

MQTT - asynchronous messaging

Overview

This project demonstrates the implementation of MQTT (Message Queue Telemetry Transport) protocol for IoT device communication. MQTT is a lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency, or unreliable networks, making it ideal for IoT applications.

System Architecture

The implementation consists of three main components:

1. **MQTT Broker (Mosquitto)**
 - Acts as the central message hub
 - Handles message routing between publishers and subscribers
 - Manages client connections and message queues
2. **Publisher (IoT Sensor Simulator)**
 - Simulates an IoT device sending sensor data
 - Publishes 1,000,000 messages to demonstrate high-volume data transmission
 - Implements message counting and delivery confirmation
3. **Subscriber (Data Consumer)**
 - Receives and processes messages from the publisher
 - Tracks message reception
 - Verifies data integrity through message counting

Command Reference

1. Start Mosquitto:
brew services start mosquitto
2. Run Subscriber
python3 mqtt_subscriber.py
3. Run Publisher
python3 mqtt_publisher.py

Implementation Features

- **Quality of Service (QoS) Level 1:** Ensures at-least-once message delivery
- **Message Tracking:** Implements counters on both publisher and subscriber sides
- **Progress Monitoring:** Reports progress every 1000 messages
- **Error Handling:** Robust error management and reporting
- **Clean Shutdown:** Proper resource cleanup and connection termination

Execution

```
Messages published: 950000
Messages published: 951000
Messages published: 952000
Messages published: 953000
Messages published: 954000
Messages published: 955000
Messages published: 956000
Messages published: 957000
Messages published: 958000
Messages published: 959000
Messages published: 960000
Messages published: 961000
Messages published: 962000
Messages published: 963000
Messages published: 964000
Messages published: 965000
Messages published: 966000
Messages published: 967000
Messages published: 968000
Messages published: 969000
Messages published: 970000
Messages published: 971000
Messages published: 972000
Messages published: 973000
Messages published: 974000
Messages published: 975000
Messages published: 976000
Messages published: 977000
Messages published: 978000
Messages published: 979000
Messages published: 980000
Messages published: 981000
Messages published: 982000
Messages published: 983000
Messages published: 984000
Messages published: 985000
Messages published: 986000
Messages published: 987000
Messages published: 988000
Messages published: 989000
Messages published: 990000
Messages published: 991000
Messages published: 992000
Messages published: 993000
Messages published: 994000
Messages published: 995000
Messages published: 996000
Messages published: 997000
Messages published: 998000
Messages published: 999000
Messages published: 1000000

Total messages published: 1000000
(dev) ➔ Assignment3 git:(main) ✖
```

```
Messages received: 948000
Messages received: 949000
Messages received: 950000
Messages received: 951000
Messages received: 952000
Messages received: 953000
Messages received: 954000
Messages received: 955000
Messages received: 956000
Messages received: 957000
Messages received: 958000
Messages received: 959000
Messages received: 960000
Messages received: 961000
Messages received: 962000
Messages received: 963000
Messages received: 964000
Messages received: 965000
Messages received: 966000
Messages received: 967000
Messages received: 968000
Messages received: 969000
Messages received: 970000
Messages received: 971000
Messages received: 972000
Messages received: 973000
Messages received: 974000
Messages received: 975000
Messages received: 976000
Messages received: 977000
Messages received: 978000
Messages received: 979000
Messages received: 980000
Messages received: 981000
Messages received: 982000
Messages received: 983000
Messages received: 984000
Messages received: 985000
Messages received: 986000
Messages received: 987000
Messages received: 988000
Messages received: 989000
Messages received: 990000
Messages received: 991000
Messages received: 992000
Messages received: 993000
Messages received: 994000
Messages received: 995000
Messages received: 996000
Messages received: 997000
Messages received: 998000
Messages received: 999000
Messages received: 1000000
```

Conclusion

This implementation demonstrates the practical application of MQTT in IoT scenarios, highlighting:

- Scalability for high-volume message handling
- Reliability through QoS implementation
- Robust error handling and monitoring
- Practical considerations for real-world deployment

The successful transmission and verification of 1,000,000 messages proves the system's capability to handle substantial IoT communication loads while maintaining reliability and performance.