

MQTT Protocol (2)



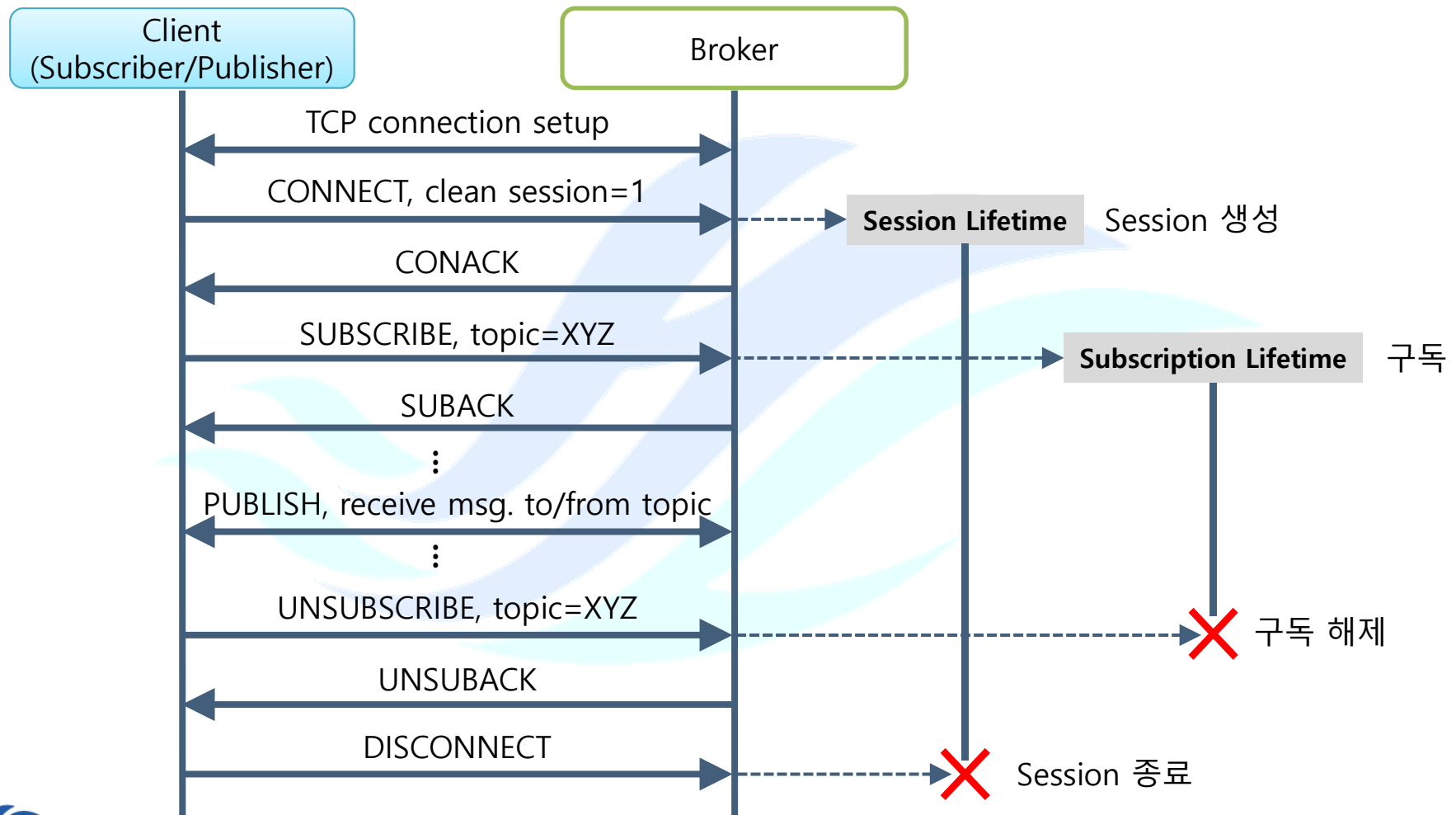
Kim, Eui-Jik

Contents

- MQTT operation
- Message format
- Fixed header
 - Message Type
 - Duplication message flag (DUP)
 - QoS Level
 - RETAIN

MQTT operation (example)

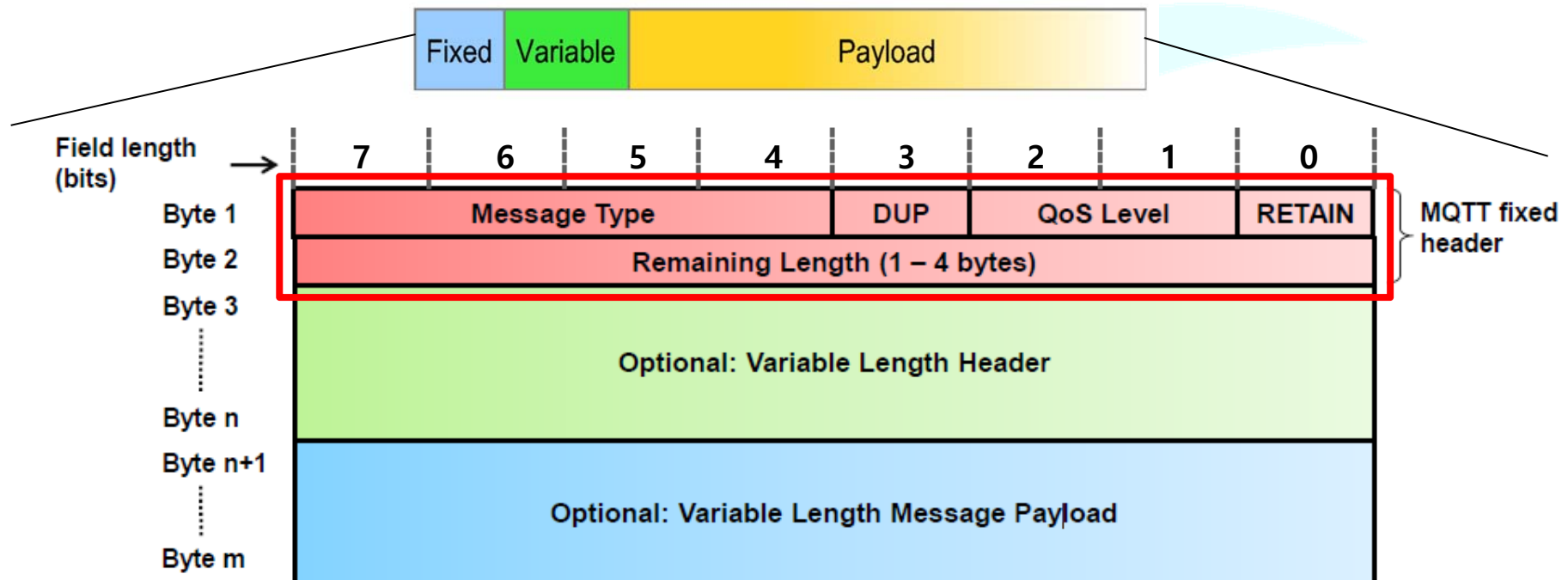
* **Case:** «Transient» Subscription (clean session flag=1)



Message format

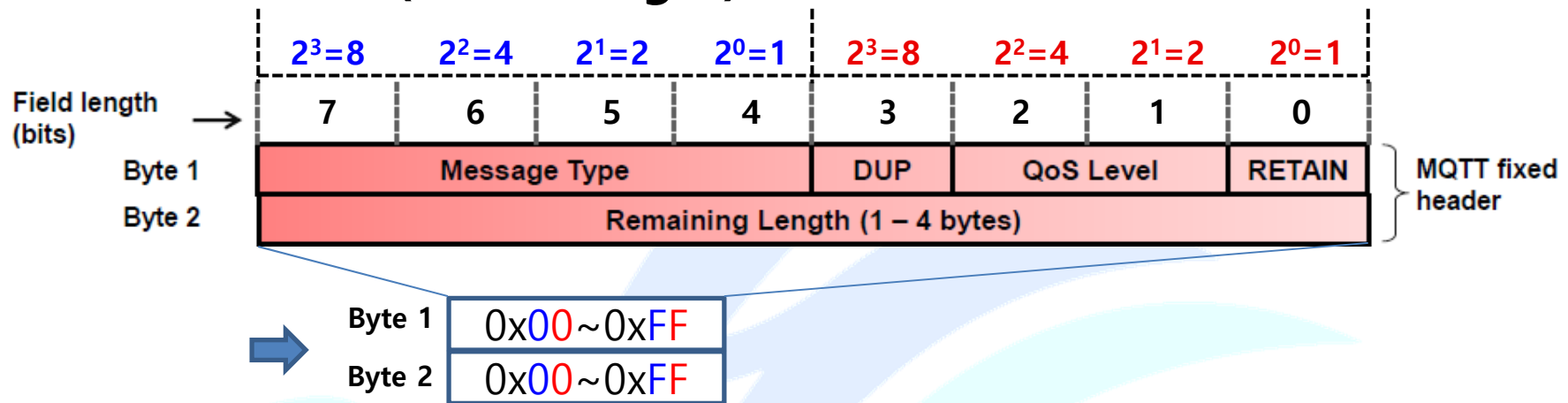
■ Message format

- Fixed header [2bytes]
 - MQTT Control Packet type [4bits], Flags [4bits], Remaining length [8bits]
- (Optional) Variable header [Variable length]
- (Optional) Payload [Variable length]



[참고] background

■ 8비트형 정수(8-bit integer)의 16진수 표현법



■ Example

- Byte 1: (1110 0000) → (14 0) → (0xE0)
 - Message Type=1110(14), DUP=0, QoS Level=00, RETAIN=0
- Byte 2: (0111 1111) → (7 15) → 0x7F
 - Remaining Length (RL) = 01111111 → 256 MBytes (의미)
 - * 다음주 내용
 - RL = Variable header length + Payload length

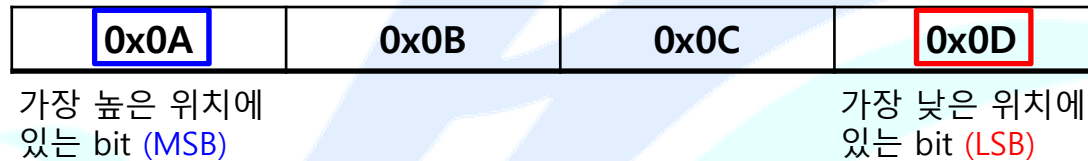
[참고] background

■ MSB와 LSB

- MSB: Most Significant Byte(최상위바이트),
LSB: Least Significant Bit(최하위바이트)

- 32-bit integer: 0x0A0B0C0D

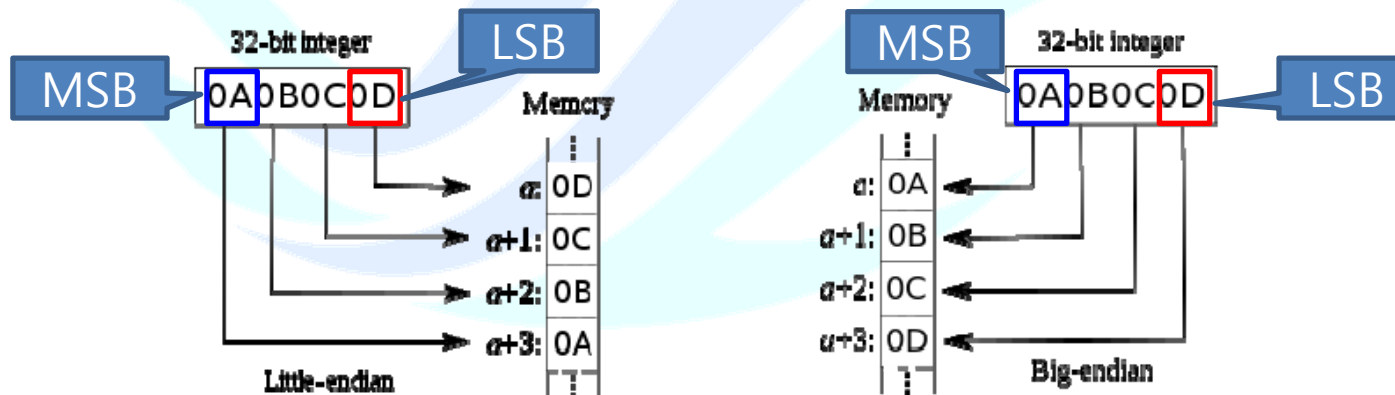
- Memory 에 저장될 때 1Byte(8bits) 단위로 잘려져 저장



[참고] background

■ 엔디안

- **Endian** : 메모리에 저장되는 바이트들의 순서 (Byte Order)를 나타냄
- **Little Endian(리틀엔디안)** : MSB(Most Significant Byte; 최상위 바이트)가 높은 주소에 위치하는 방식, x86 CPU
- **Big Endian(빅엔디안)** : MSB(Most Significant Byte; 최상위 바이트)가 낮은 주소에 위치하는 방식, RISC CPU



[참고] background

■ Little Endian & Big Endian 예시

■ Little Endian

- LSB (least significant byte)부터 저장하는 방식
- 낮은(시작) 주소에 하위 바이트부터 기록, Intel CPU 계열

예) 32비트형 4바이트 값 : 0x12345678

하위 주소	0x0001	0x0002	0x0003	0x0004	상위 주소
하위바이트	78	56	34	12	상위바이트
< -----					

■ Big Endian

- MSB (most significant byte)부터 저장하는 방식
- 패킷 분석을 하면서 많이 보게 되는 네트워크 프로토콜은 기본적으로 빅 엔디안 표현
- 낮은(시작) 주소에 상위 바이트부터 기록, RICS CPU

예) 32비트형 4바이트 값 : 0x12345678

하위 주소	0x0001	0x0002	0x0003	0x0004	상위 주소
상위바이트	12	34	56	78	하위바이트
----- >					

Fixed header-Message Type

■ Message type field

■ 4-bit unsigned value

- 총 16가지의 타입을 정의할 수 있으며, 0을 제외한 15개의 Type을 정의

■ 메시지의 타입을 결정할 때 사용

- 메시지 타입마다 서로 다른 사용 목적을 가지며, 고정 값으로 표현
- 메시지 타입마다 전송 흐름을 정의 (Client → Server 또는 Server → Client)

■ 메시지 타입 상세

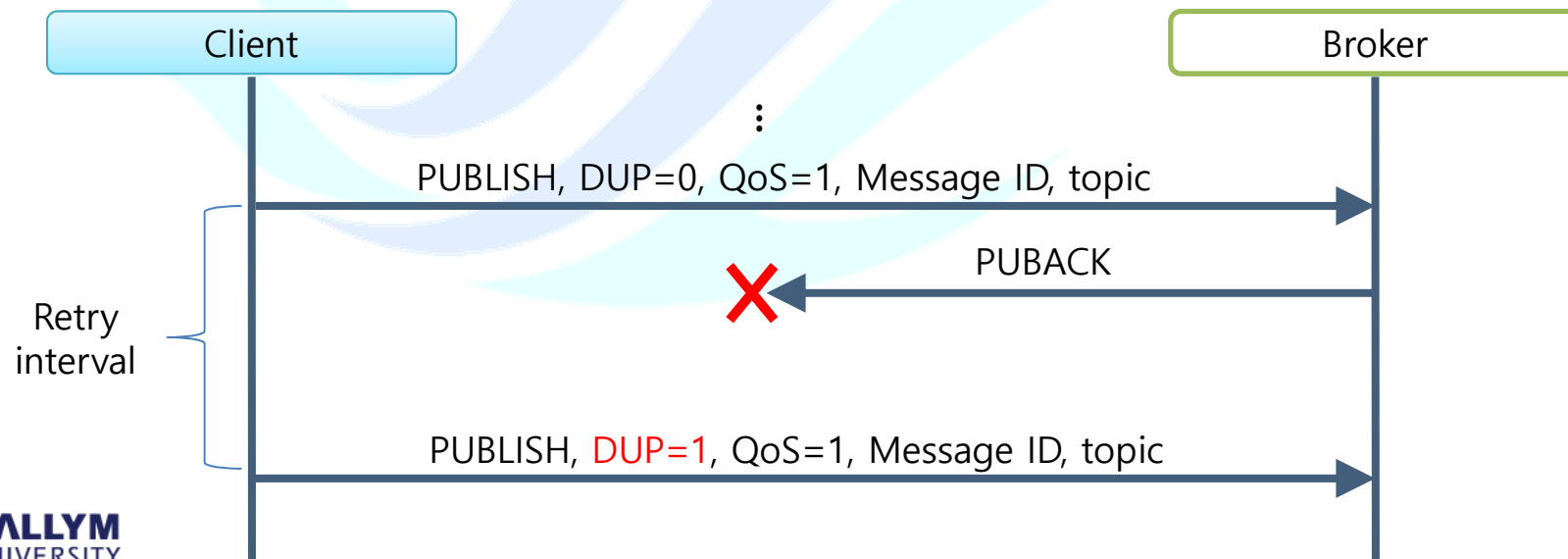
Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client → Server	Connection request
CONNACK	2	Server → Client	Connect acknowledgment
PUBLISH	3	Both Direction	Publish message
PUBACK	4	Both Direction	Publish acknowledgment [QoS level 1]
PUBREC	5	Both Direction	Publish received (assured delivery part 1) [QoS level 2]
PUBREL	6	Both Direction	Publish release (assured delivery part 2) [QoS level 2]
PUBCOMP	7	Both Direction	Publish complete (assured delivery part 3) [QoS level 2]
SUBSCRIBE	8	Client → Server	Subscribe request
SUBACK	9	Server → Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client → Server	Unsubscribe request
UNSUBACK	11	Server → Client	Unsubscribe acknowledgment
PINGREQ	12	Client → Server	PING request
PINGRESP	13	Server → Client	PING response
DISCONNECT	14	Client → Server	Disconnect notification
AUTH	15	Both Direction	Authentication exchange

Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-DUP

■ Duplicate message flag (DUP) field

- 1-bit unsigned value
- 송신 메시지의 중복 여부를 알릴 때 사용 (재전송 시 사용)
 - Client 또는 Broker가 이미 송신된 특정 메시지에 대한 응답 확인(ACK 등)을 정해진 기간 동안 받지 못했을 경우, 동일 메시지를 재전송하며, 이 때 DUP 필드를 1로 채워서 해당 메시지를 재전송
 - PUBLISH에서만 사용
 - QoS Level이 0보다 클 때만 사용 (QoS 1 또는 QoS 2)
 - Message ID로 중복 메시지를 식별하기 때문에, Message ID를 반드시 포함
 - Message ID는 variable header에, topic은 payload에 포함됨.



Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-QoS Level

■ QoS Level field

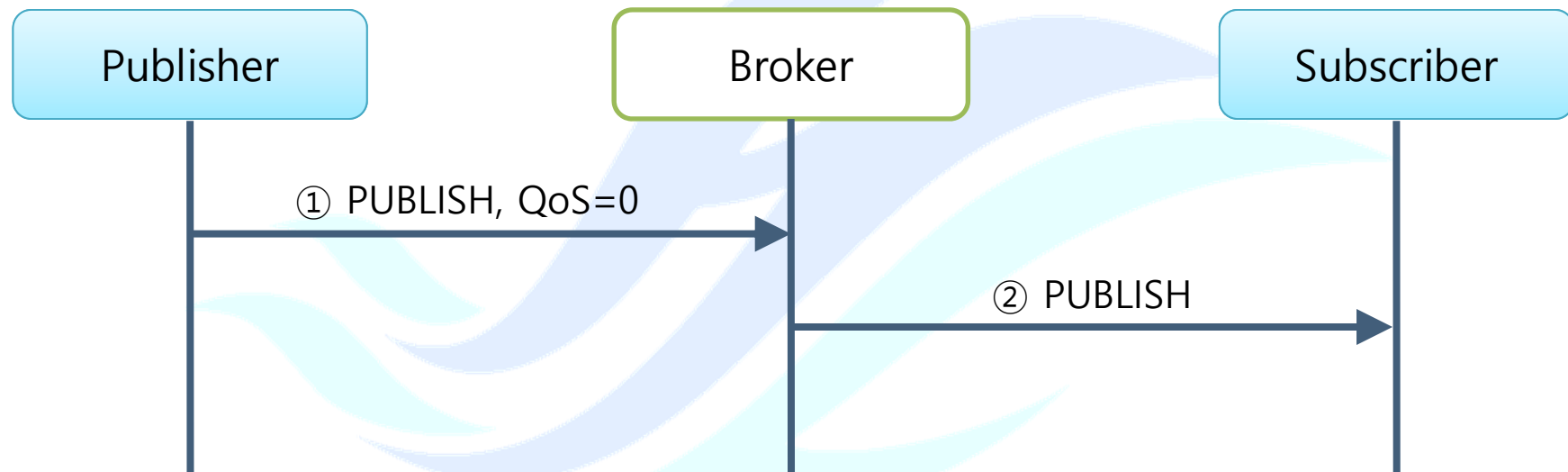
- 2-bit unsigned value
- 혼잡하고 제한적인 네트워크 환경에서 신뢰적인 메시지 전송을 보장하기 위해 사용
- 3단계 QoS 지원
 - QoS 0: 전송 후 응답을 받지 않음. 전송 보장하지 않음 (Fire and forget)
 - QoS 1: 최소 1회 전달. ACK 미수신시 재전송 (Acknowledged delivery)
 - 응답이 올 때까지 계속 전송하기 때문에, 중복 문제가 발생할 수 있음.
 - QoS 2: 정확히 1회 전달. Message ID 저장을 통한 중복 검사 (Assured delivery)
 - 제어 메시지를 여러 번 전송하기 때문에 오버헤드가 높음.
 - PUBLISH인 경우에만 사용

QoS value	bit 2	bit 1	Description
0	0	0	At most once Fire and Forget <=1
1	0	1	At least once Acknowledged delivery >=1
2	1	0	Exactly once Assured delivery =1
3	1	1	Reserved

Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-QoS Level

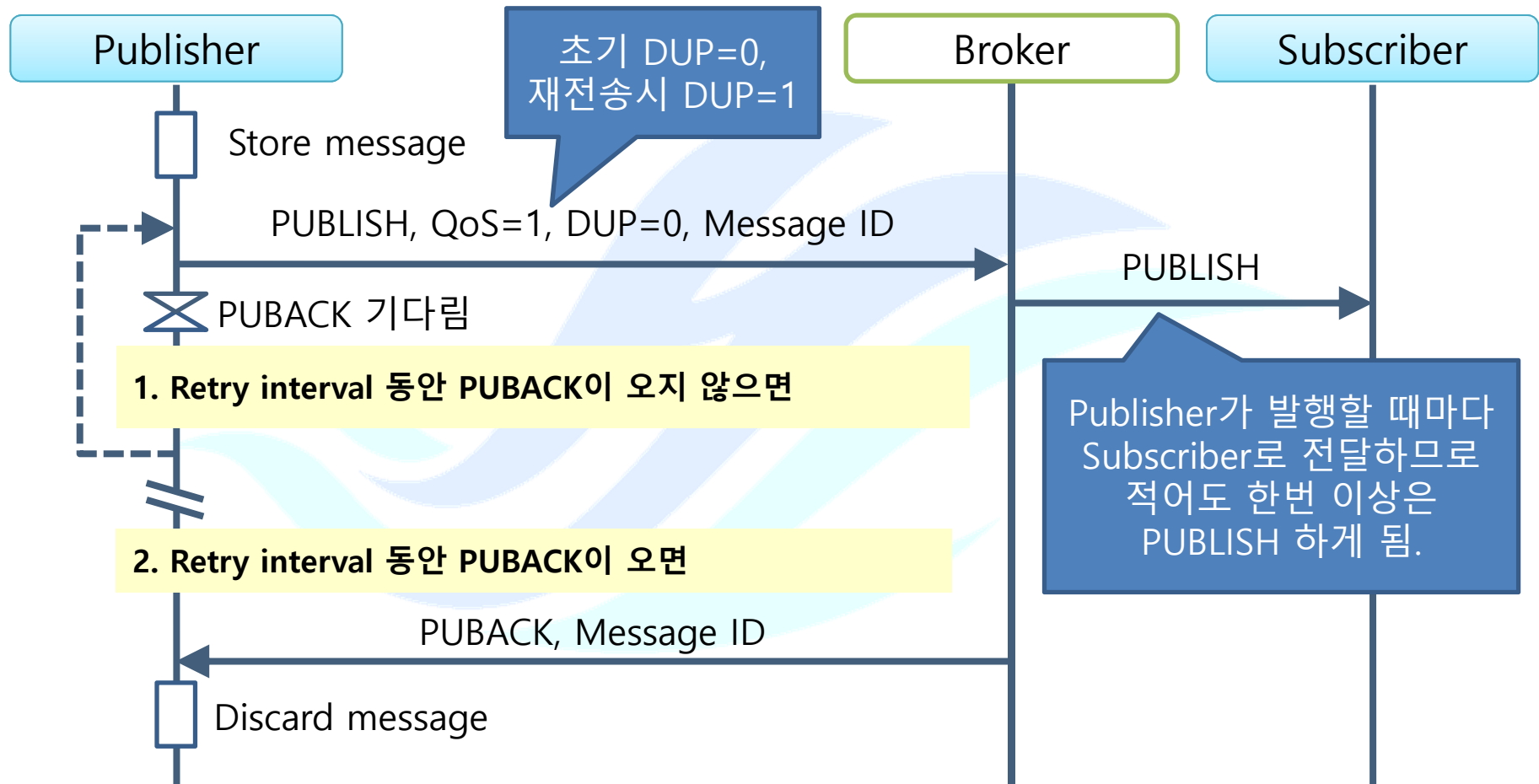
- Example of QoS 0
 - PUBLISH 예제



Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-QoS Level

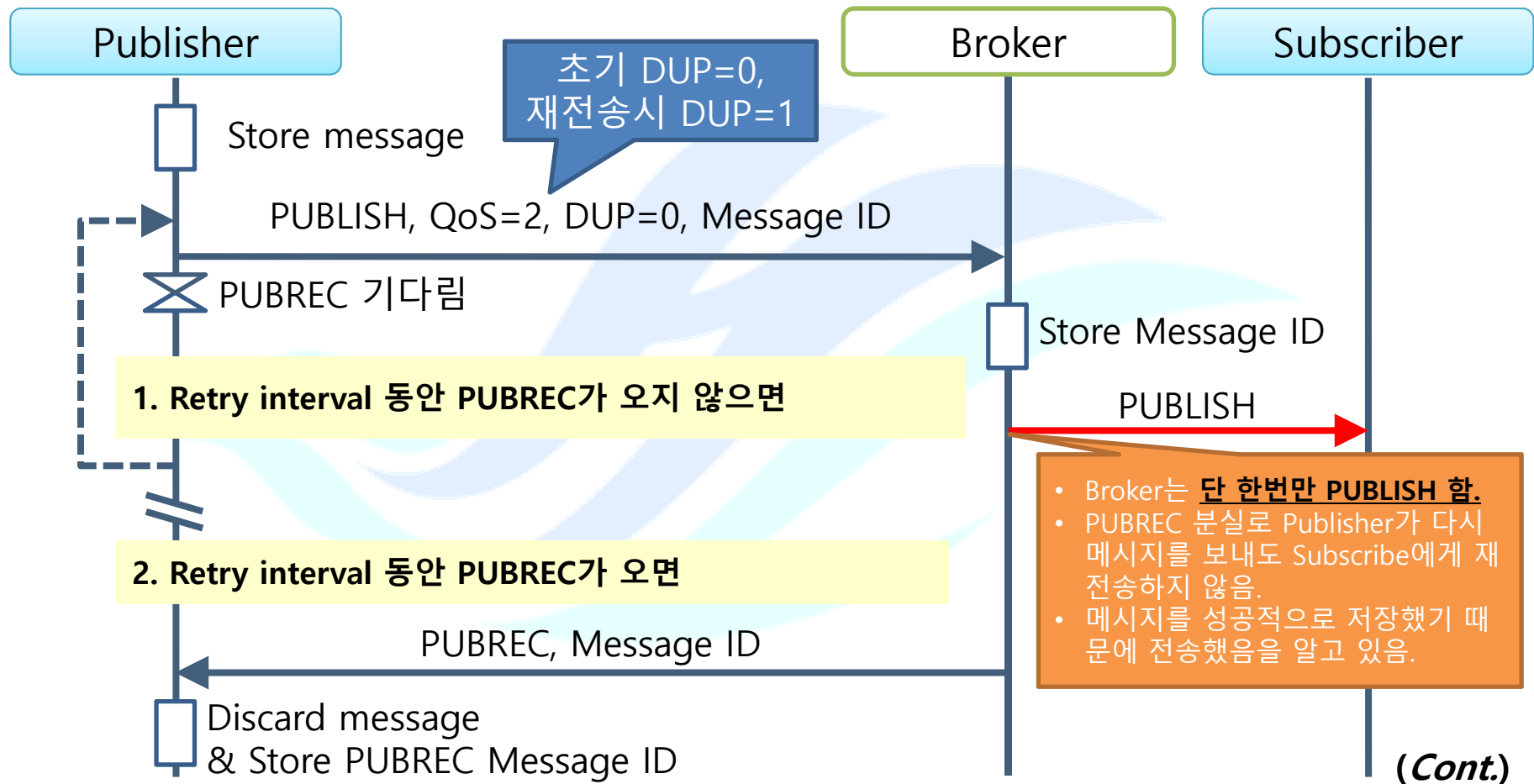
■ Example of QoS 1



Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 - 4 bytes)			

Fixed header-QoS Level

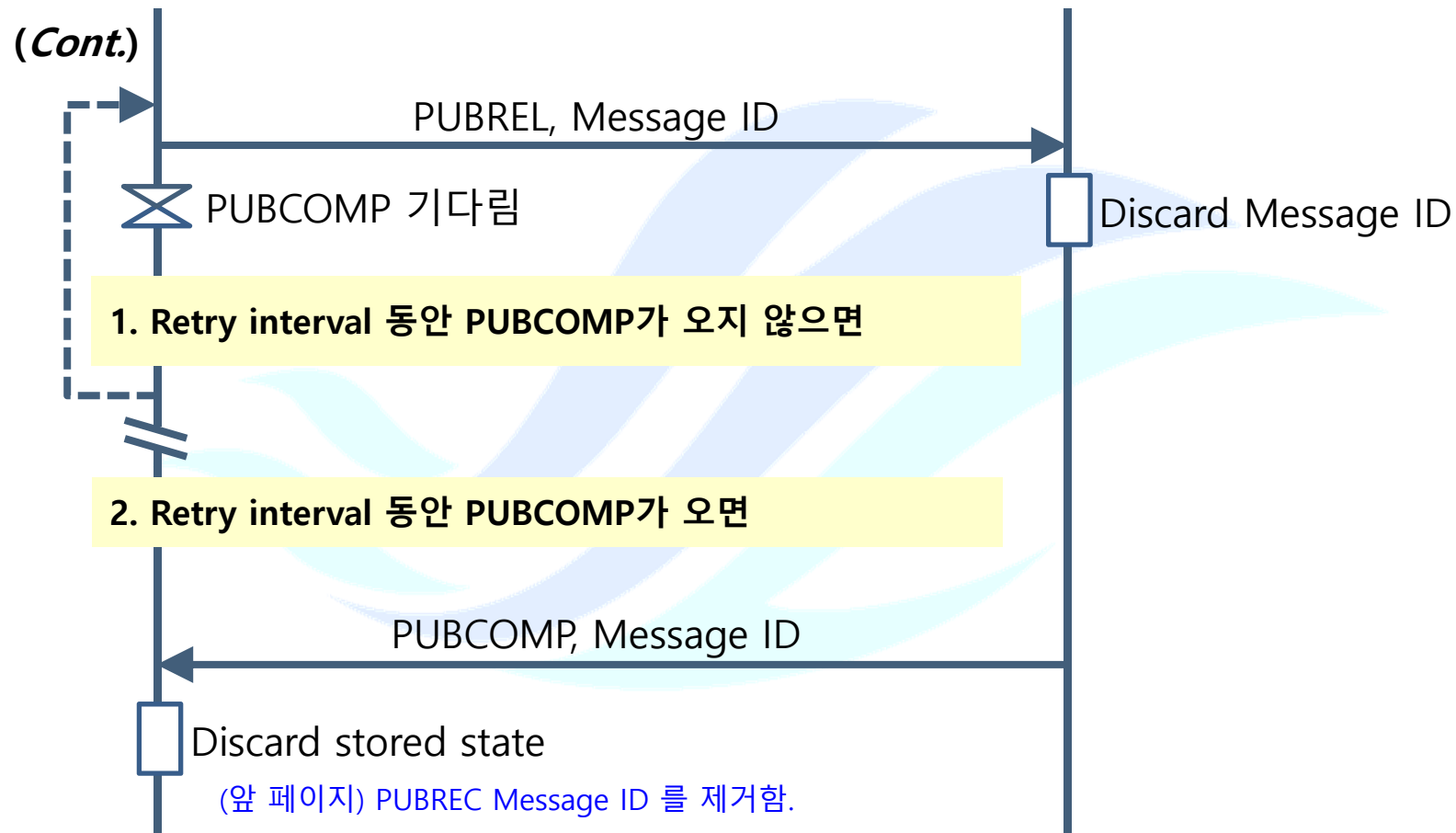
■ Example of QoS 2 (1/2)



Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-QoS Level

■ Example of QoS 2 (2/2)



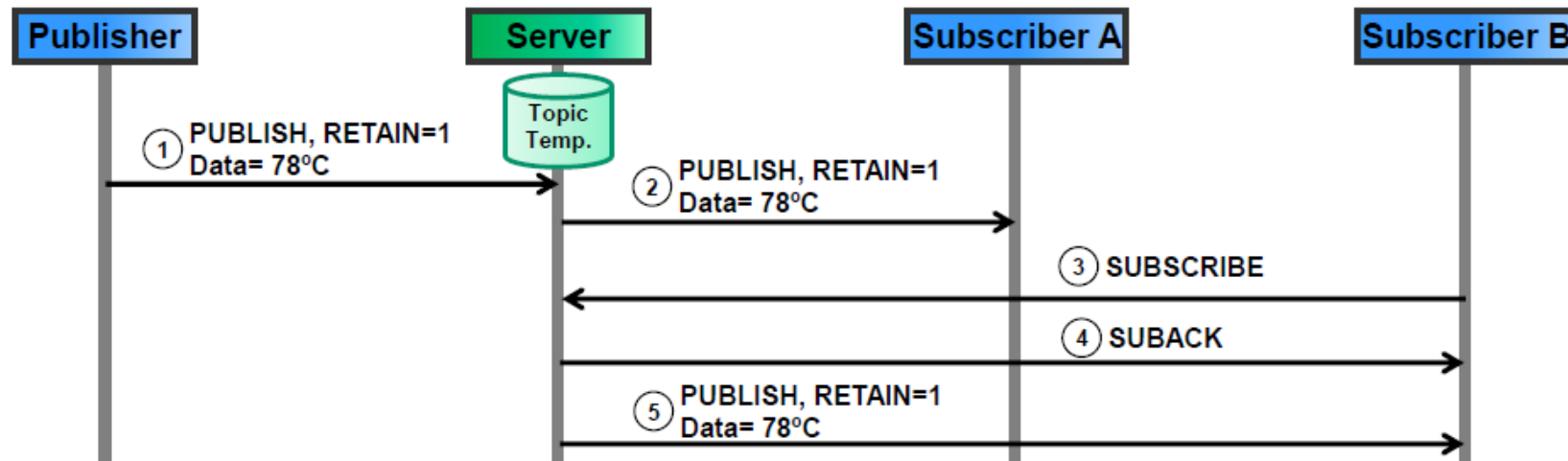
Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 – 4 bytes)			

Fixed header-RETAIN

■ RETAIN field

- 1-bit unsigned value
- Publisher가 보낸 가장 최근 메시지를 Broker가 유지하기 위해서 사용하는 필드
 - PUBLISH 메시지 타입에서만 사용함.
 - RETAIN=1 인 경우 Broker는 Publisher가 보낸 해당 최근 값을 유지함.
 - Subscriber가 신규로 구독요청을 하면 Broker는 자신이 유지하고 있는 값을 바로 보내줌.

■ RETAIN example



Fixed header

■ Flag bits of the fixed header

MQTT Control Packet	Fixed Header flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT v5.0	DUP	QoS		RETAIN
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0
AUTH	Reserved	0	0	0	0

