



MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输协议), 是一种基于发布/订阅 (publish/subscribe) 模式的"轻量级"通讯协议, 该协议构建于TCP/IP协议上, 由IBM在1999年发布。

MQTT最大优点在于, 可以以极少的代码和有限的带宽, 为连接远程设备提供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, 使其在物联网、小型设备、移动应用等方面有较广泛的应用。



MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT协议是轻量、简单、开放和易于实现的, 这些特点使它适用范围非常广泛。在很多情况下, 包括受限的环境中, 如: 机器与机器 (M2M) 通信和物联网 (IoT)。其在, 通过卫星链路通信传感器、偶尔拨号的医疗设备、智能家居、及一些小型化设备中已广泛使用。

## MQTT主要特性

按 Esc 即可退出全屏模式

### (1) 使用发布/订阅消息模式, 提供一对多的消息发布, 解除应用程序耦合。

这一点很类似于XMPP, 但是MQTT的信息冗余远小于XMPP, 因为XMPP使用XML格式文本来传递数据。

### (2) 对负载内容屏蔽的消息传输。

### (3) 使用TCP/IP提供网络连接。

主流的MQTT是基于TCP连接进行数据推送的, 但是同样有基于UDP的版本, 叫做MQTT-SN。这两种版本由于基于不同的连接方式, 优缺点自然也就各有不同了。

### (4) 有三种消息发布服务质量:

- "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。
- "至少一次", 确保消息到达, 但消息重复可能会发生。
- "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, 可以使用此级别。

### (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。

这就是为什么在介绍里说它非常适合"在物联网领域, 传感器与服务器的通信, 信息的收集", 要知道嵌入式设备的运算能力和带宽都相对薄弱, 使用这种协议来传递消息再适合不过了。

### (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。

Last Will: 即遗言机制, 用于通知同一主题下的其他设备发送遗言的设备已经断开了连接。

Testament: 遗嘱机制, 功能类似于Last Will。

## MQTT:

0	1	2	3	4	5	6	7
Message Type				UDP	QoS Level		Retain
Remaining Length (1~4 bytes)							
Variable Length Header (Optional)							
Variable Length Message Payload (Optional)							

### MQTT关键词

按 Esc 即可退出全屏模式



#### • 一、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。

#### • 二、会话 (Session)

每个客户端与服务器建立连接后就是一个会话，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连接。

#### • 三、主题名 (Topic Name)

连接到一个应用程序消息的标签，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每个客户端。

#### • 四、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。

#### • 五、负载 (Payload)

消息订阅者所具体接收的内容。

8:09 / 37:03

1x

自动