

# 雲南大學

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

班 级： 19 计算机科学与技术

姓 名： 李泽昊

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<sup>[1]</sup>

（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<sup>[3]</sup>

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

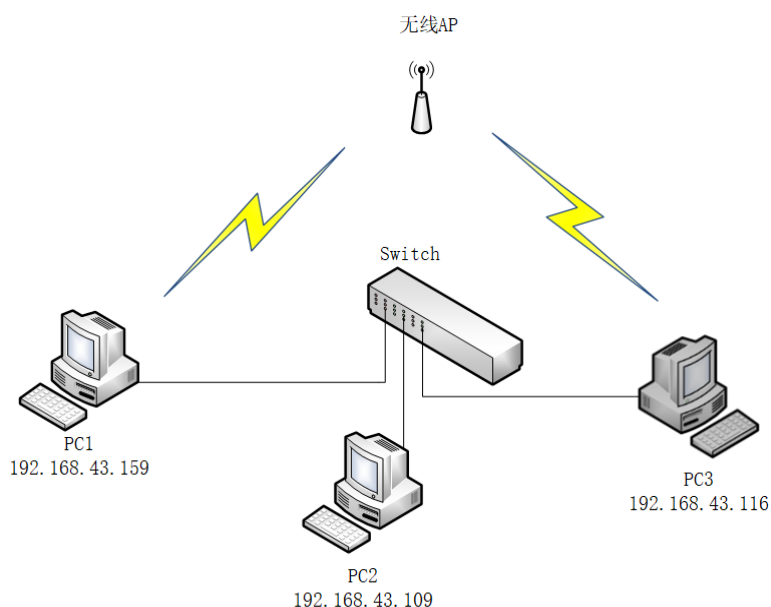


图 1 网络拓扑图



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：

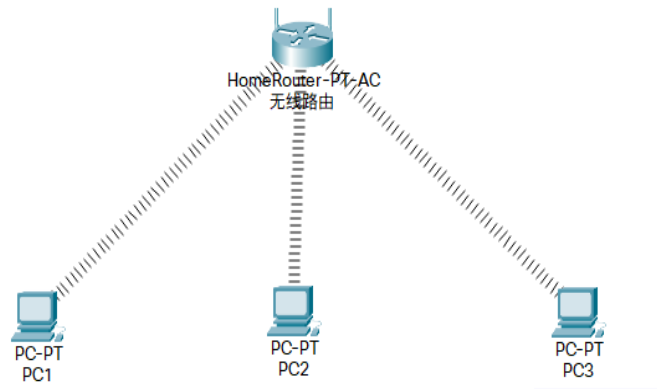


图 2 CiscoPacketTracer 无线连接拓扑图

有线连接拓扑图：

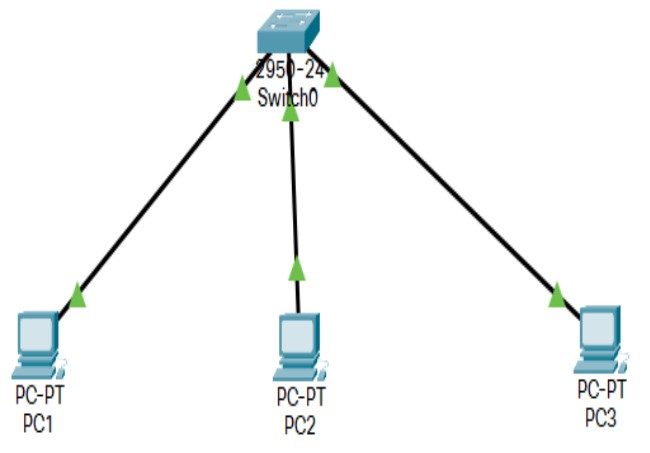


图 3 CiscoPacketTracer 有线连接拓扑图

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：

Simulation Panel				
Event List				
Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC1	ICMP
	0.001	PC1	无线路由	ICMP
	0.003	--	无线路由	ICMP
	0.004	无线路由	PC1	ICMP
	0.004	无线路由	PC2	ICMP
	0.009	--	PC2	ICMP
	0.010	PC2	无线路由	ICMP
	0.012	--	无线路由	ICMP
	0.013	无线路由	PC1	ICMP
	0.013	无线路由	PC2	ICMP

图 4 无线连接条件下的 ICMP 报文交换过程

有线连接：

Simulation Panel				
Event List				
Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC1	ICMP
	0.001	PC1	Switch0	ICMP
	0.002	Switch0	PC2	ICMP
	0.003	PC2	Switch0	ICMP
	0.004	Switch0	PC1	ICMP

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)

3. 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)

4. 采用 ping 命令进行二者之间的通信，测试是否连通

```
C:\Users\HP>ping 192.168.43.2

正在 Ping 192.168.43.2 具有 32 字节的数据:
来自 192.168.43.2 的回复: 字节=32 时间=2ms TTL=128
来自 192.168.43.2 的回复: 字节=32 时间=3ms TTL=128
来自 192.168.43.2 的回复: 字节=32 时间=2ms TTL=128
来自 192.168.43.2 的回复: 字节=32 时间=2ms TTL=128

192.168.43.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 2ms, 最长 = 3ms, 平均 = 2ms
```

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

无线局域网适配器 WLAN:

```
连接特定的 DNS 后缀 . . . . . :  
本地连接 IPv6 地址. . . . . : fe80::788c:74c1:c51a:7fdf%2  
IPv4 地址 . . . . . : 192.168.43.116  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.43.1
```

图 9 查看设备 IP 地址(1)

无线局域网适配器 WLAN:

```
连接特定的 DNS 后缀 . . . . . :  
IPv4 地址 . . . . . : 192.168.43.109  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.43.1
```

图 10 查看设备 IP 地址(2)

```
bwq@ubuntu:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.43.159 netmask 255.255.255.0 broadcast 192.168.43.255  
    inet6 fe80::c04a:694:1853:12ba prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:bf:16:26 txqueuelen 1000 (以太网)  
    RX packets 14622 bytes 19345102 (19.3 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6954 bytes 632347 (632.3 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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"

```
PS C:\Users\刘二蛋\Desktop> c::; cd 'c:\Users\刘二蛋\Desktop'; & 'python' 'c:\Users\刘二蛋\.vscode\extensions\ms-python.f  
b\python\debugpy\launcher' '8167' '--' 'c:\Users\刘二蛋\Desktop\socket.py'  
请输入服务器ip:192.168.43.159  
请输入服务器端口:7788  
input your data:bwq,yyds  
收到数据: hello here is bwq
```

图 12 CLIENT 端通信界面

```
bwq@ubuntu:~/桌面/Python$ python3 tcp_server.py
-----等待新的客户端连接-----
('192.168.43.116', 1293)
-----已经为当前客户服务完毕-----
```

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。

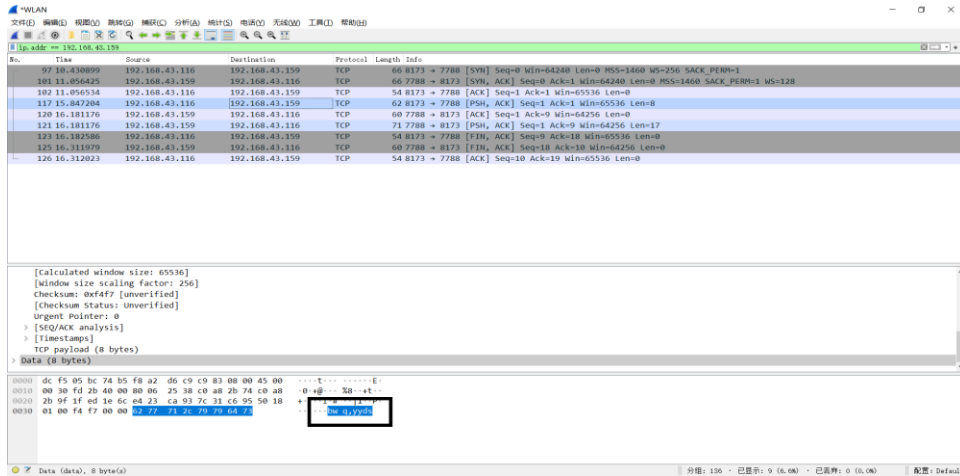


图 14 Wireshark 抓取发送信息数据包

### 5.3.1 物理层分析

```
> Frame 56: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{01302D9B-81C5-43D1-AB6A-953EB092CF32}
  Interface id: 0 (\Device\NPF_{01302D9B-81C5-43D1-AB6A-953EB092CF32})
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 10, 2021 00:09:30.744577000 中国标准时间
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1636474170.744577000 seconds
    [Time delta from previous captured frame: 0.000672000 seconds]
    [Time delta from previous displayed frame: 0.000672000 seconds]
    [Time since reference or first frame: 21.814481000 seconds]
    Frame Number: 56
    Frame Length: 71 bytes (568 bits)
    Capture Length: 71 bytes (568 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:data]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
```

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

```
▼ Ethernet II, Src: AzureWav_bc:74:b5 (dc:f5:05:bc:74:b5), Dst: LiteonTe_c9:c9:83 (f8:a2:d6:c9:c9:83)
  > Destination: LiteonTe_c9:c9:83 (f8:a2:d6:c9:c9:83)
  > Source: AzureWav_bc:74:b5 (dc:f5:05:bc:74:b5)
  Type: IPv4 (0x0800)
```

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(dcf5:05: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

```
▼ Internet Protocol Version 4, Src: 192.168.43.159, Dst: 192.168.43.116
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 57
  Identification: 0x504c (20556)
  > Flags: 0x40, Don't fragment
  Fragment Offset: 0
  Time to Live: 64
  Protocol: TCP (6)
  Header Checksum: 0x120f [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.43.159
  Destination Address: 192.168.43.116
```

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

```
▼ Transmission Control Protocol, Src Port: 7788, Dst Port: 12857, Seq: 1, Ack: 9, Len: 17
  Source Port: 7788
  Destination Port: 12857
  [Stream index: 3]
  [TCP Segment Len: 17]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3580580533
  [Next Sequence Number: 18 (relative sequence number)]
  Acknowledgment Number: 9 (relative ack number)
  Acknowledgment number (raw): 2011511873
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0x22b6 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
  TCP payload (17 bytes)
```

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,

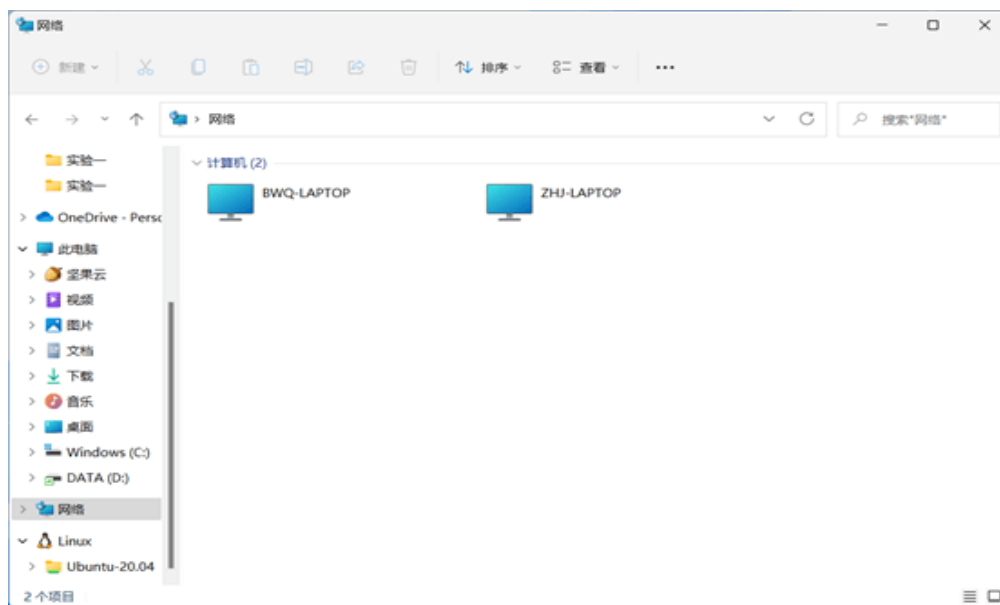


图 22 计算机“网络”界面



- 通过微软账号和密码可以进入 ZHJ-LAPTOP，访问其共享的文件夹

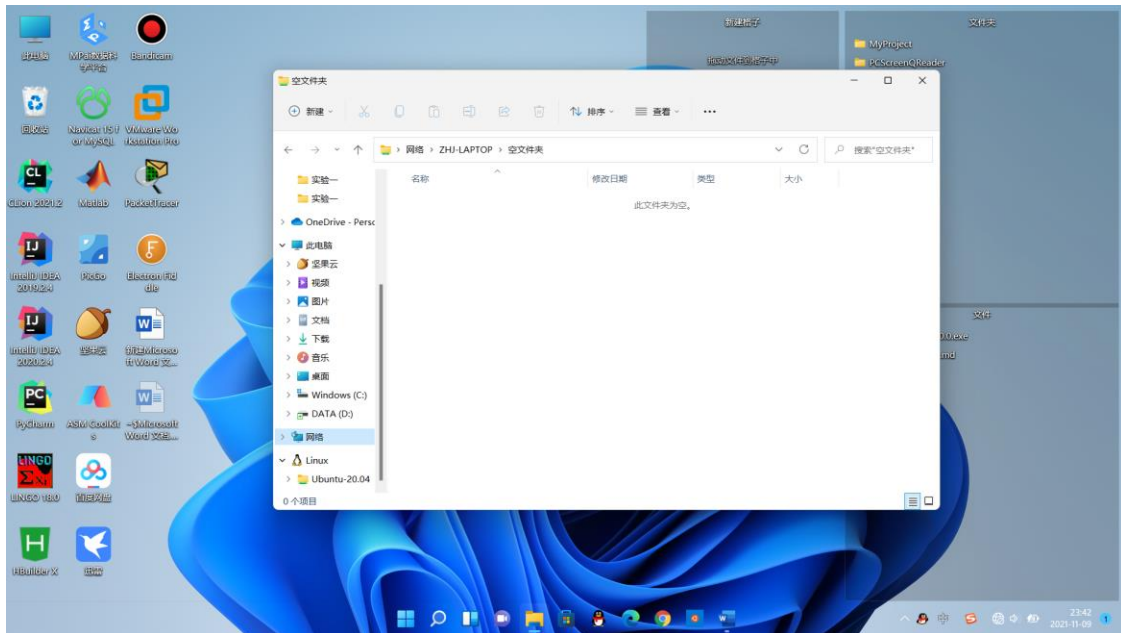


图 23 BWQ-LAPTOP 电脑上查看 ZHJ-LAPTOP 共享文件夹

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。

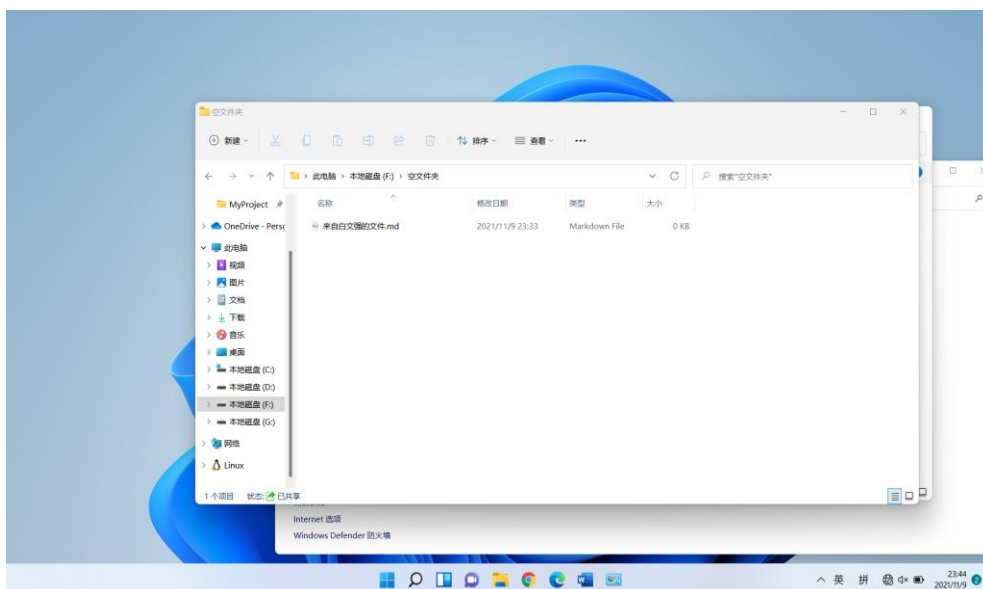


图 24 传输文件后查看 ZHJ-LAPTOP 的共享文件夹

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。

#### 2. 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。

#### 3. 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，并顺利获得 IP 地址。

#### 4. Ping 网络上的设备

一旦获取 DHCP 地址之后，即可使用工具 ping 局域网外的设备，可确认 DHCP 服务器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。

#### 5. 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。

#### 6. 网络流量监控

通过在线工具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

### 实验随想

①

在本次网络工程实验中,我接触到了很多在我之前看来很陌生的软件及实验相关专业名词,如Wireshark, VISIO等。幸运的是,实验完成的时间相对宽裕,这也就意味着我可以有许多时间来学习,熟悉这些软件的使用。此次实验中,我承担的主要任务是使用Wireshark对各层次涉及到的协议进行抓包分析及利用Cisco对实验设计图进行具象实现,但除了自己承担的任务,我也在其他部分进行了尝试,如制作双绞线等动手操作。经过本次实验,我对于网络工程课程有了相较之前更全面、更清晰的认识。

在我看来,我在第一次实验中最大的收获就是对协议分析的更深入理解及利用Cisco时制作实验拓扑图并使其正常运行时对该软件的更娴熟的使用。如在验证无线路由是否正常工作是利用Cisco的简单报文工具对报文发送的每一步进行认真观察,使我对路由工作的原理有了更深刻的印象;在协议分析时面对繁多且复杂的各类参数,筛选出有价值的部分参数。以上种种,都需要我对自己知识体系进行扩充,才能很好地完成自己负责的实验工作。

张朝

20191060046

19级计科

2021.11.10.



# 云南大学

所谓工程,即要真刀真枪动手实践。在本次对等网络实验中,我们用自己的笔记本电脑及手机,搭建了一个简单的局域网并成功让处在局域网中的笔记本电脑实现了文件共享、消息传输、图片传输等。

我们学习了双绞线的制作,剥皮→排序→理直→剪齐→插入→压线,即使步骤写的很清楚,理论准备充分,但实际操作跟理论还是不一样的,缺乏实际操作经验的我们在第一次制作的时候还是失败了。吸取第一次的经验教训后,第二次制作的时候,我们成功了,并实现了将两台电脑相连,并进行文件共享。

我们还自学了 socket 编程,虽然我们从未接触过,但经过上网学习,我们还是利用 python 进行编程,成功实现了一个简单的聊天室功能。

我认为实验课最重要的就是提升自己的动手能力,在实践过程中,我们会去学习理论知识在实践中的应用,会为了实验去补习平时没学过的知识,在实践中提升自我。

白文强 20191060064





### 实验一心得

实验一的内容是基于Windows平台的对等网络的搭建,在这次的实验中,我们小组成员分别通过有线和无线两种形式对P2P网络进行了搭建,并实现了在自组网络中的资源共享和信息的传输。

首先我们是在Visio上进行了一个对等网络的拓扑图的设计和规划,由于我们的设计网络经验的不足,当我们完成初稿交付给老师时,老师针对我们的拓扑图指出了很多的问题,包括文字标注的标准问题和有线连接接口的严谨性问题。通过拓扑图的设计,我们就可以学习到,对待一件事情,我们必须要注意涉及到的所有细枝末节,任何一点小问题最后都有可能会酿成大错。在设计完拓扑图后,我们开始使用Cisco Packet Tracer软件进行仿真验证,确保我们的设计是可行的。如果缺乏这个步骤,当后期工程开始实际搭建的时候再发现问题会浪费大量的物力、人力,因此虚拟仿真验证是搭建网络工程中非常关键的一个环节。接下来就到了有线和无线网络搭建的过程,我们小组成员亲自动手制作了双绞线,在制作的过程中更体会到了动手能力是多么重要,在制作的过程中,因为制作双绞线的要求是十分精细的,我们每完成一根双绞线的制作都会损坏好多根,由此可见动手能力是我们急切需要提高的。





# 云南大学

在无线网络搭建的过程中，我们学习了套接字编程和 Wireshark 对不同层的协议进行分析。

对于实验一的内容本身来说，实验本身难度不大，但它教会了我们要真正完成一个网络工程需要哪些步骤，教会了我们认真对待一件事情应如何考虑细节，完成一个整体性的任务应考虑上下系统的结构性和一致性。要想成为一名合格的工程师，需不断追逐细节的完美和整体的完整性，这样才能使得我们的结果越来越完美。

2021. 11. 14





## 实验小结

“纸上得来终觉浅，绝知此事要躬行。”本次的实验描述起来并不复杂，只需要建立一个端对端的小型网络，理论部分老师之前讲解过，但在具体的实现过程中却遇到了相当多的麻烦，顿觉实践与理论之间的差距。

早在上学期的计算机网络这门课中，网络交叉线的规格就学过，568A、568B的标准早已了然于心。但在真正拿起水晶头、拿起网线时，才发现自己存在的实践与理论之间的偏差。包括在PT上模拟过更加复杂的网络拓扑结构，但到了具体实践连线传输文件时，却又出现了诸多状况。

这次看似简单的实验，让我更深一步了解了网络搭建的全流程，以小见大地了解了一番工程中专业知识的使用方式，理论与实践相结合，真正对网络连接，对网线制作有了自己的认识。

2019/10/60061

刘明浩.





## 实验一心得体会

通过本次和小组成员共同完成的对等网实验，我收获颇丰。学会了基于 windows 平台在一个局域网内搭建对等网并进行资源共享的方法。通过对等网的搭建，每一个终端既能作为服务端，又能做为客户端。实验过程中，首先进行软件仿真，以确定初步设想，并进一步完善网络设计，在硬件上通过自己制作双绞线完成了有线网络通信，使计算机之间实现了连通，最后使用 python 进行套接字编程实现了传输层信息传输。通过这次实验，我对网络不同层间的信息传输有了深入了解，并对对等网的建立、硬件的配置、编程命令等有了进一步的学习。同时，在团队合作、动手能力、创新能力等方面也得到了很好的锻炼。

贾润堃

20191060062



赵尚杰 20191060074

本次做了“基于 Windows 平台的对等网络连接”实验；这次由于我们没有交换机设备，所以有线连接只做了两台电脑之间的网络连接，并且使用无线网进行多台电脑的网域建立。

在进行有线网的 P2P 实验之前，使用老师带来的材料进行了网线的制作，不由得感觉到什么内容都不容易啊，第一次制作网线在夹水晶头时因为松了一下而失败了，在第二次制作才成功，学习到了 568A 和 568B 标准，在同型设备之间需要两头的标准不一样，异型设备之间两头用同样的标准。

在成功建成网域以后，通过套接字编程实现了消息、图片、文件的共享，在本次实现实验目标的过程中学习的知识不仅促进了实验的完成，也对理论知识进行了巩固，获益匪浅。

## 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()

```