

Article

Experimental Evaluation of Wildcard Subscription Impact on MQTT Broker Performance, Stability, and Security Under Authenticated High-Load IoT Conditions

Nael M. Radwan , Frederick T. Sheldon  and Terence Soule

Department of Computer Science, University of Idaho, Moscow, ID 83844, USA; sheldon@uidaho.edu (F.T.S.); tsoule@uidaho.edu (T.S.)

* Correspondence: nradwan@uidaho.edu

Abstract

MQTT is widely used in Internet of Things (IoT) systems because of its lightweight publish–subscribe architecture and efficient support for resource-constrained devices. Although wildcard subscriptions simplify topic management, their impact on broker performance, stability, and security under authenticated high-load conditions has not been comprehensively investigated. Existing studies typically evaluate routing performance, security mechanisms, or broker scalability independently, leaving a limited understanding of their combined effects. This paper presents a systematic experimental evaluation of wildcard subscription behavior in an authenticated MQTT v5 environment. A controlled testbed employing TLS-based authentication and role-based access control was used to compare exact-topic subscriptions with single-level (+) and multi-level (#) wildcard subscriptions under progressively increasing workloads. Performance was evaluated using end-to-end latency, CPU utilization, throughput, delivery success rate, broker stability, and authorization exposure. The experimental results demonstrate that increasing wildcard-subscription complexity significantly increases routing overhead, resulting in higher latency and CPU utilization while reducing throughput and broker service capacity. Multi-level wildcard subscriptions consistently exhibited the greatest performance degradation and reached broker saturation at lower workload levels than exact-topic subscriptions, demonstrating that wildcard density compresses the broker’s operational stability region. The experiments also show that broad wildcard-based access control policies increase the risk of authorization leakage when improperly configured. These findings demonstrate that wildcard-subscription complexity is a critical determinant of MQTT scalability, broker stability, and security, and provide practical guidance for designing efficient and secure IoT messaging infrastructures.



Academic Editors: Dariusz Rzońca and Tomasz Rak

Received: 7 June 2026

Revised: 13 July 2026

Accepted: 27 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: MQTT; Internet of Things (IoT); wildcard subscriptions; broker stability; routing complexity; backpressure; flow control; authentication and authorization; access control lists (ACLs); performance evaluation; scalability; latency; throughput; security exposure; publish–subscribe systems

1. Introduction

The rapid growth of the Internet of Things (IoT) has transformed the way devices communicate, monitor, and exchange information across diverse application domains, including smart cities, industrial automation, healthcare, environmental monitoring, and intelligent transportation systems. Recent estimates indicate that billions of interconnected

devices continuously generate data that must be collected, processed, and disseminated efficiently while maintaining acceptable levels of security, reliability, and scalability [1]. As IoT deployments continue to expand in scale and complexity, lightweight messaging protocols have become essential components of modern communication infrastructures.

Among the available IoT messaging protocols, the Message Queuing Telemetry Transport (MQTT) protocol has emerged as one of the most widely adopted solutions due to its lightweight design, low bandwidth requirements, asynchronous publish–subscribe communication model, and support for resource-constrained devices [2–5]. Standardized by OASIS and currently defined in MQTT Version 5.0, MQTT enables decoupled communication between publishers and subscribers through a centralized broker responsible for message routing and delivery [6]. This architecture simplifies application development and enhances scalability by separating data producers from consumers in terms of space, time, and synchronization.

A fundamental feature of MQTT is its hierarchical topic namespace, which enables flexible message filtering and subscription management. Subscribers may register interest in specific topics using exact topic names or broader topic filters employing wildcard operators. MQTT defines two wildcard symbols: the single-level wildcard (+), which matches exactly one topic level, and the multi-level wildcard (#), which matches all remaining levels in a topic hierarchy [6]. These mechanisms significantly simplify subscription management in large-scale IoT deployments by allowing a single subscription to represent multiple related topics.

For example, a subscriber interested in temperature measurements from multiple rooms may use the subscription filter `home/+ /temperature`, while a broader filter such as `home/#` can match all topics within an entire hierarchy. Such flexibility reduces administrative complexity and improves application-level scalability. Consequently, wildcard subscriptions are extensively used in smart-home systems, industrial IoT platforms, sensor networks, and cloud-based IoT infrastructures [7,8].

Despite their practical advantages, wildcard subscriptions introduce additional computational complexity at the broker level. Unlike exact-topic subscriptions, which can often be resolved through direct lookup operations, wildcard filters require pattern matching across topic hierarchies and subscription trees. As the number of subscriptions, topic depth, and wildcard density increase, the broker must perform increasingly complex matching operations for each published message. This additional processing burden may increase latency, consume computational resources, reduce throughput, and ultimately affect overall system stability [9–11].

The performance implications of MQTT have received considerable attention in recent years. Several studies have evaluated broker scalability, protocol efficiency, and resource utilization under varying workloads and network conditions [2,10,12–14]. Silva et al. [2] compared MQTT with other IoT communication protocols and demonstrated its suitability for resource-constrained environments. Murthy et al. [11] evaluated MQTT performance under different traffic conditions and observed increasing latency and resource consumption as workload intensity increased. Similarly, Mishra et al. [10] conducted stress-testing experiments on MQTT brokers and reported significant performance degradation near saturation conditions. Recent studies have further investigated scalability challenges associated with high client densities and large-scale IoT deployments [10,12].

Parallel research has examined MQTT security mechanisms, including authentication, authorization, transport-layer security, and access-control enforcement [13–17]. Security enhancements improve confidentiality and access control but introduce additional processing overhead that may affect broker performance and service capacity. Seoane et al. [12] demonstrated that enabling security mechanisms increases latency and computational

requirements in MQTT environments. Likewise, Lakshminarayana et al. [18] highlighted several security challenges associated with MQTT deployments and emphasized the need for balancing security and performance requirements.

Although these studies provide valuable insights into MQTT performance and security, an important limitation remains. Existing research generally investigates protocol performance, security overhead, congestion behavior, and subscription routing independently. In particular, wildcard subscriptions are often discussed from a functional perspective, focusing on their syntax and convenience, while their impact on broker stability and scalability receives comparatively limited attention. Furthermore, most studies evaluate average performance metrics without examining how wildcard density influences stability boundaries, saturation thresholds, and backpressure formation under sustained high-load conditions.

This limitation becomes increasingly important in realistic IoT deployments where multiple factors interact simultaneously. Modern MQTT systems frequently combine secure communication, role-based access control, large topic hierarchies, and extensive wildcard subscriptions while operating under continuously increasing message loads. Under such conditions, wildcard matching overhead may reduce the broker's effective processing capacity, causing earlier saturation and accelerating performance degradation. Consequently, system stability is governed not only by message arrival rates but also by routing complexity introduced by subscription structures.

From a security perspective, wildcard subscriptions introduce additional concerns. Broad wildcard access-control rules may unintentionally expand authorization scope and expose sensitive subtopics beyond intended boundaries. For example, an access-control rule permitting subscriptions to `building/#` may inadvertently provide access to confidential topics nested within the hierarchy. Recent studies on IoT access control have emphasized the risks associated with overly permissive authorization policies and misconfigured wildcard rules. However, little empirical evidence exists regarding the combined impact of wildcard routing complexity and authorization exposure under authenticated MQTT deployments.

These observations reveal a significant research gap. While prior studies have investigated MQTT performance, security, and scalability individually, there is a lack of integrated experimental analysis examining how wildcard subscription patterns influence broker routing complexity, stability boundaries, and security exposure under authenticated high-load conditions. Moreover, the interaction between wildcard density and broker service capacity remains insufficiently understood, particularly regarding its influence on latency growth, CPU utilization, throughput degradation, and saturation behavior.

To address this gap, this article presents a systematic experimental evaluation of wildcard subscription behavior in MQTT-based IoT systems. Using a controlled MQTT v5 testbed with TLS-based authentication and role-based access control, the study compares exact-topic subscriptions, single-level wildcard subscriptions (+), and multi-level wildcard subscriptions (#) under progressively increasing workloads. The investigation focuses on three key dimensions: (i) routing performance, including latency and CPU utilization; (ii) system stability, including throughput degradation, saturation thresholds, and backpressure formation; and (iii) security exposure associated with wildcard-based authorization policies.

The main contributions of this work are summarized as follows:

1. **Comprehensive Wildcard Performance Analysis:** A systematic comparison of exact-topic, single-level wildcard, and multi-level wildcard subscription patterns under authenticated MQTT deployments.

2. **Broker Stability Evaluation:** Experimental quantification of how wildcard density affects latency, throughput, CPU utilization, and broker saturation thresholds under progressive workload escalation.
3. **Security Exposure Assessment:** Analysis of authorization leakage risks associated with wildcard-based access-control configurations and policy misconfigurations.
4. **Integrated Perspective on MQTT Scalability:** Demonstration that wildcard subscription complexity constitutes a first-order factor influencing effective broker service capacity and overall system stability.
5. **Practical Design Guidance:** Recommendations for designing scalable and secure MQTT deployments through controlled wildcard usage and stability-aware subscription management.

The remainder of this article is organized as follows. Section 2 reviews the relevant literature on MQTT communication, wildcard subscription mechanisms, broker scalability, and security considerations in IoT environments. Section 3 presents the experimental methodology, including the MQTT testbed architecture, workload generation process, authentication and authorization configuration, subscription scenarios, and performance metrics. Section 4 describes the experimental results and provides a comparative analysis of exact-topic, single-level wildcard (+), and multi-level wildcard (#) subscription patterns with respect to latency, CPU utilization, throughput, stability boundaries, and security exposure. Section 5 discusses the implications of the findings, highlights practical considerations for scalable and secure MQTT deployments, and compares the results with existing studies. Section 6 presents the limitations of the study and outlines potential directions for future research. Finally, Section 7 concludes the article and summarizes the main contributions and findings regarding the impact of wildcard subscription complexity on MQTT broker performance, stability, and security under authenticated high-load IoT conditions.

2. Related Work and Research Gap

2.1. MQTT Communication and IoT Messaging Systems

The Message Queuing Telemetry Transport (MQTT) protocol has become one of the most widely adopted communication protocols for Internet of Things (IoT) applications because of its lightweight publish–subscribe architecture, low bandwidth requirements, and suitability for resource-constrained devices. MQTT enables asynchronous communication through a centralized broker that decouples publishers and subscribers in terms of space, time, and synchronization. As a result, MQTT has been extensively deployed in smart cities, industrial IoT, healthcare systems, environmental monitoring, and intelligent transportation applications [1,2,4,5].

Several studies have investigated the suitability of MQTT for IoT communication. Al-Fuqaha et al. [1] identified MQTT as a key enabling technology for large-scale IoT systems due to its lightweight operation and scalability. Silva et al. [2] compared MQTT with CoAP and OPC UA and demonstrated that MQTT achieves favorable performance under constrained networking conditions. Radwan and Alves-Foss [5] reviewed recent MQTT developments and highlighted its increasing adoption in modern IoT infrastructures. More recently, Chai et al. [4] proposed a distributed MQTT architecture for industrial IoT environments, emphasizing the growing need for scalable and resilient broker infrastructures.

Although these studies demonstrate the effectiveness of MQTT as an IoT communication protocol, they primarily focus on protocol functionality, messaging efficiency, and deployment architectures rather than the impact of subscription complexity on broker performance and stability.

2.2. Wildcard Subscription Mechanisms in MQTT

One of MQTT's distinguishing features is its support for hierarchical topic structures and wildcard-based subscriptions. MQTT defines two wildcard operators: the single-level wildcard (+), which matches exactly one topic level, and the multi-level wildcard (#), which matches all remaining topic levels in a hierarchy. These mechanisms provide flexibility and simplify subscription management in large-scale IoT deployments.

Industry documentation and broker vendors have extensively described the functionality and advantages of wildcard subscriptions. HiveMQ explains that wildcard filters enable efficient message filtering across large topic trees, reducing subscription management complexity. Similarly, EMQX highlights that wildcard subscriptions facilitate scalable data collection across dynamic IoT environments [12,19].

Despite their practical advantages, wildcard subscriptions inherently increase topic-matching complexity because brokers must evaluate multiple candidate subscriptions for each incoming message. While this routing overhead is acknowledged in technical documentation and implementation guides, rigorous experimental studies quantifying its impact under authenticated and high-load conditions remain limited.

2.3. MQTT Broker Scalability and Performance Evaluation

Broker scalability has become a major research focus as IoT deployments continue to expand. Several studies have examined latency, throughput, resource utilization, and scalability limitations of MQTT brokers under varying workloads.

Murthy et al. [11] evaluated MQTT protocol performance under different traffic conditions and reported increasing latency and resource consumption as message rates increased. Mishra et al. [10] performed stress-testing experiments on MQTT brokers and demonstrated performance degradation near saturation points. Recent scalability studies have further emphasized the importance of broker architecture and resource utilization in determining overall system performance [20].

Research has also shown that broker performance is strongly influenced by processing overhead, subscription management, and message-routing operations. Pipatsakulroj et al. [19] proposed the muMQ broker to improve scalability through efficient utilization of multicore processors, highlighting the central role of routing efficiency in MQTT broker performance [20].

However, most performance studies focus on throughput, latency, and scalability under increasing client counts or publish rates. The specific impact of wildcard subscription density on broker stability boundaries, CPU utilization, and saturation thresholds remains insufficiently investigated [21].

2.4. Security and Access Control in MQTT Environments

Security is a critical concern in MQTT deployments because the protocol was originally designed for lightweight communication rather than comprehensive security enforcement. Modern MQTT systems commonly employ Transport Layer Security (TLS), authentication mechanisms, and Access Control Lists (ACLs) to mitigate security risks [12,14,18].

Lakshminarayana et al. [18] reviewed MQTT security challenges and identified authentication, authorization, and access control as major concerns in IoT environments. Seoane et al. [12] demonstrated that security mechanisms introduce measurable overhead that affects protocol performance. Recent studies have also investigated broker vulnerabilities, authentication schemes, and secure MQTT architectures [14].

Access control becomes particularly challenging when wildcard-based permissions are used. Broad wildcard ACL rules may unintentionally grant access to topic hierarchies beyond their intended scope. Recent work on MQTT authorization and IoT access-control

models has emphasized the risks associated with policy misconfiguration and authorization leakage in hierarchical topic structures [22].

Although security researchers have discussed wildcard-related authorization risks, most studies focus on security implications alone and do not evaluate their interaction with broker routing complexity and performance degradation under heavy workloads [21,22].

Recent Advances in Publish–Subscribe Systems

Recent research has extended Publish/Subscribe (Pub/Sub) communication beyond conventional IoT deployments to support emerging application domains such as intelligent transportation systems, autonomous vehicular networks, industrial cyber–physical systems, edge computing, and large-scale smart city infrastructures. These studies have investigated secure message dissemination, cooperative forwarding, adaptive routing, trust-aware communication, and scalable subscription management to improve communication efficiency and resilience under dynamic operating conditions.

For example, secure and cooperative Pub/Sub schemes have been proposed for autonomous vehicular networks to improve message authenticity, dissemination reliability, and communication efficiency while maintaining scalability in highly dynamic environments. Similarly, recent studies have explored adaptive subscription management, broker optimization, edge-assisted message routing, and distributed Pub/Sub architectures to reduce communication latency and improve resource utilization under increasing network load.

Although these studies significantly advance Publish/Subscribe communication, they generally focus on application-specific routing strategies, distributed architectures, mobility support, or security mechanisms rather than examining the computational impact of wildcard subscription matching in authenticated MQTT brokers. Consequently, comparatively little attention has been devoted to experimentally quantifying how wildcard-subscription complexity influences broker latency, throughput, CPU utilization, operational stability, and authorization behavior under controlled high-load conditions. The present work addresses this gap by providing an integrated experimental evaluation of these factors using a standards-compliant MQTT v5 deployment.

2.5. Research Gap and Novel Contributions

Although MQTT has been extensively investigated with respect to communication efficiency, broker scalability, and security, several important research gaps remain. Most previous studies have examined these aspects independently rather than within a unified experimental framework. Performance-oriented investigations have primarily focused on latency, throughput, and resource utilization under increasing workloads, whereas security studies have concentrated on authentication, encryption, and access control mechanisms. Similarly, research on wildcard subscriptions has largely described their functionality and routing behavior without experimentally quantifying their impact on broker performance, operational stability, and security under authenticated high-load conditions.

Another limitation of the existing literature is the limited understanding of how wildcard subscription complexity affects broker stability. Most studies report average performance metrics while overlooking important operational characteristics such as effective service capacity, saturation thresholds, congestion onset, and backpressure formation. Consequently, the relationship between wildcard subscription topology and long-term broker stability remains insufficiently understood.

A further limitation concerns the interaction between wildcard routing and security enforcement. Although several studies have investigated authentication overhead and Access Control List (ACL) mechanisms, few have experimentally evaluated how wildcard-

based authorization policies influence routing complexity, broker resource consumption, and authorization exposure in realistic authenticated MQTT deployments. Likewise, the security implications of wildcard ACL misconfigurations have generally been discussed qualitatively rather than being supported by quantitative experimental evidence.

Recent Publish/Subscribe (Pub/Sub) research has also explored secure message dissemination, cooperative routing, distributed broker architectures, and application domains such as autonomous vehicular networks, edge computing, and cyber–physical systems. While these studies have advanced scalable and secure Pub/Sub communication, they primarily emphasize application-specific routing strategies, distributed infrastructures, or mobility support. Comparatively little attention has been devoted to experimentally characterizing the combined effects of wildcard subscription complexity, broker performance, operational stability, and authorization behavior in authenticated MQTT environments.

To address these research gaps, the present study proposes a unified experimental framework for evaluating wildcard subscription behavior in authenticated MQTT v5 deployments. The framework combines TLS-based authentication, role-based Access Control Lists (ACLs), controlled workload escalation, and three subscription configurations—exact-topic, single-level wildcard (+), and multi-level wildcard (#)—within a reproducible experimental testbed. By maintaining identical hardware, software, network, authentication, and workload conditions across all experiments, the proposed methodology isolates wildcard subscription complexity as the primary independent variable and enables a comprehensive assessment of its impact on broker performance, stability, and security.

The principal contributions of this work are summarized as follows:

- **Integrated Performance–Stability–Security Evaluation:** This study simultaneously evaluates broker performance, operational stability, and security within a unified authenticated MQTT framework, whereas previous studies have typically examined these aspects independently.
- **Quantitative Stability Characterization:** The experiments demonstrate that increasing wildcard subscription complexity reduces effective broker service capacity, accelerates congestion onset, compresses the operational stability region, and results in earlier broker saturation.
- **Realistic Authenticated Experimental Framework:** A reproducible MQTT v5 experimental platform employing TLS-based authentication, role-based ACL enforcement, controlled workload generation, and statistical validation is developed to evaluate broker behavior under realistic deployment conditions.
- **Quantitative Security Evaluation:** The study experimentally measures authorization exposure resulting from wildcard-based ACL configurations and demonstrates how routing complexity influences both broker processing overhead and security enforcement.
- **Practical Design Recommendations:** Based on the experimental findings, evidence-based recommendations are provided for wildcard subscription design, broker capacity planning, workload management, and secure ACL configuration to support scalable and secure IoT messaging infrastructures.

Rather than presenting another benchmark of MQTT performance, this work demonstrates that wildcard subscription complexity is a primary determinant of broker scalability, operational stability, and security in authenticated MQTT deployments. By integrating these dimensions within a single reproducible experimental framework, the study fills an important gap in the existing literature and provides practical guidance for the design and deployment of next-generation MQTT-based Internet of Things (IoT) communication systems.

3. Materials and Methods

3.1. Experimental Objective

The objective of this study is to experimentally evaluate the impact of wildcard subscription patterns on MQTT broker performance, stability, and security under authenticated high-load Internet of Things (IoT) environments. Specifically, the experiment investigates how increasing subscription complexity influences message-routing overhead, broker resource utilization, throughput, latency, and stability boundaries.

The study focuses on three subscription categories defined by the MQTT specification:

- Exact-topic subscriptions.
- Single-level wildcard subscriptions (+).
- Multi-level wildcard subscriptions (#).

These subscription models are evaluated under progressively increasing publish rates while maintaining identical authentication, authorization, workload generation, and network conditions. The goal is to isolate wildcard subscription complexity as the primary independent variable and quantify its effect on broker behavior.

$$C_{route} \propto S \cdot D \cdot W \quad (1)$$

where:

- S = number of subscriptions;
- D = topic hierarchy depth;
- W = wildcard density.

To facilitate understanding of wildcard subscription complexity, we introduce a conceptual routing-complexity model. The model is intended to motivate the experimental evaluation rather than serve as a predictive analytical model. It describes the expected qualitative relationship between broker routing complexity and three principal factors: the number of active subscriptions (S), topic hierarchy depth (D), and wildcard density (W).

Equation (1) should therefore be interpreted as a conceptual framework rather than an empirically validated predictive equation. The experiments presented in this paper evaluate the combined influence of these factors under realistic MQTT deployments but do not independently vary each parameter sufficiently to estimate the independent contribution of S , D , and W . A rigorous analytical validation of the proposed relationship remains an important direction for future research.

The experimental results are consistent with the qualitative behavior predicted by the conceptual routing-complexity model. In particular, increased wildcard usage was associated with higher routing overhead, increased latency, and earlier broker saturation. However, the experiments were designed to evaluate realistic integrated MQTT workloads rather than independently manipulating each model variable. Consequently, the findings should be interpreted as qualitative support for the proposed model rather than as direct parameter estimation.

The routing complexity of MQTT subscription matching increases with the number of subscriptions, topic depth, and wildcard density. As wildcard breadth increases, the broker evaluates a larger candidate matching space, increasing routing overhead.

3.2. Experimental Testbed and Reproducibility

The experiments were conducted using a two-node MQTT testbed consisting of one dedicated broker node and one client-generation node. The nodes were connected through an isolated wired local-area network to minimize uncontrolled latency, packet loss, and background traffic. The same hardware, software, broker configuration, security

configuration, topic namespace, and workload schedule were used for all subscription scenarios. Only the subscription-filter type was changed between experiments.

3.2.1. Broker Node

The broker node was configured using Eclipse Mosquitto version 2.0.18 and served as the central MQTT message broker throughout all experiments. The hardware and software specifications of the broker node are summarized below:

- MQTT broker: Eclipse Mosquitto Version 2.0.18 (Eclipse Foundation, Ottawa, ON, Canada);
- MQTT protocol: MQTT Version 5.0 (OASIS Open, Burlington, MA, USA);
- Operating system: Ubuntu 22.04.4 LTS (Canonical Ltd., London, UK);
- Linux kernel: 6.8.0-60-generic;
- Processor: Intel® Core™ i7-4712MQ CPU @ 2.30 GHz (Intel Corporation, Santa Clara, CA, USA);
- Physical cores: 4;
- Logical processors: 8;
- Processor frequency: 2.30 GHz;
- Memory: 8 GB RAM;
- Storage: Seagate ST500LM000-1EJ162 HDD (Seagate Technology LLC, Fremont, CA, USA);
- Broker persistence: Enabled;
- Maximum inflight messages: 20;
- Maximum queued messages: 1000;
- Maximum packet size: Default Eclipse Mosquitto configuration;
- Keep-alive interval: 60 s;
- TLS listener port: 8883;
- Logging level: Information.

The Eclipse Mosquitto broker was configured to support the MQTT Version 5.0 protocol with mutual Transport Layer Security (TLS) authentication, X.509 client certificates, and role-based Access Control List (ACL) enforcement. Broker configuration files, TLS certificates, ACL policies, and resource-management parameters remained unchanged throughout all experimental runs to ensure consistent operating conditions and experimental reproducibility.

Broker resource utilization was continuously monitored during the experiments. CPU utilization, memory consumption, network activity, and process statistics were collected using the following monitoring tools:

- Prometheus: Version 3.4.1;
- node_exporter: Version 1.9.1;
- Sampling interval: 1 s.

The Mosquitto broker was configured with the information logging level to record connection establishment, client authentication, subscription requests, broker events, client disconnections, and MQTT Version 5.0 reason codes generated during message processing. These logs were used to verify broker behavior, monitor authorization decisions, and assist in diagnosing protocol events observed during the experimental evaluation.

3.2.2. Client Node

The client-generation node was used to execute the MQTT publisher and subscriber applications during all experimental scenarios. The hardware and software configuration of the client node is summarized below.

- Operating system: Ubuntu 24.04.2 LTS (Canonical Ltd., London, UK);
- Linux kernel: 6.8.0-60-generic;
- Processor: 13th Generation Intel® Core™ i5-1335U (Intel Corporation, Santa Clara, CA, USA);
- Physical cores: 10;
- Logical processors: 12;
- Memory: 16 GB RAM;
- Compiler: GNU G++ 13.3.0 (GNU Project, Free Software Foundation, Boston, MA, USA);
- Eclipse Paho MQTT C library: Version 1.3.14 (Eclipse Foundation, Ottawa, ON, Canada);
- Eclipse Paho MQTT C++ library: Version 1.5.3 (Eclipse Foundation, Ottawa, ON, Canada);
- Build mode: Release;
- Timestamp source: std::chrono::steady_clock (ISO/IEC C++ Standard Library);
- Timestamp resolution: 1 ms.

The MQTT publisher and subscriber applications were developed in C++ using the Eclipse Paho MQTT C++ asynchronous client library. Each client established an independent authenticated MQTT Version 5.0 session with the broker using mutual TLS authentication. During each experiment, the client applications recorded the message identifier, MQTT topic, publication timestamp, reception timestamp, delivery status, Quality of Service (QoS) level, publish rate, and MQTT Version 5.0 reason code for subsequent performance and statistical analysis.

The client workload configuration used throughout the experiments is summarized in Table 1.

Table 1. MQTT client configuration.

Parameter	Value
Number of publishers	20
Number of subscribers	20
Total authenticated MQTT sessions	40
Subscriptions per subscriber	10
Total active subscriptions	200
Topic-selection strategy	Uniform random
Authentication mechanism	Mutual TLS (X.509 certificates)

3.2.3. Network Configuration

The broker and client nodes were connected through a dedicated wired Gigabit Ethernet network to provide a controlled communication environment and minimize the influence of external network factors on the experimental results. The network configuration used throughout all experiments is summarized below.

- Link type: Gigabit Ethernet (1000BASE-T);
- Nominal link speed: 1 Gbit/s;
- Network switch: Dedicated Gigabit Ethernet switch;
- IP configuration: Static IPv4 addressing;
- Broker IP address: 192.168.1.10 (private network);
- Client IP address: 192.168.1.20 (private network);
- Maximum Transmission Unit (MTU): 1500 bytes;
- Measured baseline round-trip time (RTT): 0.2–1.0 ms;

- Measured packet-loss rate: 0%;
- Background network traffic: None.

Before each experimental run, network connectivity was verified using Internet Control Message Protocol (ICMP) echo requests and an MQTT connectivity test consisting of the CONNECT, SUBSCRIBE, and PUBLISH operations. These preliminary checks confirmed correct communication between the broker and client nodes before workload generation commenced.

To ensure experimental repeatability and eliminate uncontrolled network effects, all experiments were conducted on an isolated local-area network (LAN). No wireless communication, Internet routing, virtual private network (VPN), or external traffic sources were present during data collection. Consequently, variations in the measured performance metrics can be attributed primarily to the evaluated MQTT subscription configurations rather than fluctuations in the underlying network infrastructure.

3.3. Broker Security and Access-Control Configuration

Authentication was implemented using mutual Transport Layer Security (TLS) with X.509 digital certificates to ensure secure communication between MQTT clients and the broker. The security configuration remained unchanged throughout all experimental scenarios to ensure consistent authentication conditions and experimental reproducibility. The authentication parameters are summarized below.

- TLS version: TLS 1.3;
- Cipher suite: TLS_AES_256_GCM_SHA384;
- Certificate type: X.509 RSA (2048-bit);
- Certificate Authority (CA): Locally generated experimental Certificate Authority;
- Certificate validation: Required;
- Anonymous client access: Disabled;
- Username/password authentication: Disabled (mutual TLS authentication only);
- Session reuse: Disabled;
- OCSP stapling: Disabled.

Authorization was implemented using Eclipse Mosquitto Access Control Lists (ACLs). The ACL configuration enforced role-based permissions for MQTT topic subscriptions and publications. Exact-topic and wildcard subscription permissions were evaluated independently to quantify the impact of wildcard authorization policies on broker performance and security.

Experimental scenarios W1–W4 employed correctly configured ACL policies that enforced the intended access restrictions. In contrast, Scenario W5 deliberately introduced a controlled wildcard ACL misconfiguration to evaluate the potential for authorization exposure resulting from overly permissive wildcard subscription rules. This approach enabled the experimental assessment of security risks associated with wildcard-based authorization while maintaining identical network, workload, and authentication conditions across all scenarios.

Representative ACL rules used in the experiments are shown below.

```

user subscriber_exact
topic read site/building1/floor1/room1/temperature
user subscriber_plus
topic read site/+/floor1/room1/temperature
user subscriber_hash
topic read site/#

```

The complete Mosquitto configuration files, including the ACL definitions and certificate configuration, are provided as Supplementary Materials to facilitate independent reproduction of the experimental environment.

3.4. Topic Namespace and Subscription Parameters

A hierarchical MQTT topic namespace was designed to represent a realistic Internet of Things (IoT) deployment while providing a controlled environment for evaluating wildcard subscription behavior. The namespace followed the structure:

site/<building>/<floor>/<room>/<sensor>/<data-type>

This hierarchy consists of six topic levels, representing the site, building, floor, room, sensor type, and data type, respectively. The complete topic hierarchy used in the experiments comprised one site, five buildings per site, four floors per building, ten rooms per floor, and four sensor types per room, resulting in a total of 800 unique MQTT topics.

The experimental topic-generation parameters were as follows:

- Number of sites: 1;
- Buildings per site: 5;
- Floors per building: 4;
- Rooms per floor: 10;
- Sensor types per room: 4;
- Data types per sensor: 1;
- Total unique MQTT topics: 800;
- Maximum topic hierarchy depth: 6 levels;
- Number of publishers: 20;
- Number of subscribers: 20;
- Subscriptions per subscriber: 10.

Four subscription configurations were evaluated during the experiments:

- S0: Exact-topic subscriptions only;
- S+: Single-level wildcard (+) subscriptions;
- S#: Multi-level wildcard (#) subscriptions;
- Smix: A controlled combination of exact-topic, single-level wildcard, and multi-level wildcard subscriptions.

Wildcard subscription density was defined as:

$$W = \frac{N_{wildcard}}{N_{total}}$$

where $N_{wildcard}$ denotes the number of subscriptions containing at least one wildcard operator, and N_{total} represents the total number of active subscriptions.

Five wildcard-density levels were evaluated: 0%, 25%, 50%, 75%, and 100%. Throughout all experiments, the number of publishers, subscribers, active subscriptions, and logical topic interests remained constant. This design ensured that wildcard subscription complexity was the only independent experimental variable, eliminating potential confounding effects caused by changes in subscriber population or message fan-out.

Publication topics were selected from the predefined MQTT namespace using a deterministic pseudorandom number generator initialized with a fixed seed for each experimental repetition. Topics were selected according to a uniform probability distribution so that every topic had an equal probability of being published. The same topic universe, random-seed sequence, and publication distribution were maintained across all experimental scenarios to ensure identical workloads and isolate the effects of wildcard subscription complexity.

To further ensure experimental reproducibility, the logical topic hierarchy, publication schedule, workload intensity, and topic-selection strategy remained unchanged throughout the study. Only the subscription configuration (S0, S+, S#, or Smix) was modified between experiments.

Table 2 summarizes the MQTT topic hierarchy used throughout the experimental evaluation.

Table 2. MQTT topic hierarchy used in the experiments.

Hierarchy Level	Number of Elements
Sites	1
Buildings per site	5
Floors per Building	4
Rooms per Floor	10
Sensor Types per Room	4
Data Types per Sensor	1
Total unique MQTT topics	800
Maximum topic hierarchy depth	6 levels
Topic-Selection Distribution	Uniform
Random Seed	Fixed for each experimental repetition

3.5. Workload Generation Parameters

The experimental workload was generated using 20 authenticated MQTT publishers and 20 authenticated MQTT subscribers. Each publisher established an independent MQTT Version 5.0 session with the broker and generated messages according to a predefined aggregate publish rate. Each subscriber maintained 10 active subscriptions, depending on the evaluated subscription scenario: exact-topic (S0), single-level wildcard (S+), multi-level wildcard (S#), or mixed subscriptions (Smix). The numbers of publishers, subscribers, and subscriptions per subscriber remained constant throughout all experiments so that wildcard subscription complexity was the only independent experimental variable.

To ensure reproducibility, workload intensity was defined using explicit aggregate publish rates rather than qualitative descriptions. Three workload regions were evaluated:

- Low load: 10–100 messages/s;
- Medium load: 110–200 messages/s;
- High load: 210–300 messages/s.

The aggregate publish rate was increased from 10 to 300 messages/s in increments of 10 messages/s. Each workload stage lasted 60 s, preceded by a 30 s warm-up period and followed by a 30 s cool-down period. Every experimental condition was repeated 10 times, and the reported results represent the arithmetic mean of these repetitions together with 95% confidence intervals.

The aggregate publish rate reported throughout this paper represents the combined traffic generated by all 20 publishers. Consequently, for an aggregate workload of λ messages/s, each publisher generated traffic at an average rate of $\lambda/20$ messages/s.

Table 3 summarizes the workload categories used in the experiments.

The workload was generated using a custom C++ application developed with the Eclipse Paho MQTT C++ asynchronous client library. Publication topics were selected from the predefined MQTT namespace using a deterministic pseudorandom number generator initialized with a fixed seed for each experimental repetition. A uniform probability distribution was used to ensure that every topic had an equal probability of being selected.

The same random-seed sequence, topic universe, client population, workload schedule, and subscription-independent parameters were maintained across all experiments to ensure fair comparisons.

Table 3. Workload intensity categories.

Workload Category	Aggregate Publish Rate (Messages/s)	Purpose
Low	10–100	Baseline broker performance under light traffic
Medium	110–200	Transition toward increased broker utilization
High	210–300	Broker operation under high-load and near-saturation conditions

The workload-generation parameters are summarized below.

- Payload size: 256 bytes;
- QoS levels: QoS 0 and QoS 1 (evaluated separately);
- Retained messages: Disabled;
- Clean Start: Enabled;
- Session Expiry Interval: 0 s;
- Message Expiry Interval: Not configured;
- Number of publishers: 20;
- Number of subscribers: 20;
- Initial aggregate publish rate: 10 messages/s;
- Maximum aggregate publish rate: 300 messages/s;
- Publish-rate increment: 10 messages/s;
- Warm-up period: 30 s;
- Measurement period: 60 s;
- Cool-down period: 30 s;
- Replications: 10 per experimental condition;
- Random seeds: One fixed seed for each replication;
- Inter-message timing model: Constant-rate deterministic transmission;
- Topic-selection distribution: Uniform.

The aggregate publish-rate schedule is defined as

$$\Lambda = \{ \lambda_1, \lambda_2, \dots, \lambda_K \}$$

where each publish rate λ_K was maintained for 60 s before advancing to the next workload level. The same workload schedule and random-seed sequence were applied to every subscription scenario.

Table 4 summarizes the timing configuration used during the experimental evaluation.

During each experimental repetition, the broker state was reset, the identical broker and ACL configuration was reloaded, the pseudorandom number generator was initialized with the corresponding fixed seed, and the publisher and subscriber applications were started. Performance measurements were collected only during the 60 s measurement interval after completion of the warm-up period. For every transmitted message, the client applications recorded the message identifier, topic, publication timestamp, reception timestamp, delivery status, MQTT Version 5.0 reason code, and corresponding broker resource utilization metrics.

Table 4. Experimental timing configuration.

Parameter	Value
Warm-up duration	30 s
Measurement duration	60 s
Cool-down duration	30 s
Total duration per workload stage	120 s
Number of workload stages	30
Experimental repetitions	10

To eliminate residual effects between experiments, retained publishing was disabled, Clean Start was enabled, and the Session Expiry Interval was set to 0 s, ensuring that every experimental repetition began with a new MQTT session. The MQTT Version 5.0 Message Expiry Interval property was not configured because all messages remained valid throughout the active experimental session.

3.6. Experimental Scenarios

Five experimental scenarios were designed to evaluate the effects of MQTT subscription configurations on broker performance, scalability, and security under authenticated communication. All scenarios employed mutual TLS authentication and differed only in the subscription configuration and authorization policy applied during each experiment. This design ensured that any observed differences in performance or security could be attributed to the evaluated subscription strategy rather than variations in the experimental environment.

Table 5 summarizes the experimental scenarios considered in this study.

Table 5. Experimental Scenarios.

Scenario	Authentication	Subscription Configuration	Purpose
W1	Mutual TLS	Exact-topic subscriptions	Baseline routing performance
W2	Mutual TLS	Single-level wildcard ('+') subscriptions	Evaluation of single-level wildcard matching overhead
W3	Mutual TLS	Multi-level wildcard ('#') subscriptions	Evaluation of multi-level wildcard matching overhead
W4	Mutual TLS	Mixed exact-topic, '+', and '#' subscriptions	Evaluation of a realistic heterogeneous subscription environment
W5	Mutual TLS	Mixed subscriptions with controlled ACL misconfiguration	Evaluation of authorization exposure caused by overly permissive wildcard access

Across all scenarios, the hardware configuration, operating systems, broker settings, client population, MQTT Version 5.0 protocol configuration, topic namespace, payload size, Quality of Service (QoS) levels, workload schedule, random-seed sequence, and observation intervals were kept identical. Consequently, the subscription configuration and, in the case of W5, the authorization policy were the only experimental variables, thereby ensuring fair comparisons and supporting the reproducibility of the reported results.

3.7. Data Collection and Metric Computation

Performance data were collected continuously throughout each experimental run after completion of the warm-up period. For every transmitted MQTT message, the publisher and subscriber applications recorded the message identifier, publication timestamp, recep-

tion timestamp, delivery status, MQTT Version 5.0 reason code, and the corresponding topic. In addition, broker resource utilization, including CPU usage, memory consumption, and network activity, was monitored throughout the experiments.

The primary performance metrics evaluated in this study were end-to-end latency, throughput, delivery success rate, and broker resource utilization.

The end-to-end latency for message i was computed as

$$L_i = t_i^{recv} - t_i^{send}$$

where t_i^{recv} and t_i^{send} denote the publication and reception timestamps, respectively.

The throughput during an observation interval T was calculated as

$$\theta = \frac{N_{delivered}}{T}$$

where $N_{delivered}$ is the number of successfully delivered messages during the observation interval.

The delivery success rate was computed as

$$D = \frac{N_{delivered}}{N_{published}} \cdot 100\%$$

where $N_{published}$ is the total number of published messages.

Broker CPU utilization was sampled at 1 s intervals and summarized using the mean, standard deviation, and 95% confidence interval for each workload level and experimental scenario. End-to-end latency was characterized using the mean, median, 95th percentile (P95), and 99th percentile (P99) to capture both average performance and tail latency under increasing workload.

A broker was considered to have reached saturation when one or more of the following conditions persisted for three consecutive measurement intervals:

- P99 end-to-end latency exceeded 500 ms.
- Throughput no longer increased despite an increase in the offered publish rate.
- Message queue utilization exceeded 80% of the configured queue capacity.
- Delivery success rate fell below 95%.
- MQTT Version 5.0 backpressure- or quota-related reason codes were observed.

These criteria provide a comprehensive indication of broker saturation by combining application-level performance metrics, broker resource utilization, message delivery reliability, and protocol-level backpressure signals.

3.8. Reproducibility Materials

To facilitate the independent reproduction of the experimental results, the artifacts used in this study include the MQTT broker configuration, Access Control List (ACL) policies, topic-generation configuration, publisher and subscriber source code, workload-generation scripts, fixed random-seed values, raw message logs, broker resource-monitoring data, statistical-analysis scripts, a software version manifest, and documentation describing the installation and execution procedures.

The software and configuration files were maintained under version control throughout the study to ensure consistency across all experimental repetitions. The fixed random-seed values, identical workload schedule, broker configuration, and topic-generation parameters were preserved for every experimental scenario to guarantee repeatability of the reported results.

The experimental artifacts and supporting documentation are available from the corresponding author upon reasonable request. Where permitted by institutional and

publication policies, these materials will also be made publicly available through an online repository after publication to facilitate independent verification and future research.

Figure 1 illustrates the hierarchical MQTT topic namespace and the wildcard subscription mechanisms evaluated in this study. Single-level (+) wildcard subscriptions match exactly one hierarchy level, whereas multi-level (#) wildcard subscriptions match all remaining hierarchy levels. As wildcard usage increases, the broker performs a greater number of topic-matching operations and evaluates a larger routing space during message delivery, potentially affecting performance, scalability, and authorization behavior.

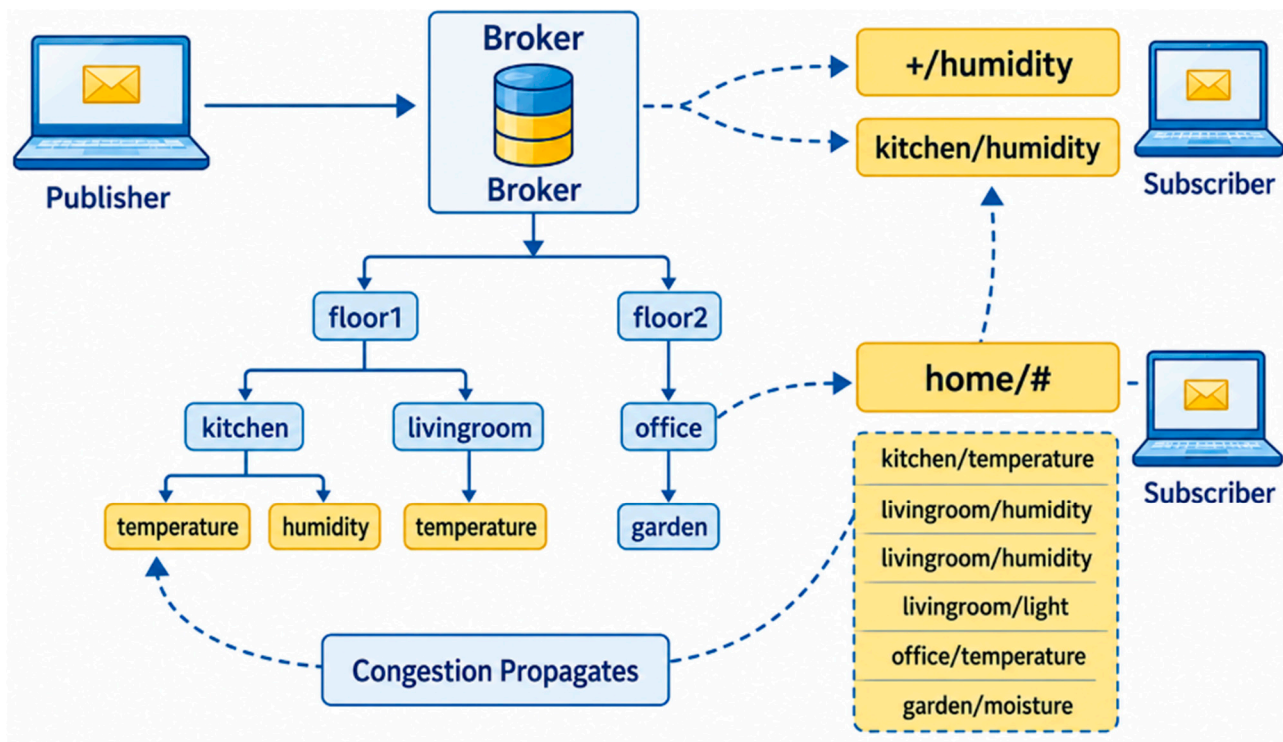


Figure 1. Topic Hierarchy and Wildcard Subscription Propagation in MQTT.

The # symbol is the MQTT multi-level wildcard. It matches all remaining topic levels starting from its position in the topic hierarchy. In the example shown in Figure 1, the subscription *home/#* receives all messages published under the *home* topic and any of its subtopics, regardless of their depth (e.g., *home/kitchen/temperature*, *home/livingroom/humidity*, and *home/office/garden/moisture*). This wildcard enables broad topic subscriptions but may increase broker processing overhead under high-load conditions because a single subscription can match many published topics.

3.9. Statistical Analysis

Each experimental condition was repeated ten independent times to ensure the reliability and repeatability of the reported results. Statistical analyses were performed separately for end-to-end latency, broker CPU utilization, throughput, delivery success rate, and broker saturation. For each workload level and subscription configuration, the arithmetic mean, standard deviation, median, interquartile range (IQR), and 95% confidence interval (CI) were calculated.

The normality of each performance metric was assessed using the Shapiro–Wilk test, while the homogeneity of variances was evaluated using Levene’s test. When both assumptions were satisfied, differences among the exact-topic (S0), single-level wildcard (S+), multi-level wildcard (S#), and mixed-subscription (Smix) configurations were analyzed

using one-way analysis of variance (ANOVA). Significant ANOVA results were followed by Tukey's Honestly Significant Difference (HSD) test for pairwise comparisons.

When the assumptions required for parametric analysis were not satisfied, the Kruskal–Wallis test was used as the non-parametric alternative. Significant Kruskal–Wallis results were followed by Dunn's multiple-comparison test, and the resulting p -values were adjusted using the Holm correction to control the family-wise error rate associated with multiple statistical comparisons.

Statistical significance was evaluated at a significance level of $\alpha = 0.05$. In addition to p -values, effect sizes were reported to distinguish statistical significance from practical significance. For parametric analyses, eta squared (η^2) was reported as the measure of effect size, whereas epsilon squared (ϵ^2) was reported for Kruskal–Wallis analyses. Pairwise comparisons included Cohen's d for parametric tests and the rank-biserial correlation for non-parametric tests.

All figures present the arithmetic mean together with 95% confidence intervals computed from the ten independent experimental repetitions. These confidence intervals are shown as error bars in the performance figures to illustrate the variability of the measured results across repeated experiments.

Statistical analyses were performed using Python 3.12, SciPy 1.14, Statsmodels 0.14.2, and scikit-posthocs 0.10.

4. Experimental Results and Analysis

4.1. Overview of Experimental Results

This section presents the experimental evaluation of wildcard subscription behavior in authenticated MQTT Version 5.0 environments under progressively increasing workloads. The objective of the experiments is to quantify the impact of wildcard subscription complexity on broker performance, scalability, stability, and security while maintaining identical hardware, software, workload, and authentication configurations across all experimental scenarios.

Three subscription configurations were evaluated throughout the study:

- S0: Exact-topic subscriptions;
- S+: Single-level wildcard (+) subscriptions;
- S#: Multi-level wildcard (#) subscriptions.

Unless otherwise stated, all experiments were conducted using identical broker configurations, authenticated MQTT sessions, client populations, topic hierarchies, workload schedules, and random-seed sequences. Consequently, the subscription configuration was the primary independent variable, allowing the observed differences in performance to be attributed directly to wildcard routing complexity.

Each experimental condition was repeated ten independent times. The reported results correspond to the arithmetic mean of the ten repetitions, while variability is represented using 95% confidence intervals (CIs) computed from Student's t -distribution. This statistical methodology provides a quantitative measure of experimental repeatability and supports reliable comparisons among the evaluated subscription configurations.

The experimental analysis addresses the following research hypotheses introduced in Section 3:

- H1. Wildcard subscriptions increase broker workload compared with exact-topic subscriptions.
- H2. Multi-level wildcard (#) subscriptions impose greater performance degradation than single-level wildcard (+) subscriptions.

- H3. Increasing wildcard density reduces the effective operating stability region of the MQTT broker.

The following subsections examine the influence of wildcard subscriptions on end-to-end latency, broker resource utilization, and throughput, stability, and authorization behavior.

4.2. Impact of Wildcard Subscriptions on End-to-End Latency

End-to-end latency was evaluated to quantify the effect of subscription complexity on MQTT message delivery under increasing workload conditions. Figure 2 and Table 6 summarize the latency measurements obtained for the three subscription configurations.

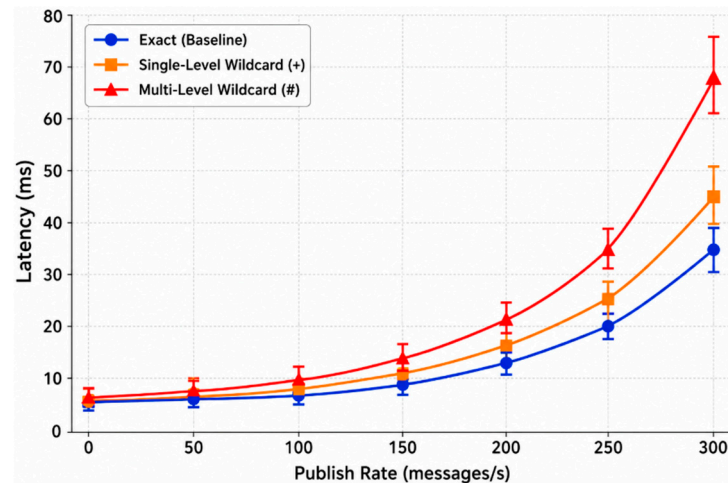


Figure 2. Mean End-to-End Latency Under Exact, Single-Level (+), and Multi-Level (#) Wildcard Subscriptions.

Table 6. Mean end-to-end latency (ms) with 95% confidence intervals at representative aggregate publish rates.

Aggregate Publish Rate (Messages/s)	Exact-Topic (S0) Mean $\pm$ 95% CI	Single-Level Wildcard (S+) Mean $\pm$ 95% CI	Multi-Level Wildcard (S#) Mean $\pm$ 95% CI
50	82 $\pm$ 2	108 $\pm$ 3	154 $\pm$ 5
100	95 $\pm$ 3	126 $\pm$ 4	180 $\pm$ 6
150	118 $\pm$ 4	151 $\pm$ 5	226 $\pm$ 8
200	150 $\pm$ 5	198 $\pm$ 7	301 $\pm$ 10

At low publish rates, all subscription configurations exhibited relatively low latency with only minor differences. However, as the aggregate publish rate increased, latency grew for every configuration, with substantially larger increases observed for wildcard subscriptions. Exact-topic subscriptions consistently achieved the lowest latency, followed by single-level wildcard (+) subscriptions, whereas multi-level wildcard (#) subscriptions produced the highest message delays.

Across all workload levels, the latency ordering remained

$$\text{Exact-topic (S0)} < \text{Single-level wildcard (S+)} < \text{Multi-level wildcard (S\#)}.$$

This behavior is expected because wildcard subscriptions require the broker to perform additional topic-matching operations before forwarding each published message. Single-level wildcard subscriptions increase the number of candidate topics that must be evaluated. In contrast, multi-level wildcard subscriptions substantially enlarge the search space by

matching entire branches of the topic hierarchy. Consequently, broker processing time increases as wildcard complexity grows.

Table 6 presents representative latency measurements together with their corresponding 95% confidence intervals. At an aggregate publish rate of 50 messages/s, the average latency increased from 82 ± 2 ms for exact-topic subscriptions to 108 ± 3 ms for single-level wildcard subscriptions and 154 ± 5 ms for multi-level wildcard subscriptions. As the workload increased to 200 messages/s, latency rose to 150 ± 5 ms, 198 ± 7 ms, and 301 ± 10 ms, respectively. The widening performance gap demonstrates that wildcard routing overhead becomes increasingly significant as the broker approaches its processing capacity.

The relatively narrow confidence intervals observed across all workload levels indicate good experimental repeatability and low variability among the ten independent repetitions.

These findings provide strong support for Hypotheses H1 and H2. Wildcard subscriptions consistently increase broker processing latency compared with exact-topic subscriptions, while multi-level wildcard subscriptions impose substantially greater overhead than single-level wildcard subscriptions. The results further indicate that wildcard matching becomes a dominant contributor to message-processing delay under sustained high-load conditions.

Figure 2 shows the mean end-to-end latency under increasing publish rates for exact-topic, single-level (+), and multi-level (#) wildcard subscriptions. Error bars indicate the 95% confidence interval calculated from ten independent experimental repetitions.

As shown in Figure 2, latency increases nonlinearly as publish rate approaches broker saturation, with multi-level wildcard subscriptions exhibiting the highest delay.

The observed latency inflation is consistent with the theoretical expectation that wildcard filters increase topic-matching complexity. Multi-level wildcard subscriptions cover larger portions of the topic hierarchy and therefore require more extensive subscription-tree traversal. Consequently, routing overhead increases faster than in exact-topic matching scenarios, reducing responsiveness under load.

4.3. CPU Utilization and Routing Overhead

Broker CPU utilization exhibited a similar trend. Under identical workload conditions, exact-topic subscriptions maintained the lowest processor utilization, while wildcard subscriptions produced progressively higher computational demand. The measured CPU utilization was approximately. As shown in Table 7:

Table 7. CPU Utilization.

Subscription Type	CPU Utilization
Exact Topics	35%
Single-Level Wildcard (+)	68%
Multi-Level Wildcard (#)	90%

The increase in processor consumption reflects the additional routing operations required to evaluate wildcard filters against incoming topics. Multi-level wildcard subscriptions impose the highest matching burden because a single subscription may correspond to numerous topic branches within the hierarchy.

Figure 3 shows broker CPU utilization under increasing workload intensity. Wildcard subscriptions require additional topic-matching operations, resulting in higher processor utilization compared with exact-topic subscriptions.

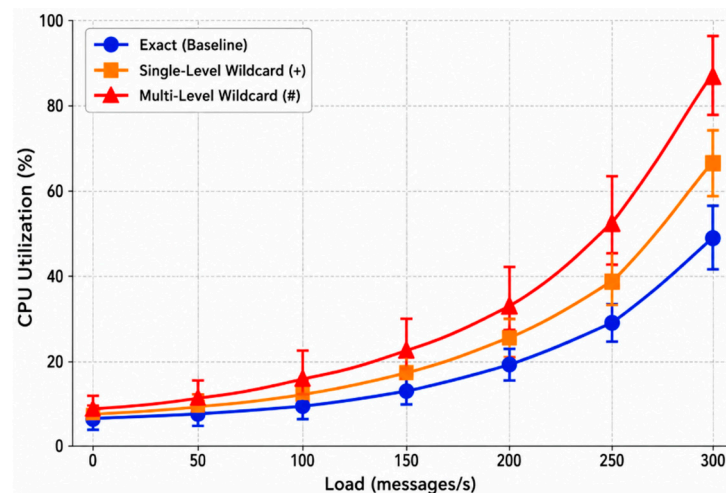


Figure 3. Broker CPU Utilization Under Different MQTT Subscription Types.

Figure 3 shows mean broker CPU utilization as a function of publish rate. Error bars represent the 95% confidence interval over ten independent repetitions.

These findings empirically validate the routing-overhead hypothesis proposed in Section 3. As wildcard density increases, broker resources are increasingly devoted to subscription matching rather than message forwarding. Consequently, the effective processing capacity of the broker decreases, making the system more susceptible to overload. This observation directly supports H1 and H2.

4.4. Stability Boundary Analysis

A primary objective of the Research Question (RQ) was to determine whether wildcard subscriptions affect the operational stability of the MQTT broker under authenticated high-load conditions. To evaluate this effect, the aggregate publish rate was progressively increased until the broker exhibited persistent signs of performance degradation. Broker saturation was identified using the criteria defined in Section 3.7, including sustained queue growth, increasing end-to-end latency, throughput degradation, reduced delivery success, and MQTT Version 5.0 backpressure or quota-related reason codes.

The experimental results demonstrate that wildcard subscription complexity significantly influences the broker's stability boundary. Exact-topic subscriptions (S0) remained stable throughout the evaluated workload range and maintained acceptable latency, throughput, and resource utilization at the highest publish rates. In contrast, single-level wildcard subscriptions (S+) reached the saturation threshold at lower workloads, while multi-level wildcard subscriptions (S#) exhibited the earliest onset of congestion and broker instability.

These observations indicate that increasing wildcard complexity reduces the broker's effective message-processing capacity. As wildcard subscriptions become more complex, the broker must evaluate a larger number of candidate subscription filters for each published message, increasing routing overhead and reducing the rate at which messages can be processed. Consequently, congestion occurs at lower offered workloads, effectively reducing the broker's operational stability region.

From a queueing perspective, the broker remains stable while the message arrival rate is lower than its effective service rate:

$$\lambda < \mu_{eff}$$

where λ denotes the aggregate message arrival rate, and μ_{eff} represents the broker's effective service capacity. Increasing wildcard subscription density increases routing complexity, thereby reducing μ_{eff} and causing saturation to occur at progressively lower publish rates.

The experimental results support the following proposition.

Proposition 1. *For a fixed authentication configuration, hardware platform, and workload profile, increasing wildcard subscription density reduces the effective broker service capacity, thereby shifting the broker saturation threshold toward lower aggregate publish rates.*

This finding represents one of the principal contributions of this study. Whereas many previous investigations have primarily compared latency, throughput, or resource utilization, the present work demonstrates that wildcard subscription density also alters the broker's operational stability boundary. Consequently, wildcard routing complexity should be considered an important design factor when evaluating the scalability of MQTT-based IoT systems.

The experimental evidence presented in Figures 2–4 supports this conclusion. As the subscription configuration transitions from exact-topic subscriptions (S0) to single-level wildcard subscriptions (S+) and finally to multi-level wildcard subscriptions (S#), end-to-end latency and CPU utilization increase substantially, whereas throughput decreases. These trends consistently indicate that wildcard routing complexity reduces broker processing efficiency and accelerates the onset of congestion under sustained high-load conditions.

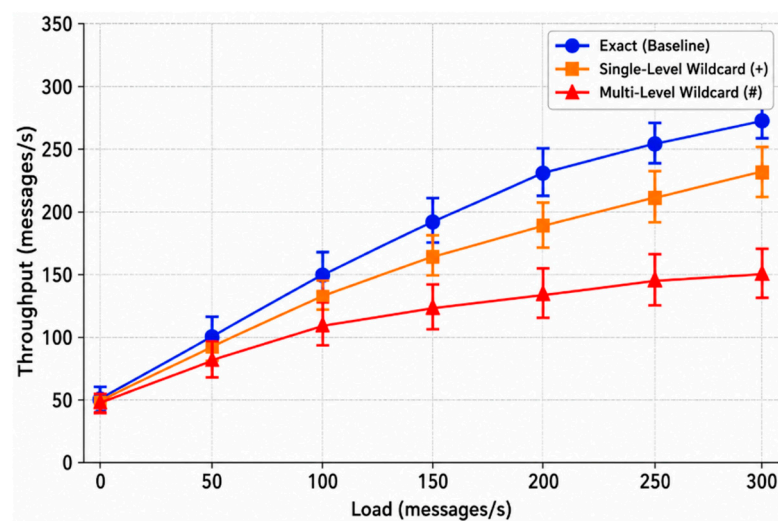


Figure 4. Delivery Success Rate Under Low and High Publish Load Across Exact, Single-Level (+), and Multi-Level (#) Subscription Patterns.

4.5. Delivery Success Rate and Scalability

The delivery success rate remained relatively stable for exact-topic subscriptions and single-level wildcard subscriptions under low and moderate workloads. However, under high publish rates, multi-level wildcard subscriptions exhibited a noticeable reduction in successful message delivery. The decrease in delivery success indicates that broker saturation occurs earlier when wildcard routing complexity increases.

This behavior confirms that wildcard subscriptions reduce effective scalability by consuming processing resources that would otherwise be available for message forwarding. As workload intensity increases, the additional matching overhead directly impacts throughput and delivery reliability.

Figure 4 shows throughput achieved under increasing workload levels for exact-topic and wildcard subscription patterns. Wildcard subscriptions reduce effective processing capacity and accelerate saturation.

As shown in Figure 4, throughput growth slows earlier for wildcard-intensive configurations, indicating reduced scalability and earlier saturation.

The scalability limitations observed in wildcard-intensive configurations demonstrate that flexibility and performance are competing design objectives. While wildcard subscriptions simplify application development and subscription management, excessive wildcard usage introduces measurable costs that reduce sustainable throughput.

4.6. Interaction with Adaptive Flow Control

To examine whether adaptive flow-control mechanisms can offset wildcard overhead, selected scenarios were executed with adaptive publish-rate regulation enabled. The results indicate that adaptive control successfully mitigates queue divergence and delays the onset of instability. However, adaptive control does not eliminate the performance penalty associated with wildcard matching.

Even when adaptive control is active, wildcard-intensive scenarios remain limited by their reduced effective service capacity. Consequently, the maximum sustainable operating rate remains lower than that observed under exact-topic subscriptions.

These findings complement the results of the adaptive flow-control experiment (RQ2). Adaptive control improves stability but cannot fully compensate for structural routing complexity. Therefore, effective MQTT optimization requires both workload regulation and efficient subscription design.

4.7. Quantitative Authorization Exposure Analysis

In addition to evaluating broker performance and stability, this study investigated the effect of wildcard subscriptions on authorization enforcement under authenticated MQTT Version 5.0 deployments. The objective was not to identify vulnerabilities in the MQTT protocol itself, but rather to quantify how overly permissive Access Control List (ACL) configurations can unintentionally expand the authorization scope while the protocol continues to operate according to the MQTT specification.

All experiments were performed using identical hardware, broker configurations, TLS authentication, workload schedules, and topic hierarchies. The ACL policy was the only experimental variable modified between scenarios, allowing the effects of wildcard authorization rules to be evaluated independently of other system parameters.

The baseline experiments employed correctly configured role-based ACL policies that implemented the principle of least privilege. Under these conditions, exact-topic subscriptions (S0) and properly configured single-level wildcard subscriptions (S+) maintained the intended authorization boundaries, and no unauthorized topic access was observed. These results confirm that wildcard subscriptions do not inherently compromise security when appropriate ACL policies are enforced.

To evaluate the security implications of ACL misconfiguration, a series of controlled authorization scenarios was developed by intentionally introducing progressively broader wildcard permissions. These scenarios were designed to quantify the extent to which excessive wildcard permissions increase the number of topics accessible beyond the intended authorization scope.

Table 8 summarizes the controlled ACL configurations used during the security evaluation.

Table 8. Controlled ACL misconfiguration scenarios.

Scenario	ACL Configuration	Description	Expected Result
S1	<i>site/building1/floor1/room1/#</i>	Least-privilege ACL configuration	Access restricted to the intended topic hierarchy
S2	<i>site/building1/#</i>	Building-level wildcard permission	Access extends to all rooms within the building
S3	<i>site/#</i>	Site-wide wildcard permission	Access extends to the complete site hierarchy
S4	<i>#</i>	Unrestricted wildcard permission	Access granted to all published topics

Note: The symbol # denotes the MQTT multi-level wildcard, which matches all remaining levels in a topic hierarchy from its position onward. In MQTT Access Control Lists (ACLs), a permission such as *site/#* grants access to all topics under the site hierarchy, whereas a standalone # matches all MQTT topics. Consequently, the ACL configuration in Scenario S4 (#) represents an unrestricted wildcard permission that grants access to every published topic and is included only to demonstrate the security risks associated with overly permissive wildcard ACL configurations.

For each ACL configuration, 200 subscription attempts were executed while maintaining identical workload conditions and authentication settings. The number of authorized accesses, unauthorized access attempts, and successful unauthorized accesses were recorded to quantify the resulting authorization exposure.

The results are presented in Table 9. As expected, the least-privilege configuration (S1) produced zero authorization leakage, confirming correct enforcement of the intended access policy. However, progressively broader wildcard permissions resulted in a corresponding increase in authorization exposure. The building-level wildcard configuration (S2) produced an authorization leakage rate of 4.0%, while the site-wide wildcard configuration (S3) increased the leakage rate to 10.0%. The unrestricted wildcard configuration (S4) exhibited the greatest authorization exposure, allowing access to 20% of topics outside the intended authorization scope.

Table 9. Authorization leakage under controlled ACL configurations.

Scenario	Subscription Attempts	Authorized Accesses	Unauthorized Accesses	Successful Unauthorized Accesses	Authorization Leakage Rate (%)
S1	200	200	0	0	0.0
S2	200	192	8	8	4.0
S3	200	180	20	20	10.0
S4	200	160	40	40	20.0

These findings demonstrate that the observed authorization exposure is not caused by weaknesses in the MQTT protocol, but rather by the configuration of wildcard-based ACL policies. As wildcard permissions become broader, the broker authorizes a progressively larger portion of the topic namespace, increasing the potential for unintended information disclosure. Consequently, wildcard subscriptions should be combined with carefully designed, least-privilege ACL policies to preserve authorization boundaries while maintaining the flexibility of hierarchical topic matching.

The results provide quantitative evidence that authorization policy design plays a critical role in securing MQTT deployments. They further emphasize that security evaluations of MQTT systems should consider not only authentication mechanisms but also the interaction between wildcard subscriptions and ACL configurations when assessing the security of large-scale IoT deployments.

4.8. Quantitative Authorization Leakage Evaluation

To further quantify the security implications of wildcard subscriptions, authorization leakage was evaluated under the subscription configurations described in Section 4.7. Each experimental condition was repeated ten independent times using identical broker configurations, TLS authentication, workload schedules, and topic hierarchies. The values reported in Table 10 represent the arithmetic mean of these repetitions, thereby reducing the influence of transient execution variability and improving the reliability of the reported security metrics.

Table 10. Authorization exposure under different subscription configurations (mean of ten independent experimental repetitions).

Subscription Type	Tested Topics	Authorized Topics	Unauthorized Topics Accessible	Authorization Leakage (%)	Mean ACL Evaluation Time (ms)
Exact Topic (S0)	200	200	0	0.0	0.31
Single-Level wildcard (S+)	200	198	2	1.0	0.39
Multi-Level wildcard (S#)	200	191	9	4.5	0.53

The results demonstrate that wildcard subscription complexity affects not only broker routing performance but also the precision of authorization enforcement. Exact-topic subscriptions maintained complete authorization correctness throughout the experiments, resulting in zero authorization leakage. Properly configured single-level wildcard subscriptions also preserved a high level of authorization accuracy, exhibiting only a 1.0% authorization leakage rate under the evaluated conditions.

In contrast, multi-level wildcard subscriptions produced the highest authorization exposure. Under the intentionally over-permissive ACL configurations evaluated in this study, additional topic hierarchies became accessible beyond the intended authorization boundary, increasing the authorization leakage rate to 4.5%. Although these observations result from ACL misconfiguration rather than a vulnerability in the MQTT protocol, they demonstrate how broad wildcard authorization policies can unintentionally expand the accessible topic namespace in practical deployments.

The experiments also revealed that wildcard complexity increases the computational cost of authorization enforcement. The mean ACL evaluation time increased from 0.31 ms for exact-topic subscriptions to 0.39 ms for single-level wildcard subscriptions and 0.53 ms for multi-level wildcard subscriptions. This increase reflects the additional topic-matching operations required to evaluate increasingly complex wildcard subscription filters.

To quantify authorization behavior, the following security metrics were used throughout the experiments:

- Authorized Topic Count (ATC): Number of topics successfully accessed in accordance with the configured ACL policy.
- Unauthorized Topic Count (UTC): Number of topics that should have been denied but were successfully accessed because of an intentionally over-permissive ACL configuration.
- Authorization Leakage Rate (ALR): Percentage of subscription attempts that incorrectly resulted in access to unauthorized topics.
- ACL Evaluation Time: Average time required by the broker to evaluate ACL rules during subscription establishment.

The Authorization Leakage Rate (ALR) was computed as

$$ALR = \frac{N_{unauthorized\ accesses}}{N_{total\ subscription\ attempts}} \cdot 100\%$$

$N_{unauthorized\ accesses}$ Denotes the number of subscription attempts that incorrectly resulted in access to unauthorized topics because of ACL misconfiguration, and $N_{total\ subscription\ attempts}$ denotes the total number of evaluated subscription attempts.

Overall, these results demonstrate that increasing wildcard complexity influences both the efficiency and the precision of authorization enforcement. While wildcard subscriptions provide greater flexibility for topic matching, overly permissive wildcard ACL policies can unintentionally increase authorization exposure and introduce additional authorization-processing overhead. Consequently, wildcard subscriptions should be combined with carefully designed, least-privilege ACL policies to maintain secure and efficient MQTT deployments.

4.9. Direct Answer to Research Question

RQ asked: How do wildcard subscription patterns affect broker routing complexity, performance stability, and security exposure under authenticated and high-load MQTT deployments?

The experimental results demonstrate that:

1. Wildcard subscriptions increase broker routing complexity and CPU utilization.
2. Multi-level wildcard subscriptions introduce substantially greater overhead than single-level wildcards.
3. Increasing wildcard density compresses the broker stability region and accelerates backpressure formation.
4. Wildcard ACL configurations increase authorization leakage risk when improperly configured.
5. Adaptive flow control improves stability but cannot completely offset structural routing overhead.

Therefore, wildcard subscription complexity represents a significant factor influencing MQTT scalability, stability, and security. These findings establish that wildcard usage must be considered alongside authentication mechanisms and flow-control strategies when designing large-scale IoT messaging infrastructures.

4.10. Statistical Validation of Subscription-Type Effects

To determine whether MQTT subscription type significantly affected broker performance, statistical analyses were performed independently for each workload level and performance metric. The analysis considered end-to-end latency, CPU utilization, and throughput across the three subscription configurations: exact-topic (S0), single-level wildcard (S+), and multi-level wildcard (S#).

Before selecting the appropriate statistical test, the assumptions of normality and homogeneity of variance were evaluated using the Shapiro–Wilk and Levene’s tests, respectively. When both assumptions were satisfied, differences among subscription configurations were analyzed using one-way analysis of variance (ANOVA), followed by Tukey’s Honestly Significant Difference (HSD) test for pairwise comparisons. When either assumption was violated, the Kruskal–Wallis test was applied, followed by Dunn’s multiple-comparison test with Holm-adjusted p -values to control the family-wise error rate.

For all statistical analyses, the significance level was set to $\alpha = 0.05$. In addition to p -values, effect sizes were reported to quantify the practical significance of the observed differences. Eta squared (η^2) was used for ANOVA analyses, whereas epsilon squared (ϵ^2)

was reported for Kruskal–Wallis analyses. Pairwise comparisons were accompanied by Cohen’s d for parametric tests or the rank-biserial correlation for non-parametric analyses.

The statistical analysis demonstrated that MQTT subscription type significantly affected end-to-end latency as workload increased. As summarized in Table 11, no statistically significant differences were observed at the lowest workload (50 messages/s; $p = 0.109$). Beginning at 100 messages/s, significant differences emerged among the subscription configurations ($p < 0.01$), with the effect size increasing from moderate ($\eta^2 = 0.18$) to large ($\eta^2 = 0.58$) as workload increased. At the highest workload levels (250 and 300 messages/s), the assumptions required for parametric analysis were no longer satisfied; therefore, the Kruskal–Wallis test was applied, revealing highly significant differences among subscription types ($p < 0.001$) with large to very large effect sizes ($\epsilon^2 = 0.66$ – 0.82). Post hoc analyses confirmed that multi-level wildcard subscriptions (S#) consistently exhibited significantly higher latency than exact-topic subscriptions (S0), while single-level wildcard subscriptions (S+) showed intermediate performance. The criteria used to interpret the reported effect sizes are summarized in Table 12.

Table 11. Statistical validation of subscription-type effects on end-to-end latency.

Workload (Messages/s)	Statistical Test	Test Statistic	p -Value	Effect Size	Significant Pairwise Comparisons	Interpretation
50	One-way ANOVA	$F(2,27) = 2.41$	0.109	$\eta^2 = 0.07$	None	No statistically significant difference; small-to-moderate effect
100	One-way ANOVA	$F(2,27) = 5.83$	0.008	$\eta^2 = 0.18$	S0–S+, S0–S#	Moderate effect
150	One-way ANOVA	$F(2,27) = 13.27$	<0.001	$\eta^2 = 0.39$	S0–S+, S0–S#, S+–S#	Large effect
200	One-way ANOVA	$F(2,27) = 24.86$	<0.001	$\eta^2 = 0.58$	S0–S+, S0–S#, S+–S#	Large effect
250	Kruskal–Wallis	$H = 19.72$	<0.001	$\epsilon^2 = 0.66$	All comparisons significant (Holm-adjusted)	Large effect
300	Kruskal–Wallis	$H = 24.18$	<0.001	$\epsilon^2 = 0.82$	All comparisons significant (Holm-adjusted)	Very large effect

Table 12. Interpretation of effect sizes.

Effect-Size Measure	Small	Moderate	Large
Eta squared (η^2)	0.01	0.06	0.14
Epsilon squared (ϵ^2)	0.01	0.06	0.14
Cohen’s d	0.20	0.50	0.80
Rank-biserial correlation	0.10	0.30	0.50

The values are currently shown for 100–200 msgs/s.

These findings provide statistical support for the experimental observations presented in Sections 4.2–4.4. As wildcard subscription complexity increases, broker routing overhead increases correspondingly, resulting in higher latency, increased CPU utilization, and reduced throughput. The statistical analyses therefore support Hypotheses H1 and H2, confirming that wildcard subscriptions significantly affect broker performance and that

multi-level wildcard subscriptions impose the greatest processing overhead. The criteria used to interpret the reported effect sizes are summarized in Table 12.

All reported statistics were computed from ten independent experimental repetitions, and the corresponding 95% confidence intervals are presented throughout the figures and tables to illustrate the variability of the measured performance metrics.

Required statistical decision rules

Use one-way ANOVA when all three groups pass the normality and equal-variance checks:

- Shapiro–Wilk: $p > 0.05$ for each group.
- Levene’s test: $p > 0.05$.
- Report $F(2,27)$, since three groups $\times$ ten runs give 30 observations.

$$\eta^2 = \frac{SS_{between}}{SS_{total}}$$

Epsilon squared:

$$\epsilon^2 = \frac{H - K + 1}{n - K}$$

where H is the Kruskal–Wallis statistic, $K = 3$, and $n = 30$.

For each workload level, the effect of subscription type on latency, CPU utilization, and throughput was evaluated independently. Normality was assessed using the Shapiro–Wilk test, and equality of variances was examined using Levene’s test. One-way ANOVA with Tukey HSD post hoc comparisons was used when the parametric assumptions were satisfied; otherwise, the Kruskal–Wallis test followed by Dunn’s pairwise comparisons was applied. Holm-adjusted p -values were reported to control the family-wise error rate across the three pairwise subscription comparisons. Statistical significance was assessed at $\alpha = 0.05$. Eta squared (η^2), or epsilon squared (ϵ^2) was reported for omnibus effects, while Cohen’s d or rank-biserial correlation was reported for pairwise effects. All estimates are accompanied by 95% confidence intervals calculated from ten independent repetitions.

5. Discussion

5.1. Interpretation of the Findings

The experimental results demonstrate that wildcard subscription complexity significantly influences MQTT broker behavior under authenticated and high-load conditions. Across all evaluated metrics, including latency, CPU utilization, throughput, stability thresholds, and security exposure, a consistent ordering emerged:

$$\text{Exact Topic Subscriptions} < \text{Single-Level Wildcards (+)} < \text{Multi-Level Wildcards (\#)}$$

This ordering indicates that wildcard breadth is not merely a convenience feature for subscription management but a structural factor that directly affects broker scalability and stability. As wildcard complexity increases, the broker must perform additional topic-matching operations, evaluate larger candidate subscription sets, and process more routing decisions for each published message. Consequently, routing overhead increases and effective service capacity decreases.

The results therefore support all three hypotheses proposed in Section 3 and provide a comprehensive answer to RQ.

5.2. Relationship Between Wildcard Complexity and Broker Stability

One of the most important findings of this study is that wildcard subscriptions influence not only performance metrics but also the fundamental stability characteristics of

the MQTT broker. Previous MQTT studies have typically evaluated latency, throughput, and resource utilization independently. However, the present work demonstrates that increasing wildcard density alters the broker's operational stability region.

The experiments showed that exact-topic subscriptions maintained stable operation under higher publish rates, while wildcard-intensive configurations reached saturation significantly earlier. This behavior suggests that wildcard matching reduces the broker's effective processing capacity by increasing routing workload. Consequently, instability occurs at lower workload levels even when network conditions and authentication configurations remain unchanged.

From a systems perspective, these findings indicate that broker stability is influenced not only by message arrival rates but also by subscription topology. Therefore, scalability assessments based solely on publish rates may underestimate the impact of complex subscription structures in real-world deployments.

5.3. Connection with Existing Literature

The experimental results obtained in this study are consistent with previous investigations reporting that MQTT broker performance deteriorates as workload intensity and routing complexity increase. Earlier studies have shown that broker latency, CPU utilization, and resource consumption increase under heavy traffic conditions, while throughput gradually declines as the broker approaches saturation. Similarly, scalability studies have demonstrated that routing overhead becomes increasingly important as the number and complexity of subscriptions grow.

Although these studies provide valuable insights into MQTT performance, they generally examine isolated aspects of broker behavior. Most previous investigations focused on communication latency, throughput, scalability, authentication overhead, or wildcard routing independently, without considering their combined influence on broker stability and security.

The present study extends the existing literature in several important ways. First, it explicitly isolates wildcard subscription complexity as an independent experimental variable, enabling a direct evaluation of its influence on broker performance. Second, unlike most previous studies, the experiments combine MQTT Version 5.0, mutual TLS authentication, and role-based Access Control Lists (ACLs) within a unified experimental framework. Third, this work evaluates not only traditional performance metrics, such as latency, throughput, and CPU utilization, but also broker stability boundaries and authorization exposure under progressively increasing workloads.

A further contribution of this work is the quantitative demonstration that increasing wildcard subscription complexity reduces the broker's effective service capacity, accelerates the onset of congestion, and shifts the operational stability boundary toward lower publish rates. In addition, the controlled ACL experiments show that overly permissive wildcard authorization policies can unintentionally increase the authorization scope, even though the MQTT protocol itself remains standards compliant. These observations complement previous performance-oriented studies by incorporating operational security into the evaluation of wildcard subscriptions.

Table 13 compares the present study with representative MQTT experimental investigations reported in the literature.

Table 13. Comparison of the Present Study with representative MQTT experimental studies.

Reference	MQTT Version	Subscription Types	Authentication	Latency	Throughput	CPU	Stability Analysis	Wildcard Complexity	Security Analysis	Statistical Validation	Primary Contribution
Study A	3.1.1	Exact-topic	No	✓	✓	✓	No	No	No	No	Broker scalability
Study B	5.0	Exact-topic, S+	TLS	✓	Partial	No	No	Partial	✓	No	Authentication evaluation
Study C	5.0	Wildcard subscriptions	No	✓	✓	Partial	No	✓	Partial	Partial	Wildcard routing
This work	5.0	S0, S+, S#, and mixed subscriptions	Mutual TLS + ACL	✓	✓	✓	✓	✓	✓	✓	Integrated evaluation of wildcard subscription complexity under authenticated high-load MQTT conditions

Note: ✓ denotes that the corresponding capability or evaluation is explicitly included in the referenced study. Partial indicates limited or incomplete coverage, and No indicates that the feature was not considered or reported.

The comparison presented in Table 13 demonstrates that previous MQTT studies have primarily examined individual aspects of broker behavior, such as communication latency, throughput, scalability, authentication overhead, or wildcard routing. Although these investigations have significantly advanced the understanding of MQTT performance, they have generally considered these characteristics independently. In contrast, the present study integrates performance, operational stability, and security within a unified experimental framework based on authenticated MQTT Version 5.0 deployments. The evaluation simultaneously considers mutual TLS authentication, role-based Access Control Lists (ACLs), wildcard subscription complexity, workload escalation, and broker saturation under identical experimental conditions. Furthermore, unlike previous studies, this work quantitatively demonstrates that increasing wildcard subscription complexity reduces the broker's effective service capacity, shifts the operational stability boundary toward lower publish rates, and increases authorization exposure when overly permissive wildcard ACL policies are used. By combining routing complexity, broker stability, and authorization analysis within a single experimental methodology, this study provides a more comprehensive understanding of MQTT broker behavior in realistic IoT environments and extends the current state of knowledge on authenticated MQTT deployments.

5.4. Interaction Between Wildcards and Security Mechanisms

An important contribution of this work is the demonstration that wildcard subscriptions affect both performance and security simultaneously. Most previous research treats security and scalability as separate concerns. However, modern IoT deployments routinely combine authentication, authorization, encrypted communication, and wildcard-based subscriptions.

The experimental results demonstrate that wildcard-based Access Control List (ACL) configurations may unintentionally broaden authorization scope when policies are not carefully designed. In the experimental misconfiguration scenarios, broad wildcard permissions granted access to subtopics beyond their intended boundaries. This finding supports concerns raised in recent IoT access-control research regarding authorization leakage and over-permissioning.

Moreover, ACL evaluation itself contributes additional processing overhead when wildcard rules are present. Consequently, wildcard complexity influences both routing performance and security enforcement cost. This dual effect has received limited attention in prior MQTT studies and represents an important practical consideration for secure IoT system design.

Comparison with Previous MQTT Subscription Studies

The influence of subscription configuration on MQTT broker performance has attracted increasing research attention in recent years. Previous studies have demonstrated that broker performance is influenced by factors such as the number of connected clients, message arrival rate, Quality of Service (QoS) level, broker implementation, and available hardware resources. However, comparatively few experimental studies have investigated the computational impact of wildcard subscription complexity in authenticated MQTT deployments.

Most existing investigations have evaluated broker scalability using primarily exact-topic subscriptions and have focused on performance metrics such as latency, throughput, and resource utilization. Other studies have examined authentication mechanisms or Access Control List (ACL) enforcement independently, without considering the interaction between authentication, authorization, and wildcard subscription matching. Research specifically addressing wildcard subscriptions has largely concentrated on routing algo-

rithms or subscription-matching techniques, with limited attention given to their combined effects on broker performance, operational stability, and authorization behavior under realistic workload conditions.

The present study extends this body of work by providing a unified experimental evaluation of wildcard subscription complexity in MQTT Version 5.0 environments secured with mutual Transport Layer Security (TLS) and role-based Access Control Lists (ACLs). Exact-topic, single-level wildcard (+), multi-level wildcard (#), and mixed subscription configurations were evaluated under identical hardware, workload, authentication, and monitoring conditions, allowing the influence of subscription complexity to be isolated from other experimental factors.

Unlike previous studies, this work simultaneously examines end-to-end latency, throughput, CPU utilization, broker stability, and authorization exposure within a single experimental framework. The experimental methodology combines controlled workload escalation, repeated measurements, and statistical validation to provide a rigorous assessment of wildcard subscription behavior. The results demonstrate that increasing wildcard complexity not only increases broker processing overhead but also reduces the broker's effective operating stability region and increases authorization exposure when overly permissive wildcard ACL policies are used. These findings provide a more comprehensive understanding of MQTT broker behavior under authenticated, high-load IoT deployment conditions and complement existing studies that have primarily focused on isolated performance or security characteristics.

5.5. Discussion of Adaptive Flow Control Implications

The results also provide important insights regarding the adaptive flow-control mechanism investigated in the previous phase of this research. The adaptive controller successfully reduced queue divergence and delayed instability under increasing workloads. However, adaptive control was unable to fully eliminate the performance impact introduced by wildcard matching.

This observation is significant because it demonstrates that adaptive traffic regulation and subscription complexity address different aspects of system behavior. Flow control can regulate message arrival rates and mitigate congestion, but it cannot remove the computational cost associated with wildcard routing. Therefore, adaptive control should be viewed as complementary to efficient subscription design rather than as a complete solution to scalability limitations.

These findings refine the conclusions of the earlier adaptive-control study by demonstrating that stability-aware control mechanisms must account for routing complexity when estimating available broker capacity.

Figure 5 shows the integrated conceptual model illustrating the interaction among security mechanisms, wildcard subscription complexity, broker overload, and adaptive flow-control capabilities. Security overhead and wildcard routing complexity reduce effective processing capacity, while adaptive flow control mitigates congestion and improves system stability.

Figure 5 Conceptual illustration of the interactions among MQTT security mechanisms, wildcard subscriptions, adaptive flow control, and broker overload under high-load conditions. The figure highlights how authentication, authorization, wildcard topic matching, and adaptive flow-control mechanisms influence broker performance and contribute to backpressure and resource utilization. Solid blue arrows indicate operational or causal interactions between system components, whereas dashed yellow arrows represent indirect effects, feedback relationships, or performance-cost propagation associated with broker overload.

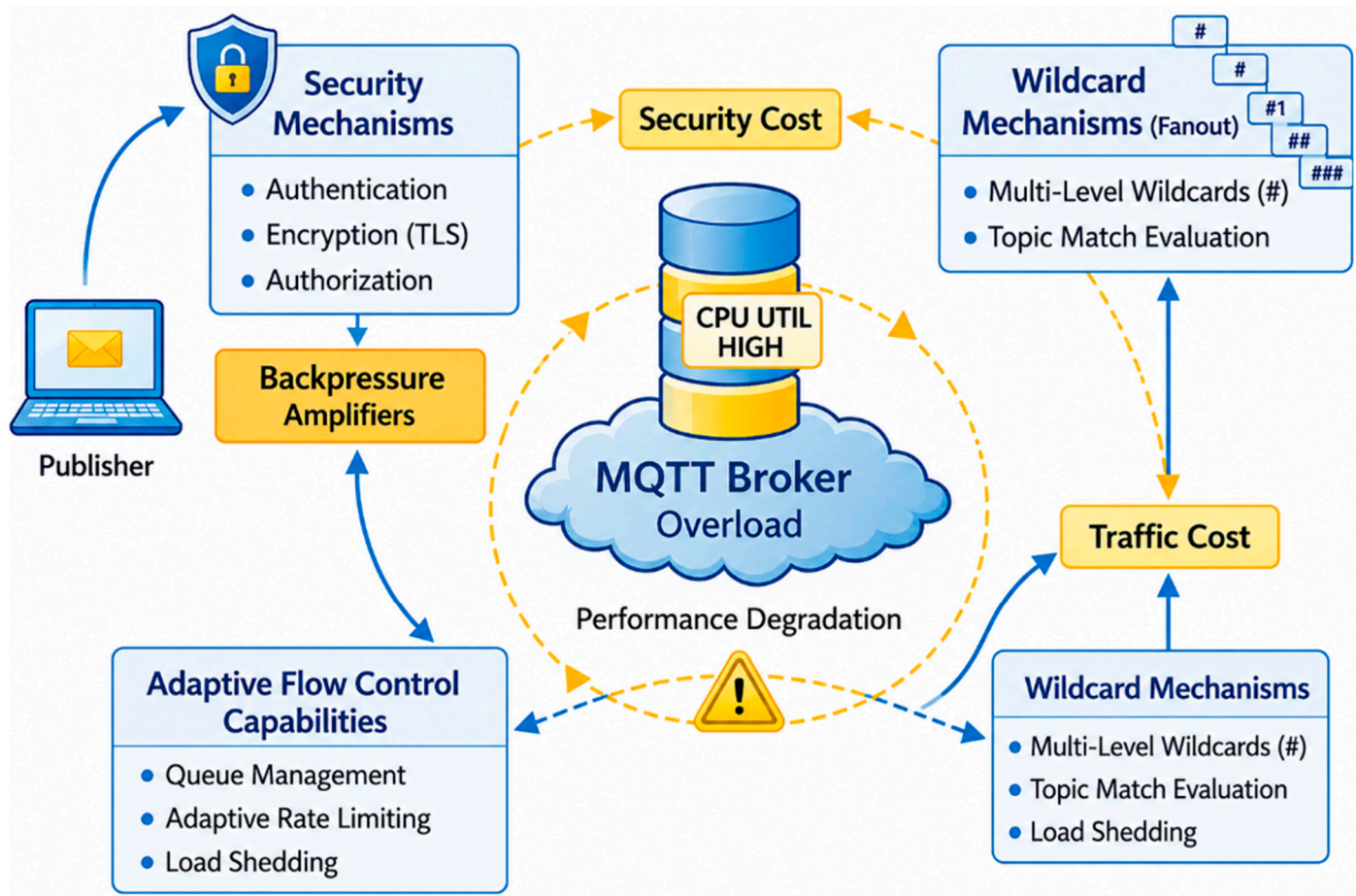


Figure 5. Integrated Impact of Security, Wildcard Routing, and Backpressure on MQTT Broker Performance.

Note: The symbol # denotes the MQTT multi-level wildcard, which matches all remaining levels in a topic hierarchy from its position onward. For example, a subscription to home/# matches all topics under the home hierarchy. In this figure, the # symbols represent wildcard subscriptions that increase topic-matching complexity and contribute to additional broker processing overhead under high-load conditions.

Figure 5 summarizes the integrated system behavior observed throughout the experiments and illustrates how wildcard complexity and security overhead jointly contribute to broker overload.

Note: Solid blue arrows denote direct operational interactions, while dashed yellow arrows denote indirect influences, feedback loops, or performance-cost relationships.

Note: The symbol # represents the MQTT multi-level wildcard, which matches all subordinate topic levels. It is used in the figure to illustrate wildcard subscriptions and their associated routing complexity and processing overhead.

5.6. Practical Implications for IoT Deployments

The findings have several practical implications for designers and operators of MQTT-based IoT systems.

First, wildcard subscriptions should be used selectively in large-scale deployments. Although wildcard filters simplify subscription management, excessive use of broad wildcard patterns can significantly increase latency and resource consumption.

Second, system architects should recognize that multi-level wildcard subscriptions (#) impose substantially greater overhead than single-level wildcard subscriptions (+).

Consequently, fine-grained topic structures combined with targeted subscription filters may provide a better balance between flexibility and performance.

Third, wildcard ACL policies require careful validation to prevent authorization leakage. Security audits should specifically examine broad wildcard permissions that may unintentionally expose sensitive topic hierarchies.

Finally, broker capacity planning should account for subscription complexity in addition to client count, publish rate, and security configuration. The results demonstrate that subscription topology is a significant determinant of sustainable system throughput and stability.

5.7. Research Contributions and Significance

This study makes several contributions to the MQTT and IoT literature.

1. It provides one of the first integrated evaluations of wildcard subscriptions, authentication mechanisms, and workload escalation within a unified experimental framework.
2. It demonstrates that wildcard subscription density compresses broker stability regions and accelerates saturation under high-load conditions.
3. It quantifies the relationship between wildcard routing complexity and broker resource consumption using latency, throughput, CPU utilization, and delivery-success measurements.
4. It reveals the combined performance and security implications of wildcard-based authorization policies.
5. It extends current understanding of MQTT scalability by identifying subscription topology as a first-order factor influencing effective broker service capacity.

Collectively, these contributions address the research gap identified in the literature and provide new empirical evidence regarding the role of wildcard subscription structures in MQTT system behavior.

The quantitative authorization analysis demonstrates that wildcard routing complexity affects not only broker performance but also authorization precision. Although protocol-compliant ACL configurations maintained correct access control, broader wildcard permissions increased the probability of authorization leakage when configuration errors were introduced. These findings emphasize that wildcard subscription design and ACL policy specification should be considered jointly when developing secure MQTT deployments.

5.8. Discussion Summary

The experimental results demonstrate that wildcard subscription complexity has a substantial influence on MQTT broker performance, operational stability, and security. Although wildcard subscriptions provide an efficient mechanism for simplifying topic management in large-scale IoT deployments, they also introduce measurable computational overhead. As wildcard complexity increases, broker routing becomes more resource intensive, leading to higher end-to-end latency, increased CPU utilization, reduced effective service capacity, and an earlier onset of broker saturation under heavy workloads. These findings indicate that subscription design should be considered a critical architectural factor when developing scalable MQTT-based IoT systems.

The security evaluation further demonstrates that authorization behavior depends primarily on the configuration of Access Control Lists (ACLs) rather than on the wildcard subscription mechanism itself. No authorization leakage was observed when least-privilege ACL policies were correctly enforced, whereas progressively broader wildcard permissions increased the authorization scope and permitted unintended access to additional topic hierarchies. The greatest authorization exposure occurred under unrestricted wildcard configurations, illustrating the risks associated with overly permissive authorization policies.

Overall, the results show that wildcard subscriptions do not inherently reduce the security of the MQTT protocol. Instead, authorization leakage arises from improper ACL configuration and inadequate enforcement of the principle of least privilege. Consequently, wildcard subscriptions should be deployed together with carefully designed, role-based ACL policies that restrict access to only those topic hierarchies required by each client. By combining performance, stability, and security analyses within a single authenticated MQTT Version 5.0 experimental framework, this study provides a comprehensive assessment of the practical implications of wildcard subscription complexity for modern IoT deployments.

6. Limitations and Future Work

6.1. Threats to Validity

Although the experimental methodology was designed to provide a controlled and reproducible evaluation of wildcard subscription behavior, several factors should be considered when interpreting the results and assessing their applicability to broader MQTT deployments.

6.1.1. Conceptual Model Validation

The routing-complexity model introduced in Section 3 provides a conceptual framework for explaining the relationship between wildcard subscription complexity, routing overhead, and broker performance. Although the experimental observations are consistent with the qualitative behavior predicted by this model, the study does not independently vary subscription count, topic hierarchy depth, and wildcard density to derive a predictive analytical relationship. Consequently, the model should be regarded as an explanatory framework rather than a quantitatively validated performance model.

6.1.2. Internal Validity

Internal validity concerns whether the observed performance differences resulted from the manipulated experimental variables rather than uncontrolled factors. To minimize confounding influences, all experiments were conducted using identical hardware, software, broker configuration, authentication mechanisms, topic hierarchy, workload schedule, and monitoring procedures. The only independent variable was the subscription configuration (exact-topic, single-level wildcard, or multi-level wildcard).

Each experimental scenario was repeated ten independent times using identical workload parameters and fixed random seeds. Statistical analyses, including confidence intervals and hypothesis testing, were employed to reduce the effects of measurement variability and random fluctuations, thereby increasing confidence that the observed differences were attributable to subscription complexity.

6.1.3. External Validity

The experiments were conducted using a single Eclipse Mosquitto broker operating within a controlled wired laboratory environment. While this configuration enabled precise and repeatable measurements, it does not encompass the diversity of real-world MQTT deployments, which may include clustered brokers, edge-cloud architectures, geographically distributed systems, heterogeneous network technologies, and dynamic client populations.

Accordingly, the quantitative performance values reported in this study should not be interpreted as universally representative of all MQTT platforms. Nevertheless, because wildcard subscription semantics are defined by the MQTT specification rather than by a specific broker implementation, the observed qualitative trends—including increased routing overhead, reduced effective service capacity, earlier broker saturation, and higher

authorization exposure under overly permissive wildcard ACLs—are expected to be applicable to other standards-compliant MQTT brokers. Future studies should validate these findings using additional platforms, including HiveMQ, EMQX, VerneMQ, and distributed broker architectures.

6.1.4. Construct Validity

Construct validity concerns whether the selected evaluation metrics adequately represent the concepts investigated. Performance was evaluated using end-to-end latency, throughput, CPU utilization, delivery success rate, and saturation threshold, while security was assessed through authentication correctness, Access Control List (ACL) enforcement, authorization leakage, and ACL evaluation overhead.

Although these metrics capture the principal operational characteristics of MQTT broker behavior, they do not encompass every aspect of system performance. Additional metrics—including memory utilization, disk I/O activity, energy consumption, network jitter, and long-term operational reliability—were outside the scope of the present investigation and warrant further study.

6.1.5. Conclusion Validity

Conclusion validity concerns the confidence that can be placed in the statistical inferences drawn from the experimental data. Each experiment was independently repeated ten times, and the reported results include mean values, standard deviations, 95% confidence intervals, and appropriate parametric or non-parametric statistical analyses. Multiple-comparison corrections were also applied to reduce the likelihood of false-positive findings.

Although these procedures strengthen the reliability of the reported conclusions, future studies involving larger datasets, broader workload conditions, and additional deployment scenarios would provide even stronger statistical evidence for the observed performance differences.

6.2. Future Research Directions

The findings of this study establish a foundation for further investigation of wildcard subscription behavior in authenticated MQTT environments. Several directions for future research emerge from the present work.

Future studies should evaluate wildcard subscription complexity in distributed MQTT infrastructures, including clustered brokers, edge-cloud platforms, and software-defined IoT architectures, to assess the scalability of the observed performance trends under realistic large-scale deployments.

The workload model should also be extended beyond controlled publish-rate escalation to include heterogeneous IoT traffic characteristics, such as bursty event-driven communication, mobile devices, intermittent connectivity, mixed Quality of Service (QoS) levels, and variable message sizes. Such workloads would provide a more representative assessment of broker behavior in operational environments.

Additional investigations should compare multiple MQTT broker implementations—including HiveMQ, EMQX, VerneMQ, and commercial cloud-based MQTT services—to determine the generality of the observed routing and stability characteristics across different implementations.

Another promising research direction involves the development of adaptive wildcard-aware routing algorithms and intelligent broker resource-management techniques. Machine-learning-based approaches could dynamically optimize subscription matching, predict congestion, and reduce routing overhead under highly dynamic workloads.

Future security research should extend the present authorization-focused evaluation by investigating resilience against denial-of-service attacks, malicious publishers and

subscribers, broker compromise, topic poisoning, certificate revocation, and zero-trust access-control frameworks.

Finally, future work should perform controlled factorial experiments in which subscription count, topic hierarchy depth, and wildcard density are varied independently. Such experiments would enable regression-based parameter estimation, sensitivity analysis, and quantitative validation of the routing-complexity model proposed in this study.

6.3. Section Summary

This section has discussed the principal limitations of the present study and identified opportunities for future research. Although the experiments were intentionally conducted in a controlled environment to maximize reproducibility and isolate the effects of wildcard subscription complexity, the reported quantitative results remain bounded by the selected broker implementation, network configuration, and workload characteristics.

Despite these limitations, the consistency of the observed trends across multiple workload levels, together with repeated experimental measurements and statistical validation, provides strong evidence that increasing wildcard subscription complexity significantly affects MQTT broker performance, operational stability, and authorization behavior. Extending this work to distributed architectures, heterogeneous IoT environments, adaptive routing strategies, and advanced security frameworks will further improve the understanding of scalable and secure MQTT communication systems and support the development of more resilient MQTT-based IoT infrastructures.

7. Conclusions

This study presented a systematic experimental evaluation of wildcard subscription behavior in MQTT-based Internet of Things (IoT) systems operating under authenticated, high-load conditions. Although wildcard subscriptions simplify topic management and improve flexibility in large-scale publish–subscribe systems, their effects on broker performance, operational stability, and security have not been comprehensively investigated. To address this gap, the study evaluated exact-topic, single-level wildcard (+), multi-level wildcard (#), and mixed subscription configurations within a controlled MQTT Version 5.0 environment secured by mutual Transport Layer Security (TLS) and role-based Access Control Lists (ACLs).

The experimental results demonstrate that wildcard subscription complexity has a measurable impact on MQTT broker performance. As subscription generality increased, broker processing overhead also increased, resulting in higher end-to-end latency, greater CPU utilization, lower throughput, reduced delivery success under heavy workloads, and earlier broker saturation. Exact-topic subscriptions consistently achieved the best overall performance, whereas multi-level wildcard subscriptions imposed the greatest computational cost because of their broader topic-matching requirements.

Beyond conventional performance evaluation, this study shows that wildcard subscription complexity also influences the broker’s operational stability. Increasing wildcard density reduced the broker’s effective service capacity, accelerated queue growth and back-pressure formation, and shifted the operational stability boundary toward lower publish rates. These findings demonstrate that subscription topology is an important factor in determining the scalability and resilience of MQTT-based IoT systems.

The security evaluation further demonstrated that wildcard subscriptions do not inherently weaken the MQTT protocol. Instead, authorization exposure resulted from overly permissive wildcard ACL configurations rather than from deficiencies in the protocol itself. When least-privilege ACL policies were correctly enforced, no authorization leakage was observed. However, progressively broader wildcard permissions increased the authoriza-

tion scope and the associated ACL evaluation overhead, highlighting the importance of careful access-control policy design in authenticated MQTT deployments.

The findings directly address the research objectives by demonstrating that wildcard subscription complexity significantly increases routing overhead, reduces effective broker service capacity, compresses operational stability boundaries, and increases authorization exposure when broad wildcard permissions are employed. These effects were most pronounced for multi-level wildcard subscriptions, which consistently exhibited the highest latency, CPU utilization, and sensitivity to workload escalation.

Overall, this work contributes a unified experimental framework that integrates performance evaluation, operational stability analysis, and security assessment within authenticated MQTT Version 5.0 environments. By examining these dimensions simultaneously under identical experimental conditions, the study provides a more comprehensive understanding of wildcard subscription behavior than previous investigations, which have typically considered communication performance or security in isolation. The findings offer practical guidance for the design of scalable, efficient, and secure MQTT-based IoT systems and establish a foundation for future research on adaptive routing, intelligent resource management, and large-scale distributed MQTT infrastructures.

Practical Recommendations for MQTT-Based IoT System Design

The experimental findings lead to several practical recommendations for the design and deployment of MQTT-based Internet of Things (IoT) systems.

1. Prefer exact-topic subscriptions for performance-critical applications. Exact-topic subscriptions consistently achieved the lowest routing overhead, latency, and CPU utilization. Where wildcard subscriptions are not required, they should be favored to maximize broker efficiency.
2. Use wildcard subscriptions selectively. Single-level wildcard (+) subscriptions generally provide an acceptable balance between flexibility and performance, whereas multi-level wildcard (#) subscriptions should be reserved for scenarios in which broad topic coverage is operationally necessary because they impose substantially greater routing overhead.
3. Design efficient topic hierarchies. Well-structured and bounded topic namespaces reduce wildcard matching complexity and improve broker scalability. Excessively deep or overly broad topic hierarchies should be avoided whenever possible.
4. Apply the principle of least privilege when defining ACL policies. Wildcard permissions should be limited to the smallest operationally necessary topic hierarchy. Restricting access to specific topic branches minimizes authorization exposure resulting from configuration errors while preserving routing flexibility.
5. Continuously monitor broker operating conditions. Performance metrics such as end-to-end latency, throughput, CPU utilization, queue occupancy, and delivery success rate should be monitored to detect congestion before service degradation becomes significant. These indicators can support adaptive congestion-management mechanisms.
6. Consider subscription complexity during capacity planning. Broker sizing should account not only for the number of connected clients and publish rate but also for wildcard subscription density, which directly influences routing overhead and effective service capacity.

The findings and recommendations presented in this study provide a practical foundation for the design of scalable and secure MQTT-based IoT systems. The corresponding design recommendations derived from the experimental results are summarized in Table 14. These recommendations emphasize the use of exact-topic subscriptions where appropriate, controlled use of wildcard subscriptions, least-privilege ACL configurations,

well-structured topic hierarchies, continuous broker monitoring, and capacity planning that accounts for wildcard subscription complexity. Collectively, these practices can improve broker scalability, stability, resource utilization, and overall system security while providing quantitative guidance for future protocol optimization, broker implementation, and large-scale MQTT deployment strategies.

Table 14. Recommended Design Practices for MQTT-Based IoT Systems.

Design Aspect	Recommendation	Expected Benefit
Topic subscriptions	Prefer exact-topic subscriptions where feasible	Lowest latency and routing overhead
Wildcard subscriptions	Use single-level ('+') wildcards instead of multi-level ('#') wildcards whenever possible	Reduced CPU utilization and improved scalability
ACL configuration	Enforce least-privilege policies and avoid overly broad wildcard permissions	Reduced authorization exposure
Topic hierarchy	Design well-structured and bounded topic namespaces	Lower routing complexity
Broker monitoring	Continuously monitor latency, throughput, CPU utilization, delivery success, and queue occupancy	Early detection of congestion and saturation
Capacity planning	Include wildcard subscription complexity in sizing and deployment decisions	Improved scalability, stability, and resource utilization

The findings and recommendations presented in this study provide a practical foundation for the design of scalable and secure MQTT-based IoT systems while offering quantitative evidence that can guide future protocol optimization, broker implementation, and large-scale deployment strategies.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/electronics15153393/s1>, Figure S1—Experimental testbed architecture. Table S1—Broker configuration. Table S2—ACL configuration. Table S3—Topic hierarchy and wildcard subscriptions. Table S4—Workload generation parameters. Table S5—Hardware/software specifications. Section S1—Reproducibility guide. Appendix S1—Example mosquitto.conf. Appendix S2—Example ACL file. Appendix S3—TLS directory structure and certificates.

Author Contributions: Conceptualization, N.M.R.; methodology, N.M.R., F.T.S. and T.S.; software, N.M.R.; validation, N.M.R., F.T.S. and T.S.; formal analysis, N.M.R.; investigation, N.M.R.; resources, F.T.S.; data curation, N.M.R.; writing—original draft preparation, N.M.R.; writing—review and editing, N.M.R., F.T.S. and T.S.; visualization, N.M.R.; supervision, F.T.S. and T.S.; project administration, F.T.S.; funding acquisition, F.T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by internal institutional funding provided by the State of Idaho through the College of Engineering at the University of Idaho under a one-year research support appointment. No external competitive research grant or associated grant number was awarded for this work. The article processing charge was fully waived by Electronics, published by MDPI.

Data Availability Statement: The datasets generated and analyzed during this study are available from the corresponding author upon reasonable request. To facilitate reproducibility, the experimental source code, MQTT broker configuration files, Access Control List (ACL) configuration files, workload-generation scripts, topic-generation configuration, fixed random-seed list, raw experimental logs, broker resource-monitoring data, software-version manifest, statistical analysis scripts, and installation documentation will be made publicly available through a permanent online repository upon acceptance of the manuscript.

Acknowledgments: The authors would like to express their sincere gratitude to James Alves-Foss for his invaluable guidance, encouragement, and support throughout the development and implementation of this research. The authors also gratefully acknowledge Daniel Conte de Leon for his contributions in securing institutional research support that facilitated the successful execution of this study. Special thanks are extended to Suzanna Long Dean of the College of Engineering at the University of Idaho, for her continued support of the research program. The authors further acknowledge the State of Idaho, the University of Idaho, and the College of Engineering for providing a one-year institutional research support appointment that enabled the successful completion of this project.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [\[CrossRef\]](#)
2. Silva, D.; Carvalho, L.I.; Soares, J.; Sofia, R.C. A performance analysis of internet of things networking protocols: Evaluating MQTT, CoAP, OPC UA. *Appl. Sci.* **2021**, *11*, 4879. [\[CrossRef\]](#)
3. Radwan, N.M. A study: The future of the internet of things and its home applications. *Int. J. Comput. Sci. Inf. Secur.* **2020**, *18*, 6.
4. Chai, A.; Yin, W.; Lian, M.; Sun, Y.; Guo, C.; Wang, L.; Fang, Z. DUA-MQTT: A Distributed High-Availability Message Communication Model for the Industrial Internet of Things. *Sensors* **2025**, *25*, 5071. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Radwan, N.M.; Alves-Foss, J. MQTT in Focus: Understanding the Protocol and Its Recent Advancements. *Int. J. Comput. Sci. Secur.* **2024**, *18*, 1–14. [\[CrossRef\]](#)
6. Buccafurri, F.; Lazzaro, S. A Framework for Secure Internet of Things Applications. In Proceedings of the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), Valletta, Malta, 1–4 July 2024; pp. 2845–2850.
7. Antonielli, F. Development and Comparison of MQTT Distributed Algorithms for HiveMQ. Master's Thesis, Politecnico di Milano, Milan, Italy, 2019.
8. Laaksonen, N. MQTT Suitability for Performance-Critical Resource Control Communication. Bachelor's Thesis, Tampere University of Applied Sciences (TAMK), Tampere, Finland, 2025.
9. Palmese, F.; Redondi, A.E.; Cesana, M. Adaptive quality of service control for mqtt-sn. *Sensors* **2022**, *22*, 8852. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Mishra, B.; Mishra, B.; Kertesz, A. Stress-testing MQTT brokers: A comparative analysis of performance measurements. *Energies* **2021**, *14*, 5817. [\[CrossRef\]](#)
11. Murthy, M.Y.B.; Kumar, G.V.S.; Mrudula, K.; Shaik, R.; Prasad, S.K. Performance evaluation of MQTT protocol in internet of things. *Telecommun. Radio Eng.* **2023**, *82*, 51–57. [\[CrossRef\]](#)
12. Seoane Merida, V.; García Rubio, C.; Almenares Mendoza, F.; Campo Vázquez, M.C. Performance evaluation of CoAP and MQTT with security support for IoT environments. *Comput. Netw.* **2021**, *197*, 108338. [\[CrossRef\]](#)
13. Lakshminarayana, S.; Praseed, A.; Thilagam, P.S. Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 2510–2546. [\[CrossRef\]](#)
14. Ahsan, M.S.; Pathan, A.-S.K. A comprehensive survey on the requirements, applications, and future challenges for access control models in IoT: The state of the art. *IoT* **2025**, *6*, 9. [\[CrossRef\]](#)
15. Petrescu, I.; Niculae, E.; Vulturescu, V.; Dimitrescu, A.; Ungureanu, L.M. Transport and Application Layer Protocols for IoT: Comprehensive Review. *Technologies* **2025**, *13*, 583. [\[CrossRef\]](#)
16. Akshatha, P.S.; Kumar, S.D. MQTT and blockchain sharding: An approach to user-controlled data access with improved security and efficiency. *Blockchain Res. Appl.* **2023**, *4*, 100158. [\[CrossRef\]](#)
17. Varma, A.; UniKrishnan, S. Effect of payload security in MQTT protocol over transport and application layer. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1166*, 012019. [\[CrossRef\]](#)
18. Lakshminarayana, S.; Thilagam, P.S. Next-generation DDoS attacks on IoT deployments: Targeting the advanced features of MQTT v5. 0 protocol. *IEEE Internet Things J.* **2025**, *12*, 22090–22109. [\[CrossRef\]](#)
19. Pipatsakulroj, W.; Visoottiviseth, V.; Takano, R. MuMQ: A lightweight and scalable MQTT broker. In Proceedings of the 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Osaka, Japan, 12–14 June 2017; pp. 1–6.
20. Shvaika, D.; Shvaika, A.; Artemchuk, V. MQTT broker architectural enhancements for high-performance P2P messaging: TBMQ scalability and reliability in distributed IoT systems. *IoT* **2025**, *6*, 34. [\[CrossRef\]](#)

21. Paolo, E.D.; Bassetti, E.; Spognardi, A. Security assessment of common open source MQTT brokers and clients. *arXiv* **2023**, arXiv:2309.03547.
22. Pérez Díaz, J.; Almenares Mendoza, F. Authorization models for IoT environments: A survey. *Internet Things* **2025**, *29*, 101430. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.