

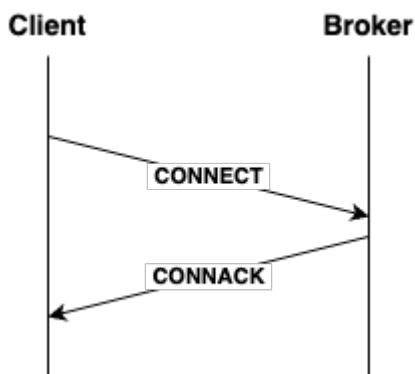
# MQTT 프로토콜

## 1. MQTT란?

- MQTT (Message Queuing Telemetry Transport)는 경량 프로토콜.
- IoT(사물 인터넷) 및 M2M(Machine-to-Machine) 통신에서 널리 사용.
- 특징
  - 저전력 & 저대역폭: 소형 디바이스에서도 원활하게 작동.
  - Pub/Sub 모델: 브로커를 통해 발행(Publish)과 구독(Subscribe) 방식으로 메시지 전달.
  - QoS 지원: 메시지 전송의 신뢰성을 보장하는 3가지 품질(QoS 0, 1, 2) 제공.
  - 지속적 연결 유지: PINGREQ/PINGRESP를 통한 연결 상태 확인 가능.

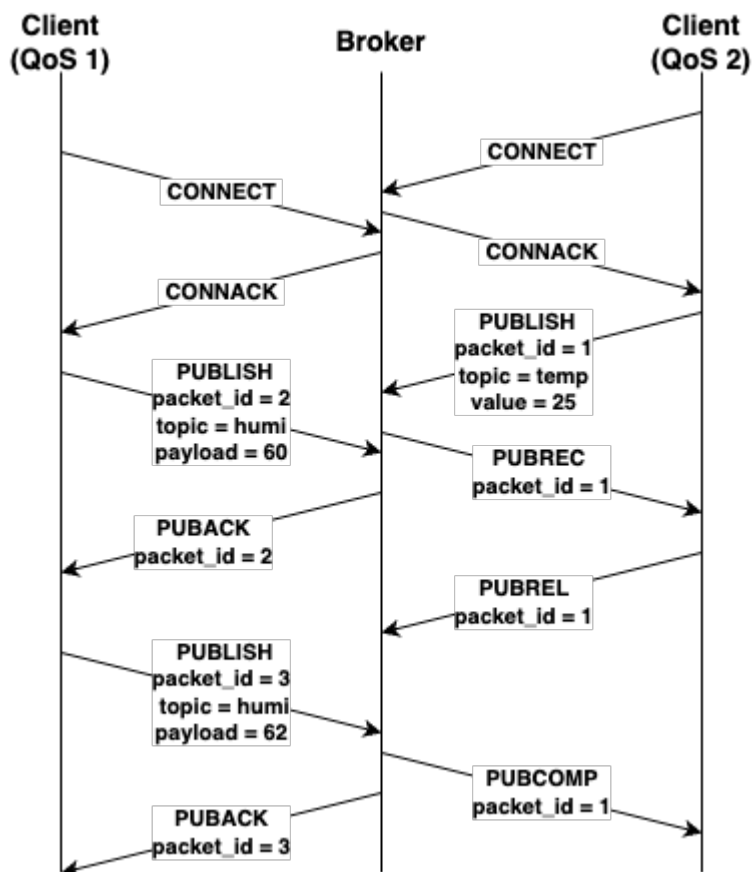
## 2. MQTT 동작 흐름

### 2.1 클라이언트 연결 (CONNECT → CONNACK)



- 절차
  1. 클라이언트가 브로커에 연결 요청  
CONNECT 패킷 전송 → 클라이언트 ID, 인증 정보, KeepAlive 설정 포함.
  2. 브로커가 요청을 승인 또는 거부  
CONNACK 패킷 응답 → 성공 시 연결 유지, 실패 시 거부 코드 전송.

### 2.2 메시지 발행 (PUBLISH → PUBACK/PUBREC/PUBREL/PUBCOMP)



#### 절차

1. 클라이언트가 브로커로 메시지 전송 (PUBLISH)

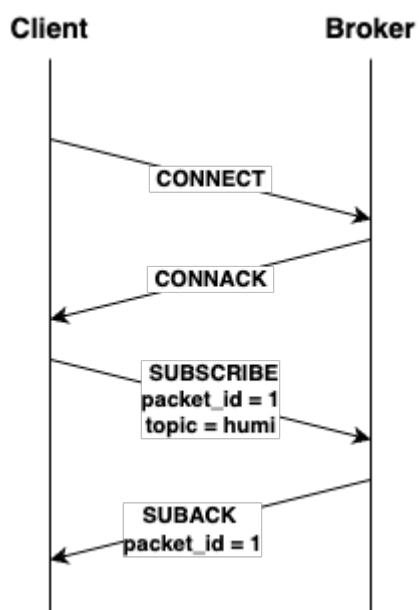
2. QoS에 따라 브로커가 메시지 확인

QoS 0: 즉시 전송, 확인 없음.

QoS 1: PUBACK으로 전송 확인.

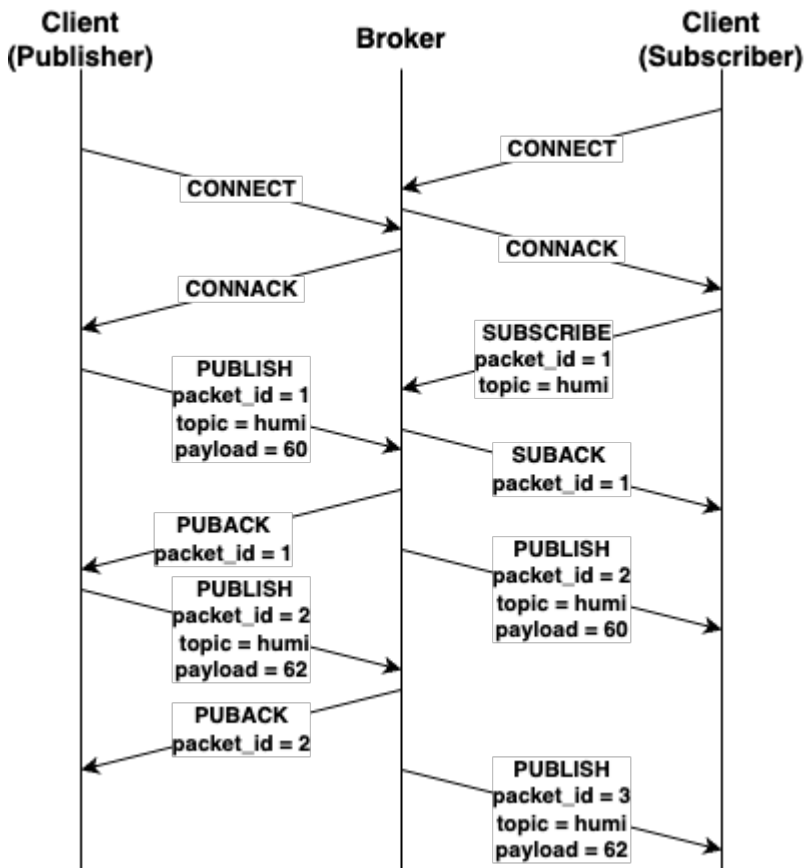
QoS 2: PUBREC → PUBREL → PUBCOMP 순서로 확인.

## 2.3 메시지 구독 (SUBSCRIBE → SUBACK)



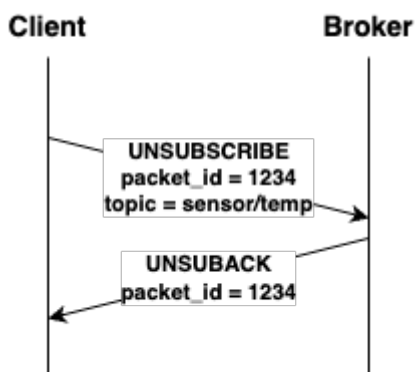
- 절차
  - 클라이언트가 브로커에 특정 토픽 구독 요청 (SUBSCRIBE)
  - 브로커가 요청을 승인하고 SUBACK 응답
    - 구독 요청에 대한 QoS 레벨을 반환.
    - 구독 실패 시 0x80 반환.

## 2.4 메시지 배포 (PUBLISH → SUBSCRIBERS)



- 절차
  - 브로커가 구독 중인 모든 클라이언트에게 메시지 전달
  - QoS에 따라 응답 확인 (QoS 1 → PUBACK, QoS 2 → PUBREC/PUBREL/PUBCOMP).

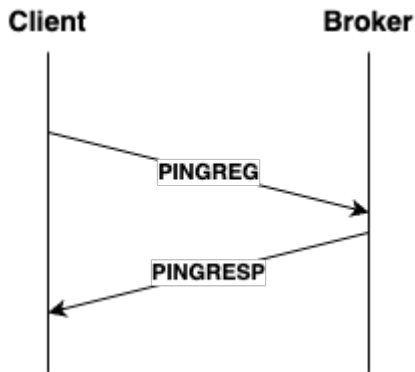
## 2.5 구독 취소 (UNSUBSCRIBE → UNSUBACK)



- 절차

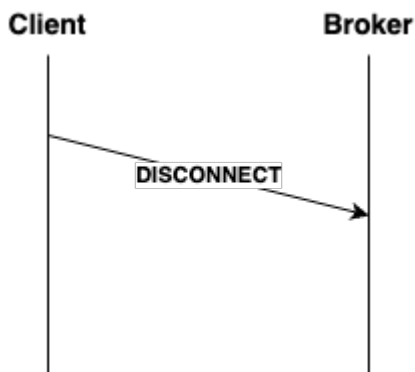
1. 클라이언트가 특정 토픽의 구독을 취소 (UNSUBSCRIBE)
2. 브로커가 UNSUBACK 패킷을 응답하여 확인.

## 2.6 연결 유지 (PINGREQ → PINGRESP)



- 절차
  1. 클라이언트가 일정 시간 동안 활동이 없으면 PINGREQ 전송
  2. 브로커가 PINGRESP 응답을 보내 연결이 정상적임을 확인.

## 2.7 연결 종료 (DISCONNECT)



- 절차
  1. 클라이언트가 연결을 종료할 때 DISCONNECT 패킷 전송
  2. 브로커는 클라이언트 세션을 종료.

## 3. QoS (Quality of Service)

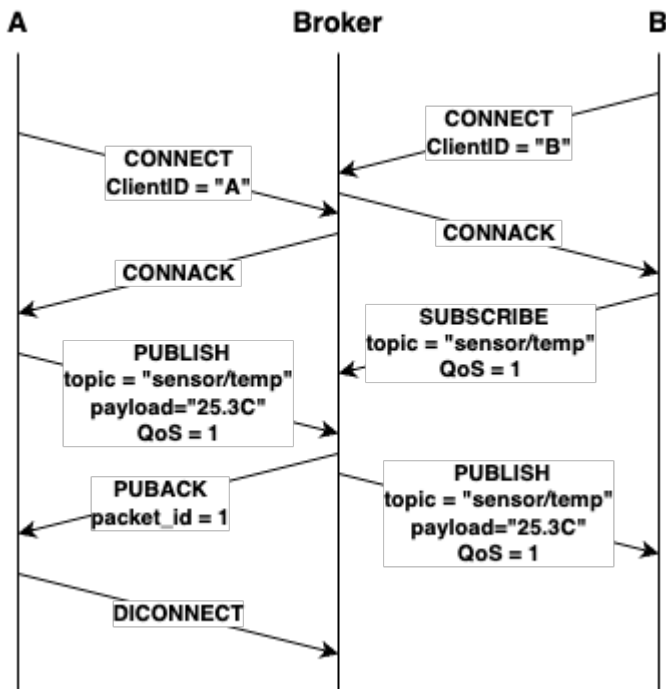
- MQTT는 메시지 전달 신뢰성을 보장하기 위해 3가지 QoS 레벨을 제공합니다.

QoS 레벨	설명	패킷 흐름
QoS 0	최소 전송 (At most once) (손실 가능)	PUBLISH
QoS 1	최소 1회 전송 (At least once)	PUBLISH → PUBACK
QoS 2	정확히 1회 전송 (Exactly once)	PUBLISH → PUBREC → PUBREL → PUBCOMP

- QoS 2는 가장 높은 신뢰성을 제공하지만, 4단계 메시지 교환이 필요하여 성능이 저하될 수 있음.

## 4. MQTT 동작 예제 (전체 흐름)

### 4.1 예제 시나리오 (QoS 1)



1. 클라이언트 A가 MQTT 브로커에 연결.
2. 클라이언트 B가 "sensor/temp" 토픽을 QoS 1로 구독.
3. 클라이언트 A가 "sensor/temp" 토픽에 "25.3C" 메시지를 QoS 1로 발행.
  - A → PUBLISH(QoS 1) → 브로커
  - 브로커 → PUBACK → A (Publisher가 PUBACK을 받음)
4. 브로커가 B에게 메시지 전달.
  - 브로커 → PUBLISH(QoS 1) → B
5. 클라이언트 A가 연결 종료.
  - 이 과정에서 MQTT는 Pub/Sub 모델을 통해 메시지를 효율적으로 전송 및 관리.

## 5. MQTT 패킷 구조

- MQTT 패킷은 고정 헤더(Fixed Header), 가변 헤더(Variable Header), 페이로드(Payload) 세 부분으로 구성됩니다.

MQTT 패킷은 다음과 같은 구조로 구성됩니다:

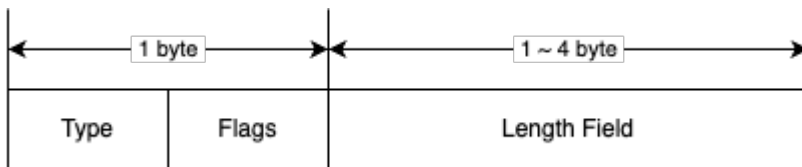
<b>Fixed Header</b>
<b>Variable Header</b>
<b>Payload</b>

- Fixed Header (고정 헤더): 패킷의 기본 정보를 포함 (패킷 타입, 플래그, 길이 등)
- Variable Header (가변 헤더): 특정 패킷에서만 사용되는 추가 정보 (예: Packet ID, Topic Name)
- Payload (페이로드): 메시지의 실제 데이터 (일부 패킷에서만 사용)

## 5.1 Fixed Header (고정 헤더)

- MQTT 패킷은 고정 크기의 헤더(Fixed Header)를 가짐.

### 5.1.1 Fixed Header 구조 (2 바이트 이상)



필드	크기	설명
Packet Type	4비트	MQTT 패킷 유형 (CONNECT, PUBLISH 등)
Flags	4비트	패킷 별 추가 플래그
Remaining Length	1~4 바이트	전체 패킷 크기 (가변 길이 인코딩)

### 5.1.2 Packet Type (4비트)

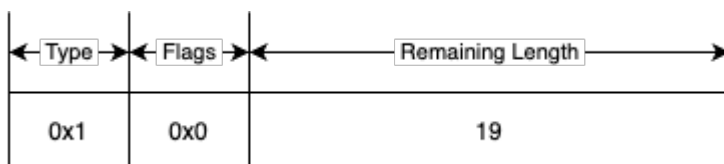
- MQTT 패킷의 종류를 나타내는 4비트 값.

패킷 타입	16진수 값	설명
CONNECT	0x1	클라이언트가 브로커에 연결 요청
CONNACK	0x2	브로커가 클라이언트의 CONNECT 요청을 승인 또는 거부
PUBLISH	0x3	클라이언트 또는 브로커가 메시지를 발행
PUBACK	0x4	PUBLISH 메시지(QoS 1)에 대한 응답 (ACK)
PUBREC	0x5	PUBLISH 메시지(QoS 2) - 1단계 (Received)
PUBREL	0x6	PUBLISH 메시지(QoS 2) - 2단계 (Release)
PUBCOMP	0x7	PUBLISH 메시지(QoS 2) - 3단계 (Complete)
SUBSCRIBE	0x8	클라이언트가 특정 토픽을 구독 요청
SUBACK	0x9	브로커가 SUBSCRIBE 요청을 승인 또는 거부
UNSUBSCRIBE	0xA	클라이언트가 구독을 취소

UNSUBACK	0xB	브로커가 UNSUBSCRIBE 요청을 승인
PINGREQ	0xC	클라이언트가 브로커에 연결 유지 요청
PINGRESP	0xD	브로커가 PINGREQ에 대한 응답
DISCONNECT	0xE	클라이언트가 연결 해제 요청

### 5.1.3 Flags (4비트)

- MQTT 메시지의 추가적인 속성을 설정하는 4비트 필드입니다.
- 예를 들어, PUBLISH 패킷에서는 QoS 레벨 및 Retain 플래그를 나타냅니다.



- DUP: 중복 메시지 여부
- QoS: 메시지의 QoS 레벨 (0, 1, 2)
- RETAIN: 메시지를 보관할지 여부

### 5.1.4 Remaining Length (1~4 바이트)

- 전체 MQTT 패킷 크기를 나타냄.
- 가변 길이 인코딩 (Variable Length Encoding) 사용.
- 7비트씩 사용하고, 마지막 바이트는 MSB=0.
- 예제
  - 0x7F (127) → 1 바이트 사용
  - 0x80 0x01 (128) → 2 바이트 사용
  - 0x80 0x80 0x01 (16384) → 3 바이트 사용

## 5.2 Variable Header (가변 헤더)

- 가변 헤더는 패킷 유형에 따라 필요할 때만 포함.
- 예제
  - CONNECT → 프로토콜 버전, 클라이언트 ID 포함
  - PUBLISH → 토픽 이름, 패킷 ID 포함

## 5.3 Payload (페이로드)

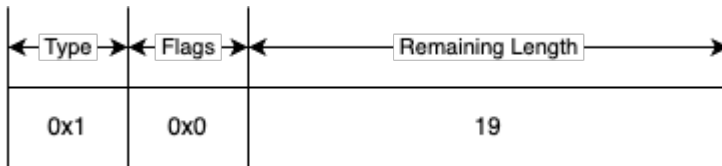
페이로드는 실제 전송할 데이터가 들어가는 영역입니다.

- 예제
  - CONNECT 패킷 → 클라이언트 ID, 사용자 이름, 비밀번호
  - PUBLISH 패킷 → 메시지 내용
  - SUBSCRIBE 패킷 → 구독할 토픽 리스트

## 5.4 MQTT 패킷 예제

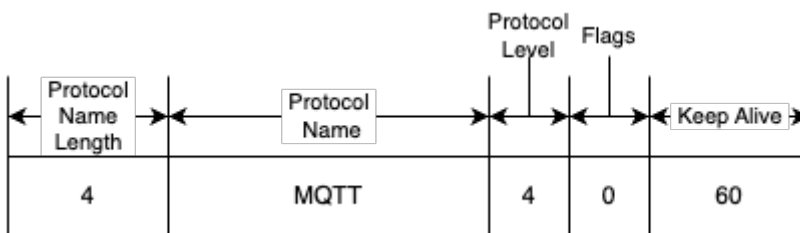
### 5.4.1 CONNECT 패킷 (클라이언트 연결 요청)

- Fixed Header



- Control Packet Type: CONNECT (0x01)
- Remaining Length : variable header 길이 + payload 길이(19)

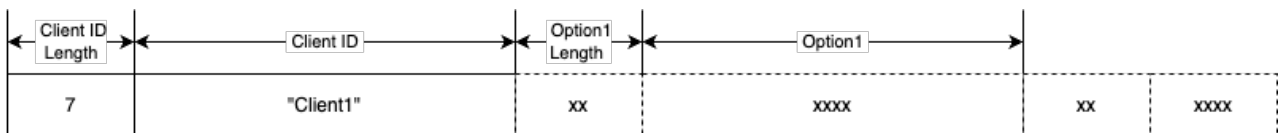
- Variable Header



- Protocol Name Length: 프로토콜 이름 길이("4")
- Protocol Name: 프로토콜 이름("MQTT")
- Flags: 사용자 이름, 패스워드, Retain, QoS, Will Flag, 클린 세션, 인증 정보 포함 여부
- KeepAlive: 연결 유지 시간(60초)

연결 유지 시간을 지나서까지 계속 유지하기 위해서는 연결 확인 메시지 전송(PINGREQ)

- Payload

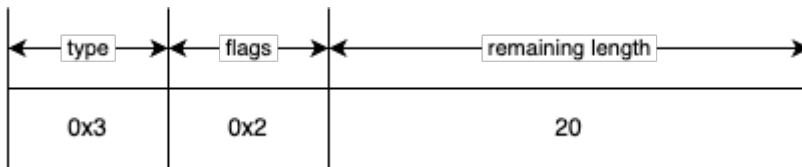


- Client ID Length: 클라이언트 ID 길이(6)
- ClientID: 클라이언트 ID("Client1")
- 옵션(길이 + 데이터)
  - User Name: 사용자 이름
  - Password: 패스워드
  - Will Topic: Will flag 설정시, 저장할 topic
  - Will Message: Will flag 설정시, 저장할 메시지

### 5.4.2 PUBLISH 패킷 (메시지 전송)

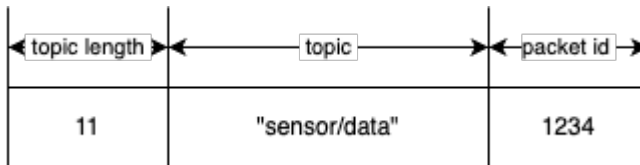
- Fixed Header





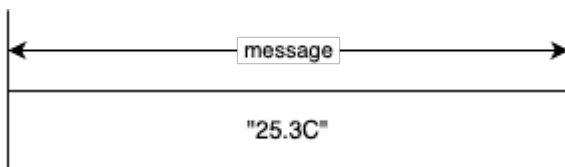
- Control Packet Type: PUBLISH(0x03)
- Flags: DUP(0), QoS(1), RETAIN(0)
- Remaining Length: variable header 길이 + payload 길이(20)

• Variable Header



- Topic Length: 메시지 주제 길이(11)
- Topic: 메시지 주제("sensor/data")
- Packet ID: 패킷 식별자(1234)

• Payload



- Message: 메시지("25.3C")

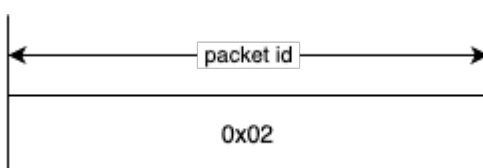
### 5.4.3 PUBACK (QoS 1 메시지 확인) 패킷 구조

- PUBACK 패킷은 QoS 1에서 PUBLISH 메시지를 정상적으로 받았음을 확인하는 응답입니다.
- FixedHeader



- Control Packet Type: PUBACK(0x4)
- Flags: 없음
- Remaining Length: variable header 길이(2)

• Variable Header

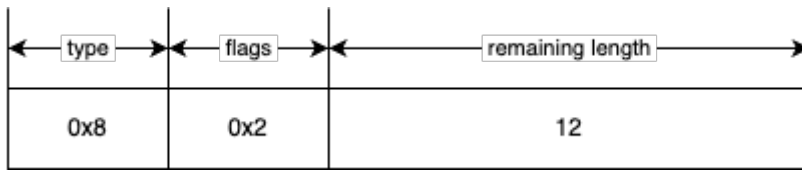


- Packet ID: PUBLISH에 포함되어 있는 패킷 식별자(2)

- PUBACK을 받으면 PUBLISH 메시지가 성공적으로 전달된 것으로 간주됨.

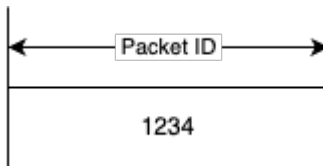
## 5.4.4 SUBSCRIBE 패킷 (구독 요청)

- Fixed Header



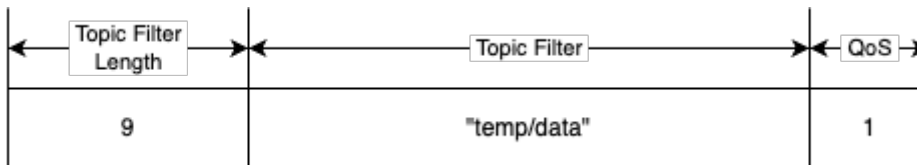
- Control Packet Type: SUBSCRIBE(0x8)
- Flags: 고정(0x2)
- Remaining Length: variable header 길이 + payload 길이(12)

- Variable Header



- Packet ID: 패킷 식별자(2 바이트)

- Payload

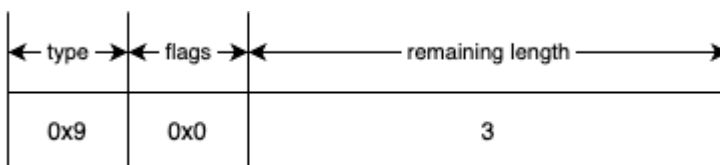


- Topic Filter Length: 수신할 주제("temp/data")의 길이
- Topic Filter: 수신할 주제("temp/data")
- Requested QoS: 메시지 송수신 품질 수준(1)

## 5.4.5 SUBACK (구독 요청 응답) 패킷 구조

- SUBACK 패킷은 클라이언트가 보낸 SUBSCRIBE 요청에 대한 브로커의 응답.

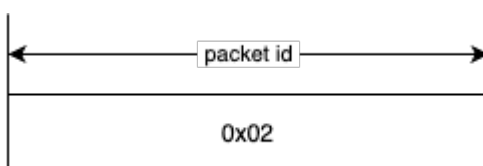
- Fixed Header



Control Packet Type: SUBACK(0x9)

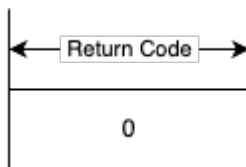
Flags: 없음 \*\* Remaining Length: variable header 길이 + payload 길이(3)

- Variable Header



- Packet ID : SUBSCRIBE 요청시 받은 패킷 식별자

- Payload



○ Return Code: 각 토픽에 대한 구독 결과 (QoS 수준 또는 실패 코드)

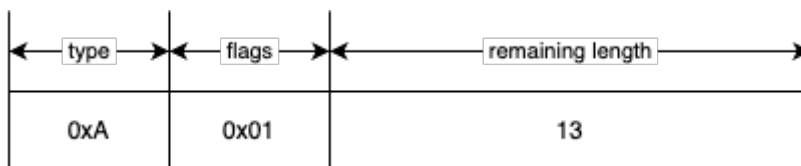
값	설명
0x00	QoS 0으로 구독 성공
0x01	QoS 1으로 구독 성공
0x02	QoS 2로 구독 성공
0x80	구독 실패

• SUBACK을 받으면 클라이언트는 해당 토픽을 성공적으로 구독한 것으로 간주.

## 5.4.6 UNSUBSCRIBE (구독 취소 요청) 패킷 구조

• UNSUBSCRIBE 패킷은 클라이언트가 특정 토픽의 구독을 취소할 때 사용.

- Fixed Header

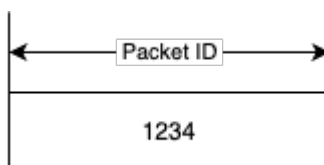


○ Control Packet Type: UNSUBSCRIBE (0xA)

○ Flags: 0x01 고정

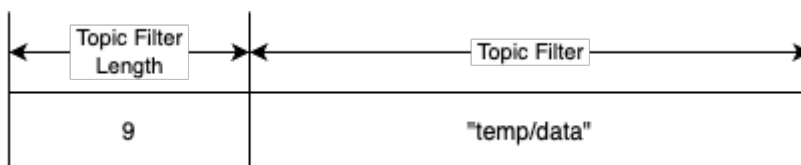
○ Remaining Length: variable header 길이 + payload 길이(13)

- Variable Header



○ Packet ID: 구독 취소 요청의 ID

- Payload



○ Topic Filters: 구독을 취소할 토픽 리스트

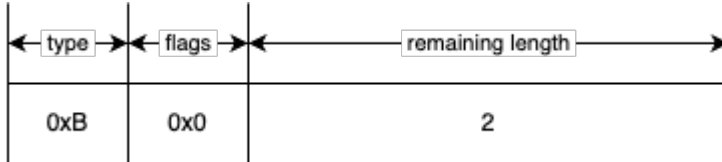
토픽 필터 길이와 토픽 필터

하나 이상 반복될 수 있음

- 브로커는 클라이언트의 UNSUBSCRIBE 요청을 처리한 후, UNSUBACK을 응답으로 보냅니다.

## 5.4.7 UNSUBACK (구독 취소 요청 응답) 패킷 구조

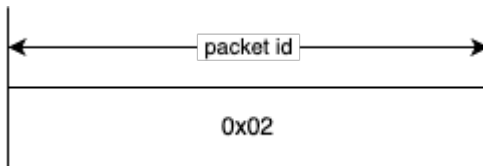
- UNSUBACK 패킷은 클라이언트가 보낸 UNSUBSCRIBE 요청에 대한 브로커의 응답.
- Fixed Header



Control Packet Type: UNSUBACK(0xB)

Flags: 없음 (0) \*\* Remaining Length: variable header 길이 (2)

- Variable Header

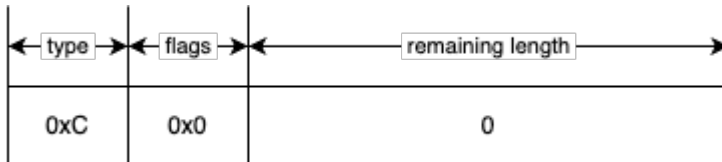


○ Packet ID : UNSUBSCRIBE 요청시 받은 패킷 식별자

- UNSUBACK을 받으면 클라이언트는 해당 토픽을 성공적으로 구독 취소한 것으로 간주.

## 5.4.8 PINGREQ (연결 유지 요청) 패킷 구조

- PINGREQ 패킷은 클라이언트가 브로커에 연결이 유지되고 있는지 확인하기 위해 보냄.
- Fixed Header



○ Control Packet Type: PINGREQ(0xC)

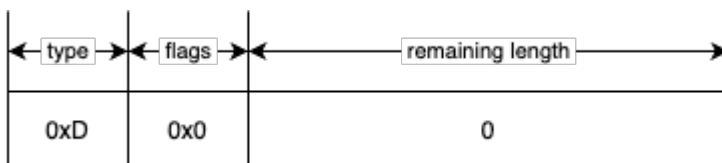
○ Flags: 0x00 고정

○ Remaining Length: 0

- 브로커는 PINGRESP 패킷을 응답으로 보냄.

## 5.4.9 PINGRESP (연결 유지 응답) 패킷 구조

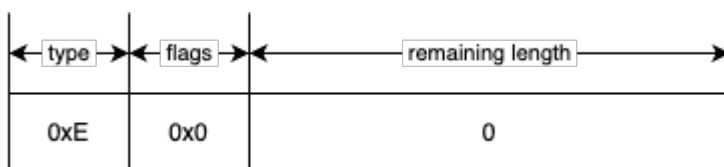
- PINGRESP 패킷은 브로커가 PINGREQ 요청을 받았을 때 응답하는 메시지입니다.
- Fixed Header



- Control Packet Type: PINGRESP(0xD)
- Flags: 0x00 고정
- Remaining Length: 0
- PINGRESP를 받으면 클라이언트는 연결이 정상적으로 유지되고 있음을 확인 가능.

## 5.4.10 DISCONNECT (연결 끊기) 패킷 구조

- DISCONNECT 패킷은 클라이언트가 브로커에게 연결을 끊으면 보내는 마지막 메시지.
- Fixed Header



- Control Packet Type: DISCONNECT(0xE)
- Flags: 0x00 고정
- Remaining Length: 0