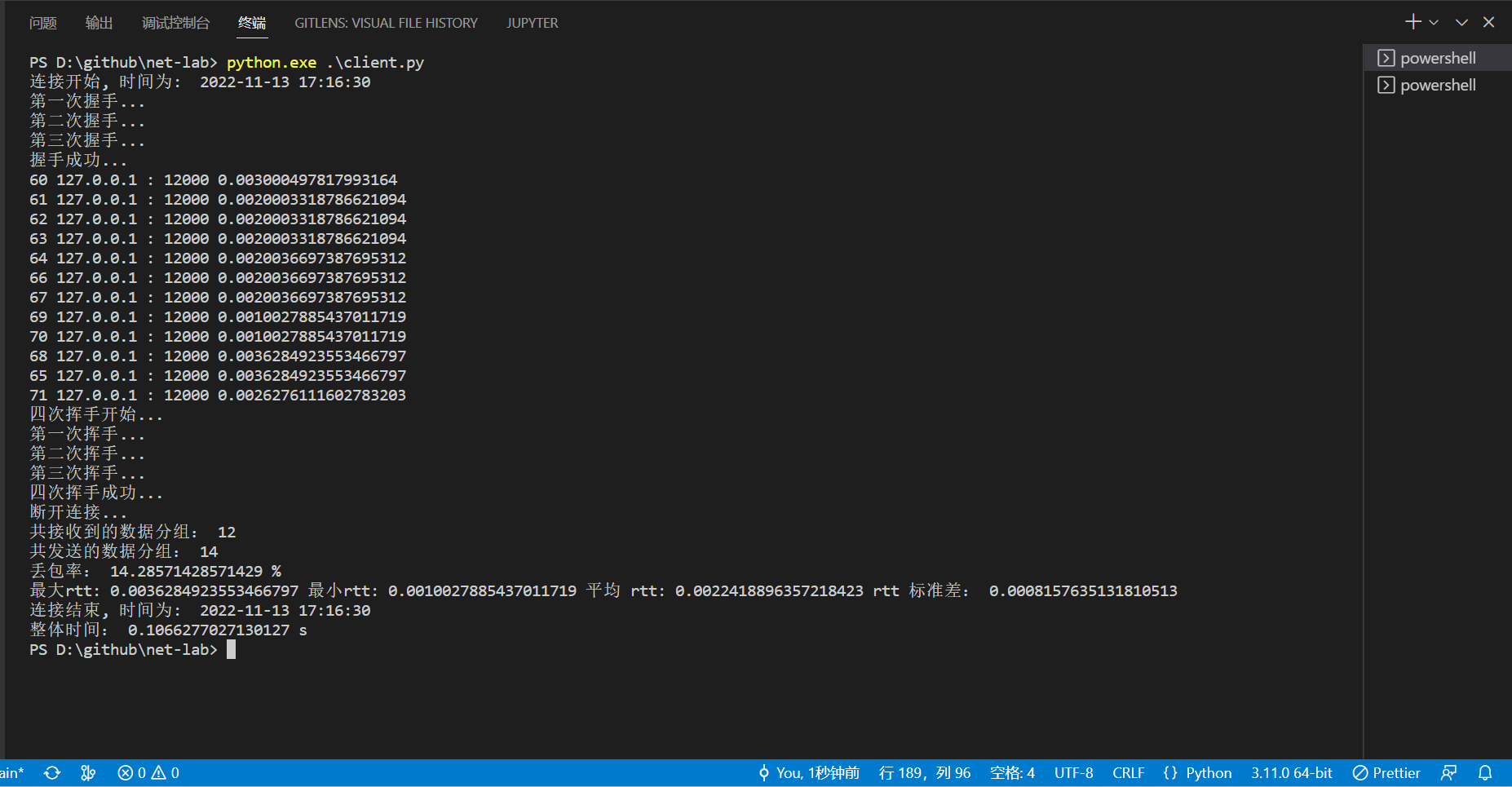
# 计算机网络课设 task2 实验报告

## 实现结果

### 客户端

***解释：***

首先开始时进行总体计时，然后开始三次握手，传输数据，添加计时器，接收数据，四次挥手断开连接。



### 服务端

***解释:***

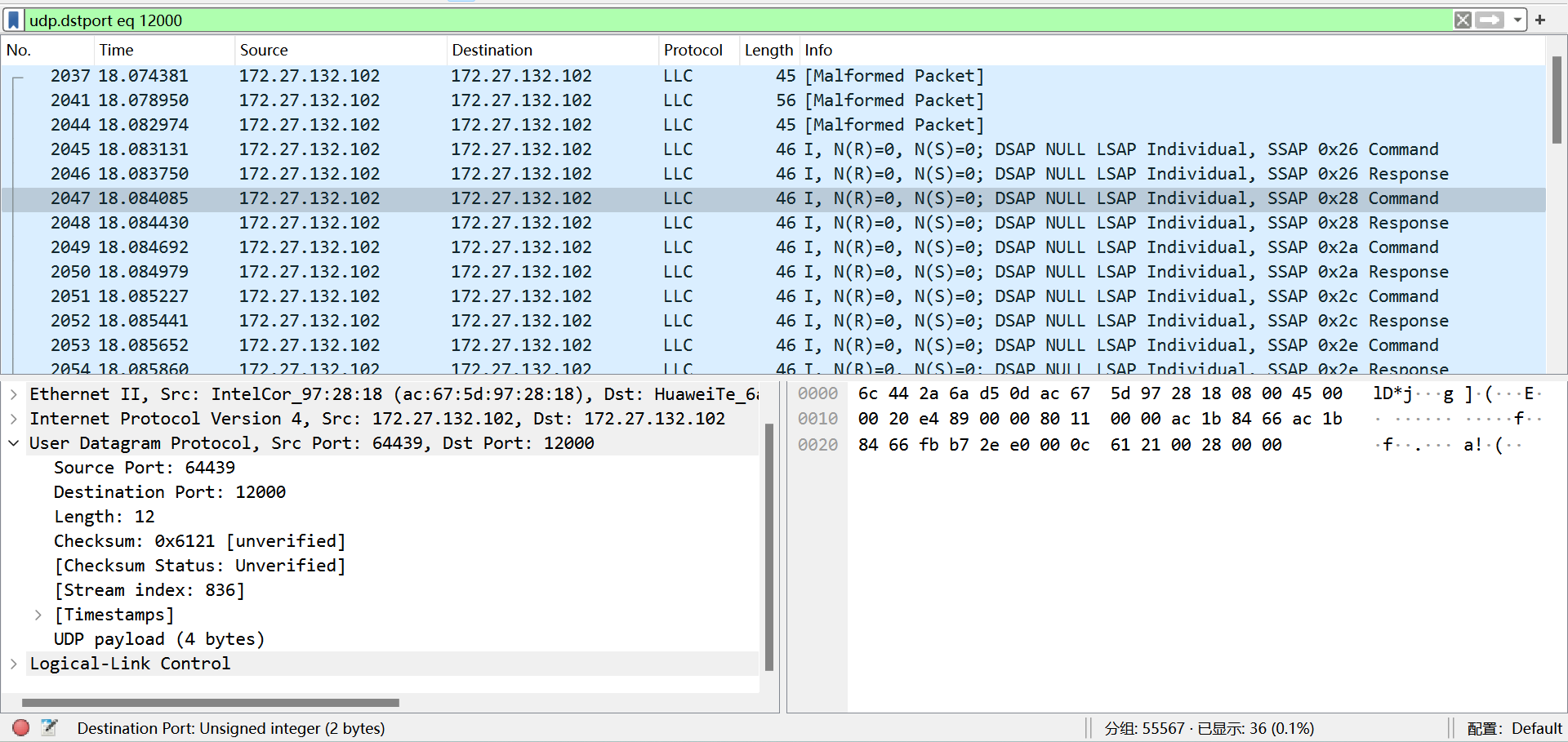
基本连接方式和客户端相同，对于每个请求使用随机数（0.8）来决定是否丢包。

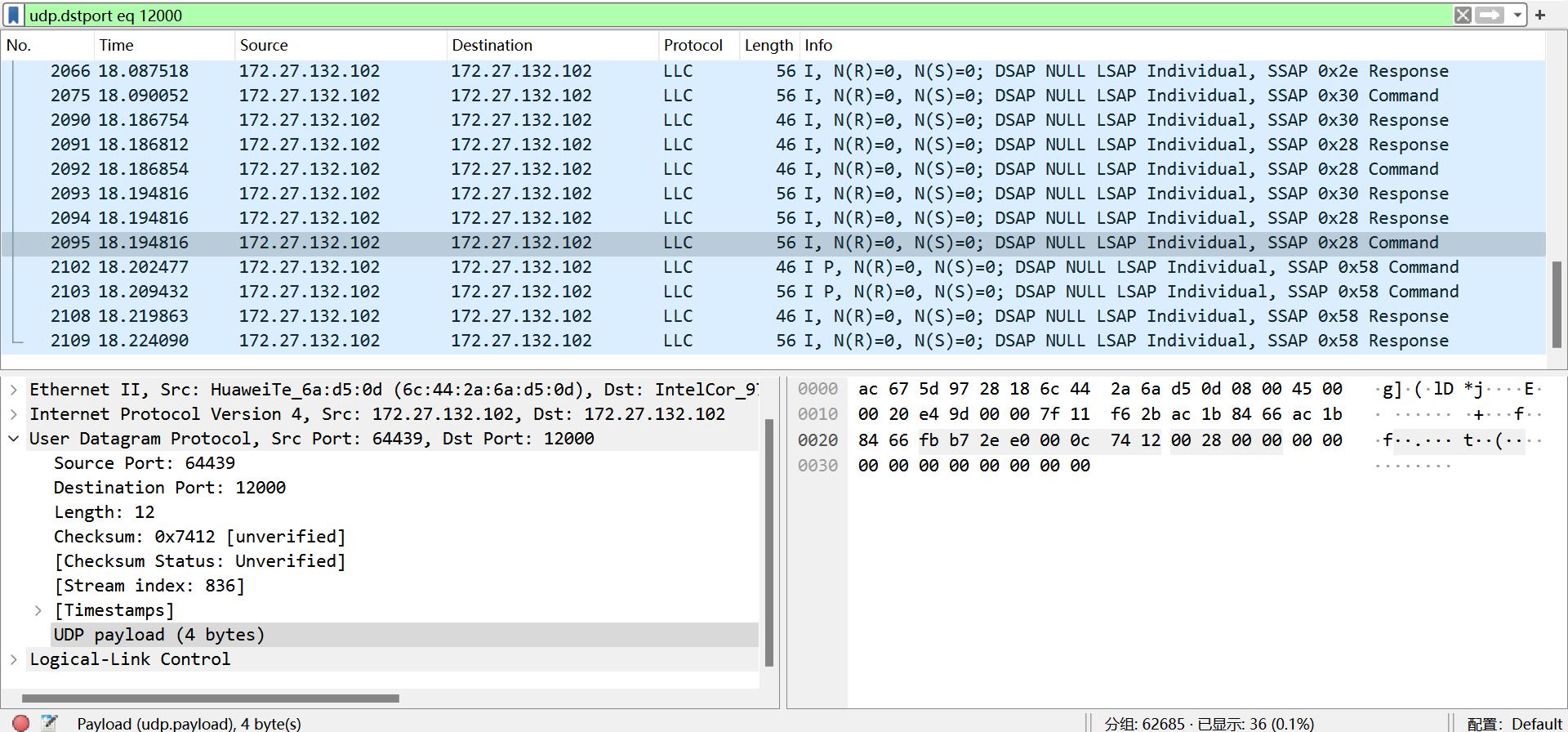


### Wire-shark

***解释：***

可以看到两次重传的报文只有数据（payload）相同，两个的报文长度不相同。





## 实现细节

### 服务端

1. 使用python创建一个udp socket，绑定固定的接口。
2. 等待客户建立连接, 提供三次握手，首先 syn = 1, 1, 0并判断seq等数据标志符号顺序后，成功建立连接。
3. 等待传输数据，对于每一个请求判断随机概率是否大于0.8来模拟丢包。
4. 若某请求 fin=1，则进入结束请求流程。四次挥手 fin，ack，fin，ack。

### 客户端

1. 创建 udp socket，请求相应的地址及端口，首先建立连接三次挥手，然后传送数据，当接受到的返回seq和发送的一致时，断开连接四次挥手。
2. 对于建立连接，与服务端基本一致。
3. 对于传送数据，发送以相应结构组织的字节流数据，对于每一个请求建立一个延时为timeout（1000ms）的Timer，如果在timeout前接收到此返回，则取消timer并将返回记录在返回窗口中，若超时则执行Timer并在Timer中新加一个timer。

## 使用知识点

1. Udp编程
2. Rtt计算
3. 三次握手建立连接
4. 超时重传
5. Python 字节流组织
6. python 定时器线程
7. 四次挥手关闭连接