ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE "GUGLIELMO MARCONI" - DEI

CORSO DI LAUREA
INGEGNERIA ELETTRONICA

CONFRONTO TRA MODBUS TLS E MQTT OVER SSL/TLS: VALUTAZIONE DELLA SICUREZZA E DELL'EFFICIENZA NELLE COMUNICAZIONI IOT INDUSTRIALI

Relatore

Prof. Roffia Luca

Presentata da

Youssefeldin Khaled Taha Ebrahim Ragab Ahmed

Sessione mese anno

Anno Accademico 2023/2024

DICHIARAZIONE DI ORIGINALITÀ DELLA TESI DI LAUREA

Io sottoscritto Youssefeldin Khaled Taha Ebrahim Ragab Ahmed nato al Giza il 05 settembre 2001 dichiaro che l'elaborato che segue è frutto del mio lavoro originale, che nessuno lo ha scritto in mia vece, che non ho copiato il lavoro di altri e che ho documentato tutte le fonti che ho utilizzato. Dichiaro anche che ho personalmente consultato tutte le fonti citate.

Dichiaro di non aver presentato questo elaborato presso altre istituzioni al fine di ottenere diplomi, lauree, certificazioni, ecc., né di averlo pubblicato in precedenza, in parte o per intero.

Dichiaro di aver letto e compreso che il 'plagio' è una "falsa attribuzione a sé di opere o scoperte delle quali spettino ad altri i diritti di invenzione o di proprietà" (Devoto-Oli, *Dizionario della Lingua italiana*, Milano, Le Monnier, 2001). Dichiaro di aver compreso che quando si elabora un lavoro che incorpori parole o idee di altri, si deve citare appropriatamente la fonte di quell'informazione. Se non lo si fa, si commette un plagio, che è un reato (legge n. 633 del 1941 sul diritto d'autore). Comprendo che l'individuazione di plagio anche in una sola parte di questa tesi potrebbe pregiudicare la presunzione di autenticità del resto. Dichiaro di aver compreso che il plagio è un atto illecito e che nel caso io abbia plagiato il lavoro altrui in una o più parti di questa tesi, il risultato comprometterà l'esito della mia laurea. Sono consapevole che in caso io abbia commesso plagio, il docente può rifiutarsi di ammettermi alla discussione della tesi di laurea.

In fede,

Youssefeldin Khaled Taha Ebrahim Ragab Ahmed

Data 12/09/2024

Indice

Elenco dei grafici

Elenco delle figure

Elenco delle tabelle

Capitolo 1: Introduzione

Attualmente, l'industria è sempre più orientata verso l'automazione della produzione, con un'intensa interconnessione tra robot e sistemi embedded all'interno delle macchine per gestire le linee di produzione con minimo o nessun intervento umano. Questa evoluzione ha portato a un significativo incremento dell'uso dell'Internet of Things (IoT) in ambito industriale. In questo contesto, i protocolli di comunicazione emergono come un fattore critico per il funzionamento efficace di qualsiasi sistema IoT. Esistono numerosi protocolli utilizzati nei sistemi IoT, tra cui Modbus, PROFINET, MQTT, DDS. Tuttavia, per l'ambito industriale, Modbus e MQTT rivestono un'importanza particolare. Questi protocolli, tra i più diffusi e rilevanti, saranno oggetto di discussione in questa tesi, specialmente considerando la loro implementazione con l'aggiunta del livello di sicurezza TLS.

1.1 Contesto e motivazione

L'Internet of Things (IoT) ha segnato una profonda trasformazione nel settore industriale, promuovendo l'evoluzione verso fabbriche sempre più connesse e automatizzate. Inizialmente focalizzato su applicazioni consumer, l'IoT ha trovato una vasta applicazione in ambienti industriali, dove la capacità di monitorare e controllare in tempo reale macchinari e processi offre vantaggi significativi in termini di efficienza e riduzione dei costi.

L'automazione industriale ha guadagnato terreno come una componente chiave delle moderne fabbriche, dove robot e sistemi embedded operano in modo sincronizzato per eseguire compiti produttivi con precisione millimetrica. Questo livello di automazione non solo ottimizza la produzione, ma riduce anche la necessità di interventi umani, minimizzando il margine di errore e aumentando la sicurezza sul lavoro.

In questo scenario, la scelta e l'implementazione di protocolli di comunicazione efficaci diventano cruciali. Protocolli come Modbus e MQTT sono ampiamente utilizzati per la loro capacità di facilitare una comunicazione affidabile tra dispositivi. Tuttavia, con l'aumento delle minacce cyber e la necessità di proteggere dati sensibili e infrastrutture critiche, la sicurezza di questi protocolli assume un ruolo primario.

La sicurezza è diventata una delle principali preoccupazioni nell'IoT industriale, con un'enfasi particolare sulla protezione contro gli attacchi cyber che possono causare danni estesi e interruzioni nelle operazioni. L'integrazione del Transport Layer Security (TLS) in protocolli standard come Modbus e MQTT rappresenta un passo fondamentale verso la creazione di un ambiente di comunicazione sicuro.

L'aggiunta di strati di sicurezza come la crittografia TLS aiuta a proteggere la trasmissione dei dati tra dispositivi IoT, garantendo che solo le parti autorizzate possano accedere e manipolare le informazioni trasmesse. Questo è particolarmente rilevante in ambienti industriali, dove la perdita o il furto di dati operativi può avere conseguenze disastrose.

1.2 Obiettivi della ricerca

Nell'ambito di questo studio, si presta particolare attenzione ai protocolli Modbus TLS e MQTT over SSL/TLS, essenziali per garantire sicurezza ed efficienza nelle comunicazioni dell'IoT industriale. Questo capitolo delinea gli obiettivi specifici della ricerca, mirando a fornire un quadro dettagliato e analitico delle prestazioni e della sicurezza di questi protocolli.

- **Valutazione della sicurezza**: Esaminare in dettaglio come il TLS è implementato in ciascun protocollo e valutare l'efficacia di questa implementazione.

- **Misurazione dell'efficienza operativa**: Misurare l'impatto dell'integrazione del TLS sulla latenza e sul throughput della comunicazione. Questo include testare come la crittografia e la decrittografia influenzano il tempo di trasmissione e la velocità di elaborazione dei dati in scenari reali. Analizzare l'uso delle risorse come CPU e memoria durante l'esecuzione dei protocolli. L'obiettivo è determinare l'efficienza di Modbus TLS e MQTT over SSL/TLS in termini di consumo di risorse hardware, essenziale per l'ottimizzazione e il dimensionamento delle soluzioni IoT.
- **Confronto e applicabilità in scenari industriali**: Identificare e descrivere scenari industriali specifici dove l'uso di Modbus TLS e MQTT over SSL/TLS è vantaggioso. Analizzare come questi protocolli si adattano a diverse condizioni e requisiti industriali. Integrare opinioni e feedback da esperti del settore per validare i risultati e garantire che le conclusioni siano ancorate a prassi e realtà operative concrete.
- **Contributo alla letteratura e alle prassi**: Compilare e discutere le teorie correnti relative ai protocolli in questione, evidenziando lacune e opportunità per ulteriori studi.

L'obiettivo di questo studio è fornire linee guida concrete per la loro implementazione in contesti industriali. Attraverso questi obiettivi, la ricerca mira a contribuire significativamente alla sicurezza delle comunicazioni IoT, fondamentale per il futuro dell'automazione industriale.

Capitolo 2: Revisione della letteratura

Prima di confrontare i protocolli Modbus e MQTT per l'IOT industriale, è essenziale fornire una panoramica teorica sui protocolli di comunicazione, spiegando le basi su cui sono progettati. Questa sezione esamina i modelli di protocolli di comunicazione utilizzati nelle applicazioni industriali, con un focus sui concetti fondamentali che guidano il loro sviluppo e la loro implementazione.

2.1 Modello di protocolli di comunicazione

In origine, le reti di comunicazione erano sistemi chiusi in cui tutti i componenti dovevano essere forniti dallo stesso costruttore. Ogni costruttore utilizzava protocolli e tecnologie proprietarie, il che creava significativi problemi di incompatibilità tra i vari dispositivi e sistemi. Questa mancanza di interoperabilità derivava dalle differenze nelle tecniche di trasmissione e nei protocolli di comunicazione utilizzati, rendendo difficile per le applicazioni operare efficacemente all'interno della rete.

A partire dal 1975 la **ISO** (International Organization for Standardizzazioni) ha iniziato a sviluppare una serie di **standard unificati** per la realizzazione e la interconnessione di reti di calcolatori **aperte** e dei relativi protocolli.

La prima cosa cha ha fatto la **ISO** è il modello **OSI-RM** (Open System Interconnection Reference Model), che è diventato lo standard internazionale nel 1984 (ISO 7498). Questo modello è basato sulla architettura suddivisa in più strati, che mi dà il vantaggio di dividere il problema di comunicazione in sottoproblemi dove ogni stratto si occupa di una funzione indipendente dall'altra e questo per porta un terzo vantaggio che gli strati posso essere sviluppati e prodotti da entità diverse.

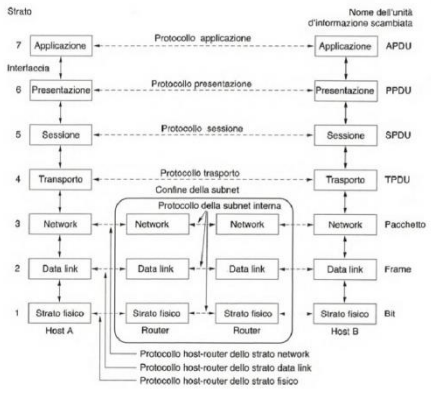

Figure 2.1- modello ISO OSI

Il modello ISO OSI è costituito da sette strati numerati da basso verso l'alto dove sono divisi in livello ed ogni livello offre un servizio allo strato più alto e si interconnessone gli strati tramite interfacce. Gli strati 1, 2 e 3 sono detti **Network Oriented Layers(Lower)**, dove il primo è lo strato fisico, il secondo è lo strato di collegamento ed il terzo è lo strato di Rete. Gli strati 5, 6, 7 sono detti **Application Oriented Layers(Upper)**, dove il quinto è lo strato di sessione, il secondo è lo strato di presentazione ed il terzo è quello di Applicazione. Lo strato quattro è lo strato importante che lega gli strati Upper agli strati Lower.

Col tempo e basando sugli standard sei IOS il IETF (Internet Engineering Task Force) ha potuto semplificare l'architettura a quattro strati solo denominata Modello TCP/IP, che sono in ordine crescente Network Access, Network, Transport ed Application.

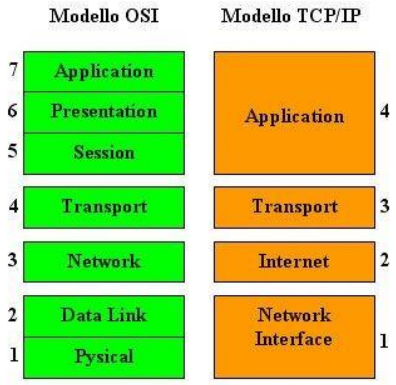

Figura 2.2- confronto modelli ISO e TCP/IP

Attualmente, nella progettazione dei protocolli di comunicazione, si utilizza quasi sempre il modello TCP/IP. Tra questi protocolli, MQTT e Modbus, che verranno discussi più approfonditamente, basano le loro prestazioni sull'architettura TCP/IP.

2.2 Panoramica di Modbus e MQTT

Panoramica di Modbus

Modbus è un protocollo di comunicazione aperto sviluppato originariamente da Modicon (ora Schneider Electric) nel 1979 per l'automazione industriale. Progettato per facilitare la comunicazione tra dispositivi elettronici in ambienti industriali, Modbus è diventato uno degli standard di fatto per il collegamento di dispositivi come controller, sensori, attuatori e altri componenti di automazione. La sua ampia adozione è dovuta alla sua semplicità, facilità di implementazione e la capacità di operare in ambienti industriali difficili.

Modbus permette lo scambio di informazioni tra un dispositivo master (controllore) e uno o più dispositivi slave (sotto-controllori) in una rete. Il protocollo specifica un formato per i messaggi che permette la comunicazione tra i dispositivi, indipendentemente dal tipo di rete utilizzata per il collegamento fisico. Questo approccio rende Modbus estremamente versatile, consentendo di operare sia su reti seriali tradizionali (come RS-232 e RS-485) che su moderne reti basate su Ethernet utilizzando Modbus TCP/IP.

Il protocollo Modbus è basato su un'architettura di tipo Client-Server, spesso descritta anche come master-slave. In questo modello, un dispositivo master (client) invia richieste di dati o comandi a uno o più dispositivi slave (server), i quali rispondono fornendo le informazioni richieste o eseguendo le azioni specificate. Questa struttura consente al master di controllare e monitorare in tempo reale tutti i dispositivi connessi in una rete industriale.

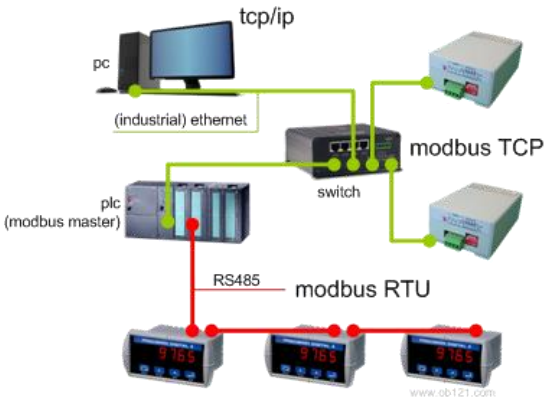

Figura 2.3- modello connessioni Modbus

Modbus supporta diversi tipi di messaggi per facilitare la comunicazione tra i dispositivi, inclusi:

- **Richieste**: Messaggi inviati dal master per richiedere dati o per inviare comandi.
- **Risposte**: Messaggi inviati dagli slave in risposta a una richiesta del master.
- **Messaggi di errore (eccezioni)**: Utilizzati quando si verifica un problema durante la comunicazione, come una richiesta non valida o la mancanza di dati disponibili.

Esistono diverse varianti del protocollo Modbus, ognuna progettata per specifici tipi di connessione e ambienti operativi:

- **Modbus RTU (Remote Terminal Unit)**: Utilizza un formato binario compatto per la trasmissione dei dati, rendendolo altamente efficiente in termini di larghezza di banda. È comunemente utilizzato su reti seriali, come RS-485, dove i dati vengono trasmessi come una serie di byte continui. La struttura di un messaggio Modbus RTU include un indirizzo (che identifica lo slave), un codice di funzione (che specifica l'operazione da eseguire), i dati trasmessi e un CRC (Cyclic Redundancy Check) per rilevare errori nella trasmissione.
- **Modbus ASCII**: Utilizza la codifica ASCII per la trasmissione dei dati, rendendo il formato dei messaggi più leggibile, ma meno efficiente rispetto a Modbus RTU. I dati vengono inviati in caratteri ASCII leggibili, con un prefisso e un suffisso di inizio e fine messaggio. Questa modalità è meno utilizzata oggi, poiché richiede una maggiore larghezza di banda rispetto a RTU e offre una minore efficienza nella comunicazione.
- **Modbus TCP/IP**: Adattamento di Modbus per le reti Ethernet, che utilizza il protocollo TCP/IP come livello di trasporto. In questa modalità, i messaggi Modbus vengono incapsulati in pacchetti TCP, permettendo di sfruttare l'infrastruttura di rete esistente e di comunicare su distanze maggiori con una maggiore velocità. Modbus TCP/IP offre un supporto nativo per la connessione a reti Ethernet, facilitando l'integrazione con tecnologie moderne dell'IoT industriale e migliorando l'interoperabilità tra dispositivi.

Modbus TCP/IP è una variante del protocollo Modbus progettata per operare su reti Ethernet, sfruttando lo stack di protocolli TCP/IP. A differenza delle modalità Modbus RTU e ASCII, che sono tipicamente utilizzate su reti seriali, Modbus TCP/IP consente una comunicazione più rapida e affidabile tra i

dispositivi attraverso le reti IP, supportando così applicazioni distribuite su scala più ampia, come quelle tipiche dell'Internet of Things (IoT) industriale.

In Modbus TCP/IP, i messaggi Modbus vengono incapsulati in pacchetti TCP. Questo significa che ogni messaggio Modbus è trattato come un payload all'interno di un pacchetto TCP, mantenendo la stessa struttura di messaggio utilizzata nelle versioni RTU o ASCII (con indirizzo, codice di funzione, dati e checksum), ma con alcune modifiche per adattarsi al protocollo TCP/IP. La porta TCP predefinita utilizzata per le comunicazioni Modbus è la porta 502, standardizzata per facilitare l'integrazione con l'infrastruttura di rete esistente.

L'utilizzo di Modbus su TCP/IP presenta diversi vantaggi significativi:

- **Interoperabilità e standardizzazione:** Modbus TCP/IP utilizza protocolli di rete standardizzati, rendendolo compatibile con una vasta gamma di dispositivi e sistemi di automazione industriale, inclusi PLC, SCADA, sensori e attuatori.
- **Facilità di integrazione con le reti moderne:** Poiché Modbus TCP/IP è progettato per funzionare su reti Ethernet, può essere facilmente integrato con reti IT e OT esistenti, supportando applicazioni distribuite e comunicazioni a lungo raggio.
- **Velocità di comunicazione:** Utilizzando TCP/IP come livello di trasporto, Modbus TCP/IP offre una velocità di comunicazione significativamente maggiore rispetto alle versioni su rete seriale. Questo è particolarmente utile per le applicazioni IoT industriali che richiedono un trasferimento di dati rapido e affidabile.
- **Supporto per connessioni multiple:** A differenza del Modbus tradizionale, che spesso opera in una configurazione master-slave unidirezionale, Modbus TCP/IP supporta connessioni multiple simultanee, consentendo comunicazioni bidirezionali tra diversi dispositivi in rete.

Nonostante i suoi vantaggi, Modbus su TCP/IP presenta alcune limitazioni:

- **Assenza di meccanismi di sicurezza intrinseci:** Modbus TCP/IP, nella sua forma base, non include funzioni di crittografia, autenticazione o controllo degli accessi. Questo lo rende vulnerabile a vari tipi di attacchi, come intercettazioni e manipolazioni di dati, soprattutto quando viene utilizzato su reti pubbliche o non protette.
- **Consumo di risorse di rete:** Sebbene Modbus TCP/IP possa operare su reti Ethernet, la sua efficienza dipende dalla qualità e dalla capacità della rete. Il protocollo TCP/IP può introdurre un sovraccarico di dati a causa della gestione delle connessioni, dei pacchetti e della trasmissione affidabile, il che può influire sulla latenza e sul throughput, specialmente in reti congestionate.

Modbus TCP/IP è ampiamente utilizzato in diverse applicazioni industriali, come il controllo di macchine, il monitoraggio ambientale, i sistemi SCADA e l'automazione di edifici. La sua capacità di sfruttare le infrastrutture Ethernet esistenti lo rende ideale per ambienti dove è richiesta una comunicazione rapida, scalabile e facilmente integrabile con altre tecnologie. Inoltre, è una scelta popolare per progetti IoT industriali che coinvolgono una rete di dispositivi distribuiti geograficamente, poiché consente la gestione remota e il monitoraggio centralizzato attraverso Internet.

Panoramica di MQTT

MQTT (Message Queuing Telemetry Transport) è un protocollo di messaggistica leggero progettato per la comunicazione machine-to-machine (M2M) e l'Internet of Things (IoT). Originariamente sviluppato da IBM negli anni '90, MQTT è diventato uno standard aperto (OASIS) ed è ampiamente utilizzato in applicazioni IoT grazie alla sua semplicità, efficienza e basso consumo di risorse. Il protocollo è progettato per funzionare in ambienti con larghezza di banda limitata, latenza elevata o connettività intermittente, rendendolo ideale per dispositivi con capacità limitate, come sensori e attuatori.

L'architettura di MQTT si basa su un modello publish/subscribe (pubblicazione/sottoscrizione), che è diverso dal tradizionale modello Client-Server utilizzato da molti altri protocolli, come Modbus. In un'architettura publish/subscribe, i client MQTT si connettono a un server centrale, chiamato broker, che gestisce la distribuzione dei messaggi tra i dispositivi. I client possono agire come publisher (pubblicatori), inviando messaggi su determinati "topic" (argomenti), o come subscriber (sottoscrittori), ricevendo messaggi su argomenti a cui si sono iscritti.

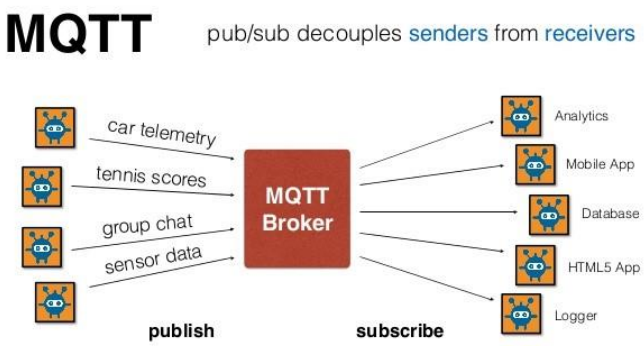

Figura 2.4- modello connessioni MQTT

- **Broker:** Il broker è un elemento centrale nell'architettura MQTT. Riceve tutti i messaggi pubblicati dai client, determina chi è abbonato a ciascun messaggio, e inoltra il messaggio ai client appropriati. I broker MQTT possono gestire migliaia di client connessi simultaneamente, offrendo scalabilità e flessibilità.
- **Publisher e Subscriber:** Un publisher invia messaggi a uno o più argomenti sul broker, senza sapere chi riceverà i messaggi. Un subscriber si iscrive a uno o più argomenti e riceve tutti i messaggi associati a quegli argomenti. Questo decoupling tra i publisher e i subscriber riduce il carico di comunicazione sui dispositivi e migliora la scalabilità della rete.

MQTT opera tipicamente su TCP/IP, utilizzando la porta 1883 per connessioni non cifrate e la porta 8883 per connessioni cifrate utilizzando TLS/SSL. La comunicazione tra i client e il broker è sempre mediata dal broker, che agisce come un punto centrale di raccolta e distribuzione dei messaggi.

MQTT supporta diversi livelli di qualità del servizio (QoS), che determinano l'affidabilità della consegna dei messaggi:

- **QoS 0 (At most once):** Il messaggio viene inviato una sola volta senza conferma di ricezione. È il livello più rapido ma meno affidabile.

- **QoS 1 (At least once):** Il messaggio viene inviato almeno una volta, garantendo che arrivi al destinatario, ma può arrivare più volte in caso di problemi di comunicazione.
- **QoS 2 (Exactly once):** Il messaggio viene garantito di essere ricevuto esattamente una volta, con il più alto livello di affidabilità e una maggiore latenza dovuta alla necessità di più scambi di messaggi per confermare la ricezione.

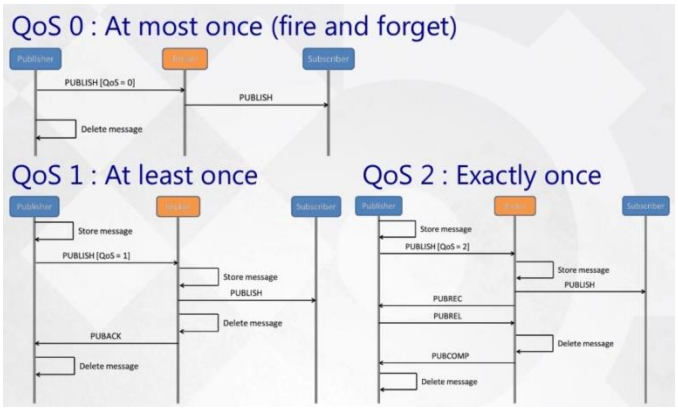

Figura 2.5-confronto livelli di affidabilità QOS

Di default, MQTT non fornisce meccanismi di sicurezza intrinseci, il che rappresenta una limitazione quando viene utilizzato in ambienti pubblici o non protetti. Tuttavia, la sicurezza può essere migliorata utilizzando SSL/TLS per criptare le connessioni tra i client e il broker, fornendo la riservatezza e l'integrità dei dati trasmessi. Inoltre, molti broker MQTT supportano meccanismi di autenticazione basati su nome utente e password o certificati digitali per garantire che solo i dispositivi autorizzati possano pubblicare o sottoscrivere messaggi.

L'utilizzo di MQTT presenta diversi vantaggi significativi:

- **Leggerezza ed efficienza:** MQTT è progettato per trasmettere messaggi con un overhead minimo, rendendolo ideale per dispositivi con risorse limitate e reti con larghezza di banda ridotta.
- **Scalabilità e flessibilità:** Il modello publish/subscribe consente una facile aggiunta e rimozione di dispositivi nella rete senza modificare la configurazione di altri dispositivi, migliorando la scalabilità.
- **Supporto per QoS:** I diversi livelli di QoS offrono una flessibilità nella gestione dell'affidabilità della consegna dei messaggi in base ai requisiti applicativi.

Nonostante i suoi vantaggi, MQTT presenta alcune limitazioni:

- **Dipendenza dal Broker:** L'intero sistema dipende dalla disponibilità e dall'affidabilità del broker, creando un potenziale punto di guasto singolo (single point of failure).
- **Gestione della rete e della banda:** L'efficienza di MQTT può essere compromessa in reti molto congestionate o instabili, dove il sovraccarico di riconnessioni e gestione dei messaggi può influire sulle prestazioni.

In sintesi, sia Modbus che MQTT sono protocolli di comunicazione ampiamente utilizzati nell'IoT industriale, ma presentano differenze significative in termini di architettura, funzionamento e applicazioni. Mentre Modbus, con la sua architettura Client-Server, è una scelta consolidata per molte applicazioni industriali tradizionali, MQTT, con il suo modello publish/subscribe, offre maggiore flessibilità e

scalabilità per le applicazioni IoT moderne che richiedono comunicazioni efficienti e affidabili in ambienti distribuiti.

Tuttavia, entrambi i protocolli presentano limitazioni in termini di sicurezza, che devono essere adeguatamente affrontate per garantire la protezione dei dati nelle reti IoT industriali. A questo proposito, è essenziale considerare l'uso di meccanismi di sicurezza avanzati, come SSL/TLS, per mitigare i rischi associati alla trasmissione dei dati su reti non protette.

Nel prossimo punto (2.3), verrà fornita una panoramica dettagliata di SSL/TLS, il protocollo crittografico comunemente utilizzato per proteggere le comunicazioni su reti TCP/IP, e come questo può essere applicato per migliorare la sicurezza sia di Modbus che di MQTT nelle applicazioni industriali.

2.3 Criptografica SSL/TLS

SSL (Secure Sockets Layer) e il suo successore TLS (Transport Layer Security) sono protocolli crittografici progettati per garantire la sicurezza delle comunicazioni su reti di computer, in particolare su Internet. Questi protocolli forniscono tre caratteristiche fondamentali per la sicurezza delle comunicazioni:

- **Crittografia:** Assicura che i dati trasmessi tra due parti non possano essere letti da terzi non autorizzati. Utilizza algoritmi di crittografia simmetrici e asimmetrici per proteggere la confidenzialità delle informazioni.
- **Integrità dei dati:** Verifica che i dati inviati non siano stati alterati durante la trasmissione, utilizzando funzioni hash crittografiche.
- **Autenticazione:** Conferma l'identità delle parti coinvolte nella comunicazione, tipicamente attraverso l'uso di certificati digitali.

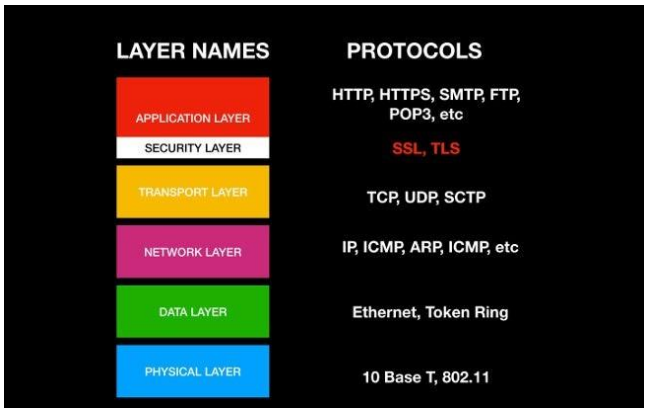

Figura 2.6- applicazione di strato TLS nel modello TCP/IP

SSL/TLS funziona attraverso un processo chiamato handshake, che avviene all'inizio di ogni sessione di comunicazione sicura. Durante l'handshake, le seguenti fasi principali si verificano:

- **Negoziazione della connessione:** Il client e il server negoziano le specifiche di sicurezza, inclusi gli algoritmi crittografici e le chiavi di sessione da utilizzare.
- **Autenticazione del server e del client:** Il server (e facoltativamente il client) presenta un certificato digitale, che viene verificato dall'altra parte per assicurarsi che la connessione sia stabilita con l'entità legittima.

- **Generazione della chiave di sessione:** Viene generata una chiave di sessione simmetrica condivisa utilizzata per cifrare e decifrare i dati scambiati durante la sessione.
- **Trasferimento dei dati cifrati:** Una volta completato l'handshake, i dati vengono trasmessi in modo sicuro utilizzando la crittografia simmetrica.

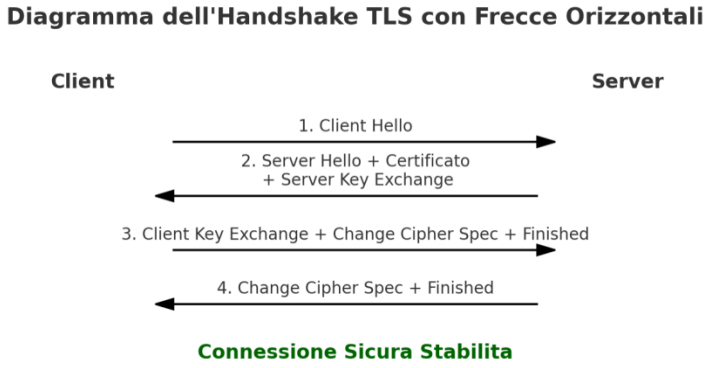

Figura 2.7- esempio handshake TLS

SSL/TLS utilizza una combinazione di crittografia asimmetrica e simmetrica per fornire una comunicazione sicura:

- **Crittografia asimmetrica:** Utilizza una coppia di chiavi (pubblica e privata) per autenticare le parti e negoziare la chiave di sessione. Questo tipo di crittografia è utilizzato durante l'handshake iniziale.
- **Crittografia simmetrica:** Una volta stabilita la connessione, viene utilizzata una chiave simmetrica per crittografare i dati trasferiti. La crittografia simmetrica è più veloce rispetto a quella asimmetrica, rendendola ideale per il trasferimento di grandi quantità di dati.

Avendo uno strato di sicurezza SSL/TLS mi favorisce i seguenti vantaggi:

- **Protezione della riservatezza dei dati:** La crittografia SSL/TLS assicura che solo le parti autorizzate possano leggere i dati trasmessi.
- **Integrità dei dati:** Le funzioni hash utilizzate in SSL/TLS rilevano eventuali modifiche ai dati durante la trasmissione.
- **Autenticazione:** L'uso di certificati digitali garantisce che la comunicazione avvenga tra parti legittime.

Nonostante i vantaggi del SSL/TLS, questo strato mi porta un po' di limitazioni:

- **Overhead di computazione:** L'implementazione di SSL/TLS richiede risorse di calcolo significative, che possono essere un limite per i dispositivi IoT con risorse limitate.
- **Gestione dei certificati:** Richiede la gestione di certificati digitali, inclusa la loro emissione, rinnovo e revoca, che può essere complesso e costoso.
- **Problemi di compatibilità:** Non tutti i dispositivi IoT supportano SSL/TLS nativamente, richiedendo aggiornamenti software o hardware.

L'adozione di SSL/TLS è essenziale per garantire la sicurezza delle comunicazioni IoT industriali, mitigando molti dei rischi associati all'uso di protocolli come Modbus e MQTT in ambienti non protetti. Tuttavia, la sua implementazione richiede una valutazione attenta delle risorse disponibili, della gestione dei certificati e delle esigenze specifiche di sicurezza dell'applicazione. Nel prossimo capitolo, analizzeremo come Modbus TLS e MQTT over SSL/TLS affrontano specificamente le sfide di sicurezza nelle applicazioni IoT industriali.

2.4 Sicurezza in Modbus TLS e MQTT over SSL/TLS

Nel contesto delle comunicazioni IoT industriali, la sicurezza è un fattore cruciale, poiché i dati trasmessi spesso contengono informazioni sensibili o critiche per il funzionamento di sistemi automatizzati. Sia Modbus che MQTT, nella loro forma base, non offrono sufficienti meccanismi di sicurezza, rendendo necessario l'uso di protocolli di crittografia come SSL/TLS per proteggere le comunicazioni. In questa sezione, analizzeremo come Modbus e MQTT utilizzano SSL/TLS per migliorare la sicurezza e quali sono le principali considerazioni da tenere in conto.

Modbus TLS

Modbus TLS è una versione sicura del protocollo Modbus che utilizza Transport Layer Security (TLS) per proteggere i dati trasmessi su reti IP. Modbus TLS introduce la crittografia per garantire la riservatezza dei dati, l'integrità e l'autenticità delle comunicazioni tra i dispositivi master e slave.

Aggiungendo il protocollo TLS al Modbus, aggiunge alle caratteristiche del Modbus TCP/IP (leggi punto 2.2) quelli dello strato SSL/TLS (leggi punto 2.3).

Questa versione di Modbus ha dei vantaggi in più da qualsiasi altra implementazione del protocollo modbus, che sono i seguenti:

- **Sicurezza:** Offre una protezione avanzata contro attacchi comuni come l'intercettazione dei dati (sniffing), l'uomo nel mezzo (man-in-the-middle), e la manipolazione dei dati.
- **Facilità di integrazione:** Modbus TLS può essere implementato sulle reti esistenti senza richiedere cambiamenti significativi nell'infrastruttura fisica.

Ma con questi vantaggi, abbiamo delle sfide da compiere:

- **Overhead delle risorse:** La crittografia e l'autenticazione richiedono risorse di calcolo significative, che possono essere un limite per dispositivi industriali con capacità limitate.
- **Gestione dei certificati:** Richiede la gestione e la manutenzione di certificati digitali, che possono essere complessi e costosi, soprattutto in grandi implementazioni.

MQTT over SSL/TLS

MQTT over SSL/TLS è una versione più sicura del protocollo MQTT che utilizza SSL/TLS per cifrare la connessione tra i client MQTT e il broker. Questo approccio garantisce che i dati pubblicati dai client e distribuiti dal broker siano protetti contro intercettazioni, alterazioni e accessi non autorizzati.

Aggiungendo il protocollo TLS al MQTT, aggiunge alle caratteristiche del MQTT (leggi punto 2.2) quelli dello strato SSL/TLS (leggi punto 2.3).

Questa versione di MQTT ha dei vantaggi in più da qualsiasi altra implementazione del protocollo modbus, che sono i seguenti:

- **Flessibilità e scalabilità:** SSL/TLS aggiunge un livello di sicurezza al protocollo MQTT senza comprometterne la leggerezza e la flessibilità. Può scalare facilmente per supportare migliaia di dispositivi.
- **Protezione contro vari tipi di attacchi:** Protegge contro attacchi di tipo man-in-the-middle, intercettazione di dati, e replay attack, assicurando che le comunicazioni rimangano private e sicure.

Ma con questi vantaggi, abbiamo delle sfide da compiere:

- **Carico computazionale aggiuntivo:** Similmente a Modbus TLS, la crittografia SSL/TLS richiede risorse computazionali significative, il che può rappresentare una sfida per i dispositivi con limitate capacità di elaborazione.
- **Gestione delle chiavi e dei certificati:** L'implementazione sicura di MQTT over SSL/TLS richiede una gestione efficace delle chiavi e dei certificati, che può diventare complessa con un numero elevato di dispositivi.

L'implementazione di SSL/TLS nei protocolli Modbus e MQTT è un passo fondamentale per garantire la sicurezza nelle applicazioni IoT industriali. Sebbene entrambi i protocolli beneficino dell'aggiunta di SSL/TLS in termini di protezione dei dati e autenticazione, la scelta tra Modbus TLS e MQTT over SSL/TLS dipenderà dalle specifiche esigenze dell'applicazione, dalle risorse disponibili e dal livello di sicurezza richiesto.

Nel prossimo capitolo, esploreremo la metodologia adottata per valutare l'efficienza di Modbus TLS e MQTT over SSL/TLS, descrivendo l'ambiente di test e le metriche di valutazione utilizzate per il confronto.

Capitolo 3: Metodologia

Per valutare in modo concreto e accurato le differenze tra i due protocolli, sono state condotte una serie di prove di prestazione. Questi test mirano a fornire una chiara comprensione delle caratteristiche di efficienza di ciascun protocollo, evidenziando le loro prestazioni in diversi scenari applicativi.

3.1 Descrizione dell'ambiente di test

L'ambiente di test è stato progettato per simulare uno scenario industriale, in particolare un sistema di monitoraggio antincendio che potrebbe essere installato in un'azienda o una fabbrica. Il sistema utilizza un sensore ME084 per rilevare la presenza di fumo e inviare un segnale analogico, e un dispositivo ESP8266 con sistema operativo in tempo reale (RTOS) che ha il compito di comunicare con il server centrale del sistema di controllo antincendio.

Questa applicazione richiede che ogni rilevazione del sensore venga inviata almeno una volta al destinatario, garantendo così un livello di affidabilità elevato. In questo contesto, i protocolli Modbus TLS e MQTT SSL/TLS rappresentano due possibili scelte, ognuna delle quali offre specifici vantaggi e limitazioni per questa applicazione.

Le caratteristiche del Software e dei componenti ed utilizzati

Il sensore **ME084** è progettato per il rilevamento del fumo in ambienti industriali, utilizzando una tecnologia a ionizzazione o ottica per garantire un'elevata sensibilità. È in grado di rilevare particelle di fumo da 0.3 a 2.5 micrometri, con un tempo di risposta inferiore a 10 secondi e una precisione del ±5%. Funziona con una tensione di alimentazione tra 3,3V e 5V, con uscita analogica proporzionale alla concentrazione di fumo, consentendo una misurazione continua e in tempo reale. Il sensore opera in un intervallo di temperatura da -10°C a +60°C, con un basso consumo energetico (<10mA), rendendolo adatto per applicazioni IoT alimentate a batteria. Con una vita operativa di oltre 5 anni, il ME084 è ideale per il monitoraggio antincendio in condizioni ambientali difficili, fornendo affidabilità ed efficienza energetica.

L'**ESP8266** è un microcontrollore Wi-Fi altamente integrato, ideale per applicazioni IoT. È dotato di un'unità di elaborazione a 32 bit basata su architettura Tensilica L106 e supporta una frequenza di clock fino a 160 MHz, offrendo prestazioni sufficienti per la maggior parte delle applicazioni embedded. Dispone di 11 GPIO, interfacce I2C, SPI e UART, e una capacità di memoria di 64 KB di RAM e fino a 4 MB di flash, che consentono un'ampia flessibilità nella gestione di sensori e attuatori. L'ESP8266 supporta reti Wi-Fi IEEE 802.11 b/g/n e offre crittografia WPA/WPA2, rendendolo adatto per trasmissioni di dati sicure. È compatibile con vari protocolli di rete, come TCP/IP, HTTP, MQTT e Modbus, ed è gestibile tramite un sistema operativo in tempo reale (RTOS). Consuma poca energia, con modalità di risparmio energetico che riducono l'assorbimento a meno di 10 µA in modalità deep sleep, rendendolo efficiente per applicazioni a batteria. Grazie alle sue dimensioni compatte, al basso costo e alla capacità di connettersi facilmente alle reti Wi-Fi, l'ESP8266 è ampiamente utilizzato in applicazioni IoT per il monitoraggio remoto e l'automazione.

Un **router** domestico è un dispositivo essenziale per creare una rete Wi-Fi locale che consente ai dispositivi di casa di connettersi a Internet e tra loro. I router moderni supportano la tecnologia dual-band,

operando sia sulla banda a 2,4 GHz, per una maggiore copertura, sia sulla banda a 5 GHz, per velocità di trasferimento dati più elevate e minori interferenze. Per garantire la sicurezza della rete, i router includono protocolli avanzati come WPA2 o WPA3, proteggendo i dati da accessi non autorizzati. La presenza di più porte Ethernet permette connessioni cablate stabili per dispositivi come PC e console di gioco, mentre le funzionalità QoS (Quality of Service) ottimizzano il traffico per applicazioni critiche, come lo streaming video e il gaming online. Molti router offrono anche strumenti di controllo parentale e gestione del traffico, nonché il supporto per il protocollo IPv6, garantendo compatibilità futura con nuovi dispositivi e servizi. Inoltre, la facilità di configurazione e gestione, spesso tramite app mobile, insieme agli aggiornamenti firmware automatici, assicura che il router rimanga sicuro e aggiornato, offrendo prestazioni ottimali in ogni momento.

Utilizziamo **un PC** con sistema operativo **Windows** che funge da **Broker** locale per l'applicazione MQTT TLS, un ruolo cruciale che richiede un dispositivo potente e affidabile. Il computer selezionato è un Lenovo IdeaPad Gaming 3-15IMH05 (tipo 81Y4), equipaggiato con un processore Intel Core i5-10300H e una scheda grafica NVIDIA GeForce GTX 1650, ideali per gestire operazioni complesse e traffico di rete intenso. Offre fino a 16 GB di memoria DDR4 per multitasking efficiente e un SSD NVMe da 1 TB per accesso rapido ai dati. Il display da 15.6 pollici Full HD facilita la gestione delle applicazioni e la visualizzazione chiara dei dati, mentre la connettività avanzata attraverso Wi-Fi 6, Bluetooth 5.0, e varie porte, inclusi USB 3.1, USB-C, HDMI, e LAN Ethernet, assicura un'integrazione fluida con altri dispositivi di rete. Operando su Windows 10 Home o Pro, questo sistema fornisce una piattaforma stabile e versatile per mantenere il broker MQTT sempre operativo e reattivo, indipendentemente dal volume delle comunicazioni.

Per la creazione del broker MQTT, è stato utilizzato Mosquitto, un broker MQTT open-source noto per la sua leggerezza e facilità di configurazione. **Mosquitto** supporta pienamente il protocollo MQTT e offre funzionalità come la gestione delle connessioni sicure tramite SSL/TLS, la gestione delle sessioni e la possibilità di scalare per supportare un elevato numero di client simultanei, rendendolo una scelta ideale per applicazioni IoT.

Per creare i certificati utilizzati per la crittografia TLSv1.2, è stato utilizzato **OpenSSL**, una libreria open-source ampiamente utilizzata per la generazione e gestione di certificati digitali. **OpenSSL** consente di creare certificati auto-firmati e file chiave necessari per abilitare le comunicazioni sicure tra i dispositivi, garantendo l'autenticità, la riservatezza e l'integrità dei dati trasmessi.

Per programmare l'ESP8266 è stato utilizzato **Arduino IDE**, un ambiente di sviluppo integrato semplice e intuitivo, ideale per lavorare con microcontrollori. **Arduino IDE** supportano nativamente l'ESP8266 attraverso l'installazione di una libreria dedicata, consentendo di scrivere, compilare e caricare facilmente il codice sull'ESP8266 utilizzando un linguaggio di programmazione basato su **C/C++.** Questa scelta facilita la configurazione del dispositivo per l'uso con protocolli come MQTT e Modbus, semplificando il processo di sviluppo e test delle applicazioni IoT.

3.2 Protocolli e configurazioni

Come evidenziato nel capitolo precedente, i nostri test si concentrano su **Modbus TLS** e **MQTT over SSL/TLS**, entrambi basati sul protocollo **TCP/IP** come strato di trasporto. Per garantire la sicurezza delle comunicazioni, è stata utilizzata la versione **TLSv1.2** come strato di sicurezza. La configurazione prevede che solo il server sia richiesto di presentare un certificato digitale, mentre il client non è soggetto a tale richiesta.

Per il test di **MQTT over TLS** è stata utilizzata la porta standard **8883** per la connessione TCP, mentre **Modbus TLS** è stato configurato per utilizzare la porta **802**.

Il **Modbus** è stato configurato in modo che i valori del sensore vengano memorizzati nel registro 0 dal lato del server (slave). Il client (master) interroga quasi istantaneamente questo registro per ottenere il valore aggiornato del sensore.

MQTT è stato configurato con un livello di affidabilità **QoS 1** sia per il publisher che per il subscriber, garantendo che ogni messaggio venga ricevuto almeno una volta. Inoltre, il broker verifica il nome utente e la password dei client per autenticare gli utenti e garantire la sicurezza della comunicazione.

3.3 Metriche di valutazione

Per valutare l'efficienza dei protocolli **Modbus TLS** e **MQTT over SSL/TLS** nelle applicazioni IoT industriali, sono state definite diverse metriche di valutazione che permettono di confrontare le loro prestazioni in termini di affidabilità, velocità.

Latenza di trasmissione

- Definizione: La latenza misura il tempo totale necessario per trasferire un pacchetto di dati dal client al server (o dal publisher al subscriber) e ricevere una risposta.
- Obiettivo: Determinare quale protocollo garantisce una comunicazione più rapida, un fattore critico nelle applicazioni industriali in cui le decisioni devono essere prese in tempo reale.
- Metodo di Misurazione: La latenza sarà misurata utilizzando strumenti di monitoraggio di rete che registrano il tempo di andata e ritorno (RTT) per ciascun messaggio inviato e ricevuto.

Throughput della rete

- Definizione: Il throughput rappresenta la quantità di dati trasmessi con successo da un punto all'altro della rete in un determinato periodo di tempo, misurata in kilobit per secondo (kbps).
- Obiettivo: Valutare quale protocollo offre una maggiore efficienza nella trasmissione dei dati, considerando anche l'overhead introdotto da TLS.
- Metodo di Misurazione: Il throughput sarà misurato inviando una serie di messaggi di dimensioni variabili attraverso ciascun protocollo e calcolando la quantità di dati trasmessi con successo rispetto al tempo impiegato.

Affidabilità della trasmissione

- Definizione: Questa metrica misura la capacità del protocollo di garantire che i messaggi vengano consegnati correttamente e completamente, utilizzando meccanismi come il QoS (Quality of Service) per MQTT.
- Obiettivo: Confrontare i protocolli in termini di capacità di fornire una consegna affidabile dei dati, essenziale nelle applicazioni critiche.
- Metodo di Misurazione: Saranno analizzati i log di trasmissione per verificare la percentuale di messaggi persi o duplicati e per confermare la consegna di ogni messaggio almeno una volta (per QoS 1 di MQTT).

Tempo di stabilimento della connessione sicura

- Definizione: Questa metrica misura il tempo necessario per stabilire una connessione sicura utilizzando TLS tra il client e il server (o tra il publisher e il broker).
- Obiettivo: Valutare l'efficienza di ciascun protocollo nell'instaurare una connessione sicura, soprattutto in situazioni in cui è necessaria una comunicazione rapida.
- Metodo di Misurazione: Il tempo di stabilimento della connessione sarà misurato dal momento in cui il client inizia il processo di handshake TLS fino alla conferma della connessione sicura stabilita.

Queste metriche forniranno un quadro completo delle prestazioni di **Modbus TLS** e **MQTT over SSL/TLS** nelle applicazioni IoT industriali, permettendo di identificare il protocollo più adatto in base alle esigenze specifiche di affidabilità, efficienza.

Capitolo 4: Implementazione

Per poter valutare ogni caratteristica di questi due protocolli sono stati implementati diversi per avere calcoli precisi.

4.1 Configurazione dei sistemi di test

Calcolo della latenza

Per il protocollo Modbus TLS, la latenza è stata calcolata utilizzando un timer sul client (master) che inizia a contare dall'istante in cui viene inviata la richiesta al server (slave) fino al momento in cui scatta il callback che segnala la ricezione della risposta. In questo modo, il timer consente di misurare con precisione il tempo totale di andata e ritorno della comunicazione.

Per quanto riguarda il protocollo MQTT over TLS, è stata utilizzata una singola scheda ESP8266 che funge sia da publisher che da subscriber al broker MQTT. Il timer sulla scheda inizia a contare dal momento in cui il messaggio viene inviato fino all'arrivo del messaggio di conferma, attivato dal callback di ricezione del messaggio MQTT.

Calcolo del throughput

Per il protocollo Modbus TLS, il master invia richieste per ottenere il valore di dieci registri quasi istantaneamente. Il master utilizza un array per salvare i valori di questi dieci registri non appena vengono ricevuti. Ogni volta che l'ultimo registro nell'array contiene un valore diverso da 0, significa che è arrivata la risposta dal server. A questo punto, il timer scatta per misurare il tempo impiegato a ricevere la lettura di tutti e dieci i registri. Con questo dato, è possibile calcolare il throughput, sapendo che ogni registro occupa 2 byte e che per trasmettere i dati è necessario un pacchetto di 12 byte.

Per il protocollo MQTT TLS, è stato configurato un singolo subscriber su un topic, con 7 publisher che inviano ciascuno un payload di 1 byte di dati. Il subscriber conta il numero di payload ricevuti ogni secondo, permettendo di calcolare il throughput in base al numero totale di payload ricevuti al secondo. Sapendo che, in media, ogni payload corrisponde a un pacchetto di lunghezza 83 byte, è possibile determinare il valore esatto del throughput della comunicazione.

Calcolo affidabilità della trasmissione

Per entrambi i protocolli, Modbus TLS e MQTT over TLS, è stata utilizzata la stessa tecnica. Il server (nel caso di Modbus) o il publisher (nel caso di MQTT) inviano un valore incrementale di un contatore che va da 0 a 10. Dal lato del client (Modbus) o del subscriber (MQTT), è stato implementato un log che monitora i valori ricevuti dall'interfaccia del protocollo. Questo log consente di verificare eventuali messaggi mancanti o duplicati, garantendo l'integrità dei dati trasmessi.

Calcolo tempo di stabilimento della connessione

Per entrambi i protocolli, Modbus TLS e MQTT over TLS, è stata utilizzata la stessa tecnica. Quando il client si connette al server (nel caso di Modbus) o il publisher/subscriber si connette al broker (nel caso di MQTT), il timer viene avviato all'inizio dell'handshake TLS e si arresta al completamento della connessione sicura.

Capitolo 5: Raccolta e Analisi dei Dati

In questo capitolo vengono presentati i risultati delle prove condotte per analizzare i dati raccolti e confrontarli, al fine di ottenere una valutazione chiara delle prestazioni di ciascun protocollo.

5.1 Presentazione dei Dati Raccolti

Sono state condotte diverse prove per valutare le caratteristiche dei protocolli, i cui risultati sono di seguito proprietà:

Latenza

Sono state effettuate oltre 500 prove per ottenere valori il più precisi possibile. Nel grafico seguente è possibile osservare il valore della latenza per ciascun protocollo per i primi cento prove, registrato ogni volta che è stata eseguita una prova.

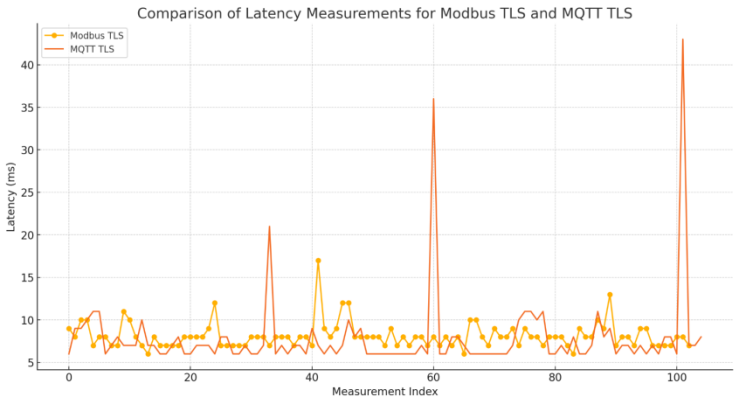

grafico 5.1- grafico confronto latenza

Throughput

Sono state effettuate oltre 500 prove per ottenere valori il più precisi possibile. Nel grafico seguente è possibile osservare il valore dello Throughput per ciascun protocollo per i primi cento prove, registrato ogni volta che è stata eseguita una prova.

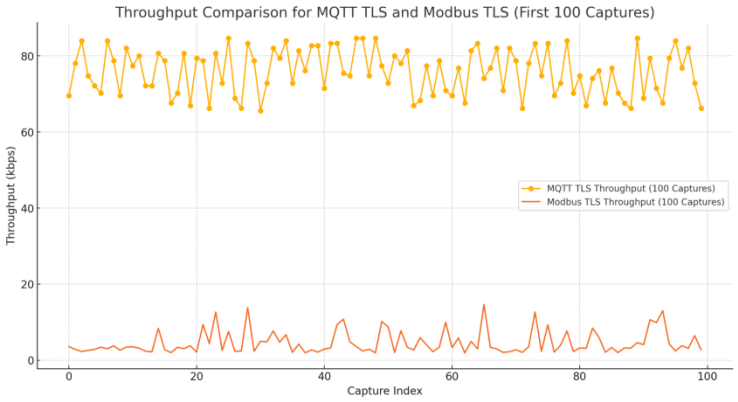

grafico 5.2- grafico confronto throughput

Affidabilità

In questo caso, sono stati analizzati i dati ricevuti dal client/subscriber. Nel grafico riportato è possibile osservare se si sono verificate duplicazioni dei dati o se alcuni dati sono andati persi.

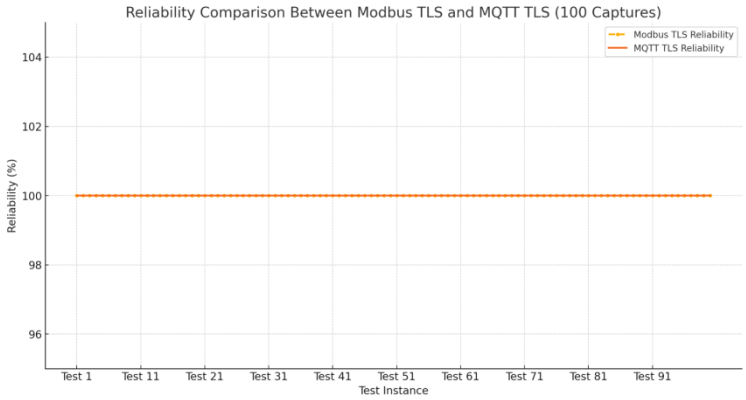

grafico 5.3- grafico confronto affidabilità

Tempo di Stabilimento della Connessione Sicura

Nel grafico seguente è possibile osservare il valore del tempo di Stabilimento della Connessione Sicura per ciascun protocollo per i primi cento prove, registrato ogni volta che è stata eseguita una prova.

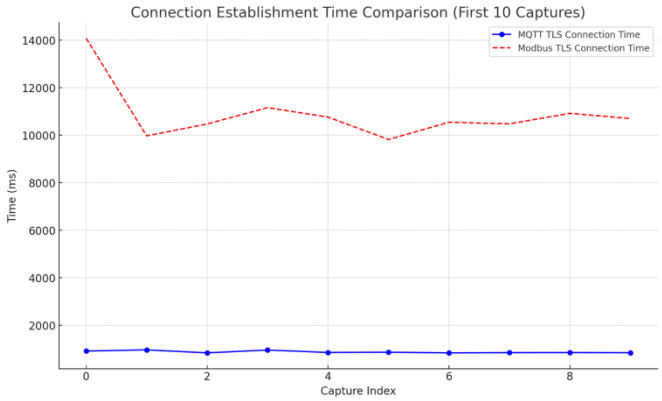

grafico 5.4- grafico confronto tempo di stabilimento della connessione sicura

5.2 Analisi statistica

Dopo aver presentato i risultati delle sperimentazioni, procederemo con un'analisi statistica dettagliata per ciascun protocollo. Questa sezione mira a fornire una comprensione più profonda delle prestazioni e affidabilità di Modbus TLS e MQTT TLS attraverso l'uso di metodi statistici.

5.2.1 Tempo di latenza

	Media	Mediana	Varianza	Deviazione Standard	Range	Quartili
Modbus TLS	8.14 ms	8.00 ms	2.35	1.53 ms	Min: 6 ms, Max: 17 ms	Q1 = 7.00 ms, Q3 = 8.00 ms
MQTT TLS	7.90 ms	7.00 ms	23.48	4.85 ms	Min: 6 ms, Max: 43 ms	Q1 = 6.00 ms, Q3 = 8.00 ms

Tabella 5.1-Risultati dei calcoli statici della latenza

Test-t per campioni indipendenti

- Statistiche t: 0.46
- P-value: 0.646

Variazione e consistenza: Modbus TLS mostra una varianza relativamente bassa (2.35) con una deviazione standard di 1.53 ms, indicando che i tempi di latenza per questo protocollo sono piuttosto consistenti e prevedibili. Al contrario, MQTT TLS presenta una varianza significativamente più alta (23.48) con una deviazione standard di 4.85 ms. Questo suggerisce che la latenza di MQTT TLS è più soggetta a fluttuazioni, il che può essere attribuito a diversi fattori come la gestione del traffico di rete, la congestione o le caratteristiche specifiche dell'implementazione.

Mediane e outliers: La mediana della latenza per Modbus TLS è di 8 ms, leggermente superiore alla mediana di MQTT TLS, che è di 7 ms. Questo suggerisce che più della metà delle misurazioni di latenza per Modbus TLS sono superiori a quelle di MQTT TLS. Tuttavia, l'ampio range di latenze registrate per MQTT TLS, che va da 6 a 43 ms, indica la presenza di outliers significativi. Questi valori estremi potrebbero essere causati da specifiche anomalie nel network o da episodi sporadici di ritardi elevati.

Significatività statistica: Il test t di Studente per campioni indipendenti ha restituito un p-value di 0.646, che è ben al di sopra della soglia convenzionale di 0.05 per rifiutare l'ipotesi nulla. Questo risultato implica che non esiste una differenza statisticamente significativa tra le medie di latenza di Modbus TLS e MQTT TLS. In termini pratici, nonostante le differenze osservate nella consistenza dei dati, non possiamo affermare con certezza che un protocollo sia generalmente più veloce o lento dell'altro basandoci esclusivamente su queste misurazioni.

Implicazioni pratiche: Questi risultati suggeriscono che entrambi i protocolli possono essere adeguati ad applicazioni che richiedono livelli moderati di prestazioni di latenza. Tuttavia, per applicazioni in cui la prevedibilità e la consistenza della latenza sono critiche, Modbus TLS potrebbe essere preferibile data la sua minore variabilità. Per situazioni in cui la tolleranza alla latenza è più flessibile, MQTT TLS potrebbe ancora essere considerato, specialmente se altri fattori come la facilità di implementazione o la funzionalità complessiva sono prioritari.

In conclusione, la scelta tra Modbus TLS e MQTT TLS dovrebbe considerare non solo la media della latenza, ma anche la variabilità e la possibilità di outliers, in base alle esigenze specifiche dell'applicazione e all'ambiente di rete in cui i protocolli devono operare.

5.2.2 Throughput

	Media	Mediana	Varianza	Deviazione Standard	Range	Quartili
Modbus TLS	7.51 kbps	6.60 kbps	10.52	3.24 kbps	Min: 1.95 kbps, Max: 15.89 kbps	Q1 = 5.00 kbps, Q3 = 9.28 kbps
MQTT TLS	75.17 kbps	75.22 kbps	10.96	3.31 kbps	Min: 66.14 kbps, Max: 83.00 kbps	Q1 = 72.63 kbps, Q3 = 77.97 kbps

Tabella 5.2-Risultati dei calcoli statici del throughput

Test-t per campioni indipendenti

- Statistiche t: 145.23
- P-value: Praticamente zero (~4.33e-203)

La differenza nel throughput tra MQTT TLS e Modbus TLS è statisticamente significativa, con un p-value estremamente basso, indicando che le differenze osservate nel throughput medio tra i due protocolli non sono casuali ma piuttosto attribuibili alle caratteristiche intrinseche dei protocolli.

Variazione e consistenza: MQTT TLS mostra una maggiore consistenza nel throughput rispetto a Modbus TLS, come indicato dalla minor variabilità (deviazione standard di 3.31 vs 3.24) e dal range più stretto. Questo suggerisce che MQTT TLS potrebbe essere più affidabile per applicazioni che richiedono una trasmissione di dati continua e prevedibile.

Mediane e outliers: Entrambi i protocolli mostrano una gamma di throughput che riflette variabilità significativa, probabilmente influenzata da fattori come la congestione della rete, la qualità del collegamento, e le implementazioni specifiche. Tuttavia, Modbus TLS ha mostrato un range di throughput estremamente ampio relativo alla sua mediana, indicando che in alcuni casi, le prestazioni possono degradare notevolmente o migliorare.

Implicazioni pratiche: In scenari dove è richiesto un throughput elevato e consistente, MQTT TLS dimostra di essere un'opzione più robusta. Tuttavia, per applicazioni che possono tollerare variazioni maggiori nel throughput e dove altri fattori (come la semplicità di implementazione o requisiti di sicurezza specifici) sono prioritari, Modbus TLS potrebbe ancora essere considerato.

5.2.3 Tempo per stabilire una connessione sicura

	Media	Mediana	Varianza	Deviazione Standard	Range	Quartili
Modbus TLS	883.3 ms	858.5 ms	2176.41	46.65 ms	Min: 842 ms, Max: 969 ms	Q1 = 850.75 ms, Q3 = 909.5 ms
MQTT TLS	10,890.6 ms	10,623.5 ms	1,274,705.64	1,129.03 ms	Min: 9,819 ms, Max: 14,079 ms	Q1 = 10,473.5 ms, Q3 = 10,879.75 ms

Tabella 5.3-Risultati dei calcoli statici del tempo per stabilire una connessione sicura

Test-t per campioni indipendenti

- Statistiche t: -26.57
- P-value: ~0.00 (6.82e-16)

Il test t ha prodotto un p-value estremamente basso, indicando che le differenze nei tempi di connessione tra MQTT TLS e Modbus TLS sono statisticamente significative. Questo significa che, con un alto grado di certezza, possiamo affermare che i tempi di connessione per Modbus TLS sono significativamente superiori rispetto a quelli di MQTT TLS.

Analisi descrittiva: MQTT TLS mostra una variazione minore nei tempi di connessione con una deviazione standard relativamente bassa, indicando una maggiore prevedibilità e stabilità nella connessione. Modbus TLS, al contrario, mostra una grande variabilità nei tempi di connessione come evidenziato dalla sua alta deviazione standard e dal vasto range. Questo suggerisce che il tempo di stabilimento della connessione può essere fortemente influenzato da vari fattori ambientali o di rete.

Implicazioni pratiche: La scelta tra MQTT TLS e Modbus TLS può dipendere significativamente dal contesto di utilizzo. Per applicazioni che richiedono connessioni rapide e affidabili, MQTT TLS può essere la scelta preferibile a causa dei suoi brevi tempi di connessione. Per Modbus TLS, mentre i tempi di connessione sono più lunghi, potrebbe essere adatto per ambienti dove la connettività è meno critica o dove altri aspetti del protocollo (come specifiche di sicurezza o compatibilità con hardware esistente) prevalgono sulla rapidità di connessione.

5.2.4 Affidabilità

Durante la valutazione dell'affidabilità di Modbus TLS e MQTT TLS, entrambi i protocolli hanno dimostrato un'eccellente capacità di mantenere l'integrità dei dati, con un tasso di affidabilità del 100% su 100 tentativi di trasmissione. Questo risultato indica che sotto le condizioni di test controllate, sia Modbus che MQTT sono stati in grado di gestire le comunicazioni senza perdite di dati né duplicazioni, evidenziando la loro robustezza e l'efficacia nella preservazione della completezza e accuratezza dei dati trasmessi.

L'analisi statistica, che ha rivelato una media, una varianza e una deviazione standard di 100%, 0 e 0 rispettivamente, conferma la costanza e l'affidabilità di entrambi i protocolli durante la fase di test. Questa uniformità nell'affidabilità offre agli sviluppatori e agli architetti di sistema la certezza che, per applicazioni dove la precisione dei dati è critica, entrambi i protocolli possono essere considerati affidabili.

Nell'ottica di un'applicazione pratica, tuttavia, è importante considerare che le condizioni reali di rete possono variare notevolmente rispetto a quelle di test. Pertanto, ulteriori test in ambienti di rete più dinamici e con carichi variabili sono consigliati per validare questi risultati. Test supplementari potrebbero includere la valutazione delle prestazioni dei protocolli sotto stress di rete, come alti livelli di traffico o interruzioni di connettività, per assicurare che l'affidabilità osservata sia mantenuta anche nelle condizioni più sfidanti.

In conclusione, mentre i risultati di test attuali forniscono un solido indicatore della robustezza di Modbus TLS e MQTT TLS, la comprensione completa della loro efficacia in scenari operativi reali richiede una considerazione più ampia delle variabili di rete e delle specifiche esigenze applicative.

Capitolo 6: Discussione

Dopo aver esaminato le statistiche e i risultati delle sperimentazioni, questo capitolo discute approfonditamente le modalità di funzionamento dei protocolli esaminati. Attraverso l'analisi dei dati raccolti, siamo in grado di trarre conclusioni informate sulle prestazioni e l'affidabilità di ciascun protocollo, fornendo una visione chiara del loro comportamento nelle diverse condizioni di test.

6.1 Analisi dei risultati

Dopo aver esaminato i risultati delle sperimentazioni, questo capitolo si dedica a discutere in dettaglio le prestazioni osservate per i protocolli Modbus TLS e MQTT TLS, focalizzandosi su latenza, throughput, affidabilità e tempo di connessione. L'analisi dei dati ci permette di trarre conclusioni informate sulle capacità e le limitazioni di ciascun protocollo in contesti specifici.

Dal punto di vista della latenza, si può notare che il protocollo MQTT TLS presenta latenze leggermente inferiori rispetto a Modbus TLS. Tuttavia, è importante considerare che i test sono stati condotti in una rete LAN, dove le distanze tra i dispositivi sono relativamente ridotte. In scenari dove le comunicazioni avvengono su aree più estese, Modbus TLS potrebbe risultare più veloce. Questo perché, a differenza di MQTT TLS che richiede l'intermediazione di un broker, Modbus TLS permette una comunicazione diretta tra i dispositivi.

Dal punto di vista del throughput, si osserva che MQTT TLS gestisce un volume di dati significativamente maggiore rispetto a Modbus TLS. Con una media di 75.17 kbps, il throughput di MQTT TLS è circa dieci volte superiore a quello di Modbus TLS, che è di 7.51 kbps. Questo dato sottolinea l'efficacia di MQTT TLS in ambienti, come quello della nostra sperimentazione, caratterizzati da un elevato flusso di dati. In tali contesti, MQTT TLS dimostra di essere particolarmente efficiente rispetto a Modbus TLS.

Dal punto di vista dell'affidabilità, entrambi i protocolli, Modbus TLS e MQTT TLS, hanno dimostrato eccellenti livelli di affidabilità nel corso delle sperimentazioni. Con un tasso di successo del 100% in tutte le prove, non sono state rilevate perdite di dati né duplicazioni, indipendentemente dal volume o dalla velocità di trasmissione dei dati. Questi risultati sono coerenti con le condizioni di test, in cui le connessioni sono state mantenute stabili e prive di traffico significativo. Tuttavia, da un punto di vista teorico, MQTT TLS può offrire un livello di affidabilità ancora maggiore, grazie alla possibilità di attivare il QoS livello 2, che garantisce la consegna del messaggio almeno una volta senza duplicazioni, rendendolo particolarmente adatto per applicazioni dove la certezza della trasmissione è critica.

Dal punto di vista del tempo di stabilimento della connessione sicura, i risultati delle sperimentazioni evidenziano differenze notevoli tra Modbus TLS e MQTT TLS. Durante i test, MQTT TLS ha registrato tempi di connessione decisamente più brevi, con una media di circa 883 ms, mentre Modbus TLS ha mostrato tempi mediamente più lunghi, circa 10,891 ms. Questa differenza sostanziale può essere spiegata dalla natura del processo di connessione di ciascun protocollo: MQTT TLS utilizza un broker che agisce come intermediario per facilitare e velocizzare l'instaurazione delle connessioni sicure. Al contrario, Modbus TLS opera tramite connessioni dirette tra i dispositivi, un metodo che tende a richiedere più tempo per negoziare la sicurezza e stabilire il collegamento, specialmente in reti di più ampia dimensione o con maggiore complessità strutturale.

In scenari dove il tempo di reazione e la rapidità nella stabilizzazione delle connessioni sono cruciali, come nei sistemi di controllo real-time o in applicazioni che necessitano di risposte immediate a seguito di cambiamenti ambientali o operativi, MQTT TLS offre vantaggi significativi. Pertanto, la scelta del protocollo dovrebbe considerare attentamente il contesto operativo e le esigenze specifiche di tempo di connessione sicura, privilegiando MQTT TLS per esigenze di rapidità e Modbus TLS dove le prestazioni temporali possono essere subordinate ad altri requisiti tecnici o di sicurezza.

6.2 Implicazioni

Quando si progetta un sistema IoT, è essenziale disporre di una guida chiara per decidere quale protocollo utilizzare in base alle specifiche esigenze di ogni elemento della rete. Per sistemi che operano in condizioni di utilizzo di RTOS e una rete LAN con area di connessione limitata, Modbus si rivela particolarmente utile. Questo protocollo è ideale per connessioni con traffico basso dove il throughput è limitato e le connessioni richieste sono tutte P2P (Peer to Peer), senza la necessità di broadcast dei dati. Un vantaggio significativo di Modbus TLS è l'eliminazione del costo del broker, rendendolo economicamente vantaggioso.

D'altra parte, MQTT TLS offre prestazioni superiori in applicazioni che richiedono un alto throughput o la capacità di effettuare broadcast. Un ulteriore vantaggio di MQTT TLS, particolarmente rilevante quando si utilizza un RTOS come l'ESP8266, è la rapidità con cui stabilisce le connessioni, circa dieci volte più veloce rispetto a Modbus TLS. Questa caratteristica lo rende ideale per applicazioni che necessitano di una configurazione veloce e frequente riconnessione.

Capitolo 7: Conclusioni e raccomandazioni

Questo capitolo riassume i principali risultati ottenuti dalla sperimentazione e analisi dei protocolli Modbus TLS e MQTT TLS, offrendo conclusioni chiave e formulando raccomandazioni basate su queste osservazioni. **MQTT TLS** ha dimostrato un throughput superiore e latenze inferiori rispetto a Modbus TLS. Questo rende MQTT TLS particolarmente adatto per ambienti che richiedono trasmissioni di dati rapide e frequenti, come nelle applicazioni di smart city o nella gestione automatizzata.
Modbus TLS, pur avendo latenze più elevate e un throughput inferiore, offre una comunicazione diretta e sicura che è preferibile in ambienti industriali tradizionali dove la semplicità e la robustezza sono prioritarie. Entrambi i protocolli hanno raggiunto un tasso di affidabilità del 100% nelle prove condotte, confermando la loro idoneità per applicazioni critiche dove l'integrità dei dati è fondamentale. **MQTT TLS** ha mostrato tempi di connessione significativamente più rapidi rispetto a Modbus TLS, una caratteristica vantaggiosa per applicazioni che necessitano di frequenti ristabilimenti della connessione o risposte tempestive.

Nonostante i risultati promettenti, ci sono alcune limitazioni che devono essere considerate e che guidano le seguenti raccomandazioni:

Le sperimentazioni sono state condotte in condizioni di laboratorio controllate che potrebbero non replicare completamente le complessità delle reti reali. Pertanto, i risultati potrebbero differire in ambienti più vari o in condizioni di rete stressate.

Valutare attentamente le specifiche esigenze di ogni progetto IoT prima di scegliere il protocollo. Per esempio, considerare MQTT TLS per nuove installazioni che richiedono flessibilità e alte prestazioni di dati, mentre Modbus TLS potrebbe essere più appropriato per aggiornamenti di sistemi esistenti che richiedono una maggiore sicurezza e meno modifiche infrastrutturali.

È essenziale condurre ulteriori test in ambienti operativi reali per verificare i risultati ottenuti e per testare la resilienza dei protocolli sotto diverse condizioni di stress di rete.

Bibliografia

1. Internet e reti : fondamenti / Achille Pattavina. - 3. Ed, Pattavina, Achille

2. IoT fundamentals : networking technologies, protocols, and use cases for the Internet of Things / by David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry Indianapolis : Cisco Press, 2017

3. Sicurezza dei computer e delle reti / William Stallings ; a cura di Alessandra De Paola e Giuseppe Lo Re, Stallings, William, 2022

4. Test t di Student per campioni indipendenti: https://paolapozzolo.it/test-t-student-campioni-indipendenti/

5. Test t: https://it.wikipedia.org/wiki/Test_t