

课程名称： 网络工程实验

实验名称： 基于Windows平台的对等网络构建

学 院： 信息学院 1

班 级： 19计算机科学与技术 1

姓 名： 李泽昊 1

白文强

刘明浩

依　阳

赵浩杰

贾润堃

# 基于Windows平台的对等网络构建

摘　要

对等网络（P2P网络），即对等计算机网络，是一种在对等者（Peer）之间分配任务和工作负载的分布式应用架构。在P2P网络环境中，彼此连接的多台计算机之间都处于对等的地位，网络中的每一台计算机既能充当网络服务的请求者，又对其它计算机的请求做出响应，提供资源、服务和内容。

本次实验是基于P2P网络的构建和协议分析实验，具体内容主要包括拓扑图网络方案设计、虚拟仿真验证、有线及无线网络搭建和协议分析四个内容。首先，通过visio设计出对等网络的架构拓扑图，并通过Cisco Packet Tracer进行可行性的虚拟仿真验证。其次，完成了有线和无线方式的构建，在有线方式下，小组成员通过制作双绞线实现了两台计算机的对等网络的有线连接。在无线方式下，小组成员采用无线热点实现了对等网络的搭建，使得方案变得更加简捷。再次，小组成员通过wireshark对网络平台的物理层、数据链路层、网络层、运输层和应用层分别截获抓取数据包流量，并对各层进行逐层分析。在物理层上，小组成员通过观察插入网线的端口亮灯情况进行物理层方面的连通性验证。在数据链路层上，通过wireshark对数据链路层的数据包截获分析其源mac地址和目的mac地址。在网络层上，通过ping命令验证两台计算机之间能否完成正常通信。在运输层上，通过基于python的 Socket 编程实现多台主机之间消息的传递以及图片的共享传输。最后，在应用层上实现多台计算机之间的文件资源共享，验证SMB协议的连通性。

通过本次实验，小组成员熟练掌握了采用无线或者有线的方式搭建适用于多台计算机资源共享的对等网络平台。通过理论分析和实验的具体操作，我们发现P2P网络作为一种没有中心服务器的资源共享网络平台，具有分布式的容错能力。但是在该种网络模式中，缺乏对所有主机的集中管理，该类型的网络模式更适合对网络安全要求较低的小型网络组网。通过本次实验，小组成员提升了自己的动手能力和总结分析能力，更深刻体会到了团队协作的重要性。

关键词： P2P网络; CiscoPacketTracer; Socket; 协议分析;

Abstract

P2P network (P2P network) is a distributed application architecture that distributes tasks and workloads among peers. In P2P network environment, multiple computers connected to each other are in a peer position, each computer in the network can act as the requestor of network services, and respond to the request of other computers to provide resources, services and content.

This experiment is based on P2P network construction and protocol analysis experiment, the specific content mainly includes topology diagram network scheme design, virtual simulation verification, wired and wireless network construction and protocol analysis. Firstly, the architecture topology of p2p network is designed by Visio, and the feasibility of virtual simulation is verified by Cisco Packet Tracer. Secondly, the construction of wired and wireless mode was completed. In wired mode, the team members realized the wired connection of two computers through making twisted-pair cable. In wireless mode, the team members used wireless hot spots to build peer-to-peer networks, which made the scheme more concise. Thirdly, wireshark was used to capture data packet traffic at the physical layer, data link layer, network layer, transport layer, and application layer of the network platform and analyze each layer layer by layer. At the physical layer, team members verify the connectivity of the physical layer by observing that the port to which the network cable is inserted is on. On the data link layer, wireshark is used to capture and analyze the source AND destination MAC addresses of packets on the data link layer. At the network layer, you can run the ping command to check whether two computers can communicate with each other. In the transport layer, message transmission and image sharing are realized between multiple hosts through Socket programming based on Python. Finally, file resource sharing between multiple computers is realized on the application layer to verify the connectivity of SMB protocol.

Through this experiment, team members have mastered the use of wireless or wired way to build peer-to-peer network platform suitable for multiple computers to share resources. Through theoretical analysis and experimental operation, we find that P2P network, as a resource sharing network platform without a central server, has distributed fault tolerance. However, this network mode lacks centralized management of all hosts. Therefore, this network mode is more suitable for small-sized networks with low requirements on network security. Through this experiment, the team members improved their hands-on ability and summarizing and analyzing ability, and realized the importance of teamwork more deeply.

Key Words: P2P; CiscoPacketTracer; Socket; Protocol analysis; Wireshark

目　录

基于Windows平台的对等网络构建 1

一、实验目的 5

二、实验环境与器材 5

2.1 实验环境 5

2.2 实验器材 5

三、实验预习及思考 5

3.1 计算机的基本结构及工作原理 5

3.2 计算机网络相关知识 5

3.3 五类双绞线的制作标准 6

3.4 常用于网络测试的DOS命令 6

四、系统方案设计 7

4.1 局域网的拓扑设计 7

4.2 CiscoPacketTracer仿真拓扑图及设备配置 8

4.2.1 利用CiscoPacketTracer构建网络拓扑图 8

4.2.2 仿真过程及结果 9

五、设计方案与实现 9

5.1 有线构建 9

5.1.1 制作双绞线 9

5.1.2 实际网络搭建 10

5.2 无线构建 11

5.3 分层测试及分析 12

5.3.1 物理层分析 13

5.3.2 数据链路层分析 14

5.3.3 网络层分析 14

5.3.4 运输层分析 14

5.3.5 应用层分析 15

六、分析和思考 17

6.1 如何制定网络连接故障时排查的策略？ 17

6.2 网络连接的四个基本要素是什么？ 17

七、实验小结 19

八、参考文献 26

九、附录 27

9.1 消息传输（Client）代码 27

9.2 消息传输（Server）代码 27

9.3 图片传输（Client）代码 28

9.4 图片传输（Server）代码 29

## 一、实验目的

* 掌握5类双绞线的制作标准和方法（TIA/EIA568A和568B）
* 熟练掌握网卡和无线AP等网络设备的使用；
* 熟练掌握对等网的概念、特点和配置方法，要求基于Windows平台进行对等网的配置；
* 理解TCP/IP协议，掌握对等网络中IP地址的两种配置方式；
* 熟练掌握套接字编程、调试工具和WireShark协议分析工具。

## 二、实验环境与器材

### 2.1 实验环境

Windows 10、 Cisco 模拟器、 Python、WireShark

### 2.2 实验器材

五类UTP网线、压线钳、水晶头、笔记本电脑、手机。

## 三、实验预习及思考

### 3.1 计算机的基本结构及工作原理

* 计算机硬件由五个基本部分组成：运算器、控制器、存储器、输入设备和输出设备。
* 计算机采用二进制来表示程序和数据。
* 采用“存储程序”的方式，将程序和数据放入同一个存储器中（内存储器），计算机能够自动高速地从存储器中取出指令加以执行。

### 3.2 计算机网络相关知识

#### 对等网

（1）从网络计算机的从属关系来看

对等网中每台计算机都是平等的，没有主从之分。也就是说每台计算机在网络中既是客户机也是服务器

（2）从资源分布情况来看

对等网中的资源分布是在每一台计算机上的。其他类型的网络中,资源一般分布在服务器上，客户机主要是使用资源而不是提供资源。

（3）从作业的集中角度来看

对等网中的每一台计算机都是客户机,所以它要完成自身的作业，同时由于它们又作为服务器，因此要满足其它计算机的作业要求。从整体角度来看，对等网中作业也是平均分布的,没有一个作业相对集中的节点。

综上，对等网就是一种资源和作业都相对平均分布的局域网类型。

#### TCP/IP协议

TCP/IP传输协议，即传输控制/网络协议，也叫作网络通讯协议。它是网络使用中基本的通信协议。并且，TCP/IP传输协议是保证网络数据信息及时、完整传输的两个重要的协议。TCP/IP传输协议是严格来说是一个四层的体系结构，应用层、传输层、网络层和数据链路层都包含其中。

#### 无线局域网（WLAN）

WLAN是Wireless Local Area Network的简称，指应用无线通信技术将计算机设备互联起来，构成可以互相通信和实现资源共享的网络体系。无线局域网本质的特点是不再使用通信电缆将计算机与网络连接起来，而是通过无线的方式连接，从而使网络的构建和终端的移动更加灵活。

### 3.3 五类双绞线的制作标准

根据TIA/EIA-568，五类线使用RJ-45作为标准连接器，可使用T568A、T568B两种引脚排序，在网络施工中，建议使用T568B线序。

表1 T568A/B引脚排序

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| EIA/TIA-568A线序 | 白绿 | 绿 | 白橙 | 蓝 | 白蓝 | 橙 | 白棕 | 棕 |
| EIA/TIA-568B线序 | 白橙 | 橙 | 白绿 | 蓝 | 白蓝 | 绿 | 白棕 | 棕 |

在整个网络布线中应用一种布线方式，但两端都有RJ-45端口的网络无论是采用端接方式A，还是端接方式B，在网络中都是通用的。如果是计算机与交换机或HUB相连，则两头都做568A或两头都做568B。如果是两台计算机互连或ADSL MODEM与HUB相连，则需要一头做568A，另一头做568B。

### 3.4 常用于网络测试的DOS命令

* Ping命令
* Ping是个使用频率极高的实用程序，用于确定本地主机是否能与另一台主机交换（发送与接收）数据报。根据返回的信息推断TCP/IP参数是否设置的正确以及运行是否正常。
* Netstat命令
* Netstat用于显示与IP、TCP、UDP和ICMP协议相关的数据统计，一般用于检验本机各端口的网络连接情况。
* ARP命令
* ARP是一个重要的TCP/IP协议，并且用于确定对应IP地址的网卡物理地址。实用arp命令，能够查看本地计算机或另一台计算机的ARP高速缓存中的当前内容。
* Tracert命令
* Tracert命令可以用来跟踪数据报使用的路由。该实用程序跟踪的路径是源计算机到目的地的一条路径，不能保证或认为数据报总遵循这个路径。如果配置使用DNS，那么常常会从所产生的应答中得到城市、地址和常见通信公司的名字。
* Route命令
* Route命令是用来达到显示、人工添加和修改路由表项目的的命令。

## 四、系统方案设计

### 4.1 局域网的拓扑设计

基于对等网理论基础设计本实验，初步考虑用三台PC、交换机和手机（无线路由器）来组建网络

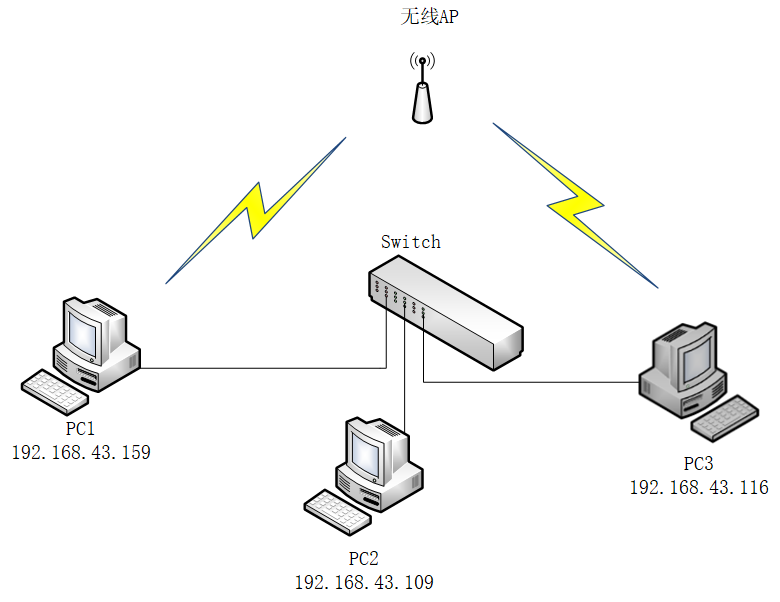


图1 网络拓扑图

表2 局域网网络配置表

|  |  |  |
| --- | --- | --- |
| 名称 | 无线IP地址 | 有线IP地址 |
| PC1 | 192.168.43.159 | 192.168.43.2 |
| PC2 | 192.168.43.109 | 192.168.43.3 |
| PC3 | 192.168.43.116 | 192.168.43.4 |
| 无线AP | 192.168.43.1 |  |

### 4.2 CiscoPacketTracer仿真拓扑图及设备配置

#### 4.2.1 利用CiscoPacketTracer构建网络拓扑图

无线连接拓扑图：

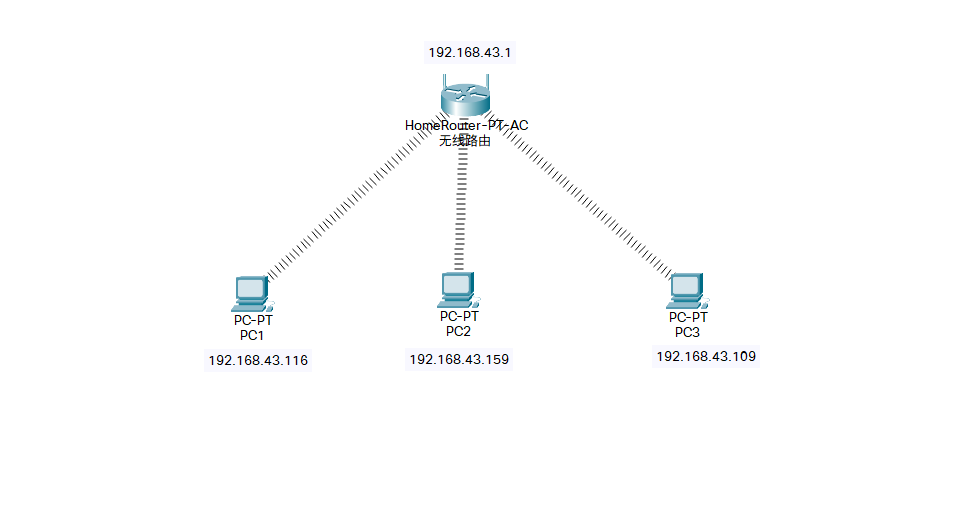


图2 CiscoPacketTracer 无线连接拓扑图

有线连接拓扑图：

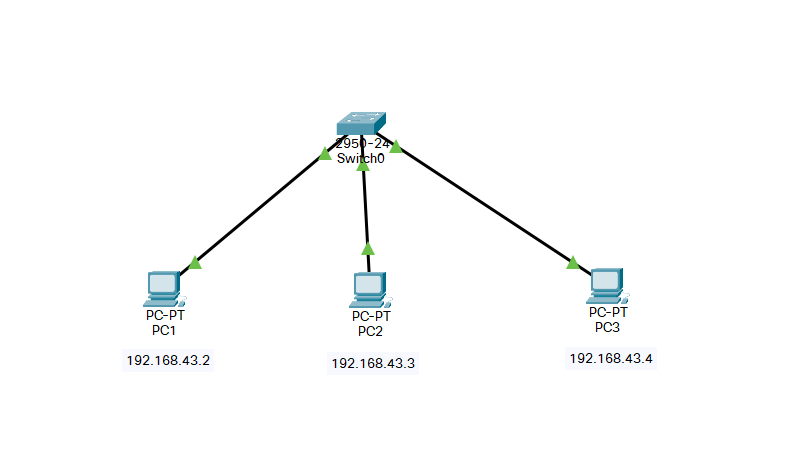


图3 CiscoPacketTracer 有线连接拓扑图

#### 4.2.2 仿真过程及结果

1.利用Cisco Packet Tracer软件，进行有线网络搭建及无线网络搭建的虚拟仿真。

2.利用简单报文工具验证主机间成功进行通信，发送简单报文时报文传输过程和结果如下：

无线连接：

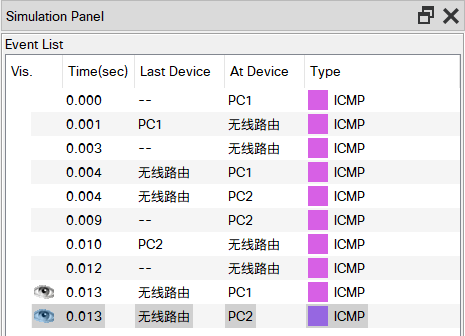


图4 无线连接条件下的ICMP报文交换过程

有线连接：



图5 有线连接条件下的ICMP报文交换过程

## 五、网络构建及分层测试

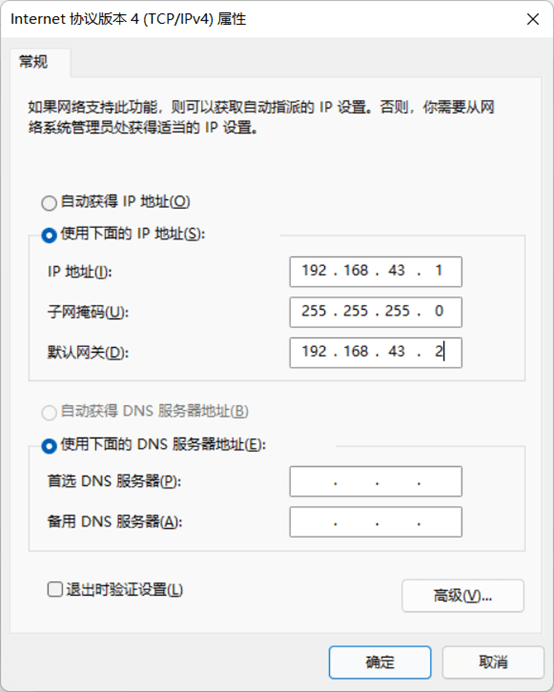
### 5.1 有线构建

#### 5.1.1 制作双绞线

* 用剥线刀将保护套管划开（小心不要将里面的双绞线的绝缘层划破），刀口距5类线的端头至少2厘米。
* 按照T568B的标准，即白橙、橙色、白绿、蓝色、白蓝、绿色、白棕、棕色的线序将导线按规定的序号排好。
* 将8根导线平坦整齐地平行排列，导线间不留空隙。
* 用压线钳的剪线刀口将8根导线剪断。
* 将剪断的电缆线放入RJ-45插头试试长短（要插到底，电缆线的外保护层最后应能够在RJ-45插头内的凹陷处被压实）。反复进行调整。
* 将RJ-45插头放入压线钳的压头槽内，用力压紧。
* 用同样的方式将双绞线的另一端制作完成。

#### 5.1.2 实际网络搭建

1. 实物连接图
2. 电脑配置：
   1. 打开两台有线连接的电脑网络和共享中心，更改适配器设置，将电脑的TCP/IPv4改为手动获取IP地址。并给两台设备设置IP地址



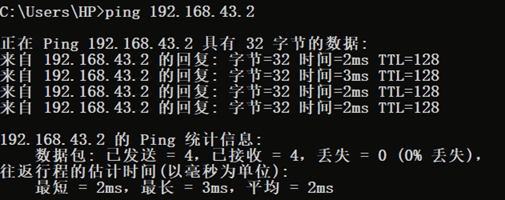
* + 图6 设备静态IP配置(1)

1. 第二台电脑进行类似设置，将上图的IP地址与默认网关对调一下即可



* 图7 设备静态IP配置(2)

1. 采用 ping 命令进行二者之间的通信，测试是否连通

* 
* 图8 ping命令测试网络连接状态

### 5.2 无线构建

利用手机热点组成一个局域网，并将三台设备连接到该网下，在三台设备之间进行消息传送与图片传输。

#### 5.2.1 查看当前设备IP地址

利用ipconfig/ifconfig 命令获取各设备的IP地址

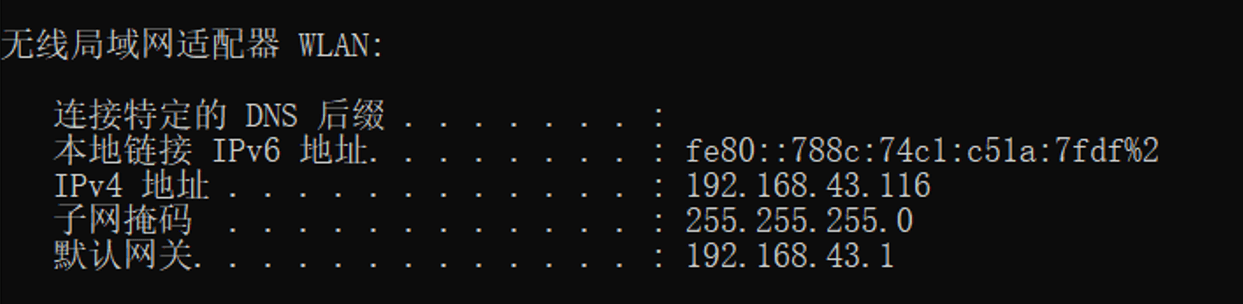


图9 查看设备IP地址(1)

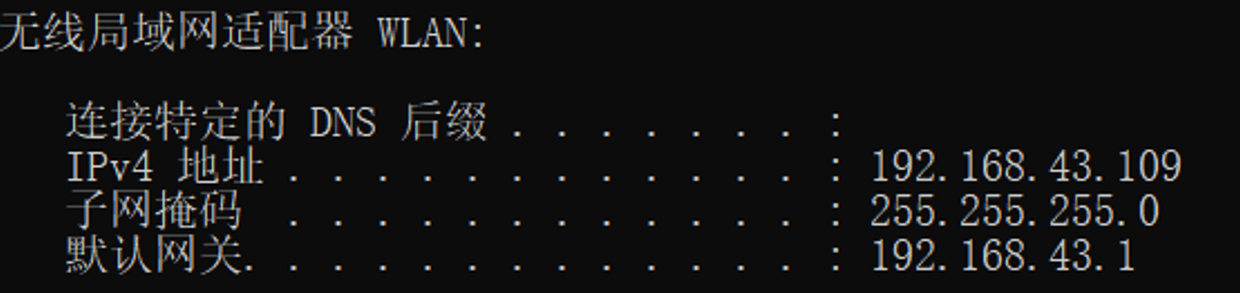


图10 查看设备IP地址(2)



图11 查看设备IP地址(3)

**5.3分层测试及分析**

进行socket编程，实现消息和图片的传输：

服务器端：192.168.43.159

客户端：192.168.43.116 和 192.168.43.109

在客户端192.168.43.116上连接服务器并发送消息"bwq,yyds"



图12 CLIENT端通信界面

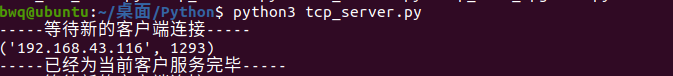


图13 SERVER端通信界面

利用wireshark软件进行数据包抓取，定向选择ip报文段，发现双方往返的报文段内容包括收发的文字段，连接成功。

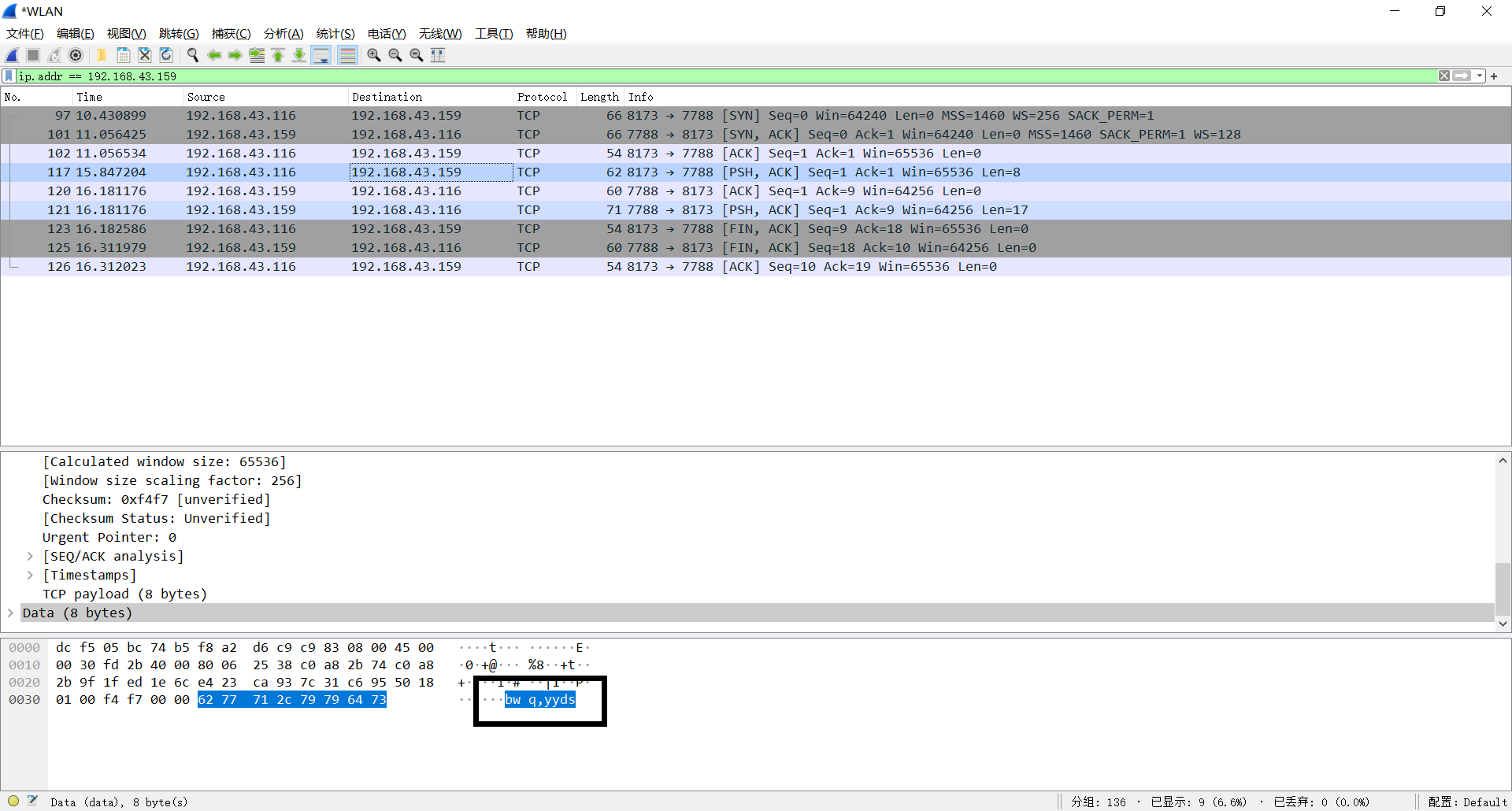
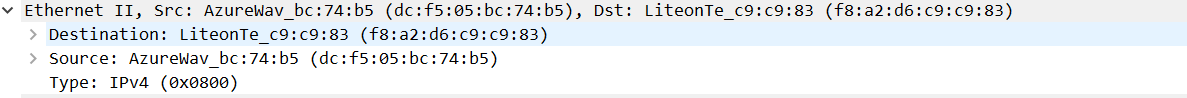


图14 Wireshark抓取发送信息数据包

5.3.1 物理层分析

* 
* 图17 数据包物理层信息分析
* 帧Frame 56表示要发送的数据块的帧号为56，捕获字节数等于传送字节数71字节。其中[Protocols in frame: eth:ip:tcp:data]表示协议帧：eth(以太网)、IP、tcp、data一项表明物理层传输向上用到的协议帧为eth、ip、tcp、data协议。

5.3.2 数据链路层分析

* 
* 图18 数据包数据链路层信息分析
* 数据链路层使用以太网，有线局域网技术，是数据链路层。其源MAC地址为(dc:f5:05:bc:74:b5), 目标MAC地址为(f8:a2:d6:c9:c9:83)。转换至的ip地址为ipv4标准

5.3.3 网络层分析

* 
* 图19 数据包网络层层信息分析
* Version: 4表示IP协议版本为IPv4；
* Source: 192.168.43.159表示源IP地址为192.168.43.159；
* Destination: 192.168.43.116表示目标IP地址为192.168.43.116；

5.3.4 运输层分析

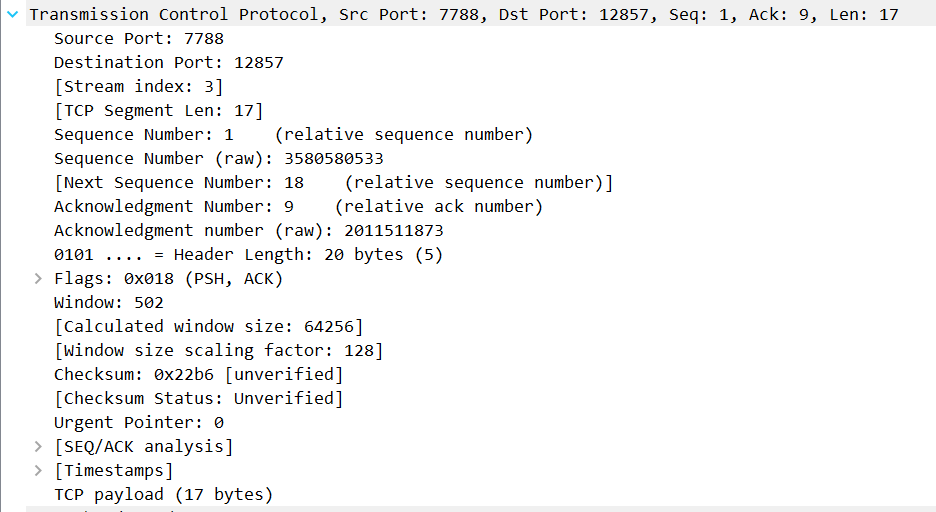
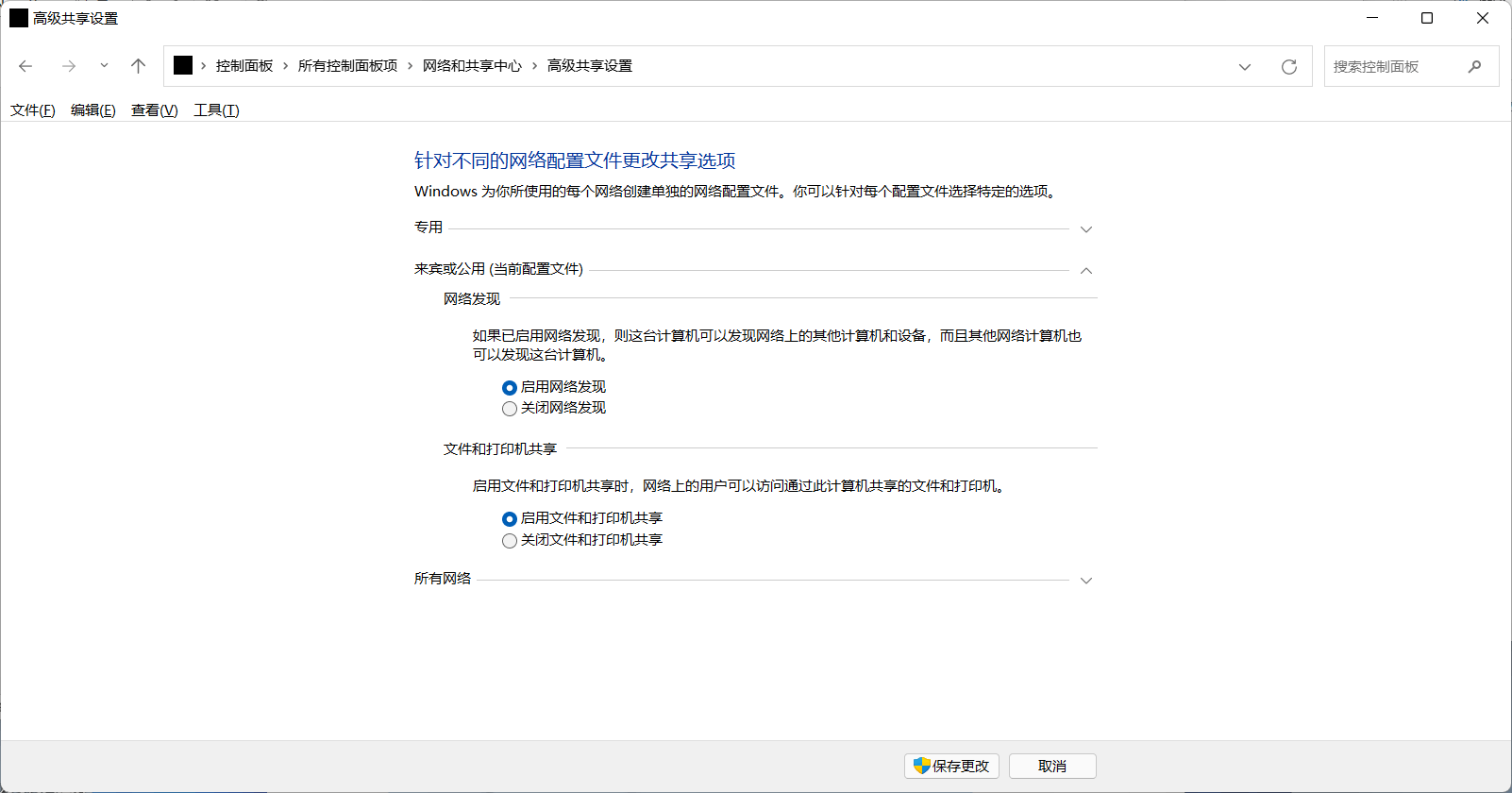
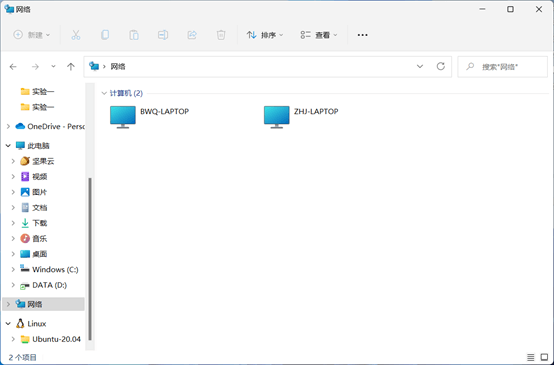
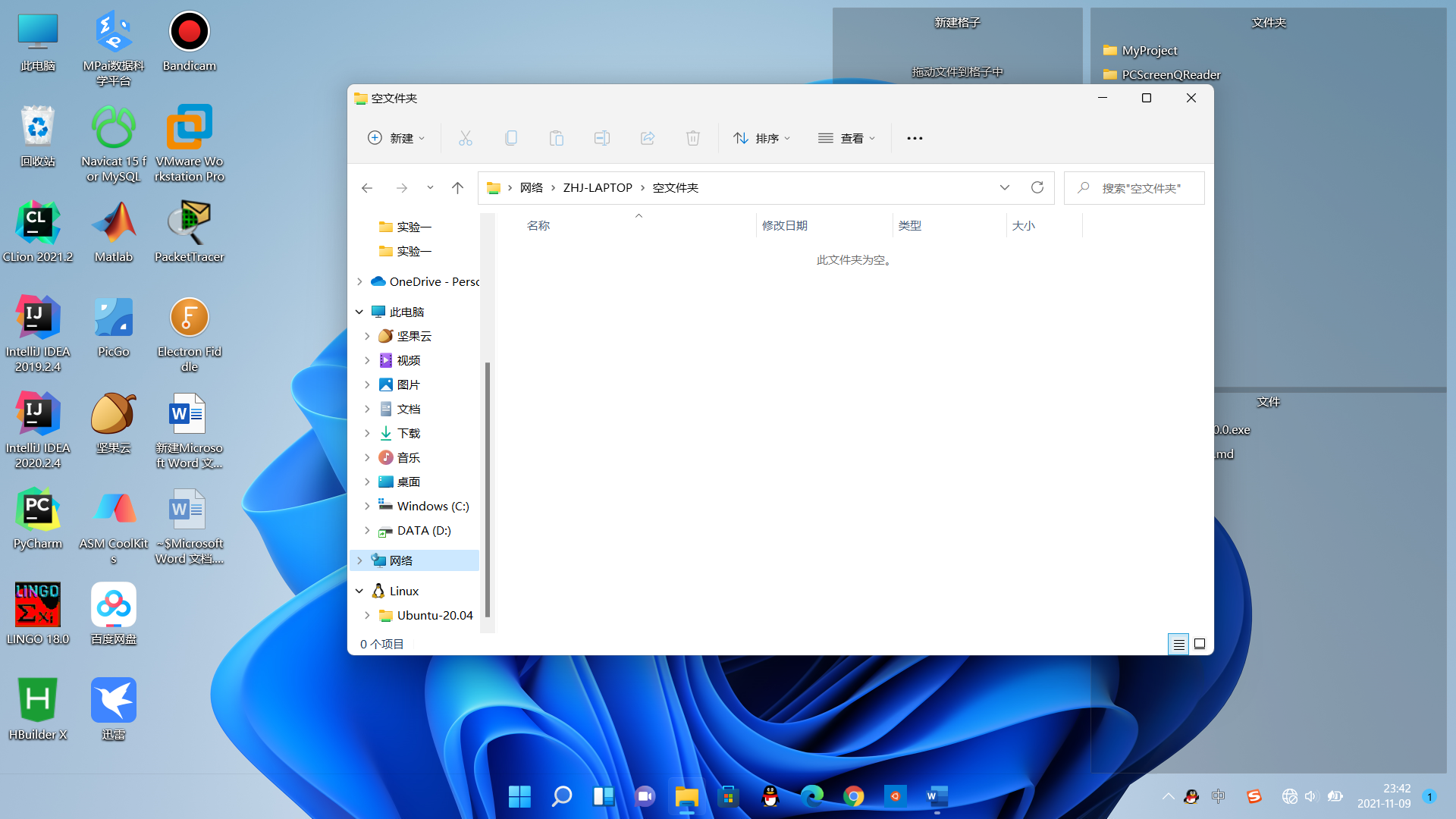
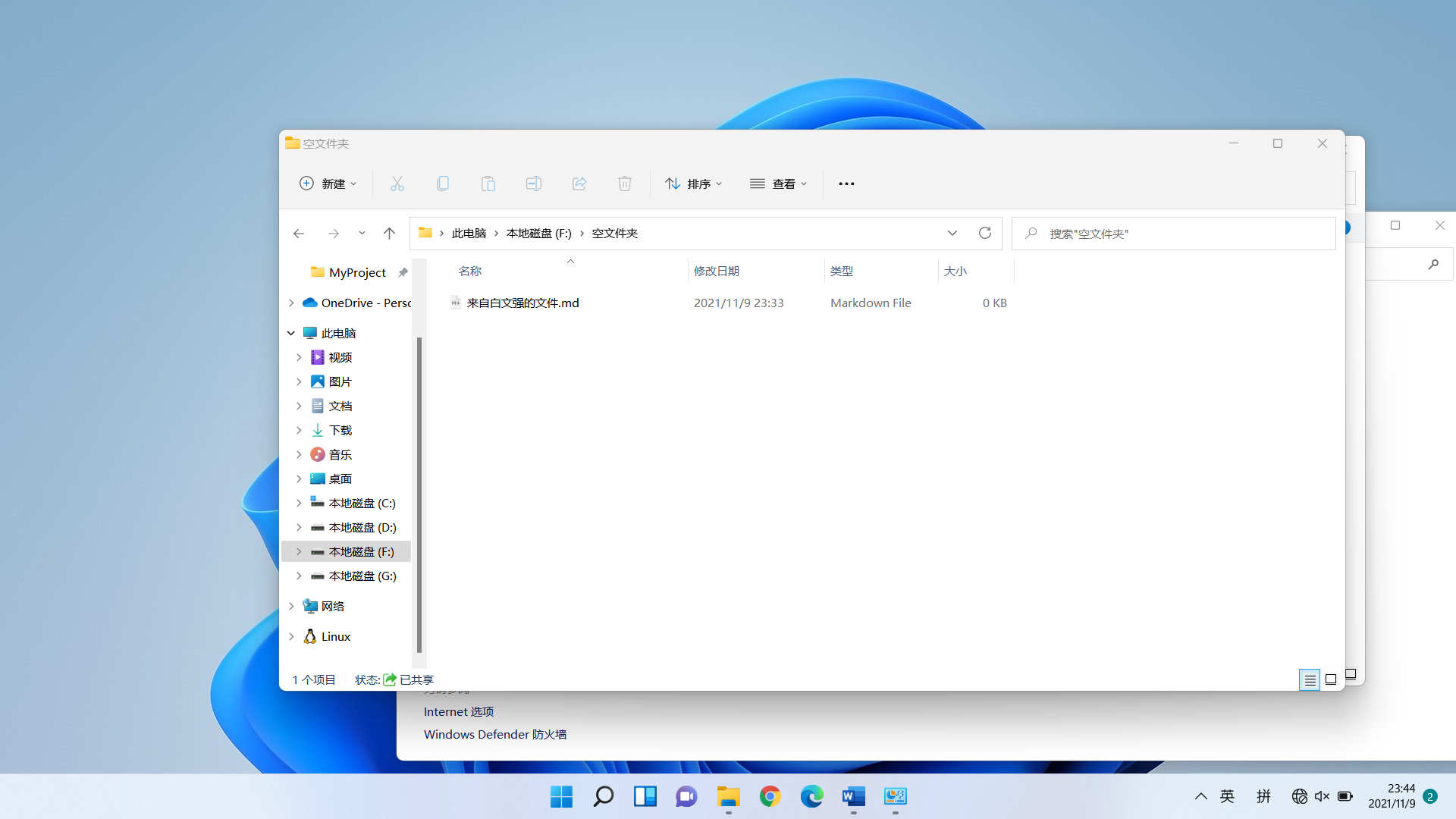
* 

图20 数据包传输层信息分析

Source Port表示连接的端口号为7788， Acknowledgment number是32位确认序列号，值等于1表示数据包收到，确认有效。

5.3.5 应用层分析

* 启用网络发现以及文件和打印机共享
* 
* 图21 网络发现配置
* 添加一个共享文件夹
* 在另一台电脑上，便可以发现该设备，
* 
* 图22 计算机“网络”界面
* 通过微软账号和密码可以进入ZHJ-LAPTOP，访问其共享的文件夹
* 
* 图23 BWQ-LAPTOP电脑上查看ZHJ-LAPTOP共享文件夹
* 可见，共享文件夹中没有文件，该设备共享了一个空文件夹。
* 在BWQ-LAPTOP上，向ZHJ-LAPTOP传输文件
* 在ZHJ-LAPTOP设备上，可以在共享文件夹中查看到BWQ-LAPTOP传送的文件。
* 
* 图24 传输文件后查看ZHJ-LAPTOP的共享文件夹
* 文件的传输成功

## 六、分析和思考

### 6.1 如何制定网络连接故障时排查的策略？

1. 检验电缆

　　检查计算机与网络插口之间的电缆。检查开路、短路和布线的工具。但是对于千兆以太网，还需要检查电缆中是否存在串扰和阻抗故障。

1. 确认连接到交换机

* 将便携式网络工具连接到办公室线路，检查是否可以建立链路，端口被管理员关闭，则工具将无法连接。

1. 申请DHCP地址

* 建立链路后可使用工具从服务器中申请DHCP地址，确保分配的地址与相应的子网相符，检验子网掩码，确认默认网关和DNS服务器地址。通过ES2网络通来检验是否能够正常接入网络，并顺顺利获得IP地址。

1. Ping网络上的设备

* 一旦获取 DHCP地址之后，即可使用工具ping台局域网外的设备，可确认DHCP服务器的的指定配置是否正确以及网络流量是否被正确的路由发送。

1. 检验速度/双工模式设置

* 某些链路性能故障如双工模式不匹配、速度不匹配以及静态配置IP地址等都只能进行在线监测。确认所连接交换机端口的速度设置和双工模式设置与相应计算机设置是否相。如果检测到双工模式不匹配，则确认计算机和交换机端口是否均已被设成自动协议。

1. 网络流量监控

* 通过在线工其检查计算机是否成功地向DHCP服务器申请并接收到地址同网络建立独立的连接，然后再与计算机建立连接，通过工具监控网络流量，查找过多的广播、冲突或错误。

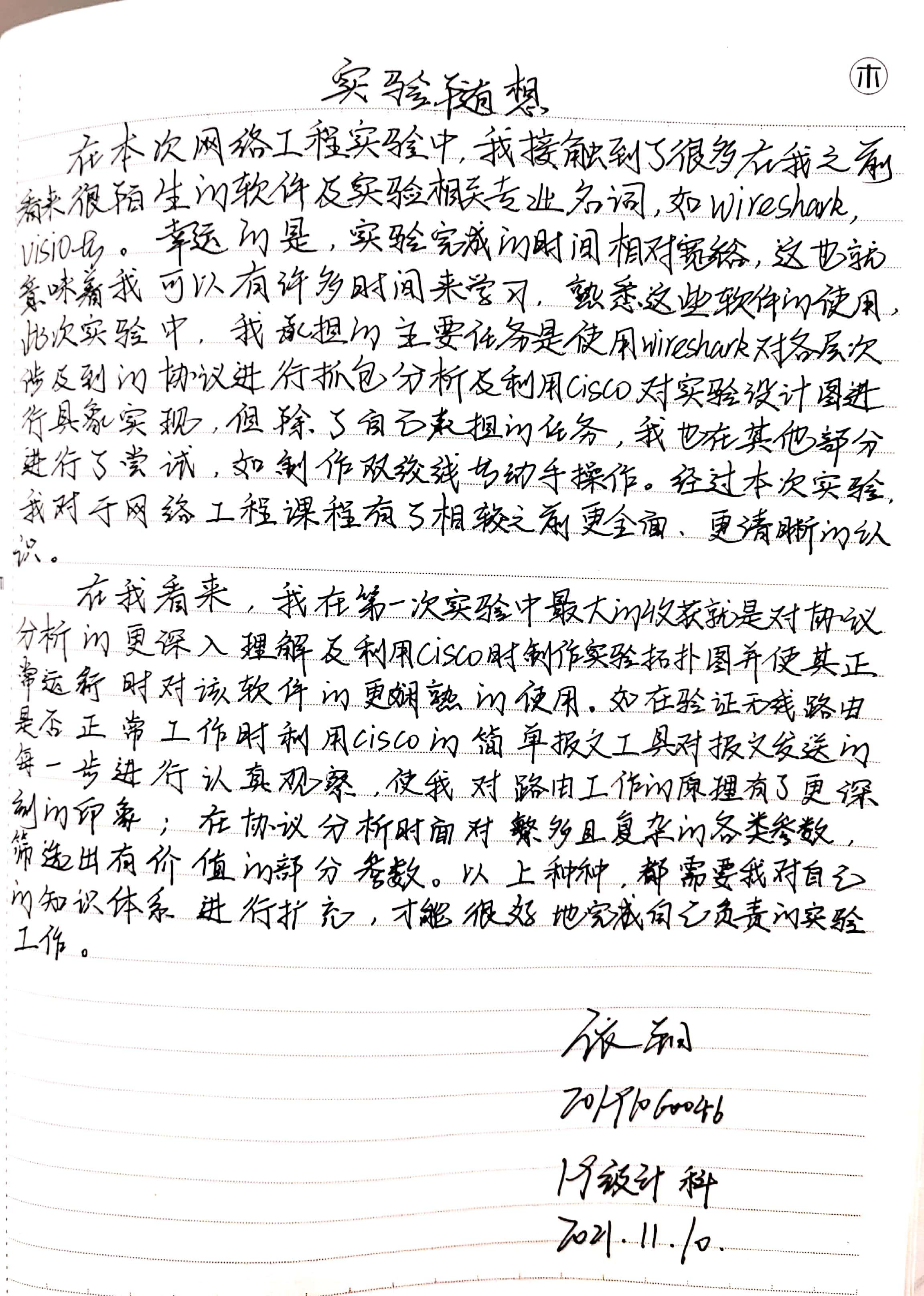
### 6.2 网络通信的三个基本要素是什么？

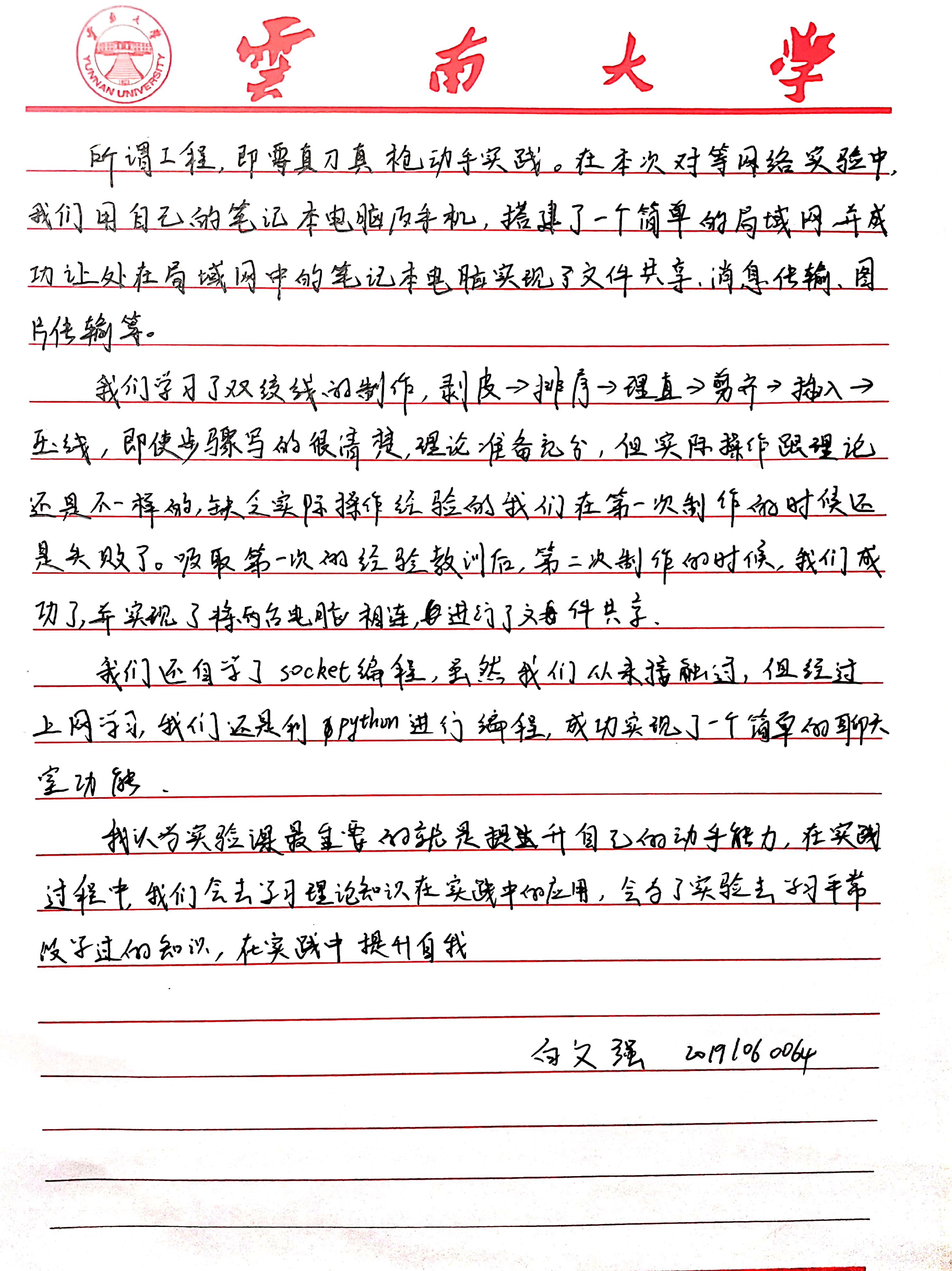
　　网络通信的三个基本要素有IP，端口，传输协议。

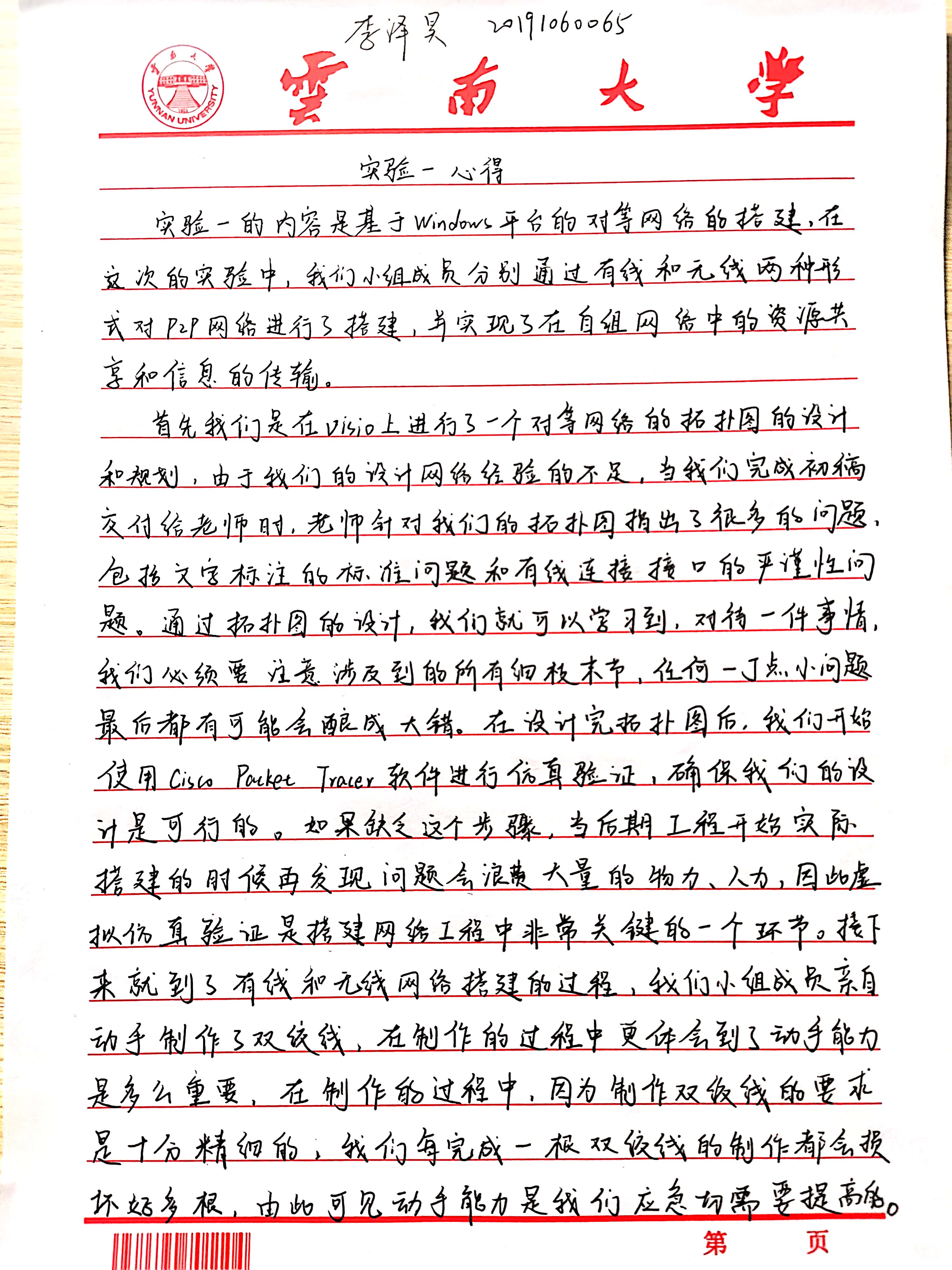
1. IP：电子设备（计算机）在网络中唯一的标识，是连入网络的识别地址。

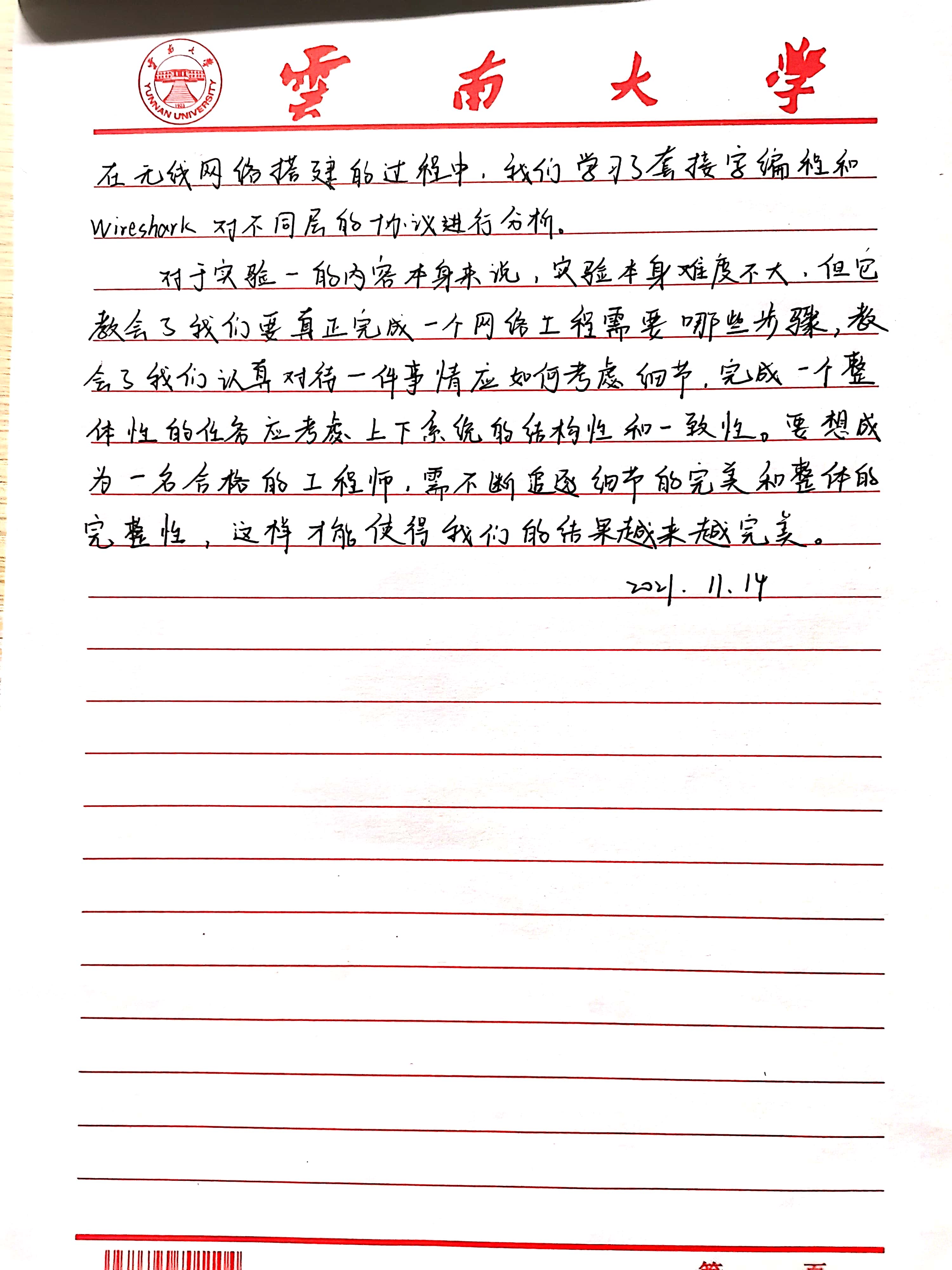
2. 端口：应用程序在计算机中唯一标识，想访问目标计算机中的某个应用程序，还需要指定端口。  
3. 传输协议：规定数据传输规则。网络通信必须遵从一定的连接协议，才能让连接端在网络连接时遵从共同规则，进行网络资源的分配。

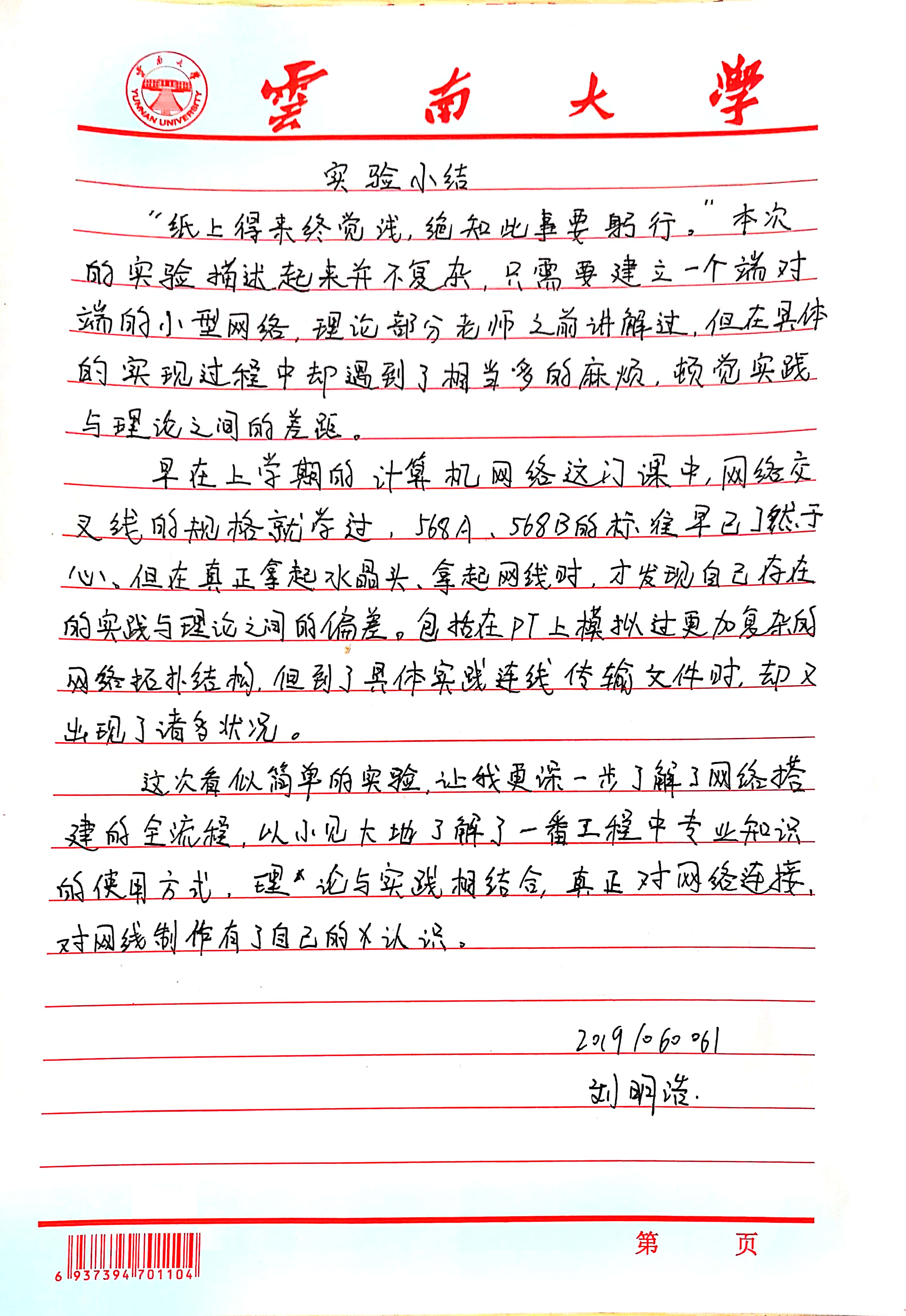
## 七、实验小结

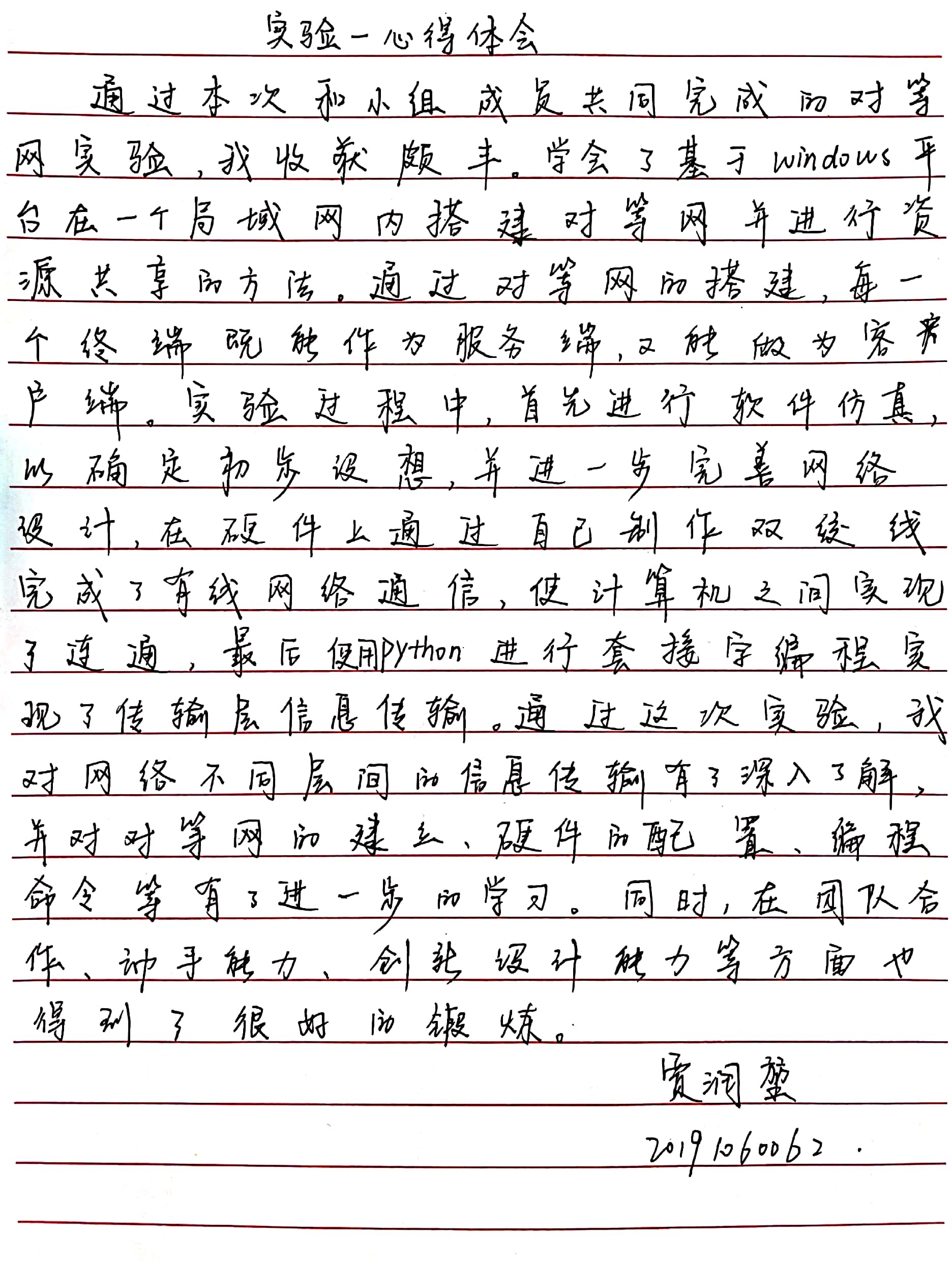


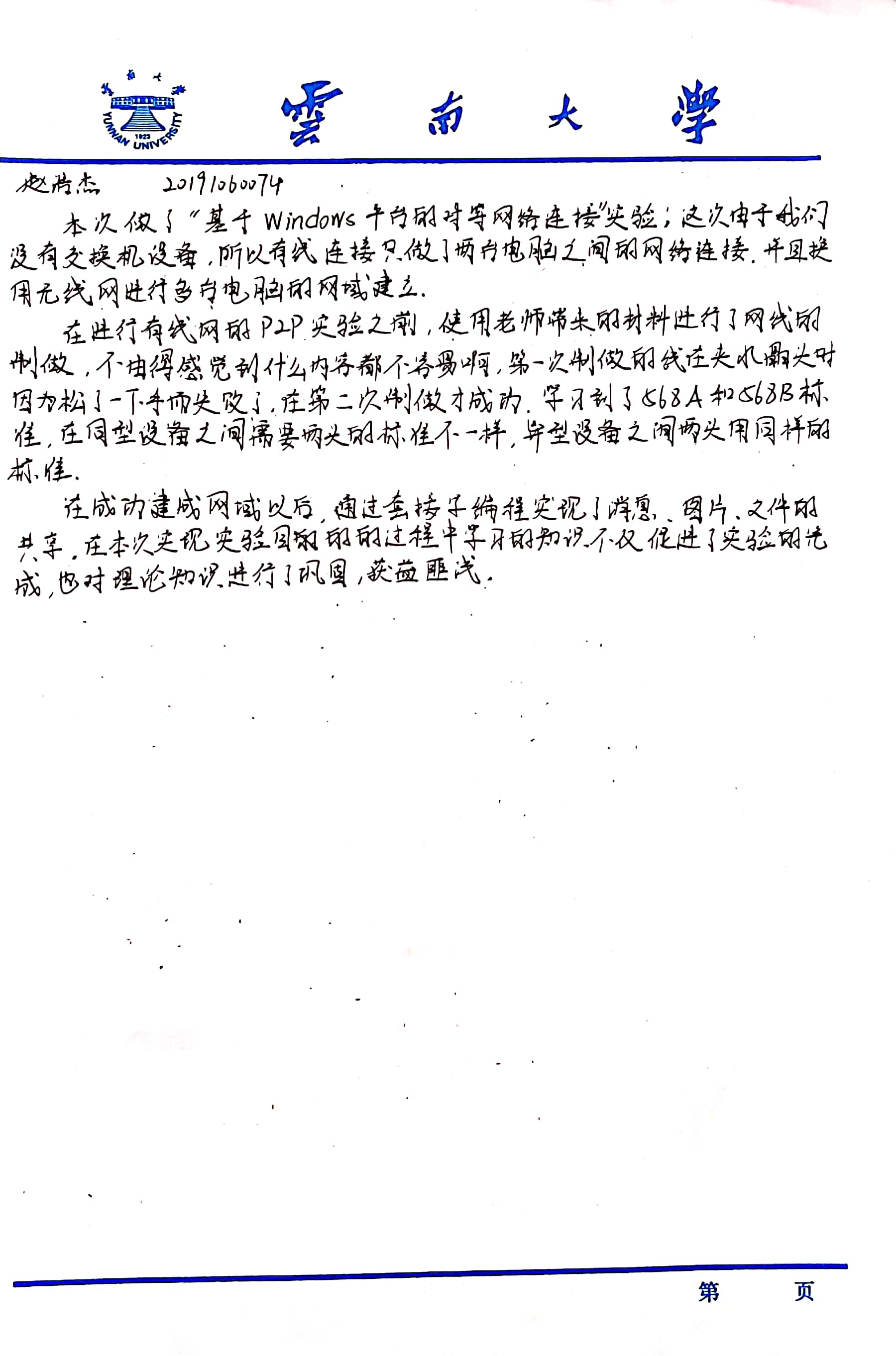












## 八、参考文献

[1]张春红. P2P 技术全面解析[M]. 人民邮电出版社, 2010.

[2]Zhang Yang et al. Research of Queuing Model Based on Request Queue in P2P Network System[J]. Radioelectronics and Communications Systems, 2021, 64(4) : 174-188.

[3]Afuwape Afeez Ajani et al. Performance Evaluation of Secured Network Traffic Classification using a Machine Learning approach[J]. Computer Standards & Interfaces, 2021, : 103545-.

[4]Shuai Zhou and Ting Zhang and Xianfu Meng. iForest : An informed resource search strategy in mobile P2P networks[J]. Peer-to-Peer Networking and Applications, 2021, : 1-16.

[5]https://wenku.baidu.com/view/ed924673f18583d04864590d.html

## 九、附录

### 9.1 消息传输（Client）代码

# 客户端  
  
import socket  
import os  
import struct  
  
def main():  
 # 创建套接字  
 tcp\_client\_socket = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)  
   
 # 目的信息  
 server\_ip = input('请输入服务器ip:')  
 server\_port = int(input('请输入服务器端口:'))  
  
 # 链接服务器  
 tcp\_client\_socket.connect((server\_ip,server\_port))  
 # 提示用户输入数据  
 send\_data = input("input your data:")  
 tcp\_client\_socket.send(send\_data.encode('utf-8'))  
 # 接收对方发过来的数据，最大接收1024字节  
 recvData = tcp\_client\_socket.recv(1024)  
 print('收到数据：',recvData.decode('utf-8'))  
  
 tcp\_client\_socket.close()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

### 9.2 消息传输（Server）代码

import socket  
  
  
def main():  
 # 创建套接字  
 tcp\_server\_socket = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)  
   
 # 绑定本地信息  
 tcp\_server\_socket.bind(("", 7788))  
   
 # 让默认的套接字由主动变为被动  
 tcp\_server\_socket.listen(128)  
   
 while True:  
 print('-----等待新的客户端连接-----')  
 # 等待客户端连接  
 new\_server\_socket, client\_addr = tcp\_server\_socket.accept()  
 print(client\_addr)  
 while True:  
 # 接收客户端发送来的请求  
 # f=open('1.jpg', 'wb')  
 recv\_data = new\_server\_socket.recv(1024)  
 # print(recv\_data)  
 if recv\_data:  
 # f.write(recv\_data)  
 # print(recv\_data.decode('utf-8'))  
 new\_server\_socket.send("hello here is bwq".encode('utf-8'))  
 else:  
 break  
  
 # 关闭套接字  
 print('-----已经为当前客户服务完毕-----')  
 new\_server\_socket.close()  
  
 tcp\_server\_socket.close()  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

### 9.3 图片传输（Client）代码

import socket  
import os  
import sys  
import struct  
  
def sock\_client\_image():  
 while True:  
 try:  
 s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 s.connect(('192.168.43.159', 7788)) #服务器和客户端在不同的系统或不同的  
 except socket.error as msg:  
 print(msg)  
 print(sys.exit(1))  
 filepath = input('input the file: ') #输入当前目录下的图片名 xxx.jpg  
 fhead = struct.pack(b'128sq', bytes(os.path.basename(filepath), encoding='utf-8'), os.stat(filepath).st\_size) #将xxx.jpg以128sq的格式打包  
 s.send(fhead)  
  
 fp = open(filepath, 'rb') #打开要传输的图片  
 while True:  
 data = fp.read(1024) #读入图片数据  
 if not data:  
 print('{0} send over...'.format(filepath))  
 break  
 s.send(data) #以二进制格式发送图片数据  
 s.close()  
 # break #循环发送  
   
if \_\_name\_\_ == '\_\_main\_\_':  
 sock\_client\_image()

### 9.4 图片传输（Server）代码

import socket  
import os  
import sys  
import struct  
  
def socket\_service\_image():  
 try:  
 s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 s.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)  
 s.bind(('', 7788))  
 s.listen(10)  
 except socket.error as msg:  
 print(msg)  
 sys.exit(1)  
  
 print("Wait for Connection.....................")  
  
 while True:  
 sock, addr = s.accept() #addr是一个元组(ip,port)  
 deal\_image(sock, addr)  
   
   
def deal\_image(sock, addr):  
 print("Accept connection from {0}".format(addr)) #发送端的ip和端口  
  
 while True:  
 fileinfo\_size = struct.calcsize('128sq')  
 buf = sock.recv(fileinfo\_size) #接收图片名  
 if buf:  
 filename, filesize = struct.unpack('128sq', buf)  
 fn = filename.decode().strip('\x00')  
 new\_filename = os.path.join('./', 'new\_' + fn)   
 recvd\_size = 0  
 fp = open(new\_filename, 'wb')  
  
 while not recvd\_size == filesize:  
 if filesize - recvd\_size > 1024:  
 data = sock.recv(1024)  
 recvd\_size += len(data)  
 else:  
 data = sock.recv(1024)  
 recvd\_size = filesize  
 fp.write(data) #写入图片数据  
 fp.close()  
 sock.close()  
 break  
   
if \_\_name\_\_ == '\_\_main\_\_':  
 socket\_service\_image()