

OpenHarmony 华为云 IoT 应用开发--MQTT 协议开发



目 录

CONTENTS

01

什么是MQTT

02

MQTT接口分析

03

实例分析

01

什么是MQTT

MQTT(消息队列遥测传输)是一个基于客户端-服务器的消息发布/订阅传输协议，是ISO标准(ISO/IEC PRF 20922)下基于发布/订阅范式的消息协议。MQTT工作在TCP/IP协议栈上，是为硬件性能低下的远程设备以及网络状况不理想的情况下而设计的发布/订阅型消息协议。

MQTT协议是为大量计算能力有限，且工作在低带宽、不可靠的网络的远程传感器和控制设备通讯而设计的协议，它具有以下主要的几项特性：

- 1、使用发布/订阅消息模式，提供一对多的消息发布，解除应用程序耦合；
- 2、使用 TCP/IP 提供网络连接；
- 3、有三种消息发布服务质量；
- 4、小型传输，开销很小（只有固定长度的头部是 2 字节），协议交换最小化，以降低网络流量；

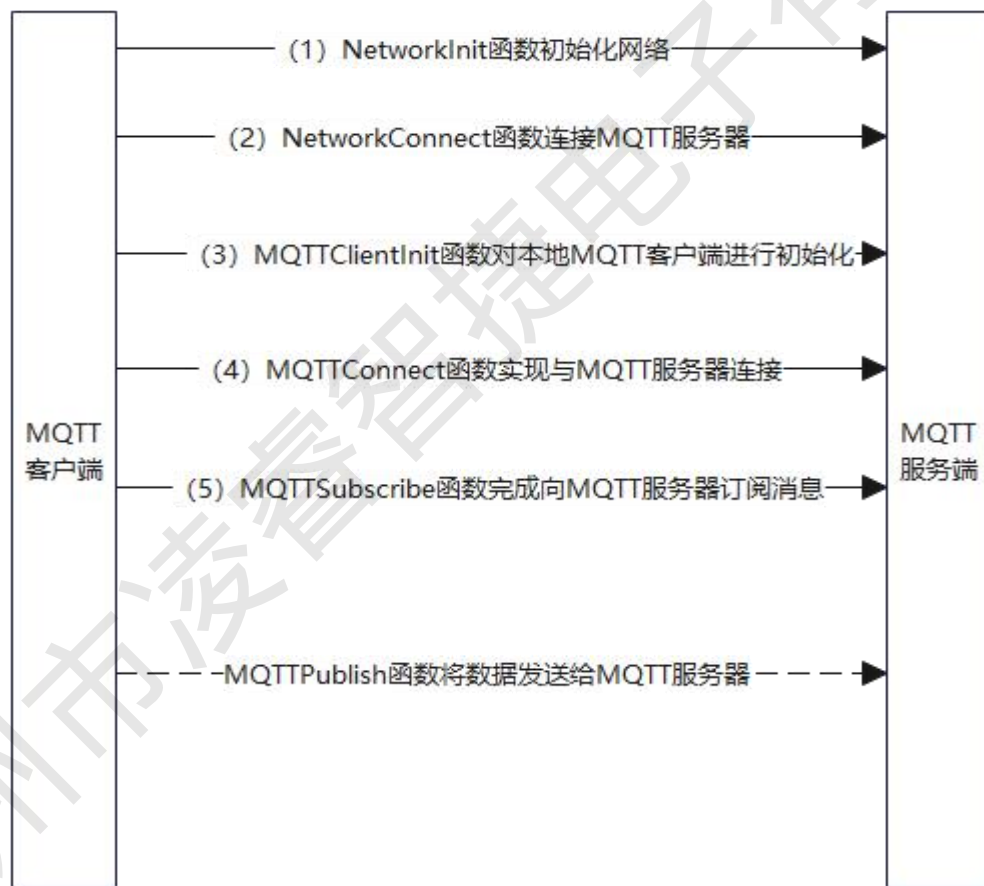
01

什么是MQTT



02

MQTT接口分析



02

MQTT接口分析

头文件: `//third_party/paho_mqtt/MQTTClient-C/src/liteOS/MQTTLiteOS.h`

```
void NetworkInit(Network* n);
```

该函数主要功能是网络初始化。该接口的头文件在OpenHarmony第三方库中，属于paho_mqtt。

■ 参数n：网络结构体指针，包含socket ID，读操作函数指针和写操作函数指针。

无返回值。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/liteOS/MQTTLiteOS.h

```
int NetworkConnect(Network* n, char* addr, int port);
```

该函数主要功能是网络连接。

- 参数n: 网络结构体指针, 包含socket ID, 读操作函数指针和写操作函数指针;
- 参数addr: MQTT服务器IP地址;
- 参数port: MQTT服务器端口。

返回值为0则成功; -1则失败。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/liteOS/MQTTLiteOS.h

```
int NetworkDisconnect(Network* n);
```

该函数主要功能是断开网络连接。

■ 参数n: 网络结构体指针, 包含socket ID, 读操作函数指针和写操作函数指针;
无返回值。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/liteOS/MQTTLiteOS.h

```
void MQTTClientInit(MQTTClient* client,  
                    Network* network,  
                    unsigned int command_timeout_ms,  
                    unsigned char* sendbuf,  
                    size_t sendbuf_size,  
                    unsigned char* readbuf,  
                    size_t readbuf_size);
```

02

MQTT接口分析

该函数主要功能是MQTT客户端初始化。

- 参数`clien`: MQTT客户端结构体指针
- 参数`network`: 网络结构体指针, 包含`socket ID`, 读操作函数指针和写操作函数指针
- 参数`command_timeout_ms`: MQTT发送命令超时时间
- 参数`sendbuf`: 发送缓冲区指针
- 参数`sendbuf_size`: 发送缓冲区大小
- 参数`readbuf`: 读缓冲区指针
- 参数`readbuf_size`: 读缓冲区大小

无返回值。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/MQTTClient.h

```
int MQTTConnect(MQTTClient* client, MQTTPacket_connectData* options);
```

该函数主要功能是断开网络连接。

■ 参数n: 网络结构体指针, 包含socket ID, 读操作函数指针和写操作函数指针;
无返回值。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/**MQTTClient.h**

```
int MQTTDisconnect(MQTTClient* client, MQTTPacket_connectData* options);
```

该函数主要功能是断开连接到MQTT服务端。

■ 参数n: 网络结构体指针, 包含socket ID, 读操作函数指针和写操作函数指针;
返回值为0 则成功, -1则失败, -2则缓冲区溢出。

02

MQTT接口分析

头文件: //third_party/paho_mqtt/MQTTClient-C/src/MQTTClient.h

```
int MQTTSubscribe(MQTTClient* c, const char* topicFilter,  
                  enum QoS qos, messageHandler messageHandler);
```

该函数主要功能是向MQTT服务端订阅消息。

- 参数client: MQTT客户端结构体指针;
- 参数topicFilter: 订阅消息名称;
- 参数qos: QOS等级;
- 参数messageHandler: 订阅消息处理函数指针。

返回值为0 则成功, -1则失败, -2则缓冲区溢出。

02

MQTT接口分析

头文件: `//third_party/paho_mqtt/MQTTClient-C/src/MQTTClient.h`

```
int MQTTPublish(MQTTClient* c, const char* topicName, MQTTMessage* message);
```

该函数主要功能是向MQTT服务端订阅消息。

- 参数client: MQTT客户端结构体指针;
- 参数topicFilter: 订阅消息名称;
- 参数message: 消息指针, 包含消息内容、消息长度、消息ID和QOS等级。

返回值为0 则成功, -1则失败, -2则缓冲区溢出。

03

实例分析

`Iot_Mqtt_Thread`函数首先调用`NetworkInit`函数初始化网络；
其次调用`NetworkConnect`函数连接MQTT服务器；
其次调用`MQTTClientInit`函数对本地MQTT客户端进行初始化；
再次调用`MQTTConnect`函数实现与MQTT服务器连接；
再次，调用`MQTTSubscribe`函数完成向MQTT服务器订阅消息；
最后通过`MQTTPublish`函数将数据发送给MQTT服务器。

谢谢聆听

单击此处添加副标题内容