**MQTT协议评估对比**

**V0.1**

# 概览

MQTT（MQ Telemetry Transport）是轻量级的、 基于代理的发布/订阅消息传输协议，比较适用于：

* 网络代价昂贵，带宽低、不可靠。
* 在嵌入设备中运行，处理器和内存资源有限。

该协议的特点包括：

* 使用发布/订阅消息模式。
* 使用TCP/IP提供基础的网络连接。
* 有三种消息传递服务质量：
* “At most once”，消息发布完全依赖于底层的TCP/IP网络， 会发生消息丢失或重复。
* At lease once，确保消息到达，但可能发生消息重复。
* Exactly once，确保消息只到达一次。
* 固定头部长度很小，只有2字节。
* 提供一种机制，使得客户端异常中断时，能够使用 LastWill 和 Testament 特性通知有关各方。

# 对比

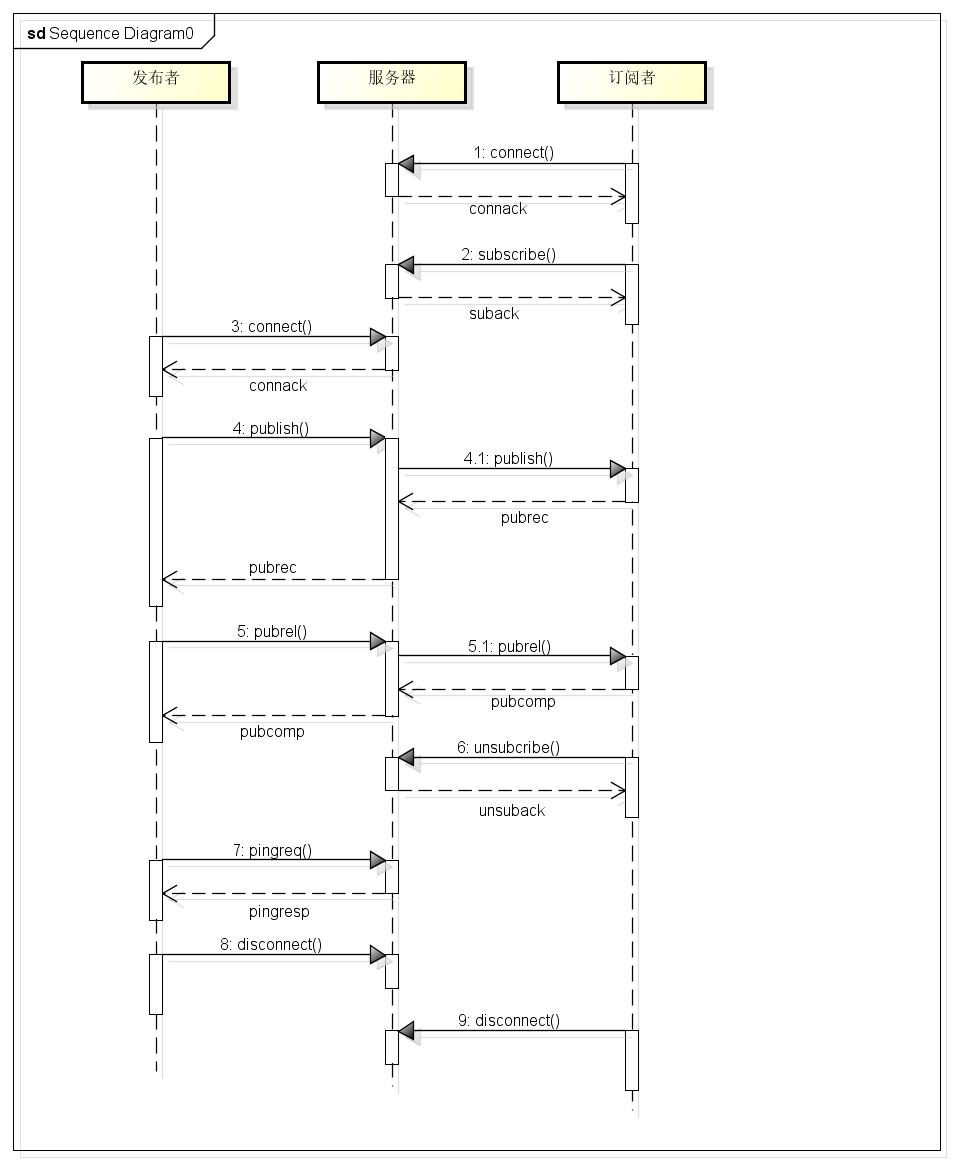
## 模型对比

* MQTT本身是一个消息订阅的协议，不太适合用作IM，尤其没有好友，群组等概念。MQTT比较适合做消息推送系统。
* MQTT的消息总类个数是有上限的，只有4个bit来表征，不好扩展。
* MQTT的典型应用场景是通过卫星和代理连接，通过拨号和医疗保健提供者连接，以及在一些自动化或小型设备上。

## 实现对比

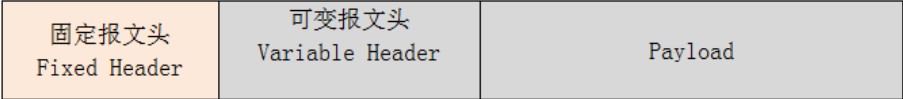
不够成熟、实现较复杂、服务端组件rsmb不开源，部署硬件成本较高

## 消息流程图



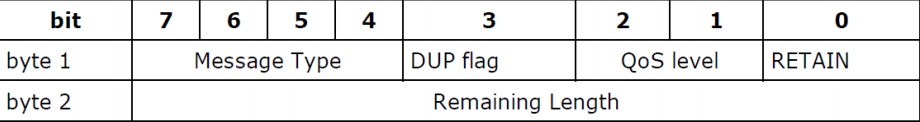
# 消息格式

每个MQTT命令消息的消息头都包含一个固定的报头。有些消息需要一个可变的报头和一个payload。



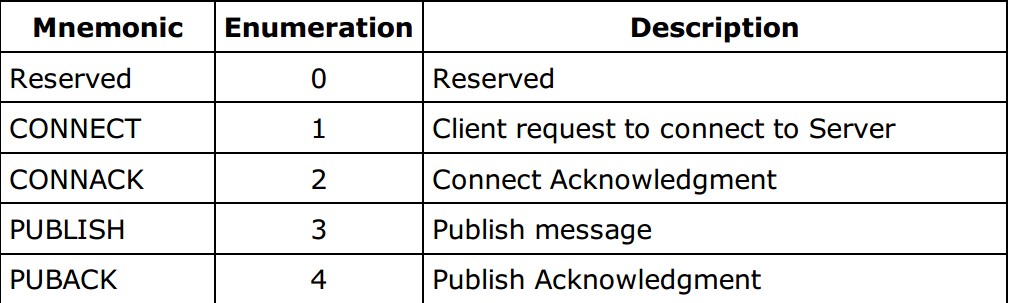
## 固定报头

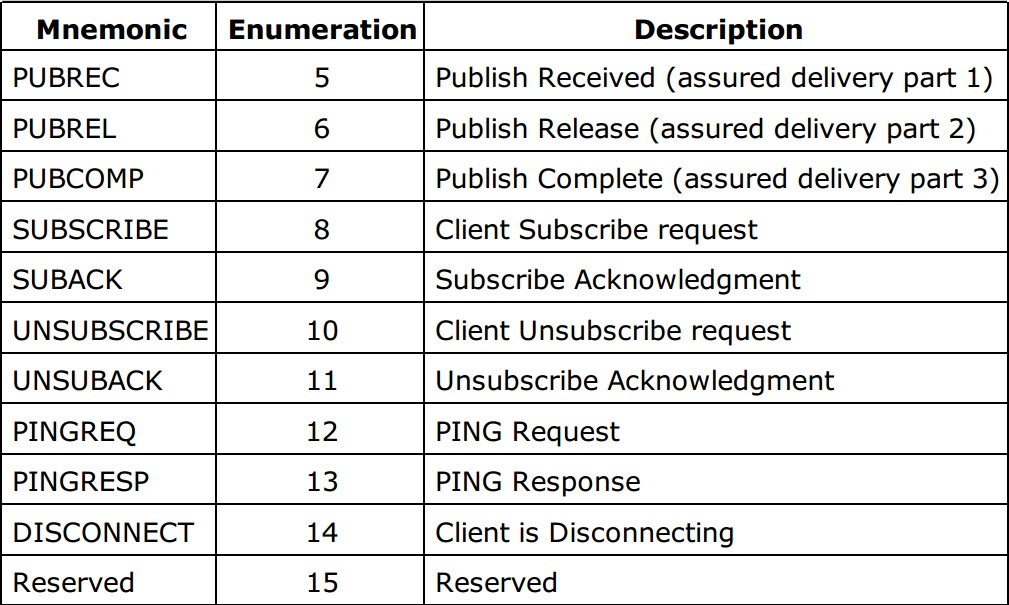
每个MQTT命令消息的消息头都包含一个固定的报头。下表显示了固定的报头格式：



### Message Type

表现为一个4-bit的无符号值，这个版本协议的枚举值如下表所示：



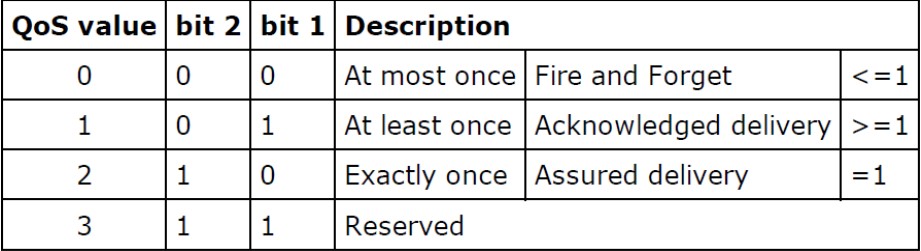


### DUP Flag

todo

### QOS Flag

QOS表明了服务质量等级：

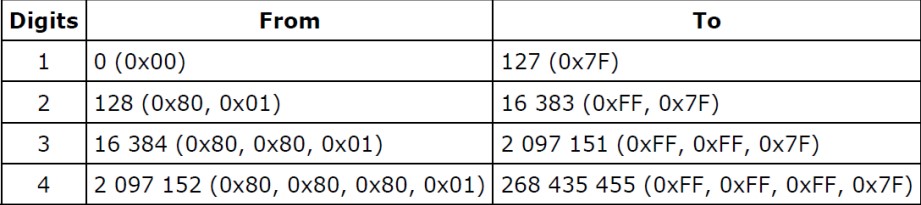


### RETAIN Flag

当客户端发布一个消息到服务器端时，如果Retain flag被设为1, 则服务器在将消息分发给当前的订阅者后应该保留这条消息,如果一个新的订阅者订阅某一个主题，就会收到这条消息。

### Remaining Length

最多4个字节，如果当前字节的首bit是1，代表下一个字节也是Remaining Length的组成部分，下表显示了增加字节数后，Remaining Length所表示的值：



## 可变报头

固定报文头后面是可变报头和负载（不是每种消息都有）。

### 协议名

出现在CONNECT消息中，代表协议名MQlsdp。

### 协议版本

8位的无符号值，当前协议为3（0x03）。

### connect的可变报头flag

#### clean session flag

如果设置这个位，那么客户端订阅者断开连接后，服务器必须存储客户端的订阅信息。通常情况下，一个客户端要么采用这个模式，要么采用另外一个模式，且不会改变。

#### Will flag

如果服务端和客户端的通讯断开，并且客户端没有发送DISCONNECT信息，服务端会代表客户端发布一个will消息。

如果设置了Will flag，Connect flags的字节中就必须有Will QoS和Will Retain字段，payload中必须有Will Topic和Will message字段。

Todo

#### Will retain flag

Will Retain flag指明了服务器是否应该保留Will消息，此Will消息是在客户端意外断开连接时由服务器代表客户端所发布的。

如果设置Will flag，则Will Retain flag是强制性的，否则Will Retain flag将被忽略。

#### User name and password flags

连接中的客户端可以指定用户名和密码，设置这两个标志位意味着一个用户和一个可选的密码会包含在CONNECT消息的payload里。

如果设置了User Name标志，则User Name字段是强制性的，否则User Name的值将被忽略。如果设置Password标志，则Password字段是强制性的，否则Password的值将被忽略。

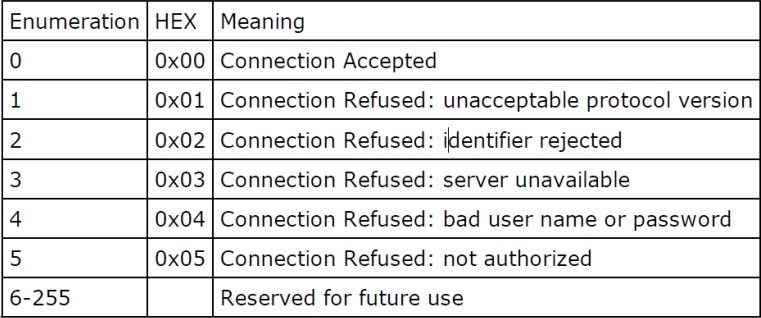
### Keep Alive timer

Keep Alive timer位于MQTT CONNECT消息的可变报文头中，是一个16位的值用于表征客户端和服务端直接的心跳时间，客户端有义务在没有业务的时候发送PINGREQ消息，而服务端回复PINGRESP。

### Connect return code

Connect return code是在CONNACK消息的可变报文头，这个字段定义了一个

无符号的byte返回码。



### Topic name

主题名位于MQTT PUBLISH消息的可变报文头中，指明了payload数据会被发布到哪个信息通道。

## 负载

下列几种类型的MQTT命令消息含有负载：

### Connect

Payload包含一个或多个UTF-8编码的字符串，为客户端指定了一个唯一的标识符 （ClientID），一个Will主题和消息 （Will topic/Will Message） 、所用的User Name和Password。 除了ClientID，其他都是可选的，是否含有可选内容是由可变报文头中的标识（flags）决定的。

### SUBACK

Todo

## 消息识别符

Todo

# 具体消息

## CONNECT

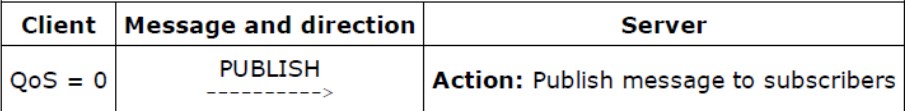
Todo

# 协议流

## QOS等级和流程

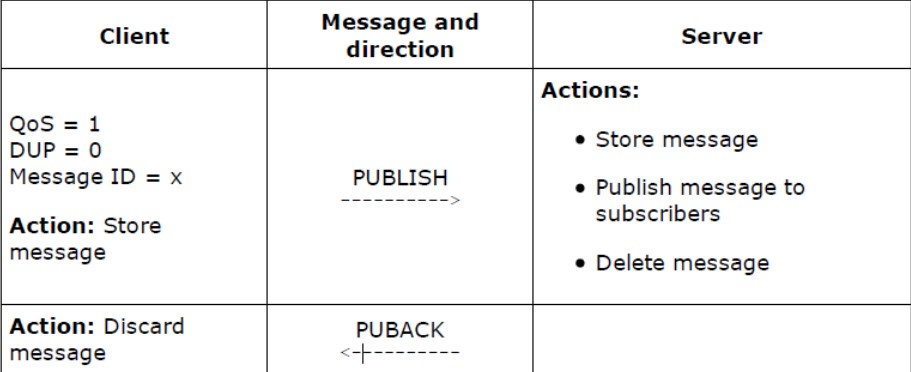
### QoS level 0: At most once delivery

消息的传递完全依赖底层的TCP/IP网络，协议里没有定义应答和重试。消息要么只会到达服务端一次，要么根本没有到达。



### QoS level 1: At least once delivery

服务器的消息接收由PUBACK消息进行确认。如果通信链路异常、发送设备异常，或者指定时间内没有收到确认消息，发送端会重发这条在消息头中设置DUP位的消息。



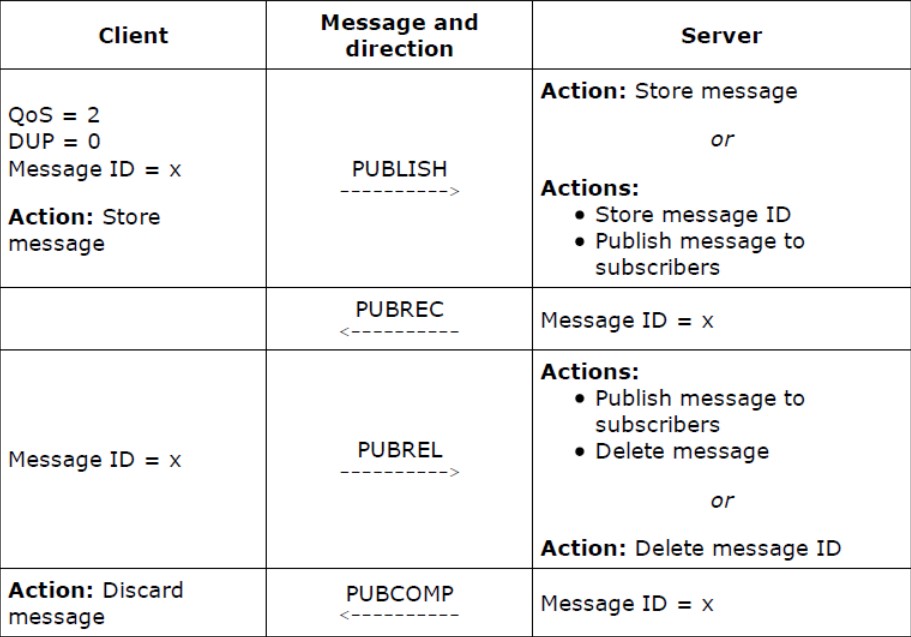
如果客户端没有收到PUBACK消息（在应用中定义的时间范围内，或者检测到了一个失败且通讯会话被重启），客户端可能重发这条设置了DUP标志的PUBLISH消息。

当从客户端收到了一条重复的消息，服务器会向订阅者重新发布消息，且会发送另外一个

PUBACK消息。

### QoS level 2: Exactly once delivery

在QoS level 1之上的附加协议流能够确保重复的消息不会被传递给正在接收的应用程序。当不允许出现重复消息时，这就是最高级别的传递。一个QoS level 2的消息在消息头中有一个Message ID。



### QOS1和QOS2的假设

MQTT假设客户端和服务器通常都是可靠的，通讯通道更可能是不可靠的。如果客户端设备发生故障，将是灾难性而不是一个暂时性的故障。从设备恢复数据的可能性非常低。有些设备有非易失性（non-volatile）存储，如flash ROM。在客户端设备上提供更持久的存储可以保护大多数重要的数据不受某些异常的而影响。除了基本的通讯线路异常外，异常模式矩阵变得很复杂，很多场景是MQTT所无法处理的。

## 消息传递重试

Todo

## 消息顺序

为了保证消息的顺序，必须保证每个消息传递流的存储都是按照想要的顺序。也就是说，每个工作流程都不能乱，都是完全按照 PUBLISH 的流程进行，虽然有的消息的某个步骤快，

有的慢。

