**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 计算机通信与网络

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

[Lab1 使用网络协议分析仪Wireshark 2](#_Toc700319996)

[1.1 环境 2](#_Toc442692613)

[1.2 实验目的 2](#_Toc1020887873)

[1.3 实验内容及步骤 2](#_Toc968489308)

[1.4 实验结果 2](#_Toc1382148275)

[1.5 实验中的问题及心得 7](#_Toc20293989)

[参考文献 7](#_Toc896903662)

[Lab2 网络模拟器Packet Tracer的使用 9](#_Toc708041631)

[2.1 环境 9](#_Toc267023353)

[2.2 实验目的 9](#_Toc937164987)

[2.3 实验内容及步骤 9](#_Toc595230852)

[2.4 实验结果 16](#_Toc1388910581)

[2.5 实验中的问题及心得 18](#_Toc1531172931)

[参考文献 18](#_Toc1272138287)

[Lab3 分析Ethernet II帧、集线器和交换机工作原理 20](#_Toc351887973)

[3.1 环境 20](#_Toc1572791002)

[3.2 实验目的 20](#_Toc1272549102)

[3.3 实验内容及步骤 20](#_Toc873536252)

[3.4 实验结果 21](#_Toc1676866470)

[3.5 实验中的问题及心得 29](#_Toc1411674895)

[参考文献 29](#_Toc154603459)

[Lab4 分析IP和ARP协议 31](#_Toc1180095941)

[4.1 环境 31](#_Toc2079991972)

[4.2 实验目的 31](#_Toc1426106199)

[4.3 实验内容及步骤 31](#_Toc302970798)

[4.4 实验结果 32](#_Toc46970353)

[4.5 实验中的问题及心得 32](#_Toc2065385334)

[参考文献 33](#_Toc739858735)

[Lab5 配置路由器的路由选择协议 35](#_Toc72588001)

[5.1 环境 35](#_Toc456412412)

[5.2 实验目的 35](#_Toc18004491)

[5.3 实验内容及步骤 35](#_Toc772907997)

[5.4 实验结果 35](#_Toc899105025)

[5.5 实验中的问题及心得 41](#_Toc1038892365)

[参考文献 41](#_Toc1741397305)

[Lab6 分析TCP特性 42](#_Toc133769652)

[6.1 环境 42](#_Toc1059186354)

[6.2 实验目的 42](#_Toc490817319)

[6.3 实验内容及步骤 42](#_Toc841811283)

[6.4 实验结果 42](#_Toc1326209708)

[6.5 实验中的问题及心得 42](#_Toc1427982306)

[参考文献 43](#_Toc1437042135)

[Lab7 利用Java(C++)开发网络应用程序 44](#_Toc567636641)

[7.1 环境 44](#_Toc811671589)

[7.2 实验目的 44](#_Toc561696775)

[7.3 实验内容及步骤 45](#_Toc919524614)

[7.4 实验结果 46](#_Toc236978944)

[7.5 实验中的问题及心得 47](#_Toc1834245877)

[参考文献 47](#_Toc1793060867)

**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 使用网络协议分析仪Wireshark

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

# Lab1 使用网络协议分析仪Wireshark

## 1.1 环境

操作系统： ArchLinux x64

网络平台： Wireshark 2.6.3

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 1.2 实验目的

1. 能够正确安装配置网络协议分析软件Wireshark。
2. 熟悉使用Wireshark分析网络协议的基本方法。
3. 加深对协议格式、 协议层次和协议交换过程的理解。

## 1.3 实验内容及步骤

1. 启动WireShark
2. 因为WireShark需要网卡权限，输入命令 sudo wireshark
3. 选择当前网卡elo1
4. 尝试不同的协议和报文进行抓包分析

## 1.4 实验结果

### 1.4.1 分析ping的报文

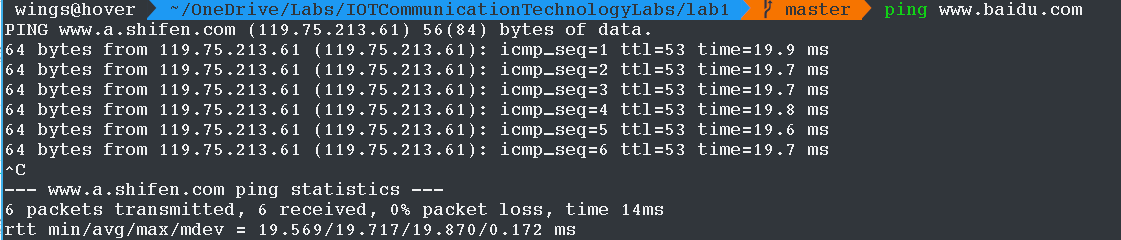


图1-1 ping [www.baidu.com](http://www.baidu.com)结果

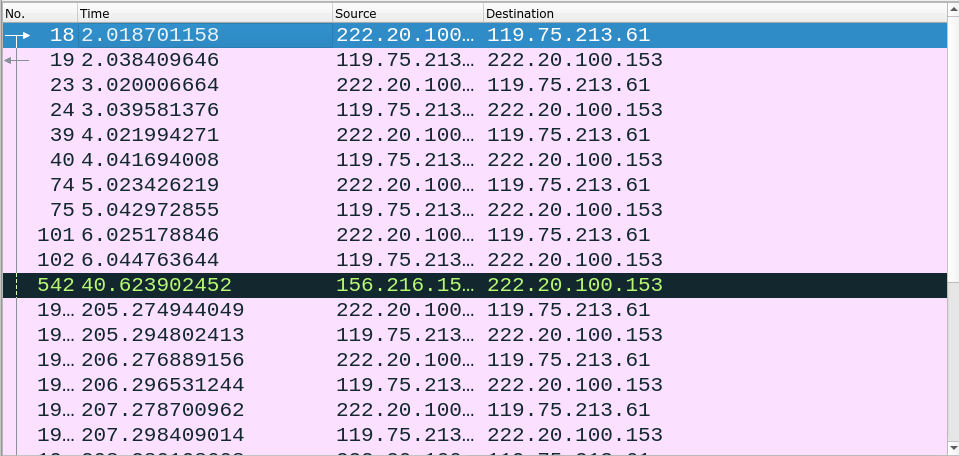


图1-2 wireshark结果



图1-3 ICMP报文结构

协议类型：V4



图1-4 协议类型

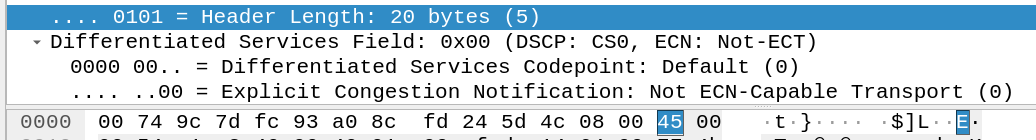


图1-5 首部长度

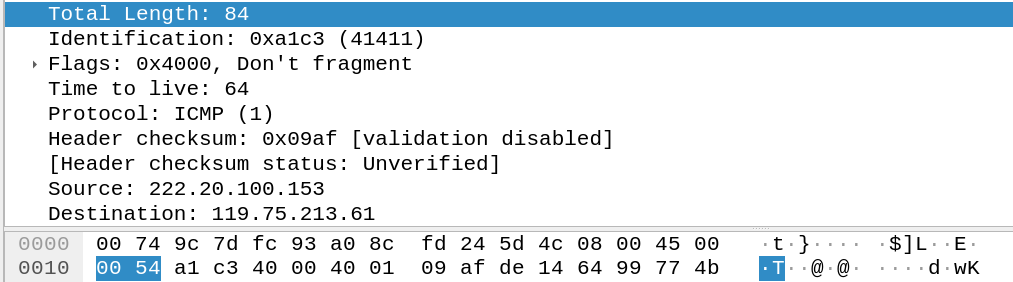


图1-6 总长度

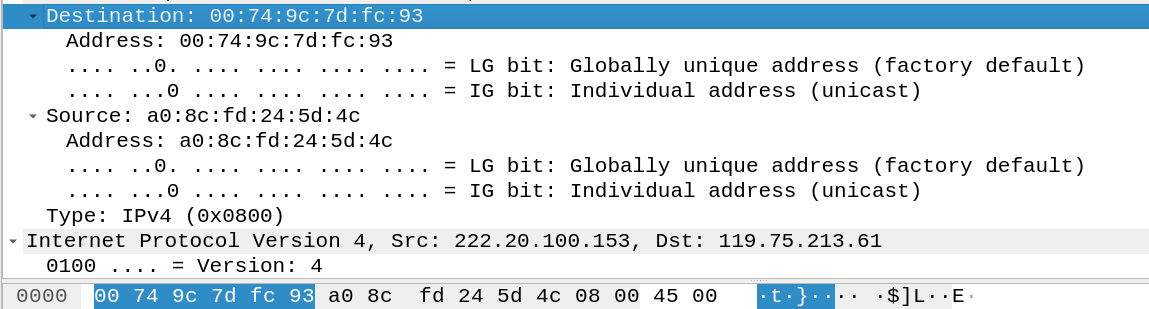


图1-7 源地址与目标地址

### 1.4.2 分析VPN

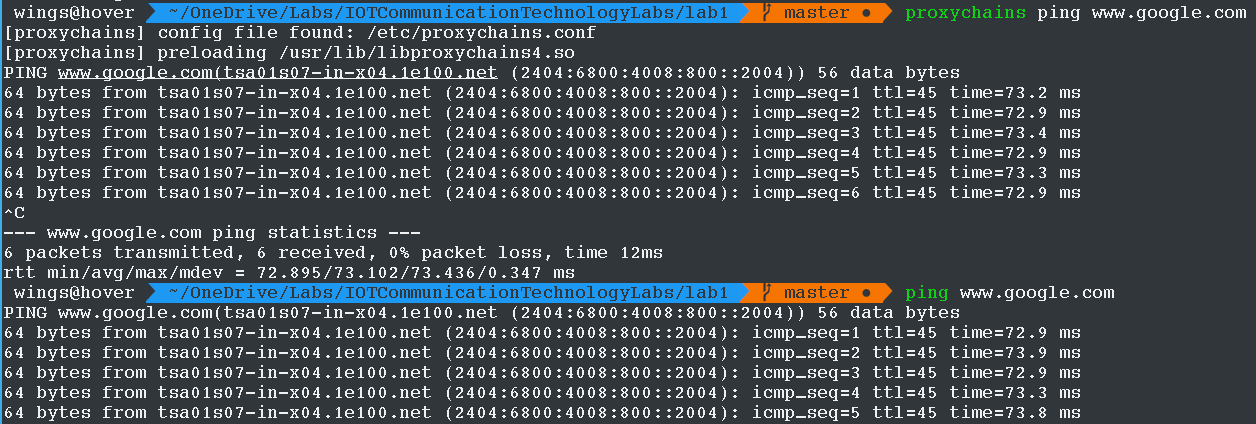


图1-8 proxychains ping

在proxychains中，对于能够直接连接且速度较快的访问请求采用直接访问形式，故实际上仍为直接访问。

对于www.google.com采用IPV6连接方式。

### 1.4.3 TraceRoute路径分析

TraceRoute的默认程序基于ICMP报文实现，而ICMP报文可能被防火墙拦截（路由器不响应）等原因无法收到反馈报文，tcpTraceRoute采用TCP “SYN”包，而如果目标地址是允许访问的，SYN标志位的数据段是TCP建立连接时进行“三次握手”的第一次握手，通常不会故增加采用了tcpTraceRoute进行测试。

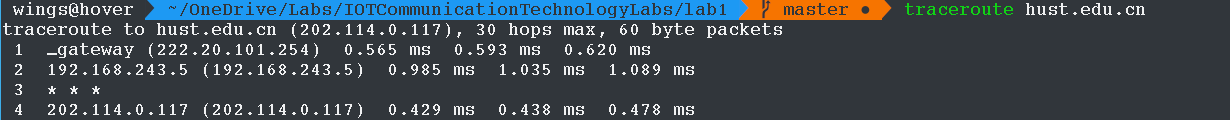


图1-9 traceroute hust.edu.cn

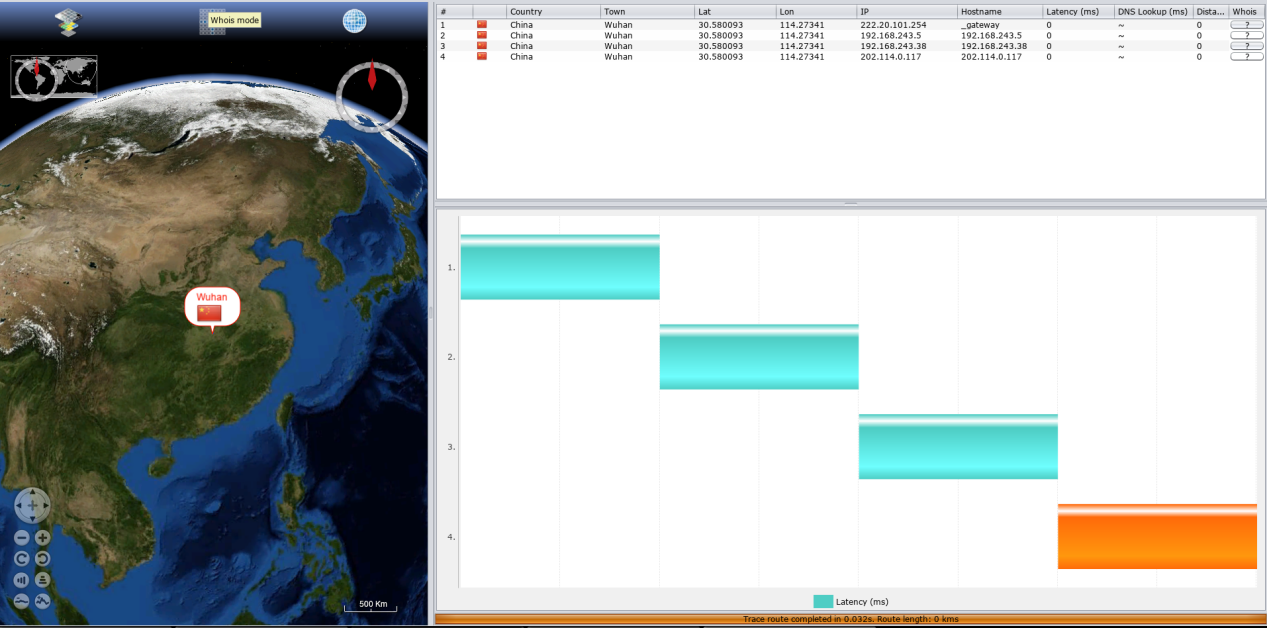


图1-10 visual-traceroute hust.edu.cn

从图中可以看出，在进行hust.edu.cn ICMP报文中，对于局域网段的返回报文不同的程序有不同的处理方式。

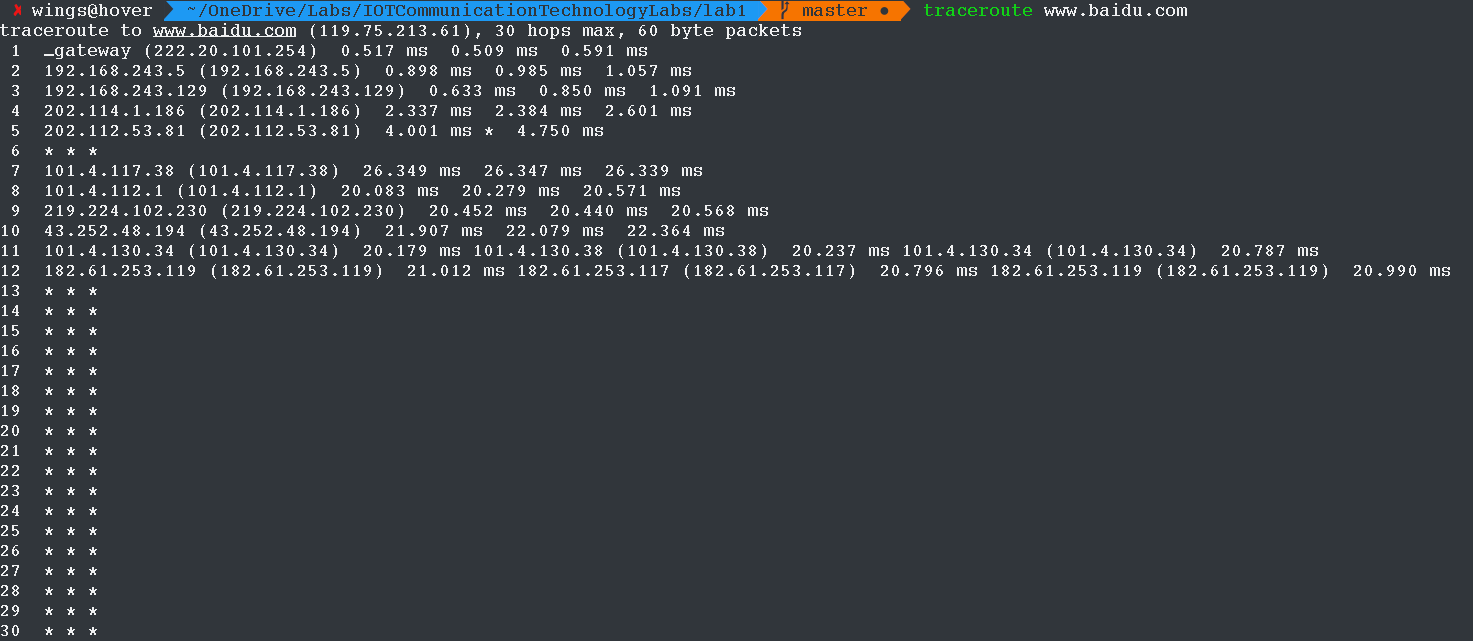


图1-11 traceroute www.baidu.com

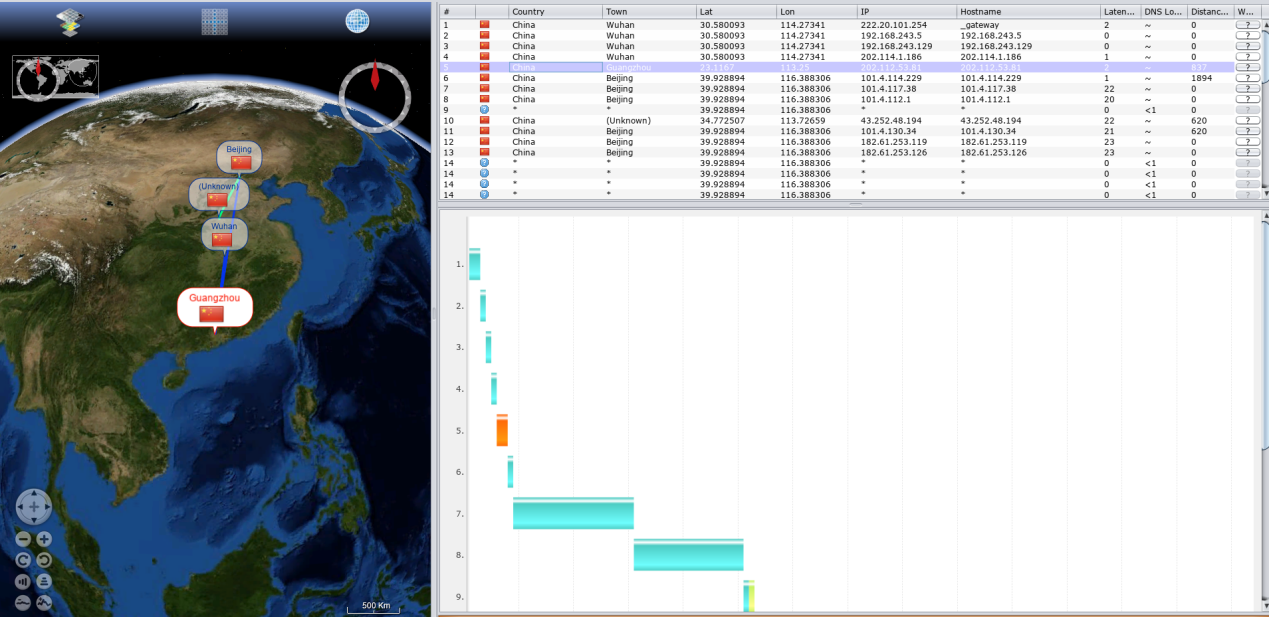


图1-12 visual-traceroute [www.baidu.com](http://www.baidu.com)

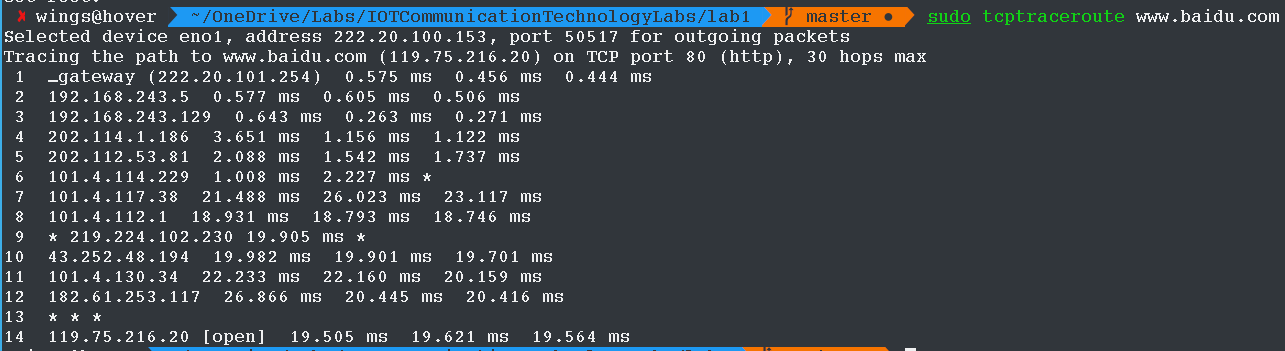


图1-13 tcptraceroute [www.baidu.com](http://www.baidu.com)

1. 与上文hust.edu.cn结果进行比较，从IP地址可以看出，均有从局域网段到广域网段的跳转。
2. 而在182.61.253.117/119处对ICMP包进行了拦截处理，故无法收到反馈报文，使用tcptraceroute可以得到反馈。
3. 可以看到使用可视化时，传播过程先到广州，再到北京，报文传输的路径选择与实际的物理距离有差异

## 1.5 实验中的问题及心得

实验过程中，因为较为熟悉网络的配置过程，故没有产生太多的困难，开始采用traceroute和ping时，想到使用proxychains进行一下测试，在过程中发现并无返回报文，应该是防火墙或者路由对与此类报文进行了阻塞处理，而进行了一次正常追踪之后，使用proxychains与正常结果相同，故猜测可能存在机制选择能够连接的方式。

分析过程使用了可视化工具帮助分析。

正常的ICMP报文对于商用服务器来说是一定的负担，故基于ICMP报文的traceroute可能被阻拦，故尝试采用基于‘SYN’包的tcptracroute，最终成功。

实验过程中，熟练了网络的debug方法，和了解了不同报文的字段结构。

## 参考文献

1. <https://wiki.archlinux.org/index.php/Network_Debugging>

**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 网络模拟器Packet Tracer的使用

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

# Lab2 网络模拟器Packet Tracer的使用

## 2.1 环境

操作系统： Manjaro x64

网络平台： PacketTracer 7.2.1

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 2.2 实验目的

1. 掌握使用Packet Tracer模拟网络场景的基本方法，加深对网络环境、 网络设备和网络协议交互过程等方面的理解。
2. 安装和配置网络模拟器软件Packet Tracer， 观察与IP网络接口的各种网络硬件及其适用场合。

## 2.3 实验内容及步骤

### 2.3.1 安装

1. 安装网络模拟器
   1. 从官网下载./tar.gz
   2. 执行./install安装
   3. Packettracer脚本运行
2. 使用网络模拟器

### 2.3.2 环境测试

两台Terminal通过Switch使用Copper Cross-over 直接连接

PC0

        IP：          192.168.1.2

        Submask：     255.255.255.0

        Gateway：     192.168.1.1

PC1

        IP：          192.168.1.3

        Submask：     255.255.255.0

        Gateway：     192.168.1.1

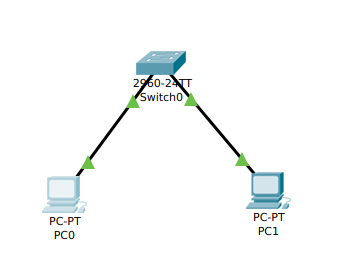


图2-1 环境测试

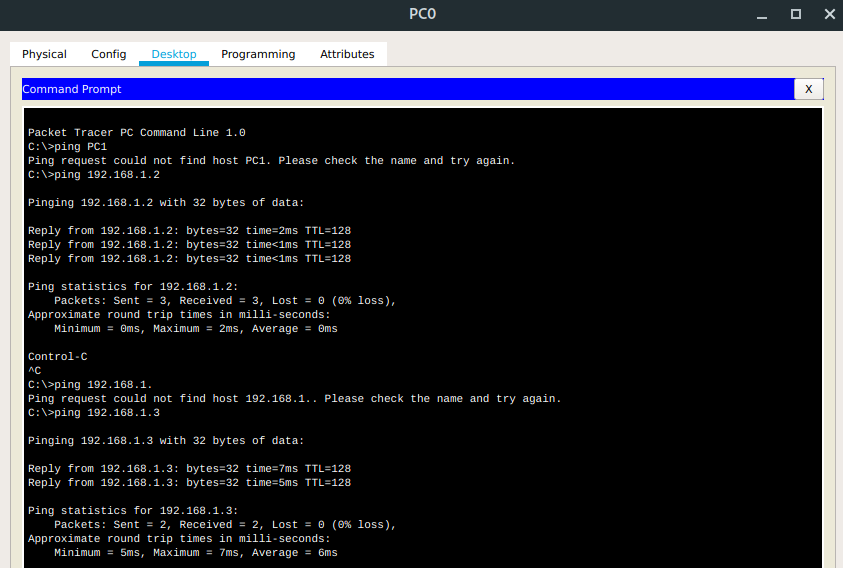


图2-2 PC0 ping PC1

### 2.3.3 交换机配置

交换机vlan 1远程管理配置

 IP：          192.168.1.3

        Submask：     255.255.255.0

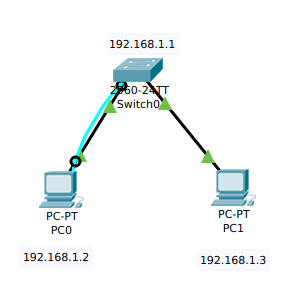


图2-3 交换机控制网络图

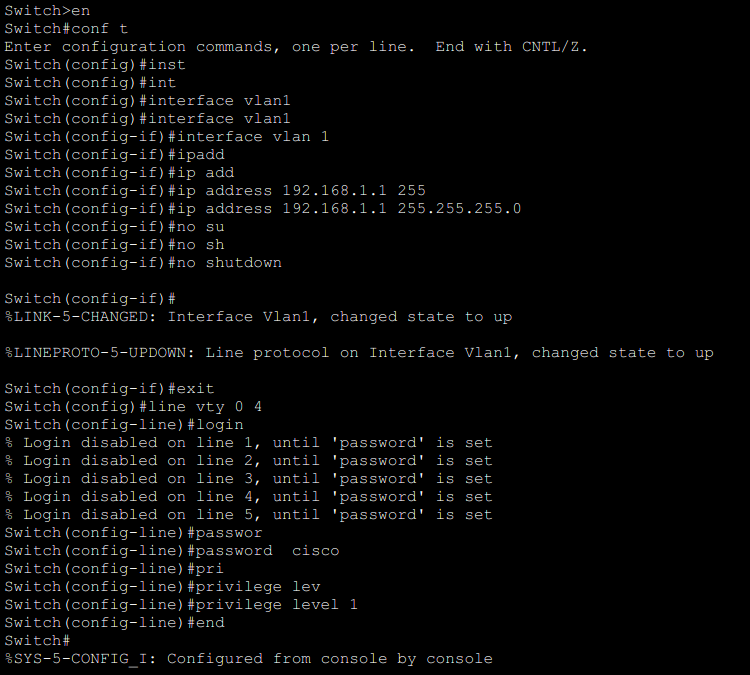


图2-4 交换机配流图

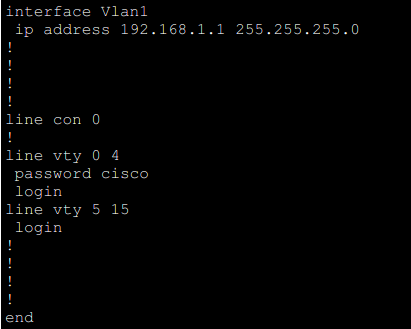


图2-5 交换机配置结果

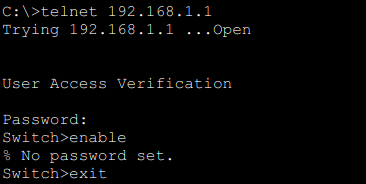


图2-6 交换机配置结果测试

### 2.3.3 链路层聚合

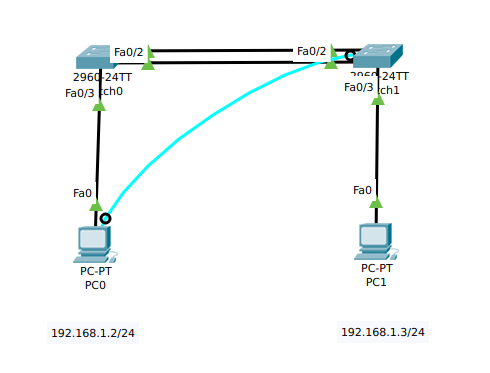


图2-7 交换机链路聚合测试网络拓扑图

交换机配置命令：

Switch>enable

Switch#config t

Switch(config)#interface range fa 0/1-2 //同时选择端口fa0/1 fa0/2

Switch(config-if-range)#Switchport mode trunk //设置端口模式为trunk

Switch(config-if-range)#channel-group 1 mode on //加入链路组1并开启

Switch(config-if-range)#exit

Switch(config)#port-channel load-balance dst-ip //按照目标主机IP地址数据分 //发来实现负载平衡

Switch(config)#exit

Switch#show etherchannel summary //显示以太信道概况

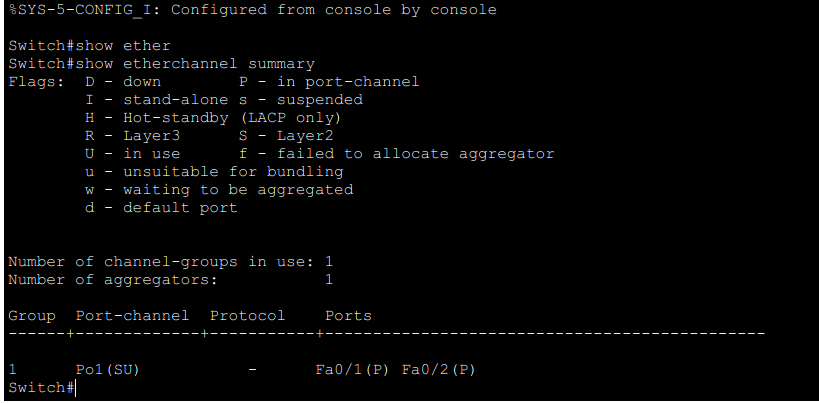


图2-8 Switch0 配置

对switch1做同样的配置

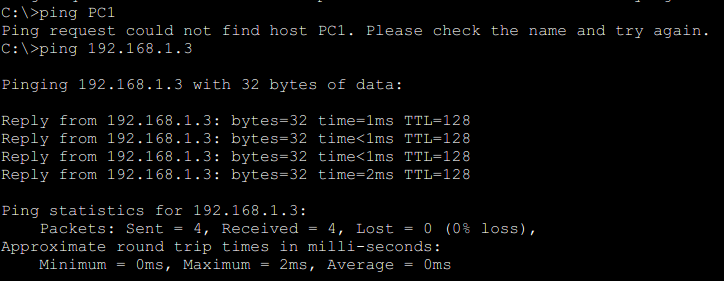


图2-9 进行ping验证

### 2.3.4 Vlan测试

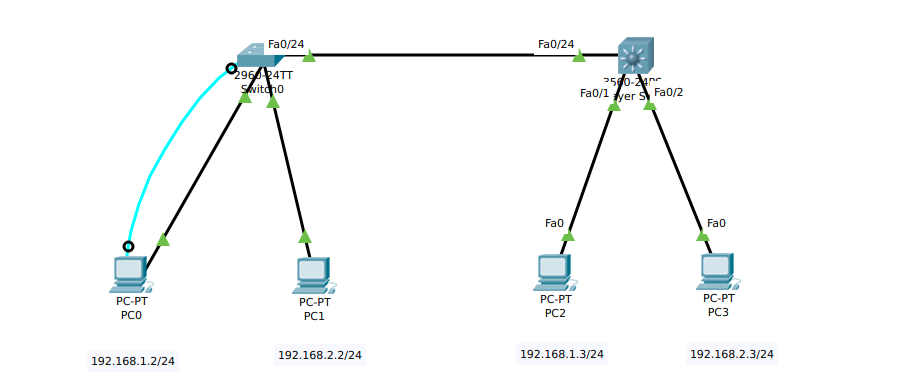


图2-10 Vlan测试网络拓扑图

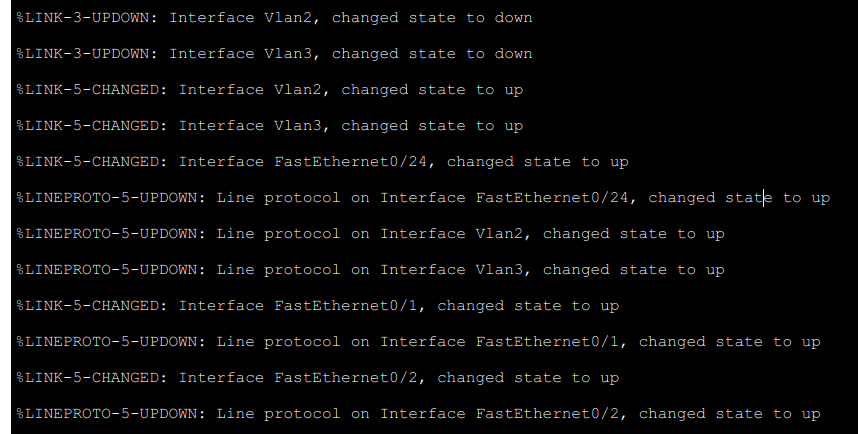


图2-11 Vlan网络配置图

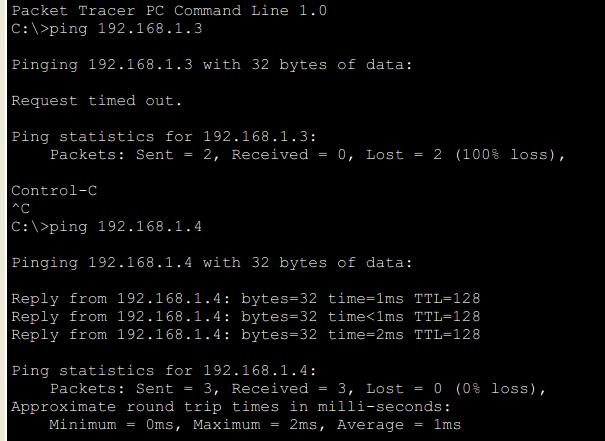


图2-12 PC0 Ping PC2 测试图

### 2.3.5 三层交换机测试

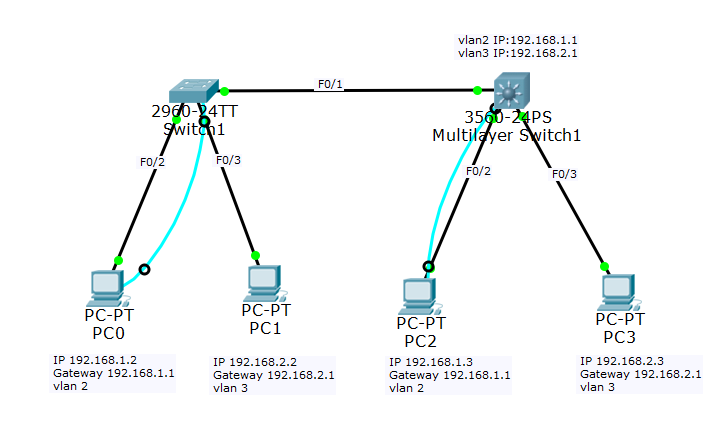


图2-13 三层交换机网络拓扑图

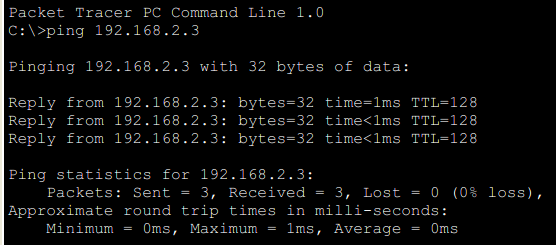


图2-14 三层交换机实现Vlan间路由转发测试图

### 2.3.4观察与IP网络接口的各种网络硬件

从PacketTracer中打开路由器2620XM的物理设备视图，仔细做下列工作：观察有关NM-1FE-FX模块描述；将其拖入设备，观察模块面板上的硬件接口情况；做笔记，并自行分析该模块的适用场合。

对路由器2620XM的NM-1FE-TX、NM-2FE2W、NM-8AM、NM cover plate模块分别做上述工作。

### 2.3.5 ping和traceroute实验

1. 创建链路
2. 配置网络
3. 配置路由器端口
4. 使用命令

## 2.4 实验结果

### 2.4.1 网络拓扑图

如实验过程所示，进行各种设备的相应配置和测试。

### 2.4.2 设备观察

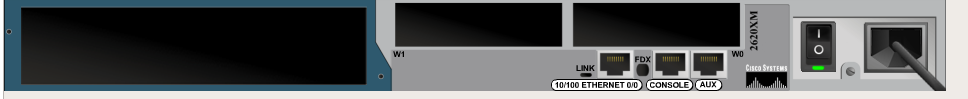


图2-15 Route2620XM缺省模块图

端口：

ETHRNET 10/100以太网端口

CONSOLE 端口

AUX 端口

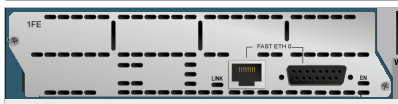


图2-16 NM-1FE-TX模块图



图2-17 NM-2FE2W模块图



图2-18 NM-8AM模块图



图2-19 NM cover plate模块图



图2-20 其余模块拓展槽

从图中我们可以看到，总共提供了Route2620三块模块拓展槽，可以使用不同的拓展模块进行接口的拓展，同时也可以使用面板覆盖进行缺省。

### 2.4.3 命令及报文跟踪

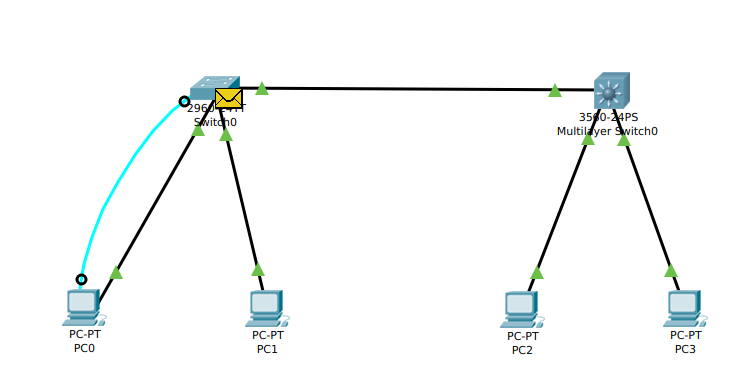


图2-21 报文追踪图

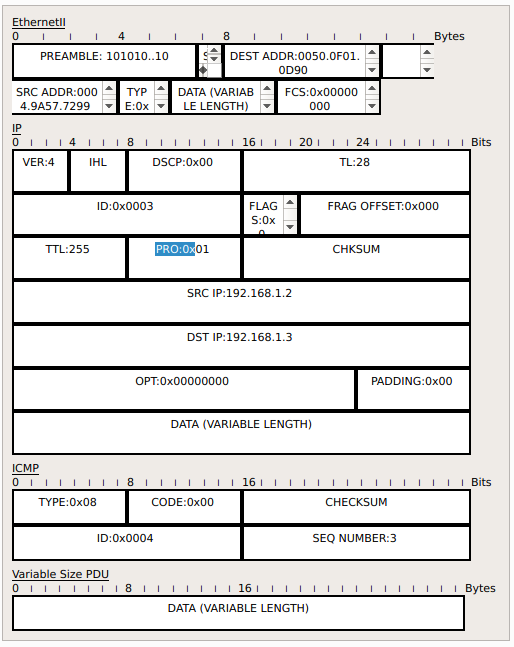
s

图2-22 报文图

## 2.5 实验中的问题及心得

Cisio Packet Tracer整个模拟软件十分有趣，实验过程中尝试了不同的器件，对于许多设备的模拟十分逼真，比如使用二层交换机的时候需要把电话的界面打开，手动插上电源线，而使用三层交换机的时候提供POE供电，此时还可以分析整个交换机的负载功率。

对于交换机，三层交换机可以充当路由功能或者交换机功能，在交换机的配置平台中可以进行切换，而交换机提供了VLAN可以灵活划分节点的广播域且容易修改，提高了灵活性，而Packet Tracer也提供了思科私有的ISL协议实现交换机之间的VLAN中继。

对于线的使用上，同类设备交叉线，异类设备直通线，这么设计的目的是因为网口标准是相同的情况下，线序是对称的，需要把发送端口接上接通端口故需要使用交叉线，而中间如果有中继设备则提供转接服务。

而且，PacketTracer中可以设置物理环境，包括光照，大气环境，辐射强度，进行真实的物理环境下的网络模拟。

报文跟踪中，可以看到，由于仿真环境，直接划分不同设备的层次，可以明显看到在路由器报文的层次变化，以及各层次的报文头。

整个实验十分有趣，学习了PacketTracer中各种元器件的仿真使用，且其命令行配置与真实环境基本一致，同时尝试了物联网设备的使用。

## 参考文献

1. CiscoPacketTracer网络实验手册
2. Cisco Packet Tracer 实验教程 https://blog.csdn.net/al\_assad/article/details/70255987

**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 分析Ethernet II帧、分析集线器和交换机工作原理

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

# Lab3 分析Ethernet II帧、集线器和交换机工作原理

## 3.1 环境

操作系统： Manjaro-4.18.10-1 x86\_64 (Arch-Based Distribution)

网络平台： Wireshark 2.6.3

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 3.2 实验目的

1. 分析Ethernet II帧
   1. 深入理解Ethernet II帧结构。
   2. 基本掌握使用Wireshark分析俘获的踪迹文件的基本技能。
2. 分析集线器和交换机工作机理
   1. 观察交换机处理广播和单播报文的过程。
   2. 比较交换机与集线器工作过程。
   3. 掌握使用Packet Tracer模拟网络场景的基本方法，加深对网络环境、 网络设备和网络协议交互过程等方面的理解。

## 3.3 实验内容及步骤

1. 分析Ethernet II帧
   1. 分析踪迹文件中的帧结构

用Wireshark俘获网络上收发分组或者打开踪迹文件，选取感兴趣的帧进行分析。

* 1. 分析以太帧结构

1. 分析集线器和交换机工作机理
   1. 在PacketTracer中配置网络拓扑
   2. 观察交换机如何处理广播和单播报文
   3. 观察交换机如何处理未知单播

## 3.4 实验结果

### 3.4.1 分析Ethernet II帧

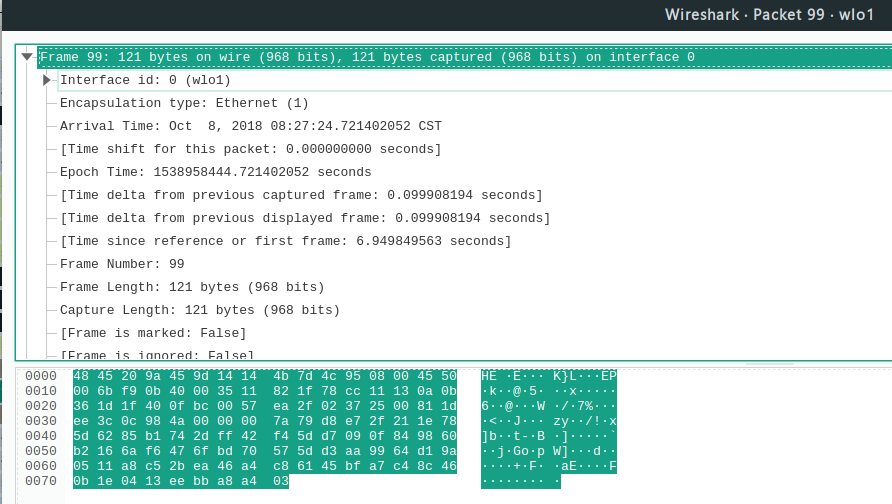


图3-1 物理层Frame结构

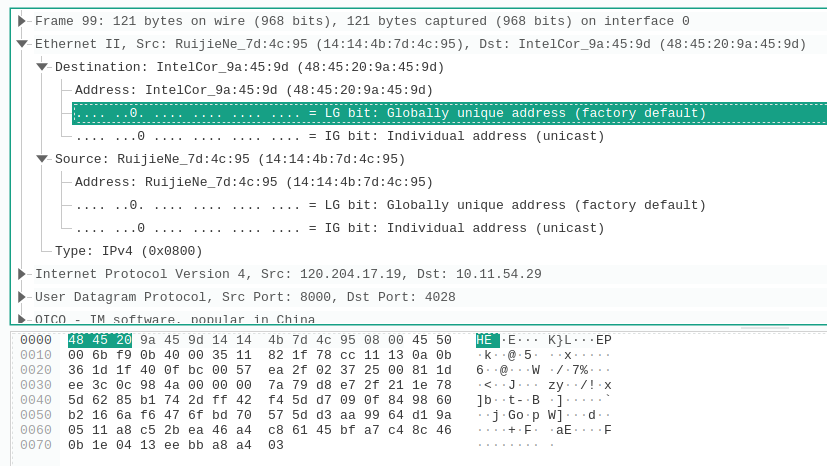


图3-1 链路层Ethernet II帧结构

可以看到，报文结构为，Frame结构为整个链路层传输的PDU，封装Ethernet II作为链路层PCI,此时传输层IP报文段作为链路层的SDU。

1. 目标MAC地址：

Destination: IntelCor\_9a:45:9d (48:45:20:9a:45:9d)

我们可以看到其目标MAC地址品牌为IntelCor。

此时需要对是否为同一子网进行区分，此时MAC地址涉及arp过程

A ping B

如果在同一网段：

* 1. 主机先会查看自己的ARP高速缓存中是否有B主机的MAC地址记录。
     1. 如果A主机的高速缓存中有B主机的记录，则直接通过这个MAC地址进行数据的传输。
     2. 如果A主机的高速缓存中没有B主机的记录，则会向局域网的所有主机广播一个ARP请求，寻找B主机  
        的MAC地址。
  2. 当B主机收到A主机广播的ARP请求后，就会直接给A主机回复一个ARP数据包。
  3. 当A主机收到B主机发送过来的请求后，将B的MAC地址写入高速缓存中，然后通过该MAC地址，A主机向B主机进行数据的传输。

如果不在同一网段：

网关把目标MAC改成下一跳路由器的MAC地址（通过ARP得到），而源端MAC改成发出端口的MAC地址，否则下一跳路由器收到目标MAC不是自己的数据包，会丢弃不予理睬，下一跳路由器再发给下一跳路由器同样要把目标MAC地址改为下一跳路由器的MAC地址再发出去。

故此处MAC地址不为实际的服务器MAC地址，而为网关地址。

1. 源目标MAC地址：

Source: RuijieNe\_7d:4c:95 (14:14:4b:7d:4c:95)

可以看到前缀反映了物理网卡的品牌信息，而这种信息对应了一定的前缀MAC地址编码。

在MAC地址中，使用LG Bit/IG Bit进行标识MAC地址类型

1. 网络协议类型：IPV4

如果为IPV6协议2字节以太类型字段中值为0x0800，它表示以太帧包含了 IP数据报，如果其中封装了IPv6协议，其值应为0x86DD。

1. 用户数据报文UDP
2. 应用层报文 OICP

### 3.4.2 分析集线器和交换机工作机理

1. 网络拓扑图及IP配置
   1. 集线器测试

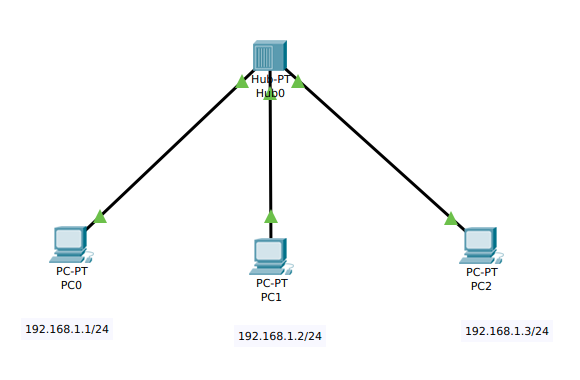


图3-1 集线器网络拓扑图

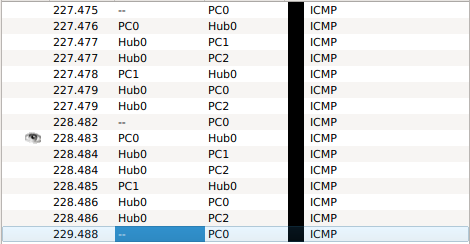


图3-1 PC0 PING PC1测试

从图中可以看出，集线器采用广播模式，对于报文对所有的端口进行转发。

* 1. 交换机环路测试

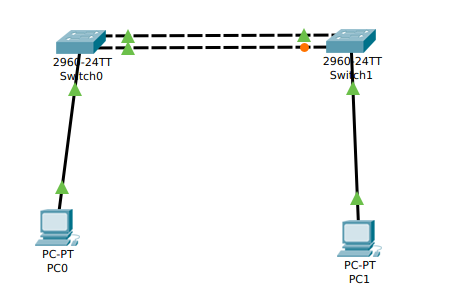


图3-1 交换机环路测

* + 1. 默认情况下STP协议是启用的。通过两台交换机之间传送BPDU协议数据单元。选出跟交换机、根端口等，以便确定端口的转发状态。图中标记为黄色的端口处于block堵塞状态。
    2. 设置RSTP。
    3. 查看交换机show spanning-tree状态，了解跟交换机和根端口情况。

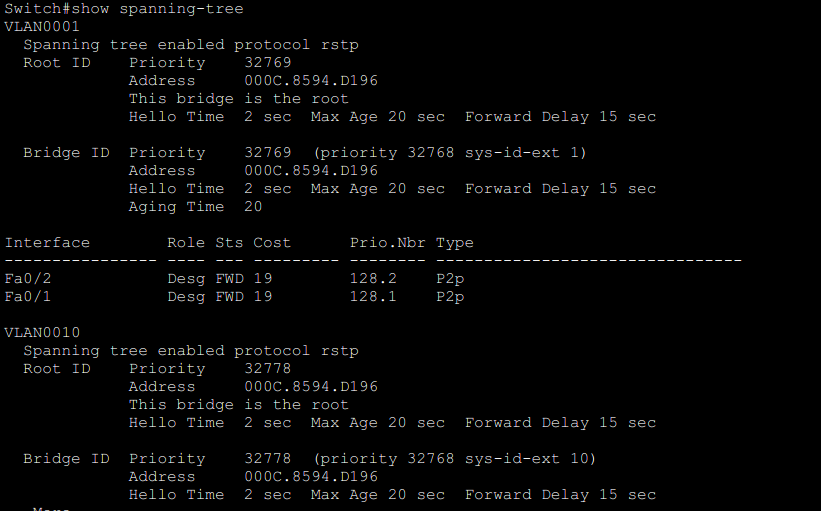


图3-1 查看生成树协议

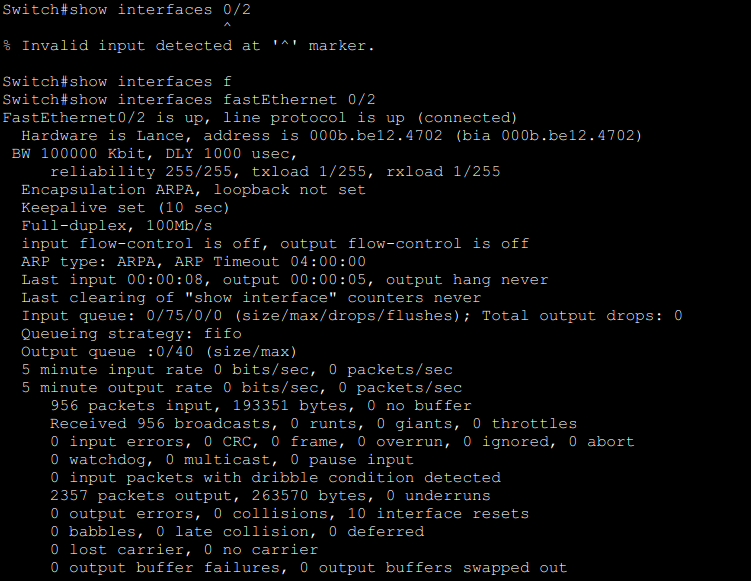


图3-2 查看端口流量

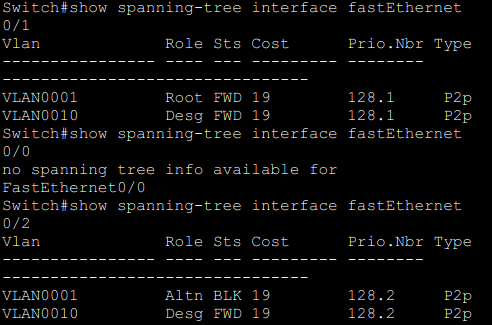


图3-3 查看端口状态

返回参数说明：

StpVersion : RSTP 生成树协议的版本

SysStpStatus : Enabled 生成树协议运行状态，disable 为关闭状态

Priority : 32768 交换机的优先级

RootCost : 200000 交换机到达根交换机的开销

RootPort : Fa0/1 交换机上的根端口

或：

RootCost: 0 交换机到达根交换机的开销，0 代表本交换机为根

RootPort: 0 交换机上的根端口，0 代表本交换机为根

可以看到0/1端口，即上端的端口具有较高的优先级(较小的Priority)，故为开启状态。而0/2端口为阻塞状态。

* + 1. 通过更改交换机生成树的优先级spanning-tree vlan 10 priority 4096可以变化跟交换机的角色。
    2. 测试。当主链路处于down状态时候，能够自动的切换到备份链路，保证数据的正常转发。

可以看出，在刚刚进行连接的时候，交换机其中的链路表未收敛，需要一段时间。

1. 广播报文
2. 单播报文

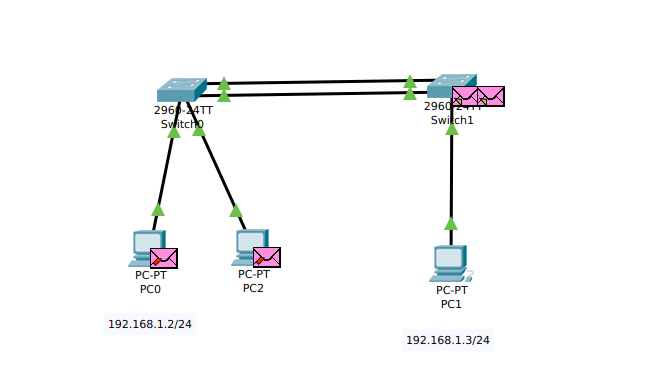


图3-4 广播报文

1. 未知单播报文

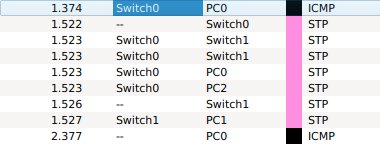


图3-5 初始交换机未知单播测试

### 3.4.3 单步调试集线器及广播报文

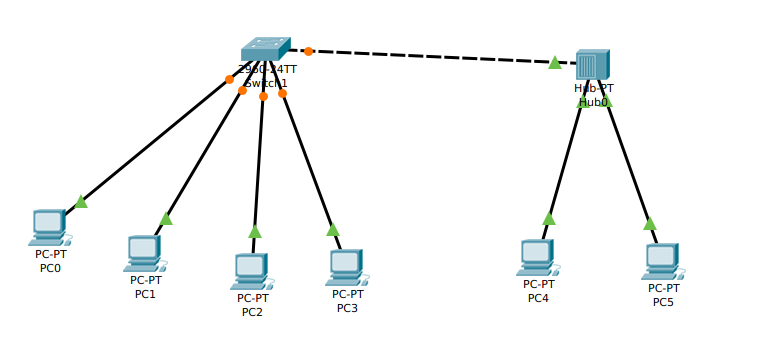


图3-6 网络的初始状态

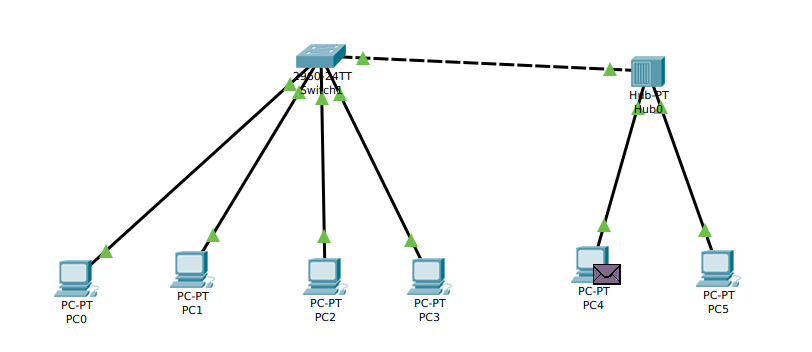


图3-7 默认执行STP后的状态



图3-8 集线器所有端口转发



图3-9 已知PC0 MAC地址的情况下直接进行转发

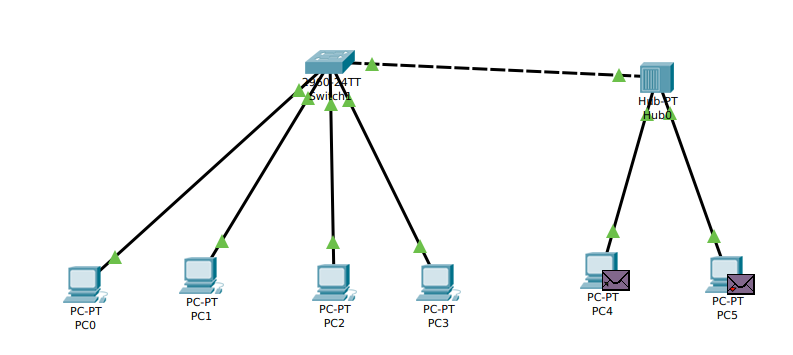


图3-10 回复报文再次进行转发

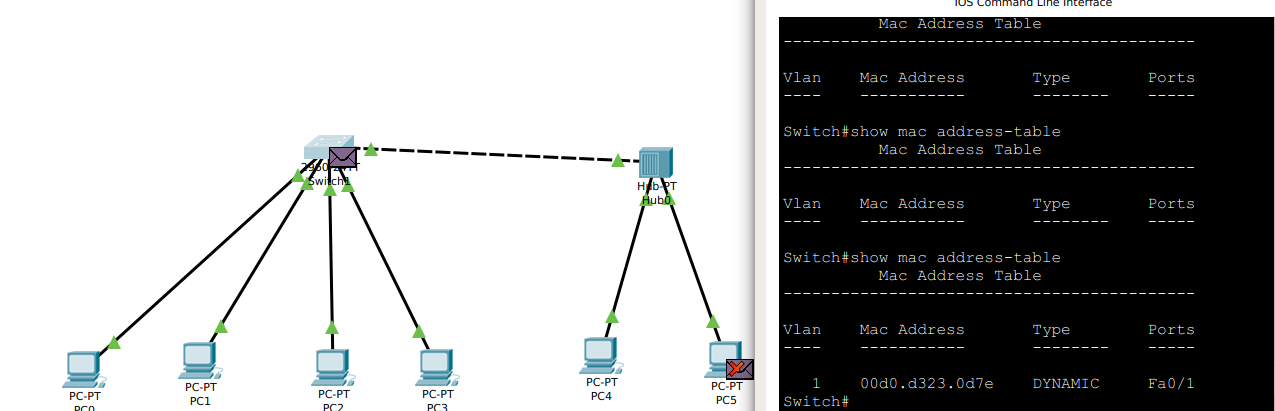


图3-11 此时添加PC5 MAC地址

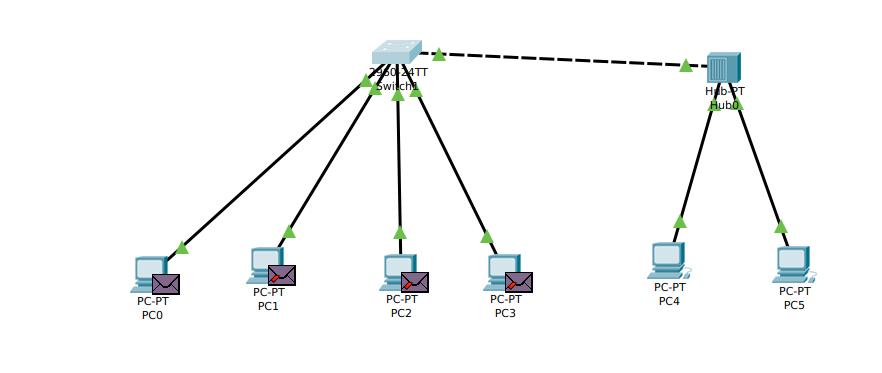


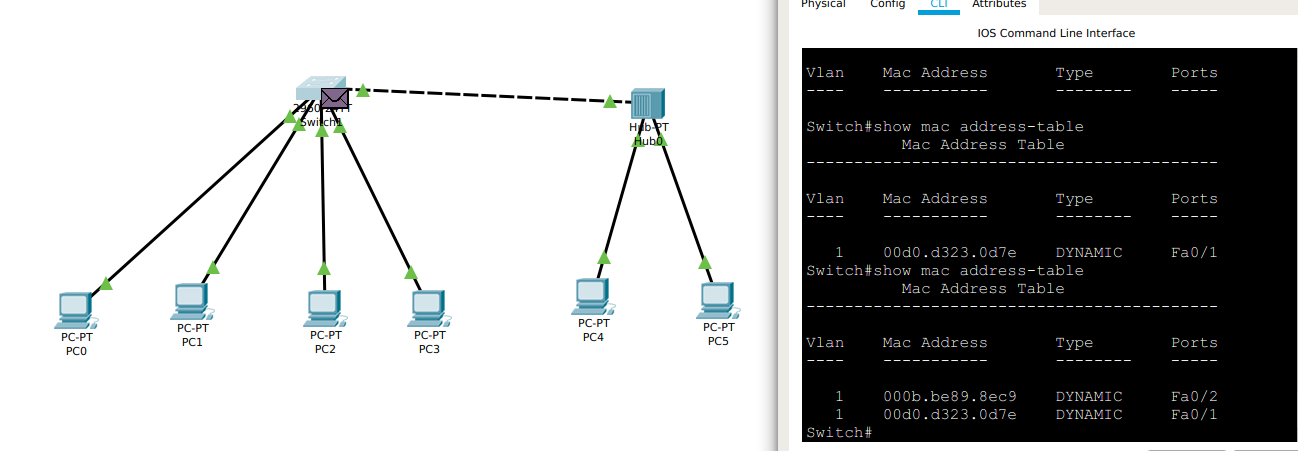
图3-12 广播报文

图3-13此时添加PC0 MAC地址

集线器工作在物理层，仅对电信号进行放大整形并向所有端口转发，不识别数据链路层的帧，不执行CSMA/CD协议。

## 3.5 实验中的问题及心得

在网络拓扑的架构中，比较标准的操作是先进行配置再进行连线，类似于电路中的先进行电路设置，再进行连线操作，因为不配置的话，容易造成广播风暴，造成交换机长时间不收敛。

普通集线器广播策略不传播至本身的原因是集线器为物理结构，不提供控制功能，故其中的线连接方式直接通过物理实现，自然不会形成回环，而在实验中也得到了验证。

而交换机采用转发表的过程，其中可能会形成环路，在未进行生成树配置的时候，可以看到存在环路的存在，默认STP开启进行处理，但是在较大的网络中国存在收敛较慢，且链路强制阻塞，效率并不高，采用RSTP进行尝试，可以看到在根节点选择不同的情况下，会有不同的链路生成树，在进行配置的时候可验证，不同的vlan共享同一个生成树，并不可按照vlan阻塞链路。

实验过程中熟悉了关于集线器，路由器，交换机不同层级的概念，加深了网络层次的理解。

## 

## 参考文献

1. CiscoPacketTracer网络实验手册

**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 分析IP和ARP协议

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

# Lab4 分析IP和ARP协议

## 4.1 环境

操作系统： Manjaro-4.18.10-1 x86\_64 (Arch-Based Distribution)

网络平台： Wireshark 2.6.3

网络环境：

inet 222.20.100.153  netmask 255.255.254.0  broadcast 222.20.101.255   
inet6 2001:250:4000:803c:e9a0:1b72:2be3:b69e  prefixlen 64  scopeid 0x0<global>   
inet6 fe80::3781:b076:5845:ec9d  prefixlen 64  scopeid 0x20<link>   
ether a0:8c:fd:24:5d:4c  txqueuelen 1000  (Ethernet)

## 4.2 实验目的

1. 深入理解IP报文结构和工作原理。
2. 深入理解ARP协议的工作原理：

理解IP和以太网协议的关系，掌握IP地址和MAC地址的映射机制，搞清楚IP报文是如何利用底层的以太网帧进行传输的。

## 4.3 实验内容及步骤

1. 分析IP报文结构
   1. 分析俘获的分组
   2. 分析IP报文结构
2. 分析ARP协议
   1. 查看本机因特网硬件地址
   2. 使用ARP命令
   3. 分析ARP协议工作过程

## 4.4 实验结果

### 4.4.1 分析IP协议

1. 选取报文分析

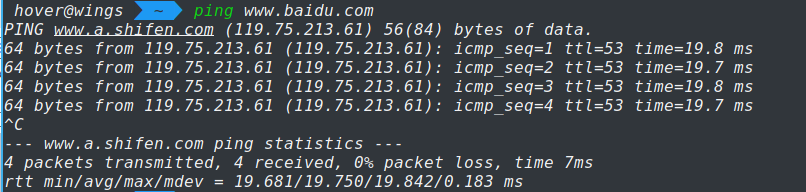


图4-1 Ping [www.baidu.com](http://www.baidu.com)

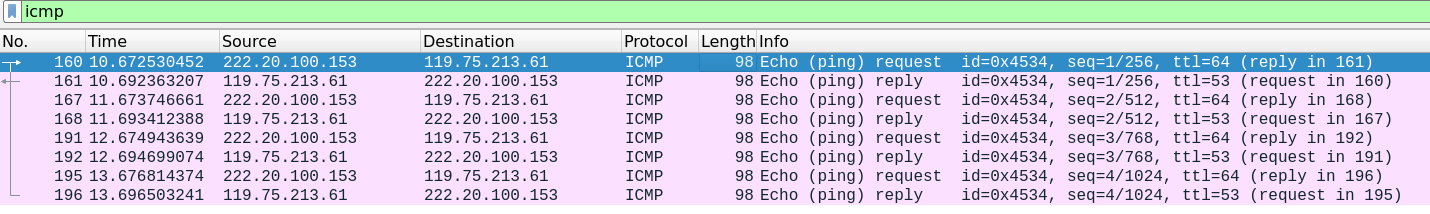


图4-2 Ping [www.baidu.com](http://www.baidu.com) wireshark抓包

1. 分析IP报文结构

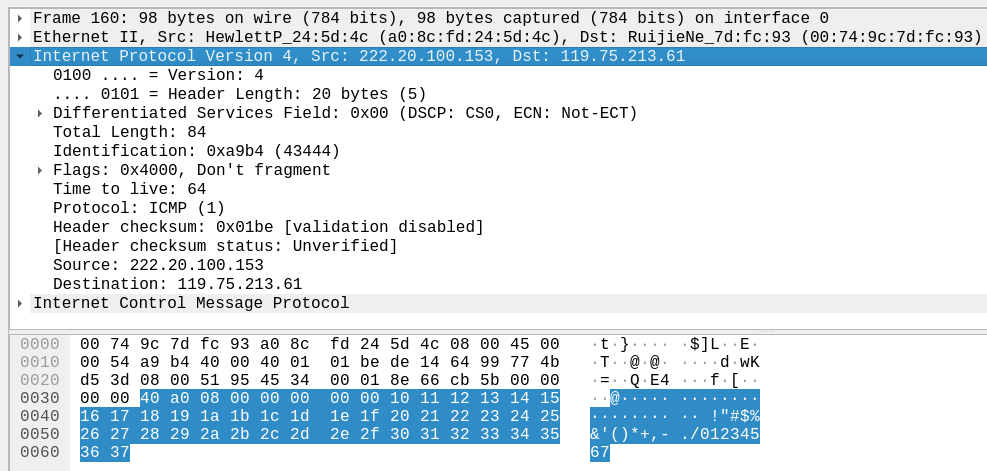


图4-3 IP部分报文选取

可以看到，报文结构为，Frame结构为整个链路层传输的PDU，封装Ethernet II作为链路层PCI,此时传输层IP报文段作为链路层的SDU，在之后

Frame报文描述：

Frame 160: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

Ethernet II报文描述：

Ethernet II, Src: HewlettP\_24:5d:4c (a0:8c:fd:24:5d:4c), Dst: RuijieNe\_7d:fc:93 (00:74:9c:7d:fc:93)

IP报文描述：

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

差分字段：

表4-1差分字段类型

| 业务类型 | DSCP PHB | DSCP值 |
| --- | --- | --- |
| Network Control | CS7(111000) | 56 |
| IP Routing | CS6(110000) | 48 |
| Interactive Voice | EF(101110) | 46 |
| Interactive Video | AF41(100010) | 34 |
| Video control | AF31(011010) | 26 |
| Transactional/Interactive(对应高优先级应用) | AF2x(010xx0) | 18、20、22 |
| Bulk Data(对应中优先级应用) | AF1x(001xx0) | 10、12、14 |
| Streaming Video | CS4(000100) | 4 |
| Telephony Signaling | CS3(000011) | 3 |
| Network Management | CS2(000010) | 2 |
| Scavenger | CS1(000001) | 1 |
| Best Effort | 0 | 0 |

此时为Best Effort，在传统的IP网络中多为Best Effort，对报文传送的可靠性、传送延迟等性能不提供任何保证。

而在特殊的情形下，需要对QoS进行保证，故需要采用标志位。

且提供了IPV4与IPV6的服务类型接口，同时在骨干网和边缘网的同时可能需要进行服务类型的转换，此时此标志位需要映射。

Total Length: 84

Identification: 0xa9b4 (43444)

标识：

当数据报由于长度超过网络的MTU而必须分片时，这个标识字段的值就被复制到所有的数据报片的标识字段中。相同的标识字段的值使分片后的各数据报片最后能正确地重装成为原来的数据报。

Flags: 0x4000, Don't fragment

1. .. .... .... .... = Reserved bit: Not set

.1.. .... .... .... = Don't fragment: Set

..0. .... .... .... = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 64

Protocol: ICMP (1)

Header checksum: 0x01be [validation disabled]

[Header checksum status: Unverified]

Source: 222.20.100.153

Destination: 119.75.213.61

之后为数据段部分，即IP协议的SDU，为整个ICMP报文段。



图4-4 ICMP报文段

### 4.4.1 分析ARP协议

1. 查看本机MAC地址

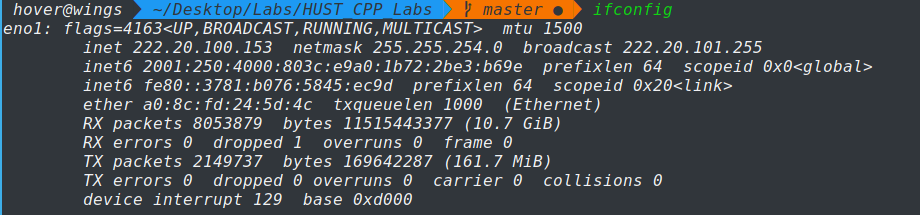


图4-5 查看本机MAC地址图

1. 查看本机ARP表

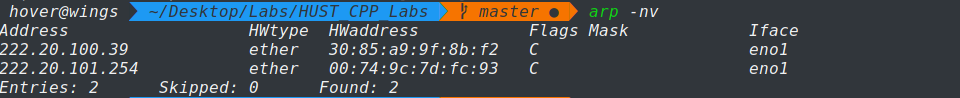


图4-6 查看本地arp表图

1. 清除ARP表

可使用ip neigh flush dev <dev>

或者arp -d 删除

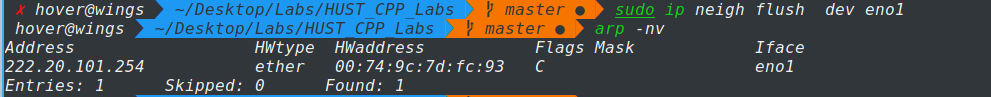


图4-7 删除本地arp表图



图4-8 删除本地arp表验证图

此时为了测试ARP表清空之后的重建过程，因为网络环境的差异，与下面实验环境稍有不同。

1. arp过程分析

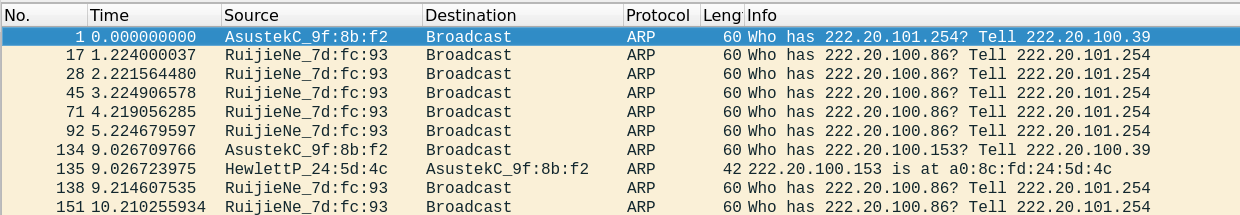


图4-9 本地arp报文抓取图

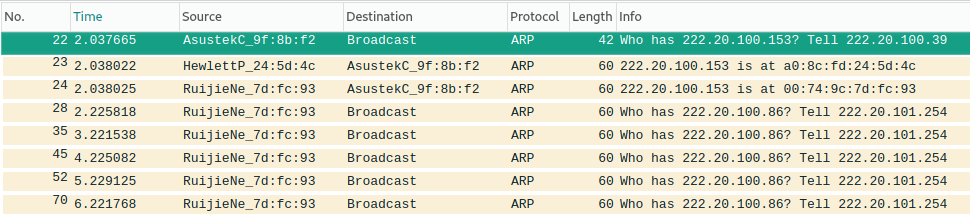


图4-10 目标主机arp报文抓取图

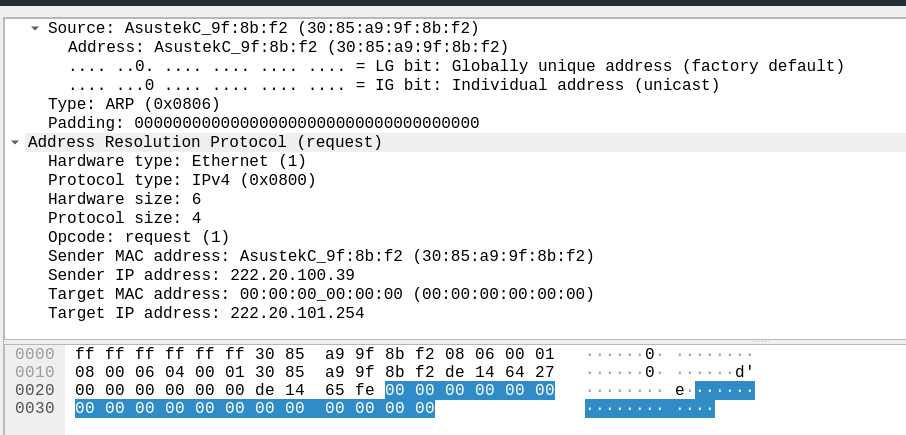


图4-11主机arp报文字段图

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: RuijieNe\_7d:fc:93 (00:74:9c:7d:fc:93)

Sender IP address: 222.20.100.39

Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)

Target IP address: 222.20.101.254

只知道IP地址而不知道广播地址，故MAC地址为缺省值

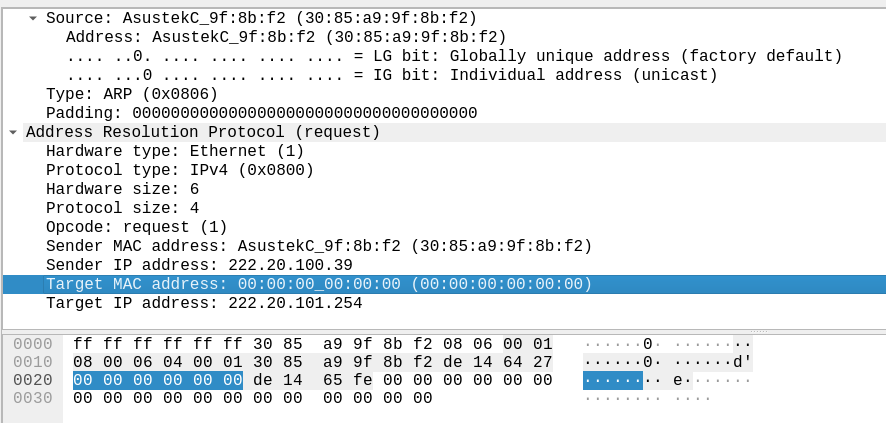


图4-12 MAC地址图

路由器接受到这样的报文，不会向网络进行转发这样的广播包的，在子网内转发包的时候，MAC地址会被修改后再转发。

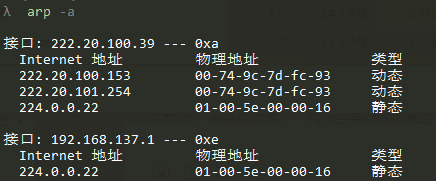


图4-13 接收端arp表图

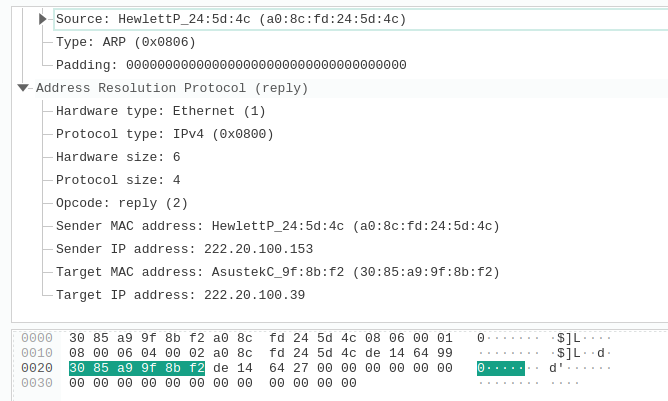


图4-14 arp回复报文图

此时已经填充目标MAC地址，不过此时为网关的MAC地址。

### 4.4.3 相关问题

1. IP地址

222.20.100.153

1. IP数据报首部，较高层协议字段中的值是什么？

icmp报文，较高层协议字段的值为1

1. IP首部字节，载荷字节

.... 0101 = Header Length: 20 bytes (5)

Total Length: 84

载荷字段有64个字节

1. 是否分段？

.1.. .... .... .... = Don't fragment: Set

1. 高层协议有用信息
   1. Hardware type
   2. Protocol type
   3. Hardware size
   4. Protocol size
   5. Opcode
   6. Sender MAC address
   7. Sender IP address
   8. Target MAC address
   9. Target IP address

## 4.5 实验中的问题及心得

在实验的过程中，曾经出现过主机A Ping 主机B超时，主机B Ping 主机A成功，此时主机A为Linux环境，B为Windows环境，考虑二者网络管理策略不同，两台主机ARP表有表项更新的情况，经过检查，当主机B开启系统自带防火墙时，作为目的节点被Ping会出现超时的情况，防火墙将ping分组拦截了下来，导致返回超时，但不影响其他数据包的通讯，故该主机仍会返回ARP应答报文，也正因如此两者能正常通讯。

在arp报文中，我们知道其MAC地址会被网关替换，实际上这是网络下层对上层透明的思想，即底层提供服务，而无需知道如何实现，对于链路层，只需要知道转发的链路即可，是不是最终的MAC地址，不重要。但此时想到该如何获取实际的目标MAC地址呢？通过查阅资料，可使用NetBIOS协议进行询问，获取远程主机的实际MAC地址。在windows环境下，提供nbstat命令进行同一子网的实际MAC地址查询，而如果非同一子网，则需要封装“UDP-NetBIOS-NS”询问包进行问询。

## 参考文献

1. 实验文档
2. IP 网络 QoS

<https://blog.csdn.net/jianchaolv/article/details/7926537>

**华中科技大学计算机学院**

**《计算机通信与网络》实验报告**

实验名称 分析IP和ARP协议

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 班 级 | 学 号 | 得 分 |
| 潘翔 | IOT1601 | U201614898 |  |

教师评语：

# Lab5 配置路由器的路由选择协议

## 5.1 环境

操作系统： Manjaro-4.18.10-1 x86\_64 (Arch-Based Distribution)

网络平台： Wireshark 2.6.3

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 5.2 实验目的

1. 深入理解路由器中路由选择协议的工作原理。
2. 能够配置路由器的路由选择协议RIP。

## 5.3 实验内容及步骤

1. 生成并配置网络拓扑
2. 配置IP
3. 配置路由器选路协议
4. 检查路由器选路协议的效果

## 5.4 实验结果

### 5.4.1 单臂路由实验

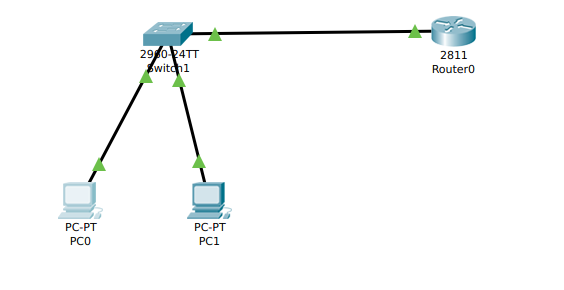


图5-1 单臂路由网络拓扑图

**Switch0配置**

en

conf t

vlan 2

exit

int fa 0/10

switchport access vlan 2

exit

int vlan 1

ip address 192.168.1.1 255.255.255.0

no shutdown

exit hpor

int vlan 2

ip address 192.168.2.1 225.255.255.0

no shutdown

end

show int vlan 1

conf t

router rip

network 192.168.1.0

network 192.168.2.0

version 2

**Router0配置**

en

conf t

host R1

inf fa 0/0

ip address 192.168.2.2 255.255.255.0

no shutdown

int fa 0/1

ip address 192.168.3.1 255.255.255.0

no shutdown

exit

router rip

network 192.168.2.0

version 2

router ospf 1

network 192.168.3.0 0.0.0.255 area 0

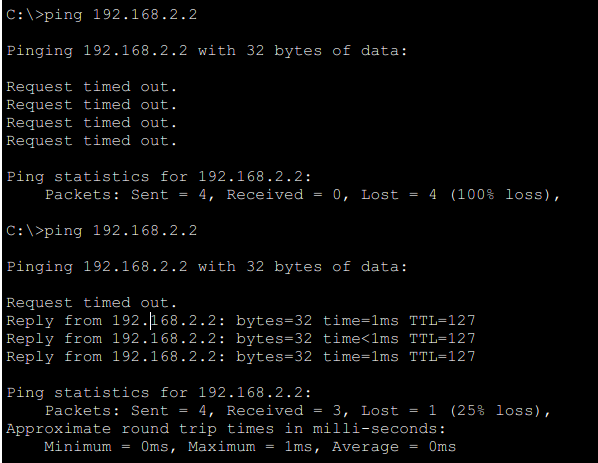


图5-2 单臂网络测试

### 5.4.2 综合路由实验

1. PC与交换机间用直连线连接；PC与路由、路由与路由之间用交叉线连接。
2. 在三层上划分2个Vlan，运行RIPV2协议；R1运行OSPF协议。
3. 在路由器R0上左侧配置RIPV2路由协议；右侧配置OSPF协议。
4. 在R0路由进程中引入外部路由，进行路由重分布。
5. 将PC1、PC2主机默认网关分别设置为与直接网络设备接口IP地址。
6. 验证PC1、PC2主机之间可以互相通信；

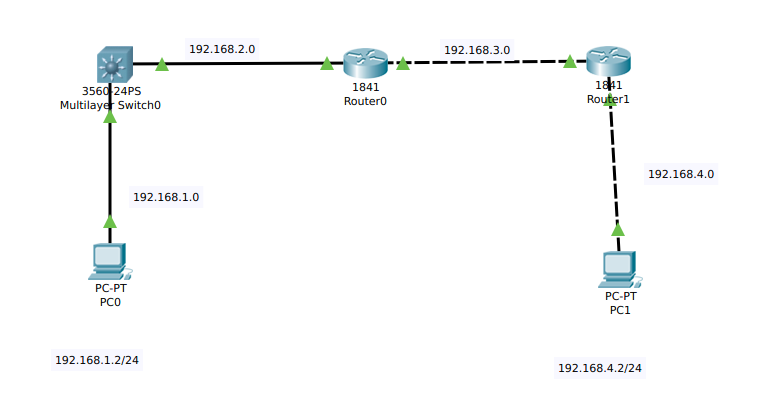


图5-3 综合路由网络拓扑图

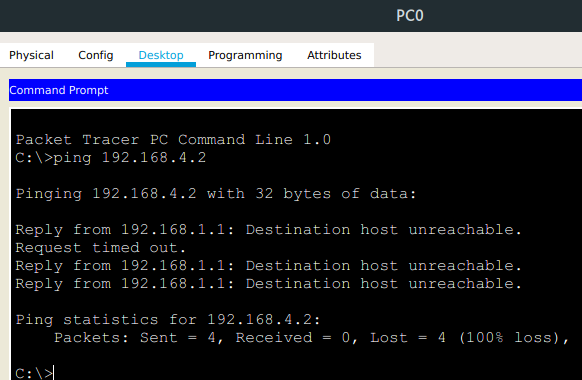


图5-4 路由未配置ospf下的反馈信息

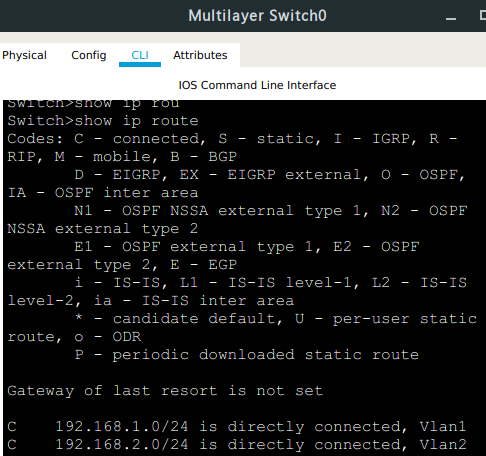


图5-5 查看路由信息



图5-6 进行PC1 PC2联通路由测试

### 5.4.3 路由环路实验

1. 网络拓扑

本次实验使用的网络拓扑中包含了3 台 CISCO 1841 路由器和 两台PC。标配的 CISCO 1841 路由器仅带有两个 10/100Mbps 的以太端口，而路由器 0 和路由器 2 需要 3 个端口，因此要为它们再增加一个以太端口。此时需在路由器的物理设备视图中关闭路由器电源，增加一个WIC-1ENET 模块后重启路由器，此时路由器增加了一个 10 Mbps 以太网接口，可用于连接 PC。此外，为了减少交换机，各网络设备之间的连接线缆使用交叉线。

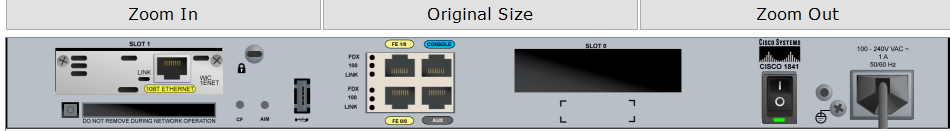


图5-7 CISCO1841路由器物理视图-添加WIC-1ENET模块

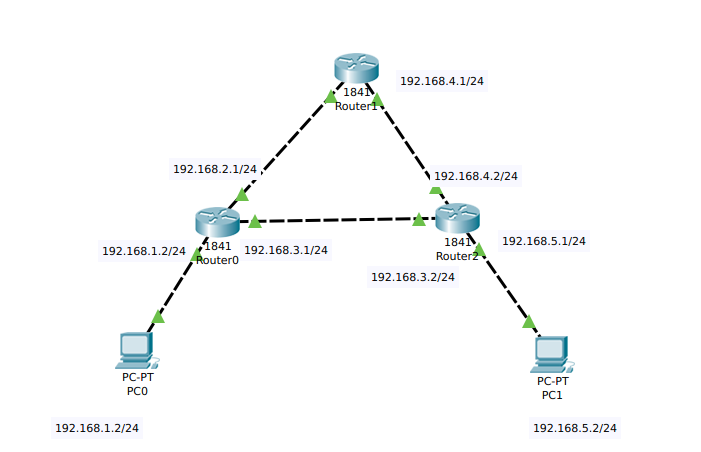


图5-7 路由网络拓扑图

进行如下的配置

表5-1 PC IP地址配置表

|  |  |  |  |
| --- | --- | --- | --- |
| 设备名 项目名 | IP地址 | 子网掩码 | 默认网关 |
| PC0 | 192.168.1.2 | 255.255.255.0 | 192.168.1.1 |
| PC1 | 192.168.5.2 | 255.255.255.0 | 192.168.5.1 |

表5-2 路由器IP地址配置表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 端口 | IP地址 | 子网掩码 |
| Router0 | FastEthernet0/0 | 192.168.1.1 | 255.255.255.0 |
| FastEthernet0/1 | 192.168.2.1 | 255.255.255.0 |
| Ethernet0/1/0 | 192.168.3.1 | 255.255.255.0 |
| Router1 | FastEthernet0/0 | 192.168.2.2 | 255.255.255.0 |
| FastEthernet0/1 | 192.168.4.1 | 255.255.255.0 |
| Router2 | FastEthernet0/0 | 192.168.4.2 | 255.255.255.0 |
| FastEthernet0/1 | 192.168.5.1 | 255.255.255.0 |
| Ethernet0/1/0 | 192.168.3.2 | 255.255.255.0 |

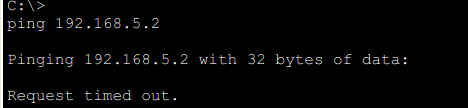


图5-8 初始情况下PC0 ping PC1

配置RIP Route选路协议

1. 回答下列问题：
2. 路由器接口不够的情况下，如何增加？

关闭路由器开关，从左侧列表选择对应模块拖入路由器的空闲插槽，重新启动路由器。

1. 网络配置完成后，如果连接显示为红色，如何解决？

检查各个设备的IP配置以及开关的状态。

1. IP层互联互通的条件是什么？

物理层连接正确，链路层配置正确，IP地址、网关等配置正确。

1. 实验成功后，给出三个路由器的路由表，并对其内容进行说明

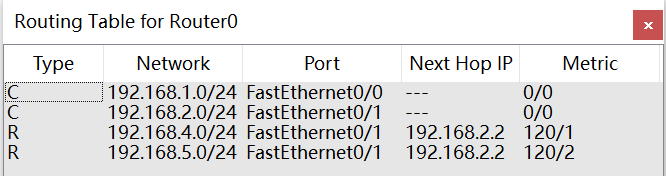


图4-7 Router0路由表

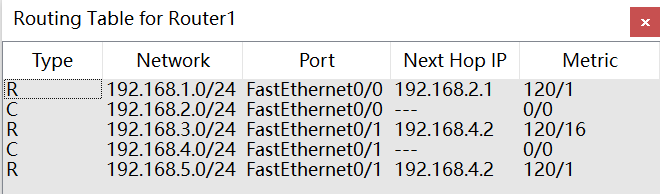


图4-8 Router1路由表

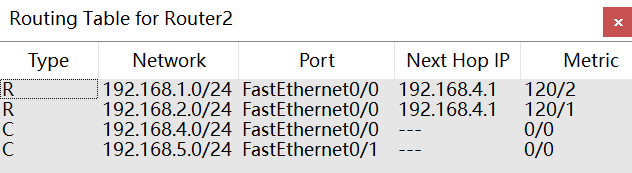


图4-9 Router2路由表

可以看到Router0的路由表中，由于人为断开了Router0和Router2之间的链接，PC0和PC1的路由信息都是空的，同样，Router2的路由表中，PC0和PC1的路由信息也是空的。

## 5.5 实验中的问题及心得

其中Route1841支持多个路由进行多个路由协议进程，系统软件提供了路由信息从一个路由进程重分布到另一个路由进程的功能。可以将OSPF路由域的路由重新分布后RIP路由域中，也可以将RIP路由域的路由重新分布后通告到OSPF路由域中。路由的相互重分布可以在所有的IP路由协议之间进行，经过测试，可以实现路由的跨协议信息通告。

在单臂路由中和

## 参考文献

1. CiscoPacketTracer网络实验手册
2. Cisco Packet Tracer 实验教程

https://blog.csdn.net/al\_assad/article/details/70255987

# Lab6 分析TCP特性

## 6.1 环境

操作系统： Manjaro-4.18.10-1 x86\_64 (Arch-Based Distribution)

网络平台： Wireshark 2.6.3

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 6.2 实验目的

1. 深入理解TCP的如下重要机制的工作原理：利用序号和确认号实现可靠数据传输，TCP拥塞控制算法（慢启动和拥塞避免），接收方通告的流量控制。
2. 掌握用Wireshark分析TCP踪迹文件的技能。

## 6.3 实验内容及步骤

1. 俘获本机与远程服务器的TCP踪迹文件

使用Wireshark俘获从本机到远程服务器之间的TCP踪迹文件，并将俘获本机收发的TCP报文段存入踪迹文件tcp.cap中，为了方便实验，此处使用已经俘获保存的踪迹文件tcp-ethereal-trace-1.pcap。

1. 熟悉TCP踪迹文件
2. 踪迹文件反应了TCP和IP之间的通信过程，时间戳基于踪迹文件开始追踪的时间

可以看到俘获机器与gaia.cs.umass.edu的Web服务器之间交互的TCP和 HTTP报文序列。从图中可以看到发起三次握手的SYN报文，也可以看到一 系列交互的HTTP报文。

1. 分析TCP序列/应答编号和流量控制
2. 分析应用层内容
3. 分析TCP拥塞控制

## 6.4 实验结果

## 6.5 实验中的问题及心得

## 参考文献

1. CiscoPacketTracer网络实验手册

# Lab7 利用Java(C++)开发网络应用程序

## 7.1 环境

操作系统: Manjaro-4.18.10-1 x86\_64 (Arch-Based Distribution)

开发语言: C/C++

编辑器: Visual Studio Code 1.27.2 x64

编译器: g++ (GCC) 8.2.1 20180831

调试器: GNU gdb (GDB) 8.2

构建工具: GNU Make 4.2.1

网络环境：

Link encap:Ethernet  HWaddr a0:8c:fd:24:5d:4c     
inet addr:222.20.100.153  Bcast:222.20.101.255  Mask:255.255.254.0   
inet6 addr: fe80::2476:27:cd9d:d75b/64 Scope:Link   
inet6 addr: 2001:250:4000:803c:e3c1:b69:d9f2:67b0/64 Scope:Global

## 7.2 实验目的

1. 基本掌握利用C++开发环境调试应用程序的方法。
2. 理解基于套接字开发网络应用程序的过程，深入理解Ping工作原理。
3. 深入理解HTTP协议的格式和工作过程，理解Web代理服务器工作原理。

## 7.3 实验内容及步骤

### 7.3.1 myping

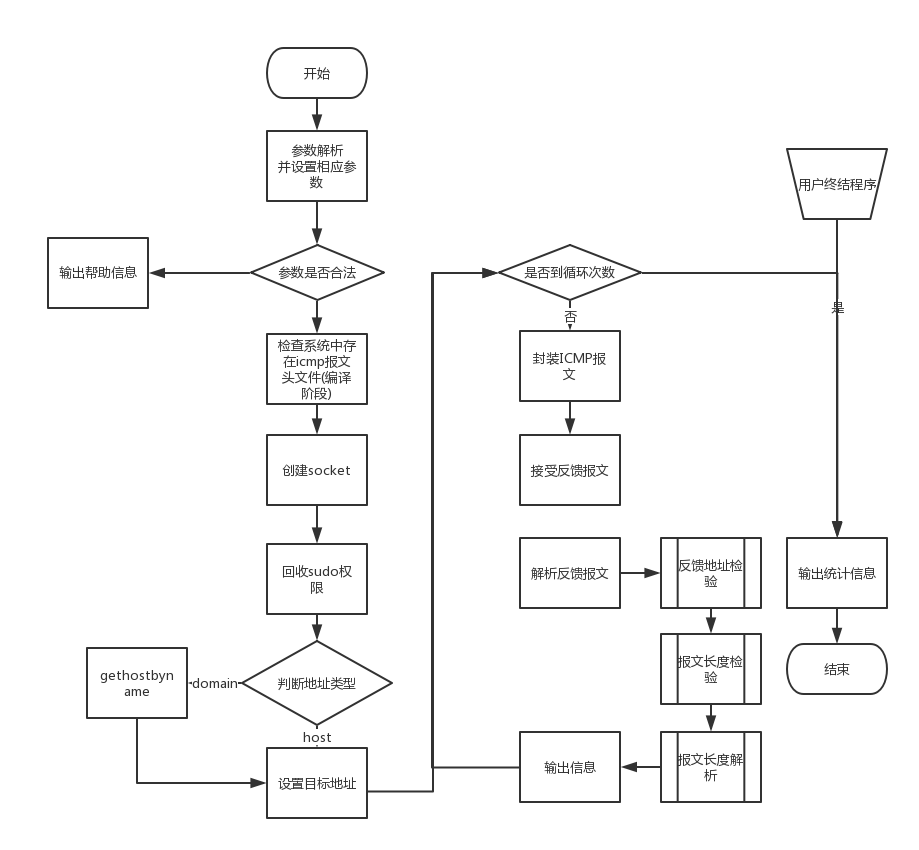


图7-1 myping函数流程图

### 7.3.2 myWebServer

1. server\_base
   1. request 结构体定义
   2. 初始化
   3. 装载资源
   4. 默认的request处理函数
2. server\_http
   1. 初始化服务器
      1. 指定端口
      2. 线程数量
   2. 重写accept函数
      1. m\_io\_service资源池中取出socket
      2. 异步接受
         1. 调用accept函数
         2. 如果错误，调用默认process\_request函数
3. server\_https
   1. 初始化服务器
      1. 指定端口
      2. 线程数量
      3. 证书文件
      4. 私钥文件
   2. 重写accept函数
      1. m\_io\_service资源池中取出socket
      2. 异步接受
         1. 利用ssl对于报文IO Stream加密
         2. 如果错误，调用默认process\_request函数

图7-2 server\_https流程图

## 7.4 实验结果

### 7.4.1 ping实验结果

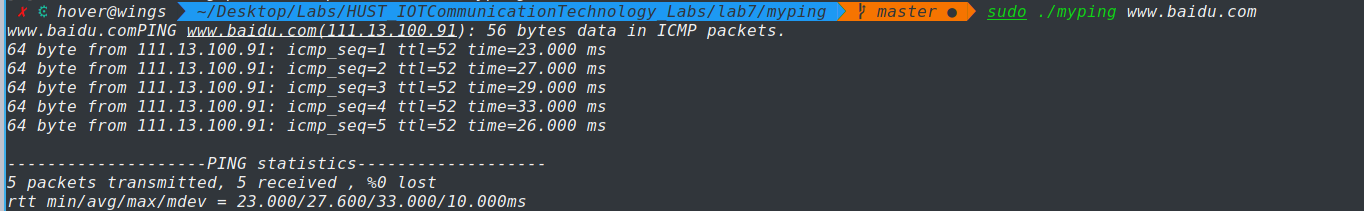


图7-3 myping [www.baidu.com](http://www.baidu.com)

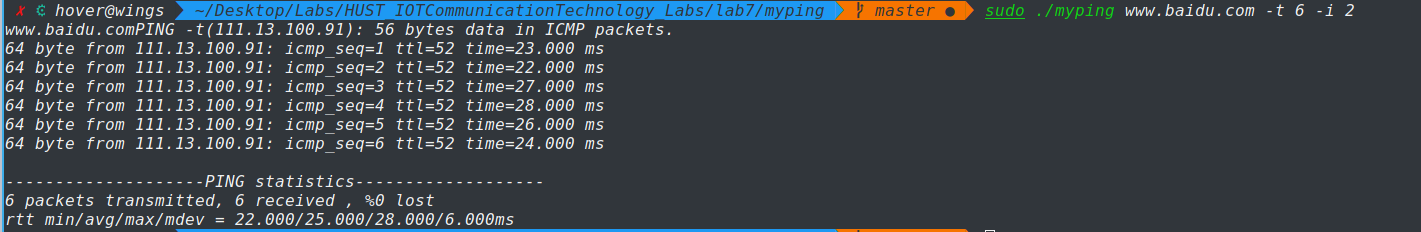


图7-4 myping [www.baidu.com](http://www.baidu.com) 参数解析测试



图7-5 ping www.baidu.com对比图

### 7.4.2 myWebServer

## 7.5 实验中的问题及心得

在ping的开发过程中，熟悉了UDP和ICMP的报文结构，其中对于大小端的处理，操作系统采用了宏判断的形式，而ICMP报文结构体内使用union形式，对是否回应做出了定义，结构清晰，且进行封装的时候不容易出错，这些值得我们自己在实现协议是进行学习，而在ping的系统实现中，和traceroute一样，可以选择指定协议类型，避免如同lab1中部分服务器出现的对于icmp报文不响应的状态。

Socket的创建需要root权限，但是系统的ping并不需要，是因为使用raw socket的使用，可以采用为应用程序授权的方式免root执行。

WebServer实现过程中，主要依赖Boost库进行，因为其有良好的多线程管理和异步IO操作，同时利用C++11特性，利用智能指针进行服务器资源管理和线程池实现，避免了引用计数错误和内存泄露问题。其中设计server\_base作为基类，定义公共属性，便于不同的服务器进行版本实现，利用server\_http和server\_https进行继承实现主要是其中构造函数和accept方法的重写。使用handler类进行方法类的抽离，减少耦合性，便于添加新的方法。

其中，Boost库为我们提供了一些很好的示范，包括在进行网络流传输的时候，接受到的流可能大于bufferSize,传统的做法为舍弃溢出部分并报错，而推荐做法为解析头文件后对于溢出部分进行拼接。

整个过程熟悉了http和https应用层协议，锻炼了网络编程技巧，并学习了相关协议的实现。

## 参考文献

1. Boost.Asio Docement

<https://www.boost.org/doc/libs/1_61_0/doc/html/boost_asio.html>

1. UNIX Network Programming,3th