BLOG / FIELD NOTES

MQTT Reconnection: A Reproducible Reasoning Framework

A simulated environment for validating connection closure, resubscription, and message recovery without involving production systems.

2026-10-18 MQTT / TCP / Reliability

> **Example article:** This uses a general local experiment to demonstrate the site's article structure. Verify every conclusion against your own experiment records before publishing.

Where the problem starts

When an application observes a disconnected connection, calling `socket.close()` only releases local connection resources. Recovery also depends on whether a new connection can be established, subscriptions become active again, and messages are handled while the connection is down.

A minimal experiment

Run a Broker and two test clients locally, then publish numbered synthetic messages at a fixed interval. At minimum, record:

- Connection state and reconnection time before and after the disconnect.
- Whether subscriptions recover as expected.
- Whether message sequence numbers contain gaps or duplicates.
- Differences across QoS and session configurations.

```
publisher → test broker → subscriber
                 ↓
           disconnect / reconnect
```

Boundaries to make explicit

Use backoff and jitter in the reconnection loop so many clients do not retry together. Validate subscription recovery, idempotent handling, and persistent sessions against the selected client library and Broker configuration. One successful reconnect is not evidence that the link is reliable.

Next steps

[Add packet captures, logs, and result screenshots from your local simulation, including software versions.]