

Message Queue Telemetry Transport (MQTT) (1)



Kim, Eui-Jik

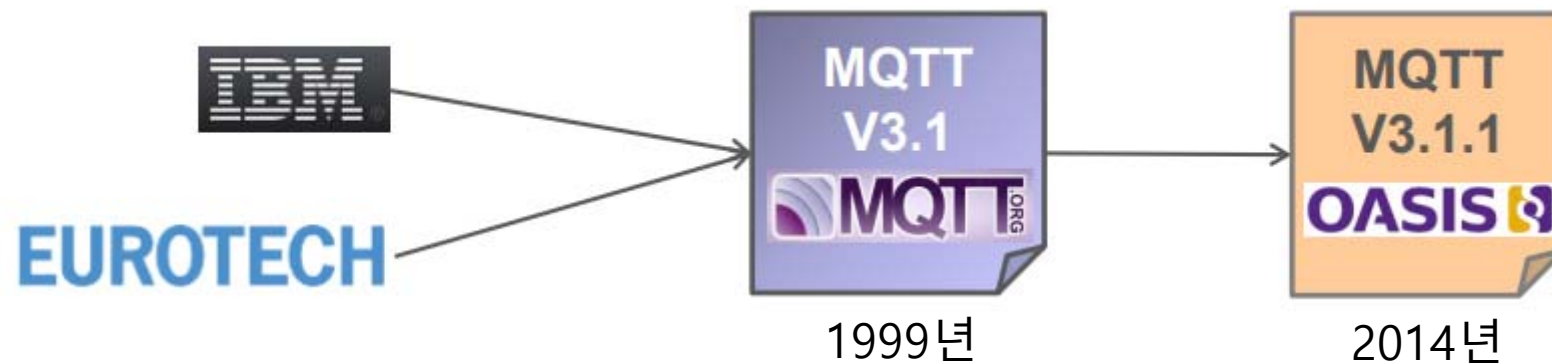
Contents

- Introduction to MQTT
- MQTT communication model
- Topic in MQTT
- Key features of MQTT

Introduction to MQTT

■ Origins and present of MQTT

- 1999년: IBM의 Andy Stanford-Clark와 Arcom (현 Eurotech) Arlen Nipper가 초기 버전의 MQTT 표준을 개발함. (MQTT Ver. 3.1)
- 2014년: 표준화 단체 Organization for the Advancement of Structured Information Standards (OASIS)가 MQTT를 공식적으로 표준화 함. (MQTT Ver. 3.1.1)



Introduction to MQTT

■ MQTT 란?

Message Queue Telemetry Transport (MQTT) is a machine-to-machine (M2M) / "Internet of Things (IoT)" connectivity protocol.

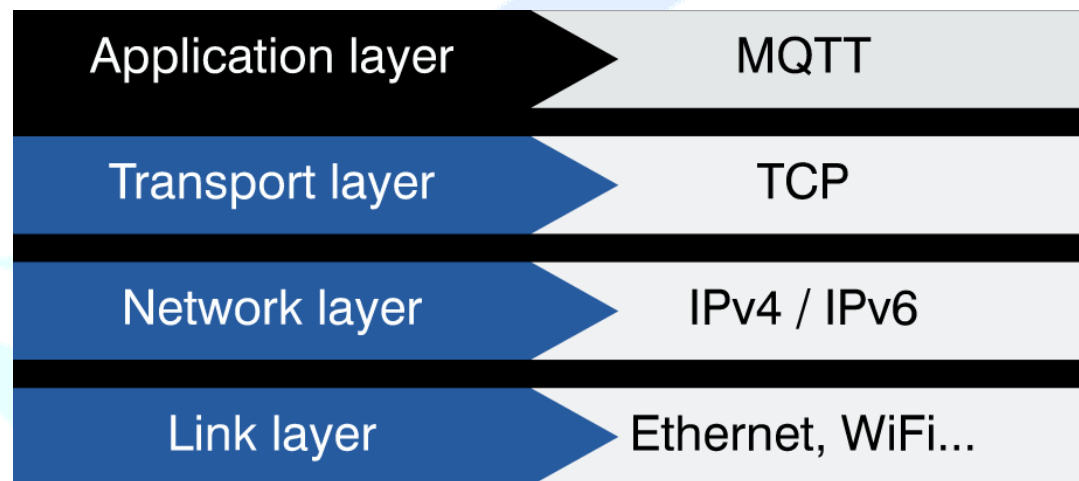
출처: <http://mqtt.org/>

- MQTT는 물리 리소스가 제한적인 디바이스 및 적은 대역폭, 긴 지연 등 신뢰성이 낮은 네트워크를 위해 고안됨.
- MQTT는 **Publish/Subscribe** 모델로 통신을 수행하며, CoAP과 더불어 IoT를 위한 대표적인 경량형 어플리케이션 프로토콜임.
- IoT 환경과 같이 다수의 저사양 센서 및 액추에이터들이 생산한 원격 측정 데이터 (Telemetry data)의 전송에 적합함.

Introduction to MQTT

■ Stack

- TCP 상에서 동작하는 Application layer 프로토콜



Introduction to MQTT

■ Message format

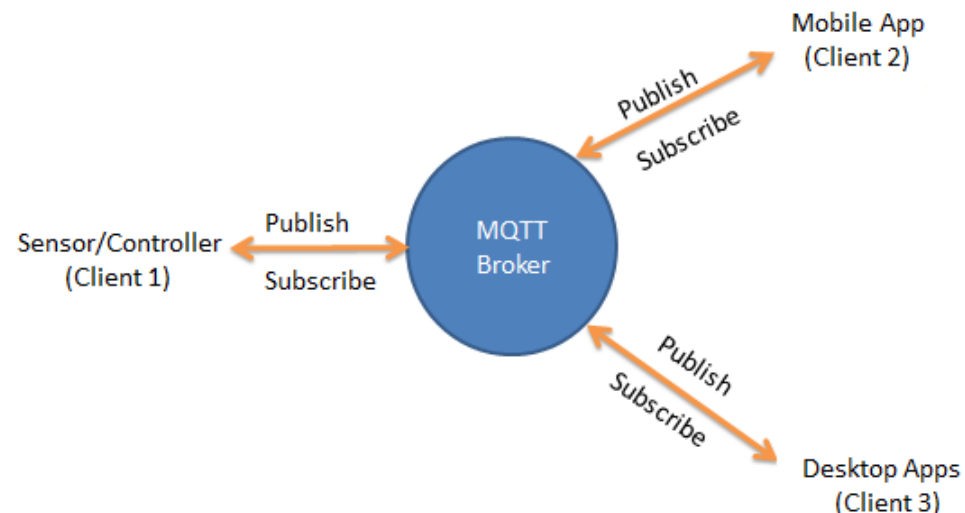
- Fixed header (2 bytes)
- Variable header (optional, 가변 길이)
- Message payload (optional, 최대 256MB)



MQTT communication model

■ Publish/Subscribe(Pub/Sub) model

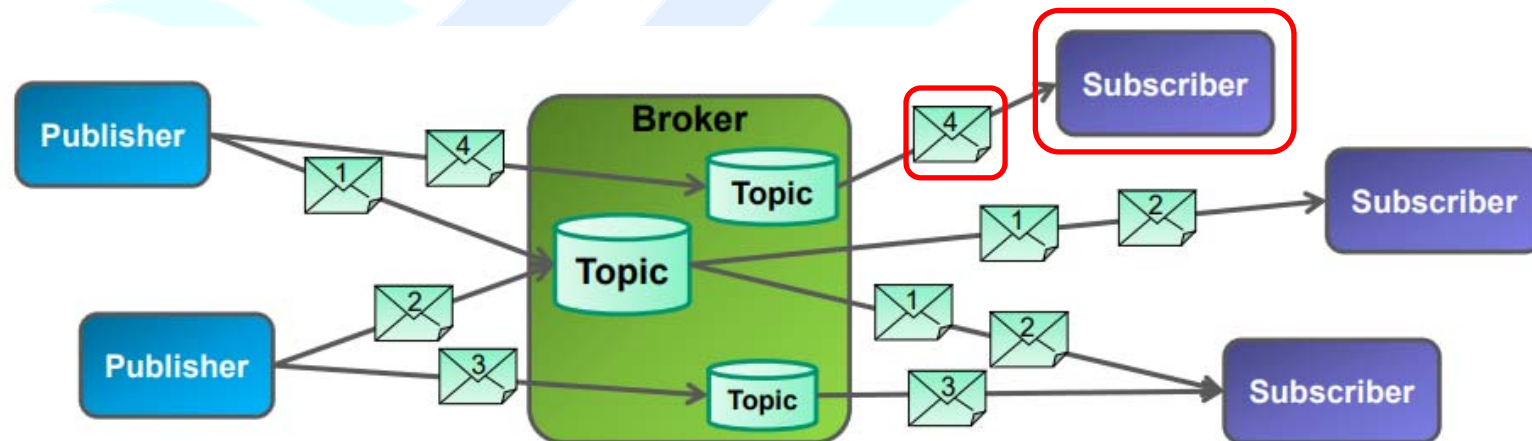
- MQTT는 Pub/Sub model을 통해 통신을 수행함.
- Client가 요청하고 Server가 데이터를 전송하는 기존의 Server/Client model은 다른 동작을 수행함.
- Publish/Subscribe model에서는 Client가 데이터를 전송하는 주체 (Publisher)가 될 수도 있고, 수신 받는 입장(Subscriber)이 될 수도 있음.
 - Client간 통신을 위해 데이터를 전달하는 **중계자(Broker)**가 존재함.



MQTT communication model

■ Topic in Pub/Sub model

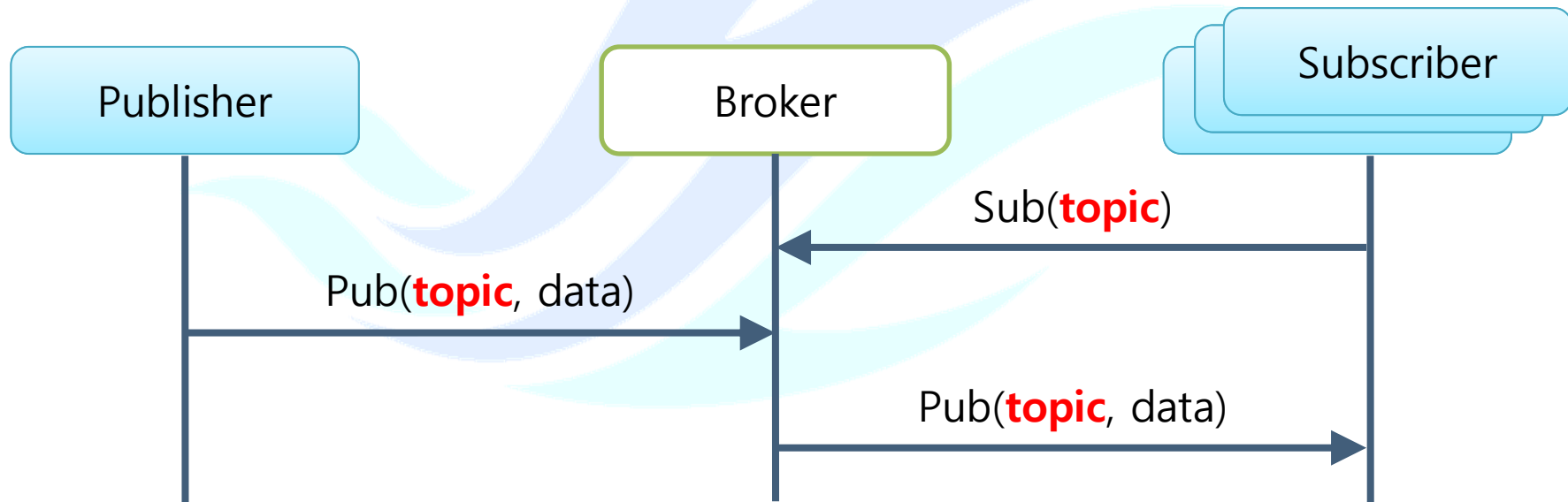
- Topic은 메시지 큐를 의미하며, 데이터마다 서로 다르게 표현될 수 있음.
- Client는 데이터 송수신의 모든 주체가 될 수 있기 때문에,
 - Topic을 통해 자신이 발행(Publish)하는 데이터를 나타낼 수 있고,
 - 원하는 데이터를 구독(Subscribe)하기 위해 특정 Topic을 선택 요청할 수 있음.
- Broker는 이러한 Topic을 중앙에서 관리하고, 구독/발행을 총괄함.
- Client는 Topic 별로 데이터를 주고 받음.
 - 특정 Subscriber가 4 라는 Topic에 해당하는 데이터 구독하고,
 - Publisher들이 1, 2, 3, 4 라는 Topic에 해당하는 데이터를 발행하면,
 - 4에 해당하는 데이터만 Broker를 통해 Subscriber로 전달됨.



MQTT communication model

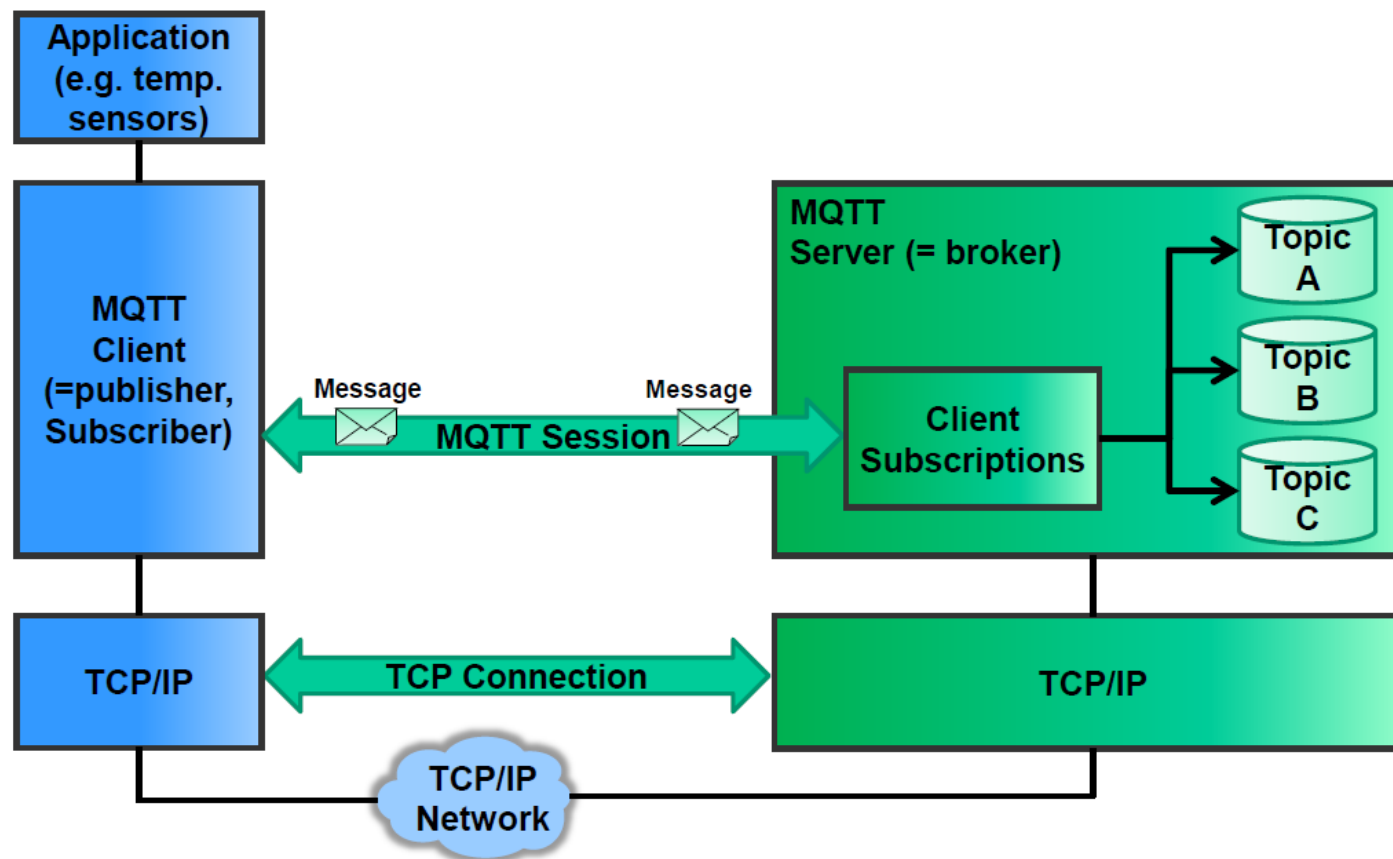
■ Operation of Pub/Sub model

- Subscriber들은 Broker에게 자신이 받을 Topic에 대한 정보를 요청
- Broker는 메모리에 Topic별로 구독하는 Subscriber들의 정보를 유지
- Publisher가 해당 Topic으로 데이터를 전송하면 구독하고 있던 Subscriber 전원에게 해당 데이터를 전달



MQTT communication model

■ MQTT communication



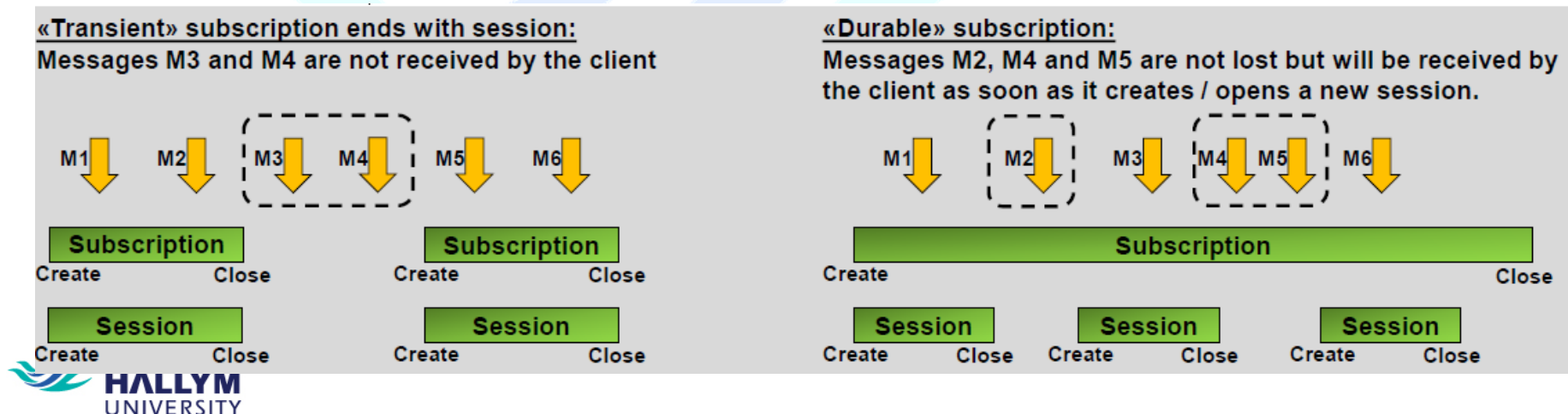
MQTT communication model

■ Session

- 세션은 Broker에 Client의 (일시적으로) 연결을 의미함.
- Client와 Broker 간의 모든 통신은 세션으로 이루어짐.

■ Subscription

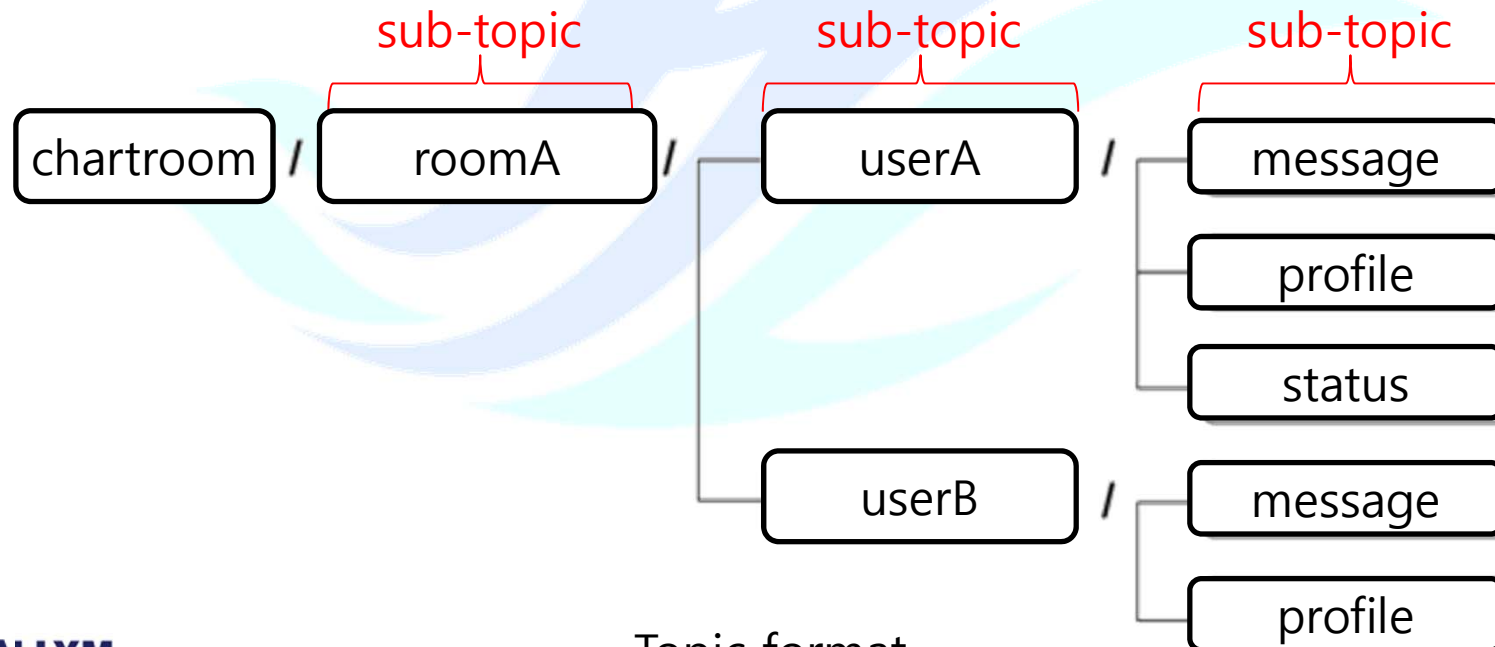
- Session과는 다르게, Subscription은 Topic과 Client의 (논리적) 연결을 의미함.
- «Transient» «Durable» 두 종류의 Subscription이 존재함.
 - Transient subscription : Session이 생성되어 있을 때만 메시지를 전송
 - Durable subscription : Session이 끊어졌을 때 메시지를 전송할 필요가 있으면, Broker는 해당 메시지를 저장 해 두었다가, 새로운 Session이 생성되었을 때 해당 메시지를 전송
 - Broker에 클라이언트의 구독 정보 저장되어, 재연결시 다시 구독하게 됨.



Topic in MQTT

■ Topic format

- Topic의 형태는 아래의 그림과 같이 트리구조를 가짐.
- Topic의 구분은 슬래쉬 "/"로 구분되며, 이는 topic level separator라고 부름.
- 각 하위는 "sub-topic" 혹은 "자식 노드"라고 부름.
- chatroom/roomA/userA/message를 구독하고자 하면 해당 Topic에 대한 데이터만 받을 수 있음.



Topic in MQTT

■ Wildcard in Topic

- 다수의 Topic을 구독하고자 할 때, 매번 요청하게 되면 높은 오버헤드가 발생할 수 있음. 이를 해결하기 위해 Wildcard를 활용할 수 있음.

■ Single Level

- "+" 라고 표기하며, 해당 레벨에 한하여 Topic을 확장할 수 있음.
- 2개 이상을 하나의 Topic에 사용할 수 있음. (ex. chatroom/+/+/profile)

chatroom/roomA/+/profile로 subscribe를 하면

chatroom/roomA/userA/message (X)
chatroom/roomA/**userA/profile** (O)
chatroom/roomA/userA/status (X)
chatroom/roomA/userB/message (X)
chatroom/roomA/**userB/profile** (O)

Topic in MQTT

■ Multi Level

- "#" 라고 표기하며, 해당 레벨 하위의 Topic을 모두 확장할 수 있음.

chatroom/roomA/# 로 subscribe를 하면

chatroom/roomA/userA/message (O)
chatroom/roomA/userA/profile (O)
chatroom/roomA/userA/status (O)
chatroom/roomA/userB/message (O)
chatroom/roomA/userB/profile (O)

- #은 Topic의 마지막에만 사용 가능함.
 - chatroom/roomA/#/profile 는 유효하지 않음.

Key features of MQTT

■ MQTT Characteristics

- 낮은 오버헤드 (2 Bytes header)
- Transmission Control Protocol (TCP) 상에서 동작
- 이벤트 (Message) 기반 비동기 통신
- Publish/Subscribe (Pub/Sub) 통신 모델
- Broker Server와 Client (Publisher/Subscriber)로 구성
- Topic (Message Queue)을 통해 Publisher 및 Subscriber가 구분

