



Đặc tả giao thức MQTT V3.1

Tác giả:

International Business Machines Corporation (IBM)
Eurotech

Khái quát chung nhất

MQ Telemetry Transport (MQTT) (tạm dịch là Giao vận tầm xa) là giao thức truyền message theo mô hình cung cấp/thuê bao (publish/subscribe). Nó dựa trên một Broker (điểm trung gian) "nhẹ" (khá ít xử lý), và được thiết kế có tính mở (tức là không đăng trưng cho ứng dụng cụ thể nào), đơn giản, "nhẹ", và dễ implement. Những đặc trưng này khiến MQTT rất lý tưởng để sử dụng trong các môi trường bị giới hạn tài nguyên tính toán và truyền dữ liệu như:

- Những nơi mà giá network đắt hoặc băng thông thấp, hay thiếu tin cậy
- Khi chạy trên một thiết bị nhúng bị giới hạn về tài nguyên tốc độ và bộ nhớ

Các đặc trưng của giao thức bao gồm:

- Dạng truyền message theo mô hình cung cấp/thuê bao (publish/subscribe) cung cấp việc truyền tin phân tán 1-nhiều, tách biệt với phần ứng dụng (trong mô hình này, thì người gửi message là publisher, còn người nhận là subscriber)
- Việc truyền message là luôn không quan tâm đến nội dung được truyền
- Sử dụng TCP/IP là giao thức nền
- Có 3 cấp độ của QoS (Qualities of Service : chất lượng truyền tin) được đưa ra:
 - "Hầu như chỉ 1 lần", message được truyền nhận dựa hoàn toàn vào tính tin cậy của TCP/IP. Việc mất hoặc lặp message có thể xảy ra. Ở QoS này, có thể ví dụ 1 trường hợp sử dụng: như trong môi trường sensor mà việc mất 1 gói dữ liệu (message) tại 1 thời điểm không ảnh hưởng đến toàn bộ quá trình sử dụng dữ liệu của sensor đó.
 - "Ít nhất 1 lần", các message được đảm bảo được nhận (bởi subscriber) nhưng có thể xảy ra lặp.
 - "Chính xác chỉ 1 lần", message được đảm bảo đến nơi đúng 1 lần. Cấp độ này được ứng dụng ở các hệ thống thanh toán, nơi mà việc lặp hay mất message có thể gây ra việc tính tiền bị sai hay những hậu quả nghiêm trọng khác.
- Phần bao bọc dữ liệu truyền (overhead) nhỏ (độ dài cố định luôn là 2 byte), và được giảm đến mức tối thiểu để giảm tải cho đường truyền.
- Một cơ chế để thông báo đến các thuê bao (subscriber) khi đường truyền bị đứt bất thường, đó là sử dụng khả năng thông báo Last Will (thông điệp cuối) và Testament feature.

Copyright Notice

© 1999-2010 Eurotech, International Business Machines Corporation (IBM). All rights reserved.

Permission to copy and display the MQ Telemetry Transport specification (the "Specification"), in any medium without fee or royalty is hereby granted by Eurotech and International Business Machines Corporation (IBM) (collectively, the "Authors"), provided that you include the following on ALL copies of the Specification, or portions thereof, that you make:

1. A link or URL to the Specification at one of the Authors' websites.
2. The copyright notice as shown in the Specification.

The Authors each agree to grant you a royalty-free license, under reasonable, non-discriminatory terms and conditions to their respective patents that they deem necessary to implement the Specification. THE SPECIFICATION IS PROVIDED "AS IS," AND THE AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE SPECIFICATION.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the Specification or its contents without specific, written prior permission. Title to copyright in the Specification will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.

Mục lục

1. Giới thiệu

Tài liệu đặc tả này được chia làm 3 chương chính:

- Định dạng message chung cho tất cả các gói dữ liệu (packet hay gói tin) sử dụng trong giao thức
- Chi tiết mỗi loại packet
- Luồng packet giữa client và server

Thông tin về cách mô tả topic (chuỗi kí tự đại diện cho một nhóm message) được cung cấp ở phần appendix.

1.1. Thay đổi mới so với phiên bản trước

Sự thay đổi từ MQTT V3 đến MQTT V3.1:

- User và password giờ có thể được gửi qua một gói tin CONNECT
- Thêm mã trả về cho packet CONNACK để sử dụng cho mục đích security
- Client sẽ không được cho biết về việc các gói chứa command PUBLISH hoặc SUBSCRIBE không được xác thực. Một luồng MQTT thông thường sẽ kết thúc thậm chí những command này không được thực hiện.
- String trong MQTT bây giờ hỗ trợ cả UTF8 thay vì chỉ tập nhỏ ASCII.

Số version của protocol sẽ có trong gói CONNECT, và trong phiên bản V3r1 này sẽ là 3. Những thực thi MQTT V3 có thể chấp nhận kết nối đến từ version này với độ dài mong muốn và bỏ qua các thông tin security khác.

2. Định dạng của message

Message của mỗi MQTT command đều chứa một phần gọi là header cố định. Một số message cũng chứa thêm phần header có độ dài thay đổi được rồi cả dữ liệu kèm theo nữa. Định dạng của mỗi message header được miêu tả trong các section sau:

2.1. Phần header cố định

Tất cả các message luôn chứa phần cố định dưới đây:

bit	7	6	5	4	3	2	1	0
byte 1	Loại Message					Cờ	QoS level	
byte 2	Độ dài còn lại							

Byte 1

Chứa loại Message và các cờ (DUP, QoS level, and RETAIN).

Byte 2

Quy định độ dài còn lại (Ít nhất 1 tức là 1 byte)

Trường này sẽ được nói đến kĩ hơn. Tất cả dữ liệu là big-endian order: Byte cao sẽ đứng trước byte thấp. Một word 16 bit sẽ được truyền trên dây thành 1 byte Most Significant (MSB), theo là byte Least Significant(LSB).

Loại Message

Vị trí: byte 1, bits 7-4.

Một số 4-bit không dấu diễn tả các giá trị được miêu tả dưới bảng sau:

Từ gợi nhớ	Giá trị thứ tự	Miêu tả
Reserved	0	Chưa dùng
CONNECT	1	Client yêu cầu kết nối đến Server
CONNACK	2	Kết nối được chấp nhận
PUBLISH	3	Xuất bản message
PUBACK	4	Xuất bản message được chấp nhận
PUBREC	5	Xuất bản đã được nhận (đảm bảo nhận được part 1)

Từ gọi nhở	Giá trị thứ tự	Miêu tả
PUBREL	6	Xuất bản release (đảm bảo nhận được part 2)
PUBCOMP	7	Xuất bản hoàn thành (đảm bảo nhận được part 3)
SUBSCRIBE	8	Yêu cầu subscribe từ client
SUBACK	9	Yêu cầu subscriber được chấp nhận
UNSUBSCRIBE	10	Yêu cầu unsubscribe
UNSUBACK	11	Yêu cầu unsubscribe được chấp nhận
PINGREQ	12	Request PING
PINGRESP	13	Response PING
DISCONNECT	14	Client đang mất kết nối
Reserved	15	Reserved

Các cờ

Các bit còn lại của byte đầu tiên tương ứng sẽ mô tả trạng thái của DUP, QoS và RETAIN. Vị trí các bit và ý nghĩa được miêu tả trong bảng dưới đây.

Vị trí bit	Tên viết gọn	Miêu tả
3	DUP	Truyền lặp lại
2-1	QoS	Quality of Service (mức độ tin cậy)
0	RETAIN	cờ RETAIN

DUP

Vị trí: byte 1, bit 3.

Cờ này được bật khi client hoặc server đang cố chuyển lại một gói PUBLISH, PUBREL, SUBSCRIBE or UNSUBSCRIBE. Giá trị này được sử dụng trong các message mà có QoS lớn hơn 0 và yêu cầu ACK. Khi bit DUP được set, phần header thay đổi sẽ là Message ID.

Phía nhận xem giá trị này là một gợi ý để tự kiểm tra xem gói tin (có Message ID) ở trên đã chuyển đến trước đó hay không. Còn chỉ từ trạng thái cờ DUP này thì chưa thể chắc chắn có gói tin lặp được.

QoS

Vị trí: byte 1, bits 2-1.

Cờ này sẽ cho biết mức độ tin cậy của việc chuyển message PUBLISH. Giá trị của QoS được miêu tả trong bảng dưới đây.

Giá trị QoS	bit 2	bit 1	Miêu tả
-------------	-------	-------	---------

Giá trị QoS	bit 2	bit 1	Miêu tả		
0	0	0	Cùng lăm là 1 lần	Gửi rồi quên ngay	≤ 1
1	0	1	Ít nhất 1 lần	Xác nhận bằng ACK	≥ 1
2	1	0	Chính xác 1 lần	Nhận đảm bảo	$= 1$
3	1	1	Chưa dùng		

RETAIN

Vị trí: byte 1, bit 0.

Cờ này chỉ được sử dụng ở message PUBLISH. Khi client gửi 1 message PUBLISH đến server, nếu cờ Retain được set (1), thì server phải hiểu rằng phải giữ message này sau khi chuyển nó đến các subscribers hiện tại.

Khi có 1 subscription (yêu cầu nhận tin từ subscriber) mới được thiết lập đến 1 topic, message cuối cùng được lưu lại của topic đó phải được gửi đến subscriber và trường Retain được set (1) trong phần header. Nếu không có message nào của topic được lưu giữ, thì không cần gửi gì hết.

Trường này sẽ có ích khi publisher gửi message để báo "report bằng ngoại lệ" , thỉnh thoảng là nơi giữa các message. Điều này cho phép những subscribers mới nhanh chóng nhận dữ cần thiết, hoặc giá trị kiểu Méo mó có hơn không.

Trường hợp mà server chuyển tiếp nội dung vừa nhận được từ một Publisher thì trường Retain sẽ không được set. Điều này sẽ giúp phân biệt được message có từ trước với message mới được publish lên.

Message Retained sẽ được giữ thậm chí sau khi restart lại server

Server sẽ xóa message được retained nếu nó nhận được một message với payload bằng zero.

Độ dài còn lại

Vị trí: byte 2.

Miêu tả độ dài bao gồm cả phần header và payload có trong message

Việc encoding với độ dài thay đổi sử dụng 1 byte để miêu tả độ dài, vì thế độ dài tối đa sẽ là 127. Những message dài hơn sẽ được miêu tả theo cách sau. 7 bit được dùng để miêu tả giá trị, bít còn lại dùng để miêu tả phía sau còn byte nào miêu tả trường này hay không. Mỗi byte tiếp sau đó cũng như vậy 7 bit để lưu giá trị, 1 bít gọi là bit tiếp tục. Giá trị được tính bằng cách nhân giá trị được diên tả bởi 7 bit và lũy thừa tăng dần của 128. Ví dụ miêu tả độ Remain Length = 64, ta chỉ cần 1 byte, trong đó 7 bytes để miêu tả giá trị 64, 1 bit còn lại bằng 0. Một ví dụ nữa, giá trị là 321 chẵng hạn $321 = 65 * 128^0 + 2 * 128^1$, ta cần 2 byte để biểu diễn. Byte đầu chứa giá trị 65 trong 7 bit và bit còn lại là 1. Byte thứ 2 chứa giá trị 2 ở 7 bit và 1 bit chứa giá trị bằng 0.

Trường này được biểu diễn tối đa trong 4 byte. Tức là cho độ dài cho phép sẽ là đến 268 435 455 (256 MB)

Bảng sau miêu tả độ dài dài có thể biểu diễn với các số byte tương ứng.

Số byte	Độ dài min	Độ dài max
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)

Thuật toán để chuyển giá trị ở hệ cơ số 10 sang dạng encode này được miêu tả bằng mã giả sau:

```

do
    digit = X MOD 128
    X = X DIV 128
    // Nếu cần nhiều byte hơn để encoding
    if ( X > 0 )
        digit = digit OR 0x80
    endif
    'output' digit
while ( X> 0 )

```

Ở đây, `MOD` là phép toán lấy phần dư (% in C), `DIV` trong số nguyên (/ in C), và `OR` phép HOẶC bit hay (`|` in C).

Thuật toán để decoding dãy byte ra giá trị thập phân ban đầu:

```

multiplier = 1
value = 0
do
    digit = 'next digit from stream'
    value += (digit AND 127) * multiplier
    multiplier *= 128
while ((digit AND 128) != 0)

```

Ở đây `AND` là phép VÀ bit (& in C).

Khi thuật toán kết thúc, giá trị còn lại trong `contains the bytes` là Remaining Length

Phần encoding của Remaining Length không phải là một phần của header. Số byte dùng để Encoding không được biết trong giá trị trường Remaining Length. Phần quy định có độ dài có thể thay đổi nằm ở Header, còn phần thay đổi thì không phải một phần của header.

2.2. Header có độ dài thay đổi được

Một vài loại MQTT message cũng chứa những phần có độ dài thay đổi được. Nó nằm giữa phần header cố định và phần payload.

Phần thay đổi được của Remaining Length sẽ không nằm ở phần header thay đổi được. Phần byte thay đổi được của Remaining Length thuộc về phần header và payload, xem

Fixed header để biết thêm chi tiết.

Định danh của các trường thay đổi được trong header được miêu tả bên dưới đây:

Protocol name

Xuất hiện trong message CONNECT. Trường này sẽ chứa tên MQIsdp ở dạng chữ hoa

Protocol version

Xuất hiện trong phần header của message CONNECT .

8 bit biểu diễn một giá trị không dấu là revision của protocol. Giá trị của trường Protocol version ở version hiện tại được miêu tả dưới bảng sau:

bit	7	6	5	4	3	2	1	0
Protocol Version								
	0	0	0	0	0	0	1	1

Connect flags

Giá trị này quy định cờ Clean Session, cờ Will, cờ Will QoS, và cờ Retain flags được chứa trong phần header thay đổi được của messageCONNECT

Clean session flag

Position: Bit đầu tiên của byte Connect flags

Nếu không được set (tức là giá trị bằng 1), thì server phải lưu lại những subscriptions sau khi client này đứt kết nối. Nó bao gồm việc tiếp tục lưu những message có QoS bằng 1 và QoS bằng 2 cho các topic đã subscribed vì thế, chúng có thể được chuyển đi khi client kết nối lại. Server cũng phải cập nhật trạng thái "lơ lửng" của message đang được chuyển đi tại thời điểm mà connection bị lost. Thông tin này phải được giữ đến khi client kết nối lại.

Nếu mà được set (tức giá trị bằng 1), thì server phải quên hết những thông tin trước đó về client, phải coi như là xóa sạch. Sever cũng phải quên hết mọi trạng thái của client này khi nó đứt kết nối.

Thông thường, client sẽ thực hiện trong 1 mode (hoặc là luôn clean hoặc là luôn không), ít khi thay đổi. Lựa chọn này phụ thuộc vào ứng dụng. Client kết nối với một clean conenction sẽ không nhận được những thông tin trước đó, nó phải subscribe lại tại mỗi lần kết nối. Client với kết nối dạng non-clean session sẽ không miss bất cứ gói tin QoS 1 hoặc QoS 2 nào, khi nó được published trong lúc client đứt kết nối. QoS 0 sẽ không bao giờ được lưu lại, vì vậy nó chỉ được chuyển đi theo cách thức cố gắng hết mức ở thời điểm nó được publish thôi.

Cờ này cũng được biết đến với một nghĩa khác "Clean start". Nó được đổi tên để phù hợp với ý nghĩa là nó được áp dụng cho toàn bộ session, chứ không riêng gì connect khởi tạo

Một server có thể cung cấp một cơ chế quản trị cho phép có xóa các thông tin đã lưu về client hay không. Nó có thể được sử dụng trong trường hợp biết chắc chắn rằng client nào đó không bao reconnect.

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
	x	x	x	x	x	x		x

Bit số 0 của byte này không được sử dụng troverserversion hiện tại của protocol. Nó có thể được sử dụng trong tương lai.

Will flag

Position: Bit thứ 2 của byte Connect flags

Trường này định nghĩa rằng, một message được published bởi server nhân danh cho một client mà server không thể giữ kết nối bằng Keep Alive hoặc có lỗi I/O trong quá trình kết nối với Client. Phải phân biệt với trường hợp server nhận được message DISCONNECT từ Client, khi đó nó sẽ không gửi Will message

Nếu cờ này được set, Will QoS và Will Retain sẽ được biểu diễn trong cờ Connect và các trường của Will Topic và Will Message phải có trong phần payload.

Định dạng của cờ Will được chỉ ra trong bảng dưới đây:.

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
	x	x	x	x	x		x	x

Bit 0 của byte này chưa được sử dụng trong phiên bản hiện tại của protocol này.

Will QoS

Position: Byte 3, 4 của byte Connect Flags

Một quy định về QoS cho message Will là trường Will QoS, message này được gửi để báo về event đứt kết nối với Client không tự nguyện. Message Will được định nghĩa trong phần payload của message of a CONNECT .

Nếu cờ Will flags được set, thì trường Will QoS sẽ là bắt buộc, ngoài ra nó sẽ bị bỏ qua.

Các giá trị của Will QoS is 0 (0x00), 1 (0x01), or 2 (0x02). Bảng miêu tả giá trị của cờ này được miêu tả dưới bảng sau.

bit	7	6	5	4	3	2	1	0

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
	x	x	x			1	x	x

Bit 0 không được dùng trong phiên bản này của protocol. Nó có thể được sử dụng trong tương lai.

Will Retain flag

Position: Bít 5 của byte Connect flags .

Cờ Will Retain flag sẽ quy định có hay không việc giữ lại message Will đã được published khi client đứt kết nối bất thường.

Cờ Will Retain flag là bắt buộc nếu Will flag được set, ngoài ra, nó sẽ bị bỏ qua. Định dạng của cờ này được miêu tả trong bảng sau.

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
	x	x		x	x	1	x	x

Bit 0 không được sử dụng trong version này của protocol. Nó có thể được sử dụng trong các phiên bản sau.

User name and password flags

Position: Bit 6, 7 của byte Connect flags .

Một kết nối từ client có thể chỉ ra user và password, thiết lập các trường để thông báo rằng có User và Password (tùy chọn) trong phần payload của Message CONNECT CONNECT .

Nếu trường User Name được set, thì trường User Name là bắt buộc, ngoài ra nó, sẽ bị bỏ qua. Nếu cờ Password được set, thì trường Password là bắt buộc, mặt khác nó sẽ bỏ qua giá trị này. Nó sẽ không quan tâm đến Password nếu không cung cấp Username.

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
			x	x	x	x	x	x

Bit 0 chưa được sử dụng trong phiên bản này của protocol. Nó có thể được sử dụng trong tương lai.

Keep Alive timer

The Keep Alive timer được chứa bên trong phần header thay đổi được của message CONNECT .

The Keep Alive timer, được tính bằng second, định nghĩa interval time lớn nhất giữa các message nhận từ client. Trường này giúp server phát hiện được khi nào kết nối đến client bị đứt mà không cần phải đợi TCP/IP timeout. Client có trách nhiệm gửi 1 message trong mỗi chu kì Keep Alive. Khi không xuất hiện các message liên quan đến Data trong chu kì này thì client phải gửi một message PINGREQ , và khi đó server sẽ trả lời bằng message PINGRESP.

Nếu server không nhận được message trong vòng 1.5 chu kì của Keep Alive (clients sẽ được "ân hạn" thêm một nửa chu kì), Nó sẽ disconnect với clients như là khi client gửi message DISCONNECT. Hành động này không ảnh hưởng đến bất kỳ subscriptions nào của client. Xem thêm về DISCONNECT để biết thêm chi tiết.

Nếu client không nhận được message PINGRESP trong khoảng 1 chu kì Keep Alive sau khi gửi một message PINGREQ, thì nó nên đóng kết nối TCP/IP lại.

Keep Alive timer là một giá trị 16 bit biểu diễn số second của chu kì này. Giá trị khi sử dụng sẽ phụ thuộc vào ứng dụng, nó thường là vài phút. Giá trị lớn nhất có thể là 18 giờ. Nếu giá trị này là Zero, có nghĩa là client không bị mất kết nối.

Định dạng của Keep Alive được chỉ ra dưới bảng sau. Thứ tự byte là byte cao trước(MSB), byte thấp sau(LSB).

bit	7	6	5	4	3	2	1	0
	Keep Alive MSB							
	Keep Alive LSB							

Connect return code

Mã kết nối được trả về được gửi trong phần header thay đổi được của message CONNACK message.

Trường này định nghĩa một bở một byte có giá trị ≥ 0 . Ý nghĩa của mỗi giá trị được miêu tả trong bảng bên dưới. Tuy giá trị trả về có ý nghĩa thay đổi theo từng loại message. Thông thường, trả về 0 có nghĩa là success.

Thứ tự	HEX	Ý nghĩa
0	0x00	Kết nối được chấp nhận.
1	0x01	Kết nối bị từ chối : phiên bản của protocol không được chấp nhận.
2	0x02	Kết nối bị từ chối: identifier bị bỏ, hay không hợp lệ
3	0x03	Kết nối bị từ chối: server không sẵn sàng
4	0x04	Kết nối bị từ chối: sai user name hoặc password
5	0x05	Kết nối bị từ chối: chưa được xác thực
6-255		Sử dụng trong tương lai

bit	7	6	5	4	3	2	1	0
	Return Code							

Topic name

The topic name được chứa trong phần header thay đổi được của một message PUBLISH .

The topic name là chìa khóa để xác định thông tin về kênh thông tin mà payload sẽ được published tới. Subscribers sử dụng trường này để xác định kênh thông tin mà nó muốn nhận được.

The topic name là một xâu UTF-encoded. Xem thêm ở chương MQTT and UTF-8 để biết thêm thông tin chi tiết. Topic name có giới hạn trên của độ dài là 32,767 kí tự.

2.3. Payload

Nhưng message dưới đây sẽ có payload:

CONNECT

Nó chứa một hoặc nhiều chuỗi UTF-8 encoded. Trong đó, chỉ định rõ một identifier duy nhất cho client, một Will topic and message and và cả user name, password nữa. Không phải tất cả chúng đều xuất hiện và sự xuất hiện của chúng được quy định bởi một tập các cờ phía trước.

SUBSCRIBE

Chứa một danh sách các topic name mà client có thể subscribe và cả QoS level nữa. Tất cả chúng cũng là UTF-encoded.

SUBACK

Payload sẽ chứa một danh sách của các QoS level được cho phép. Có một mức QoS level mà administrators của server cho phép client sử dụng để subscribe đến một Topic name cụ thể. Những QoS level này được liệt kê cùng thứ tự tương ứng với danh sách các Topic Name trong message SUBSCRIBE.

Phần payload của message PUBLISH chỉ chứa dữ liệu cho ứng dụng cụ thể. Không thể giả định chúng được tạo bởi dữ liệu tự nhiên hay nội dung của dữ liệu "sống", phần này sẽ được hiểu là một BLOB.

Nếu bạn muốn ứng dụng thực hiện việc nén dữ liệu đưa vào trong payload data, bạn cần định nghĩa trong ứng dụng một cờ payload tương ứng để xử lý nhưng dữ liệu payload này. Bạn không thể định nghĩa một cờ chỉ cho ứng dụng cụ thể bên trong bất cứ thành phần nào của header.

2.4. Message identifiers

The message identifier được xuất hiện trong phần header của các message dưới đây: PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP, SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK.

Trường Message Identifier (Message ID) chỉ xuất hiện trong message khi giá trị QoS bit miêu tả trong header chỉ định đến giá trị QoS 1 hoặc 2. Xem thêm ở Quality of Service levels and flows để biết thêm thông tin chi tiết.

The Message ID là một giá trị integer 16 bit không dấu và phải là duy nhất trong tập các message ở một hướng kết nối cụ thể nào đó. Thông thường người ta thường tăng lên sau mỗi message nhưng việc tăng này là không bắt buộc.

A client sẽ phải maintain một danh sách các message IDs để tách biệt với Message OIDs được sử dụng server mà nó kết nối đến. Điều này cho phép việc gửi một PUBLISH với message ID bằng 1 tại đúng thời điểm nhận một message được PUBLISH lên server với message ID bằng 1.

Thứ tự 2 byte ở đây là MSB , sau đó đến LSB

Đừng sử dụng Message ID là 0. Nó thường được sử dụng là Invalid Message ID

bit	7	6	5	4	3	2	1	0
Message Identifier MSB								
Message Identifier LSB								

2.5. MQTT and UTF-8

UTF-8 is là một encoding hiệu quả của chuỗi Unicode character-strings, nó vẫn tối ưu việc Encoding các kí tự ASCII trong môi trường giao tiếp dạng text tuần túy.

Trong MQTT, mỗi xâu được gắn với một phần prefixed có độ dài 2 byte, biểu diễn độ dài của toàn bộ xâu được Encoded, như bảng dưới đây:

bit	7	6	5	4	3	2	1	0
byte 1	String Length MSB							
byte 2	String Length LSB							
bytes 3 ...	Encoded Character Data							

String Length là độ dài byte của xâu sau khi được encoded, chứ không phải là số lượng kí tự, not the number of characters. Ví dụ như này, xâu "OTWP" được Encoded trong UTF-8 như bảng dưới đây.

bit	7	6	5	4	3	2	1	0
byte 1	Message Length MSB (0x00)							
	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
byte 2	Message Length LSB (0x04)							
	0	0	0	0	0	1	0	0
byte 3	'O' (0x4F)							
	0	1	0	0	1	1	1	1
byte 4	'T' (0x54)							
	0	1	0	1	0	1	0	0
byte 5	'W' (0x57)							
	0	1	0	1	0	1	1	1
byte 6	'P' (0x50)							
	0	1	0	1	0	0	0	0

Data stream của Java sử dụng `writeUTF()` and `readUTF()` theo định dạng như trên.

2.6. Unused bits

Những bit được đánh dấu không sử dụng thường được set bằng (0).

3. Command messages

3.1. CONNECT - Client yêu cầu connect đến server

Khi một kết nối TCP/IP được thiết lập từ client đến server, thì một session ở mức protocol cũng được tạo sử dụng luồng CONNECT.

Header cố định

Định dạng được thấy dưới bảng sau:

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (1)				DUP flag	QoS level		RETAIN
	0	0	0	1	x	x	x	x
byte 2	Remaining Length							

Các DUP, QoS, and RETAIN không được sử dụng trong message CONNECT .

Remaining Length là độ dài của phần header thay đổi được(12 bytes) và độ dài . Nó có thể là nhiều byte.

Variable header

Một ví dụ về định dạng của phần header biến đổi được có thể thấy trong bảng sau.

	Description	7	6	5	4	3	2	1	0
Protocol Name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (6)	0	0	0	0	0	1	1	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	'I'	0	1	0	0	1	0	0	1
byte 6	's'	0	1	1	1	0	0	1	1
byte 7	'd'	0	1	1	0	0	1	0	0
byte 8	'p'	0	1	1	1	0	0	0	0
Protocol Version Number									

	Description	7	6	5	4	3	2	1	0
byte 9	Version (3)	0	0	0	0	0	0	1	1
Connect Flags									
byte 10	User name flag (1) Password flag (1) Will RETAIN (0) Will QoS (01) Will flag (1) Clean Session (1)	1	1	0	0	1	1	1	x
Keep Alive timer									
byte 11	Keep Alive MSB (0)	0	0	0	0	0	0	0	0
byte 12	Keep Alive LSB (10)	0	0	0	0	1	0	1	0

User name flag

Set (1).

Password flag

Set (1).

Clean Session flag

Set (1).

Keep Alive timer

Set to 10 seconds (0x000A).

Will message

- Will flag is set (1)
- Will QoS field is 1
- Will RETAIN flag is clear (0)

Payload

Payload của message CONNECT chưa một hoặc nhiều xâu UTF-8 dựa trên các cờ ở phần header thay đổi được. Nhưng xâu này, nếu có sẽ xuất hiện theo thứ tự sau:

Client Identifier

Xâu được encode UTF đầu tiên. The Client Identifier (Client ID) có độ dài từ 1 đến 23 kí tự, và nó xác định một định danh duy nhất của client kết nối đến server. Nó phải là duy nhất khi so sánh với các clients khác, và nó là key để chuyển các message với QoS là 1 và 2. Nếu Client ID chứa nhiều hơn 23 kí tự, server sẽ trả lời yêu cầu CONNECT với mã là 2: Không chấp nhận định danh.

Will Topic

nếu Will Flag được set, thì đây sẽ là chuỗi UTF-8 tiếp theo. Message Will sẽ được published đến Will Topic. QoS level sẽ được xác định bằng Will QoS field , và RETAIN

status sẽ được xác định bằng cờ Will RETAIN flag trong phần header thay đổi.

Will Message

Nếu cờ Will Flag được set, thì đây là xâu UTF-8 tiếp theo. Message Will định nghĩa nội dung của message cái mà sẽ được published đến Will Topic nếu client bị disconnect bất thường. Xâu này cũng có thể là xâu rỗng tức là độ dài bằng 0.

Mặc dù message Will Message được encoded trong message CONNECT, khi nó published đến Will Topic thì chỉ bytes của messages được gửi, không phải là 2 byte quy định độ dài đầu tiên. Phần message này phải chỉ có thể chứa các kí tự ASCII 7 bit.

User Name

Nếu User Name flag được set, đây sẽ là xâu UTF-encoded tiếp theo. Trường này xác định tên của user đang kết nối đến, được sử dụng để authention. Dù không bắt buộc nhưng user name nên có độ dài từ 12 kí tự trở xuống.

Chú ý rằng, vì lý do tương thích với MQTT version 3, trường Remaining Length từ phần header cố định được ưu tiên so với cờ User Name. Server phải có khả năng cho phép trường hợp mà cờ User Name được set nhưng trường User Name bị thiếu vẫn hợp lệ. Nó nên được đánh giá là hợp lệ và vẫn cho kết nối được tiếp tục.

Password

Nếu Password flag được set, đây sẽ là xâu UTF-8 kế tiếp. Password cho user tương ứng đang kết nối đến, được sử dụng cho việc authentication. Nên để password có độ dài từ 12 kí tự trở xuống dù điều này không bắt buộc.

Chú ý rằng, vì lý do tương thích với MQTT version 3, trường Remaining Length field trong phần header cố định sẽ được ưu tiên hơn trạng thái cờ Password flag. Server nên cho phép trường hợp Password flag được set, nhưng không có xâu Password sẽ vẫn hợp lệ. Nó nên tiếp tục cho phép kết nối và thực hiện các message khác.

Response

Server sẽ gửi message CONNACK để trả lời message CONNECT từ client.

Nếu server không nhận được mesage CONNET từ client trong một khoảng thời gian nào đó sau khi thiết lập kết nối TCP/IP, thì server nên đóng kết nối đó lại.

Nếu client không nhận được một message CONNACK từ server trong một khoảng thời gian nhất định, thì client cũng nên đóng kết nối đó lại, và restart session bằng một socket mới đến server rồi tiếp tục gửi yêu cầu kết nối bằng gói CONNECT.

Trong cả 2 trường hợp trên, thời gian chờ để nhận được message CONNECT hoặc CONNACK phụ thuộc vào ứng dụng và điều kiện kết nối.

Nếu một client kết nối bằng một Client ID đang được kết nối với Server rồi, thì client trước đó phải được disconnect bắt buộc bởi server trước khi thực hiện luồng CONNECT với client mới .

Nếu client gửi một message CONNECT không hợp lệ, server nên đóng kết nối luôn. Không hợp ở đây bao gồm việc khác nhau về Protocol Name hoặc Protocol Version Numbers. Nếu server đã parse message CONNECT rồi mới phát hiện ra có một trường nào đó không hợp lệ, nó nên gửi lại message CONNACK chứa nội dung mã có nội dung "Kết nối bị từ chối: phiên bản protocol không được chấp nhận" trước khi hủy kết nối này.

3.2. CONNACK - Acknowledge connection request

Message CONNACK được gửi bởi server như để trả lời một yêu cầu a CONNECT từ client.

Phần header cố định

Có định dạng như bảng dưới đây.

bit	7	6	5	4	3	2	1	0
byte 1	Message type (2)				DUP flag	QoS flags		RETAIN
	0	0	1	0	x	x	x	x
byte 2	Remaining Length (2)							
	0	0	0	0	0	0	1	0

Các cờ DUP, QoS and RETAIN không được sử dụng trong message CONNACK.

Phần header thay đổi

Định dạng được chỉ ra ở bảng dưới đây.

	Description	7	6	5	4	3	2	1	0
Topic Name Compression Response									
byte 1	Reserved values. Not used.	x	x	x	x	x	x	x	x
Connect Return Code									
byte 2	Return Code								

Mỗi giá trị trả về của Connect được chỉ ra dưới bảng sau

Enumeration	HEX	Meaning
0	0x00	Kết nối được chấp nhận
1	0x01	Kết nối bị từ chối: Phiên bản protocol không được chấp nhận.
2	0x02	Kết nối vị từ chối: Định danh không hợp lệ
3	0x03	Kết nối vị từ chối: server không phục vụ được
4	0x04	Kết nối bị từ chối: Sai user name hoặc password

Enumeration	HEX	Meaning
5	0x05	Kết nối bị từ chối: Chưa được xác thực
6-255		Dành cho tương lai

Mã trả về là 2 sẽ (định danh bị từ chối) sẽ được gửi nếu định danh duy nhất cho 1 client có độ dài không nằm trong khoảng từ 1 đến 23.

Payload

Không có payload.

3.3. PUBLISH - Message Publish

Một message PUBLISH được gửi từ một client đến server để phân tán chúng đến các subscriber cần chúng. Mỗi message luôn gắn liền với một topic name (cũng được hiểu là Subject hoặc Channel). Topic là có dạng không gian phân cấp, nó định nghĩa ra một cách phân loại nguồn thông tin để cho các subscribers có thể subscribe nếu muốn. Một message được published đến một topic nào đó sẽ được phân phát đến các subscriber quan tâm đến topic đó.

Nếu 1 client subscribe một hoặc nhiều topic, thì mọi message được published đến những topic đó được gửi bởi server đến client như là một message PUBLISH.

Fixed header

Bảng dưới đây chỉ ra định dạng phần header cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message type (3)				DUP flag	QoS level		RETAIN
	0	0	1	1	0	0	1	0
byte 2	Remaining Length							

QoS level

Set là 1. Xem thêm về QoS để biết thêm chi tiết.

DUP flag

Set là (0). Có nghĩa là message được gửi ở lần đầu tiên. Xem thêm DUP để biết chi tiết.

RETAIN flag

Set là zero. Có nghĩa là không giữ lại. Xem thêm Retain để biết thêm chi tiết.

Remaining Length field

Độ dài của phần header thay đổi cộng phần payload. Nó có thể là trường nhiều byte

Variable header

Phần header có độ dài thay đổi được chứa các thành phần sau:

Topic name

Một xâu UTF.

Nó không chứa kí tự đại diện trong topic .

Khi nhận được yêu cầu subscribe từ client thì topic đó của thẻ chứa kí tự đại diện, Chuỗi này hoàn toàn được xác định bởi publisher chưa không phải là topic mà client đã subscribe.

Message ID

Được sử dụng cho các message có QoS là 1 hoặc 2. Xem Message identifiers để biết thêm chi tiết.

Bảng sau đây chỉ ra một ví dụ về header có độ dài thay đổi cho PUBLISH message.

Field	Value
Topic Name:	"a/b"
QoS level	1
Message ID:	10

Định dạng của phần header thay đổi trong trường hợp này được chỉ ra dưới bảng sau:.

	Description	7	6	5	4	3	2	1	0
Topic Name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (3)	0	0	0	0	0	0	1	1
byte 3	'a' (0x61)	0	1	1	0	0	0	0	1
byte 4	'/' (0x2F)	0	0	1	0	1	1	1	1
byte 5	'b' (0x62)	0	1	1	0	0	0	1	0
Message Identifier									
byte 6	Message ID MSB (0)	0	0	0	0	0	0	0	0
byte 7	Message ID LSB (10)	0	0	0	0	1	0	1	0

Payload

Chưa phần dữ liệu cho việc publishing. Nội dung và định dạng của dữ liệu là do ứng dụng quy định. Trường Remaining Length field trong phần độ dài cố định sẽ bao gồm cả phần độ dài thay đổi và phần payload. Như vậy, hoàn toàn hợp lệ khi một message PUBLISH chứa độ dài payload là 0.

Response

Response cho PUBLISH message phụ thuộc vào level của QoS. Bảng dưới đây sẽ chỉ ra chi tiết.

QoS Level	Giá trị trả về mong muốn
QoS 0	None
QoS 1	PUBACK
QoS 2	PUBREC

Actions

Các message PUBLISH được gửi từ Publisher đến server, hoặc là từ Server đến Subsciber. Việc tiếp thu khi nó nhận được messsage phụ thuộc vào QoS level của message:

QoS 0

Phải đảm bảo chắc chắn rằng các message đến các bên quan tâm.

QoS 1

Giữ một bản của messsage trên bộ nhớ ngoài, tất nhiên cũng phải đảm bảo nó đến các bên quan tâm, và trả lại một message PUBACK đến bên gửi.

QoS 2

Giữ lại một bản trên bộ nhớ ngoài, dừng gửi nó đến các bên quan tâm , trả về message PUBREC đến bên gửi.

Ở đây, nếu server nhận được message, các bên quan tâm nghĩa là các subscriber đến topic của message PUBLISH đó. Nếu một subscriber nhận một message, thì các bên quan tâm ám chỉ đến các ứng dụng bên phía client mà đã subscribe một hoặc nhiều topics và đang đợi một message đến từ server.

Xem thêm Quality of Service levels and flows để biết chi tiết.

Chú ý rằng nếu một server mà không xác nhận được một PUBLISH từ một client, thì nó không có cách nào để thông báo đến client đó. Vì thế nó phải đảm bảo một mức hiểu biết nhất định, tùy thuộc vào các rule QoS, client sẽ *không* được báo rằng việc xác thực message PUBLISH nó gửi đi.

3.4. PUBACK - Publish acknowledgment

Một message PUBACK sẽ được gửi trả cho một message PUBLISH với QoS level là 1. Một message PUBACK sẽ được gửi bởi serrver để trả lời một PUBLISH message, and by a

subscriber in response to a PUBLISH message from the server.

Fixed header

Bảng dưới đây chỉ ra cấu trúc của phần header cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (4)				DUP flag	QoS level		RETAIN
	0	1	0	0	x	x	x	x
byte 2	Remaining Length (2)				0	0	1	0
	0	0	0	0	0	0	1	0

QoS level

Không sử dụng.

DUP flag

Không sử dụng.

RETAIN flag

Không sử dụng.

Remaining Length field

Đây là độ dài của phần header thay đổi được. (2 bytes). Nó có thể là trường nhiều byte.

Variable header

Chứa phần Message Identifier (Message ID) cho message PUBLISH cái được trả lời ACK. Bảng sau chỉ ra định dạng của phần header thay đổi được.

bit	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							
byte 2	Message ID LSB							

Payload

Không có payload.

Actions

Khi client nhận một message PUBACK, nó sẽ hủy message đã PUBLISH vì message đó đã được nhận bởi server.

3.5. PUBREC - Chắc chắn nhận được publish (phần 1)

Một message PUBREC trả lời cho một message PUBLISH với QoS level là 2. Nó là message thứ 2 trong flow mà QoS level = 2. Mesage này sẽ coi như là một trả lời cho mesggePUBLISH đến từ client, or hoặc bởi một subscriber khi trả lời một message PUBLISH từ Server.

Fixed header

Bảng dưới đây chỉ ra định dạng của header có độ dài cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (5)					DUP flag	QoS level	RETAIN
	0	1	0	1	x	x	x	x
byte 2	Remaining Length (2)							
	0	0	0	0	0	0	1	0

QoS level

Không sử dụng.

DUP flag

Không sử dụng.

RETAIN flag

Không sử dụng.

Remaining Length field

Độ dài của phần header thay đổi được (2 bytes). Nó có thể là trường nhiều byte.

Variable header

Phần header thay đổi được chứa Message ID để trả lời cho message PUBLISH. Bảng sau đây chỉ ra định dạng của phần header thay đổi được.

bit	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							
byte 2	Message ID LSB							

Payload

Không có payload.

Actions

Khi nhận được một message PUBREC, bên nhận sẽ gửi một message PUBREL đến bên gửi với cùng Message ID với PUBREC message.

3.6. PUBREL - Release Publish đảm bảo(Phần 2)

Message PUBREL message là trả lời từ một publisher cho message PUBREC được gửi đến từ server, or hoặc được gửi server để trả lời cho message PUBREC đến từ một subscriber. Đây là message thứ 3 trong flow QoS Level = 2.

Fixed header

Bảng dưới đây chỉ ra định dạng của header có độ dài cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (6)				DUP flag	QoS level		RETAIN
	0	1	1	0	0	0	1	x
byte 2	Remaining Length (2)							
	0	0	0	0	0	0	1	0

QoS level

Các message PUBREL sử dụng QoS level = 1 như là một công nhận được mong muốn dạng một PUBCOMP. Các retry sẽ được xử lý như với message PUBLISH .

DUP flag

Set là zero (0). Có nghĩa là message này được gửi cho lần đầu. Xem thêm DUP để biết thêm chi tiết.

RETAIN flag

Không sử dụng.

Remaining Length field

Độ dài của phần header thay đổi được (2 bytes). Nó có thể là trường nhiều byte.

Variable header

Phần header thay đổi được chứa cùng Message ID với message PUBREC cái được hiểu như là ACK. Bản sau mô tả định dạng của trường header có độ dài thay đổi được.

bit	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							
byte 2	Message ID LSB							

Payload

Không có payload.

Actions

Khi server nhận được một message PUBREL từ publisher, server sẽ đưa message đó đến các subscriber quan tâm, và gửi message PUBCOMP với cùng một Message ID đến publisher. Khi một subscriber nhận một message PUBREL từ server, thì nó có thể sử dụng message đã nhận được cho các ứng dụng phía trên, đồng thời cũng gửi một message PUBCOMP đến server.

3.7. PUBCOMP - Publish đảm bảo hoàn thành (Phần 3)

Message này hoặc là trả lời từ server cho message PUBREL từ publisher, hoặc trả lời từ subscriber cho một message PUBREL từ server. Nó là message cuối cùng trong flow QoS level = 2.

Fixed header

Bảng dưới đây chỉ ra định dạng của phần header thay đổi được.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (7)				DUP flag	QoS level		RETAIN
	0	1	1	1	x	x	x	x
byte 2	Remaining Length (2)							
	0	0	0	0	0	0	1	0

QoS level

Không sử dụng.

DUP flag

Không sử dụng.

RETAIN flag

Không sử dụng.

Remaining Length field

Độ dài của phần header thay đổi được. (2 bytes). Nó có thể là trường nhiều byte.

Variable header

Phần header thay đổi chưa cùng Message ID với message PUBREL đã chấp nhận trước đó.

bit	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							
byte 2	Message ID LSB							

Payload

Không có payload.

Actions

Khi client nhận được một message PUBCOMP, nó sẽ quên message chính bởi vì nó đã được chuyển đến server chính xác một lần.

3.8. SUBSCRIBE - Subscribe to named topics

Message SUBSCRIBE cho phép client subscribe một hoặc nhiều topics với server. Message được published lên server sẽ được chuyển đến client bằng message PUBLISH. Message SUBSCRIBE cũng chỉ ra QoS level mà subscriber muốn nhận message.

Fixed header

Phần header cố định được miêu tả ở bảng dưới đây.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (8)				DUP flag	QoS level		RETAIN
	1	0	0	0	0	0	1	x
byte 2	Remaining Length							

QoS level

Message SUBSCRIBE sử dụng QoS level là 1 để chấp nhận nhiều yêu cầu subscription. Message SUBACK tương ứng được xác định bằng việc so sánh Message ID. Retries được handle giống với message PUBLISH.

DUP flag

Set là zero (0). Có nghĩa là message được gửi lần đầu. Xem DUP để biết thêm chi tiết hơn.

RETAIN flag

Không sử dụng.

Remaining Length field

Độ dài của payload. Có thể chứa nhiều byte.

Variable header

Phần header thay đổi chứa Message ID bởi vì một message SUBSCRIBE có QoS level là 1. Xem Message identifiers để biết thêm chi tiết.

Bảng dưới đây miêu tả một ví dụ về phần header có độ dài thay đổi được, với Message ID bằng 10.

	Description	7	6	5	4	3	2	1	0
Message Identifier									
byte 1	Message ID MSB (0)	0	0	0	0	0	0	0	0
byte 2	Message ID LSB (10)	0	0	0	0	1	0	1	0

Payload

Payload của một message SUBSCRIBE chứa một danh sách các topics mà các client đã subscribe, và QoS level ở mức mà client muốn. Xâu được encode ở dạng UTF, và QoS level chiếm 2 bits trong 1 single byte. Topic có thể chứa những kí tự đại diện đặc biệt để biểu diễn một tập các Topic. Cặp topic/QoS được nhóm liên tiếp nhau trong ví dụ của hình dưới đây.

Topic name	"a/b"
QoS đã yêu cầu	1
Topic name	"c/d"
QoS đã yêu cầu	2

Topic names trong message SUBSCRIBE sẽ không được nén.

Ví dụ về cấu trúc của payload được mô tả ở bảng dưới đây.

	Description	7	6	5	4	3	2	1	0
Topic name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (3)	0	0	0	0	0	0	1	1
byte 3	'a' (0x61)	0	1	1	0	0	0	0	1
byte 4	'/' (0x2F)	0	0	1	0	1	1	1	1
byte 5	'b' (0x62)	0	1	1	0	0	0	1	0
Requested QoS									
byte 6	Requested QoS (1)	x	x	x	x	x	x	0	1
Topic Name									
byte 7	Length MSB (0)	0	0	0	0	0	0	0	0
byte 8	Length LSB (3)	0	0	0	0	0	0	1	1
byte 9	'c' (0x63)	0	1	1	0	0	0	1	1
byte 10	'/' (0x2F)	0	0	1	0	1	1	1	1
byte 11	'd' (0x64)	0	1	1	0	0	1	0	0
Requested QoS									

	Description	7	6	5	4	3	2	1	0
byte 12	Requested QoS (2)	x	x	x	x	x	x	1	0

Giả định rằng QoS level yêu cầu được chấp nhận, client nhận message PUBLISH messages có QoS nhỏ hơn hoặc bằng level nó yêu cầu, sự trên các QoS từ các message gốc được published lên server. Ví dụ, nếu một client có QoS level = 1 cho 1 subscription đến một topic cụ thể, thì một message PUBLISH cho topic nào đó có QoS level = 0 sẽ được chuyển đến các client có QoS level = 0. Một message PUBLISH có QoS level = 2, đến cùng topic đó sẽ bị downgrade về QoS level = 1 khi chuyển đến client (yêu cầu QoS = 1).

Một hệ quả cho điều này là việc subscribe ở QoS Level = 2 sẽ tương đương với việc nói rằng :"Tôi muốn nhận các message ở topic này với các QoS giống với message nó được published lên".

Điều này có nghĩa rằng publisher coi QoS level như là QoS level lớn nhất mà message nó publish tới, nhưng một subscriber có thể downgrade QoS Level để phù hợp với mục đích sử dụng của nó. The QoS of a message is never upgraded.

Trường Requested QoS được chứa trong 1 byte kèm theo topic name được encode dạng UTF như bảng dưới đây.

bit	7	6	5	4	3	2	1	0
	Reserved	QoS level						
	x	x	x	x	x	x		

6 bit của byte này không được sử dụng trong phiên bản hiện tại của protocol. Chúng có thể được sử dụng trong tương lai.

Khi không có 2 bit QoS level thì packet nên được hiểu là hỏng và kết nối coi như bị đóng.

Response

Khi nhận được một message SUBSCRIBE message từ, server trả lời bằng message SUBACK.

Một server bắt đầu gửi một message PUBLISH cho yêu cầu của về subscription của client thậm chí trước khi client nhận được message SUBACK.

Chú ý rằng nếu một server không xác nhận một yêu cầu SUBSCRIBE từ client, thì nó không có cách nào thông tin cho client. Vì thế nó tạo ra message SUBACK như là để xác nhận, và client sẽ không được thông báo rằng nó không được xác thực.

Một server có quyền chỉ định một level of QoS thấp hơn client yêu cầu. Điều này có thể xảy ra nếu server không thể cung cấp các levels of QoS cao hơn. Ví dụ, nếu server không cung cấp một cơ chế lưu trữ tin cậy nào đó nó có thể chỉ cấp cho các subscriptions với QoS là 0.

3.9. SUBACK - Subscription acknowledgement

Message SUBACK message được gửi bởi server đến client để báo với client rằng nó đã nhận message SUBSCRIBE.

Message SUBACK chứa một danh sách QoS levels đã được cấp. Thứ tự các QoS levels bên trong message SUBACK sẽ khớp theo thứ tự của các topic name tương ứng trong message SUBSCRIBE.

Fixed header

Bảng dưới đây chỉ ra định dạng phần header cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (9)				DUP flag	QoS level		RETAIN
	1	0	0	1	x	x	x	x
byte 2	Remaining Length							

QoS level

Không sử dụng.

DUP flag

Không sử dụng.

RETAIN flag

Không sử dụng.

Remaining Length field

Độ dài của payload. Nó có thể là nhiều byte.

Variable header

Phần header thay đổi được chứa Message ID của message SUBSCRIBE, cái đang được xác nhận. Bảng dưới đây chỉ ra định dạng của phần header thay đổi được.

	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							
byte 2	Message ID LSB							

Payload

Phần payload chứa một vector của các QoS levels được cho phép. Mỗi level tương ứng với mỗi topic name trong message SUBSCRIBE. Thứ tự của QoS levels trong message SUBACK khớp với thứ tự topic name và cặp QoS yêu cầu trong message SUBSCRIBE. Message ID trong phần header thay đổi cho phép match message SUBACK với các message SUBSCRIBE tương ứng.

Bảng dưới đây chỉ ra trường QoS được encode trong 1 byte.

bit	7	6	5	4	3	2	1	0
	Reserved	QoS level						
	x	x	x	x	x	x	x	

6 bit đầu của byte không được sử dụng trong phiên bản hiện của trình duyệt. Có thể sẽ được sử dụng trong tương lai.

Bảng dưới đây chỉ ra một ví dụ của payload.

Granted QoS	0
Granted QoS	2

The table below shows the format of this payload.

	Description	7	6	5	4	3	2	1	0
byte 1	Granted QoS (0)	x	x	x	x	x	x	0	0
byte 1	Granted QoS (2)	x	x	x	x	x	x	1	0

3.10. UNSUBSCRIBE - Unsubscribe from named topics

Một message UNSUBSCRIBE được gửi bởi client đến server để unsubscribe một topic name nào đó.

Fixed header

Bảng sau chỉ ra một ví dụ của phần header cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (10)				DUP flag	QoS level		RETAIN
	1	0	1	0	0	0	1	x
byte 2	Remaining Length							

QoS level

Các message UNSUBSCRIBE sử dụng QoS level 1 để chấp nhận nhiều unsubscribe request. Message UNSUBACK tương ứng được xác định bởi Message ID. Các Retries được thực hiện theo cùng một cách với message PUBLISH .

DUP flag

Set là zero(0). Có nghĩa là message này được gửi lần đầu. Xem message DUP for để biết thêm chi tiết.

RETAIN flag

Không sử dụng.

Remaining Length

Đây là độ dài của Payload. Nó có thể là trường nhiều byte.

Variable header

Phần header thay đổi chứa Message ID bởi vì một message UNSUBSCRIBE message có QoS level of 1. Xem thêm Message identifiers để biết thêm chi tiết.

Bảng sau chỉ ra một ví dụ về định dạng của phần header thay đổi với Message ID là 10.

	Description	7	6	5	4	3	2	1	0
Message Identifier									
byte 1	Message ID MSB (0)	0	0	0	0	0	0	0	0
byte 2	Message ID LSB (10)	0	0	0	0	1	0	1	0

Payload

Client unsubscribes từ một danh sách topics named trong phần payload. Chuỗi được encode theo dạng UTF và được đóng gói liên tiếp nhau. Các topic name trong message UNSUBSCRIBE sẽ không được nén. Bảng dưới đây mô tả một ví dụ của payload.

Topic Name	"a/b"
Topic Name	"c/d"

Bảng sau chỉ ra định dạng của payload.

	Description	7	6	5	4	3	2	1	0
Topic Name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (3)	0	0	0	0	0	0	1	1
byte 3	'a' (0x61)	0	1	1	0	0	0	0	1
byte 4	'/' (0x2F)	0	0	1	0	1	1	1	1
byte 5	'b' (0x62)	0	1	1	0	0	0	1	0
Topic Name									
byte 6	Length MSB (0)	0	0	0	0	0	0	0	0
byte 7	Length LSB (3)	0	0	0	0	0	0	1	1
byte 8	'c' (0x63)	0	1	1	0	0	0	1	1

	Description		7	6	5	4	3	2	1	0
byte 9	'/' (0x2F)		0	0	1	0	1	1	1	1
byte 10	'd' (0x64)		0	1	1	0	0	1	0	0

Response

The server sends an UNSUBACK to a client in response to an UNSUBSCRIBE message.

3.11. UNSUBACK - Unsubscribe acknowledgment

The UNSUBACK message is sent by the server to the client to confirm receipt of an UNSUBSCRIBE message.

Fixed header

The table below shows the fixed header format.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (11)				DUP flag	QoS level		RETAIN
	1	0	1	1	x	x	x	x
byte 2	Remaining length (2)							
	0	0	0	0	0	0	1	0

QoS level

Not used.

DUP flag

Not used.

RETAIN flag

Not used.

Remaining Length

The length of the Variable Header (2 bytes).

Variable header

Phần header thay đổi được chứa Message ID cho message UNSUBSCRIBE được chấp nhận. Bảng sau chỉ ra phần header thay đổi được.

bit	7	6	5	4	3	2	1	0
byte 1	Message ID MSB							

bit	7	6	5	4	3	2	1	0
byte 2	Message ID LSB							

Payload

Không có payload.

3.12. PINGREQ - PING request

Message PINGREQ có nghĩa là message "Ông vẫn sống đấy chứ?" được gửi từ một client đã kết nối đến server.

Xem Keep Alive timer để biết thêm chi tiết.

Fixed header

Bảng sau chỉ ra định dạng của phần header cố định.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (12)				DUP flag	QoS level		RETAIN
	1	1	0	0	x	x	x	x
byte 2	Remaining Length (0)							
	0	0	0	0	0	0	0	0

Các cờ DUP, QoS, and RETAIN không được sử dụng.

Variable header

Không có phần header thay đổi.

Payload

Không có payload.

Response

Trả lời cho message PINGREQ là một message PINGRESP.

3.13. PINGRESP - PING response

Một message PINGRESP được gửi từ server cho một message PINGREQ và nó có nghĩa là "Vâng, tôi vẫn sống đây".

Xem thêm Keep Alive timer để biết thêm chi tiết.

Fixed header

Bảng sau chỉ ra định dạng của một header cố định:

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (13)				DUP flag	QoS level		RETAIN
	1	1	0	1	x	x	x	x
byte 2	Remaining Length (0)							
	0	0	0	0	0	0	0	0

Các cờ DUP, QoS, and RETAIN không sử dụng.

Payload

Không có payload.

Variable header

Không có phần header thay đổi.

3.14. DISCONNECT - Disconnect notification

Message DISCONNECT được gửi từ client đến server để báo rằng nó sẽ đóng kết nối TCP/IP đang kết nối. Cái này cho phép một clean disconnection, chứ không chỉ là hủy kết nối.

Nếu client đang kết nối với cờ clean session được set, thì tất cả những thông tin trước đó về client sẽ bị quên hết.

Một server không nên để việc đóng kết nối này cho phía client sau khi nhận message DISCONNECT.

Fixed header

Định dạng của phần header cố định được chỉ ra dưới bảng sau.

bit	7	6	5	4	3	2	1	0
byte 1	Message Type (14)				DUP flag	QoS level		RETAIN
	1	1	1	0	x	x	x	x
byte 2	Remaining Length (0)							
	0	0	0	0	0	0	0	0

Các cờ DUP, QoS, and RETAIN không được sử dụng trong message DISCONNECT.

Payload

Không có payload.

Variable header

Không có phần header thay đổi.

4. Flows

4.1. Quality of Service levels and flows

MQTT phân phối các message dựa trên mức độ tin cậy được định nghĩa là Quality of Service (QoS). Các level được định nghĩa như sau:

QoS level 0: At most once delivery

Message được phân phối dựa trên best efforts của tầng mạng TCP/IP bên dưới. Một response sẽ không được expected and không có một ngữ cảnh về việc gửi lại được định nghĩa trong giao thức. Các message đến server hoặc chỉ 1 lần hoặc không bao giờ.

Bảng dưới đây chỉ ra chi tiết về QoS level 0 trong luồng protocol.

Client	Message and direction	Server
QoS = 0	PUBLISH ----->	Action: Publish message to subscribers

QoS level 1: At least once delivery

Việc nhận được message bên phía server được xác nhận bởi một message PUBACK. Nếu có lỗi do kết nối hoặc gửi đến device, hoặc message xác nhận không nhận được sau một khoảng thời gian nhất định, sender sẽ gửi lại message với và set DUP bit trong phần header của message header. Message đến server ít nhất 1 lần. Cả message SUBSCRIBE and message UNSUBSCRIBE đều sử dụng QoS level là 1.

Một message mà có QoS level 1 trong phần header sẽ giống như bảng sau.

Bảng dưới đây chỉ ra luồng protocol khi QoS level set bằng 1.

Client	Message and direction	Server
QoS = 1 DUP = 0 Message ID = x Action: Store message	PUBLISH ----->	Actions: <ul style="list-style-type: none"> • Store message • Publish message to subscribers • Delete message

Client	Message and direction	Server
Action: Discard message	PUBACK ----->	

Nếu client không nhận message PUBACK (hoặc là trong một khoảng thời gian được định nghĩa bởi ứng dụng, hoặc nếu có lỗi xảy ra và communications session bị restart), client có thể gửi lại message PUBLISH với cờ DUP được set.

Khi nhận được một message lặp lại từ phía client, server sẽ publish các message đến các subscribers, và gửi một message PUBACK khác.

QoS level 2: Exactly once delivery

Một luồng được thêm vào luồng QoS level bằng 1 ở trên để đảm bảo rằng message bị lặp lại không bị chuyển đến ứng dụng. Đây là mức độ cao nhất khi phân phối message, không message lặp nào được chấp nhận. Lưu lượng mạng sẽ tăng lên, nhưng nó thường được chấp nhận vì mức độ quan trọng của nội dung message.

Một message với QoS level = 2 có một Message ID trong phần header của nó.

Bảng sau chỉ ra luồng của protocol khi QoS level = 2. Có 2 semantics để xác định làm thế nào luồng PUBLISH được handler bởi các bên. Chúng ảnh hưởng đến các điểm nằm trong flow đưa các message đến các subscriber. Việc lựa chọn semantics nào thì phụ thuộc vào thực thi của giao thức và không ảnh hưởng đến tính đảm bảo của luồng QoS level = 2 .

Client	Message and direction	Server
QoS = 2 DUP = 0 Message ID = x Action: Store message	PUBLISH ----->	Action: Store message <i>or</i> Actions: <ul style="list-style-type: none">• Store message ID• Publish message to subscribers
	PUBREC -----<	Message ID = x
Message ID = x	PUBREL ----->	Actions: <ul style="list-style-type: none">• Publish message to subscribers• Delete message <i>or</i> Action: Delete message ID

Client	Message and direction	Server
Action: Discard message	PUBCOMP -----	Message ID = x

Nếu phát hiện lỗi, hoặc sau một khoảng thời gian nhất định, luồng protocol sẽ được thực hiện lại từ kết quả của message xác nhận cuối cùng; hoặc là PUBLISH, hoặc là PUBREL. Xem Message delivery retry để biết thêm chi tiết. Luồng protocol đảm bảo rằng message đến các subscriber chỉ đúng 1 lần.

Assumptions for QoS levels 1 and 2

Trong môi trường mạng, việc đứt kết nối, hay có lỗi kết nối là điều luôn có khả năng xảy ra. Nếu xảy ra điều này, một bên sẽ không biết có điều gì xảy ra với bên đang giao tiếp với mình; Nó được gọi là *của sổ in doubt(nghi ngờ)*. Trong tình huống của giao thức này, giả định rằng là mọi thứ được thực hiện trên mức độ tin tưởng vào network, device,...

MQTT giả định rằng client và server nói chung là tin tưởng được nhau, và và các kênh truyền (communications channel) thì sẽ được coi như là không tin tưởng. Nếu thiết bị phía client gặp lỗi, thì thông thường nó sẽ là một lỗi nghiêm trọng, chứ không phải là một lỗi có thể bỏ qua được. Khả năng phục hồi dữ liệu từ thiết bị gặp lỗi là thấp. Một vài thiết bị có trang bị cả non-volatile storage, ví dụ flash ROM. Việc trang bị thêm persistent storage trên thiết bị phía client sẽ bảo vệ hầu hết các dữ liệu quan trọng trong một vài trường hợp lỗi.

Hiểu những điều cơ bản về các lỗi có thể phát sinh trong the communications link, mà trên các lỗi sẽ trở nên phức tạp, dẫn đến các tình huống sẽ phát sinh vượt quá phạm vi mà MQTT có thể handle.

4.2. Message delivery retry

Mặc dù TCP thông thường đảm bảo việc giao nhận của các packets, nhưng cũng có nhiều tình huống mà message MQTT không được nhận. Trong trường hợp của message MQTT, nó yêu cầu một response (bằng QoS >0 PUBLISH, PUBREL, SUBSCRIBE, UNSUBSCRIBE), nếu không nhận được response trong một khoảng thời gian nhất định, thì bên gửi có thể thực hiện lại việc giao nhận. Và bên gửi cũng set cờ DUP trong message gửi lại đó.

Time để thực hiện việc gửi lại này nên được set bằng các tùy chọn cấu hình. Tuy nhiên phải các rằng message không bị timeout chừng nó vẫn đang được gửi. Ví dụ, gửi một message lớn qua trong một mạng tốc độ chậm đương nhiên sẽ mất thời gian lâu hơn việc gửi một message nhỏ trong một mạng tốc độ nhanh. Việc gửi đi gửi lại các message bị timeout có thể dẫn đến kết quả rất tồi tệ vì thế sử dụng một chiến lược như là tăng giá trị timeout sau mỗi lần retry qua mỗi lần gửi lại thì nên được cân nhắc.

Khi client kết nối lại, nếu nó không được đánh dấu là clean session, thì cả client và server nên tổ chức giao nhận các message lơ lửng trước đó.

Ngoài hành động "on reconnect" này ra, client sẽ không được yêu cầu nhận các message trước đó. Tuy nhiên, Brokers, nên gửi lại bất cứ message nào mà chưa được báo đã nhận.

4.3. Message ordering

Thứ tự các message có thể bị ảnh hưởng của vô vàn các yếu tố, như là Có bao nhiêu message PUBLISH lơ lửng trong luồng mà một client chấp nhận, rồi thì phía client và single-thread hay multi-thread. Trong giới hạn của protocol, client được giả định là single-threaded trên quan điểm về packet khi nhận và gửi từ network.

Một thực thi của giao thức khi cung cấp phải đảm bảo dù với bất cứ thứ tự nào của message nó phải chắc chắn rằng luồng giao nhận được thực hiện hoàn tất theo thứ tự nó đã bắt đầu Ví dụ, trong luồng QoS level 2, luồng PUBREL phải được gửi cùng flow với luồng PUBLISH flows:

Client	Message and direction	Server
	PUBLISH 1 -----> PUBLISH 2 -----> PUBLISH 3 ----->	
	PUBREC 1 <----- PUBREC 2 <-----	
	PUBREL 1 ----->	
	PUBREC 3 <-----	
	PUBREL 2 ----->	
	PUBCOMP 1 <-----	
	PUBREL 3 ----->	
	PUBCOMP 2 <----- PUBCOMP 3 <-----	

Số lượng message lơ lửng được cho phép cũng có hiệu quả trong các loại đảm bảo mà có thể đưa ra:

- Với giới hạn cho các message in-flight là 1, thì mỗi luồng giao nhận sẽ được hoàn thành trước khi bước tiếp theo bắt đầu. Điều này đảm bảo các message được nhận theo đúng thứ tự chúng được đưa lên.
- Với giới hạn cho các message in-flight lớn hơn 1, thứ tự message chỉ đảm bảo trong 1 với một QoS Level nhất định.

Appendix A - Topic wildcards

Một subscription có thể có thể chứa nhiều kí tự đặc biệt, cho phép các subscriber có thể subscribe nhiều topic một lúc.

Kí tự phân các topic level được sử dụng để phân tách cấu trúc một topic, và và có thể được chỉ định bên trong topic đó. Một wildcard nhiều level và wildcard đơn level có thể được sử dụng trong các subscriptions, nhưng chúng không thể được sử dụng trong một topic trong message của publisher.

Topic level separator

Kí tự slash xuôi (/) được sử dụng để phân tách mỗi level bên trong một cây topic và cung cấp một cấp trúc phân cấp cho không gian. Việc sử dụng kí tự phân tách này có ý nghĩa khi 2 kí tự phân tách mâu thuẫn nhau .

Multi-level wildcard

Kí tự thăng (#) là kí tự đại diện và nó matches với bất kì level nào trong topic. Ví dụ, nếu bạn subscribe `finance/stock/ibm/#`, bạn sẽ nhận được các message ở trong topic sau:

```
finance/stock/ibm
finance/stock/ibm/closingprice
finance/stock/ibm/currentprice
```

The multi-level wildcard có thể biểu diễn zero hoặc nhiều level. Vì vậy, `finance/#` có thể match riêng cho `finance`, khi đó # biểu diễn level zero. Kí tự phân tách level không có ý nghĩa để phân tách trong trường hợp này.

The multi-level wildcard có thể có thể chỉ định chính nó hoặc gần với kí tự đại diện topic sau nó. Vì thế, # và `finance/#` đều hợp lệ, nhưng `finance#` không hợp lệ. Kí tự đại diện multi-level phải là kí tự cuối cùng trong dãy miêu tả topic. Ví dụ, `finance/#` là hợp lệ nhưng `finance/#/closingprice` thì không hợp lệ.

Single-level wildcard

Dấu cộng (+) là một kí tự đại diện nó phù hợp chỉ với một topic level. Ví dụ, `finance/stock/+` sẽ matches `finance/stock/ibm` và `finance/stock/xyz`, nhưng không matches `finance/stock/ibm/closingprice`. Cũng bởi vì, kí tự đại diện single-level chỉ matches với single level, `finance/+` sẽ không matches `finance`.

Kí tự đại diện single-level có thể được sử dụng ở bất kì level trong topic tree, và có thể kết hợp với multi-level wildcard. Nó phải nằm sau một kí tự phân cách level, trừ khi nó chỉ định chính nó. Vì thế, + and `finance/+` thì hợp lệ, nhưng `finance+` thì không hợp lệ. Kí tự single-level có thể sử dụng ở cuối topic tree hoặc là bên trong topic tree. Ví dụ, `finance/+` và `finance/+/ibm` là không hợp lệ.

Topic semantics and usage

Khi build một ứng dụng, việc thiết kế cây topic nên tính đến các nguyên tắc sau của tên topic name và semantics:

- Một topic nên có ít nhất 1 kí tự.
- Topic name là phân biệt chữ hoa, chữ thường. Ví dụ, ACCOUNTS and Accounts là 2 cái tên khác nhau.
- Topic name có thể bao gồm dấu cách. Ví dụ, Accounts payable là hợp lệ.
- Việc bắt đầu bằng dấu "/" sẽ tạo ra một topic khác. Ví dụ, /finance sẽ khác với finance. /finance sẽ matches "+/+ and "/+", nhưng không matches với "+".
- Đừng có đưa kí tự null (Unicode \x0000) vào trong bất kì topic nào.

Những nguyên tắc dưới đây được áp dụng cho việc tạo và cho nội dung của cây topic :

- Độ dài được giới hạn là 64k nhưng bên trong đó không có giới hạn về số level của topic.
- Bao nhiêu root nodes cũng được; bao nhiêu topic tree cũng được.