

计网课程设计实验报告

题目： 计算机网络课程设计

**组长姓名 张坤**

**组员姓名**  **缪健**

**提交日期** 2021年1月14日

目录

[一、 任务一 3](#_Toc14591)

[二、 任务二 7](#_Toc11099)

[1. 在交换机上实现VLAN配置 7](#_Toc15726)

[2. 基于 Console控制台登录配置路由器，学习路由器配置相关命令 10](#_Toc10218)

[A. 基本配置 10](#_Toc12748)

[B. 接口配置 11](#_Toc30169)

[3. 基于 packet tracer构建网络环境，分别进行静态路由配置和基于RP的动态路由配置。 11](#_Toc2947)

[1) 静态路由配置 11](#_Toc20548)

[2) 动态路由配置 14](#_Toc12089)

[三、 任务三 18](#_Toc3932)

[1. 运行环境 18](#_Toc10712)

[2. 设计目标 18](#_Toc21478)

[3. 核心思想 18](#_Toc10919)

[4. 流程图 18](#_Toc9043)

[5. 关键问题,核心代码 19](#_Toc3957)

[6. 运行界面 20](#_Toc8656)

[7. 抓包分析 21](#_Toc6890)

[1) 第一次抓包 22](#_Toc16848)

[2) 第二次: 22](#_Toc28269)

[3) 第三次 22](#_Toc15575)

[至此三次握手完毕 22](#_Toc27724)

[4) 第四次 22](#_Toc1924)

[5) 第五次 23](#_Toc3075)

[下面是四次挥手 23](#_Toc17215)

[7) 第一次挥手: 23](#_Toc14804)

[8) 第二次 23](#_Toc491)

[9) 第三次 23](#_Toc25617)

[10) 第四次 23](#_Toc5259)

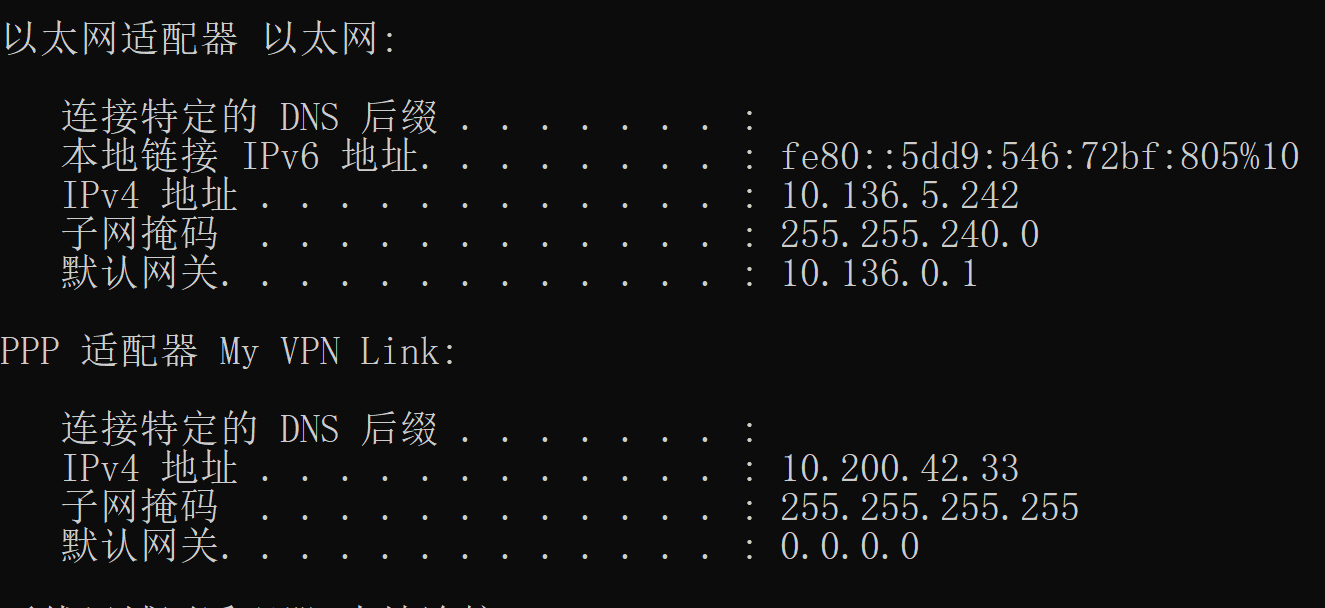
[8. 实验遇到的问题: 23](#_Toc11208)

[1 服务器只能连接一个客户端: 23](#_Toc9156)

[2 服务器无法返回数据给客户端 23](#_Toc7083)

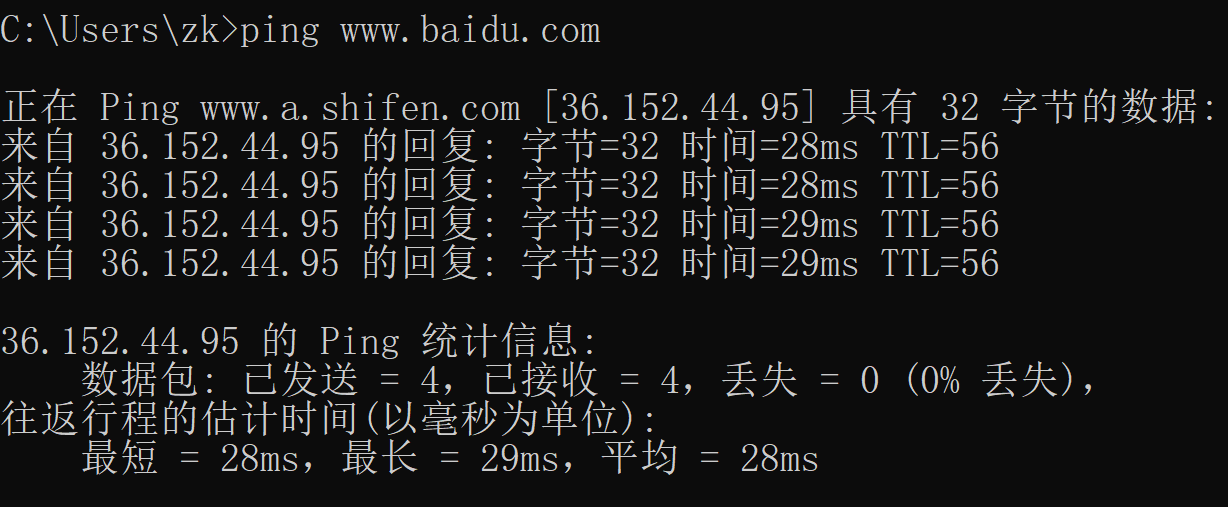
[9. 实验总结 23](#_Toc2452)

1. **任务一**
2. 常用网络命令 ipconfig,ping, netstat. tracert,arp, telnet的功能
3. ipconfig:用来显示主机内IP协议的信息,包括以太网,以太网配置器,PPP适配器等信息



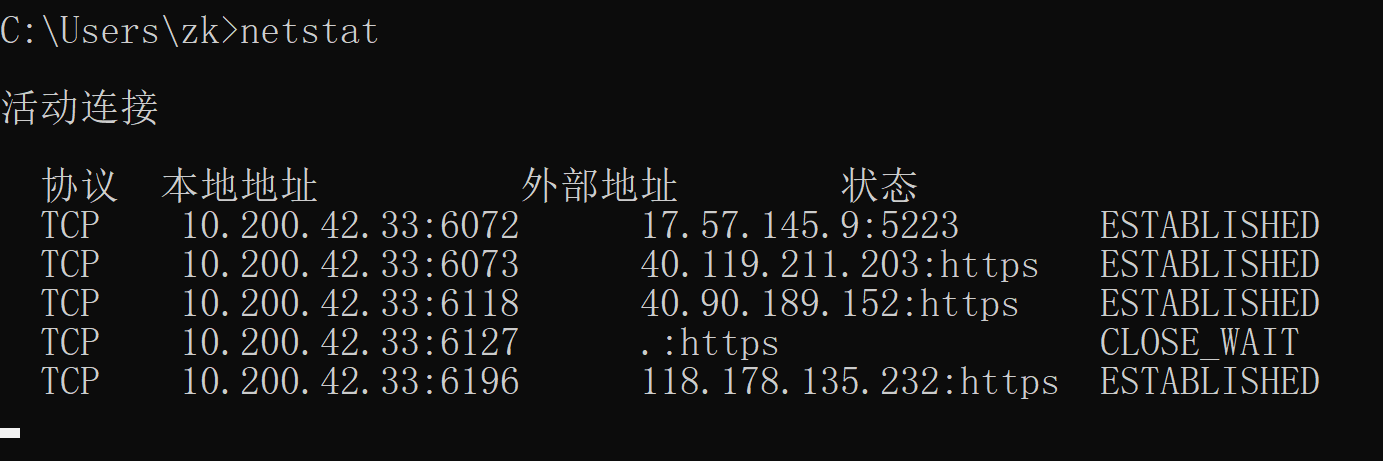
如上,可以从中查看本机的IP地址,子网掩码和默认网关等等

1. ping:用来检测两台主机相互通讯是否成功,需要多少时间.以ping [www.baidu.com](http://www.baidu.com)为例



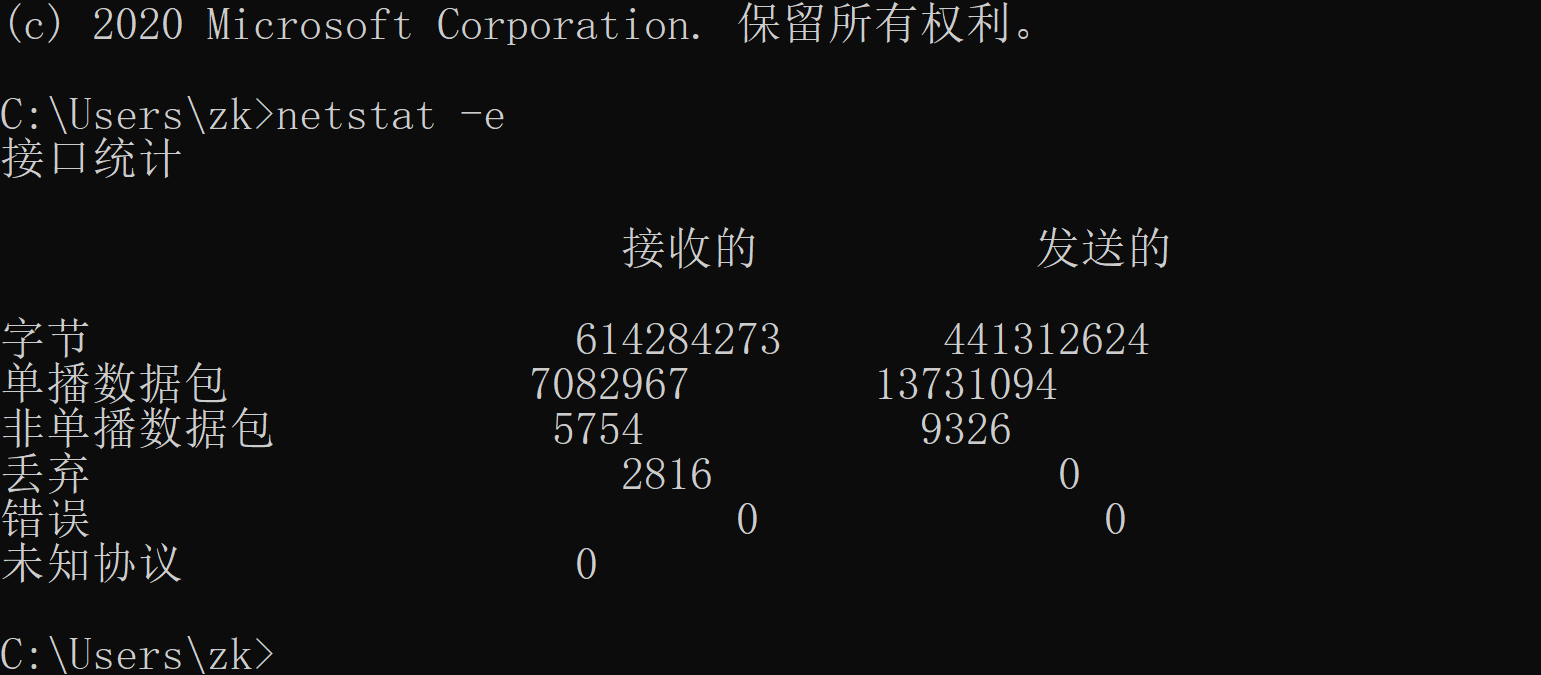
显然可以ping通[www.baidu.com](http://www.baidu.com),一共发送四个数据包,用时约28ms

1. netstat:帮助我们了解网络的整体使用状况,可以显示当前正在活动的网络连接详细信息

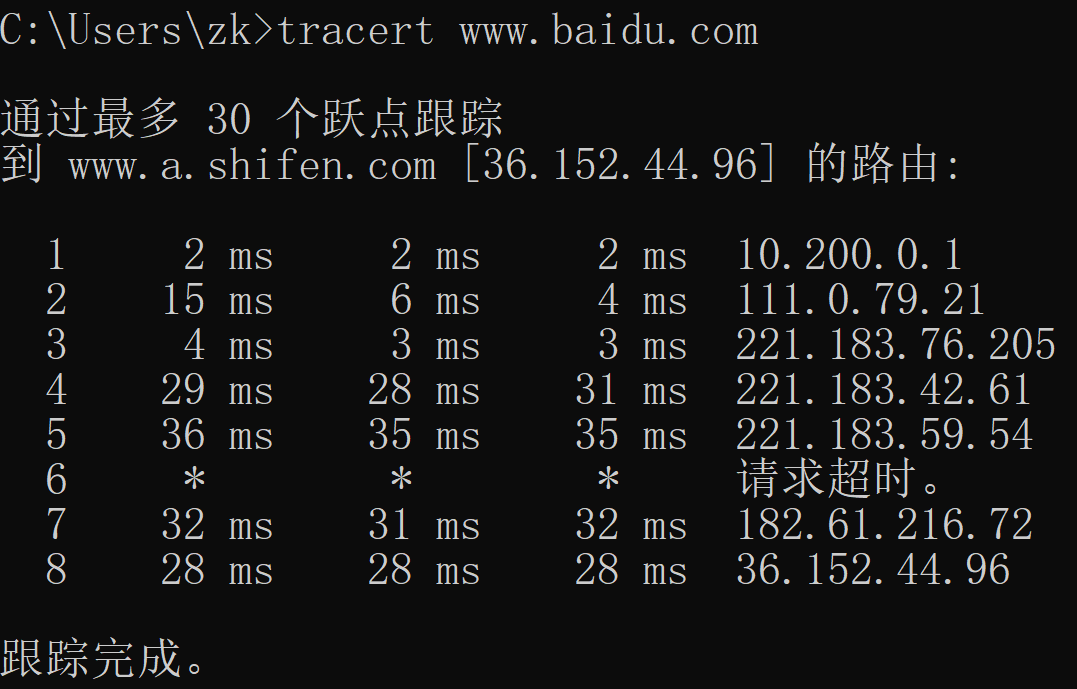


以部分信息为例,其显示结果包含:协议名称,本地地址,外部地址,状态.

还有netstat-e,netstat -n等命令.如下



1. tracert: 判定数据包到达目的主机所经过的路径、显示数据包经过的中继节点清单和到达时间.以tracert [www.baidu.com](http://www.baidu.com)为例



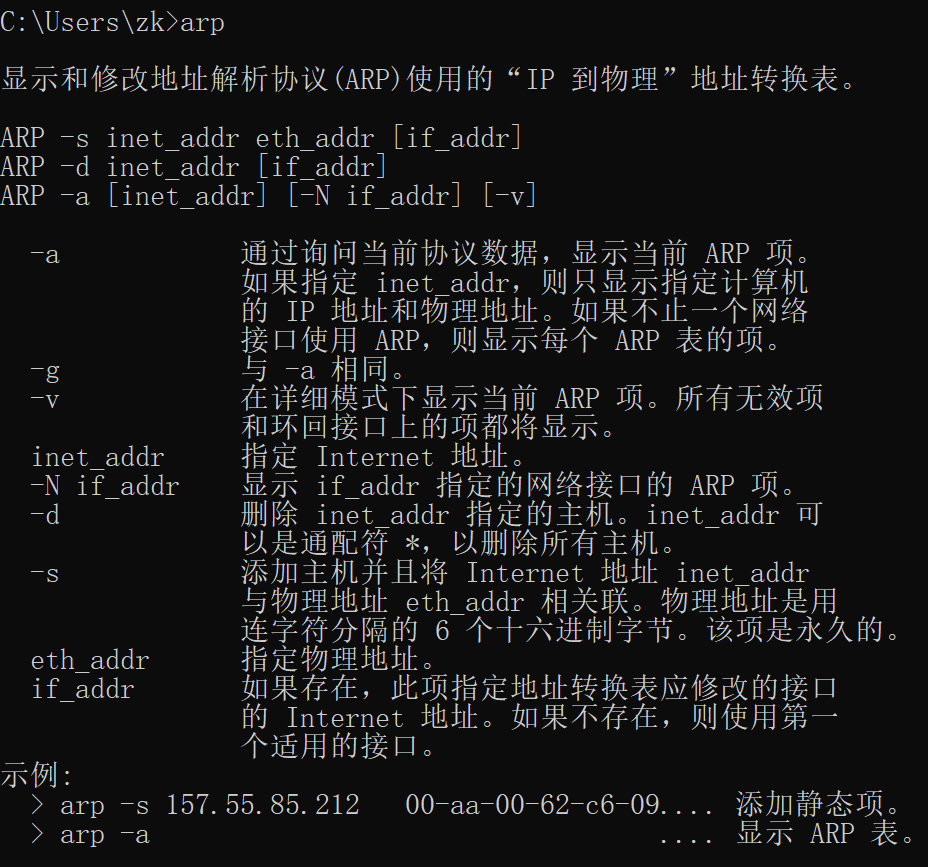
其连接过程如上,一共跟踪八个跃点

Tracert同样也有后续命令

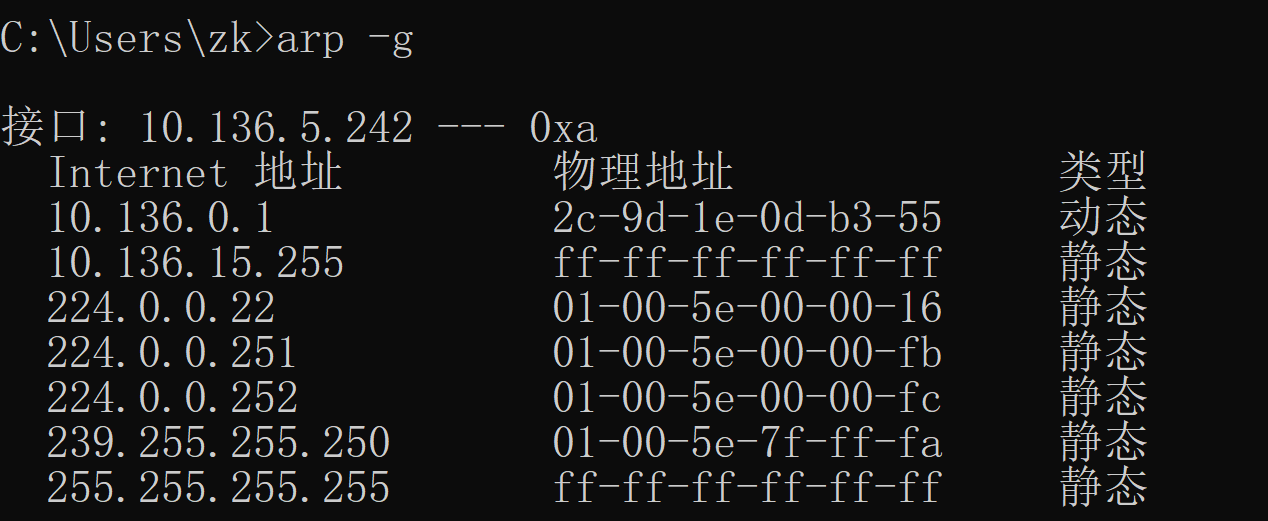
1. -d:指定不将IP地址解析到主机名称
2. -h maximum\_hops:指定跃点数以跟踪target\_name的主机的路由
3. -j host-list:指定tracert使用程序数据报所采用的路径中的路由器接口列表
4. -w timeout:等待timeout为每次回复所指定的毫秒数

1. arp: 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.

仅输入arp命令会提示如下



那么以arp -g为例



后文太长不放了,现在可以看出目前本机的arp高速缓存如上

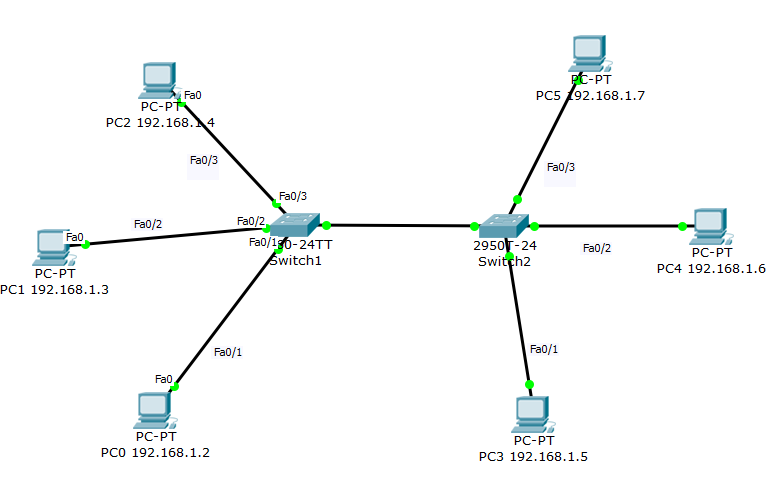
同样arp也有后续命令:

* + 1. -a:通过询问TCP/IP显示当前arp项。如果制定了inet\_addr，则只显示指定计算机的IP地址和物理地址。
    2. -g: 通过询问TCP/IP显示当前arp项。如果制定了inet\_addr，则只显示指定计算机的IP地址和物理地址。
    3. inet\_addr:以带点的十进制标记指定Internet地址
    4. -d：删除由inet\_addr指定的项
    5. -s:在arp缓存中添加项，将Internet地址inet\_addr和物理地址eth\_addr相关联。
    6. 物理地址由以连字符分隔的6个十六进制字节给定。使用带点的十进制标记指定internet地址，在超市到期后项自动从缓存删除。

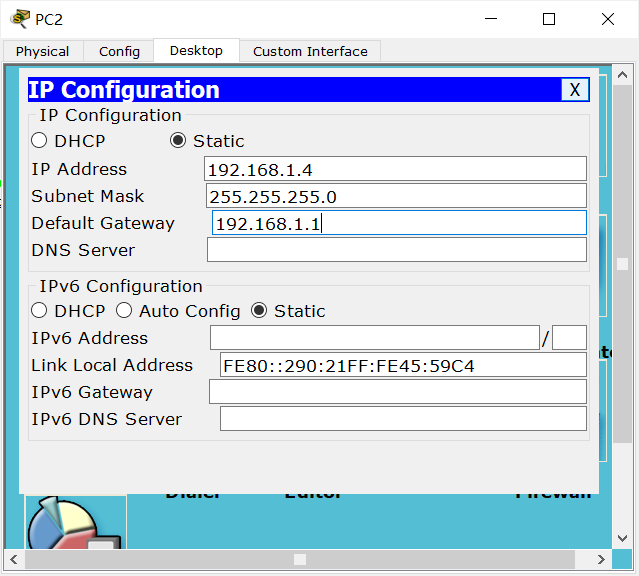
1. telnet:远程控制服务器,允许用户登录进入远程主机系统,远程操作等等.其常用命令行如下:
2. open : 使用 openhostname 可以建立到主机的 Telnet 连接。
3. close : 使用命令 close 命令可以关闭现有的 Telnet 连接。
4. display : 使用 display 命令可以查看 Telnet 客户端的当前设置。
5. send : 使用 send 命令可以向 Telnet 服务器发送命令。支持以下命令：
6. ao : 放弃输出命令。
7. ayt : “Are you there”命令。
8. esc : 发送当前的转义字符。
9. ip : 中断进程命令。
10. synch : 执行 Telnet 同步操作。
11. brk : 发送信号。

1. **任务二**
2. 在交换机上实现VLAN配置

拓扑结构如下

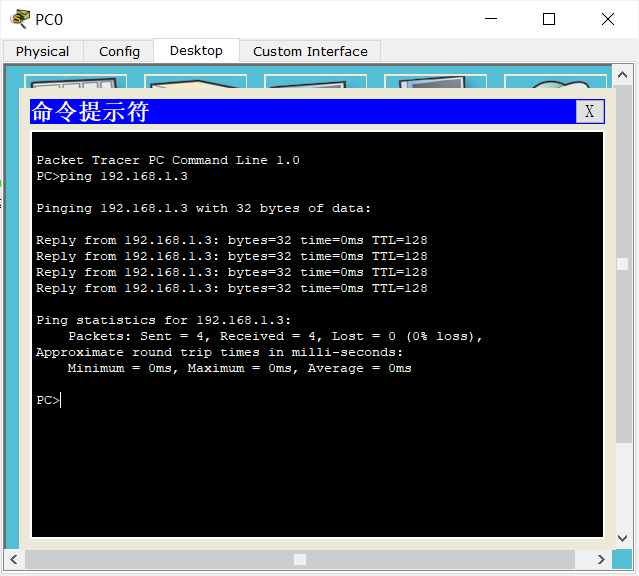


PC的IP配置如下



其余如图不赘述

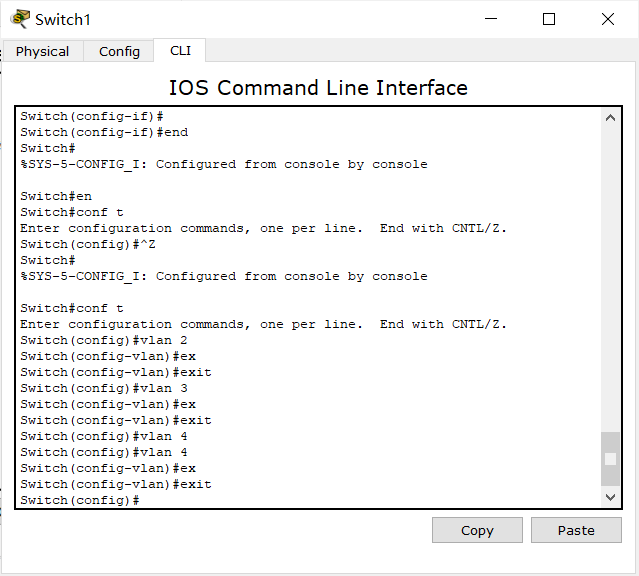
先尝试ping命令如下



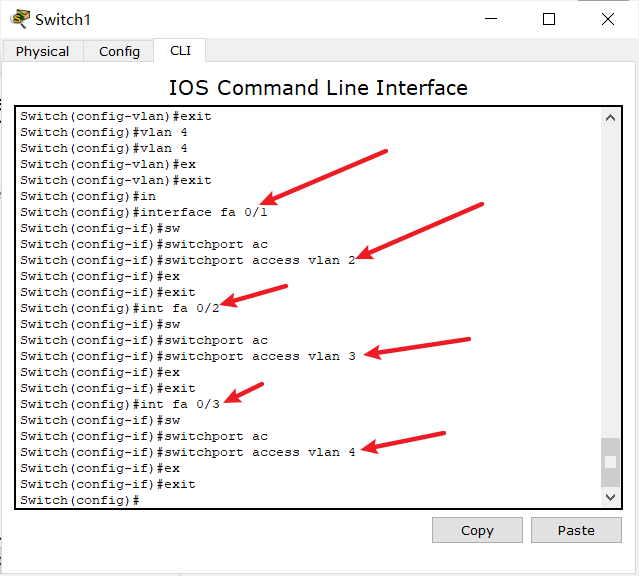
4个PC均可通讯

对交换机使用命令行如下

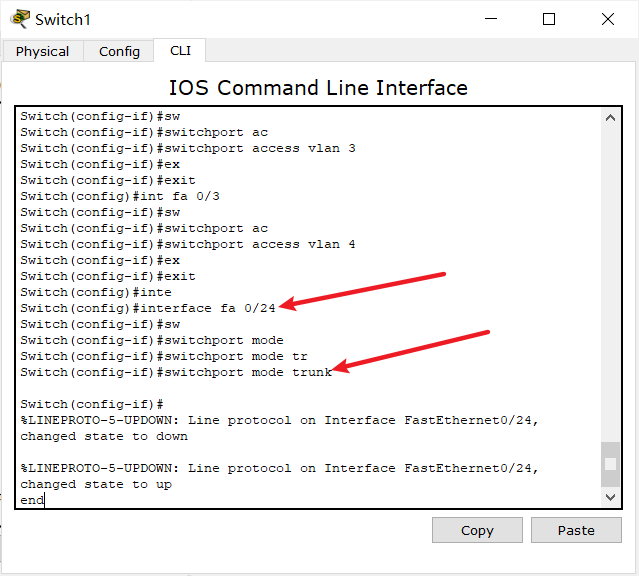
先新建三个VLAN



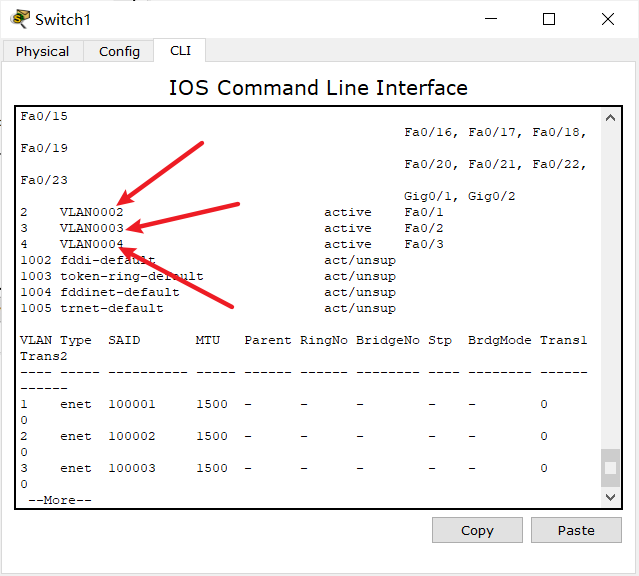
把vlan配置给线路如下:



最后配置两交换机之间的线路0/24



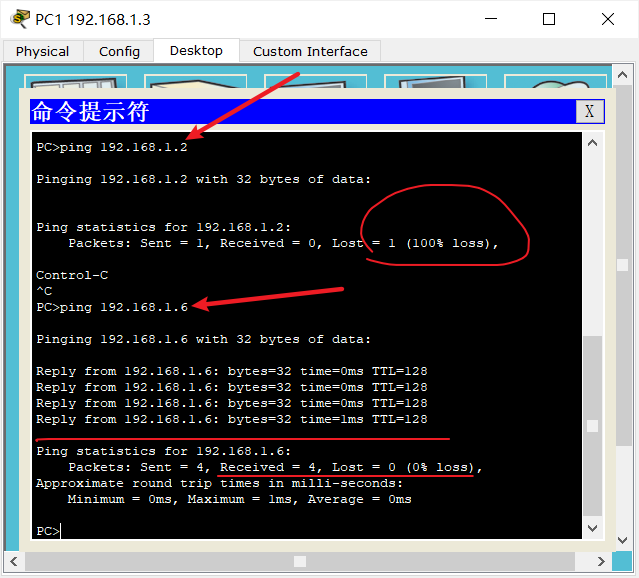
Show VLAN 如下



对另一台交换机使用同样命令行,不再赘述

那么测试ping命令查看六台PC是否可以通讯

192.168.1.3 ping 192.168.1.2如下

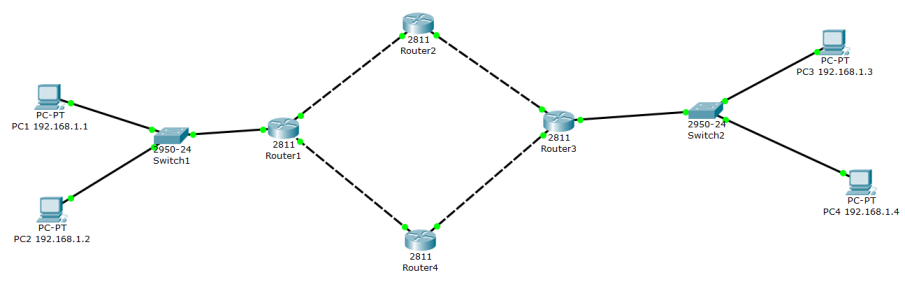


不可ping通,因为主机处于不同的VLAN中

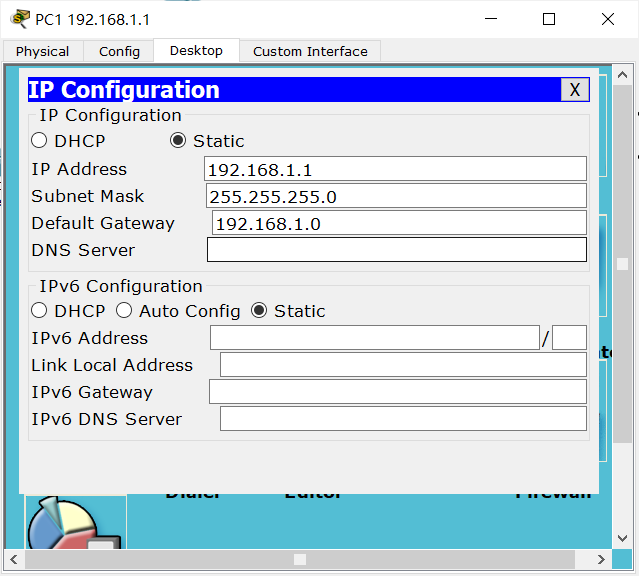
而192.168.1.3 ping 192.168.1.6则可以,因为处于相同VLAN中

1. 基于 Console控制台登录配置路由器，学习路由器配置相关命令
2. 基本配置
3. router>enable /进入特权模式
4. router#conf t /进入全局配置模式
5. router(config)# hostname xxx /设置设备名称
6. router(config)#enable password  /设置特权口令
7. router(config)#no ip domain lookup /不允许路由器缺省使用DNS解析命令
8. router(config)# Service password-encrypt /对所有在路由器上输入的口令进行暗文加密
9. router(config)#line vty 0 4 /进入设置telnet服务模式
10. router(config-line)#password xxx /设置telnet的密码
11. router(config-line)#login /使能可以登陆
12. router(config)#line con 0 /进入控制口的服务模式
13. router(config-line)#password xxx /要设置console的密码
14. router(config-line)#login /使能可以登陆
15. 接口配置
16. router(config)#int s0 /进入接口配置模式 serial 0 端口配置（如果是模块化的路由器前面加上槽位编号，例如serial0/0 代表这个路由器的0槽位上的第一个接口）
17. router(config-if)#ip add xxx.xxx.xxx.xxx xxx.xxx.xxx.xxx  /添加ip 地址和掩码
18. router(config-if)#enca hdlc/ppp 捆绑链路协议 hdlc 或者 ppp 思科缺省串口封装的链路层协议是HDLC所以在show run配置的时候接口上的配置没有，如果要封装为别的链路层协议例如PPP/FR/X25就是看到接口下的enca ppp或者enca fr
19. router(config)#int loopback /建立环回口(逻辑接口)模拟不同的本机网段
20. router(config-if)#ip add xxx.xxx.xxx.xxx xxx.xxx.xxx.xxx  /添加ip 地址和掩码给环回口在物理接口上配置了ip地址后用no shut启用这个物理接口反之可以用shutdown管理性的关闭接口
21. 基于 packet tracer构建网络环境，分别进行静态路由配置和基于RP的动态路由配置。要求：静态路由配置拓扑中至少4个路由器；RP动态路由配置中源站和目的站之间设置两条跳数不同的路径，通过RIP配置后查看选择的是哪条路径要求写出相应的步骤，给出截图和文字说明。
22. 静态路由配置

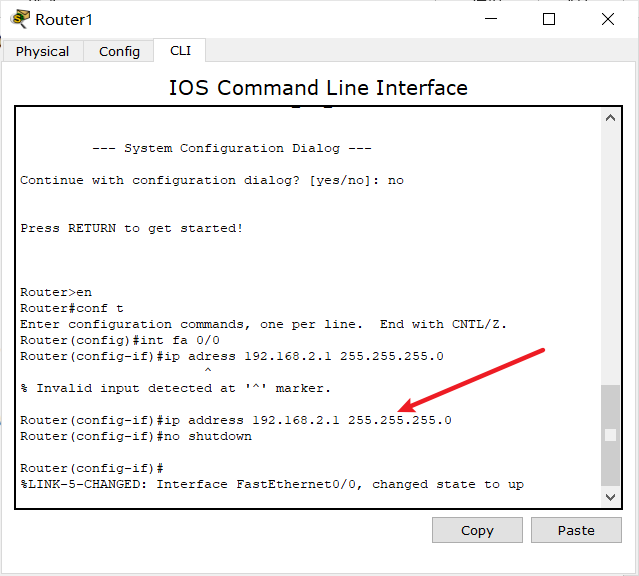
拓扑结构如下



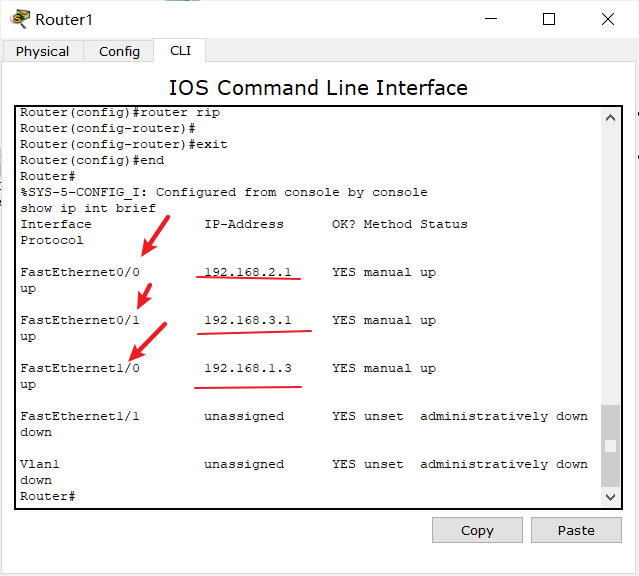
首先设置PC的IP地址如下,以PC1为例:



接着设置四个路由器的接口如下,以Router1为例



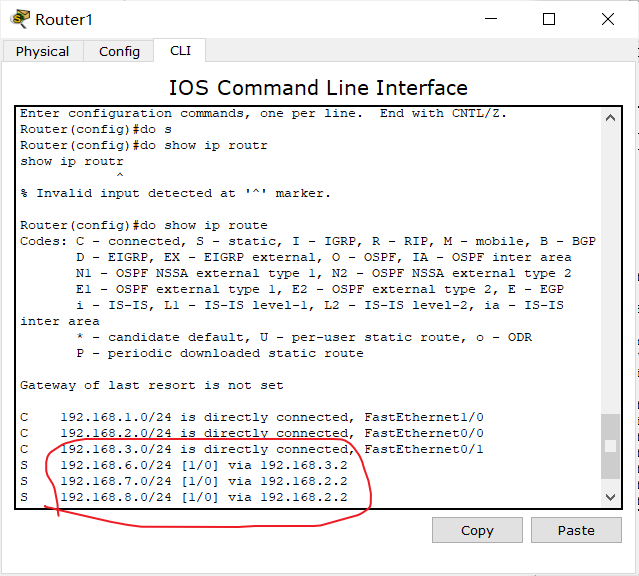
需要对四个路由器设置,用命令show ip int brief 查看如下



接着设置静态路由



用命令行do show ip route查看



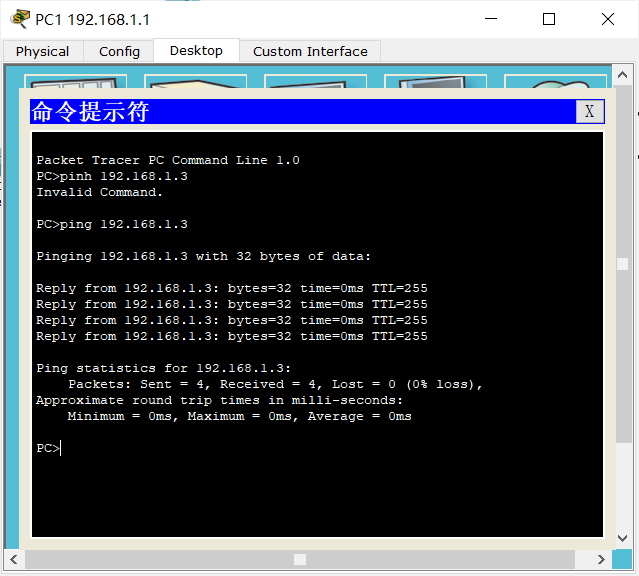
对其余三台路由表同样设置

接着尝试ping命令

一开始是无法ping通的,但是第二次就可以了

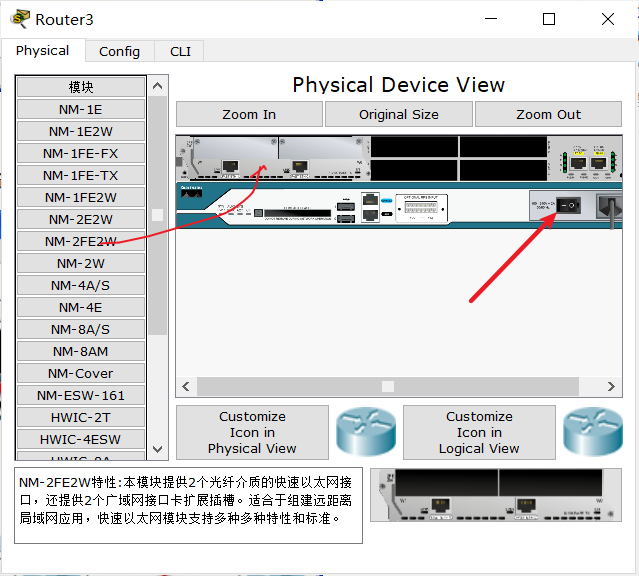
此处忘记截图了

PC1 ping PC3

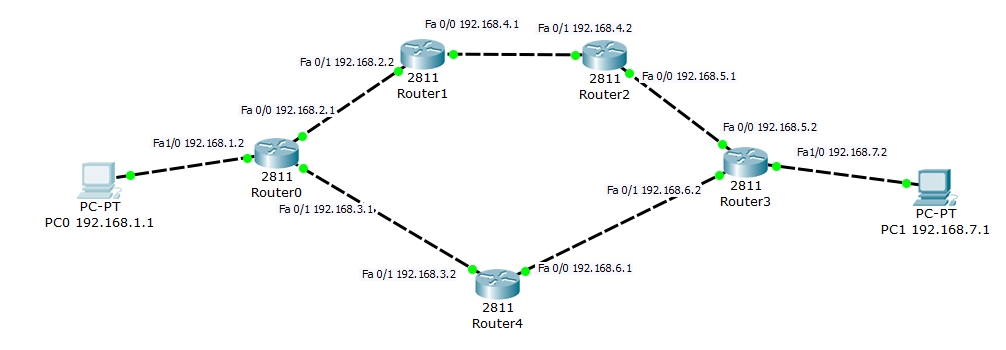


1. 动态路由配置

首先增加部分路由器端口,先关闭电源,再把NM-2FE2W拖拽到如图所示位置

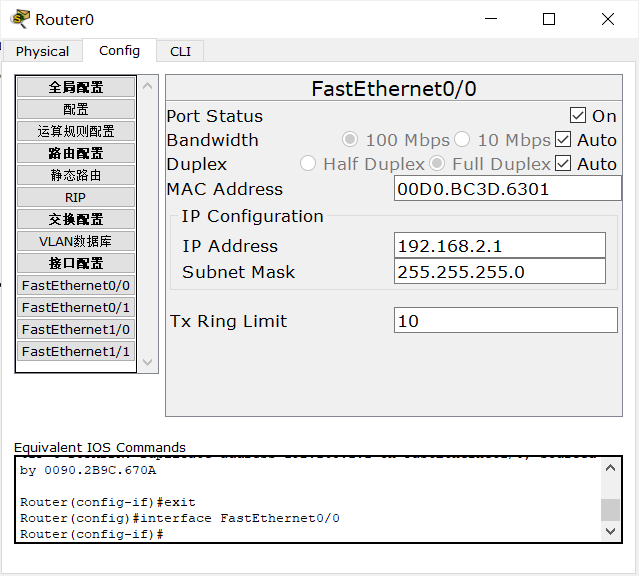


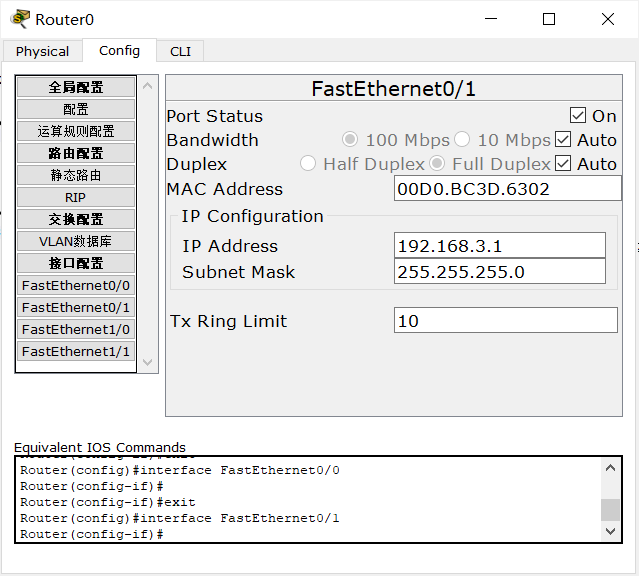
拓扑结构如图:

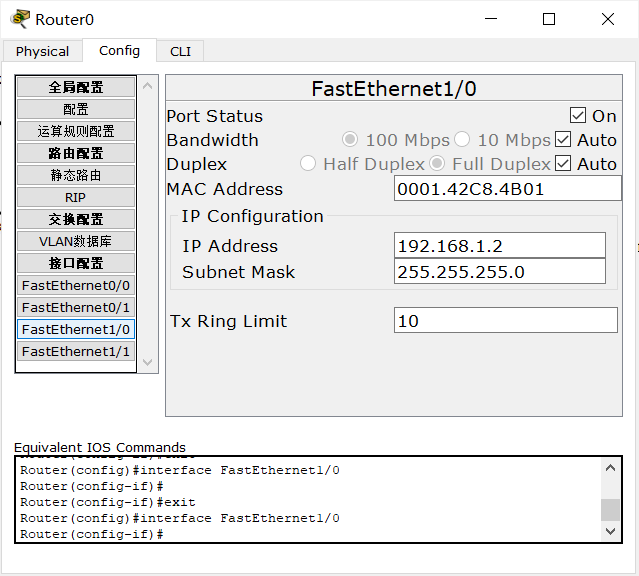


除了命令行外可以在config界面直接设置

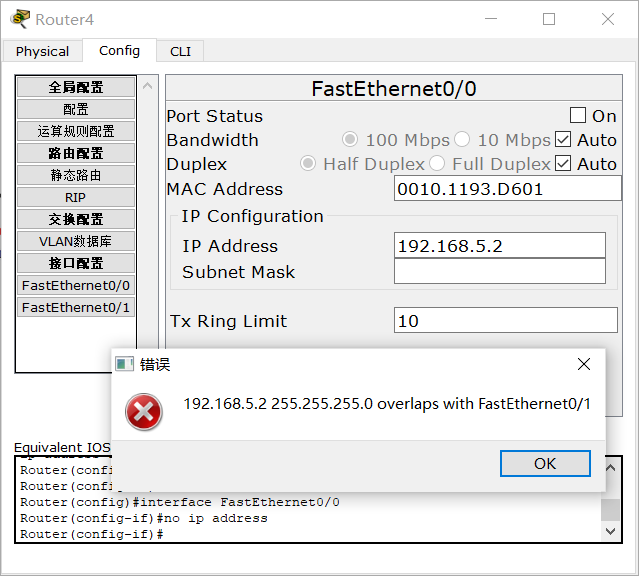
以Router0为例:



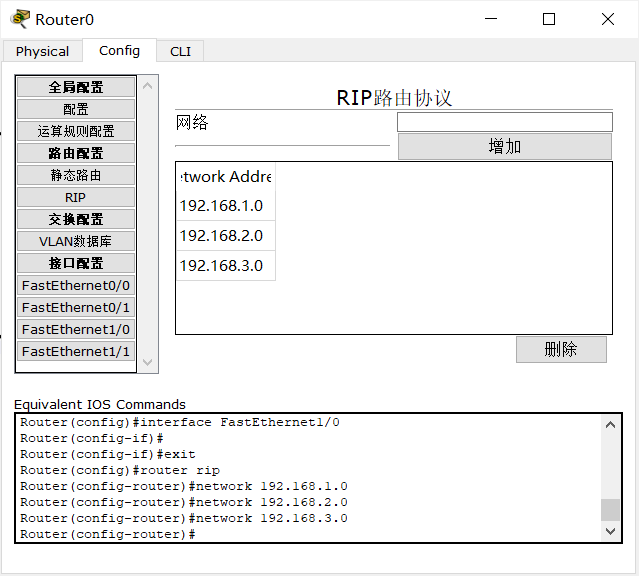




可能会出现这样的错误,只要令IP地址的第三个数字与其他端口不相同即可

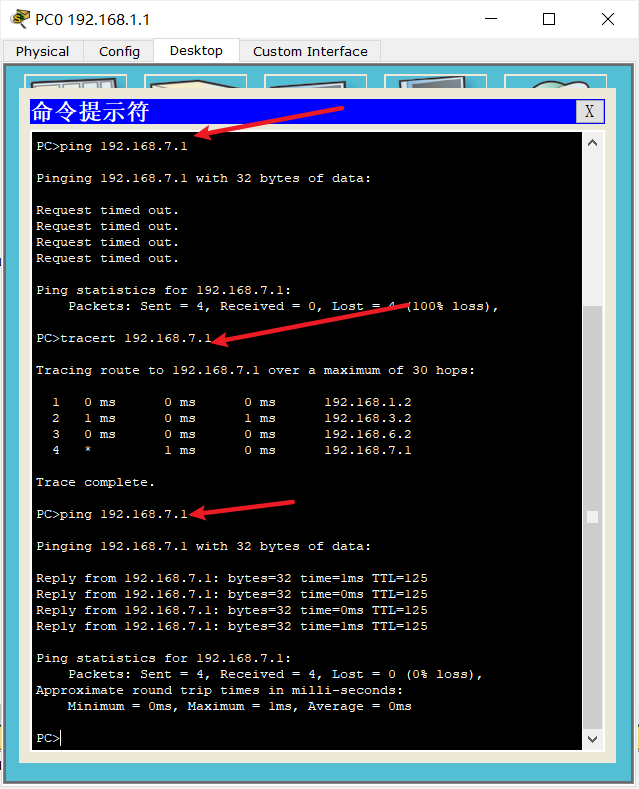


接着在config界面给路由器配置RIP路由,以Router0为例:



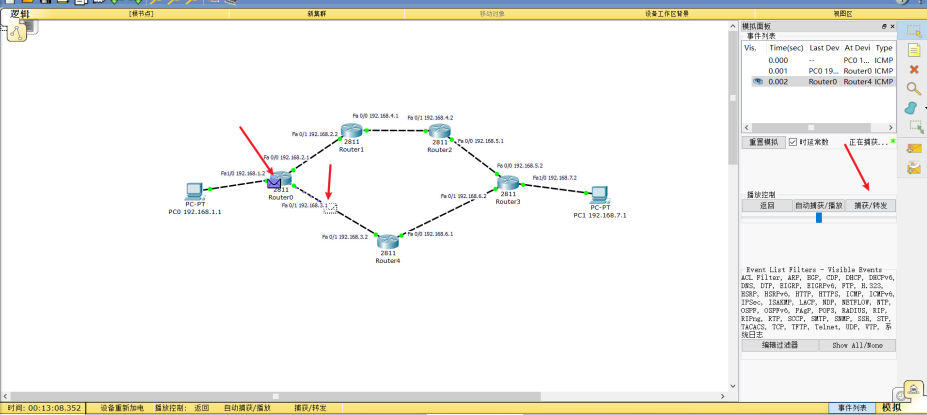
其余不赘述

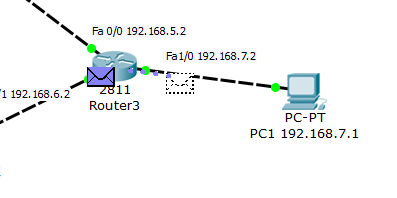
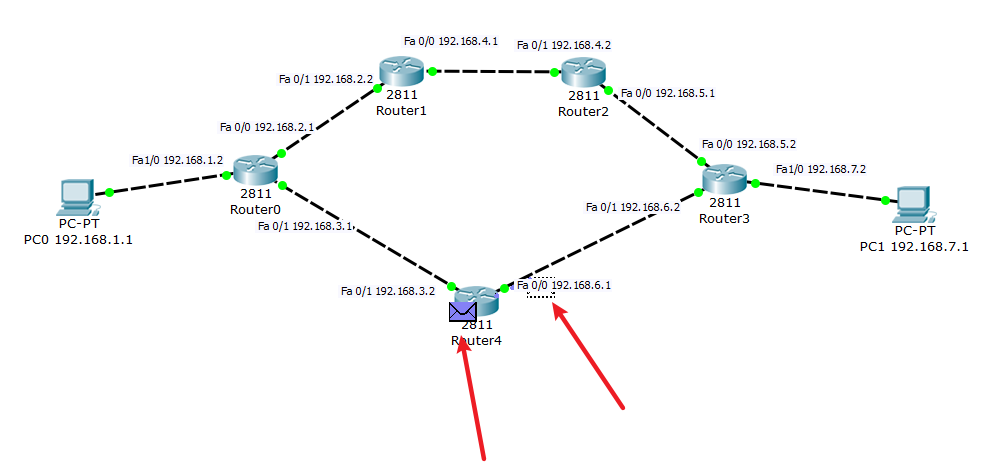
那么全部设置成功后.第一次ping会失败,因为此时会建立线路如图



那么输入tracert 192.168.7.1 命令后可以看到路径是沿着下面的路发送ip数据包的

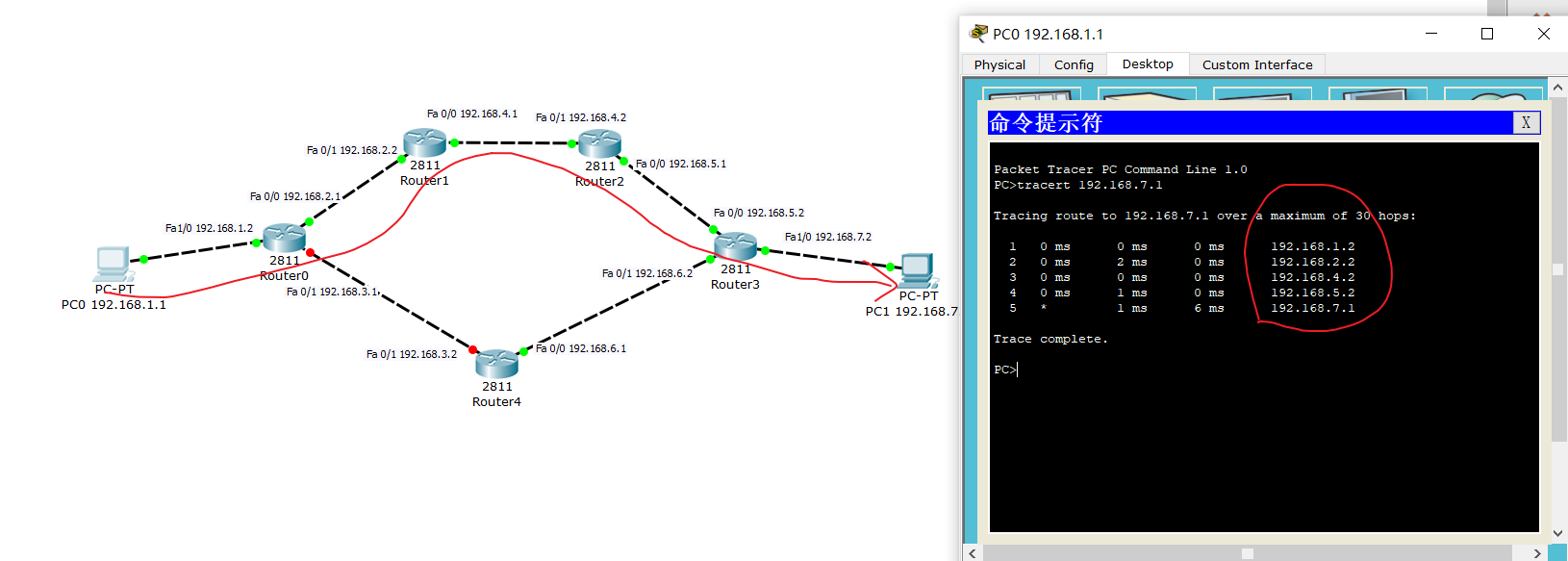
在思科界面中也可以看到





即IP数据包发送路径如上

那么我们尝试断开下面的路段,继续查看数据包发送路径如下



1. **任务三**

编程要求：两位同学一组，一位同学做TCP客户端，一位同学做TCP服务器。要求实现客户端和服务器端的文本通信，在客户端输入文本后，能够将该文本发送至服务器端正确显示。同样，在服务器端输入文本后，能够将该文本发送至客户端正确显示。完成程序后，要求对所编写程序产生的网络数据进行抓包分析。要求必须输出以下字段：TCP：源端口、目的端口、顺序号、确认号、标志位，IP：版本号、总长度、标志位、片偏移、协议、源地址和目的地址。对抓包结果进行解析，对所编写的程序进行理解，并解释关键语句的作用，以及其产生的数据包，并完成详细的说明文档。文档中内容包括所使用的实验软件和操作系统、程序的设计思想、流程图、关键问题和关键语句、程序注释和对捕获包的解析截图/与程序语句的关联、总结和心得体会等

1. **运行环境**

操作系统:WIN10

编译环境:IDEA 2020.3

处理器Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.60 GHz

1. **设计目标**

基于TCP的Socket类和Java多线程实现类似QQ的多人聊天工具

1. **核心思想**

主要利用ServerSocket和Socekt类连接程序,并依靠其DataOutputStream的writeUTF

和readUTF函数, BufferedWriter的write和flush函数传输数据

1. **流程图**

服务器

客户端上传数据 服务器回传数据

客户端

用户输入

1. **关键问题,核心代码**
2. 服务器怎么连接客户端:

经过网上多个博客的学习,服务器与客户端的连接主要依靠ServerSocket类和Socekt类,如下



首先用serverSocket用于等待客户端的连接,其accept函数会阻塞这个过程直到客户端连接.

1. 服务器怎么连接多个客户端:

主要依靠两个:while循环和Java多线程

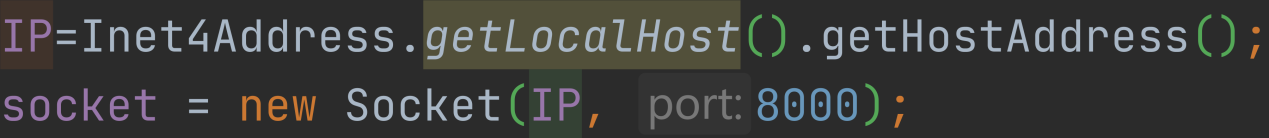
其中while循环是为了不停地接受客户端连接请求,而多线程是为了开启多个线程,接受客户端数据



不过这会有一个弊端,就是服务器会卡在这里死循环,其UI界面会卡住.

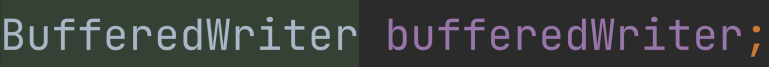
1. 客户端怎么连接服务器:

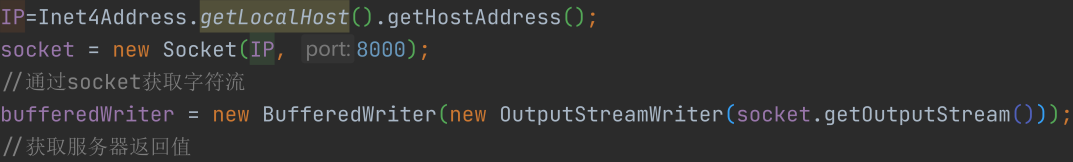
客户端连接服务器需要端口号,IP地址,代码如下:

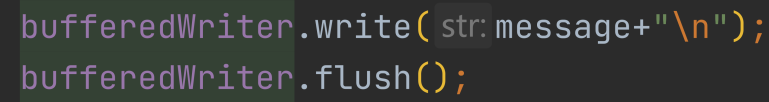


1. 客户端怎么上传数据给服务器
2. 当客户端连接上服务器后就可以传输数据,代码如下

首先依赖BufferWriter类接受Socket的输出流,并将数据输入到Socket中,再flush出去

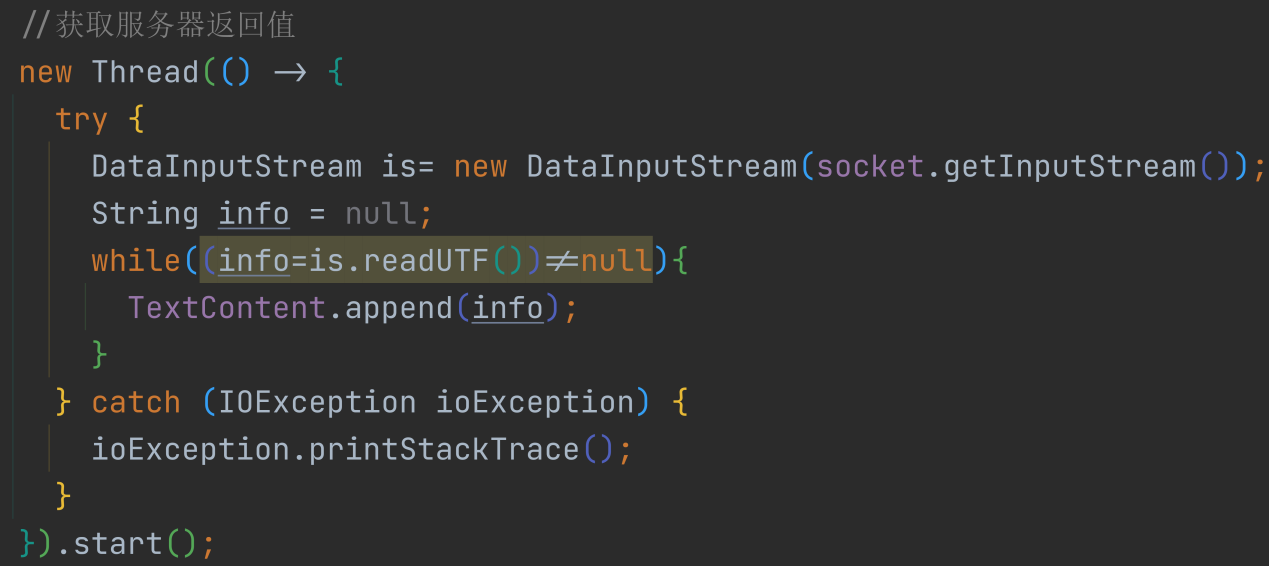




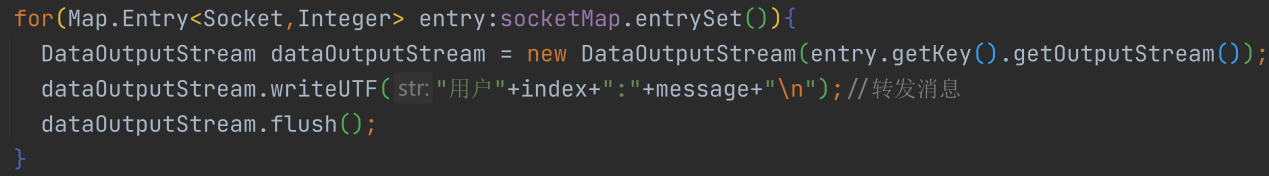


1. 服务器怎么回传数据给客户端

首先客户端需要在登录的时候开启一个线程,用于接收服务器传回来的数据:

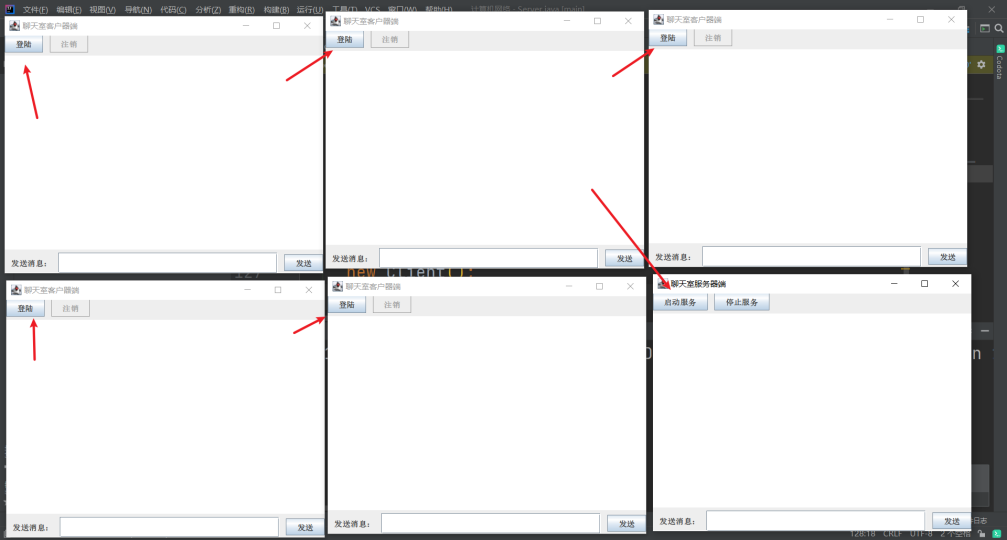


1. 其次服务器需要一个存放客户端Socket的数据,这里我使用了Map类,既存放了Socket数据,也存放了他们对应的客户端编号,用循环遍历和DataOutputStream类把数据输出到Socket中去

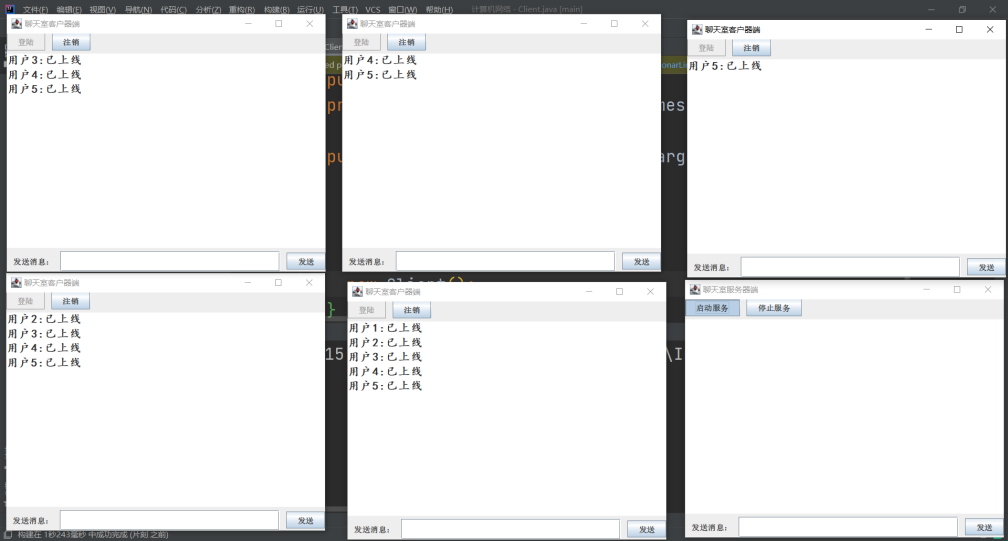


1. 阿巴巴巴
2. **运行界面**

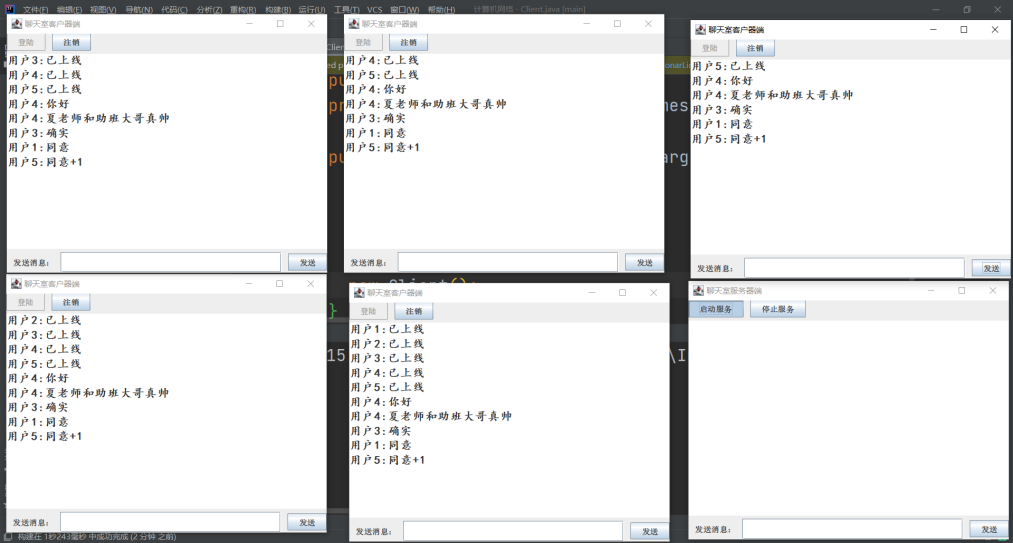
首先需要服务器运行服务,然后客户端登录



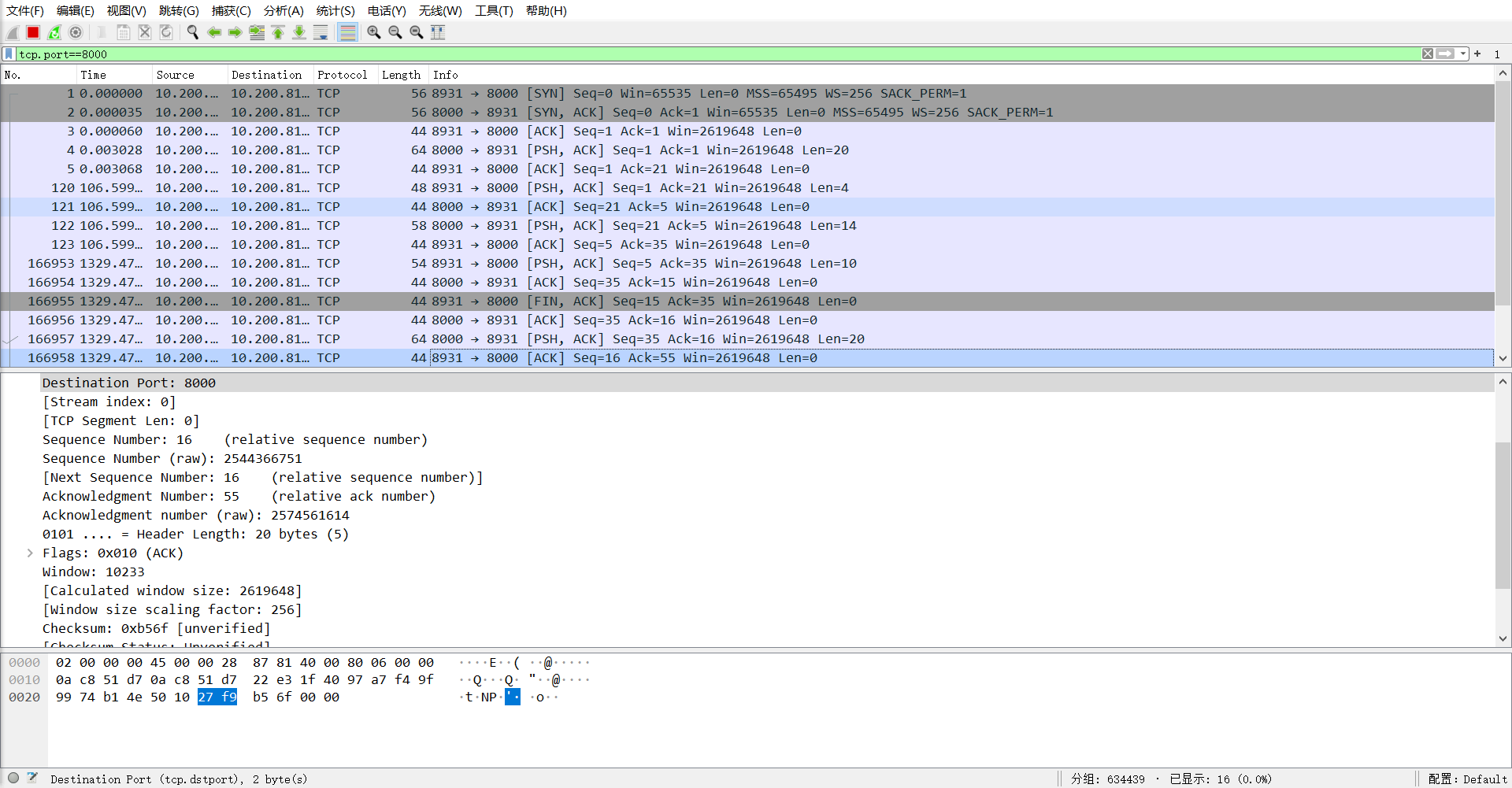
用户登录



然后用户发送数据:



1. **抓包分析**



首先确认的是前三个包1-3:三次握手.

其次是服务器返回给所有客户端的一个数据和客户端返回的数据包4-5(只登陆了一个客户端),

接着是客户端发向服务器的一个聊天记录(内容111),然后服务器返回一个确认包120-121,

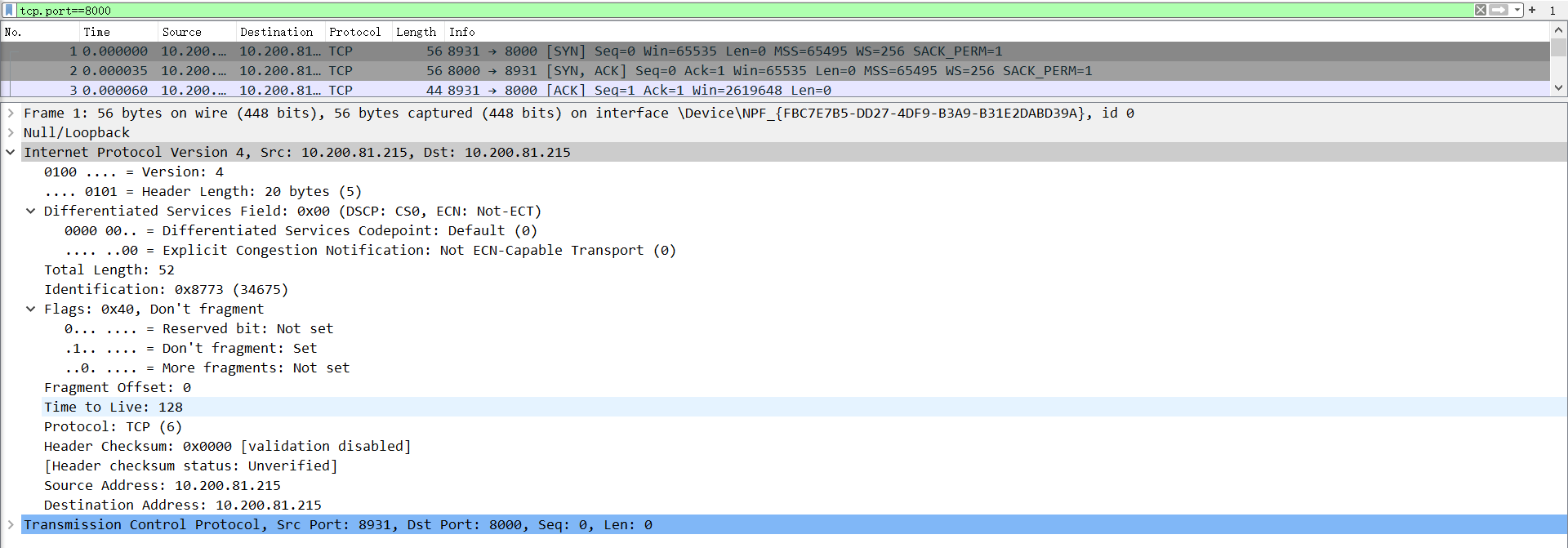
然后是服务器返回给所有客户端的一个数据和客户端返回的数据包122-123(只登陆了一个客户端),

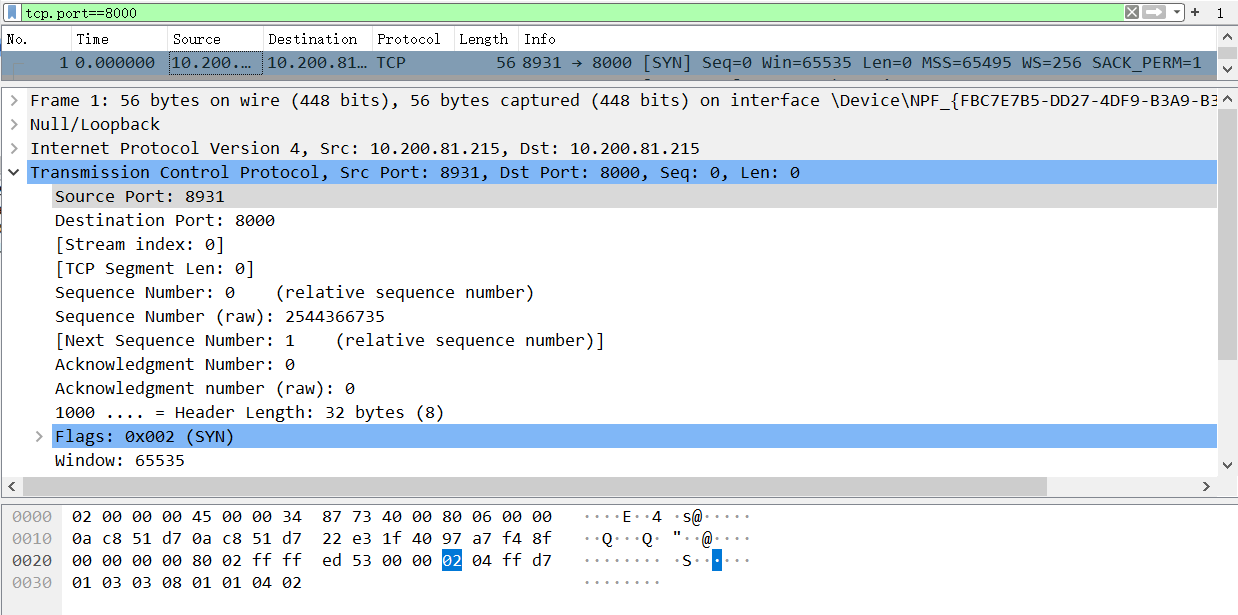
然后是服务器返回给所有客户端的一个数据(用于客户端提示下线)和客户端返回的数据包166954-166954(只登陆了一个客户端),

最后客户端断开连接166955-166958,对应四次挥手

1. 第一次抓包

截图如下,其余抓包截图不再放了





TCP：源端口:8931;目的端口8000;序列号: Seq=0;确认号:0;标志位:SYN=1,其余无;

IP：版本号(Version4);总长度(52);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第二次:

TCP：源端口:8000;目的端口8931;序列号: Seq=0;确认号:1;标志位:SYN=1,ACK=1其余无;

IP：版本号(Version4);总长度(52);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第三次

TCP：源端口:8000;目的端口8931;序列号: Seq=1;确认号:1;标志位: ACK=1其余无;

IP：版本号(Version4);总长度(40);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

**至此三次握手完毕**

那么第四次和第五次的报文分别为1:客户端连接服务器后,服务器给客户端的一个响应,用于提示用户上线

2:客户端接收4后的回应.

1. 第四次

TCP：源端口:8000;目的端口8931;序列号: Seq=1;确认号:21;标志位:PSH=1ACK=1其余无;

IP：版本号(Version4);总长度(60);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第五次

TCP：源端口: 8931;目的端口8000;序列号: Seq=1;确认号:21;标志位:ACK=1其余无;

IP：版本号(Version4);总长度(40);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 对应序号为4-5,120-121, 166954-166954的数据包和第四次第五次的是几乎一样的,这里就不赘述了,

**下面是四次挥手**

1. 第一次挥手:

TCP：源端口: 8931;目的端口8000;序列号: Seq=1;确认号:35;标志位:FIN=1,ACK=1其余无;

IP：版本号(Version4);总长度(40);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第二次

TCP：源端口: 8000;目的端口8931;序列号: Seq=1;确认号:16;标志位: ACK=1其余无;

IP：版本号(Version4);总长度(40);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第三次

TCP：源端口: 8000;目的端口8931;序列号: Seq=1;确认号:16;标志位:PSH=1,ACK=1其余无;

IP：版本号(Version4);总长度(60);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

1. 第四次

TCP：源端口: 8931;目的端口8000;序列号: Seq=1;确认号:55;标志位: ACK=1其余无;

IP：版本号(Version4);总长度(40);标志位DF=1片偏移0协议TCP6源地址(10.200.81.255),目的地址(10.200.81.255)

至此四次挥手分析完毕

1. **实验遇到的问题:**
2. 服务器只能连接一个客户端:

解决方案:利用多线程,为每一个连接到服务器的客户端分配一个线程用于接收信息,详细过程见上文

1. 服务器无法返回数据给客户端

原来以为需要客户端再建一个Socket用来连接服务器,但其实我上网查询资料后得知,服务器的serverSocket所创建的Socket在连接上客户端后,其实不但可以接收客户端的数据,还可以传输数据给客户端,只需要使用DataOutputStream把数据传回去即可,详细见上文;

1. **实验总结**

通过这次实验,我通过上网查询到了许多原本不会的知识和技能,我学会了如何在思科软件上模拟路由器,交换机,PC的实现,还学会了Java的一个重要的类Socket的使用,通过这个类的学习,我对Socket编程的工作原理有了更深的理解,还有cmd的命令,包括ipconfig,ping, netstat. tracert,arp, telnet等等,还有利用Wireshark软件对数据包进行抓取,并借这个软件的强大功能进行了TCP的三次握手,发送的数据包和四次挥手的分析,加深了我对TCP和IP的理解.以上种种,不但需要复习课本内容,理解其概念意义,还需要自行上网查询资料,查看不同的人的思路,即使前人已经把路铺好了,但是我还是走过许多的坑,被好多bug绊倒,但是依然完成了这次课设.感谢这次课设,加深了我对计算机网络的理解,.