# Task3 TCP Socket programming

## 1 目的

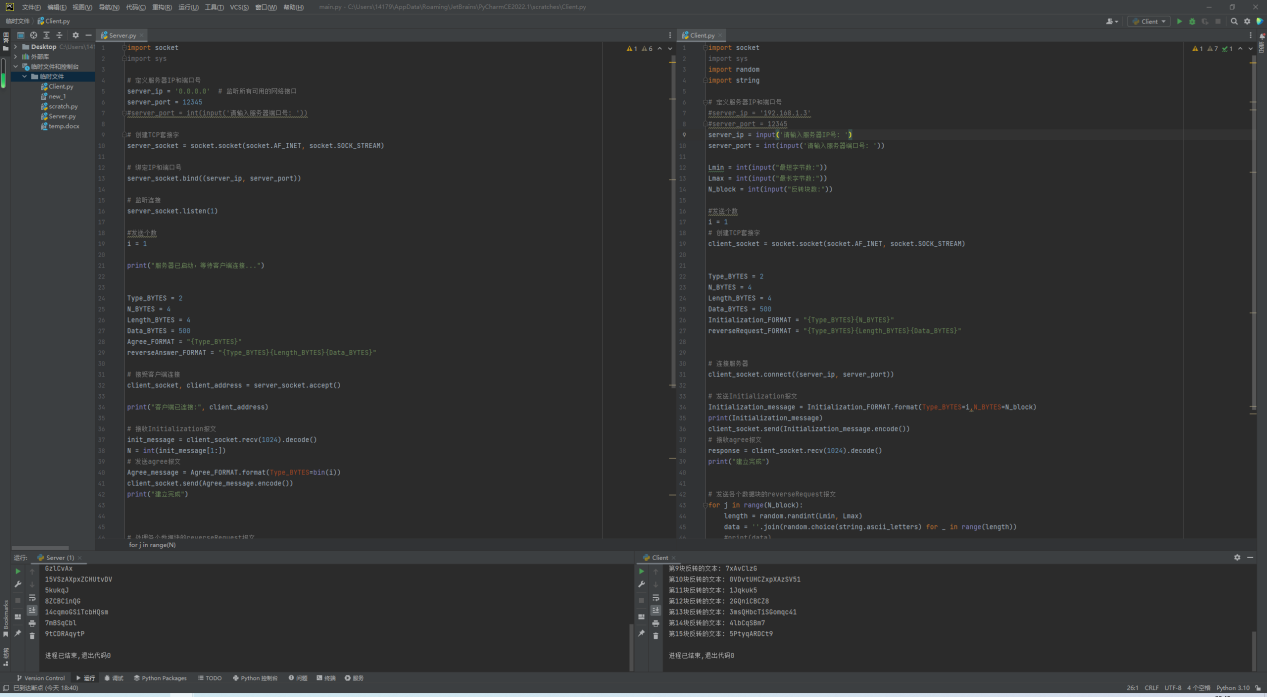
理解并掌握 TCP socket 程序设计的基本结构及工作过程。

掌握构造应用层自定义的报文结构、报文交互。

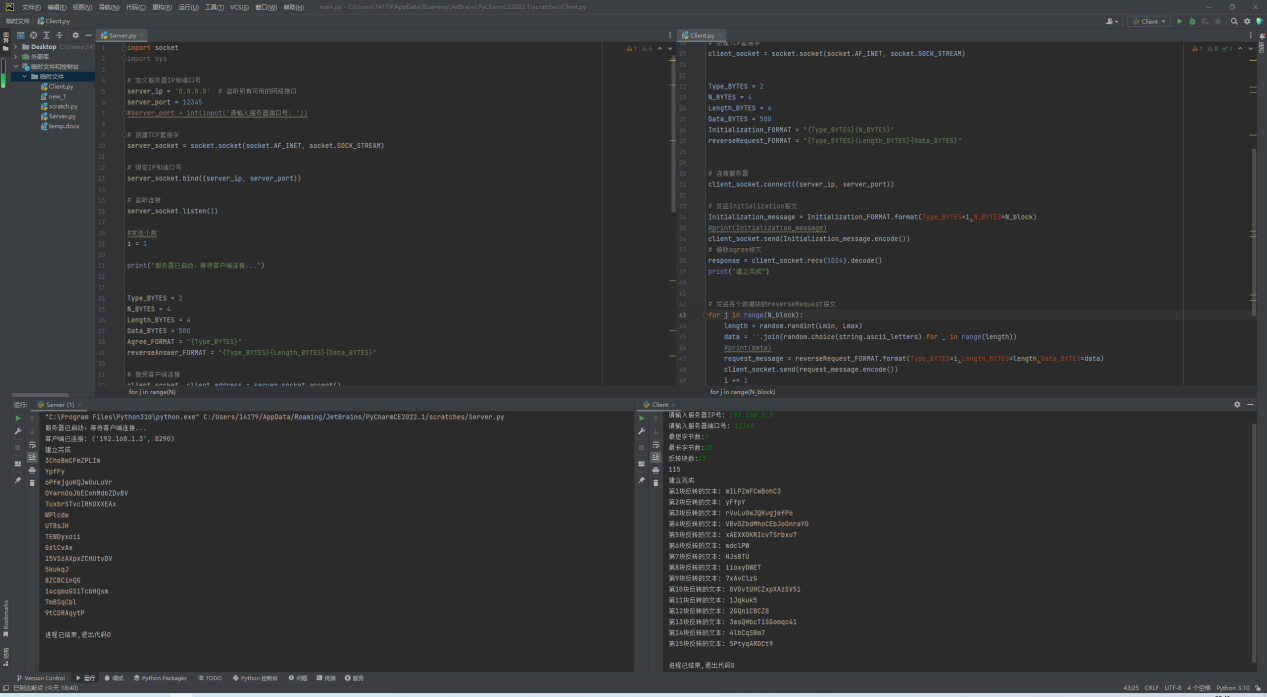
## 2描述

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

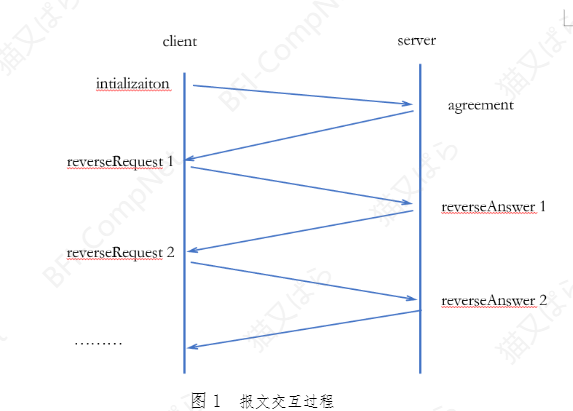
1. client和server 都是基于TCP的命令行程序。client 运行在host os，server运行在guest os。

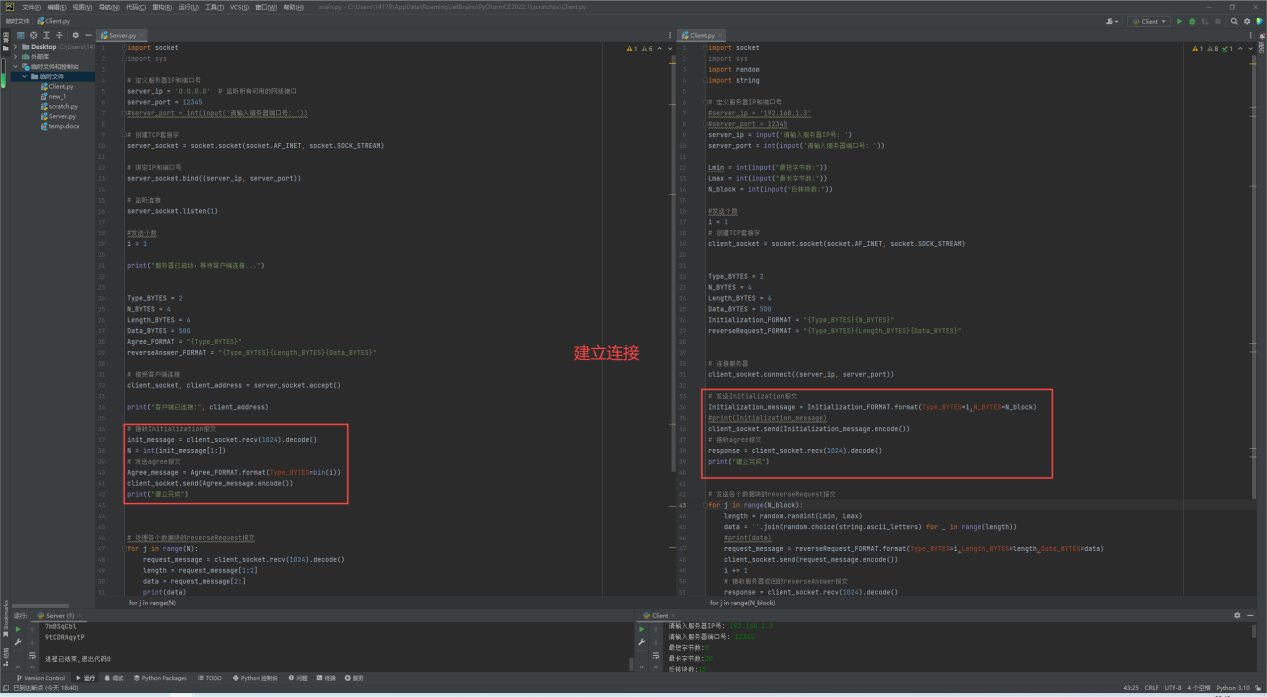


1. 有一个全英文可打印字符的 ASCII文件(这个文件你要自己构造),client 每次发送这个文件“不定长的一段/一块”给 server，的server把这块文本reverse(例如，“alittle monkey“变成“yeknom elttil a”，并返回给client，client 在终端打印显示。最终 client 要输出一个文本文件，该文件是原始文件的全部反转。

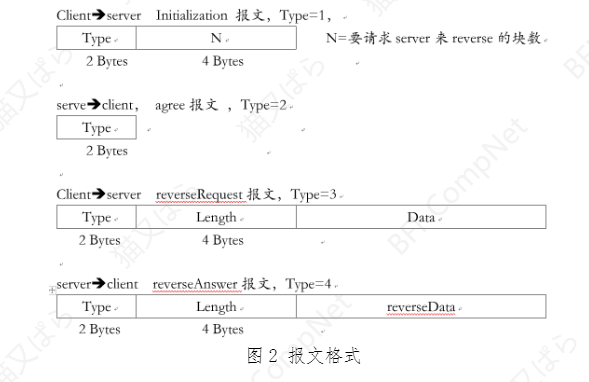


（3）client 和 server 之间的交互如图1:

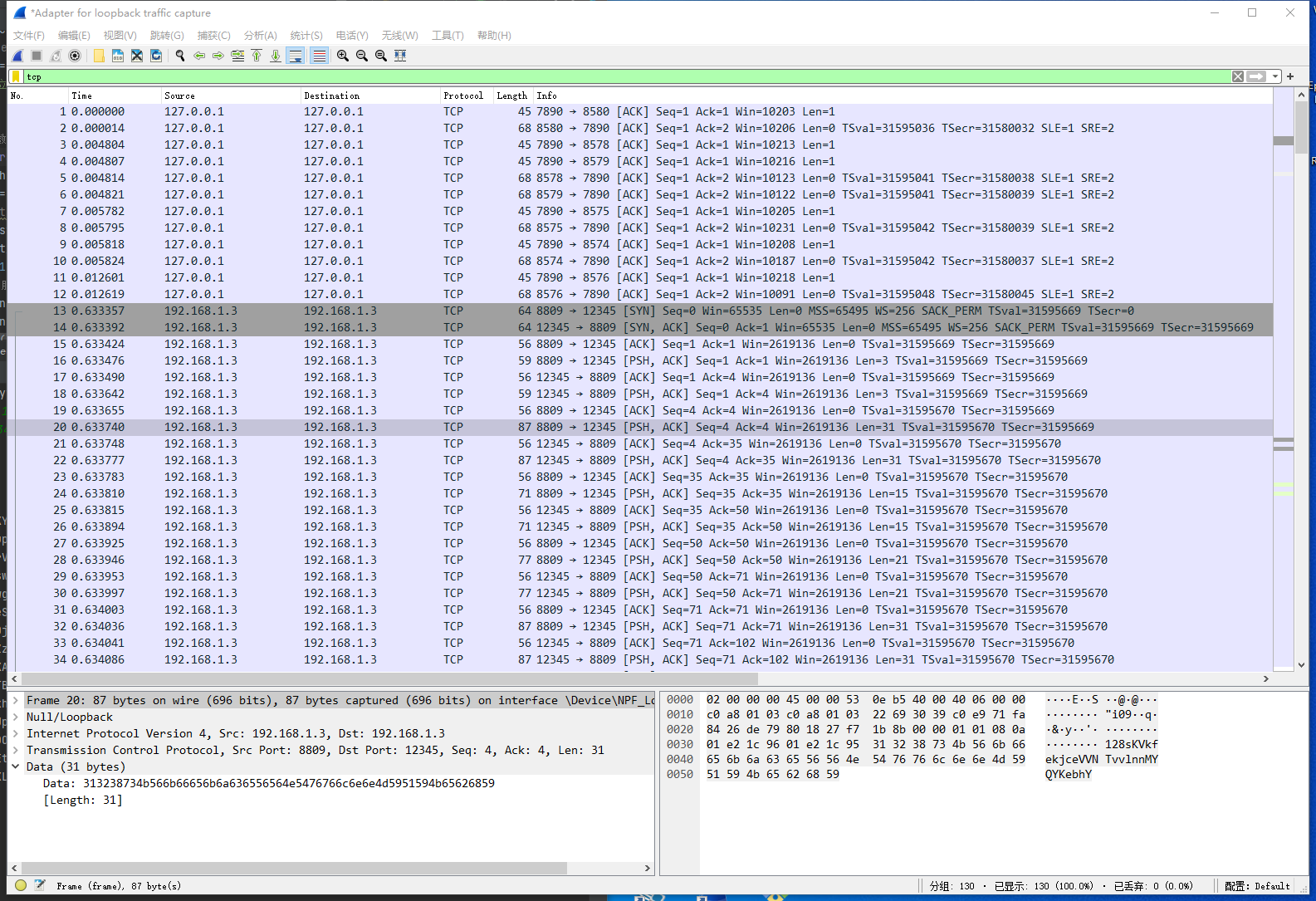




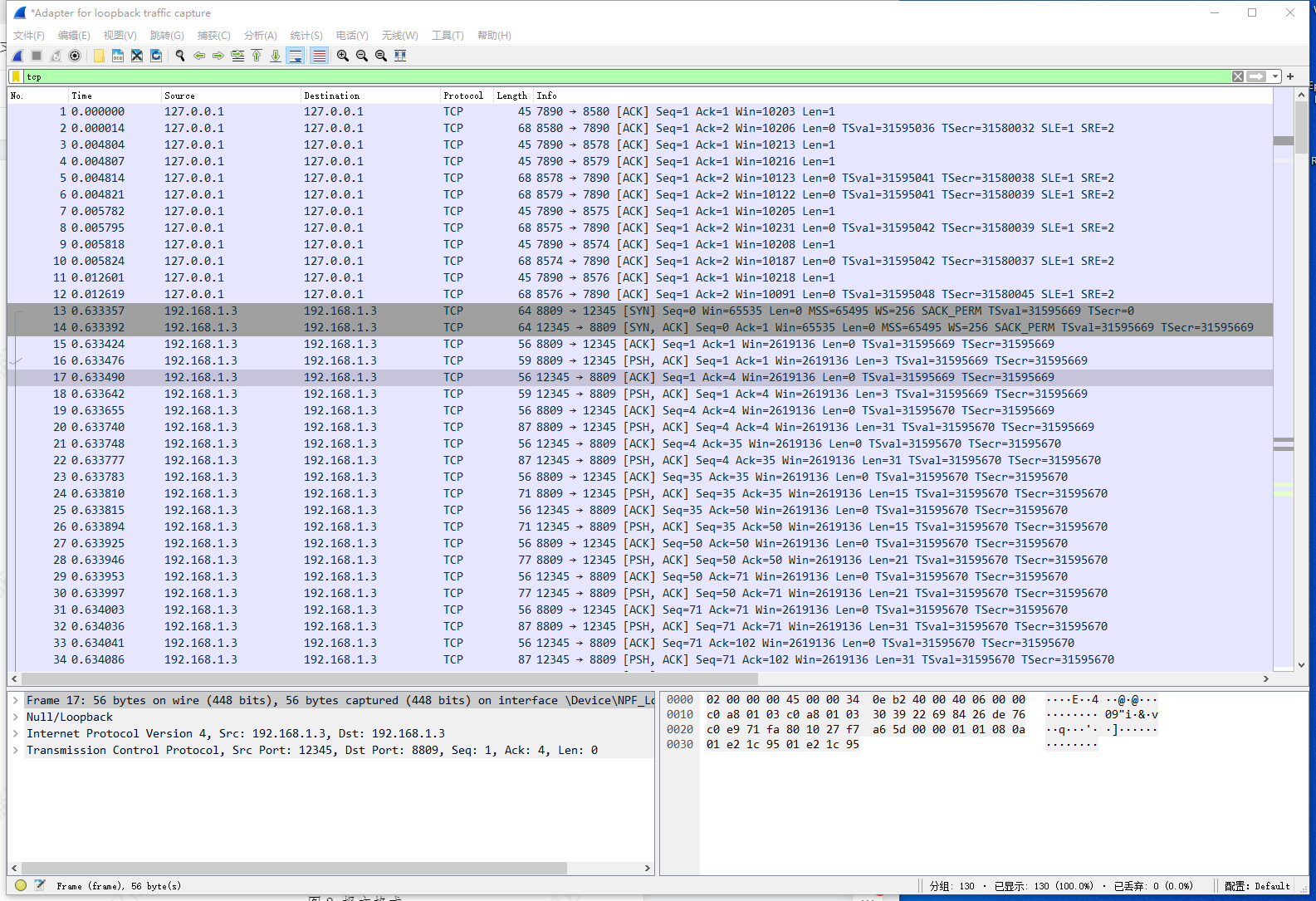
1. client和server之间共4种类型报文，Initialization、agree、reverseRequest、reverseAnswer。各报文封装如图2所示:



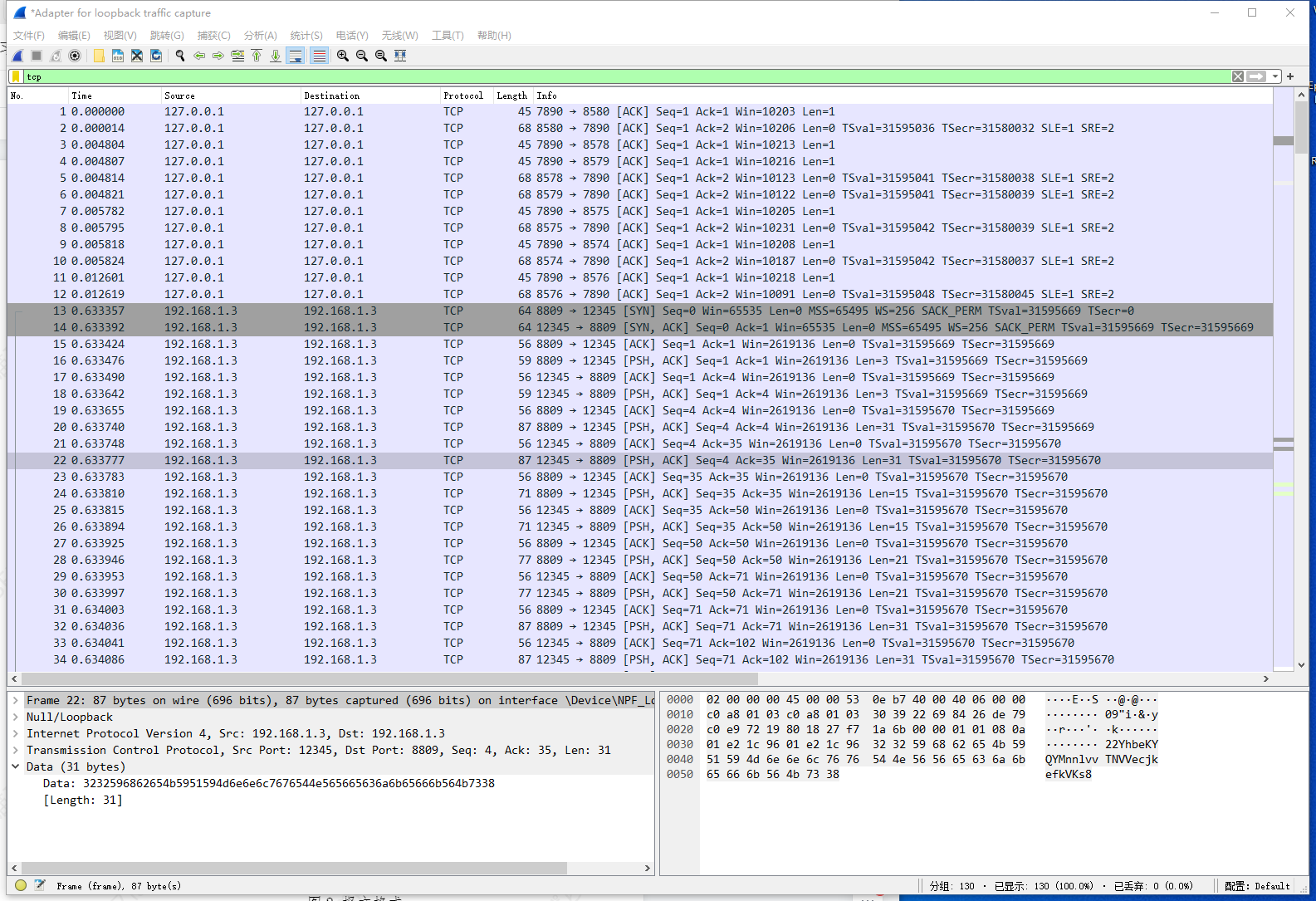
Initialization：



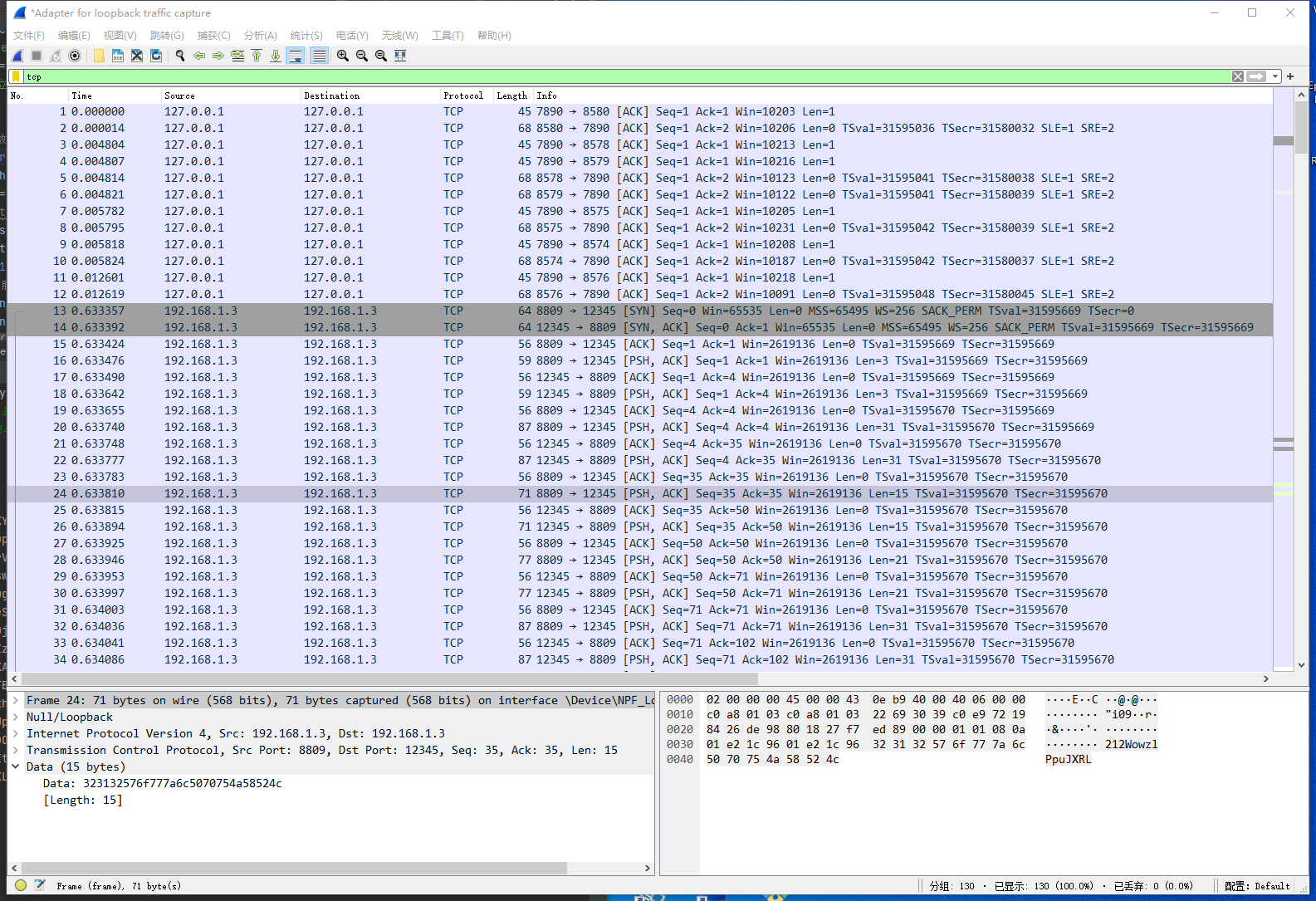
Agree:



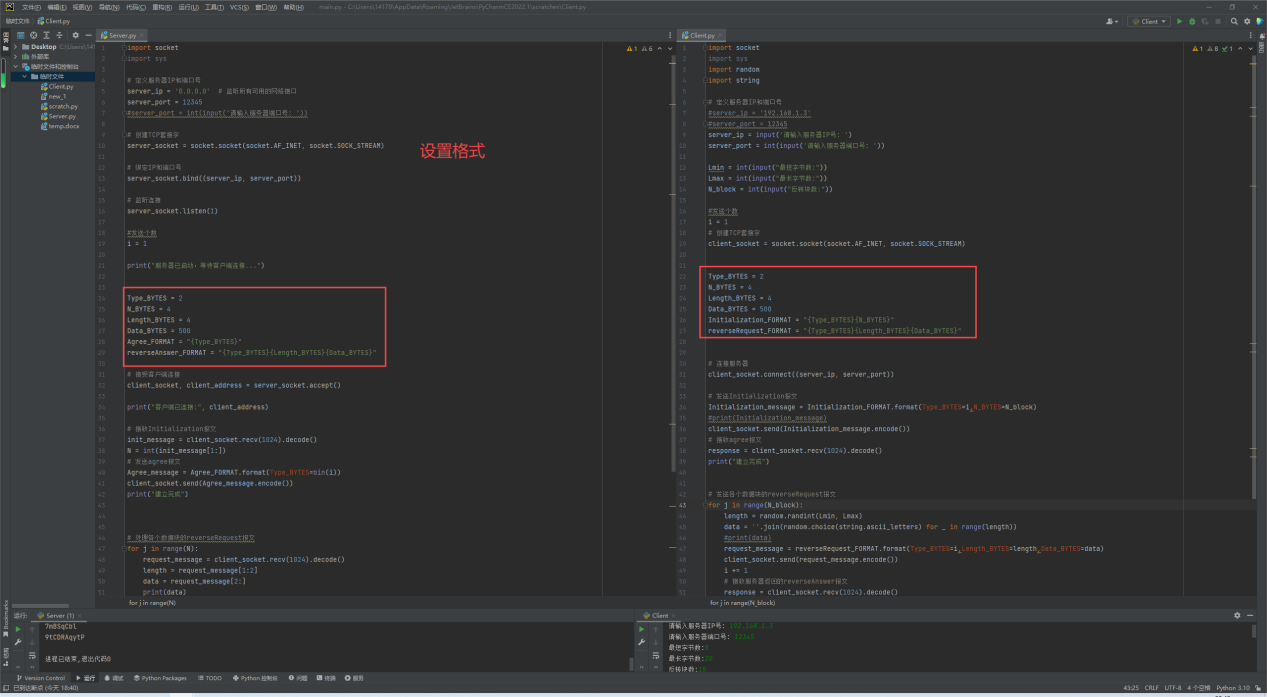
reserveRequest:



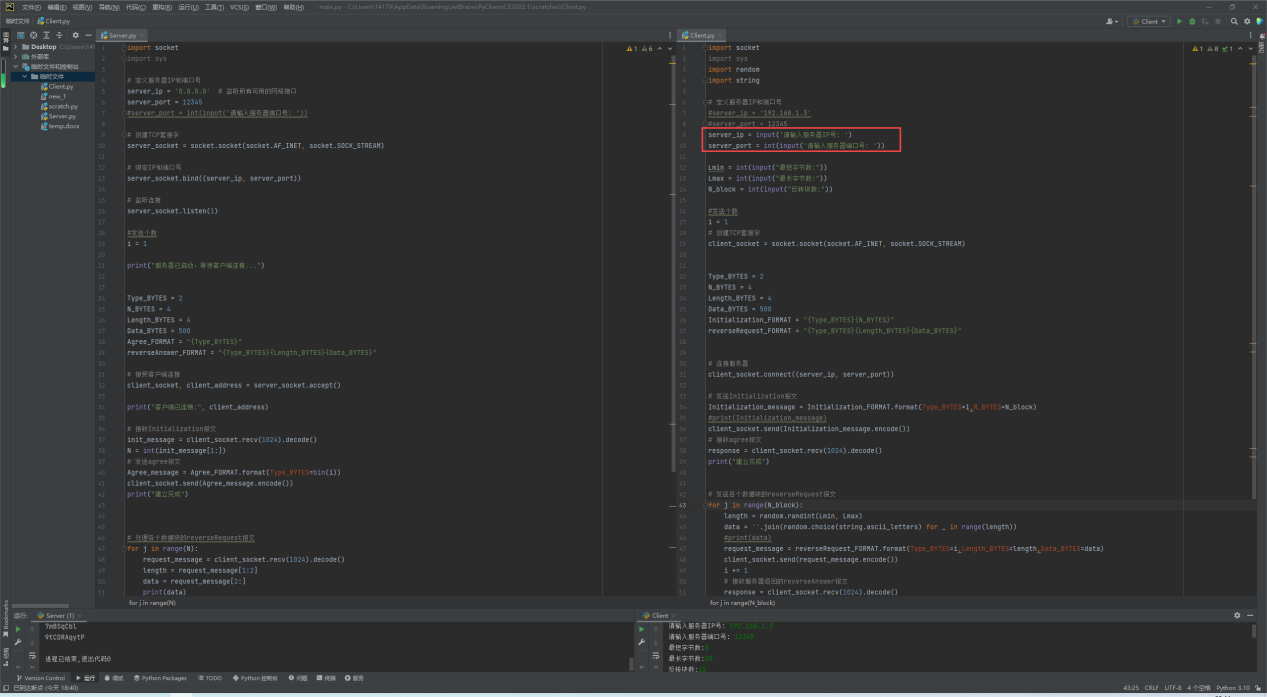
reserveAnswer:



1. 对图2的文字描述:四种报文类型，通过Type字段来区分。client 端向 server 端发送 Initialization 报文，告知 server 要做reverse的块数N，server回复agree报文。client 逐次向server 发送reverseRequest报文，包含了要做 reverse 的数据;server 端返回reverseAnswer报文，其中包含了reverse之后的数据。reverseRequest、reverseAnswer报文中Length字段分别指的是 Data和reverseData的长度，单位是Byte。



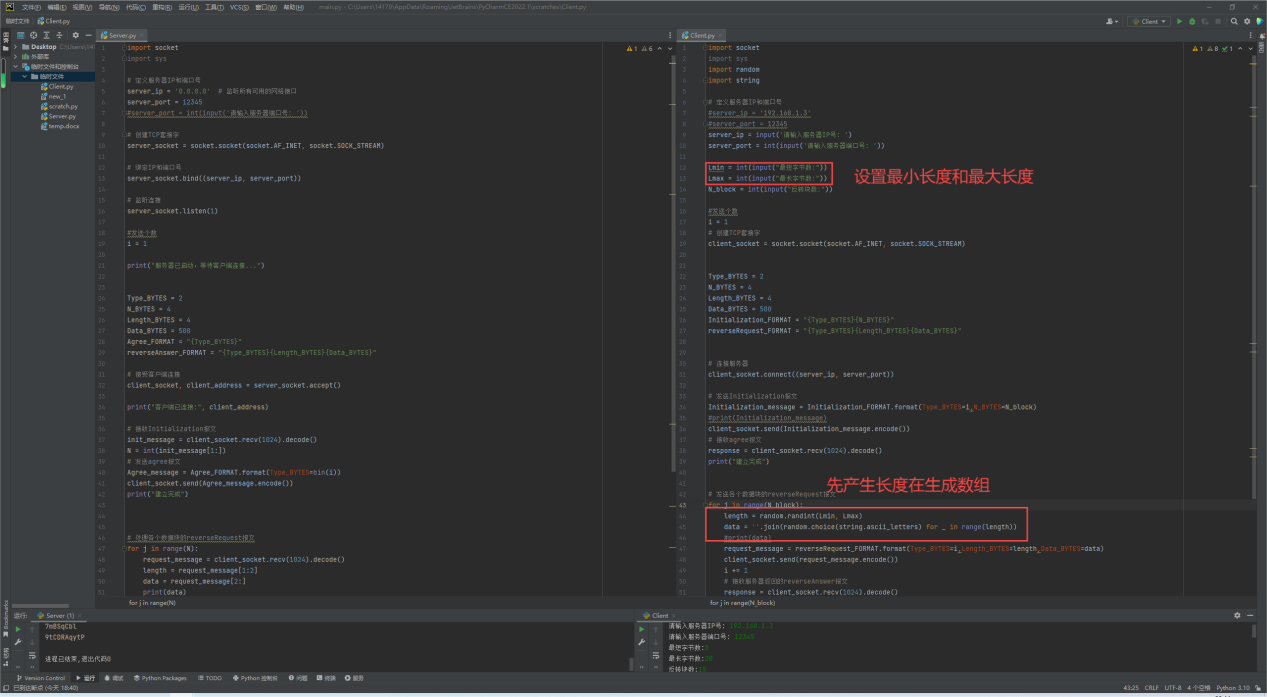
1. client 端在程序启动时，在命令行中指定serverIP、serverPort。



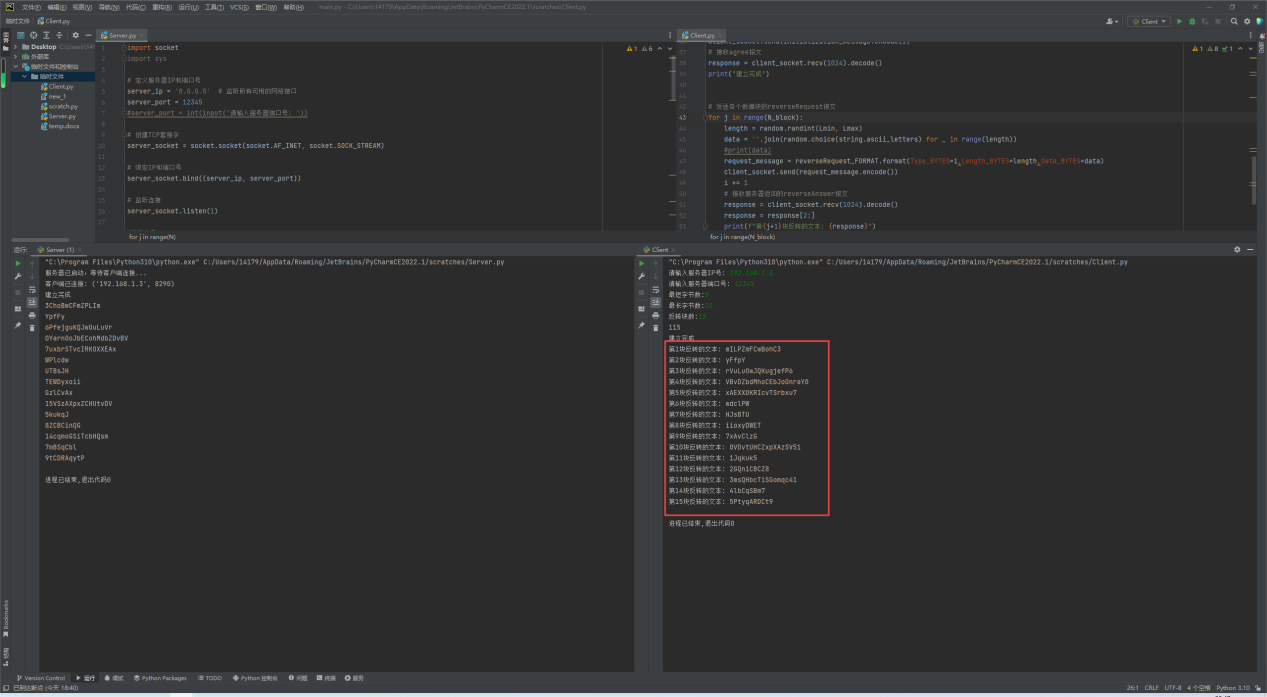
（7）client 每次发送的reverseRequest 中的长度都不一样(即不是例如每次固定发送500字节)，长度限定在一个[Lmin，Lmax]之间最后一块除外。

A.(推荐但非必须)Lmin、Lmax 在命令行中指定.

B.在开始发送数据块前，要先随机确定待发送File中的各块的 Bytes 长度，再计算块数N，才能执行”client->server的 Initialization 报文。



（8）每次client 收到来自server的reverseAnswer 报文，都在命令行(8)下打印出来“第x块:反转的文本”。如“8:yeknom elttil a。



1. (非必须实现，bonus项:)server能够同时处理2个及以上client的同时请求，即非阻塞，non-blocking。(处理non-blocking，有若干种实现方式，例如select、epoll等。)

## 3 关键点和对应解决方案

实现报文交互：客户端和服务器之间需要进行报文的发送和接收。可以使用TCP socket编程来实现报文的发送和接收操作。客户端可以使用socket库的函数来建立与服务器的连接，并通过套接字发送报文。服务器端可以监听来自多个客户端的连接，并接收和处理报文。

## 4 掌握的知识点

TCP是一种面向连接的协议，这意味着在通信之前，发送方和接收方需要建立一个连接。连接的建立经历了"三次握手"的过程，包括客户端发送连接请求，服务器回复确认，以及客户端再次回复确认。

TCP提供可靠的数据传输。它通过使用确认、重传和序列号等机制来确保数据的可靠性。接收方会确认已经接收到的数据，并且如果有任何数据丢失或损坏，发送方会重新发送这些数据。

## 5 git的url

https://github.com/nekomatapara/Task3-TCP-Socket-programming