expired 5 second timer
48 2024-11-19 12:48:30 INFO : pR_Srv : Data id=154 in data/pR.csv
```

```
49 2024-11-19 12:48:30 INFO : <module> : Returned from [2] - pR_Srv
50 2024-11-19 12:48:37 INFO : <module> : Selected [2] - pR_Srv
51 2024-11-19 12:48:37 INFO : pR_Srv : Sent parameters request
52 2024-11-19 12:48:37 INFO : pR_Srv : Started 5 second timer
53 2024-11-19 12:48:38 INFO : pR_Srv : Data packet received from 0x0001
54 2024-11-19 12:48:38 INFO : pR_Srv : Data packet received from 0x0007
55 2024-11-19 12:48:38 INFO : pR_Srv : Data packet received from 0x0008
56 2024-11-19 12:48:38 INFO : pR_Srv : Data packet received from 0x000a
57 2024-11-19 12:48:38 INFO : pR_Srv : Data packet received from 0x0005
58 2024-11-19 12:48:42 INFO : pR_Srv : Expired 5 second timer
59 2024-11-19 12:48:42 INFO : pR_Srv : Data id=516 in data/pR.csv
60 2024-11-19 12:48:42 INFO : <module> : Returned from [2] - pR_Srv
61 2024-11-19 12:51:04 INFO : <module> : Selected [2] - pR_Srv
62 2024-11-19 12:51:04 INFO : pR_Srv : Sent parameters request
63 2024-11-19 12:51:04 INFO : pR_Srv : Started 5 second timer
64 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x0001
65 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x0005
66 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x000a
67 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x0007
68 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x0008
69 2024-11-19 12:51:05 INFO : pR_Srv : Data packet received from 0x000e
70 2024-11-19 12:51:06 INFO : pR_Srv : Data packet received from 0x0009
71 2024-11-19 12:51:09 INFO : pR_Srv : Expired 5 second timer
72 2024-11-19 12:51:09 INFO : pR_Srv : Data id=318 in data/pR.csv
73 2024-11-19 12:51:09 INFO : <module> : Returned from [2] - pR_Srv
74 2024-11-19 12:51:43 INFO : <module> : Selected [2] - pR_Srv
75 2024-11-19 12:51:43 INFO : pR_Srv : Sent parameters request
76 2024-11-19 12:51:43 INFO : pR_Srv : Started 5 second timer
77 2024-11-19 12:51:43 INFO : pR_Srv : Data packet received from 0x0001
78 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x0007
79 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x000e
80 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x0005
81 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x0008
82 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x000a
83 2024-11-19 12:51:44 INFO : pR_Srv : Data packet received from 0x0009
84 2024-11-19 12:51:48 INFO : pR_Srv : Expired 5 second timer
85 2024-11-19 12:51:48 INFO : pR_Srv : Data id=765 in data/pR.csv
86 2024-11-19 12:51:48 INFO : <module> : Returned from [2] - pR_Srv
87 2024-11-19 12:51:56 INFO : <module> : Selected [2] - pR_Srv
88 2024-11-19 12:51:56 INFO : pR_Srv : Sent parameters request
89 2024-11-19 12:51:56 INFO : pR_Srv : Started 5 second timer
90 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x0001
91 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x0007
92 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x000e
93 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x000a
94 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x0005
95 2024-11-19 12:51:56 INFO : pR_Srv : Data packet received from 0x0009
96 2024-11-19 12:51:57 INFO : pR_Srv : Data packet received from 0x0008
97 2024-11-19 12:52:01 INFO : pR_Srv : Expired 5 second timer
98 2024-11-19 12:52:01 INFO : pR_Srv : Data id=69 in data/pR.csv
```

```
99 2024-11-19 12:52:01 INFO : <module> : Returned from [2] - pR_Srv
100 2024-11-19 12:55:00 INFO : <module> : Selected [2] - pR_Srv
101 2024-11-19 12:55:00 INFO : pR_Srv : Sent parameters request
102 2024-11-19 12:55:00 INFO : pR_Srv : Started 5 second timer
103 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x0001
104 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x0007
105 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x000a
106 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x000e
107 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x0006
108 2024-11-19 12:55:01 INFO : pR_Srv : Data packet received from 0x0008
109 2024-11-19 12:55:02 INFO : pR_Srv : Data packet received from 0x0005
110 2024-11-19 12:55:02 INFO : pR_Srv : Data packet received from 0x000b
111 2024-11-19 12:55:02 INFO : pR_Srv : Data packet received from 0x0009
112 2024-11-19 12:55:05 INFO : pR_Srv : Expired 5 second timer
113 2024-11-19 12:55:05 INFO : pR_Srv : Data id=721 in data/pR.csv
114 2024-11-19 12:55:05 INFO : <module> : Returned from [2] - pR_Srv
115 2024-11-19 12:55:19 INFO : <module> : Selected [2] - pR_Srv
116 2024-11-19 12:55:19 INFO : pR_Srv : Sent parameters request
117 2024-11-19 12:55:19 INFO : pR_Srv : Started 5 second timer
118 2024-11-19 12:55:19 INFO : pR_Srv : Data packet received from 0x0001
119 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x000a
120 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x0008
121 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x0006
122 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x0005
123 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x000e
124 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x0007
125 2024-11-19 12:55:20 INFO : pR_Srv : Data packet received from 0x000b
126 2024-11-19 12:55:22 INFO : pR_Srv : Data packet received from 0x0009
127 2024-11-19 12:55:24 INFO : pR_Srv : Expired 5 second timer
128 2024-11-19 12:55:24 INFO : pR_Srv : Data id=132 in data/pR.csv
129 2024-11-19 12:55:24 INFO : <module> : Returned from [2] - pR_Srv
130 2024-11-19 12:56:07 INFO : <module> : Selected [2] - pR_Srv
131 2024-11-19 12:56:07 INFO : pR_Srv : Sent parameters request
132 2024-11-19 12:56:07 INFO : pR_Srv : Started 5 second timer
133 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0001
134 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0005
135 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0006
136 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0009
137 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x000b
138 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0007
139 2024-11-19 12:56:08 INFO : pR_Srv : Data packet received from 0x0008
140 2024-11-19 12:56:09 INFO : pR_Srv : Data packet received from 0x000a
141 2024-11-19 12:56:10 INFO : pR_Srv : Data packet received from 0x000e
142 2024-11-19 12:56:12 INFO : pR_Srv : Expired 5 second timer
143 2024-11-19 12:56:12 INFO : pR_Srv : Data id=670 in data/pR.csv
144 2024-11-19 12:56:12 INFO : <module> : Returned from [2] - pR_Srv
145 2024-11-19 12:58:28 INFO : <module> : Selected [2] - pR_Srv
146 2024-11-19 12:58:28 INFO : pR_Srv : Sent parameters request
147 2024-11-19 12:58:28 INFO : pR_Srv : Started 5 second timer
148 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0001
```

```

149 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x000b
150 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x000d
151 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x000c
152 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x000a
153 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0007
154 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x000e
155 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0008
156 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0005
157 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0006
158 2024-11-19 12:58:29 INFO : pR_Srv : Data packet received from 0x0009
159 2024-11-19 12:58:33 INFO : pR_Srv : Expired 5 second timer
160 2024-11-19 12:58:33 INFO : pR_Srv : Data id=352 in data/pR.csv
161 2024-11-19 12:58:33 INFO : <module> : Returned from [2] - pR_Srv
162 2024-11-19 12:58:47 INFO : <module> : Selected [2] - pR_Srv
163 2024-11-19 12:58:47 INFO : pR_Srv : Sent parameters request
164 2024-11-19 12:58:47 INFO : pR_Srv : Started 5 second timer
165 2024-11-19 12:58:47 INFO : pR_Srv : Data packet received from 0x0001
166 2024-11-19 12:58:48 INFO : pR_Srv : Data packet received from 0x000e
167 2024-11-19 12:58:48 INFO : pR_Srv : Data packet received from 0x0009
168 2024-11-19 12:58:48 INFO : pR_Srv : Data packet received from 0x000b
169 2024-11-19 12:58:49 INFO : pR_Srv : Data packet received from 0x0005
170 2024-11-19 12:58:49 INFO : pR_Srv : Data packet received from 0x0007
171 2024-11-19 12:58:49 INFO : pR_Srv : Data packet received from 0x0006
172 2024-11-19 12:58:49 INFO : pR_Srv : Data packet received from 0x000a
173 2024-11-19 12:58:49 INFO : pR_Srv : Data packet received from 0x000c
174 2024-11-19 12:58:50 INFO : pR_Srv : Data packet received from 0x000d
175 2024-11-19 12:58:52 INFO : pR_Srv : Expired 5 second timer
176 2024-11-19 12:58:52 INFO : pR_Srv : Data id=659 in data/pR.csv
177 2024-11-19 12:58:52 INFO : <module> : Returned from [2] - pR_Srv
178 2024-11-19 12:59:49 INFO : <module> : Selected [2] - pR_Srv
179 2024-11-19 12:59:49 INFO : pR_Srv : Sent parameters request
180 2024-11-19 12:59:49 INFO : pR_Srv : Started 5 second timer
181 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x0001
182 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x0005
183 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x000d
184 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x000a
185 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x000b
186 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x000c
187 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x0006
188 2024-11-19 12:59:50 INFO : pR_Srv : Data packet received from 0x0008
189 2024-11-19 12:59:51 INFO : pR_Srv : Data packet received from 0x0009
190 2024-11-19 12:59:52 INFO : pR_Srv : Data packet received from 0x000e
191 2024-11-19 12:59:52 INFO : pR_Srv : Data packet received from 0x0007
192 2024-11-19 12:59:54 INFO : pR_Srv : Expired 5 second timer
193 2024-11-19 12:59:54 INFO : pR_Srv : Data id=805 in data/pR.csv
194 2024-11-19 12:59:54 INFO : <module> : Returned from [2] - pR_Srv

```

Listing B.1: Log dati raccolti per analisi dell'esperimento 2

B.3 File *parameterRequest.csv*

```
1 id,timestamp,service,this_node_addr,cid,pid,net_key,app_key,net_retransmit_count,
  net_retransmit_interval_ms,relay_on,ttl,tx_def_power
2 950,2024-11-19 12:45:41,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
3 950,2024-11-19 12:45:41,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
4 950,2024-11-19 12:45:41,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
5 408,2024-11-19 12:46:05,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
6 408,2024-11-19 12:46:05,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
7 408,2024-11-19 12:46:05,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
8 701,2024-11-19 12:46:35,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
9 701,2024-11-19 12:46:35,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
10 701,2024-11-19 12:46:35,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
11 892,2024-11-19 12:48:12,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
12 892,2024-11-19 12:48:12,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
13 892,2024-11-19 12:48:12,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
14 892,2024-11-19 12:48:12,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
15 892,2024-11-19 12:48:12,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
16 154,2024-11-19 12:48:30,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
17 154,2024-11-19 12:48:30,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
18 154,2024-11-19 12:48:30,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
19 154,2024-11-19 12:48:30,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
20 154,2024-11-19 12:48:30,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
21 516,2024-11-19 12:48:42,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
  ,2,20,True,7,7
22 516,2024-11-19 12:48:42,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
  ,2,20,True,7,7
23 516,2024-11-19 12:48:42,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
24 516,2024-11-19 12:48:42,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
  ,2,20,True,7,7
```


25 516,2024-11-19 12:48:42,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

26 318,2024-11-19 12:51:09,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

27 318,2024-11-19 12:51:09,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

28 318,2024-11-19 12:51:09,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

29 318,2024-11-19 12:51:09,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

30 318,2024-11-19 12:51:09,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

31 318,2024-11-19 12:51:09,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

32 318,2024-11-19 12:51:09,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

33 765,2024-11-19 12:51:48,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

34 765,2024-11-19 12:51:48,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

35 765,2024-11-19 12:51:48,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

36 765,2024-11-19 12:51:48,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

37 765,2024-11-19 12:51:48,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

38 765,2024-11-19 12:51:48,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

39 765,2024-11-19 12:51:48,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

40 69,2024-11-19 12:52:01,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

41 69,2024-11-19 12:52:01,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

42 69,2024-11-19 12:52:01,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

43 69,2024-11-19 12:52:01,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

44 69,2024-11-19 12:52:01,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

45 69,2024-11-19 12:52:01,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

46 69,2024-11-19 12:52:01,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

47 721,2024-11-19 12:55:05,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

48 721,2024-11-19 12:55:05,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

49 721,2024-11-19 12:55:05,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

50 721,2024-11-19 12:55:05,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

51 721,2024-11-19 12:55:05,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

52 721,2024-11-19 12:55:05,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

53 721,2024-11-19 12:55:05,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

54 721,2024-11-19 12:55:05,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

55 721,2024-11-19 12:55:05,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

56 132,2024-11-19 12:55:24,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

57 132,2024-11-19 12:55:24,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

58 132,2024-11-19 12:55:24,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

59 132,2024-11-19 12:55:24,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

60 132,2024-11-19 12:55:24,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

61 132,2024-11-19 12:55:24,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

62 132,2024-11-19 12:55:24,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

63 132,2024-11-19 12:55:24,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

64 132,2024-11-19 12:55:24,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

65 670,2024-11-19 12:56:12,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

66 670,2024-11-19 12:56:12,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

67 670,2024-11-19 12:56:12,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

68 670,2024-11-19 12:56:12,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

69 670,2024-11-19 12:56:12,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

70 670,2024-11-19 12:56:12,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

71 670,2024-11-19 12:56:12,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

72 670,2024-11-19 12:56:12,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

73 670,2024-11-19 12:56:12,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

74 352,2024-11-19 12:58:33,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

75 352,2024-11-19 12:58:33,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

76 352,2024-11-19 12:58:33,discoverTopology_Service,0x000d,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

77 352,2024-11-19 12:58:33,discoverTopology_Service,0x000c,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

78 352,2024-11-19 12:58:33,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

79 352,2024-11-19 12:58:33,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

80 352,2024-11-19 12:58:33,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

81 352,2024-11-19 12:58:33,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

82 352,2024-11-19 12:58:33,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

83 352,2024-11-19 12:58:33,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

84 352,2024-11-19 12:58:33,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

85 659,2024-11-19 12:58:52,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

86 659,2024-11-19 12:58:52,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

87 659,2024-11-19 12:58:52,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

88 659,2024-11-19 12:58:52,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

89 659,2024-11-19 12:58:52,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

90 659,2024-11-19 12:58:52,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
,2,20,True,7,7

91 659,2024-11-19 12:58:52,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

92 659,2024-11-19 12:58:52,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

93 659,2024-11-19 12:58:52,discoverTopology_Service,0x000c,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

94 659,2024-11-19 12:58:52,discoverTopology_Service,0x000d,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

95 805,2024-11-19 12:59:54,discoverTopology_Service,0x0001,0x02e5,0x0000,0x0084,0x0090
,2,20,True,7,7

96 805,2024-11-19 12:59:54,discoverTopology_Service,0x0005,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

97 805,2024-11-19 12:59:54,discoverTopology_Service,0x000d,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

98 805,2024-11-19 12:59:54,discoverTopology_Service,0x000a,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

99 805,2024-11-19 12:59:54,discoverTopology_Service,0x000b,0x02e5,0x0000,0x00f8,0x0056
,2,20,True,7,7

```
100 805,2024-11-19 12:59:54,discoverTopology_Service,0x000c,0x02e5,0x0000,0x00f8,0x0056
    ,2,20,True,7,7
101 805,2024-11-19 12:59:54,discoverTopology_Service,0x0006,0x02e5,0x0000,0x00f8,0x0056
    ,2,20,True,7,7
102 805,2024-11-19 12:59:54,discoverTopology_Service,0x0008,0x02e5,0x0000,0x00f8,0x0056
    ,2,20,True,7,7
103 805,2024-11-19 12:59:54,discoverTopology_Service,0x0009,0x02e5,0x0000,0x00f8,0x0056
    ,2,20,True,7,7
104 805,2024-11-19 12:59:54,discoverTopology_Service,0x000e,0x02e5,0x0000,0x00f8,0x0056
    ,2,20,True,7,7
105 805,2024-11-19 12:59:54,discoverTopology_Service,0x0007,0x02e5,0x0000,0x002c,0x008a
    ,2,20,True,7,7
```

Listing B.2: Dati raccolti per analisi dell'esperimento 2

Abbreviazioni e acronimi

IoT *Internet of Things*. 3

WSN *Wireless Sensor Network*. 3

Bluetooth SIG *Bluetooth Special Interest Group*, definito anche come *BSIG* o in questo elaborato anche solo come *SIG*. 5

BLE *Bluetooth Low Energy, Bluetooth LE*. 5

PDU *Protocol Data Unit*, l'unità d'informazione scambiata tra due entità in un protocollo di comunicazione di un'architettura di rete a strati. 7

TTL *Time to Live*, meccanismo che determina il tempo di vita di un messaggio in una rete. 8

LPN *Low Power Node*. 9

ack *Acknowledged*. 9

unack *Unacknowledged*. 9

UUID *Universally Unique Identifier*, utilizzato per identificare individualmente i dati nei sistemi informatici. 10

SPOF *Single point of failure*. 16

AR/VR/MR *Augmented reality, Virtual reality, Mixed reality*. 17

UML *Unified Modeling Language*. 20

RSSI *Received signal strength indicator* (o *indication*), valore che indica la potenza del segnale ricevuto. 21

CID *Company Identifier*. 21

PID *Product Identifier*. 21

CSV *Comma-separated values*, formato di file basato su file di testo per gestione di dati in forma tabellare. 22

UART *Universal Asynchronous Receiver/Transmitter*, definisce un protocollo di comunicazione hardware, che converte flussi di bit di dati da un formato parallelo a un formato seriale asincrono e viceversa. 25

Bibliografia

- [1] Krista Sorri, Navonil Mustafee e Marko Seppänen. **Revisiting IoT definitions: A framework towards comprehensive use**. In: *Technological Forecasting and Social Change* 179 (2022), p. 121623. ISSN: 0040-1625. DOI: <https://doi.org/10.1016/j.techfore.2022.121623>. URL: <https://www.sciencedirect.com/science/article/pii/S004016252200155X>.
- [2] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer e Pat Doody. **Internet of Things Strategic Research Roadmap**. In: (gen. 2009). URL: https://www.researchgate.net/publication/267566519_Internet_of_Things_Strategic_Research_Roadmap.
- [3] Roberto Minerva, Abyi Biru e Domenico Rotondi. **Towards a definition of the Internet of Things (IoT)**. In: *IEEE Internet Initiative* 11 (2015), pp. 1–86. URL: https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf.
- [14] Seyed Mahdi Darroudi e Carles Gomez. **Bluetooth Low Energy Mesh Networks: A Survey**. In: *Sensors* 177 (2017). ISSN: 1424-8220. DOI: 10.3390/s17071467. URL: <https://www.mdpi.com/1424-8220/17/7/1467>.
- [15] Iynkaran Natgunanathan, Niroshinie Fernando, Seng W. Loke e Charitha Weerasuriya. **Bluetooth Low Energy Mesh: Applications, Considerations and Current State-of-the-Art**. In: *Sensors* 234 (2023). ISSN: 1424-8220.

DOI: 10.3390/s23041826. URL: <https://www.mdpi.com/1424-8220/23/4/1826>.

- [16] Seyed Mahdi Darroudi, Carles Gomez e Jon Crowcroft. **Bluetooth Low Energy Mesh Networks: A Standards Perspective**. In: *IEEE Communications Magazine* 584 (2020), pp. 95–101. DOI: 10.1109/MCOM.001.1900523.
- [17] Muhammad Rizwan Ghori, Tat-Chee Wan e Gian Chand Sodhy. **Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols**. In: *Sensors* 2012 (2020). ISSN: 1424-8220. DOI: 10.3390/s20123590. URL: <https://www.mdpi.com/1424-8220/20/12/3590>.
- [18] Alexandre Adomnicai, Jacques J. A. Fournier e Laurent Masson. **Hardware Security Threats Against Bluetooth Mesh Networks**. In: *2018 IEEE Conference on Communications and Network Security (CNS)*. 2018, pp. 1–9. DOI: 10.1109/CNS.2018.8433184.
- [19] Raúl Rondón, Aamir Mahmood, Simone Grimaldi e Mikael Gidlund. **Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis**. In: *IEEE Internet of Things Journal* 73 (2020), pp. 2089–2101. DOI: 10.1109/JIOT.2019.2960248.
- [20] Bing Lin e Hang Su. **Experimental Evaluation and Performance Improvement of Bluetooth Mesh Network With Broadcast Storm**. In: *IEEE Access* 11 (2023), pp. 137810–137820. DOI: 10.1109/ACCESS.2023.3339795.
- [21] Ziyad Khalaf Farej e Azhar Waleed Talab. **Performance Validation and Hardware Implementation of a BLE Mesh Network by Using ESP-32 Board**. In: *Proceedings of 3rd International Conference on Mathematical Modeling and Computational Science*. A cura di Sheng-Lung Peng, Noor Zaman Jhanjhi, Souvik Pal e Fathi Amsaad. Singapore: Springer Nature Singapore, 2023, pp. 331–343. ISBN: 978-981-99-3611-3.
- [22] Archit Gajjar, Yunxiang Zhang e Xiaokun Yang. **A smart building system integrated with an edge computing algorithm and IoT mesh networks: demo abstract**. In: *Proceedings of the Second ACM/IEEE Sym-*

- posium on Edge Computing*. SEC '17. San Jose, California: Association for Computing Machinery, 2017. ISBN: 9781450350877. DOI: 10.1145/3132211.3132462. URL: <https://doi.org/10.1145/3132211.3132462>.
- [23] Leonardo Eras, Federico Domínguez e Caril Martinez. **Viability characterization of a proof-of-concept Bluetooth mesh smart building application**. In: *International Journal of Distributed Sensor Networks* 185 (2022), p. 15501329221097819. DOI: 10.1177/15501329221097819. eprint: <https://doi.org/10.1177/15501329221097819>. URL: <https://doi.org/10.1177/15501329221097819>.
- [24] Flor Álvarez, Lars Almon, Hauke Radtke e Matthias Hollick. **Bluemergency: Mediating Post-disaster Communication Systems using the Internet of Things and Bluetooth Mesh**. In: *2019 IEEE Global Humanitarian Technology Conference (GHTC)*. 2019, pp. 1–8. DOI: 10.1109/GHTC46095.2019.9033063.
- [25] Terence Ching Yang Lam, Sherlyn Shi Ling Yew e Sye Loong Keoh. **Bluetooth Mesh Networking: An Enabler of Smart Factory Connectivity and Management**. In: *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2019, pp. 1–6. DOI: 10.23919/APNOMS.2019.8893091.
- [26] Paul Seymer, Duminda Wijesekera e Cing-Dao Kan. **Secure Outdoor Smart Parking Using Dual Mode Bluetooth Mesh Networks**. In: *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. 2019, pp. 1–7. DOI: 10.1109/VTCspring.2019.8746459.
- [27] Leonardo Montecchiari, Angelo Trotta, Luciano Bononi e Marco 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)*. 2022, pp. 193–199. DOI: 10.1109/CCNC49033.2022.9700518.

- [28] Yuri Murillo, Brecht Reynders, Alessandro Chiumento, Salman Malik, Pieter Crombez e Sofie Pollin. **Bluetooth now or low energy: Should BLE mesh become a flooding or connection oriented network?** In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–6. DOI: 10.1109/PIMRC.2017.8292705.
- [29] Adriano Souza Brandão, Matheus Cordeiro Lima, Claudia Jacy Barenco Abbas e Luis Javier García Villalba. **An Energy Balanced Flooding Algorithm for a BLE Mesh Network.** In: *IEEE Access* 8 (2020), pp. 97946–97958. DOI: 10.1109/ACCESS.2020.2993819.
- [30] Rui Li, Xiaohui Li e Yueming Ding. **Link Prediction Algorithm for BLE Mesh Network in Health Monitoring System.** In: *2020 Chinese Control And Decision Conference (CCDC)*. 2020, pp. 1997–2001. DOI: 10.1109/CCDC49329.2020.9164868.
- [41] Keran Jegasothy. **Un sistema smart di auto-configurazione di reti BLE Mesh - progettazione implementazione ed analisi.** In: (). URL: <https://amslaurea.unibo.it/id/eprint/28910/>.

Sitografia

- [4] IBM. **Che cos'è l'Internet of Things (IoT)?** Accessed 15/11/2024. URL: <https://www.ibm.com/it-it/topics/internet-of-things>.
- [5] Bluetooth SIG Inc. **Learn about Bluetooth.** Accessed 17/11/2024. URL: <https://www.bluetooth.com/learn-about-bluetooth/>.
- [6] Bluetooth SIG Inc. **Auracast.** Accessed 17/11/2024. URL: <https://www.bluetooth.com/auracast/>.
- [7] Bluetooth SIG Inc. **Learn about Direction Finding.** Accessed 17/11/2024. URL: <https://www.bluetooth.com/learn-about-bluetooth/feature-enhancements/direction-finding/>.
- [8] Bluetooth SIG Inc. **Mesh Profile Bluetooth Specification v1.0.1.** Accessed 20/11/2024. URL: <https://www.bluetooth.com/specifications/specs/mesh-profile-1-0-1/>.
- [9] Bluetooth SIG Inc. **Mesh Model Bluetooth Specification v1.1.** Accessed 20/11/2024. URL: <https://www.bluetooth.com/specifications/specs/mesh-model-1-1/>.
- [10] Bluetooth SIG Inc. **Bluetooth Mesh Directed Forwarding and Managed Flooding.** Accessed 22/11/2024. URL: <https://www.bluetooth.com/mesh-directed-forwarding/>.
- [11] Wikipedia contributors. **Piconet** — **Wikipedia, The Free Encyclopedia.** [Online; accessed 21/11/2024]. 2024.

- [12] Bluetooth SIG Inc. **Enhancing Bluetooth Technology: Specifications in Development**. Accessed 21/11/2024. URL: <https://www.bluetooth.com/specifications/specifications-in-development/>.
- [13] **Google Scholar**. Accessed 21/11/2024. URL: <https://scholar.google.com/>.
- [31] IEEE Standard Association. **Registration Authority Company ID (CID)**. Accessed 23/11/2024. URL: <https://standards.ieee.org/products-programs/regauth/cid/>.
- [32] Espressif Systems. **ESP32**. Accessed 24/11/2024. URL: <https://www.espressif.com/en/products/socs/esp32>.
- [33] Espressif Systems. **ESP-IDF Repository**. Accessed 24/11/2024. URL: <https://github.com/espressif/esp-idf>.
- [34] Espressif Systems. **ESP-IDF Programming Guide**. Accessed 24/11/2024. URL: <https://docs.espressif.com/projects/esp-idf/en/v5.3.1/esp32/index.html>.
- [35] Espressif Systems. **ESP-IDF Extension for VSCode**. Accessed 24/11/2024. URL: <https://docs.espressif.com/projects/vscode-esp-idf-extension/en/latest/>.
- [36] Nordic Semiconductor. **ESP-IDF Extension for VSCode**. Accessed 24/11/2024. URL: <https://www.nordicsemi.com/Products/Development-tools/nRF-Mesh>.
- [37] Espressif Systems. **Establish Serial Connection with ESP32**. Accessed 24/11/2024. URL: <https://docs.espressif.com/projects/esp-idf/en/v5.3.1/esp32/get-started/establish-serial-connection.html>.
- [38] Espressif Systems Github Repository. **ESP BLE Mesh Vendor Client Example**. Accessed 24/11/2024. URL: https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/esp_ble_mesh/vendor_models/vendor_client.

- [39] Espressif Systems Github Repository. **ESP BLE Mesh Vendor Server Example**. Accessed 24/11/2024. URL: https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/esp_ble_mesh/vendor_models/vendor_server.
- [40] Wikipedia contributors. **End-to-end delay** — **Wikipedia, The Free Encyclopedia**. [Online; accessed 30-November-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=End-to-end_delay&oldid=1233342500.

