MQTT（Message Queuing Telemetry Transport）是一种专为解决物联网（IoT）应用中的通信问题而设计的协议。下面是有关MQTT的体系结构、工作原理、相关概念术语、核心协议结构以及通信过程的详细信息：

解决的问题：

MQTT旨在解决物联网应用中的通信需求，其中设备可能存在低带宽、高延迟、不稳定网络连接等特点。MQTT的设计目标是提供一种轻量级、可靠、高效的通信协议，以便物联网设备可以进行可靠的消息传递，同时尽量减少网络和设备资源的占用。

体系结构：

MQTT采用了发布-订阅模型的体系结构。在这个模型中，有一个MQTT代理（Broker）作为中介，接收设备发布的消息并将其传递给订阅了相应主题的设备。设备可以是消息的发布者（Publisher）、订阅者（Subscriber）或兼具两者角色。

工作原理：

发布者将消息发布到特定的主题（Topic）。订阅者通过订阅感兴趣的主题来表达对消息的兴趣。MQTT代理接收发布者发送的消息，并将其传递给订阅了相应主题的订阅者。消息可以根据设备和应用程序的需要选择不同的消息质量等级（QoS）。设备之间通过建立到MQTT代理的TCP连接进行通信。

相关概念术语：

客户端（Client）：参与MQTT协议的设备或应用程序。

代理（Broker）：MQTT消息传递的核心组件，接收发布者的消息并将其传递给订阅者。

主题（Topic）：消息的标识符，用于将消息分类和传递给订阅者。

QoS（Quality of Service）：消息质量等级，用于控制消息传递的可靠性和效率。

会话（Session）：客户端与代理之间的连接状态，用于保持订阅和发布状态。

保留消息（Retained Message）：发布者可以选择将保留消息发送到特定主题，新的订阅者连接时会立即接收到该消息。

核心层的协议结构：

MQTT协议由三个核心层组成：

TCP/IP传输层：MQTT使用基于TCP的可靠连接进行通信。

MQTT控制包：协议定义了一系列的控制包，用于在设备和代理之间传递消息、状态和控制信息。

MQTT应用层：定义了MQTT消息的格式、主题的结构和消息质量等级。

通信过程：

设备建立到MQTT代理的TCP连接。

设备发送CONNECT控制包以建立会话。

设备可以选择订阅一个或多个主题，发送SUBSCRIBE控制包。

发布者发送PUBLISH控制包将消息发布到指定主题。

代理接收到发布的消息后，将其传递给订阅了相应主题的订阅者。

订阅者接收到消息后，可以发送ACKNOWLEDGE控制包确认消息的接收。

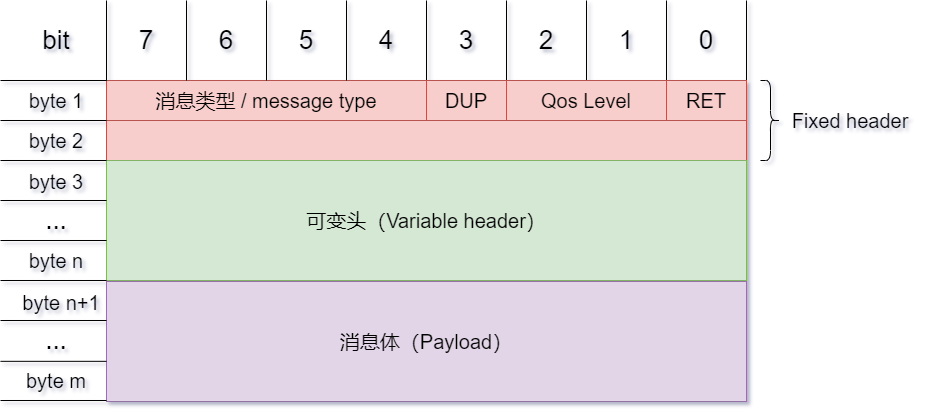
设备可以随时发送DISCONNECT控制包断开与代理的连接。

MQTT数据包结构

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；

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；

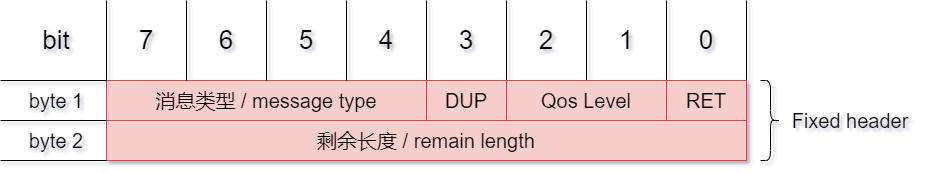
消息体（Payload），存在于部分MQTT数据包中，表示客户端收到的具体内容；



固定头：MQTT消息类型 / message type

\*\*位置：\*\*byte 1, bits 7-4。

4位的无符号值



标识位 / DUP

\*\*位置：\*\*byte 1, 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

DUP：发布消息的副本。用来在保证消息的可靠传输，如果设置为 1，则在下面的变长中增加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。

QoS发布消息的服务质量（前面已经做过介绍），即：保证消息传递的次数

00：最多一次，即：<=1

01：至少一次，即：>=1

10：一次，即：=1

11：预留

RETAIN：发布保留标识，表示服务器要保留这次推送的信息，如果有新的订阅者出现，就把这消息推送给它，如果设有那么推送至当前订阅者后释放。

剩余长度（Remaining Length）

位置：byte 1。

固定头的第二字节用来保存变长头部和消息体的总大小的，但不是直接保存的。这一字节是可以扩展，其保存机制，前7位用于保存长度，后一部用做标识。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 / Variable header**

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，较常的应用是做为包的标识

#### Payload消息体

Payload消息体是MQTT数据包的第三部分，CONNECT、SUBSCRIBE、SUBACK、UNSUBSCRIBE四种类型的消息 有消息体：

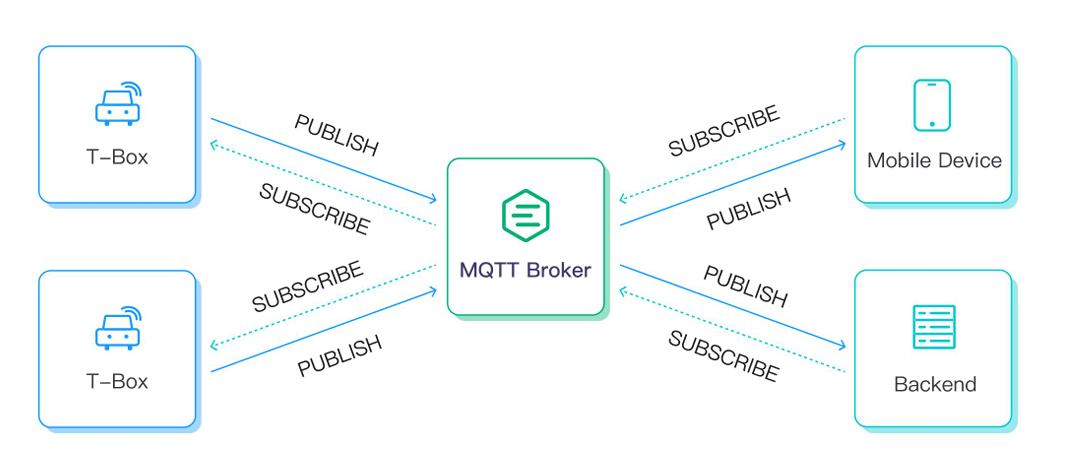
CONNECT，消息体内容主要是：客户端的ClientID、订阅的Topic、Message以及用户名和密码

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。

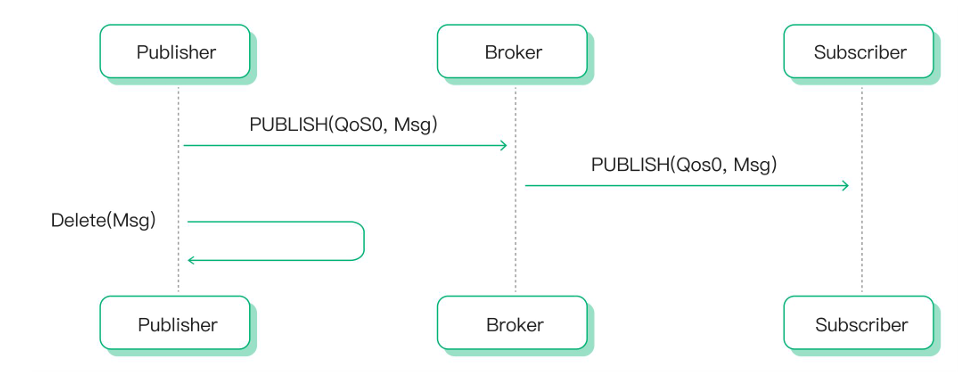
UNSUBSCRIBE，消息体内容是要订阅的主题。

MQTT



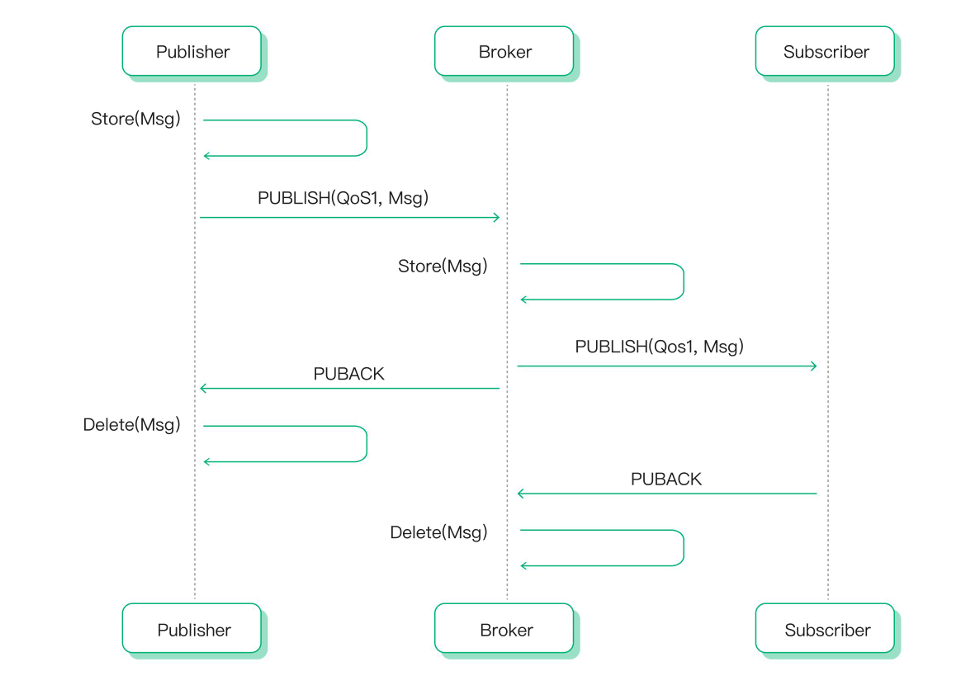
**QoS  0**

消息最多传递一次，如果当时客户端不可用，则会丢失该消息。Sender(可能是 Publisher 或 者 Broker) 发送一条消息之后，就不再关心它有没有发送到对方，也不设置任何重发机制。



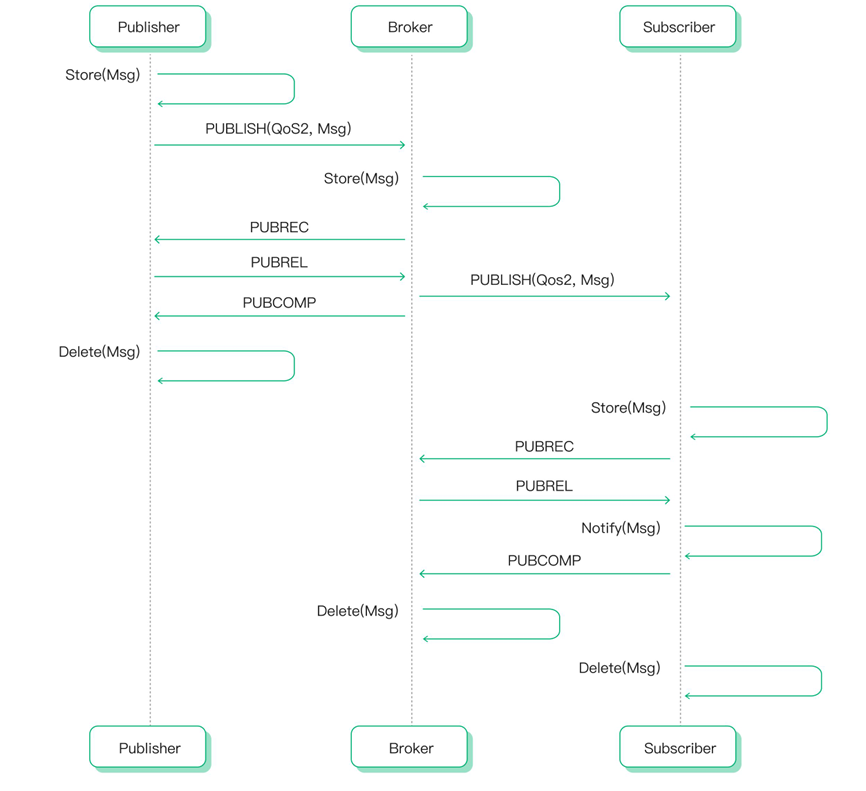
**QoS  1**

消息传递至少 1 次。包含了简单的重发机制，Sender 发送消息之后等待接收者的 ACK，如 果没收到 ACK 则重新发送消息。这种模式能保证消息至少能到达一次，但无法保证消息重复。



**QoS  2**

消息仅传送一次。设计了重发和重复消息发现机制，保证消息到达对方并且严格只到达一次。



车联网场景中的消息QoS设计首先需要明确的是QoS级别越高，消息交互越复杂，系统资源消耗越大，所以QoS等级不是设置的越高越好。应用程序可以根据自己的网络场景和业务需求，选择合适的QoS级别。

根据车联网信息服务相关数据的属性和特征，我们可以将其分为六类：基础属性类数据、车辆工控类数据、环境感知类数据、车控类数据、应用服务类数据和用户个人信息。

那么在不同的车联网场景中如何选择 MQTT QoS 等级呢？

以下情况下可以选择 QoS0

可以接受消息偶尔丢失的场景下可以选择 QoS0。车联网提供的与娱乐相关的多媒体服务，如天气预报等数据等。还有部分涉车服务类数据，如 车辆历史行车数据的上报、历史行车操作数据等

以下情况下可以选择 QoS1

车联网大部分场景都是选用 QoS1，它实现了系统资源性能和消息实时性、可靠性最优化。 QoS1 广泛运用于控车消息、行车上报数据（含新能源国标和企标）、交通安全管控类数据，和交通安全、道路安全相关的预警数据

以下情况下可以选择 QoS2

对于不能忍受消息丢失，且不希望收到重复的消息，数据完整性与及时性要求较高的场景，可 以选择 QoS2。 车联网场景中 QoS2 的应用并不多，虽然其可以增加消息可靠性，但同时也使资源消耗和消 息时延大幅增加。 QoS2 主要运用于对数据完整性与及时性要求较高的银行、消防、航空等行业，有些主机厂 的行车告警和车辆充电桩计费费单消息会选择采用 QoS2。

车联网场景中的MQTT协议

MQTT协议适合车联网吗？

整个车联网业务架构复杂，涉及多个通信环节，我们讨论的是车联网平台主要负责的云-端消息接入模块。

MQTT 是基于发布/订阅模式的物联网通信协议，具有简单易实现、支持 QoS、报文紧凑等特点，占据了物联网协议的半壁江山。

在车联网场景中，MQTT 依然能够胜任海量车机系统灵活、快速、安全接入，并保证复杂网络环境下消息实时性、可靠性，其主要应用优势如下：

1. 开放消息协议，简单易实现。市场上有大量成熟的软件库与硬件模组，可以有效降低车机 接入难度和使用成本；

2. 提供灵活的发布订阅和主题设计，能够通过海量的 Topic 进行消息通信，应对各类车联网 业务；

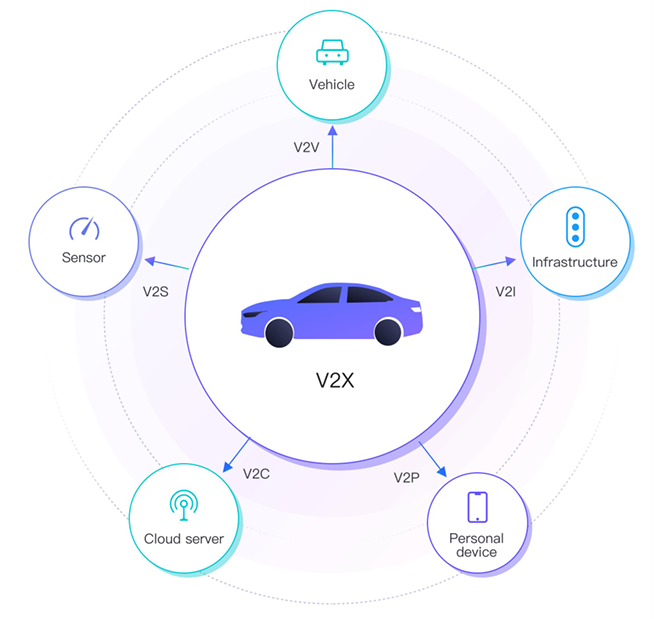
3. Payload 格式灵活，报文结构紧凑，可以灵活承载各类业务数据并有效减少车机网络流量；

4. 提供三个可选的 QoS 等级，能够适应车机设备不同的网络环境；

5. 提供在线状态感知与会话保持能力，方便管理车机在线状态并进行离线消息保留。

综上，如果配以具备海量车端连接、软实时、高并发数据吞吐以及多重安全保障能力的消息中 间件产品，MQTT 协议无疑将为车联网平台的搭建带来极大便利





车联网（vehicle-to-everything，V2X）是指车与云、车与网、车与车、车与路、 车与人、车与传感设备等交互，实现车辆与公众网络通信的动态移动通信系统，是为了满足与 车有关的每一个环节中的效率、安全、管理等元素而建立起的异构通信网络。而运行于其中的 通信协议就成为车联网系统建设的关键和核心。

两种主流的通信技术，对车联网整体发展起到了推动作用：

DSRC

C-V2X

