**浙大城市学院实验报告**

课程名称 物联网技术与应用 实验项目 实验12 MQTT协议分析

专业班级 计算机1803 学号 31801150 姓名 张帅

指导老师（签名 ） 蔡建平 日期 实验成绩

**一、实验目的：**

 了解MQTT协议；

 熟悉WireShark抓包工具。

**二、实验内容：**

1. 使用WireShark捕获协议包，分析MQTT和CoAP协议:

2. MQTT网络协议分析

 了解MQTT的连接建立和断开过程；

 了解MQTT发布消息；

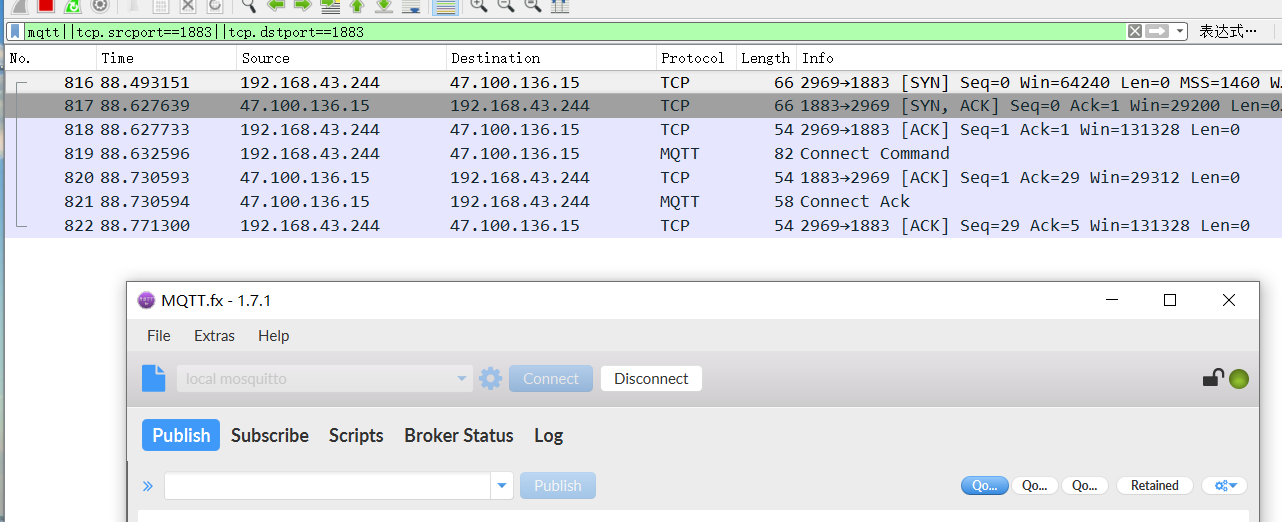
 了解MQTT订阅消息；

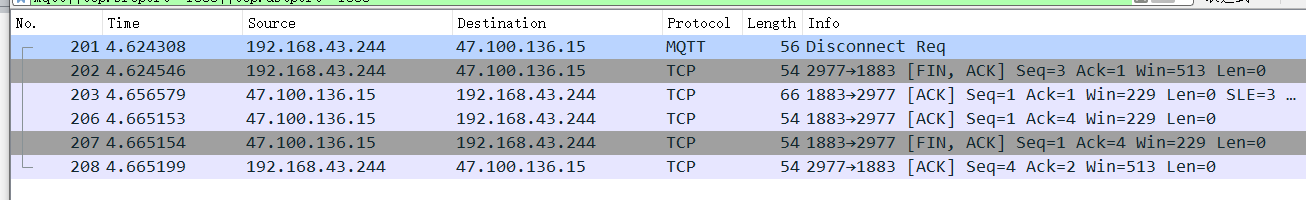
 了解MQTT协议的服务质量Qos。

**三、实验步骤：**

1. MQTT连接报文和断开连接报文

抓包截图：





分析：

连接过程：

前三条信息：服务端与代理进行三次握手建立连接

第四条信息：服务端向代理发送connect command命令进行连接

第五条信息：TCP确认收到信息

第六条信息：MQTT连接确认，发送connect ACK确认报文

第七条信息：TCP确认收到信息

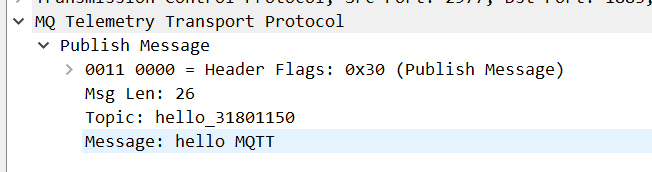
断开连接：  
第一条信息客户端向代理发送disconnect Req信息断开连接

接下来4条报文为服务器与代理进行四次挥手断开TCP连接

1. MQTT发布消息

抓包截图：





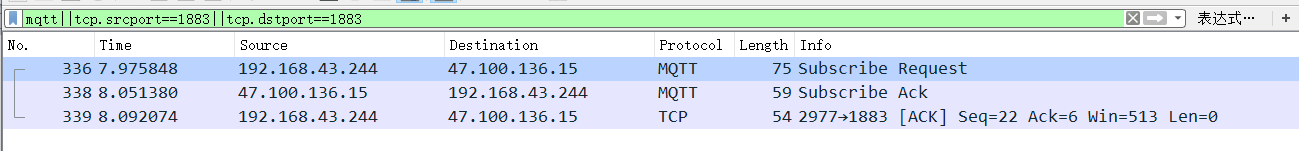
分析：

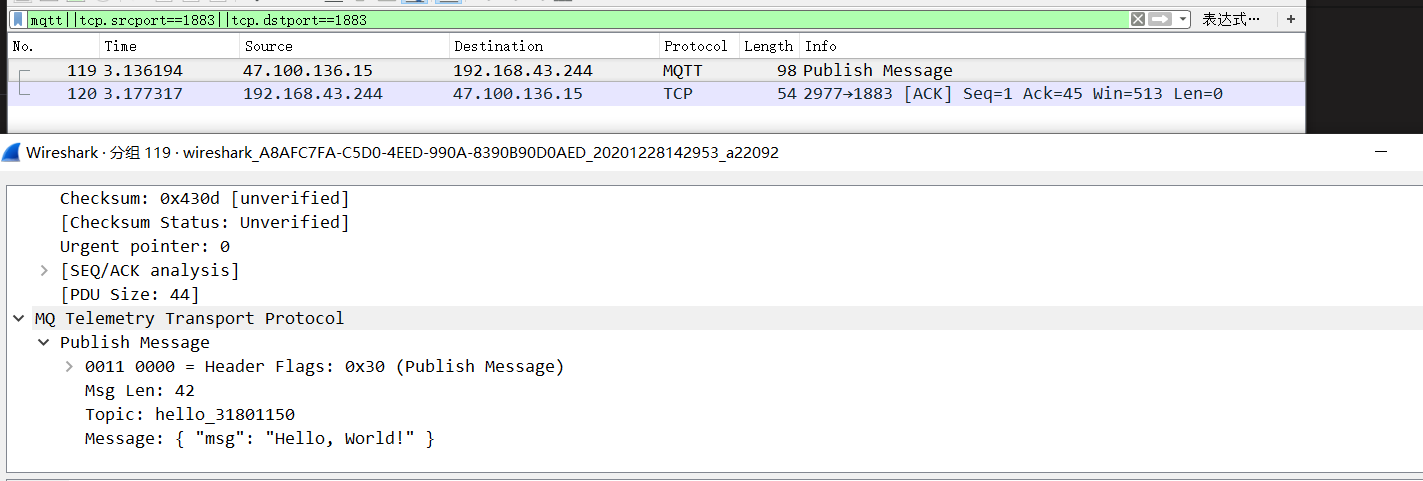
第一条信息：服务端向代理发送Publish message报文，报文包含topic和message

第二条信息：代理收到信息后发送TCP ACK确认报文

1. MQTT订阅消息

抓包截图：





分析：

订阅：

1.第一条信息客户端向代理发送订阅的请求Subscribe Request

2.第二条信息代理收到请求后向用户发送Subscribe ACK订阅确认

3.第三条信息客户端收到确认以后发送tcp确认报文

收到消息：  
1.第一条信息为客户端收到MQTT代理的Publish Message报文，内容为订阅主题刚刚发送的信息，客户收到的报文与发送的报文一致

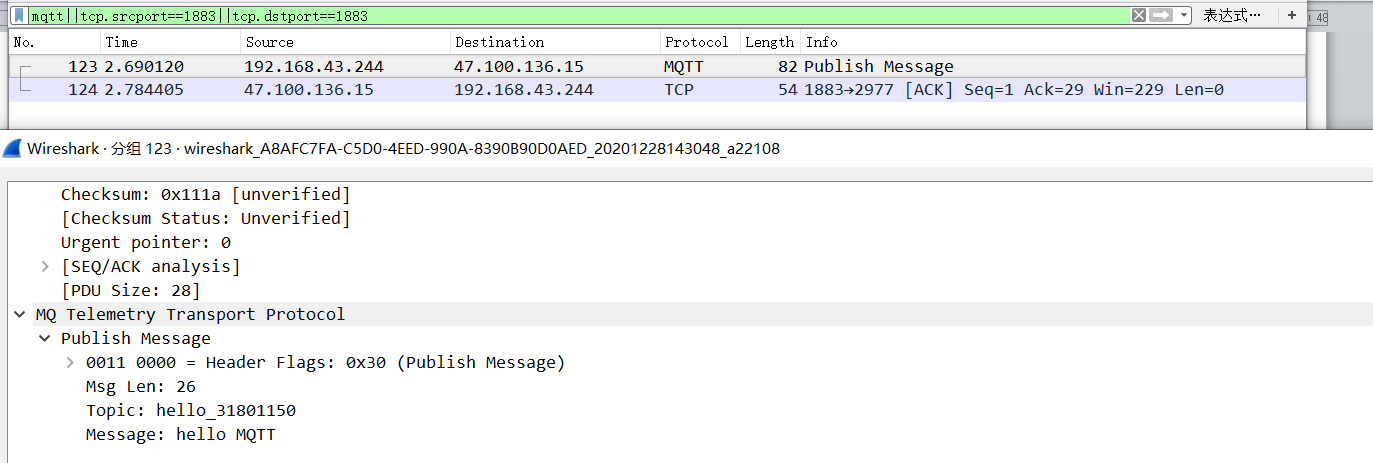
2.第二条信息为收到报文后发送的的TCP ACK确认

1. 服务质量Qos对MQTT Publish报文的影响

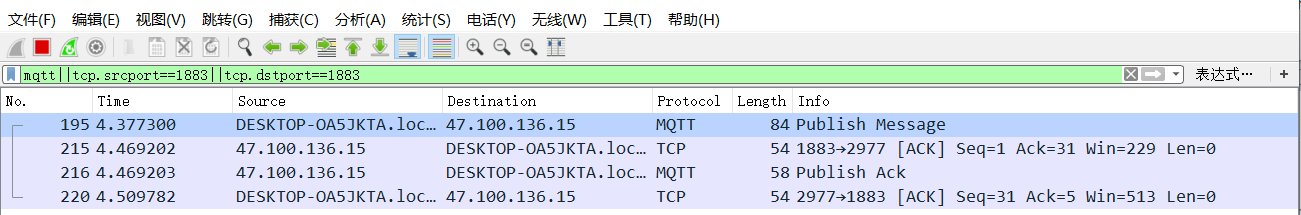
分别设置QoS为0、1、2，观察WireShark捕获的报文。

抓包截图：

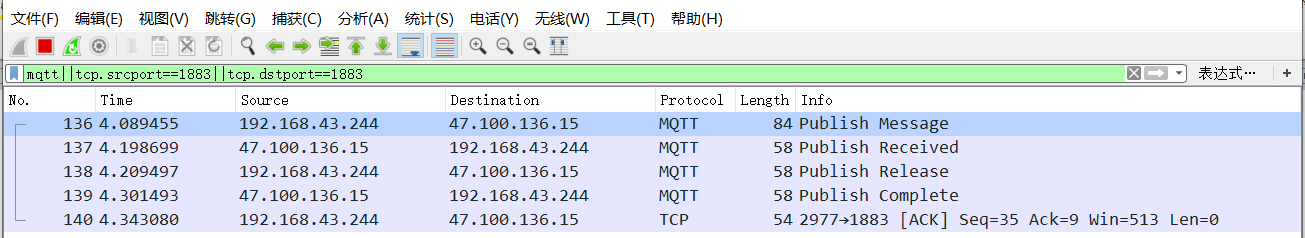
Qos：0



Qos：1



Qos：2



分析：

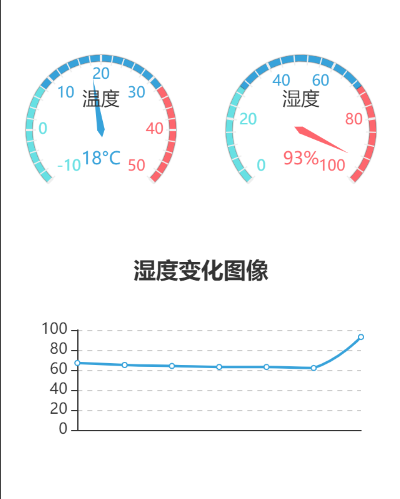
QoS为0时，只发送一次，只有客户端向代理发送报文，以及代理发送TCP的ACK回复

QoS为1时，至少发送一次，客户端向代理发送报文后，需要等待代理发送的Publish ACK回复，收到回复后才停止发送，可能重复发送

QoS为2时，确保消息不会丢失，客户端先向代理发送报文，客户端收到后发送Publish Received报文说明自己收到，客户端收到消息后发送 Publish Release报文释放连接，代理收到后发送Publish Complete报文结束过程，保证信息会发送到代理而且只发送一次。

5.（选做）完成MQTT消息的echarts可视化图表显示；

Echarts图表代码



//index.js

//获取应用实例

//import \* as echarts from '../../node\_modules/echarts'

import \* as echarts from '../../ec-canvas/echarts';

const app = getApp()

var mqtt = require('../../utils/mqtt.js');

var chart1 = null;

var chart2 = null;

var chart3 = null;

function initChart(canvas, width, height, dpr) {

  chart1 = echarts.init(canvas, null, {

    width: width,

    height: height,

    devicePixelRatio: dpr // new

  });

  canvas.setChart(chart1);

  var option = {

    backgroundColor: "#ffffff",

    color: ["#37A2DA", "#32C5E9", "#67E0E3"],

    series: [{

      name: '温度',

      type: 'gauge',

      min:-10,

      max:50,

      splitNumber: 6,

      fontSize:10,

      detail: {

        formatter: '{value}%',

        fontSize:14

      },

      axisLine: {

        show: true,

        lineStyle: {

          width: 5,

          shadowBlur: 1,

          color: [

            [0.3, '#67e0e3'],

            [0.7, '#37a2da'],

            [1, '#fd666d']

          ]

        }

      },

      pointer: {

        width: 3, //指针的宽度

        length: "70%", //指针长度，按照半圆半径的百分比

      },

      splitLine: { //分割线样式

        length: 5

     },

      data: [{

        value: 0,

        name:"温度",

        fontSize:8,

      }]

    }]

  };

  chart1.setOption(option, true);

  return chart1;

}

function initChart3(canvas, width, height, dpr) {

  chart3 = echarts.init(canvas, null, {

    width: width,

    height: height,

    devicePixelRatio: dpr // new

  });

  canvas.setChart(chart3);

  var option = {

    color: "#37A2DA",

    legend: {

      data: '湿度',

      top: 50,

      left: 'center',

      z: 100

    },

    title: {

      text: '湿度变化图像',

      left: 'center'

    },

    grid: {

      containLabel: true

    },

    tooltip: {

      show: true,

      trigger: 'axis'

    },

    xAxis: {

      type: 'category',

      boundaryGap: false,

      data: [],

      // show: false

    },

    yAxis: {

      x: 'center',

      type: 'value',

      splitLine: {

        lineStyle: {

          type: 'dashed'

        }

      }

      // show: false

    },

    series: [{

      name: 'A',

      type: 'line',

      smooth: true,

      data: []

    }]

  };

  chart3.setOption(option);

  return chart3;

}

function initChart2(canvas, width, height, dpr) {

  chart2 = echarts.init(canvas, null, {

    width: width,

    height: height,

    devicePixelRatio: dpr // new

  });

  canvas.setChart(chart2);

  var option = {

    backgroundColor: "#ffffff",

    color: ["#37A2DA", "#32C5E9", "#67E0E3"],

    series: [{

      name: '湿度',

      type: 'gauge',

      min:0,

      max:100,

      splitNumber: 5,

      fontSize:10,

      detail: {

        formatter: '{value}%',

        fontSize:14

      },

      axisLine: {

        show: true,

        lineStyle: {

          width: 5,

          shadowBlur: 1,

          color: [

            [0.3, '#67e0e3'],

            [0.7, '#37a2da'],

            [1, '#fd666d']

          ]

        }

      },

      pointer: {

        width: 3, //指针的宽度

        length: "70%", //指针长度，按照半圆半径的百分比

      },

      splitLine: { //分割线样式

        length: 5

     },

      data: [{

        value: 0,

        name:"湿度",

        fontSize:8,

      }]

    }]

  };

  chart2.setOption(option, true);

  return chart2;

}

Page({

  data: {

    sub:false,

    sub\_text:'订阅',

    but\_type:'primary',

    humidity:0,

    temperature:0,

    date:null,

    humidity\_list:[],

    topic:'espZS/RE',

    ec: {

      onInit: initChart

    },

    ec3:{

      onInit: initChart3

    },

    ec2: {

      onInit: initChart2

    }

  },

  echartInit (e) {

    initChart(e.detail.canvas, e.detail.width, e.detail.height);

  },

  onLoad: function() {

    //this.initSocket();

  },

  TopicInput:function(e){

    this.setData({

     topic:e.detail.value

    })

  },

  //接收信息函数

  mqttsub: function() {

    let that = this;

    if(that.data.sub == true){

      app.globalData.client.unsubscribe(this.data.topic)

      that.setData({

        but\_type:'primary',

        sub\_text:'订阅',

        sub:false

      })

      return;

    }

    let x\_list = that.data.humidity\_list;

    app.globalData.client.subscribe(this.data.topic);

    app.globalData.client.on('message', function(topic, payload) {

      var receive=JSON.parse(payload);

      var time=new Date();

      //赋值

       that.setData({

          temperature:receive.temperature,

          humidity:receive.humidity,

       })

       if(that.data.humidity\_list.length<20){

        x\_list = that.data.humidity\_list;

        x\_list.push(that.data.humidity);

        that.setData({

          humidity\_list:x\_list

       })

      }

       else{

        x\_list = that.data.humidity\_list;

        x\_list.shift();

        x\_list.push(that.data.humidity);

        that.setData({

          humidity\_list:x\_list

       })

       }

       let option = {

        backgroundColor: "#ffffff",

        color: ["#37A2DA", "#32C5E9", "#67E0E3"],

        series: [{

          name: '温度',

          type: 'gauge',

          min:-10,

          max:50,

          splitNumber: 6,

          fontSize:10,

          detail: {

            formatter: '{value}°C',

            fontSize:14

          },

          axisLine: {

            show: true,

            lineStyle: {

              width: 5,

              shadowBlur: 1,

              color: [

                [0.3, '#67e0e3'],

                [0.7, '#37a2da'],

                [1, '#fd666d']

              ]

            }

          },

          pointer: {

            width: 3, //指针的宽度

            length: "70%", //指针长度，按照半圆半径的百分比

          },

          splitLine: { //分割线样式

            length: 5

         },

          data: [{

            value: that.data.temperature,

            name:"温度",

            fontSize:8,

          }]

        }]

      };

      let option2 = {

        backgroundColor: "#ffffff",

        color: ["#37A2DA", "#32C5E9", "#67E0E3"],

        series: [{

          name: '湿度',

          type: 'gauge',

          min:0,

          max:100,

          splitNumber: 5,

          fontSize:10,

          detail: {

            formatter: '{value}%',

            fontSize:14

          },

          axisLine: {

            show: true,

            lineStyle: {

              width: 5,

              shadowBlur: 1,

              color: [

                [0.3, '#67e0e3'],

                [0.7, '#37a2da'],

                [1, '#fd666d']

              ]

            }

          },

          pointer: {

            width: 3, //指针的宽度

            length: "70%", //指针长度，按照半圆半径的百分比

          },

          splitLine: { //分割线样式

            length: 5

         },

          data: [{

            value: that.data.humidity,

            name:"湿度",

            fontSize:8,

          }]

        }]

      };

      let option3 = {

        color: "#37A2DA",

    legend: {

      data: '湿度',

      top: 50,

      left: 'center',

      z: 100

    },

    title: {

      text: '湿度变化图像',

      left: 'center'

    },

    grid: {

      containLabel: true

    },

    tooltip: {

      show: true,

      trigger: 'axis'

    },

    xAxis: {

      type: 'category',

      boundaryGap: false,

      data: [],

      // show: false

    },

    yAxis: {

      x: 'center',

      type: 'value',

      splitLine: {

        lineStyle: {

          type: 'dashed'

        }

      }

      // show: false

    },

    series: [{

      name: 'A',

      type: 'line',

      smooth: true,

      data: that.data.humidity\_list

    }]

      };

    chart1.setOption(option);

    chart2.setOption(option2);

    chart3.setOption(option3);

    })

    that.setData({

      but\_type:'warn',

      sub\_text:'取消订阅',

      sub:true

    })

  }

})

6.（选做）在node-red环境中完成MQTT消息可视化显示；

