# 计算机网络

# 实验报告

**实验名称： tcp抓包**

**实验地点： 233**

**实验日期： 2023.12.19**

**学生姓名： 李泽祥**

**学生学号： 21020007048**

# 一、实验目的

（1）通过实验熟悉 Wireshark 抓包软件的使用方法，理解 TCP 传输过程，以及慢启动、拥塞避免等相关技术。

（2）通过实验熟悉 Wireshark 抓包软件的使用方法，理解 TCP 传输过程，以及慢启动、拥塞避免等相关技术。熟悉在两台计算机之间进行TCP传输的过程。

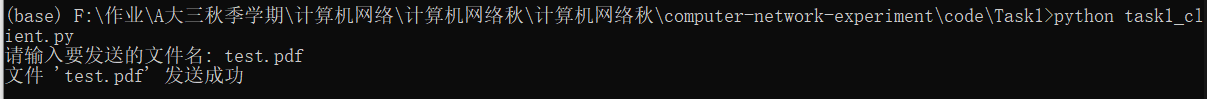
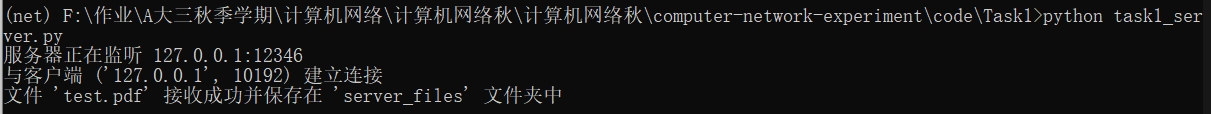
……

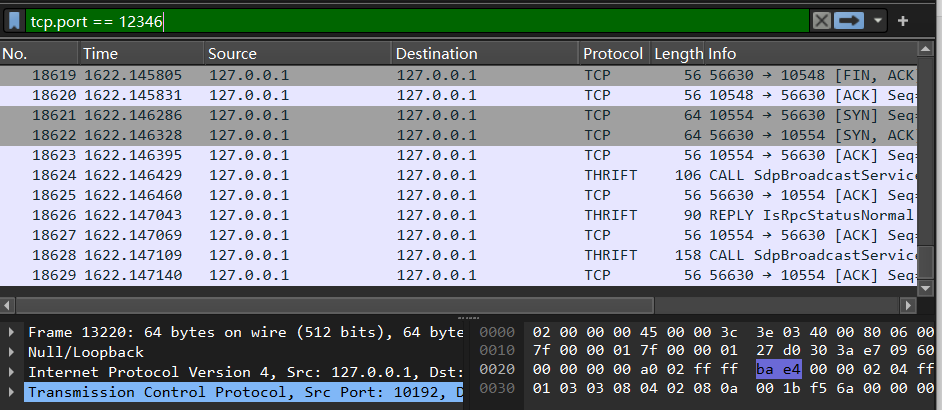
# 二、实验内容

### （一）实验一

说明你的实验步骤并提供截图证明

1. 第一步：安装wireshark，并配置python环境

2. 第二步：运行task1\_server.py监听12346端口，接着运行task1\_client.py实现建立连接成功发送test.pdf

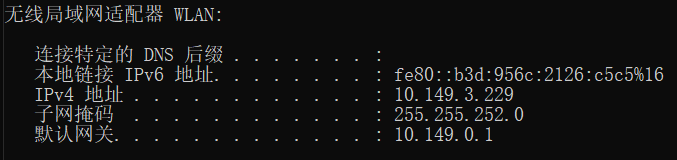
3. 第三步：用wireshark查看12346端口的tcp抓包情况  


……

### （二）实验二

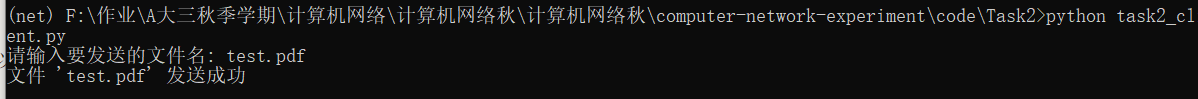
说明你的实验步骤并提供截图证明

组队：李泽祥、李迅

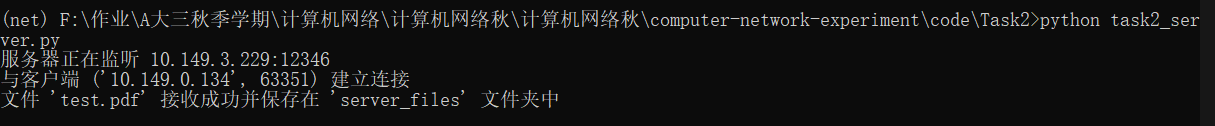
1. 第一步：查看校园网下的ip地址为10.149.3.229

2. 第二步：先修改client处的ip为同学的ip地址。然后我作为客户端，成功发送test.pdf



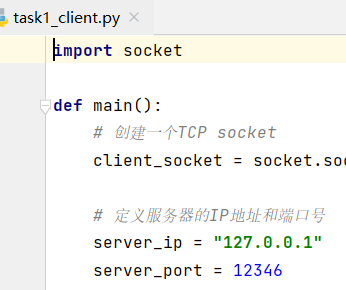


3. 第三步：先修改server.py处的ip为我校园网下的ip地址：我作为服务端，成功接收test.pdf



回答下列问题（提供截图证明）

1. 在第一个实验中的客户端程序发送数据的端口是多少，这个端口是谁来决定的？



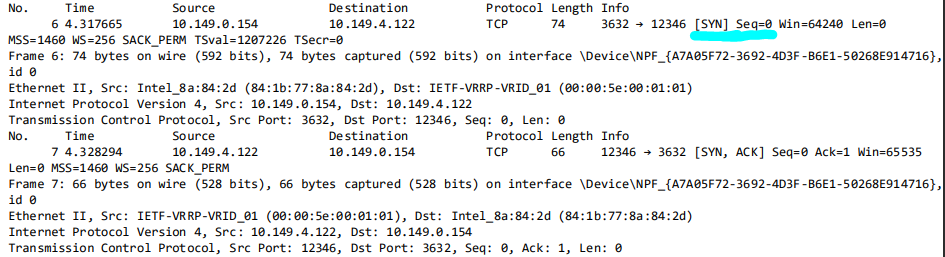
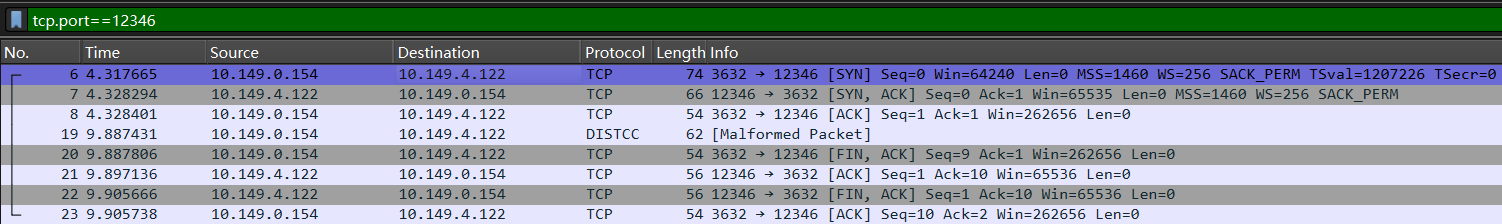
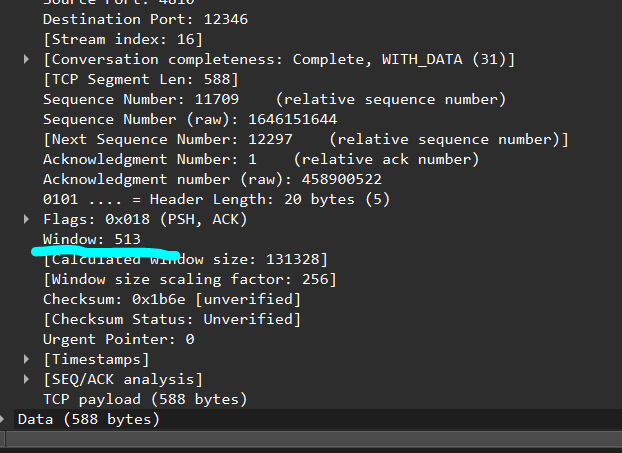
12346；我在python程序中决定的

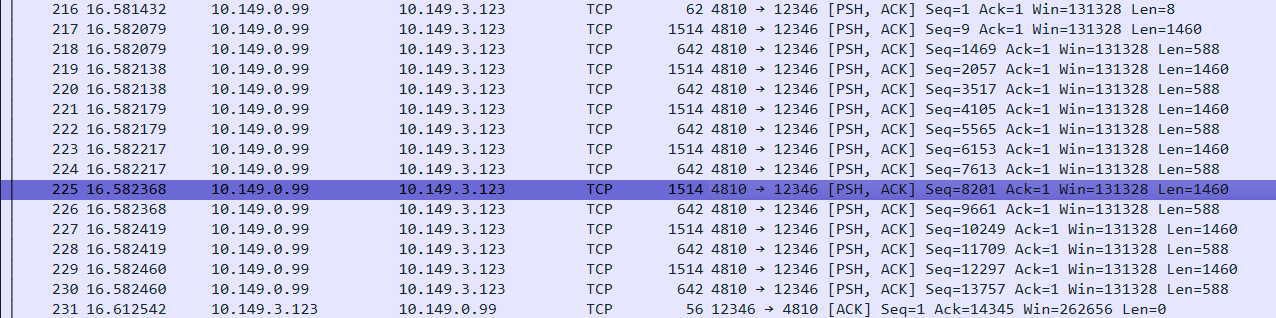
1. 在第二个实验中服务端程序和客户端程序的ip地址和端口号是什么？



客户端如图  


服务端如图

1. 为什么在第一个实验中不需要设置端口的防火墙的通行权限？  
   这两个程序之间的通信是基于本地主机（127.0.0.1）进行的，也就是在同一台计算机上运行的客户端和服务器程序之间的通信。由于它们位于同一台计算机上，所以不需要通过防火墙进行网络通信。防火墙主要用于控制网络流量和保护计算机免受未经授权的访问。当计算机上运行的程序尝试与外部网络进行通信时，防火墙通常会起到限制或允许该通信的作用。但是，在同一台计算机上运行的程序之间进行通信时，数据传输是在本地进行的，不需要通过网络接口进行传输。
2. 在第二个实验中，用来初始化客户端和服务端的 TCP 连接的TCP SYN 报文段的序号是什么？在报文段中，哪个地方表明这是一个 SYN 报文段？  
   （以下的ip和上面的不一样是因为又重新做了一遍，传了一个自己新建的temp.txt文件【无内容】）  
     
   ‘[SYN]’表示SYN报文段，序号是Seq=0
3. 开始的 6 个 TCP 报文段的长度各自是多少？  
     
   74、66、54、62、54、56、56、54
4. 在跟踪文件中，有重传的报文段么？回答这个问题，你需要检查哪个地方？  
   有重传；需要检查的是TCP报文段的INFO是否是RETRANSMISSION（或者说在wireshark中是否是红色的条）  
   
5. 接收方在一个 ACK 中，通常确认多少数据？你能辨别出这样一种情形吗：即接收方对收到的报文段，每隔一个确认一次？  
     
   确认513bit



采用延迟确认，将多个ACK合并为一个

接收方发送的ACK报文中Acknowledgement Number连续增加，但确认号并不是连续的。

接收方发送的ACK报文中，ACK标志位被设置为1，表示确认收到数据。

接收方发送的ACK报文中，确认号指示了已成功接收的连续数据块的结束位置。

1. 这个 TCP 连接的吞吐量（每单位时间传输的字节数）是多少？解释你是如何计算这个数值的？



开始时间：16.581432

  
结束时间：16.902431

Seq=726576

所以吞吐量：726576/(16.902431-16.581432)= 2,263,483.686865068

# 三、实验体会

可根据“实验思考”部分作答，也可以根据个人具体体会作答。

这次实验让我对上课讲过的tcp相关内容，有了更深的认识，也顺便巩固了一下上课学过的内容。对于滑动窗口、慢启动、拥塞避免等技术更明白了一些。

通过实验一，我安装了Wireshark，并配置了Python环境。然后我运行了task1\_server.py和task1\_client.py两个程序，成功建立了连接并发送了test.pdf文件。通过Wireshark查看12346端口的TCP抓包情况，我能够清楚地看到数据包的传输过程。

在实验二中，我与我的队友一起进行了实验。我们首先查看了校园网下的IP地址，并进行了相应的修改。我作为客户端成功发送了test.pdf文件，而我的队友作为服务端成功接收了文件。

回答问题方面，第一个实验中客户端程序发送数据的端口是12346，这个端口是我在Python程序中决定的。在第二个实验中，服务端程序的IP地址是10.149.3.229，端口号是12346；客户端程序的IP地址是我的同学的IP地址，端口号未提供。

在第一个实验中不需要设置端口的防火墙通行权限，是因为客户端和服务器程序之间的通信是基于本地主机（127.0.0.1）进行的，即在同一台计算机上运行的程序之间的通信，不需要通过网络接口进行传输。

在第二个实验中，用来初始化客户端和服务端的TCP连接的TCP SYN报文段的序号是0，并且在报文段中有一个标志位显示这是一个SYN报文段。

开始的6个TCP报文段的长度分别是74、66、54、62、54和56。而在跟踪文件中，存在重传的报文段，可以通过检查TCP报文段的INFO是否是RETRANSMISSION来确定。

接收方在一个ACK中通常确认513bit的数据。同时，我也能辨别出接收方对收到的报文段每隔一个确认一次的情形，这是采用了延迟确认的策略，将多个ACK合并为一个。

最后，根据实验数据，我计算了这个TCP连接的吞吐量为2,263,483.686865068（字节/秒），通过计算传输的数据量与传输时间的比值得出。