注：MQTT协议3.0 PDF下载连接　参考视频连接　ｏｎｅＮＥＴ云

一、理解MQTT报文结构

什么是报文？报文(message)是网络中交换与传输的数据单元，即站点一次性要发送的数据块。报文包含了将要发送的完整的数据信息，其长短很不一致，长度不限且可变。简单来理解类似两个人在说话，报文就是两个人说话的内容，而这两个人就是服务器和客户端。

在MQTT中一共只有16种报文，如连接服务器报文，订阅报文等等。报文有自己的结构，就像我们的语言在组成上可以分为主语谓语宾语，而报文结构分为：固定报头，可变报头和有效载荷三部分。这三部分组成了一条报文的基本结构，其中固定包头是每一条报文中必不可少的，可变报头和有效载荷在有的报文里没有。这些的用法和使用规范都可以在MQTT协议的官方文档中找到。

> Fixed header 固定报头，所有控制报文都包含
>
> Variable header 可变报头，部分控制报文包含
>
> Payload 有效载荷，部分控制报文包含

报文	描述	流向	值	固定报头	可变报头	负载
CONNECT	客户端请求与服务端建立连接	C->S	1	有	有	有
CONNACK	服务端确认连接建立	S->C	2	有	有	有
PUBLISH	发布消息	C⇔S	3	有	有	有
PUBACK	收到发布消息确认（QoS1 等级）	C⇔S	4	有	有	无
PUBREC	发布消息收到（QoS2 等级）	C⇔S	5	有	有	无
PUBREL	发布消息释放（QoS2 等级）	C⇔S	6	有	有	无
PUBCOMP	发布消息完成（QoS2 等级）	C⇔S	7	有	有	无
SUBSCRIBE	订阅请求	C->S	8	有	有	有
SUBACK	订阅确认	S->C	9	有	有	有
UNSUBSCRIBE	取消订阅	C->S	10	有	有	有
UNSUBACK	取消订阅确认	S->C	11	有	有	无
PING	客户端发送 PING(连接保活)命令	C->S	12	有	无	无
PINGRSP	PING 命令回复	S->C	13	有	无	无
DISCONNECT	断开连接	C->S	14	有	无	无

1、固定报头

简单来理解报文的固定报头部分：每个MQTT控制报文都包含一个固定报头。根据协议书里面的要求，报头一共需要2个字节共16bit二进制数，如下：

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文的类型				用于指定控制报文类型的标志位			
byte 2...	剩余长度							

可以看成第一个字节是MQTT报文控制类型和标志位，第二个BIT是此条报文的总长度。而报文控制类型和标志位都可以通过下面两个表格进行查询。需要用到什么报文就使用哪个报文类型和标志位放在一起组成第一个8BIT数据也就是该条报文的第一个字节数据。

表格 **2.1 -**控制报文的类型

名字	值	报文流动方向	描述
Reserved	0	禁止	保留
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
Reserved	15	禁止	保留

表格 2.2 - 标志位 Flag Bits

控制报文	固定报头标志	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP[1]	QoS[2]	QoS[2]	RETAIN[3]
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0

这里举一个例子，比如想连接服务器，那么需要发送CONNECT报文，去上表种中查询可以找到CONNECT报文的报文类型是1，1转换成16进制还是1，标志位是二进制数0000，转换成16进制还是0，那么本条报文的第一个bit数据就是10。剩余长度需要这条报文的全部长度都确定下来才能确定，那么CONNECT报文的固定报头部分就已经基本去确定下来了：10 ?? 问号是目前不能确定的1字节的报文剩余长度的值。

报文的剩余长度从第2个字节开始。剩余长度（Remaining Length）表示当前报文剩余部分的字节数，包括可变报头和负载的数 据。剩余长度不包括用于编码剩余长度字段本身的字节数。

> 例如，十进制数64会被编码为一个字节，数值是64，十六进制表示为0x40,。十进制数字321(=65+2*128)被编码为两个字节，最低有效位在前。第一个字节是 65+128=193。注 意最高位为1表示后面至少还有一个字节。第二个字节是2。

2、可变报头

某些MQTT控制报文包含一个可变报头部分。它在固定报头和负载之间。可变报头的内容根据 报文类型的不同而不同。可变报头还包含了使用的协议的使用信息，用16进制数据来表示。可变报头的长度不固定，报文可变报头的报文标识符（Packet Identifier）字段存在于在多个类型 的报文里。

规定报文标识符的第一部分是报文标识符的报文标志位，用来区一整条报文的结构，服务器根据标志位来确定报文的该位置是报文的可变报头部分。可变报头标识符可以参考下面的表格来确定：

图例 2.3 -报文标识符字节 Packet Identifier bytes

Bit	7 - 0
byte 1	报文标识符 MSB
byte 2	报文标识符 LSB

具体在使用过程中还需要增加可变报头的长度值和报文协议标准相关的信息等等。

3、有效负载

某些MQTT控制报文在报文的最后部分包含一个有效载荷，对于 PUBLISH来说有效载荷就是应用消息。有效负载的具体内容需要根据不同的报文来确定。

二、实际操作16条报文

1、CONNECT——连接服务器

想要自己的终端设别连接到服务器，需要先和服务器建立TCP连接，在此基础上传输MQTT协议才是有效的。客户端到服务端的网络连接建立后，客户端发送给服务端的第一个报文必须是CONNECT报文。在一个网络连接上，**客户端只能发送一次CONNECT报文。**服务端必须将客户端发送的第二 个CONNECT报文当作协议违规处理并断开客户端的连接。

CONNECT报文的固定包头和可变报头都是确定的，有效载荷包含一个或多个编码的字段。包括客户端的唯一标识符，Will主题，Will消息，用户 名和密码。除了客户端标识之外，其它的字段都是可选的，基于标志位来决定可变报头中是 否需要包含这些字段。

a)固定包头：

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT报文类型 (1)				Reserved 保留位			
	0	0	0	1	0	0	0	0
byte 2...	剩余长度							

第一个字节可以到高四位位1 ，低四位是0，第二个字节保留长度待定，所以得到固定包头**10 ??**

b)可变报头：

CONNECT报文的可变报头按下列次序包含四个字段：协议名（Protocol Name），协议级别（Protocol Level），连接标志（Connect Flags）和保持连接（Keep Alive）。

协议名 Protocol Name

	说明	7	6	5	4	3	2	1	0
协议名									
byte 1	长度 MSB (0)	0	0	0	0	0	0	0	0
byte 2	长度 LSB (4)	0	0	0	0	0	1	0	0
byte 3	‘M’	0	1	0	0	1	1	0	1
byte 4	‘Q’	0	1	0	1	0	0	0	1
byte 5	‘T’	0	1	0	1	0	1	0	0
byte 6	‘T’	0	1	0	1	0	1	0	0

根据表格得到协议名的第一个字节是00，第二个字节是04，字符串”MQTT“转换成四个字节16进制码数据是 4D 51 54 54。最后得到协议名：00 04 4D 51 54 54

协议级别 Protocol Level

	说明	7	6	5	4	3	2	1	0
协议级别									
byte 7	Level(4)	0	0	0	0	0	1	0	0

协议级别就是协议的版本。客户端用8位的无符号值表示协议的修订版本。对于3.1.1版协议，协议级别字段的值是 4(0x04)。如果发现不支持的协议级别，服务端必须给发送一个返回码为0x01（不支持的协议 级别）的CONNACK报文响应CONNECT报文，然后断开客户端的连接。

根据表格，固定协议级别的数据就是1字节数据 04

连接标志 Connect Flags

连接标志字节包含一些用于指定MQTT连接行为的参数。它还指出有效载荷中的字段是否存在。

Bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
byte 8	X	X	X	X	X	X	X	0

这里需要根据自己的设别的相关信息比如账户密码，设备号等来确定表格中的X的值，比如第7个bit数，User Name Flag 用户名标志位，如果你需要在报文中增加你的登陆服务器时候的使用到的用户名，那么你需要把这个标志位设置为1。通常登录阿里云腾讯云等都需要账户密码，这里我需要登录移动的物联网云平台，因此我需要把用户标志位，密码标志位都设置为1。

第5432标志位分别设置了服务器和客户端连接过程中突发问题的相关设置，比如突然离线，服务器需不要保留原来要发送却没来得及发送的数据等。这里把这四个标志位全设置为0 。

第1位Clean Session，会话保留位。取定客户端离线后数据是否保留，这里不保留，设置为0。

最后得到连接标志的完整数据是11000000 也就是C2

保持连接 Keep Alive

保持连接（Keep　Alive）是一个以秒为单位的时间间隔，表示为一个16位的字，它是指在客户 端传输完成一个控制报文的时刻到发送下一个报文的时刻，两者之间允许空闲的最大时间间隔。客户端负责保证控制报文发送的时间间隔不超过保持连接的值。

简单来说它就表示多少秒，过了这个时间服务器再没有接收到客户端收到的消息就会把客户点从服务器断开连接，来节省资源。保持连接的值为零表示关闭保持连接功能。这意味着，服务端不需要因为客户端不活跃而断 开连接。注意：不管保持连接的值是多少，任何时候，只要服务端认为客户端是不活跃或无 响应的，可以断开客户端的连接。

图例 **3.5**保持连接字节

Bit	7	6	5	4	3	2	1	0
byte 9	保持连接 Keep Alive MSB							
byte 10	保持连接 Keep Alive LSB							

这里按照120秒为标准设置保持连接时间，120换成16进制就是78，因为120 不够大不够用到保持位的高字节，因此是00。最后可以得到这两个字节的数据：00 78

可变报头最终结果

以上就是CONNECT报文中可变报头所有的组成，把每一个字节的数据放在一起就是：

协议名 00 04 4D 51 54 54 + 协议级别 04 + 连接标志 C2 + 保持连接 00 78 —》 **00 04 4D 51 54 54 04 C2 00 78**

c) 有效载荷

在CONNECT报文中的有效负载可分为三部分，分别是：客户端标识符 用户名 密码。(在协议中其实还有遗嘱，遗留报文等组成部分，但是由于在连接标志中没有设置，所以这里就没有相对应的内容)

CONNECT报文的有效载荷（payload）包含一个或多个以长度为前缀的字段，**可变报头中的 标志决定是否包含这些字段。如果包含的话，必须按这个顺序出现：客户端标识符，遗嘱主 题，遗嘱消息，用户名，密码**

这里需要写设备的客户端标识符、用户名、密码这三项。这里以我在移动oneNET上申请了一个账户为例：

> 客户端标识符：　设备ID　　587239503　　35 38 37 32 33 39 35 30 33　　9个字节
>
> 用户名：　　产品D　323685　　33 32 33 36 38 35　　6个字节
>
> 密码：　　鉴权信息 tianxiaohua　　74 69 61 6E 78 69 61 6F 68 75 61　　11个字节

在书写有效负载的时候不但需要把上面这些信息按照规定顺序转换成16进制放在一起，还需要在每端数据前标记这段信息的字节长度，表示长度的数据需要两个自己。比如客户端标识符有9个字节，那么这段信息就需要在前面加上00 09

最终得到的有效负载：

00 09 35 38 37 32 33 39 35 30 33 00 06 33 32 33 36 38 35 00 0b 74 69 61 6E 78 69 61 6F 68 75 61

d) 得到的最终的CONNECT报文

固定包头+可变报头+有效负载：

```
10 ?? 00 04 4D 51 54 54 04 C2 00 78 00 09 35 38 37 32 33 39 35 30 33 00 06 33 32
33 36 38 35 00 0b 74 69 61 6E 78 69 61 6F 68 75 61
```

从整个过程就可以发现服务器是如何读取报文的。当接收到报文时，会先确定第一个字节的值，根据序号就是本条报文是用来干什么的，是连接服务器还是发送推送等等。第二个字节和第一个字节都是固定报头的组成成分，第二个字节代表了整条报文的剩余长度，也就是可变报头和有效负载加在一起的总长度，在这里我的CONNECT报文的问号部分也就是剩余长度的是42，转换成16进制是2a。所以得到的最终的报文如下：

```
10 2a 00 04 4D 51 54 54 04 C2 00 78 00 09 35 38 37 32 33 39 35 30 33 00 06 33 32
33 36 38 35 00 0b 74 69 61 6E 78 69 61 6F 68 75 61
```

读取到固定包头后，服务器会按照顺序从可变报头的第一个字节读到最后一个字节，依次得到四个字段的信息。最后剩下的字段就是有效负载的部分。有效负载的每部分字段的前两个字节信息是该条字段信息的长度，而该字段代表的含义又可以在可变报头的连接标志中找到对应的部分。

我使用了网络调试助手来验证我的报文：

需要连接的服务器的地址为：

```
183.230.40.39:6002
```

使用TCP正常连接到oneNET云以后就可以通过发送CONNECT报文再和物联网服务器建立连接。

发送CINNECT报文以后服务器会返回个我们一个数据：20 02 01 00 表示已经成功连接。而这段数据正是CIONNACK报文，表示服务器和客户端正确连接，来到oneNET平台也可以看到设备已经上线。

2、CONNACK——服务端确认连接

a)固定报头

图例 3.8 – CONNACK 报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT报文类型 (2)				Reserved 保留位			
	0	0	1	0	0	0	0	0
byte 2...	剩余长度 (2)							
	0	0	0	0	0	0	1	0

报文的固定包头全都确定了：20 02

b）可变报头

	描述	7	6	5	4	3	2	1	0
连接确认标志		Reserved 保留位							SP[1]
byte 1		0	0	0	0	0	0	0	X
连接返回码									
byte 2		X	X	X	X	X	X	X	X

SP1根据CONNECT报文返回确认是01

表格 **3.1** –连接返回码的值

值	返回码响应	描述
0	0x00连接已接受	连接已被服务端接受
1	0x01连接已拒绝，不支持的协议版本	服务端不支持客户端请求的MQTT协议级别
2	0x02连接已拒绝，不合格的客户端标识符	客户端标识符是正确的UTF-8编码，但服务端不允许使用
3	0x03连接已拒绝，服务端不可用	网络连接已建立，但MQTT服务不可用
4	0x04连接已拒绝，无效的用户名或密码	用户名或密码的数据格式无效
5	0x05连接已拒绝，未授权	客户端未被授权连接到此服务器
6-255		保留

最后两位可以根据上图确认，客户端没有问题一般就是00，表示服务器接收客户端的连接。

c）最终得到的CONNACK报文

因为CINNACK报文没有有效载荷，所以最后得到的报文就是20 02 01 00

3、DISCONNECT –断开连接

a）固定报头

3.14.1 固定报头

图例 3.35 – DISCONNECT报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (14)				保留位			
	1	1	1	0	0	0	0	0
byte 2	剩余长度 (0)							
	0	0	0	0	0	0	0	0

可以确定报头为：E0 00

此报文没有可变报头和有效载荷，因此此报文的最终结果就是 E0 00

4、PINGREQ – 心跳请求

在CONNECT报文中我设定的连接时时长是200秒。如果客户端和服务器之间超过200秒没有发送任何数据，服务器就会自动断块和客户端的连接。因此使用心跳请求来保持客户端和服务器之间的有效连接。

a）固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (12)				保留位			
	1	1	0	0	0	0	0	0
byte 2	剩余长度 (0)							
	0	0	0	0	0	0	0	0

根据上表可以确定本条报文的固定包头是 C0 00

没有可变报头和有效负载

5、PINGRESP – 心跳响应

当服务器收到客户端发过来的心跳请求，需要给服务器发送一个心跳响应报文来确认连接。

a）固定报头

图例 3.34 – PINGRESP报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (13)				保留位			
	1	1	0	1	0	0	0	0
byte 2	剩余长度 (0)							
	0	0	0	0	0	0	0	0

根据上表可以确认固定报头为：D0 00

同时PINGRESP没有可变报头和有效负载

6、SUBSCRIBE - 订阅主题

用来关注某一个设备，也就是建立和某一个客户端设别的通讯，当你订阅的客户端的报文有消息发送出来以后，服务器会根据你的订阅主题自动把这条消息发送给你。

a）固定包头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (8)				保留位			
	1	0	0	0	0	0	0	0
byte 2	剩余长度							

可以得到固定报头： 80 ??

b)可变报头

	描述	7	6	5	4	3	2	1	0
报文标识符									
byte 1	报文标识符 MSB (0)	0	0	0	0	0	0	0	0
byte 2	报文标识符 LSB (10)	0	0	0	0	1	0	1	0

如上表，可变报文的前两个字节分别是00 0A 代表了这条报文的序号。加入需要发送十条主题报文，那么可以把报文的可变报头的报文报文标识符从第一篇报文开始标号一直标号到最后一片报文，也可以全都标号相同。使用官方给出了例子是这条订阅主题是第0A号也就是第10号报文

c）有效载荷

描述	7	6	5	4	3	2	1	0
主题过滤器								
byte 1	长度 MSB							
byte 2	长度 LSB							
byte 3..N	主题过滤器（Topic Filter）							
服务质量要求（Requested QoS）								
	保留位						服务质量等级	
byte N+1	0	0	0	0	0	0	X	X

可以看见有效载荷和前面的结构都是一样的，前两个字节还是有效载荷的长度，接下来是主题的内容，最后的一个字节是主题的服务质量，根据下表来确定自己想要什么样的服务质量：

表格 3.2 -服务质量定义

QoS 值	Bit 2	Bit 1	描述
0	0	0	最多分发一次
1	0	1	至少分发一次
2	1	0	只分发一次
-	1	1	保留位

d）最终得到的SUBSCRIBE报文

还是那我的oneNET设备举例。这里订阅一个由一个温度采集客户端发送的订阅消息temperature：

74 65 6D 70 65 72 61 74 75 72 65，长度是11 也就是00 0B，载加上服务等级，这里设置等级0或者1

因此得到最终的报文是

```
80 10 00 0A 00 0B 74 65 6D 70 65 72 61 74 75 72 65 00
```

7、SUBACK – 订阅确认

服务器根据客户端发来的订阅主题来回复客户端一个订阅确认代表已经订阅成功

a）固定报头

图例 3.24 – SUBACK报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (9)				保留位			
	1	0	0	1	0	0	0	0
byte 2	剩余长度							

根据表格得到固定包头为90 ??

b）可变报头

图例 3.25 – SUBACK报文可变报头

Bit	7	6	5	4	3	2	1	0
byte 1	报文标识符 MSB							
byte 2	报文标识符 LSB							

报文标识符是订阅报文时的报文标识符，根据订阅报文来确认。

c）有效载荷

Bit	7	6	5	4	3	2	1	0
	返回码							
byte 1	X	0	0	0	0	0	X	X

允许的返回码值：
0x00 - 最大QoS 0

0x01 - 成功 – 最大QoS 1

0x02 - 成功 – 最大 QoS 2

0x80 - Failure 失败

d）最终报文

90 03 00 ?? ??

8、UNSUBSCRIBE –取消订阅

a）固定报头

图例 3.28 – UNSUBSCRIBE报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (10)				保留位			
	1	0	1	0	0	0	1	0
byte 2	剩余长度							

A2 ??

b）可变报头

图例 3.29 – UNSUBSCRIBE报文可变报头

Bit	7	6	5	4	3	2	1	0
byte 1	报文标识符 MSB							
byte 2	报文标识符 LSB							

00 ??

c）有效载荷

报文的有效载荷包含客户端想要取消订阅的主题

d）最终报文

9、UNSUBACK – 取消订阅确认

a）固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (11)				保留位			
	1	0	1	1	0	0	0	0
byte 2	剩余长度 (2)							
	0	0	0	0	0	0	1	0

B0 02

b）可变报头

Bit	7	6	5	4	3	2	1	0
byte 1	报文标识符 MSB							
byte 2	报文标识符 LSB							

00 ??

c）最终报文

B0 02 00 ??

10 、PUBLISH – 发布消息

PUBLISH控制报文是指从客户端向服务端或者服务端向客户端传输一个应用消息

a）固定报头

图例 3.10 – PUBLISH报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT控制报文类型 (3)				DUP	QoS-H	QoS-	RETAIN
	0	0	1	1	X	X	X	X
byte 2...	剩余长度							

DUP位是根据订阅时确认的，如果是Qos0等级，DUP就是0；如果DUP标志被设置为0，表示这是客户端或服务端第一次请求发送这个PUBLISH报文。如 果DUP标志被设置为1，表示这可能是一个早前报文请求的重发。

Qos-是服务等级，这里设置为0；

RETAIN保留标志，如果客户端发给服务端的PUBLISH报文的保留（RETAIN）标志被设置为1，服务端必须存储 这个应用消息和它的服务质量等级（QoS），以便它可以被分发给未来的主题名匹配的订阅者。这里不需要保留，设置为0；

得到最后的固定包头就是 30 ??

b）可变报头

可变报头按顺序包含主题名和报文标识符。

主题名（Topic　Name）用于识别有效载荷数据应该被发布到哪一个信息通道。
主题名必须是PUBLISH报文可变报头的第一个字段。只有当QoS等级是1或2时，报文标识符（Packet Identifier）字段才能出现在PUBLISH报文 中。因为我们没有使用的是等级0，所以没有报文标识符。

	描述	7	6	5	4	3	2	1	0
Topic Name 主题名									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (3)	0	0	0	0	0	0	1	1
byte 3	‘a’ (0x61)	0	1	1	0	0	0	0	1
byte 4	‘/’ (0x2F)	0	0	1	0	1	1	1	1
byte 5	‘b’ (0x62)	0	1	1	0	0	0	1	0
报文标识符									
byte 6	报文标识符 MSB (0)	0	0	0	0	0	0	0	0
byte 7	报文标识符 LSB (10)	0	0	0	0	1	0	1	0

因此最后得到的可变报文只包括主题长度+主题名：

这里我设置自己的主题名代表温度temperature，转换成二进制就是：74 65 6D 70 65 72 61 74 75 72 65

可变报头：00 0B 74 65 6D 70 65 72 61 74 75 72 65

c）有效载荷

有效载荷包含将被发布的应用消息。数据的内容和格式是应用特定的。这里假设我发送的应用消息是25摄氏度，25转换成16进制是19

d）最终报文

30 0E 00 0B 74 65 6D 70 65 72 61 74 75 72 65 19

这里进行一个测试主题名：temperature

我再次从oneNET平台尽力一个温度采集传感器客户端：

温度采集客户端：

> 客户端标识符：设备ID 587275938 35 38 37 32 37 35 39 33 38 9个字节
>
> 用户名： 产品D 323685 33 32 33 36 38 35 6个字节
>
> 密码： 鉴权信息tianxiaohua2 74 69 61 6E 78 69 61 6F 68 75 61 32 12个字节

CONNECT 报文：连接服务器

```
10 2B 00 04 4D 51 54 54 04 C2 00 78 00 09 35 38 37 32 37 35 39 33 38 00 06 33 32
33 36 38 35 00 0C 74 69 61 6E 78 69 61 6F 68 75 61 32
```

SUBSCRIBE 报文：订阅主题

```
80 10 00 0A 00 0B 74 65 6D 70 65 72 61 74 75 72 65 00
```

e）实现过程

打开两个TCP窗口，一个是电脑客户端，一个是温度传感器客户端，分别让两个客户端连接到oneNET：

电脑客户端使用CONNECT报文连接到服务器：

```
10 2a 00 04 4D 51 54 54 04 C2 00 78 00 09 35 38 37 32 33 39 35 30 33 00 06 33 32
33 36 38 35 00 0b 74 69 61 6E 78 69 61 6F 68 75 61
```

温度传感器窗口使用CONNECT报文连接到服务器：

```
10 2B 00 04 4D 51 54 54 04 C2 00 78 00 09 35 38 37 32 37 35 39 33 38 00 06 33 32
33 36 38 35 00 0C 74 69 61 6E 78 69 61 6F 68 75 61 32
```

在电脑窗口服务器端口使用SUBSCRIBE报文订阅温度传感器发布的温度消息：

```
80 10 00 0A 00 0B 74 65 6D 70 65 72 61 74 75 72 65 00
```

在温度传感器窗口使用发布消息报文PUBLISH 发布温度值 的报文消息，温度是25 16进制是19

```
30 0E 00 0B 74 65 6D 70 65 72 61 74 75 72 65 19
```

可以看到最终电脑端接收到了温度信息。在这个过程中，由温度采集客户端发送出去的消息由客户端接收到并可以保存一段时间。同时通过电脑窗口服务器你可以通过订阅温度消息来接收温度值，服务器不但起到了存放消息的功能还实现了发送消息的功能。

f）改变发送等级

订阅主题就需要把等级变为等级一，改变最后一个字节为1：

```
80 10 00 0A 00 0B 74 65 6D 70 65 72 61 74 75 72 65 01
```

在固定报文的第一个字节把固定报头的等级部分改成等级1 ，固定包头就会变成 32 ?? 还需要在后面增加报文标识符，就是标记是第几号消息。这样在消息的接发工程中就可以确认自己发送的是哪一条消息以及接收到的是哪一条消息。并且根据等级，服务器还会返回给消息发送的客户端一个发布确认的报文。

如上的发布消息写成等级1的形式是：

```
32 10 00 0B 74 65 6D 70 65 72 61 74 75 72 65 00 01 19
```

在服务器发送了发布消息报文后，服务器就会返回给发送消息的客户端 40 02 00 01 也就是PUBACK

11、PUBACK –发布确认

a）固定报头

图例 3.12 - PUBACK报文固定报头

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT报文类型 (4)				保留位			
	0	1	0	0	0	0	0	0
byte 2...	剩余长度							
	0	0	0	0	0	0	1	0

40 02

b）可变报头

包含等待确认的PUBLISH报文的报文标识符。

图例 3.13 – PUBACK报文可变报头

Bit	7	6	5	4	3	2	1	0
byte 1	报文标识符 MSB							
byte 2	报文标识符 LSB							

根据发布消息的等级来确定，如果是等级1就是00 01

c）最终报文

40 02 00 01

12、PUBREC–发布收到（QoS2第一步）

把发布消息的等级设置为2，就会收到发布收到的报文信息：50 02 00 01 表示签收

13、PUBREL–发布释放（QoS2第二步）

把发布消息的等级设置为2，就会收到发布收到的报文信息：62 02 00 01 表示回复

14、PUBCOMP–发布完成（QoS2第三步）

把发布消息的等级设置为2，就会收到发布收到的报文信息：70 02 00 01 表示彻底完成