# Task2 UDP socket programming

## 1 目的：

（1）掌握 UDP socket 程序结构。

（2）掌握自定义应用层报文格式及报文交互过程，即自定义应用层协议。

## 2描述

以下即是对总体工作过程的描述，也包含待完成的功能点，请仔细阅读

1. client和 server 都是基于UDP的命令行程序。client 运行在host os，server运行在guestos。client 端启动时在命令行方式下指定 serverIP, serverPort。
2. 模拟 TCP 连接建立过程，即建立完毕连接以后，才进入到数据传输阶段。

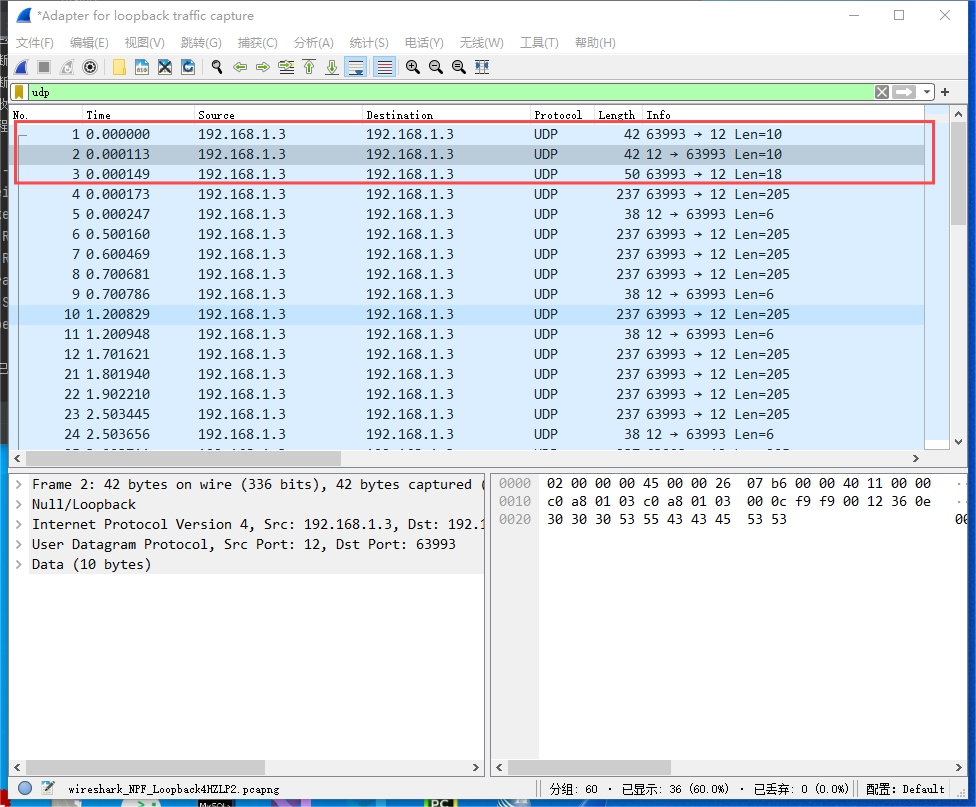
Client:



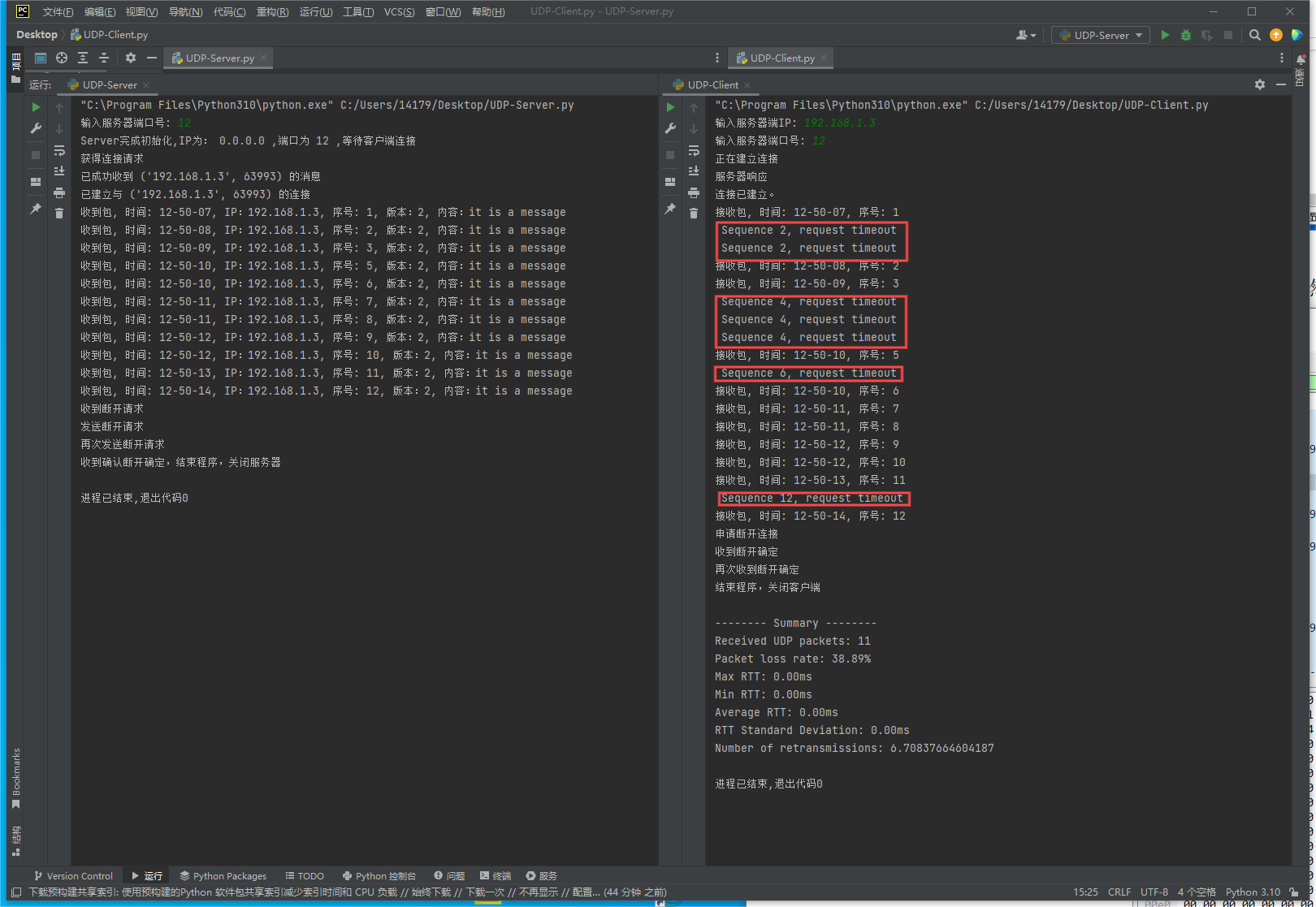
Server:



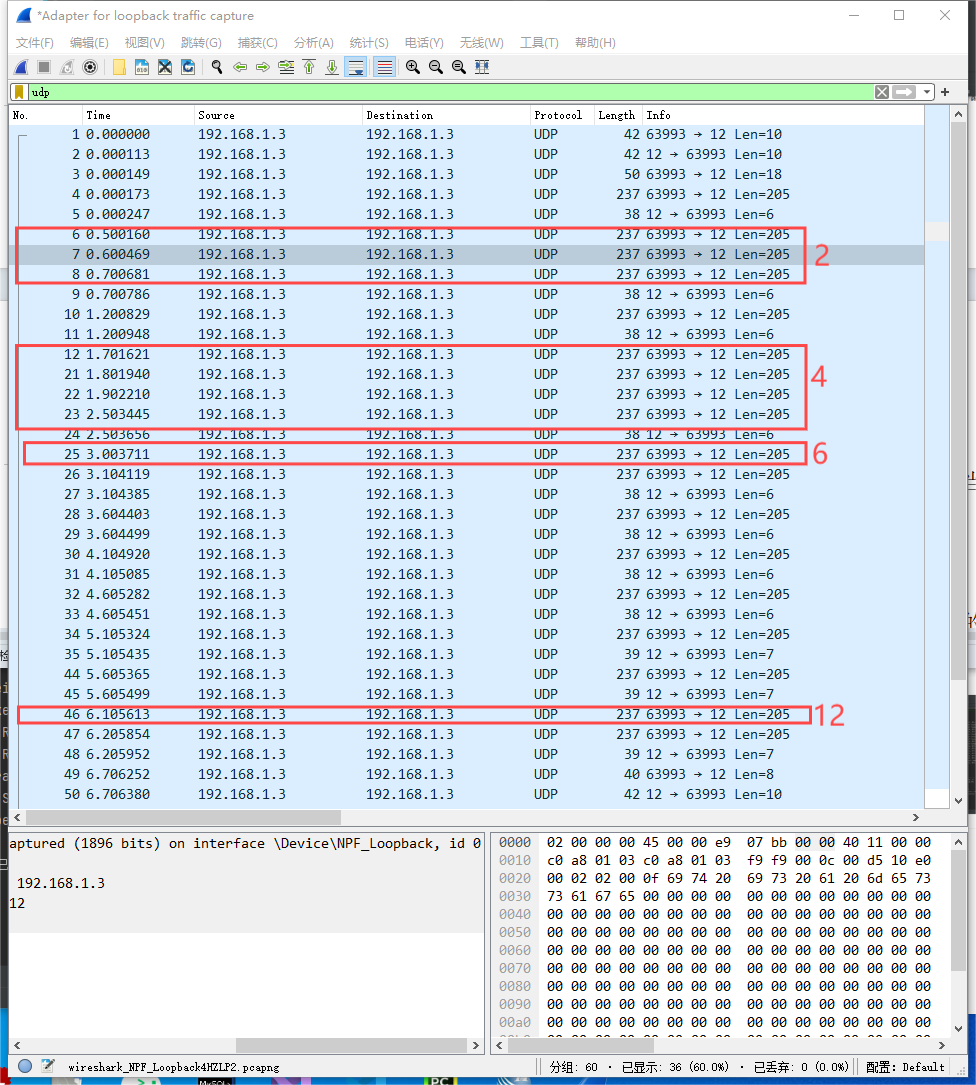
Wireshark:



1. client发送12个request数据包，对每个reqeust数据包，server 端采用随机不响应 client 请求的方式，来模拟丢包场景,丢包率自定。(说明:UDP 虽然不可靠，但指望发生UDP 丢包事件也是概率极低的，所以“丢包”这种事，得自己在应用层程序udpserver中来构建。)



在程序的运行当中看见Client有超时现象，说明出现了丢包现象，此时需要到Wireshark当中，图中标注分别是序号，



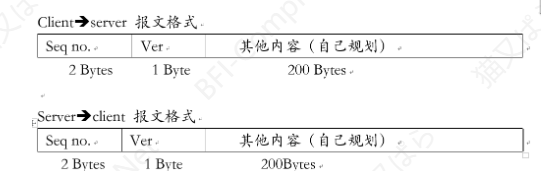
（4）给client 设定一个超时时间，100ms。在超时时间内:

a.若能接收到来自于 server 的响应报文，则计算本次交互的RTT(单位:ms)以及sever的系统时间(hh-mm-ss)。

b.若没有收到来自于 server 的回复，则认为发了丢包。若发生丢包，则重传;两次重传失败，则放弃本次重传。



（5）client 端和 server 端的交互及(你正在写的)本应用层的报文格式至少应该包括 Seq no、ver 字段,“其他内容”根据题目需求自已规划:

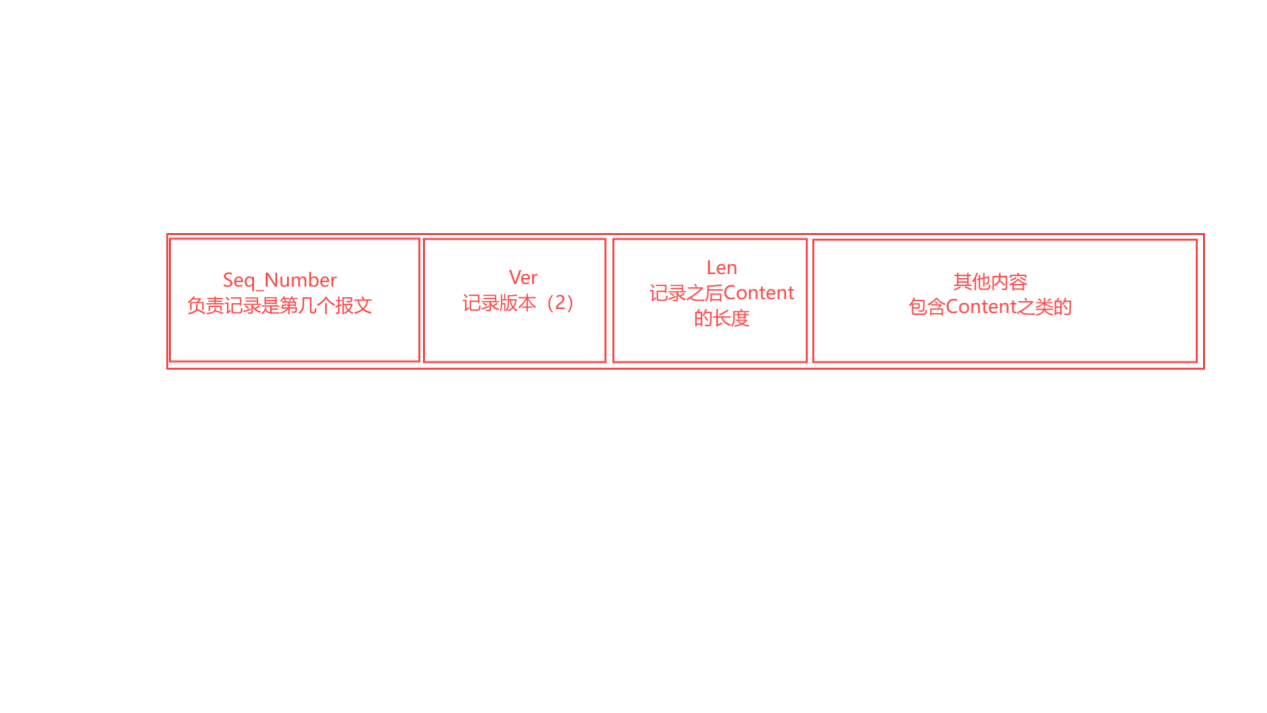


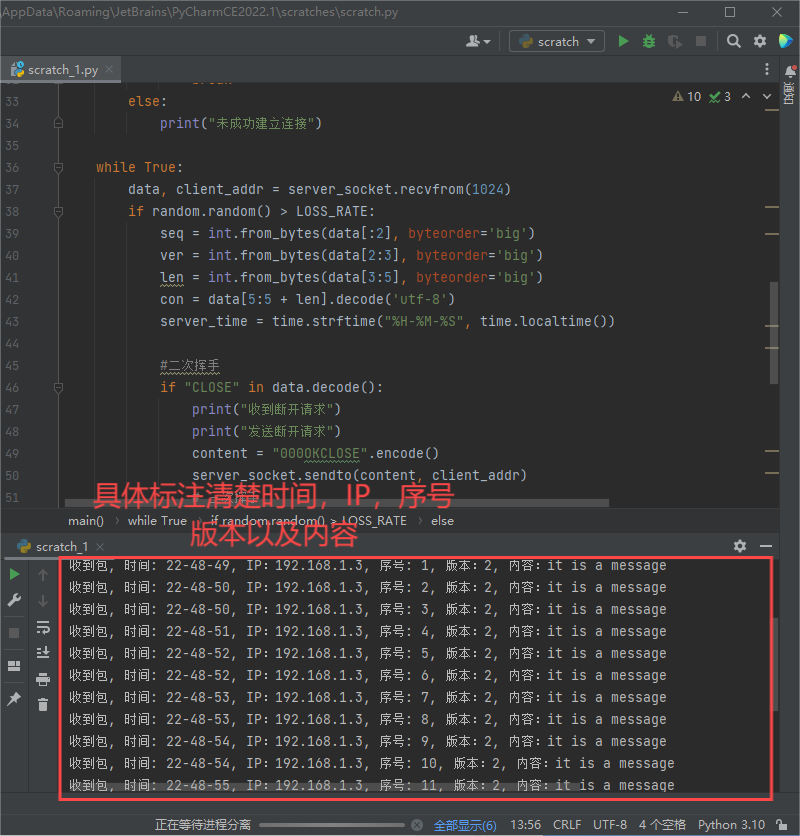
a.发送的报文中要包含sequencenumber、ver，sequence number 指的是发送的第几个报文，从1开始，按序递增。ver指的是版本号，定为2

b.Server->client 报文(若有回复，也可能没有)中Seqno.与刚刚从client 接收到的报文相对应。

c.“其他内容(自己规划)”:例如，server-client 中至少要存放系统时间等;client->server 可以存放无意义的字母序列作为填充等。

报文结构：





1. client 端每发送一个报文，若收到了来自于server 端的response,则print出“sequence no，serverIP:Port、RTT”。若发生超时即在100ms内没有收到来自于server的response，则print出sequence no,request time out”



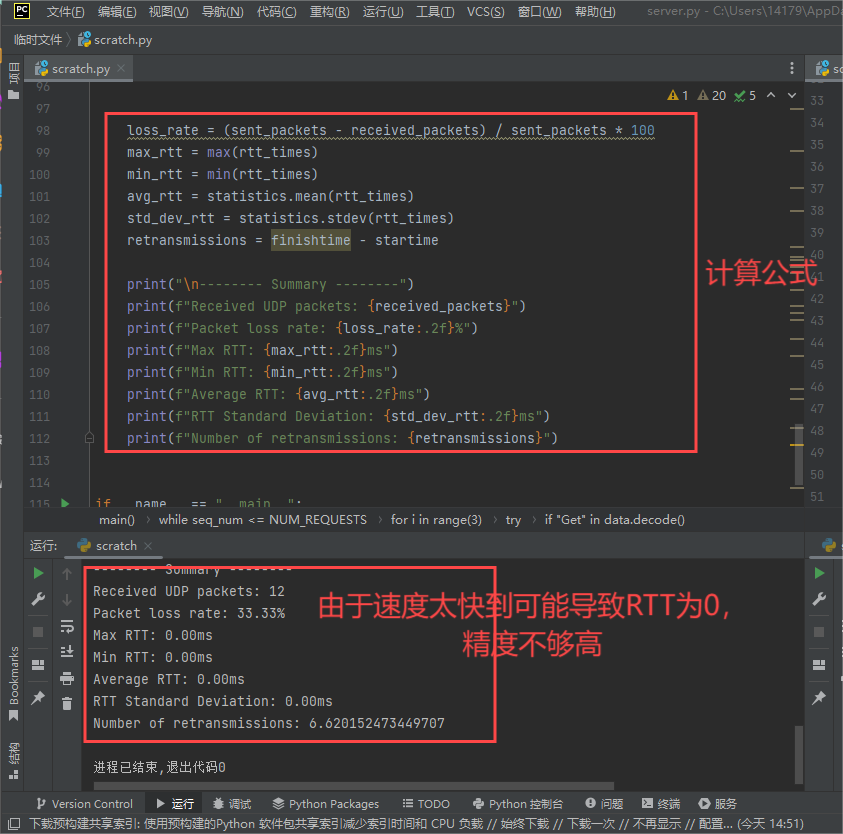
（7）当发送完毕12个request 报文后，client 端，打印出如下【汇总】信息:

·接收到的 udp packets 数目

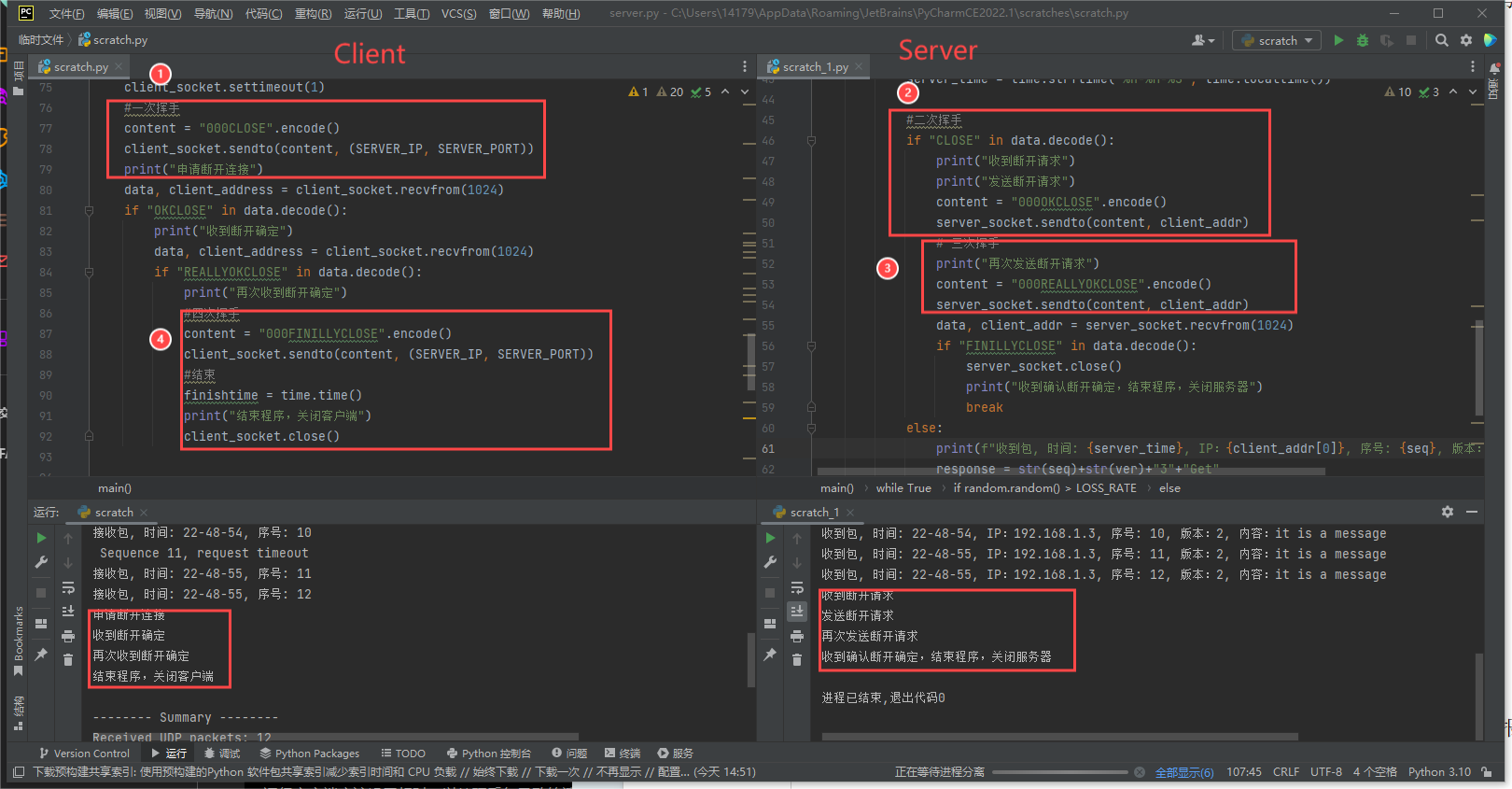
·丢包率(百分比:1-接收到的udppacket/发送的udppacket)

·最大 RTT、最小 RTT、平均RTT、RTT 的标准差

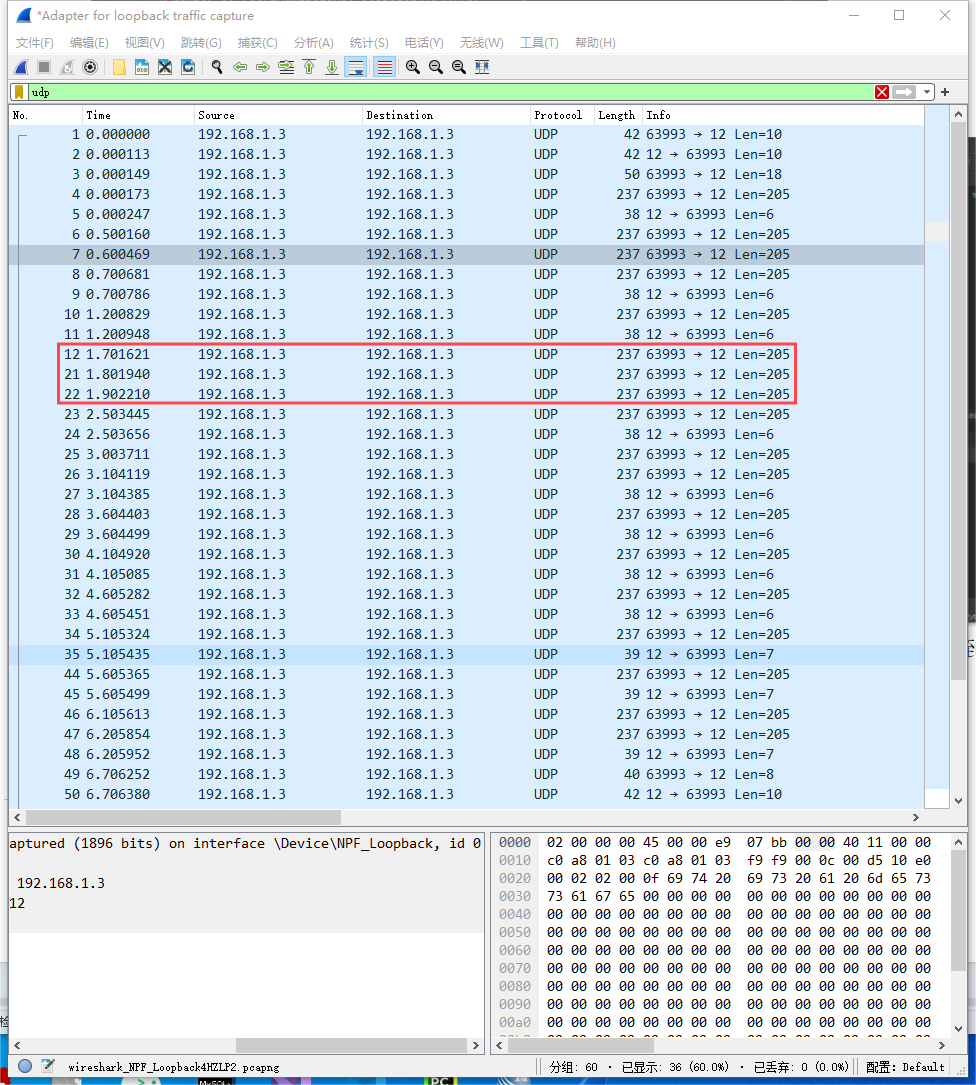
·server 的整体响应时间(server最后一次response 的系统时间与第一次 response 的系统时间之差)



1. 模拟 TCP 连接释放过程，关闭client-server 之间的交互



1. 在client-server 交互过程中，使用wireshark抓包，找到有两次重传的报文。为捕获到这种行为，你可能需要多次运行client-server



一次正常传+两次重传

## 3 遇见的关键点和解决办法

在进行实验的时候感受最大的一点就是UDP断开连接的时候，一旦断开的报文的丢包就会导致整个程序运行不起来（也即是Server和Client端一直都等待这彼此发报文，但由于结束报文丢包了导致进入了死锁状态）对于这个点的话我的解决办法就是设置两个超时时间，一个超时时间表示数据传说的超时，如果超过了这个的话就说明数据没有传输过来发生丢包，继续等待。第二个超时表示这结束的超时，一旦超过了这个的超时时间则默认断开连接。

## 4 掌握的知识点

无连接性：UDP是一种无连接协议，意味着在发送数据之前不需要先建立一个连接。

简单性：由于UDP不提供连接建立和关闭的过程，其协议开销较小，这使得数据传输更加高效。

数据传输：UDP将数据打包成数据报发送，每个数据报包含目的地的IP地址和端口，以及源的IP地址和端口。

数据校验：UDP通过使用一个简单的校验和（Checksum）机制来检测传输过程中的数据错误。

## 5 git的url

https://github.com/nekomatapara/file