

MQTT Protocol #3



Kim, Eui-Jik

Contents

- Fixed header
 - Remaining Length
- Variable header & Payload
- Message flow
 - Session creation & termination
 - CONNECT
 - CONNACK
 - DISCONNECT

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

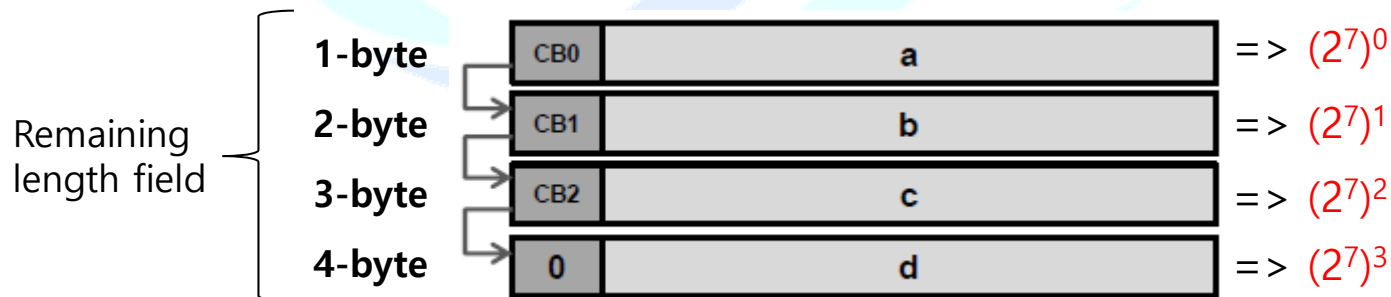
Fixed header-Remaining length

■ Remaining length (RL) field

- 8-bit unsigned value
 - Variable header 및 payload의 길이를 정의
 - 최대 256 Mbytes 까지 정의 가능
 - byte 단위로 RL의 길이를 표기 (즉, 해당 필드의 값이 1이면 1byte 길이 의미)
- 8bits로 256 Mbytes까지 정의하므로, 고유의 표현법을 사용
 - 이를 위해, 해당 필드의 길이는 **1byte ~ 4 bytes**로 가변

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

- $256 \text{ Mbytes} = 256 \times 2^{20} \text{ bytes} = 128 \times 2^{21} \text{ bytes} = 2^7 \times (2^7)^3 \text{ bytes} = 268,435,456 \text{ bytes}$



CB: Continuation Bit

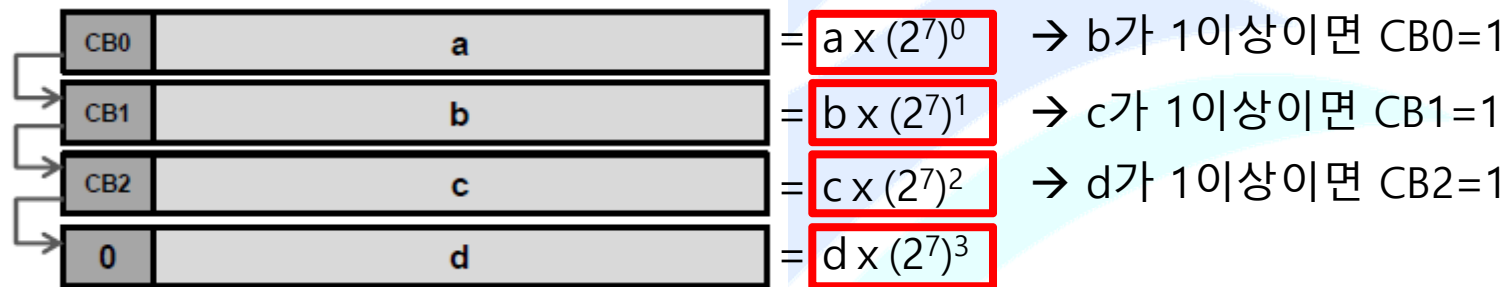
Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 - 4 bytes)			

Fixed header-Remaining length

■ Remaining length (RL) field

■ 길이 표기 및 계산법

- 해당 필드는 byte 순으로(아래→위) 표기, 실제 길이는 역순으로(위→아래) 각 단위가 커짐
- 즉, 실제 RL의 길이는 $a \times (2^7)^0 + b \times (2^7)^1 + c \times (2^7)^2 + d \times (2^7)^3$ 로 계산
 - a, b, c, d는 7비트로 표현되기 때문에 0 ~127 값을 가짐



■ CB: Continuation bit (1bit)로 다음 바이트의 사용이 필요할 때 1로 표기

- 즉, 128 byte 이상을 표기하고 싶을 때, 해당 필드의 값을 CB 값을 1로 정의
- 예를 들어, 130 bytes 길이를 정의하기 위해서는...
 - 2개 (2 bytes)의 RL field를 사용해야 하며, 첫 번째 byte의 CB (즉 CB0)의 값은 1로 표기 되고, 두 번째 byte의 CB(CB0)의 값은 0으로 표기함.
- 즉, (1000 0010) (0000 0001) → (0x82) (0x01) 로 표기 되어야 함.

Fixed header-Remaining length

■ Remaining length (RL) field

- RL 필드의 각 byte로 결정할 수 있는 최대/최소 길이

Digits	From (bytes)	To (bytes)
1	0 (0x00)	127 (0x7F)
2	128 (0x80, 0x01)	16,383 (0xFF, 0x7F)
3	16,384 (0x80, 0x80, 0x01)	2,097,151 (0xFF, 0xFF, 0x7F)
4	2,097,152 (0x80, 0x80, 0x80, 0x01)	268,435,455 (0xFF, 0xFF, 0xFF, 0x7F)

Fixed header-Remaining length

■ Remaining length (RL) field

■ Example 1: RL 필드로 364 bytes를 정의하고 싶은 경우

- $364 / 128 = 2.xxx \rightarrow CB0 = 1$, 두번째 byte 값 2 ($b=2$) (몫이 128보다 작으므로 $CB1=0$)
- $364 - (128 \times 2) = 108 \rightarrow$ 첫번째 byte의 값 ($a=108$)
- $364 = 108 \times (2^7)^0 + 2 \times (2^7)^1$

CB0=1	a=108
CB1=0	b=2
CB2=0	c=0
0	d=0

■ Example 2: RL 필드로 25,897 bytes를 정의하고 싶은 경우

- $25897 / 128 = 200.xxx \rightarrow CB0 = 1$ (몫이 128보다 크므로 한번 더 나누기)
- $200 / 128 = 1.xxx \rightarrow CB1 = 1$, 세번째 byte 값 1 ($c=1$) (몫이 128보다 작으므로 $CB2=0$)
- $25897 - 128^2 = 9513$
- $9513 / 128 = 74.xxx \rightarrow$ 두번째 byte 값 74 ($b=74$)
- $9513 - (128 \times 74) = 41 \rightarrow$ 첫번째 byte 값 41 ($a=41$)
- $25897 = 41 \times (2^7)^0 + 74 \times (2^7)^1 + 1 \times (2^7)^2$

CB0=1	a=41
CB1=1	b=74
CB2=0	c=1
0	d=0

Message Type	DUP	QoS Level	RETAIN
Remaining Length (1 - 4 bytes)			
Optional: Variable Length Header			
Optional: Variable Length Message Payload			

Variable header & Payload

■ Variable header

- Message type에 따라 서로 다른 길이 및 필드를 정의하며, 필요시 사용됨

■ Payload

- 정보(온도값, 토픽 등)를 담는 필드로, 필요시 사용됨

$\max(\text{Variable header length} + \text{Payload length}) = 256 \text{ Mbytes}$

Message Type = 1					-	-	-
Remaining Length							
Protocol name UTF-8 encoded (e.g. «Light_Protocol»), prefixed with 2 bytes string length (MSB first)							
Protocol version (value 0x03 for MQTT version 3)							
Username Flag	Password Flag	Will Retain	Will QoS	Will Flag	Clean Session	Reserved	
Keep Alive Timer MSB							
Keep Alive Timer LSB							
Client Identifier							
Will Topic							
Will Message							
Username							
Password							

CONNECT

Message Type = 3	DUP	QoS Level	RETAIN
Remaining Length			
Topic Name String Length (MSB)			
Topic Name String Length (LSB)			
Topic Name			
Message ID (MSB)			
Message ID (LSB)			
Publish Message			

PUBLISH

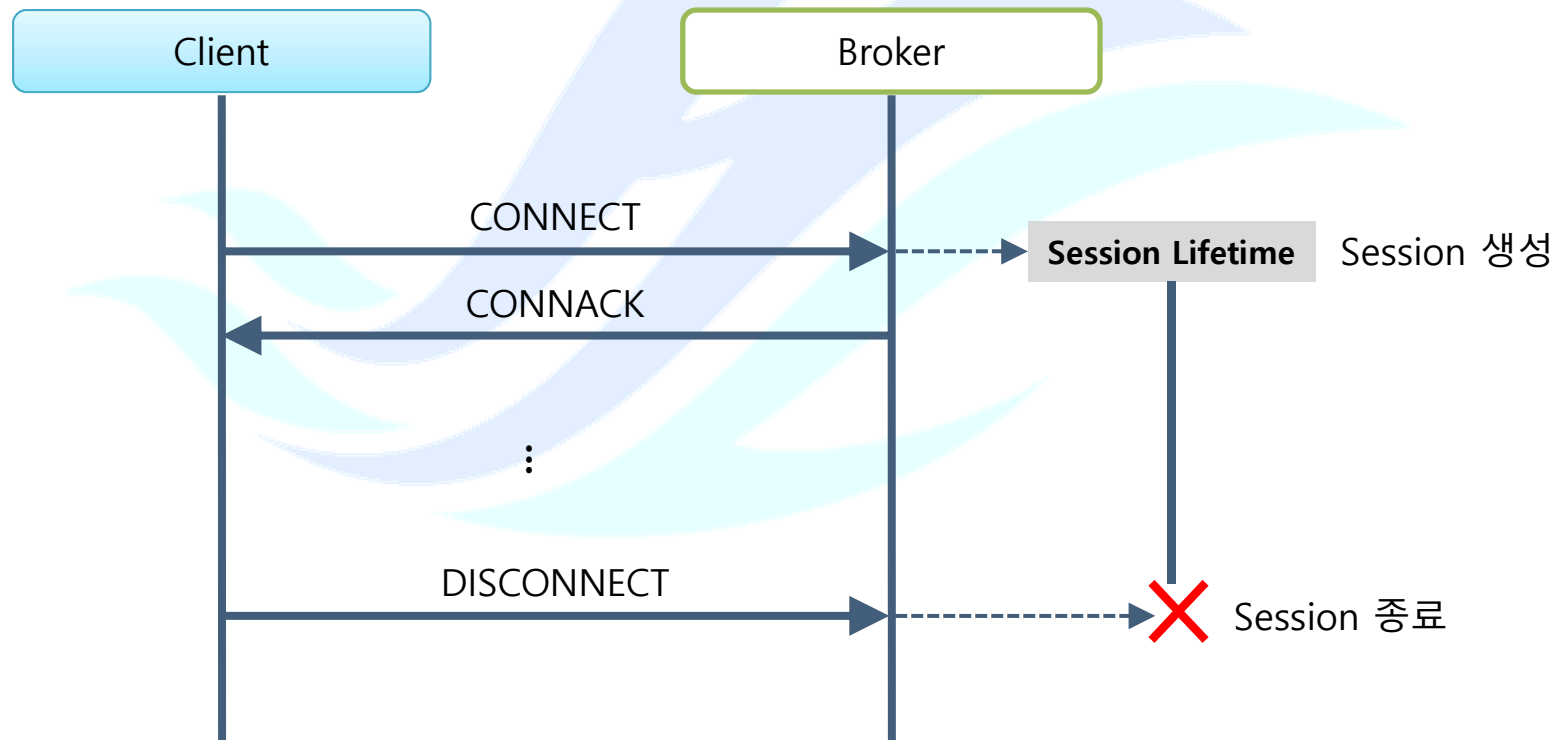
Message Type = 8	DUP	QoS Level	-
Remaining Length			
Message ID (MSB)			
Message ID (LSB)			
Topic Name String Length (MSB)			
Topic Name String Length (LSB)			
Topic Name			
Reserved (not used)		QoS Level	

SUBSCRIBE

Message flow

■ Session creation & termination

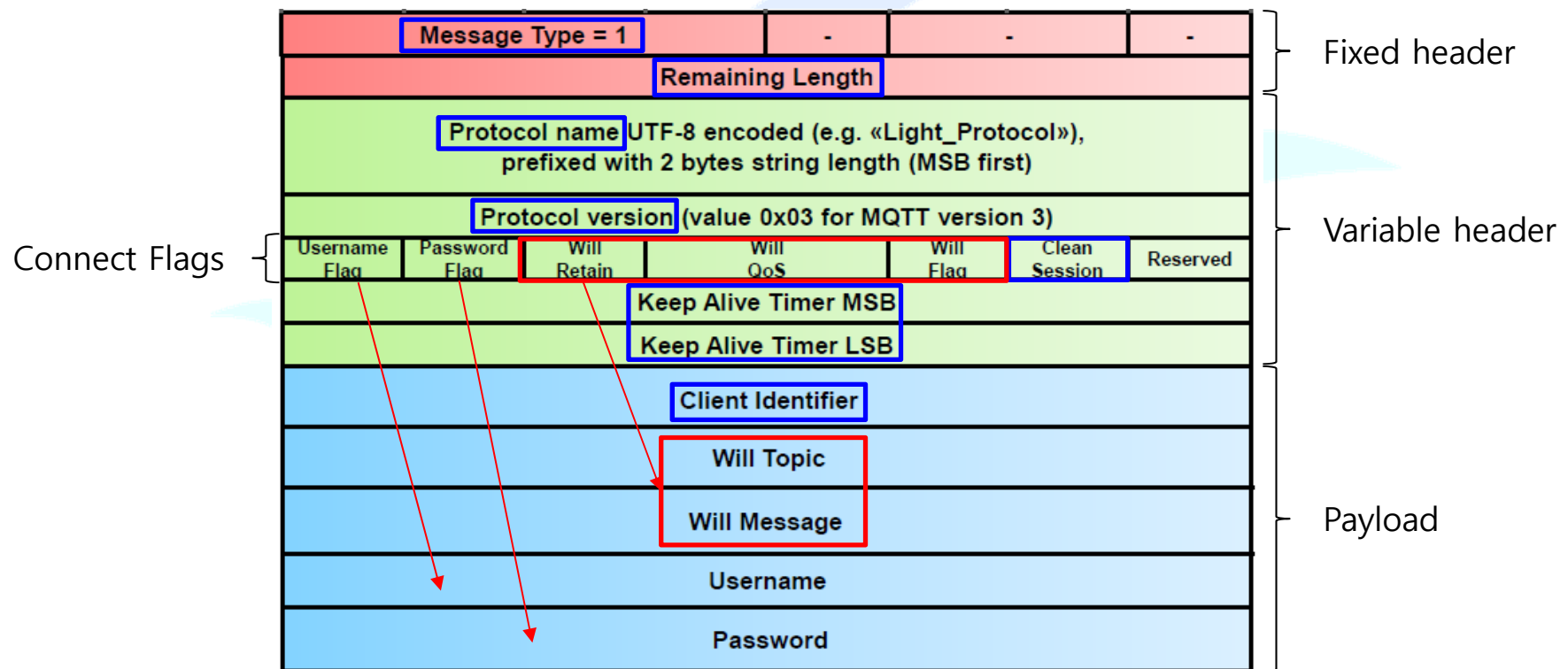
- Client는 CONNECT 메시지로 Session 생성을 요청함.
- Broker는 CONNACK 메시지로 요청에 대해 응답함.
- Client는 DISCONNECT 메시지로 Session을 종료함.



Message flow

■ CONNECT message

- Client는 CONNECT message로 Session 생성을 요청함.
- CONNECT 메시지에는 다음과 같은 필드를 포함함.



Message flow

■ Variable header in CONNECT message

■ Protocol name

- 프로토콜의 이름을 정의하며, 2bytes로 표현되는 프로토콜 이름의 길이와 UTF-8로 인코딩된 문자열로 구성됨. (문자열의 숫자값은 미국정보교환표준부호 (ASCII) 값임)

■ Protocol version

- 1byte로 표현되며, MQTT version 3은 0x03, MQTT version 3.1.1은 0x04, MQTT version 5는 0x05를 사용함.

[Example]

Protocol Name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (4)	0	0	0	0	0	1	0	0
byte 3	'M'	0	1	0	0	1	1	0	1
byte 4	'Q'	0	1	0	1	0	0	0	1
byte 5	'T'	0	1	0	1	0	1	0	0
byte 6	'T'	0	1	0	1	0	1	0	0
Protocol Version									
byte 7	Version (5)	0	0	0	0	0	1	0	1

[ASCII]

모양	이진법	십진법	십육진법	모양	이진법	십진법	십육진법
A	1000001	65	41	N	1001110	78	4E
B	1000010	66	42	O	1001111	79	4F
C	1000011	67	43	P	1010000	80	50
D	1000100	68	44	Q	1010001	81	51
E	1000101	69	45	R	1010010	82	52
F	1000110	70	46	S	1010011	83	53
G	1000111	71	47	T	1010100	84	54
H	1001000	72	48	U	1010101	85	55
I	1001001	73	49	V	1010110	86	56
J	1001010	74	4A	W	1010111	87	57
K	1001011	75	4B	X	1011000	88	58
L	1001100	76	4C	Y	1011001	89	59
M	1001101	77	4D	Z	1011010	90	5A

Message flow

■ Variable header in CONNECT message

■ Connect Flags

- 1byte로 구성되며 다음의 필드를 포함함.

* Will message는 추후 설명

필드명	필드 크기	내용
User name flag	1bit	Payload에 사용자 이름을 포함시킬 때 1로 정의
Password flag	1bit	Payload에 사용자 Password을 포함시킬 때 1로 정의
Will retain	1bit	Will message를 사용하는 경우, Will message의 retain여부를 정의할 때 사용하며, 값이 1일 때 broker에 유지
Will QoS	2bit	Will message를 사용하는 경우, Will message의 QoS를 정의할 때 사용하며, 1~3까지 정의
Will flag	1bit	Will message의 사용여부를 표기할 때 사용
Clean session	1bit	새로운 Session을 생성할 때 해당 필드를 1로 정의, 이전 세션의 구독정보를 유지하고 싶을 때는 해당 필드를 0으로 정의
Reserved	1bit	사용하지 않는 비트, 0 으로 고정

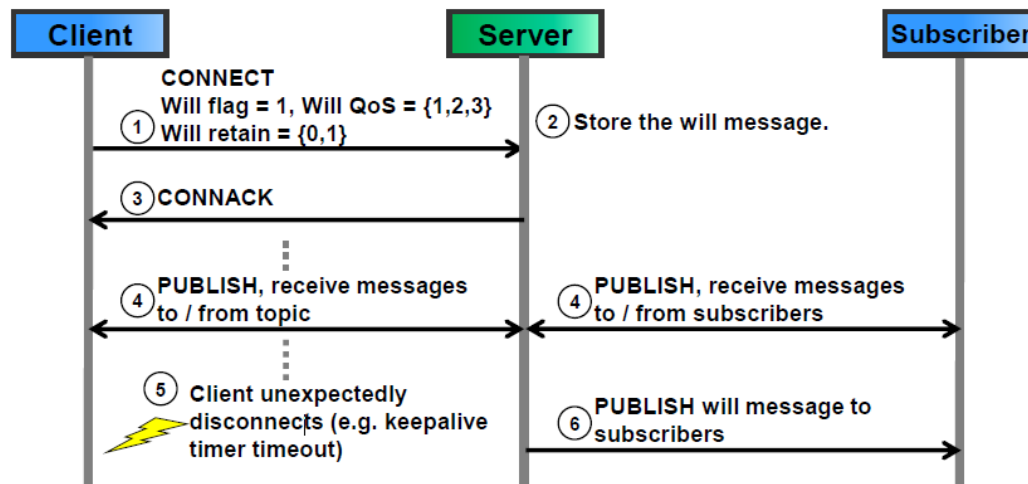
[Example]

Connect Flags									
byte 8	User Name Flag (1)								
	Password Flag (1)								
	Will Retain (0)								
	Will QoS (01)	1	1	0	0	1	1	1	0
	Will Flag (1)								
	Clean Start(1)								
	Reserved (0)								

Message flow

■ Will in CONNECT message

- Will은 '유언'이라는 의미로 사용
- 특정 Publisher의 Topic을 구독하고 있는 Subscriber는 Publisher와 Broker간 연결 상태를 알 수 없음
- Publisher와 Broker가 예기치 않게 연결이 종료 되었을 경우, Publisher로부터 전달받은 Will 정보를 활용하여, Broker는 Will message를 Topic을 구독하고 있는 모든 Subscriber에게 전송하여 연결이 끊어졌다고 알림
 - 이를 Last will and testament 라고도 함. '유언을 남긴다' 라고 해석



발행토픽: HQ/mattdamon
발행메시지: I'M PRETTY MUCH FUCKED.
월토픽: HQ/die
월메시지: will-message "matt damon"

[결과 텍스트]
HQ/mattdamon I'M PRETTY MUCH
FUCKED.

(연결 끊어짐)

HQ/die matt damon

Message flow

■ Variable header in CONNECT message

■ Keep alive timer

- Client와 Broker간 Session(Connection)이 살아 있는지 여부를 판단하기 위해 사용하는 Timer이며, Broker가 연결상태를 확인하기 위해 Timer를 유지
- 2 byte로 표현하며, 해당 필드의 1은 1초를 의미

[Example]

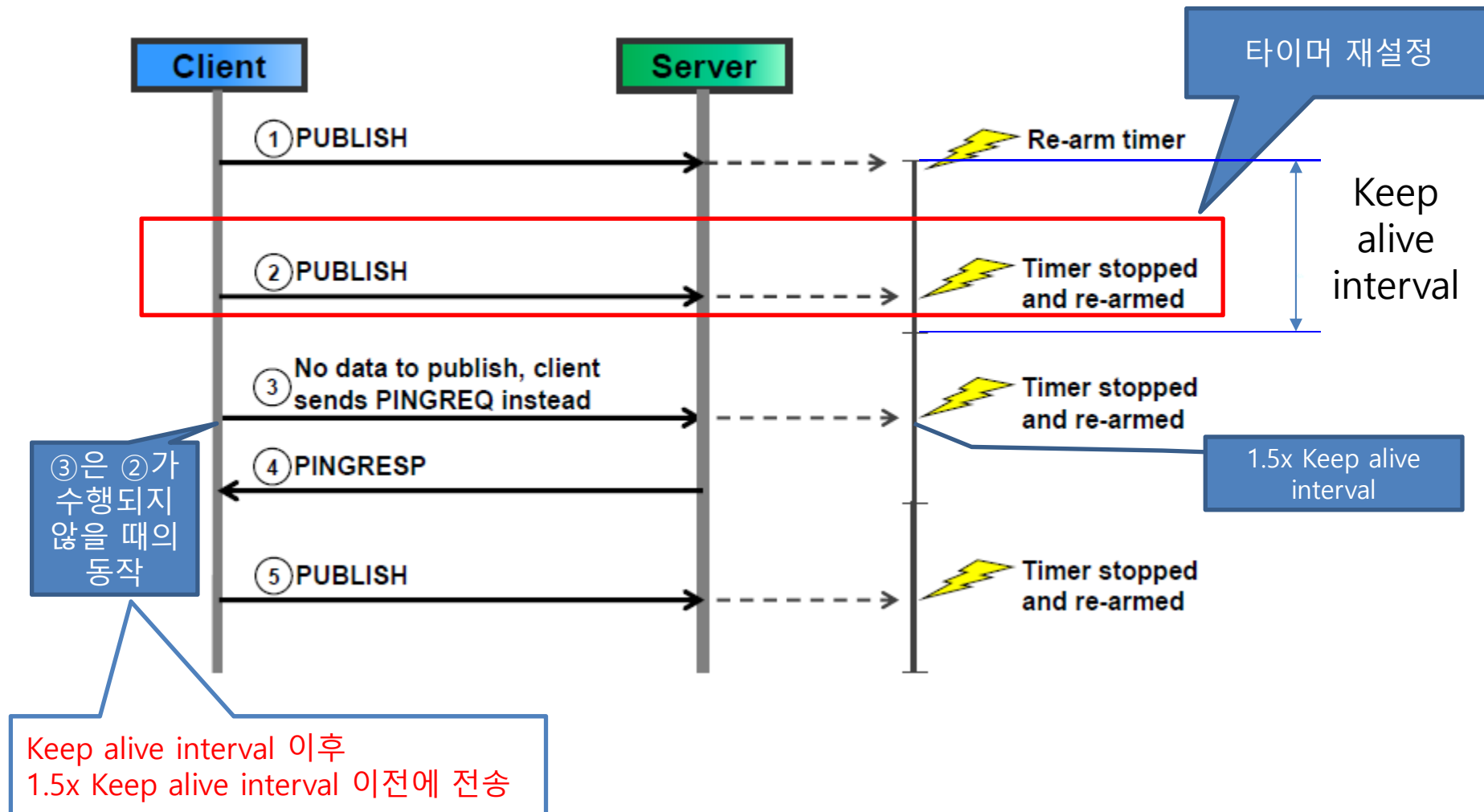
Keep Alive Timer									
byte 9	Keep Alive Timer MSB (0)	0	0	0	0	0	0	0	0
byte 10	Keep Alive Timer LSB (10)	0	0	0	0	1	0	1	0

■ 동작

- Broker는 Keep Alive Timer에 정의된 시간의 1.5배 시간만큼 Client의 Published Message를 기다리고, Broker는 **1.5 x Keep Alive Interval** 시간이 지났을 때, 어떠한 Message도 받지 못하면 **둘간의 연결이 끊겼다고 판단**
- 새로운 Published Message를 받을 때마다 해당시간은 초기화 되고, Client는 Publish할 Message가 없을 때 Keep alive interval 이후 1.5x Keep alive interval 이전에 PINGREQ 메시지를 전송하여 연결이 살아 있다는 것을 Broker에게 알리고, Broker는 이에 대한 회신으로 PINGRESP를 회신

Message flow

■ Example of keep alive timer



Message flow

■ Payload in CONNECT message

■ Client Identifier (ClientID)

- Client의 ID를 Broker에게 알려주기 위해 사용
- 1~23 bytes로 정의, UTF-8로 인코딩된 문자열로 구성(ASCII)
- 숫자 및 알파벳으로만 구성 가능

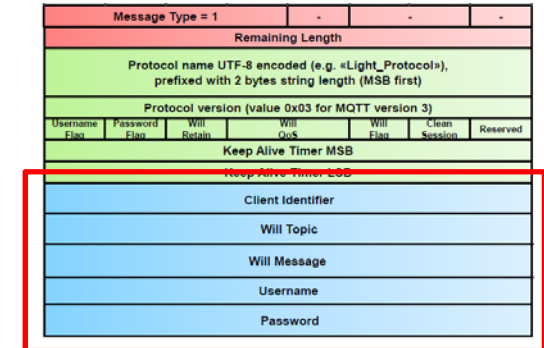
"0123456789abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ"

■ Will Topic, Will Message (Will Payload)

- Variable header의 Will Flag 값이 1일 때 사용
- 각각 Will topic 및 message 내용을 포함
- UTF-8로 인코딩된 문자열로 구성(ASCII)

■ Username & Password

- Variable header의 Username Flag 및 Password Flag 값이 1일 때 사용
- 허가된 사용자에게만 서비스 제공을 하기 위해, 사용자 이름 및 토큰 값 등을 정의하기 위해 사용
- UTF-8로 인코딩된 문자열로 구성(ASCII)



Message flow

■ Example for CONNECT message size

	Description	Hex value
CONNECT message fixed header		
byte 1	Control packet type	0x10
byte 2	Remaining length	0x10
CONNECT message variable header		
Protocol name		
byte 1	Length MSB (0)	0x00
byte 2	Length LSB (4)	0x04
byte 3	(M)	0x4D
byte 4	(Q)	0x51
byte 5	(T)	0x54
byte 6	(T)	0x54
Protocol level		
byte 7	Level (4)	0x04
CONNECT Flags		
byte 8	CONNECT flags byte	0x02
Keep alive		
byte 9	Keep Alive MSB (0)	0x00
byte 10	Keep Alive LSB (60)	0x3C
CONNECT message Payload		
Client ID		
byte 1	Length MSB (0)	0x00
byte 2	Length LSB (4)	0x04
byte 3	(D)	0x44
byte 4	(I)	0x49
byte 5	(G)	0x47
byte 6	(I)	0x49

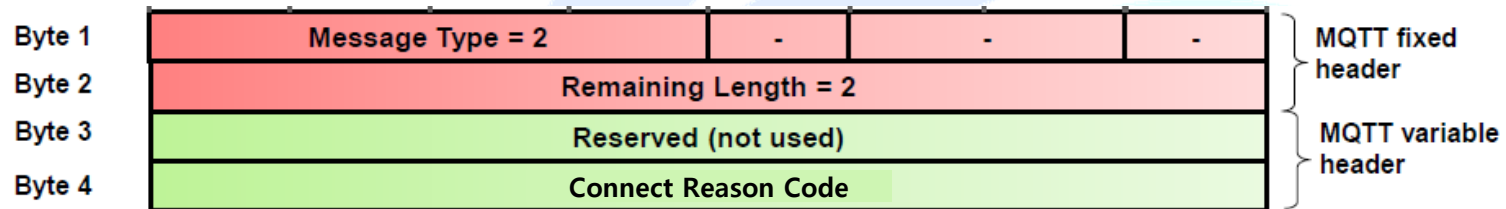
CONNECT message size
 = Fixed header + Variable header + Payload
 = 2 bytes + 10 bytes + 6 bytes
 = 18 bytes

CONNECT Flag Bit(s)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
User name flag	0							
Password flag		0						
Will retain			0					
Will QoS				0	0			
Will flag						0		
Clean session							1	
Reserved								0

Message flow

■ CONNACK message

- Broker는 CONNECT message에 대한 응답으로 CONNACK를 전송함.
- Payload는 포함하지 않으며, Variable header에 Connect Reason Code를 포함하여 전송함.



Message flow

■ Connect Reason Code in CONNACK message

Value	Hex	Reason Code name	Description
0	0x00	Success	The Connection is accepted.
128	0x80	Unspecified error	The Server does not wish to reveal the reason for the failure, or none of the other Reason Codes apply.
129	0x81	Malformed Packet	Data within the CONNECT packet could not be correctly parsed.
130	0x82	Protocol Error	Data in the CONNECT packet does not conform to this specification.
131	0x83	Implementation specific error	The CONNECT is valid but is not accepted by this Server.
132	0x84	Unsupported Protocol Version	The Server does not support the version of the MQTT protocol requested by the Client.
133	0x85	Client Identifier not valid	The Client Identifier is a valid string but is not allowed by the Server.
134	0x86	Bad User Name or Password	The Server does not accept the User Name or Password specified by the Client
135	0x87	Not authorized	The Client is not authorized to connect.
136	0x88	Server unavailable	The MQTT Server is not available.
137	0x89	Server busy	The Server is busy. Try again later.
138	0x8A	Banned	This Client has been banned by administrative action. Contact the server administrator.
140	0x8C	Bad authentication method	The authentication method is not supported or does not match the authentication method currently in use.
144	0x90	Topic Name invalid	The Will Topic Name is not malformed, but is not accepted by this Server.
149	0x95	Packet too large	The CONNECT packet exceeded the maximum permissible size.
151	0x97	Quota exceeded	An implementation or administrative imposed limit has been exceeded.
153	0x99	Payload format invalid	The Will Payload does not match the specified Payload Format Indicator.
154	0x9A	Retain not supported	The Server does not support retained messages, and Will Retain was set to 1.
155	0x9B	QoS not supported	The Server does not support the QoS set in Will QoS.
156	0x9C	Use another server	The Client should temporarily use another server.
157	0x9D	Server moved	The Client should permanently use another server.
159	0x9F	Connection rate exceeded	The connection rate limit has been exceeded.

Message flow

■ DISCONNECT, PINGREQ, PINGRESP messages

- 모두 Fixed header만을 전송
- 연결 해제 시 Message Type = 14 (DISCONNECT)
- Ping 요청 시 Message Type = 12 (PINGREQ)
- Ping 요청응답 시 Message Type = 13 (PINGRESP)

