

第三方设备接入协议规范

文档修订记录

日期	版本号	描述	著者	审阅者
2022/05/11	V1.0	文档创建	李军辉	
2024/03/18	V1.1	增加 MQTT 说明	李军辉	

1. MQTT 介绍

MQTT 是一个轻量级协议，基于 TCP/IP 协议的发布（Publish）/订阅（Subscribe）消息转发模式，在物联网应用中大规模使用。MQTT 协议的中心是 broker（服务器/代理），客户端通过订阅消息和发布消息进行数据交互

2. MQTT 说明

要了解 MQTT 的工作原理，首先需要掌握以下几个概念：MQTT 客户端、MQTT Broker、发布-订阅模式、主题、QoS。

MQTT 客户端

任何运行 MQTT 客户端库的应用或设备都是 MQTT 客户端。例如，使用 MQTT 的即时通讯应用是客户端，使用 MQTT 上报数据的各种传感器是客户端，各种 MQTT 测试工具也是客户端。

MQTT Broker

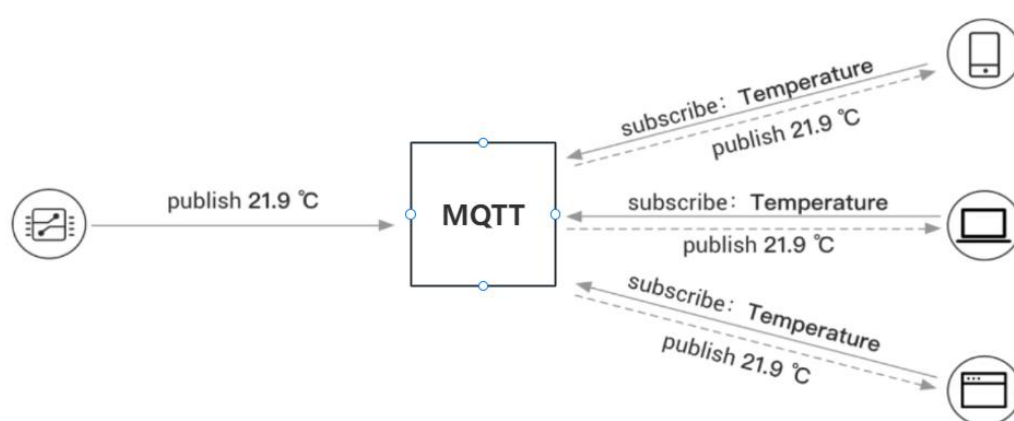
MQTT Broker 是负责处理客户端请求的关键组件，包括建立连接、断开连接、订阅和取消订阅等操作，同时还负责消息的转发。一个高效强大的 MQTT Broker 能够轻松应对海量连接和百万级消息吞吐量，从而帮助物联网服务提供商专注于业务发展，快速构建可靠的 MQTT 应用。

发布-订阅模式

发布-订阅模式与客户端-服务器模式的不同之处在于，它将发送消息的客户端（发布者）和接收消息的客户端（订阅者）进行了解

耦。发布者和订阅者之间无需建立直接连接，而是通过 MQTT Broker 来负责消息的路由和分发。

下图展示了 MQTT 发布/订阅过程。温度传感器作为客户端连接到 MQTT Broker，并通过发布操作将温度数据发布到一个特定主题（例如 Temperature）。MQTT Broker 接收到该消息后会负责将其转发给订阅了相应主题（Temperature）的订阅者客户端。



主题

MQTT 协议根据主题来转发消息。主题通过 / 来区分层级，类似于 URL 路径，例如：

```
chat/room/1
sensor/10/temperature
sensor+/temperature
```

MQTT 主题支持以下两种通配符：+ 和 #。

- +：表示单层通配符，例如 a/+ 匹配 a/x 或 a/y。
- #：表示多层通配符，例如 a/# 匹配 a/x、a/b/c/d。

注意：通配符主题只能用于订阅，不能用于发布。

QoS

MQTT 提供了三种服务质量（QoS），在不同网络环境下保证消息的可靠性。

- QoS 0：消息最多传送一次。如果当前客户端不可用，它将丢失这条消息。
- QoS 1：消息至少传送一次。
- QoS 2：消息只传送一次。

3. MQTT 的工作流程

客户端使用 TCP/IP 协议与 Broker 建立连接，可以选择使用 TLS/SSL 加密来实现安全通信。客户端提供认证信息，并指定会话类型（Clean Session 或 Persistent Session）。

客户端既可以向特定主题发布消息，也可以订阅主题以接收消息。当客户端发布消息时，它会将消息发送给 MQTT Broker；而当客户端订阅消息时，它会接收与订阅主题相关的消息。

MQTT Broker 接收发布的消息，并将这些消息转发给订阅了对应主题的客户。它根据 QoS 等级确保消息可靠传递，并根据会话类型为断开连接的客户端存储消息。

4. MQTT 协议中的方法

MQTT 协议中定义了一些方法（也被称为动作），表示对确定资源进行操作。资源指服务器上的文件或输出。主要方法有：

- (1) Connect。等待与服务器建立连接。
- (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或多个 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 协议通过交换预定义的 MQTT 控制报文来通信。

一个 mqtt 数据包结构包括：固定头部，可变头部，消息体 (payload) 三部分组成，mqtt 数据包结构如下：

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。只有 CONNECT、SUBSCRIBE、SUBACK、UNSUBSCRIBE 四种类型的消息有 Payload 消息体：

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。

SUBSCRIBE，消息体内容是一系列的要订阅的主题以及 QoS

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。

UNSUBSCRIBE，消息体内容是要取消订阅的主题。

控制报文的类型

名字	值	报文流动方向	描述
CONNECT	1	客户端到服务端	客户端请求连接服务端
CONNACK	2	服务端到客户端	连接报文确认
PUBLISH	3	两个方向都允许	发布消息
PUBACK	4	两个方向都允许	QoS 1 消息发布收到确认
PUBREC	5	两个方向都允许	发布收到 (保证交付第一步)
PUBREL	6	两个方向都允许	发布释放 (保证交付第二步)
PUBCOMP	7	两个方向都允许	QoS 2 消息发布完成 (保证交互第三步)
SUBSCRIBE	8	客户端到服务端	客户端订阅请求
SUBACK	9	服务端到客户端	订阅请求报文确认
UNSUBSCRIBE	10	客户端到服务端	客户端取消订阅请求
UNSUBACK	11	服务端到客户端	取消订阅报文确认
PINGREQ	12	客户端到服务端	心跳请求
PINGRESP	13	服务端到客户端	心跳响应
DISCONNECT	14	客户端到服务端	客户端断开连接

5. 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 协议中有三种身份：发布者 (Publish)、代理 (Broker) (服务器)、订阅者 (Subscribe)。其中，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) 两部分：

1) **Topic**, 可以理解为消息的类型, 订阅者订阅 (Subscribe) 后, 就会收到该主题的消息内容 (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 (限制大小)。

MQTT 消息发送与接收的流程:

ClientA 链接到 Broker;

ClientB 链接到 Broker, 并订阅主题 Topic1;

ClientA 发送给 Broker 一条消息, 主题为 Topic1;

Broker 收到 ClientA 的消息, 发现 ClientB 订阅了 Topic1, 而后将消息转发到 ClientB;

ClientB 从 Broker 接收到该消息。

6. MQTT 网关数据接入

物联终端设备和平台互联的数据互联皆是通过 MQTT 协议来完成的, 通过它们之间的配合, 可以实现

- **data**, 意寓数据的意思, 用于数据推送。
- **cmd**, 意寓指令的意思, 用于数据下发。
- **result**, 意寓执行成功与否返回的结果。

数据推送说明

假设现有一台网关设备，终端编号为“1000”，设备品牌型号为“HMISG”，进行数据推送网云平台/设备端，组包含两个变量，

通道标识	类型	当前值
Y0	开关型	1
V0	整型	99

网关设备会发布如下数据，主题为“data/1000/HMISG/1000”，内容格式是 json，{'变量': '值'}的形式推送数据

```
{  
  "Y0" : "1",  
  "V0" : "99"  
}
```

远程写入说明

当变量需要被改动，需要下发数据修改的消息。举个例子，现有一个 MQTT 工程，终端编号为“1000”，设备品牌型号为“HMISG”。有一个变量的值被改变，该变量信息如下，

通道标识	类型	新值
V0	整型	99

如果要远程写入变量的值，操作端需要向 MQTT 服务器发布的主题为“cmd/1000/HMISG/1000”的消息，内容为：

数据格式:

```
{  
  "type": "set_var",  
  "payload": {  
    "V0": "99"  
  }  
}
```

数据格式为 json。“set_var”表示修改变量，“payload”内是要修改的变量。“V0”通道内修改变量名，99 是需要修改的值。