MQTT Broker Load Testing Tool

Project Abstract — projectech.in (built to order) · K-1631

Introduction / Background

MQTT carries the telemetry of countless IoT deployments, and every serious deployment must answer how many clients its broker can serve before latency explodes. This project builds a load-testing studio: a virtual broker engine with configurable capacity and latency, live charts of throughput and p50/p95/p99 latency, three test scenarios, automatic pass/fail verdicts, and CSV/JSON export — all in a single-file web app.

Problem Statement

Load-testing against a production broker is risky and needs real client fleets; students without that infrastructure either skip testing or fake numbers. A virtual broker that honestly models capacity ceilings and queueing latency lets the methodology — profiles, scenarios, percentiles, verdicts — be built and demonstrated properly, with a documented path to hardware-in-the-loop runs.

Objectives

- Model a virtual MQTT broker: capacity, latency, sessions, accept behavior.
- Generate configurable client load with ramp-up and QoS mixes.
- Chart throughput, latency percentiles, connections and QoS mix live.
- Run throughput, connection-flood and fan-out scenarios.
- Render automatic pass/fail verdicts and export full time series.

Proposed Methodology

Each 500 ms tick ramps clients online, computes offered load (with fan-out amplification where applicable), and compares it against a jittered capacity ceiling; excess counts as dropped. Latency combines base latency, a queueing term active above ~85% utilization, and Gaussian jitter; percentiles come from a 400-sample sliding window. Verdicts compare sustained throughput, mean p99, loss and ramp completion against targets.

System Architecture / Working

Single-file web app: control layer (broker config, load profile, scenario cards), engine layer (tick loop, capacity model, latency model, percentile windows), visualization layer (hand-rolled canvas line charts, donut, KPI cards), and export layer (CSV/JSON serializers). No dependencies; the model is documented as a teaching simplification in the UI and report.

Hardware / Software Requirements

- Hardware: any laptop/desktop with a modern browser (2 GB RAM minimum).
- Software: current Chrome, Edge, Firefox or Safari; no installation, no broker needed for the demo.
- Hardware-in-the-loop (documented): any MQTT broker + bridge adapter.

Expected Outcomes

- A working studio that demonstrates load-testing methodology end to end.
- Visible knee curves linking load, throughput and tail latency.
- Exportable datasets ready for the project report.
- Complete viva kit: report, PPT and Q&A.

Applications

- CS/IT coursework on networks and distributed systems.
- IoT courses teaching MQTT at scale.
- Pre-deployment capacity planning exercises.

Future Scope

- Real-broker bridge with live packet capture.
- Retained-message and wildcard subscription scenarios.
- TLS handshake overhead modeling.
- Distributed load generation across machines.
