MQTT 协议

概述

MQTT（Message Queue Telemetry Transport， 消息队列遥测传输协议），给予发布/订阅（public/subscribe）模式的“轻量级”通讯协议，该协议构建于 TCP/IP 协议上。

优点：可以以极少的代码和有限的贷款，为链接远程设备童工实时可靠的消息服务。作为一种低开销、低带宽占用的即时通讯协议，在物联网、小型设备、移动应用等方面有广泛应用。

MQTT 协议是轻量、简单、开放和易于实现。在很多情况下，包括受限的环境中，如：机器与机器通信（M2M）和物联网（IoT）。在卫星链路通信传感器、偶尔拨号的医疗设备、智能家具、一些小型设备中广泛应用。

设计原则

因物联网环境的特别，MQTT遵循一下设计原则：

1. 精简，不添加可有可恶的功能；
2. 发布/订阅（pub/sub）模式，方便消息在传感器之间传递；
3. 允许用户动态创建主题，零运维成本；
4. 把传输量降到最低以提高传输效率；
5. 把低带宽、高延迟、不稳定的网络等因素考虑在内；
6. 支持连续的回话控制；
7. 理解客户端计算能力可能很低；
8. 提供服务质量管理；
9. 假设数据不可知，不强求传输数据的类型和格式，保持灵活性。

特性

MQTT 协议工作在低带宽、不可靠的网络的远程传感器和控制设备通讯二设计的协议，具有一下几项特性：

1. 使用发布/订阅消息模式，提供一对多的消息发布，接触应用程序的耦合；
2. 对负载内容屏蔽的消息传输；
3. 使用TCP/IP提供网络链接；
4. 三种消息发布服务质量：
    1. 至多一次：

       消息发布完全依赖底层TCP/IP网络。会发生消息丢失或重复。这一级别可用于环境传感器数据，丢失一次读记录无所谓，因为不就后还会有第二次发送。这一种方式主要普通APP的推送，倘若你的智能设备在消息推送时，未联网，推送过去没收到，再次联网也就收不到了。
    2. 至少一次

确保消息到达，但消息重复可能会发生

3. 只有一次

   确保消息到达一次，在一些要求比较严格的计费系统中，可以使用次级别。在计费系统中，消息重复或丢失会导致不确定的结果。这种最高质量的消息发布服务还可以用于即时通讯类的APP推送，确保用户收到且只会收到一次。

5. 小型传输，开销很小（固定长度的头部是2字节），协议交换最小化，以降低网络流量

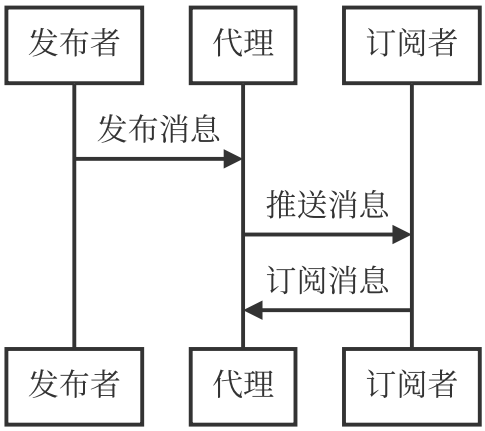
   这就是为什么非常适合在物联网（IoT）领域，传感器与服务器通信，信息的收集。嵌入式设备的运算能力和贷款相对薄弱，使用这种协议来传递消息非常适合。

6. 使用 `Last Will` 和 `Testament` 特性通知有关各方客户端异常中断的机制。

   1. Last Will 遗言机制，用于通知同意主题下的其他设备发送遗言的设备已经断开了链接；
   2. Testament 遗嘱机制，功能类似于 Last Will。

MQTT 协议原理

MQTT 协议实现方式



实现MQTT协议需要客户端和服务端通讯完成，在通讯过程中，MQTT协议中有是那种身份，发布者（public）、代理（broker）（服务器）、订阅者（Subscribe）。消息的发布者和订阅者都是客户端，消息代理是服务器，消息发布者可以同时是订阅者。

MQTT 传输的消息分为：主题（Topic）和负载（payload）

1. Topic 可以理解为消息的类型，订阅者订阅（Subscribe）后，就会收到该消息内容（payload）
2. payload 可以理解为消息的内容，是指订阅者具体要使用的内容

网络传输与应用消息

MQTT 会构建底层网络传输，将奖励客户端到服务器的链接，提供两者之间的一个有序的、无损的、基于字节流的双向传输。

当应用数据通过 MQTT 网络发送是， MQTT 会把与之相关的服务质量（QoS）和主题名（Topic）相关联。

MQTT 客户端

一个使用 MQTT 协议的应用程序或设备，总是建立到服务器的网络链接。客户端可以：

1. 发布其他客户端可能会订阅的消息；
2. 订阅其他客户端发布的消息；
3. 退订或删除应用程序消息；
4. 断开于服务器链接；

MQTT服务器

MQTT 服务器以称为"消息代理"（Broker）可以是一个应用程序或一台设备，它是位于消息发布者和订阅者之间，可以：

1. 接受来自客户端的网络链接；
2. 接受客户端发布的应用消息；
3. 处理来自客户端的订阅和退订请求；
4. 向订阅的客户转发应用程序消息。

MQTT 协议中的订阅、主题、会话

1. 订阅（Subscription）

   订阅包含主题筛选器（Topic Filter）和最大服务质量（QoS）订阅会与一个会话（Session）关联。一个会话可以包含多个订阅。每一个会话中的每个订阅都有一个不同的主题筛选器。

2. 会话（Session）

   每个客户端与服务器建立链接后就是一个会话，客户端和服务器之间有状态交互。会话存在于一个网络之间，也可能在客户端和服务器之间跨越多个连续的网络链接。

3. 主题名（Topic Name）

   链接到一个应用程序的标签，该标签与服务器的订阅相匹配。服务器会将消息发送给订阅所匹配标签的客户端。

4. 主题筛选器（Topic Filter）

   一个对主题名通配符筛选器，在订阅表达式中使用，表示订阅所匹配到的多个主题。

5. 负载（payload）

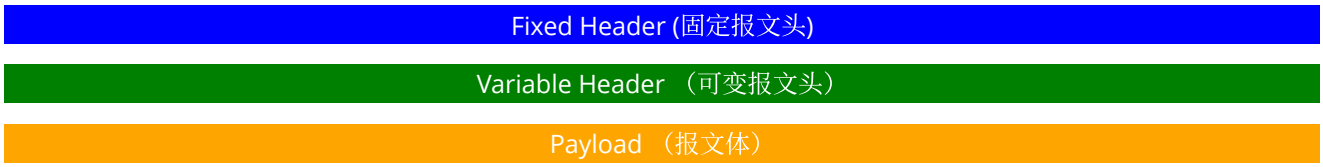
   消息订阅者所具体接受的内容。

MQTT 协议中的方法

MQTT 协议中定义了一些方法（也叫动作），用来表示对确定资源所进行操作。这个资源可以代表预先存在数据或动态生成数据，取决于服务器的实现。通常来说，资源指服务器上的文件或输出，主要方法有：

1. Connect： 等待与服务器建立链接
2. Disconnect：等待MQTT客户端完成所做的工作，并与服务器断开TCP/IP会话
3. Subscribe： 等待完成订阅
4. UnSubscribe： 等待服务器取消客户端的一个活多个 Topics 订阅
5. Publish： MQTT客户端发送消息请求，发送完成后返回应用程序线程

MQTT 协议数据包结构



1. 固定头（Fixed header）存在于所有MQTT数据包中，表示数据包类型及数据包的分组类标识
2. 可变头（Variable header）存在于部分MQTT数据包中，数据包类型决定了可变头是否存在及其具体内容
3. 消息体（Payload）存在于部分MQTT数据包中，表示客户端收到的具体内容。

MQTT 固定头

固定头存在于MQTT数据包中，结构如下

地址\bit	7	6	5	4	3	2	1	0
Byte 1	MQTT数据包类型				不同类型MQTT数据包的具体标识			
Byte 2---	剩余长度							

MQTT数据包类型

位置：Byte 1 中 bits 7 - 4。

相于一个4位的无符号值，类型、取值及描述如下：

名称	值	流方向	描述
Reserved	0	不可用	保留位
CONNECT	1	客户端到服务器	客户端请求链接到服务器
CONNACK	2	服务器都客户端	链接确认
PUBLISH	3	双向	发布消息
PUBACK	4	双向	发布确认
PUBREC	5	双向	发布收到（保证第1部分到达)
PUBREL	6	双贿	发布释放（保证第2部分到达）
PUBCOMP	7	双向	发布完成（保证第3部分到达）
SUBSCRIBE	8	客户端到服务器	客户端请求订阅
SUBACK	9	服务器到客户端	订阅确认
UNSUBSCRIBE	10	客户端到服务器	请求取消订阅
UNSUBACK	11	服务器到客户端	取消订阅确认
PINGEREQ	12	客户端到服务器	PING请求
PINGRESP	13	服务器到客户端	PING应答
DISCONNECT	14	客户端到服务器	中断连接
Reserved	15	不可用	保留位

标识位

位置：Byte 1 中 bits 3-0

在不使用标识位的消息类型中，标识位被作为保留位。如果收到无效的标志时，接收端必须关闭网络连接。

数据包	标识位	bit 3	bit 2	bit 1	bit 0
CONNECT	保留位	0	0	0	0
CONNACK	保留位	0	0	0	0
PUBLISH	MQTT 3.1.1 使用	DUP～1	QoS～2	QoS～2	retain～3
PUBACK	保留位	0	0	0	0
PUBREC	保留位	0	0	0	0
PUBCOMP	保留位	0	0	0	0
SUBSCRIBE	保留位	0	0	0	0
SUBBACK	保留位	0	0	0	0
UBSUBCRIBE	保留位	0	0	0	0
UNSUBBACK	保留位	0	0	0	0
PINGREQ	保留位	0	0	0	0
PINGRESP	保留位	0	0	0	0
DISCONNECT	保留位	0	0	0	0

1. DUP 发布消息的副本。用来保证消息的可靠传出，如果设置为1，则在下面的辩题中增加 `MessageID` ，并且需要回复确认，以保证消息传输完成，但不能用于检测消息重复发送。
2. QoS 发布消息的服务质量， 即：保证消息传递的次数
    1. Ø00：最多一次，即：<=1
    2. Ø01：至少一次，即：>=1
    3. Ø10： 一次，即：=1
    4. Ø11：预留
3. RETAIN：发布保留标识，表示服务器要保留这次推送的消息，如果有新的订阅者出现，就把这消息推送给他，如果没有呢么推送至当前订阅者后释放。

剩余长度（Remaining Length）

地址： Byte2

固定头的第二字节用来保存变长头部和消息体的总大小，但不是直接保存的。这一字节是可扩展，其保存机制，前 7 位用于保存长度，后一部分做标识。当最后一位为1时，表示长度不足，需要使用二个字节继续保存。例如：计算出后面的大小为0.

MQTT 可变头

MQTT 数据包中包含一个可变头，它驻位于固定的头和负载之间。可变头的内容因数据包类型而不同，较常的应用是作为包的标识：

bit	7	6	5	4	3	2	1	0	存在的报文类型
1~8 byte	Protocol name(协议名）使用utf8数据流格式，固定字符串MQLSDP								CONNECT
1 byte	Protocol Version(版本号）目前固定版本号 3								CONNECT
1 byte	Connect flags （链接标记）								CONNECT
	User Name Flage	Password Flag	Will Retain	WillQos		Will Flag	Clean Session	Reserved	
2 byte	Keep Alive timer(心跳时长)								CONNECT
1 byte	Connect return code （链接返回码)								CONNECT
3~32767 byte	Topic name（主题名称）								PUBLISH
8 byte	Message ID （消息ID）								PUBLISH（QoS>0), PUBACK, PUBREC,PUBREL, PUBCOMP,SUBSCRIBE, SUBACK,UNSUBSCRIBE, UNSUBACK

很多类型数据包中都包含一个2字节数据包标识字段，这些包括有：
PUBLISH（QoS>0),PUBACK,PUBREC,PUBREL,PUBCOMP,SUBSCRIBE,SUBACK,UNSUBSCRIBE,UNSUBACK

PayLoad 消息体

bit	7	6	5	4	3	2	1	0	存在的报文类型
3~25 byte	Client ID(客户端位移标识）使用utf8数据流格式，长度1~23个字符								CONNECT
3~32767 byte	Will Topic 使用utf 8 数据流格式								CONNECT， 并设置了 will flag
? byte	Will Message 使用 UTF8 数据流格式								CONNECT，并且设置了 Will flag
3~32767 byte	User Name （用户名）								CONNECT，并且设置了 User Name flag
3~32767 byte	Password （密码）								CONNECT，并且设置了 Password Flag
3~0x7F - 2 byte	Subscribe Payload (订阅体）								SUBSCRIBE
	Subscribe Ack QoS(订阅回执 QoS）								SUBACK
	UnSubscribe Topics（取消订阅主题）								UNSUBSCRIBE
	Data（数据）								PUBLISH

payload 消息体位 MQTT数据包的第三部分，包含 CONNECT，SUBSCRIBE，SUBACK，UNSUBSCRIBE 四种消息类型：

1. CONNECT：消息体内容主要是：客户端的 ClientID，订阅的Topic，Message 以及用户名和密码
2. SUBSCIRBE：消息体内容是一系列的要订阅的主题以及 QoS
3. SUBACK：消息体内容是服务器对于 SUBSCRIBE所申请的主题及QoS进行确认和回复
4. UBSUBSCRIBE：消息体内容是要订阅的主题。