

NoMQ: A Lightweight and Secure Messaging Protocol for IoT Devices with Broadcast and P2P Capabilities

Arman Ghabadi

June 2025

Abstract

NoMQ is a lightweight, secure, and brokerless messaging protocol tailored for Internet of Things (IoT) devices running MicroPython on resource-constrained platforms like ESP32 and ESP8266. It supports reliable message delivery with Quality of Service (QoS) levels, AES-CBC encryption, HMAC-SHA256 authentication, and both broadcast and peer-to-peer (P2P) communication. Devices can broadcast messages to a channel, which subscribed devices receive, while unsubscribed devices discard irrelevant data. Additionally, P2P communication allows direct messaging using specific IP addresses. This paper presents NoMQ’s architecture, operational logic, and network topology, with diagrams illustrating its components, message flow, and communication modes. Developed by Arman Ghabadi, NoMQ is designed for industrial IoT applications requiring real-time, secure, and efficient communication.

1 Introduction

The growth of IoT devices in industrial environments necessitates communication protocols that are lightweight, secure, and adaptable to dynamic network topologies. Traditional protocols like MQTT rely on centralized brokers, introducing latency and single points of failure. NoMQ, developed by Arman Ghabadi, is a decentralized, brokerless messaging protocol for MicroPython-based devices. It integrates AES-CBC encryption, HMAC-SHA256 authentication, and multiple QoS levels (0, 1, and 2) for secure and reliable communication. A key innovation is its support for both broadcast and peer-to-peer (P2P) communication. Devices can broadcast messages to a channel (e.g., using IP 255.255.255.255), which subscribed devices receive, while unsubscribed devices drop irrelevant data. P2P communication enables direct messaging to specific IP addresses, enhancing flexibility. This paper details NoMQ’s architecture, operational logic, and network topology, with TikZ-generated diagrams.

2 Architecture of NoMQ

NoMQ’s architecture is optimized for resource-constrained IoT devices, balancing security, efficiency, and flexibility. The system comprises the following components, grouped by functionality:

- **Network Layer:** Handles UDP sockets (IPv4/IPv6) for low-overhead communication, supporting broadcast and P2P modes.
- **Security Module:** Implements AES-CBC encryption (256-bit key) and HMAC-SHA256 authentication.
- **Channel Management:** Manages up to 20 channels with SHA-256-derived identifiers.
- **Message Processing:** Handles retained (up to 5) and pending (up to 50) messages, with filtering for unsubscribed data.
- **QoS Handler:** Supports three QoS levels for reliable delivery.
- **Logging:** Uses SimpleLogger for system monitoring.

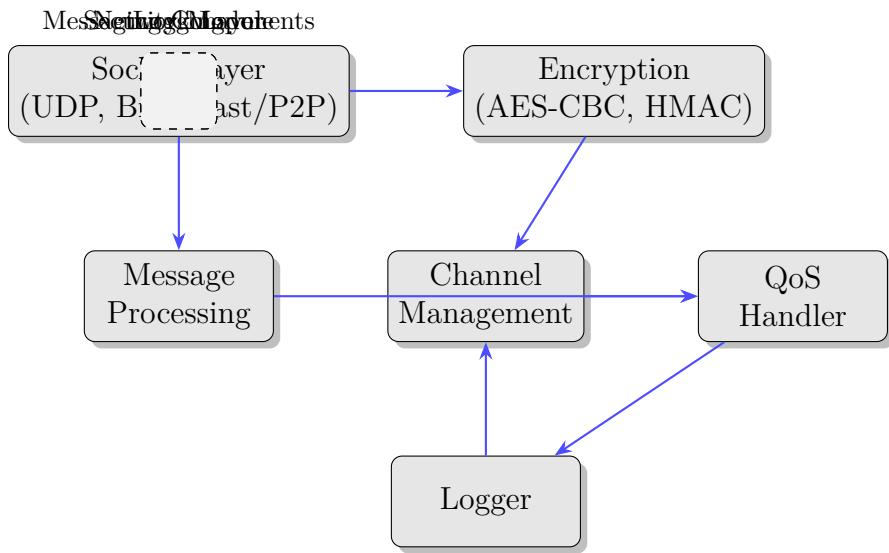


Figure 1: NoMQ System Architecture

Figure 1 shows NoMQ’s modular architecture. The network layer handles UDP-based communication, supporting broadcast to channels and P2P via direct IP addressing. The security module ensures data integrity and confidentiality. Messaging components manage channels, messages, and QoS, while the logger monitors operations.

3 Network Topology

NoMQ operates in a decentralized topology, supporting both broadcast and P2P communication. In broadcast mode, a device publishes messages to a channel using a broadcast IP (e.g., 255.255.255.255). Subscribed devices receive and process these messages, while unsubscribed devices discard them, ensuring efficient bandwidth usage. In P2P mode, devices communicate directly using specific IP addresses, ideal for targeted messaging. This dual-mode approach enhances flexibility and scalability in industrial IoT networks.

Figure 2 illustrates NoMQ’s topology. Device 1 broadcasts to Channel A, received by Devices 2, 3, and 4 (subscribers), but dropped by Device 5 (unsubscribed). Device 3 communicates directly with Device 4 via P2P on Channel B using a specific IP address, demonstrating targeted messaging.

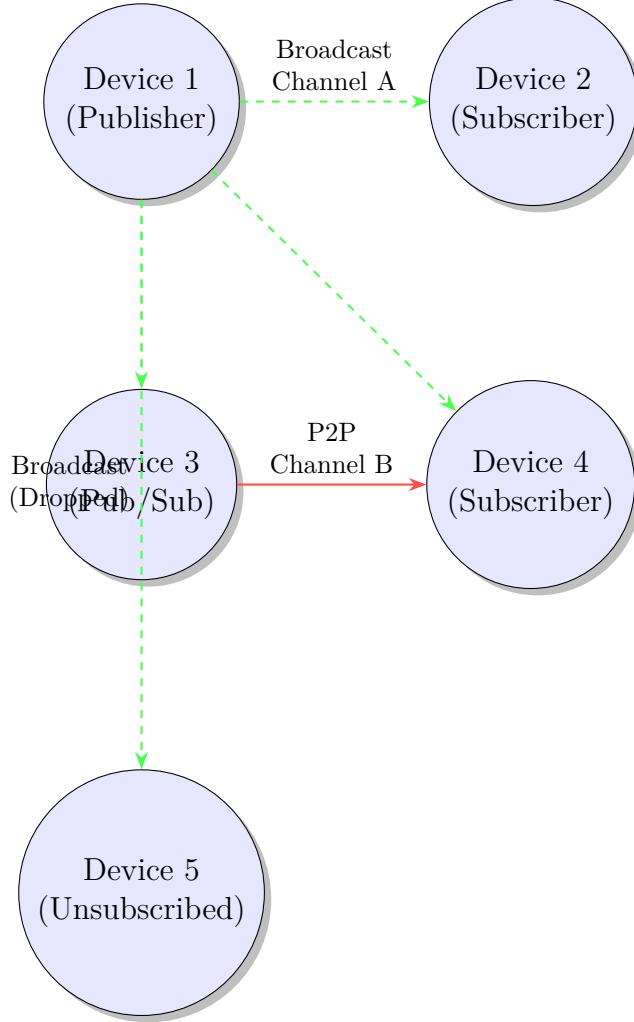


Figure 2: NoMQ Network Topology with Broadcast and P2P Communication

4 Operational Logic

NoMQ ensures secure and reliable message delivery through a structured packet format and QoS mechanisms. The protocol supports three QoS levels:

- **QoS 0:** Fire-and-forget delivery.
- **QoS 1:** At-least-once delivery with ACK.
- **QoS 2:** Exactly-once delivery via a four-way handshake (PUBREC, PUBREL, PUBCOMP).

Figure 3 shows the QoS 2 handshake, ensuring exactly-once delivery.

4.1 Packet Structure

The packet structure is compact and secure, comprising:

- **Control Header** (15 bytes): Magic number, version, type, flags, packet ID, session ID, TTL.
- **Security Header** (20 bytes): Nonce and timestamp.

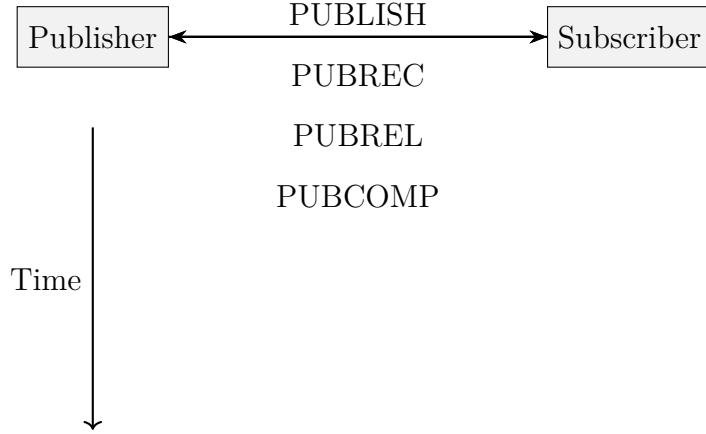


Figure 3: QoS 2 Message Flow

- **Data Header** (18 bytes): Channel ID and payload length.
- **Payload**: AES-CBC encrypted data.
- **HMAC**: 32-byte SHA256 signature.

Magic (2)	Ver (1)	Type (1)	Flags (1)	Packet ID (4)	Session ID (4)	TTL (2)	Nonce (16)	TS (4)	Channel ID (16)	Payload Len (2)	Payload (var)	HMAC (32)
Control Header				Security Header					Data Header			

Figure 4: Packet Structure

Figure 4 depicts the compact packet structure, fitting within page margins.

4.2 Code Example

A sample NoMQ usage, including broadcast and P2P:

```

1 import asyncio
2 from nomq import NoMQ
3
4 async def main():
5     nomq = NoMQ(config_file='nomq_config.json')
6     await nomq.subscribe('test/channel', priority=5)
7     # Broadcast message
8     await nomq.publish('test/channel', 'Broadcast Msg', qos=1, ip='
255.255.255.255')
9     # P2P message
10    await nomq.publish('test/channel', 'P2P Msg', qos=1, ip='
192.168.1.100')
11    listener = await nomq.listen()
12    while True:
13        msg = listener.mssg()
14        if msg:
15            print(f'Received: {msg["message"]} on {msg["channel"]}')
16            await asyncio.sleep(0.1)
17
18 asyncio.run(main())

```

Listing 1: NoMQ Usage with Broadcast and P2P

5 Conclusion

NoMQ is a versatile messaging protocol for IoT devices, offering secure, lightweight, and brokerless communication. Its support for broadcast and P2P modes, combined with AES encryption, HMAC authentication, and QoS, makes it ideal for industrial applications. Future enhancements may include advanced routing and additional encryption options.

References

- [1] Arman Ghabadi, *NoMQ: Lightweight Messaging Protocol for IoT*, 2025.