

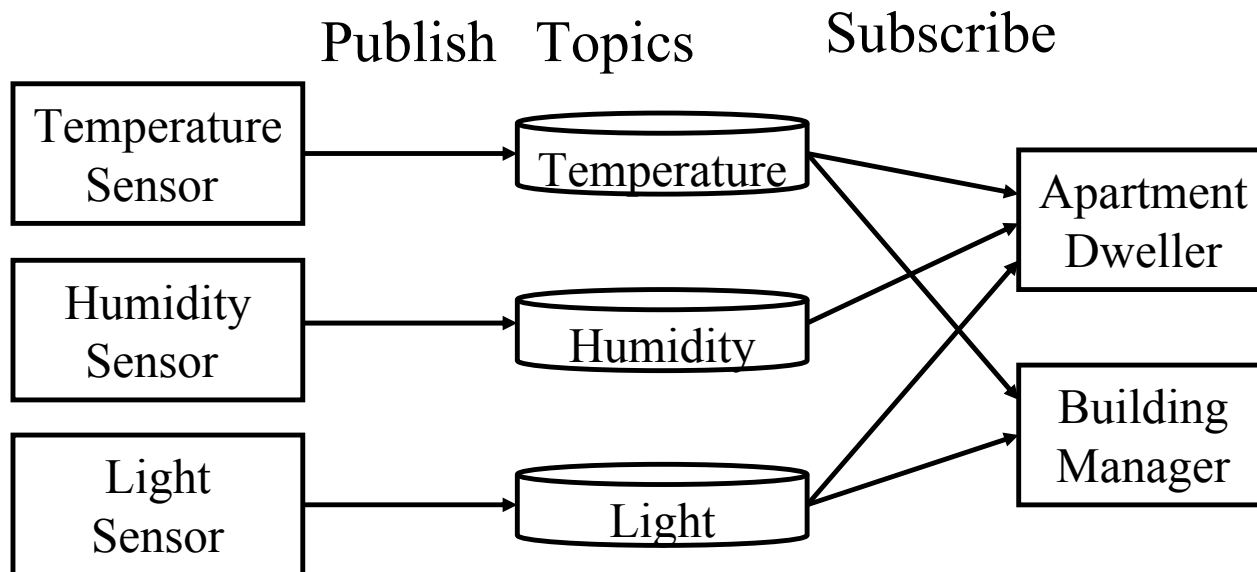
MQTT

Vấn đề cần giải quyết

- Kết nối giữa các thiết bị và giữa các thiết bị với máy chủ
- Mức năng lượng của nhiều thiết bị bị giới hạn do chạy pin
- Dữ liệu được truyền trên các mạng không có độ tin cậy cao (Wireless, cellular)
- Không cần phát triển giao thức từ đầu cho mỗi ứng dụng IoTs

MQTT

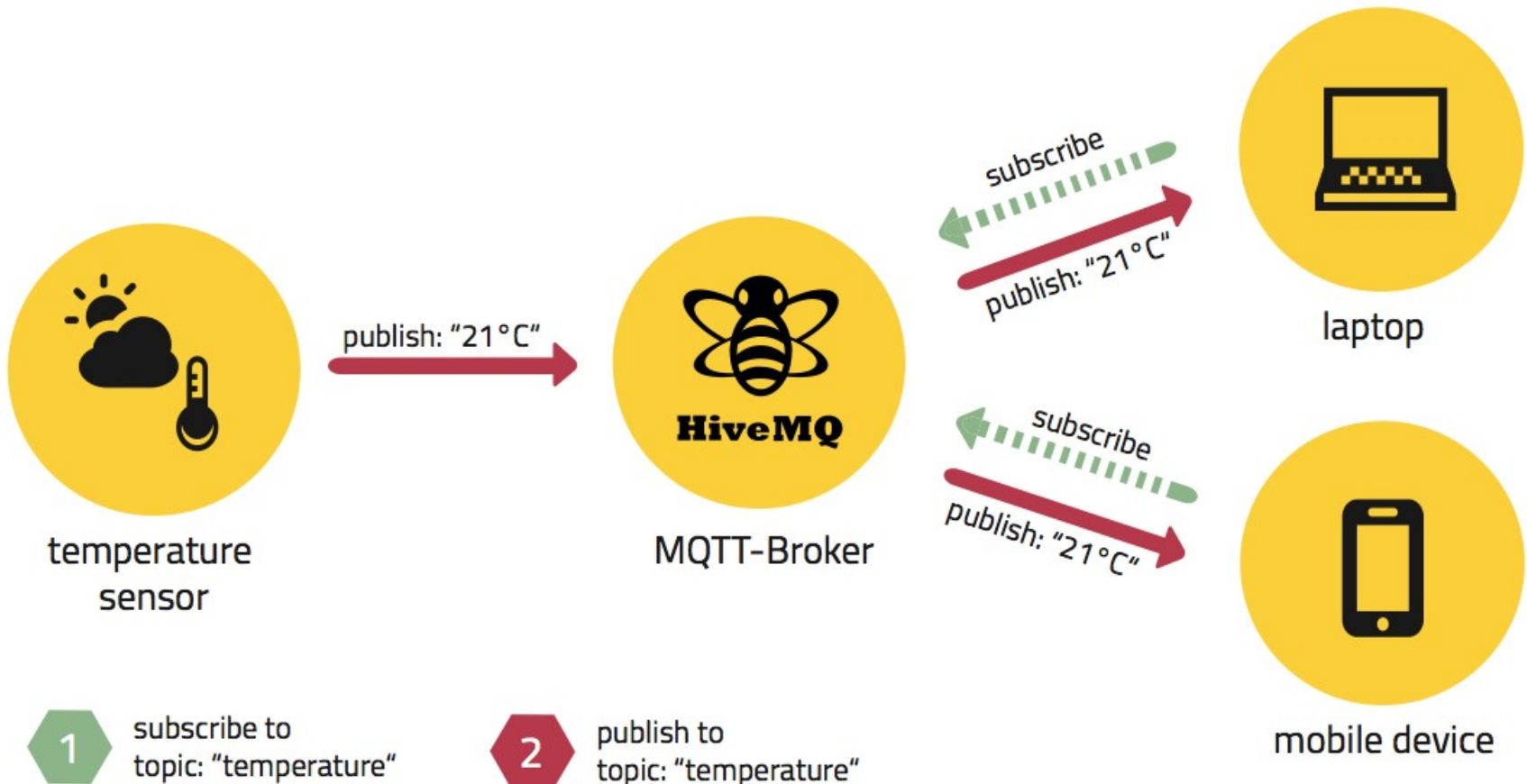
- MQTT = MQ Telemetry Transport
- Giao thức gọn nhẹ kiểu publish/subscribe đảm bảo truyền tin tin cậy
- Tạo ra năm 1999 bởi Andy Stanford-Clark (IBM) and Arlen Nipper
 - Mục tiêu ban đầu là để gửi dữ liệu sensor từ các ống dầu qua đường vệ tinh



Các đặc trưng

- Thiết kế để truyền dữ liệu kích thước lớn qua mạng có băng thông thấp
- Dữ liệu điều khiển nhỏ
- Chạy phía trên giao thức TCP/IP
- Hướng sự kiện
- Không có polling
- Có các cơ chế: phục hồi, lưu và chuyển, publish/subscribe

Publish / Subscribe



Publish/subscribe

- Tách publisher ra khỏi subscriber
- Clients luôn kết nối với một broker
 - Publishers và subscribers đều là clients
- Một bên gửi (publisher), một hoặc nhiều bên nhận (subscribers)
- Sử dụng "topics" (chủ đề) để xác định bên nhận thông tin
- Chủ đề có cấu trúc phân tầng

Broker

- Lỗi của giao thức publish/subscribe
- Nhận tất cả thông báo từ clients, lọc chúng và gửi thông báo cho tất cả các clients có quan tâm đến topic
- Xử lý việc chứng thực/chứng nhận giữa các clients
- Khả năng mở rộng cao, dễ dàng tích hợp vào các hệ thống backend, chịu lỗi tốt

Kết nối, luôn khởi tạo bởi client

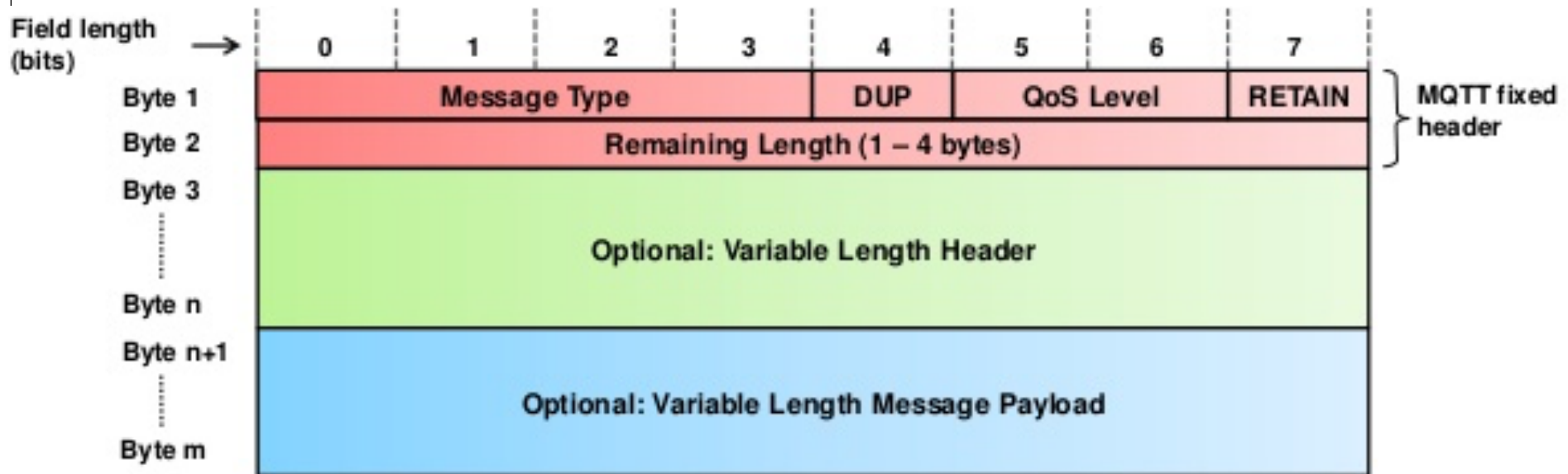


MQTT Client



MQTT Broker

Định dạng thông báo MQTT



Gói tin CONNECT

- Fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (1)				Reserved			
	0	0	0	1	0	0	0	0
byte 2...	Remaining Length							

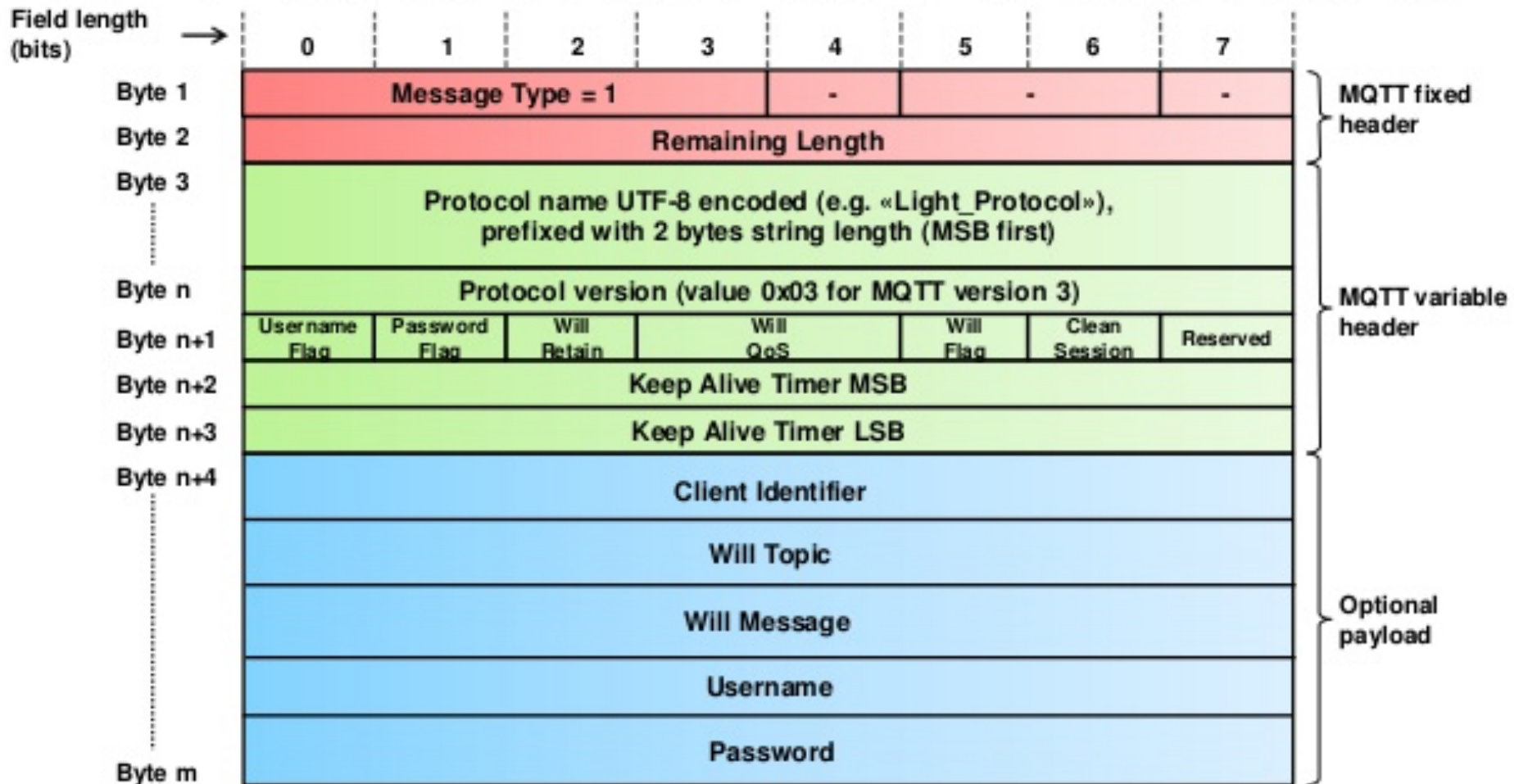
- Variable header

- Protocol Name, Protocol Level, Connect Flags, and Keep Alive.

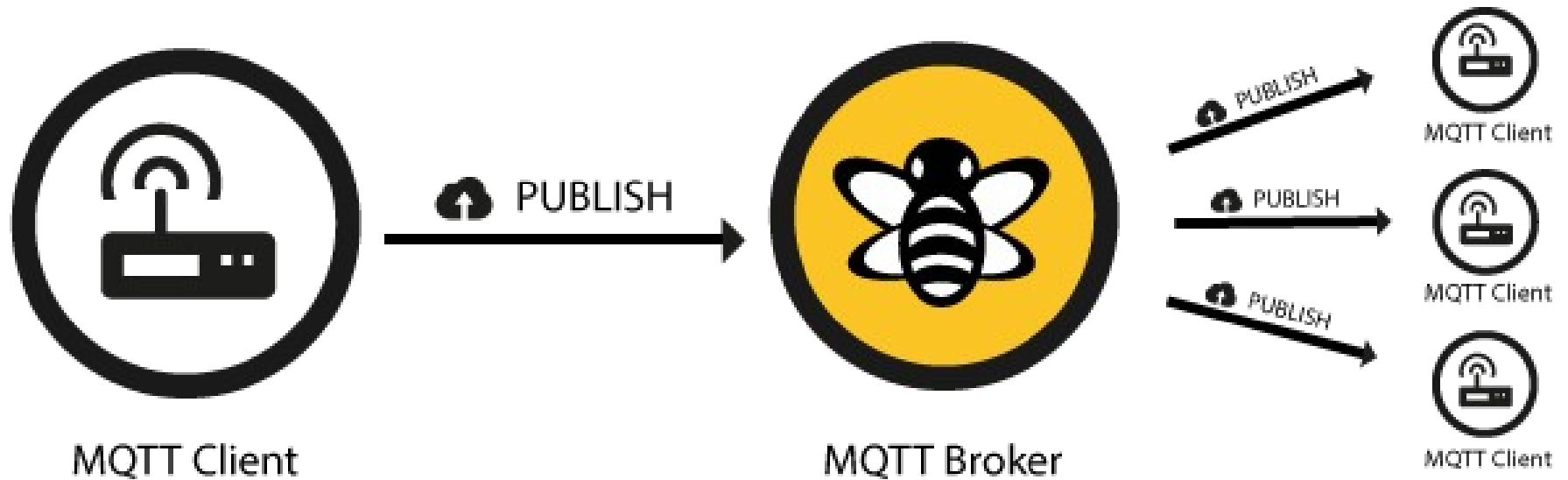
Gói tin CONNECT

- ClientId: Unique ID với mỗi broker.
- Clean Session: Cờ hiệu để chỉ ra phiên làm việc có bền (persistent) không.
 - persistent : broker sẽ lưu tất cả thông tin của client và các thông báo chưa gửi
 - non-persistent: broker không lưu thông tin gì về client
- Username/Password: Gửi dạng plaintext. Ứng dụng phải mã hóa
- Will Message: Thông báo các client khác khi một client bị mất kết nối. Broker sẽ thay mặt client gửi thông báo.
- KeepAlive: Khoảng thời gian client gửi PING đến broker

Định dạng gói tin CONNECT



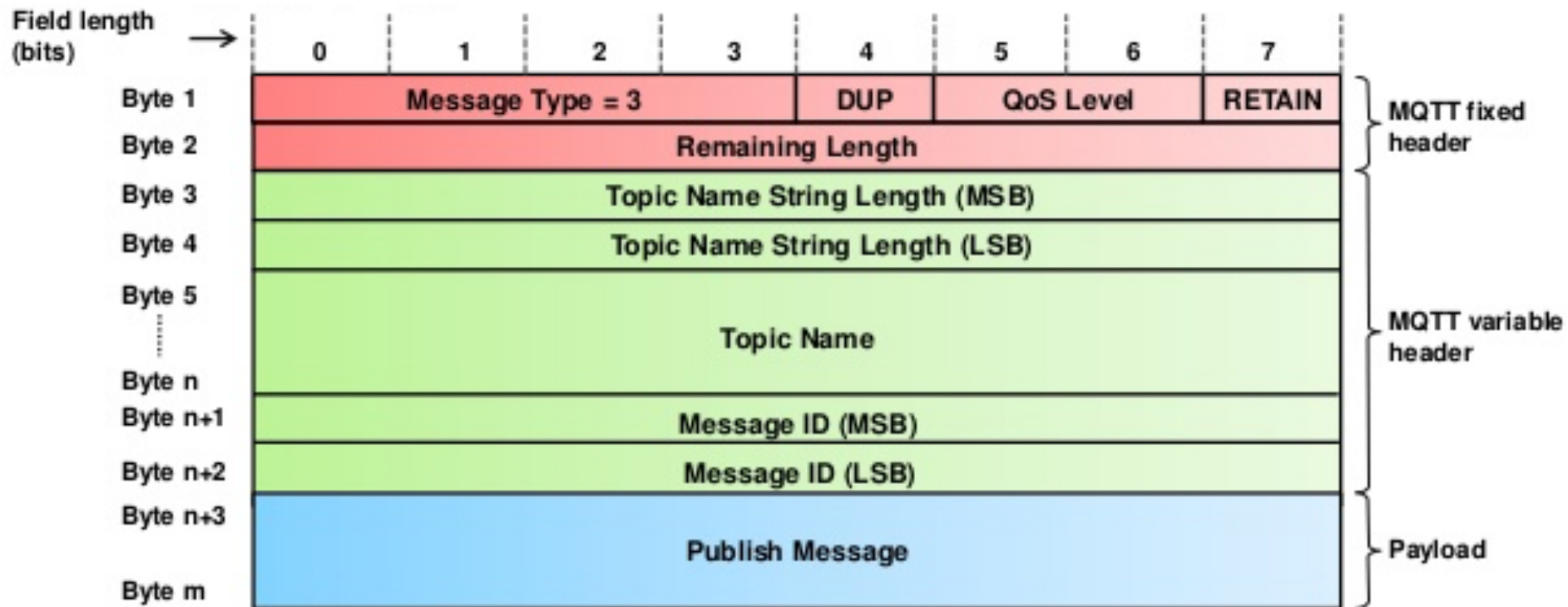
Publish, subscribe and unsubscribe



Publish packet

- Topic Name: Chuỗi ký tự có cấu trúc phân tầng, phân tầng bởi ký tự, VD "building/room_number/humidity"
- QoS: Quality of Service Level (0,1,2).
- Retain-Flag: Thiết lập nếu broker lưu thông báo gần nhất của một topic như giá trị mới nhất. Client đăng ký mới vào topic sẽ nhận được ngay thông báo này.
- Payload: dạng binary
- Packet Identifier: Định danh duy nhất giữa client và broker để phân biệt các thông báo. Chỉ dùng với QoS 1 và 2
- DUP flag: Chỉ ra rằng thông báo này được gửi lại vì bên gửi chưa nhận được ACK. Chỉ dùng với QoS 1 và 2.

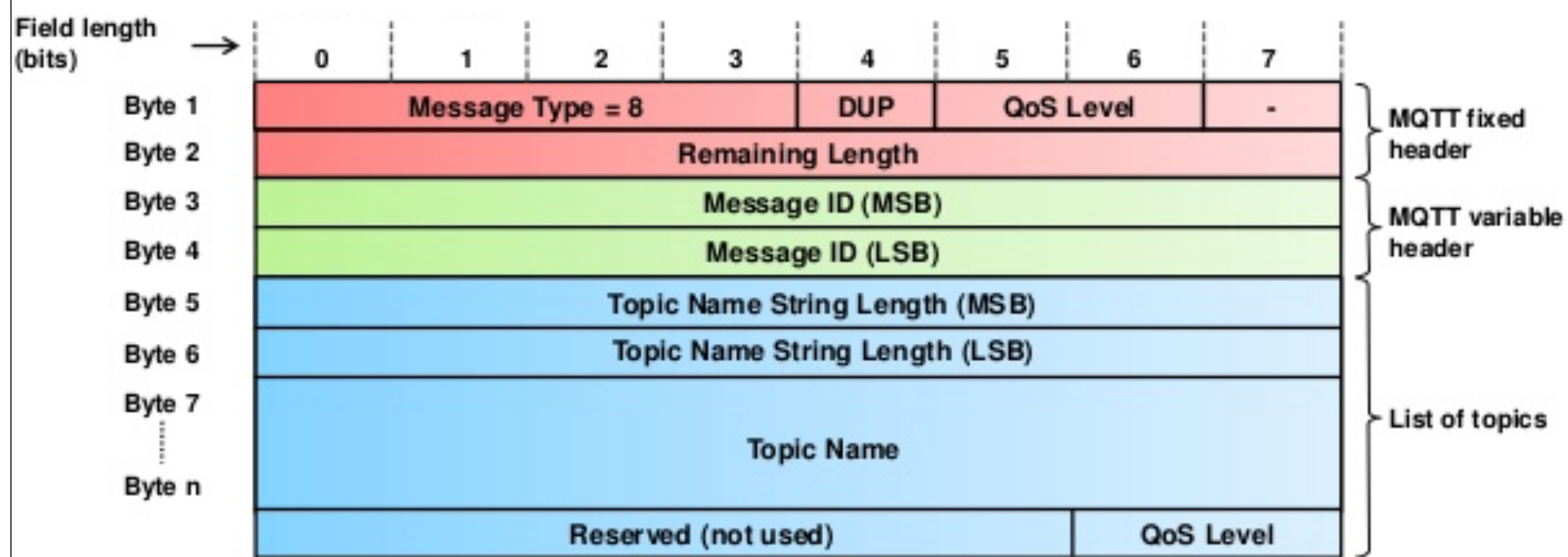
PUBLISH message format



Subscribe packet

- Packet Identifier: Chỉ cần với QoS > 0.
- List of Subscriptions: Một SUBSCRIBE message có thể chứa số lượng tùy ý các đăng ký (subscriptions). Mỗi đăng ký chứa topic và QoS level.

Định dạng gói tin SUBSCRIBE



Unsubscribe packet

- Packet Identifier: The ACK của một UNSUBSCRIBE packet sẽ có dùng packet id.
- List of Topics: Danh sách topics ngừng đăng ký

Topics

- Case sensitive
- UTF-8
- Wildcards
 - Single level: building/+/humidity
 - building/room_4/humidity
 - building/room_67/humidity
 - building/room_78/humidity
 - Multiple level (only at the end): building/room_number/#
 - building/room_4/wall/temperature
 - building/room_4/wall/humidity
 - building/room_4/ceiling/temperature
 - building/room_4/ceiling/humidity

Quality of Service (QoS)

- Thiết lập mức độ đảm bảo khi truyền thông báo:
 - 0) At most once: bên nhận không gửi ack và bên gửi không gửi lại thông báo
 - 1) At least once: Thông báo gửi ít nhất một lần, nhưng có thể gửi nhiều hơn một
 - 2) Exactly once: Mức độ QoS cao nhất, nhưng chậm nhất
- QoS được thiết lập bởi client, broker sẽ xác nhận lại. Vì vậy, mức QoS có thể bị giảm so với thiết lập của client

Quản lý hàng đợi và phiên làm việc (persistent session)

- Phiên được định danh bởi clientID
- Các tham số sau được lưu trong mỗi phiên :
 - Sự tồn tại của một phiên. Các đăng ký(subscriptions).
 - Tất cả các thông báo với QoS = 1 or 2 mà chưa nhận được xác nhận của client.
 - All new QoS 1 or 2 messages, which the client missed while offline.
 - All received QoS 2 messages, which are not yet confirmed to the client.

MQTT vs HTTP

- Push delivery of data / events:
 - MQTT low latency push from client to server and from server to client.
 - HTTP: Push from client to server but poll from server to client.
- Efficient use of network:
 - MQTT requires around 5 times less bytes than HTTP.
 - Reliable delivery: keeps QoS even across connection breaks.

MQTT is being used in:

- POS.
- Slot machines.
- Automotive / Telematics.
- Medical.
- Home Automation.
- Railway.
- Asset tracking / management. - Fire & Gas testing
- Amazon, Facebook, ...