**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](#_Toc360302089)

[1.1 环境 2](#_Toc894587175)

[1.2 实验目的 2](#_Toc1353334990)

[1.3 实验内容及步骤 2](#_Toc835080477)

[1.4 实验结果 2](#_Toc1156627571)

[1.5 实验中的问题及心得 6](#_Toc397333046)

[参考文献 7](#_Toc1420440947)

[Lab2 网络模拟器Packet Tracer的使用 9](#_Toc1471312194)

[2.1 环境 9](#_Toc1946670349)

[2.2 实验目的 9](#_Toc479187345)

[2.3 实验内容及步骤 9](#_Toc299595065)

[2.4 实验结果 12](#_Toc1021258584)

[2.5 实验中的问题及心得 13](#_Toc670189836)

[参考文献 13](#_Toc637193378)

[Lab3 分析Ethernet II帧、集线器和交换机工作原理 14](#_Toc1961992066)

[3.1 环境 14](#_Toc991856536)

[3.2 实验目的 14](#_Toc1636354207)

[3.3 实验内容及步骤 14](#_Toc1294913427)

[3.4 实验结果 15](#_Toc1270332580)

[3.5 实验中的问题及心得 15](#_Toc1719787243)

[参考文献 15](#_Toc9336447)

[Lab4 分析IP和ARP协议 16](#_Toc682374080)

[4.1 环境 16](#_Toc292445933)

[4.2 实验目的 16](#_Toc772572762)

[4.3 实验内容及步骤 16](#_Toc2065279157)

[4.4 实验结果 16](#_Toc591974057)

[4.5 实验中的问题及心得 17](#_Toc153285672)

[参考文献 17](#_Toc571374237)

[Lab5 配置路由器的路由选择协议 18](#_Toc1350948290)

[5.1 环境 18](#_Toc274060473)

[5.2 实验目的 18](#_Toc2134028110)

[5.3 实验内容及步骤 18](#_Toc1711250380)

[5.4 实验结果 18](#_Toc1168647648)

[5.5 实验中的问题及心得 18](#_Toc1339879452)

[参考文献 18](#_Toc398847209)

[Lab6 分析TCP特性 19](#_Toc177791571)

[6.1 环境 19](#_Toc1737212498)

[6.2 实验目的 19](#_Toc1819288156)

[6.3 实验内容及步骤 19](#_Toc1649103765)

[6.4 实验结果 19](#_Toc1536399199)

[6.5 实验中的问题及心得 19](#_Toc150991853)

[参考文献 19](#_Toc1948698831)

[Lab7 利用Java(C++)开发网络应用程序 20](#_Toc410174135)

[7.1 环境 20](#_Toc821181690)

[7.2 实验目的 20](#_Toc438408561)

[7.3 实验内容及步骤 21](#_Toc224682554)

[7.4 实验结果 22](#_Toc1813038226)

[7.5 实验中的问题及心得 23](#_Toc2074762768)

[参考文献 23](#_Toc1519595981)

**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

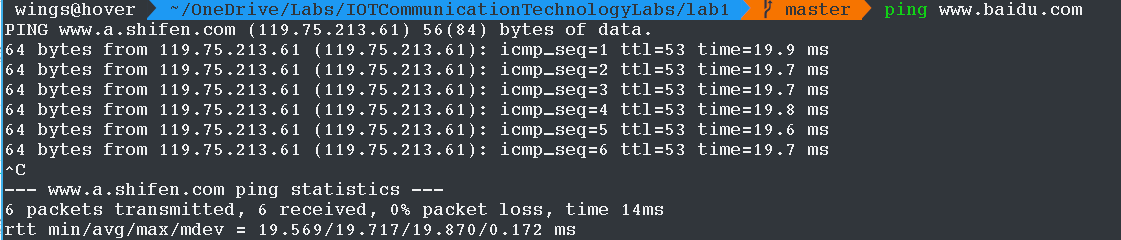


图1-1 ping [www.baidu.com](http://www.baidu.com)结果

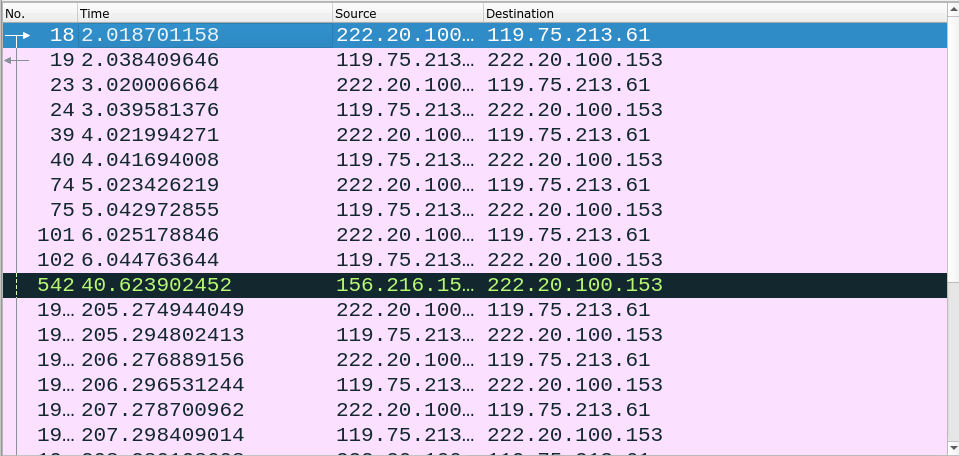


图1-2 wireshark结果



图1-3 ICMP报文结构

协议类型：V4

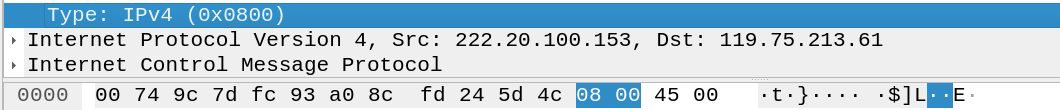


图1-4 协议类型

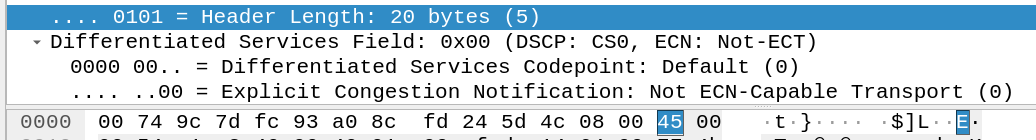


图1-5 首部长度

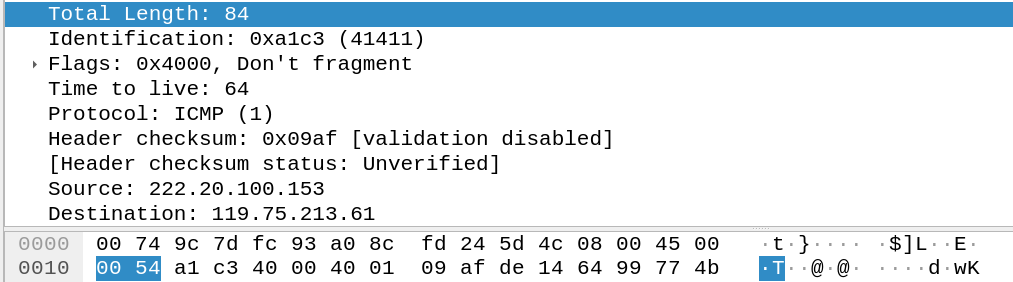


图1-6 总长度

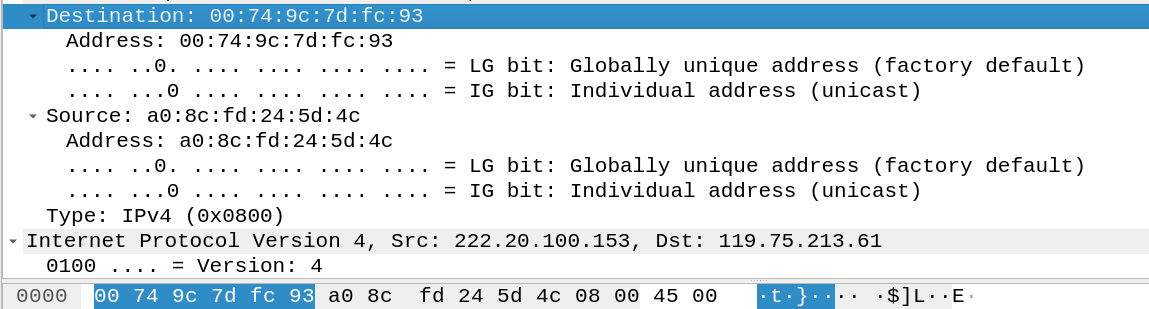


图1-7 源地址与目标地址

### 1.4.2 分析VPN

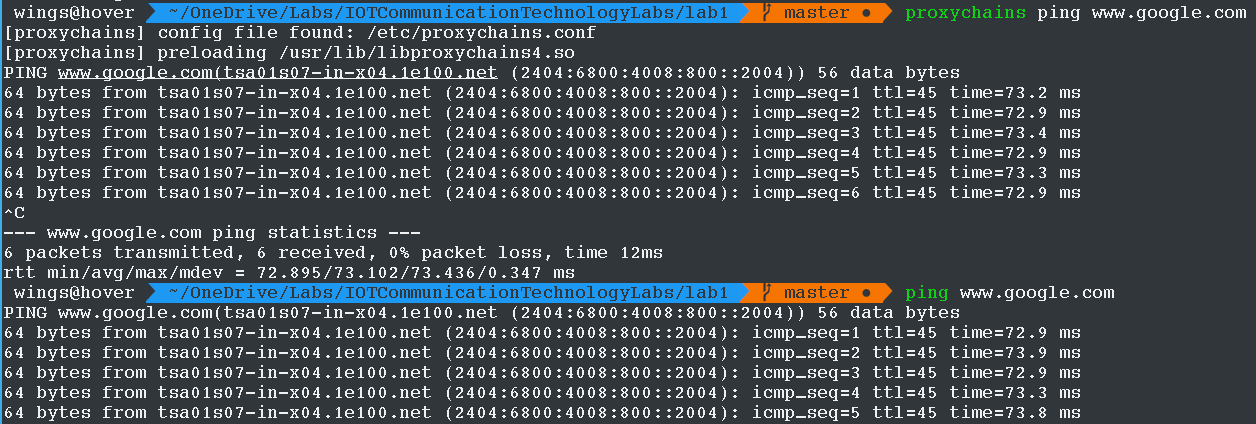


图1-8 proxychains ping

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。

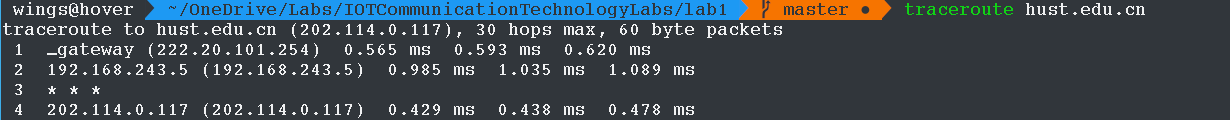


图1-9 traceroute hust.edu.cn

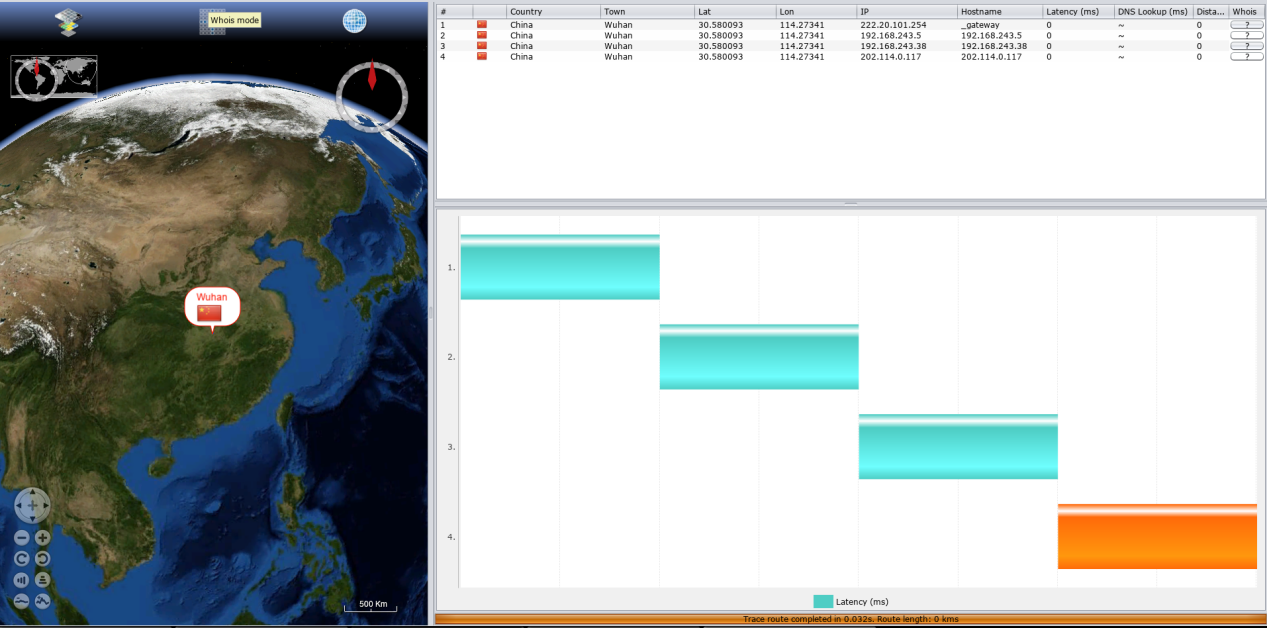


图1-10 visual-traceroute hust.edu.cn

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。

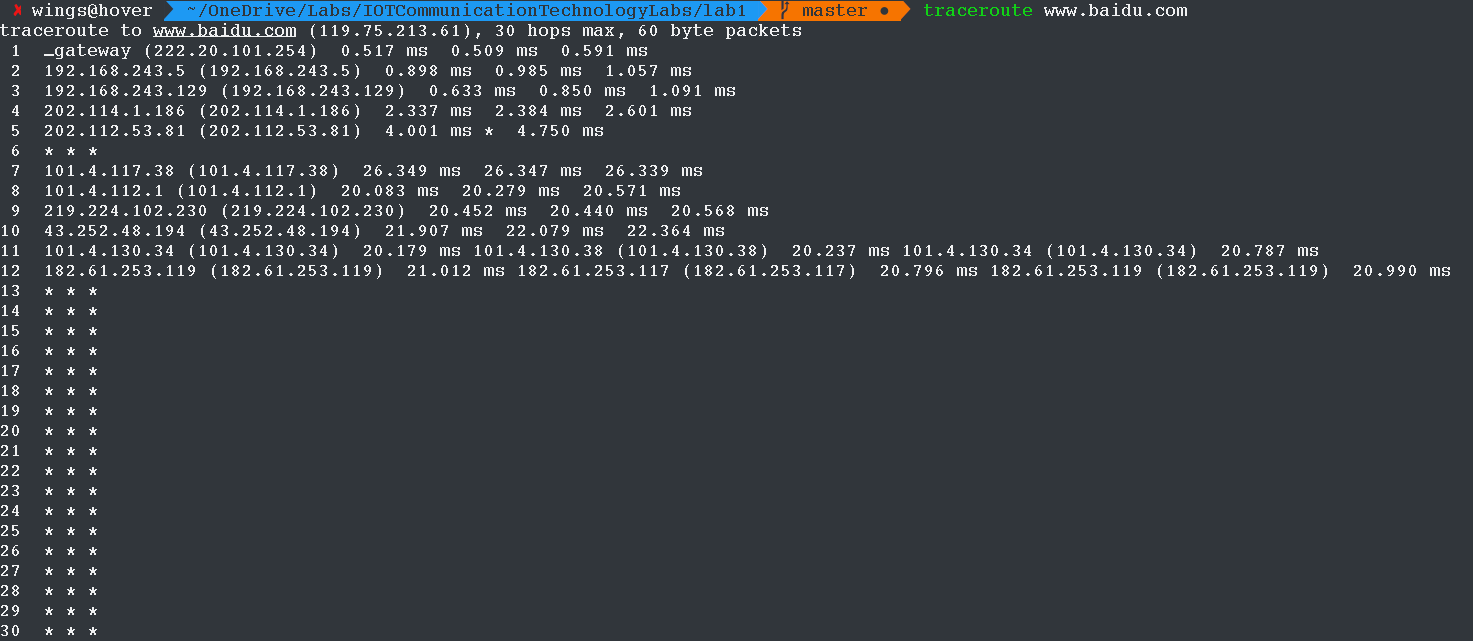


图1-11 traceroute www.baidu.com

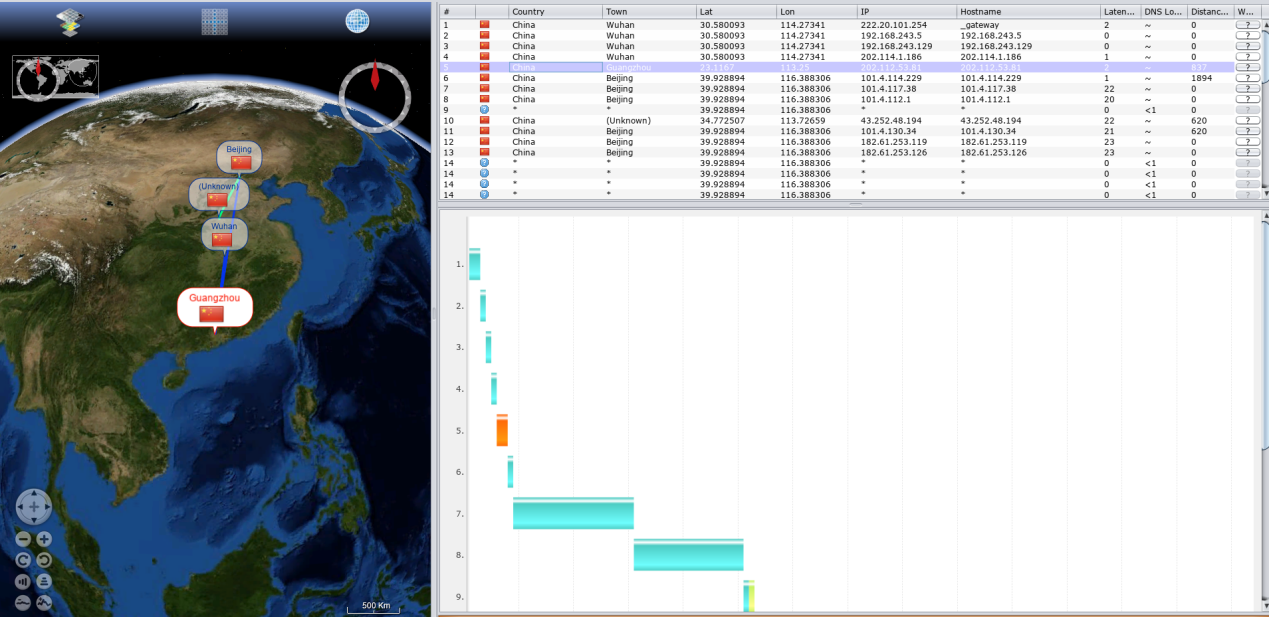


图1-12 visual-traceroute [www.baidu.com](http://www.baidu.com)

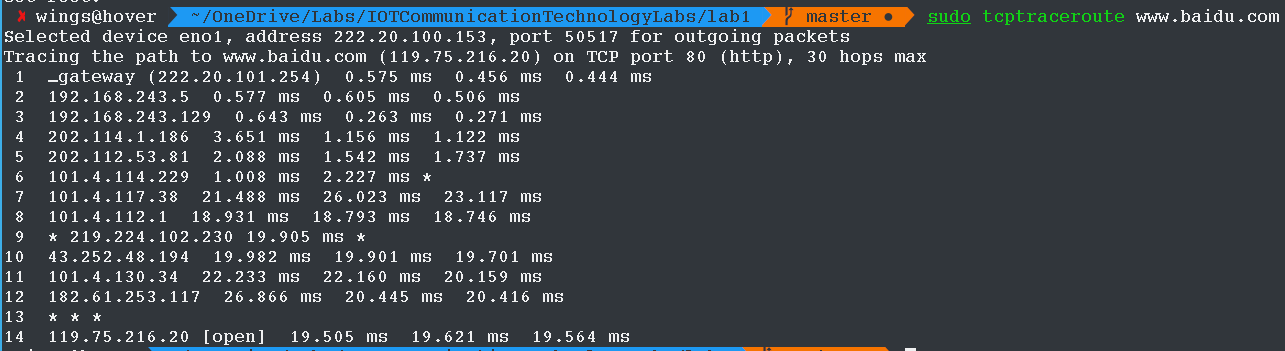


图1-13 tcptraceroute [www.baidu.com](http://www.baidu.com)

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 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：

# 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

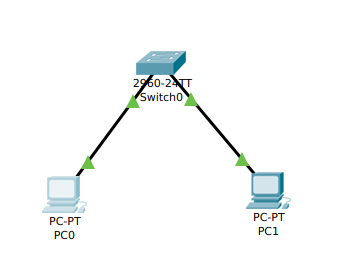


图2-1 环境测试

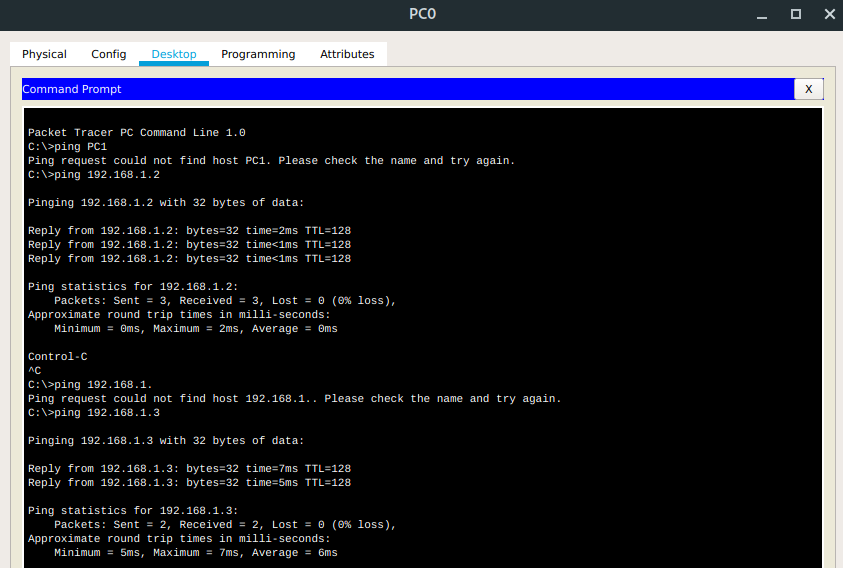


图2-2 PC0 ping PC1

### 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

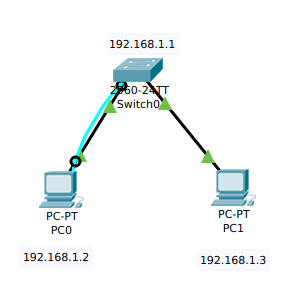


图2-3 交换机控制网络图

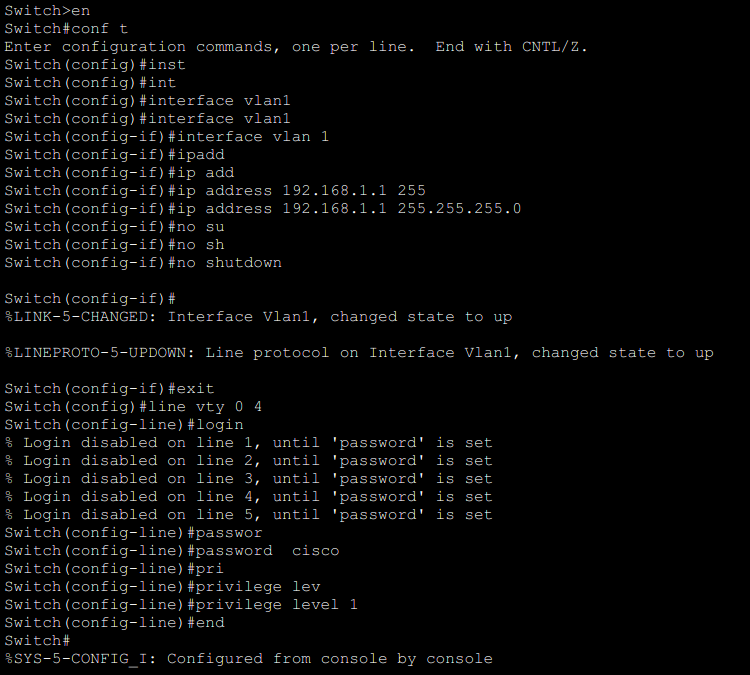


图2-4 交换机配流图

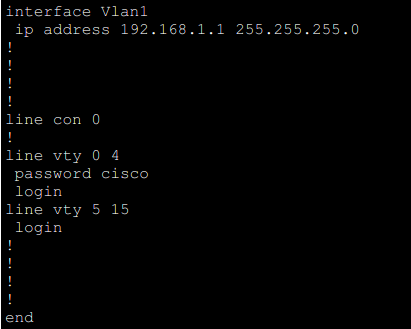


图2-5 交换机配置结果

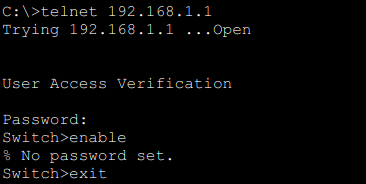


图2-6 交换机配置结果测试

### 2.3.4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

### 2.4.2 设备观察

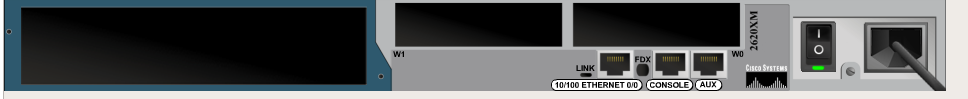


图2-1 Route2620XM模块图

### 2.4.3 命令及报文跟踪

## 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。

报文跟踪中

## 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

# 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

### 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
2. 广播报文
3. 单播报文
4. 未知单播报文

## 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。

## 参考文献

1. CiscoPacketTracer网络实验手册

# 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：

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

## 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IP硬件地址
   2. 使用ARP命令
   3. 分析ARP协议工作过程
   4. 选择两台相连的计算机，清楚ARP表中的所有项目。选择IP地址为222.20.104.243的主机A和222.20.104.248的主机B进行实验，首先在主机A和与B上分别执行“arp -d”命令清楚ARP表中的所有项目。
   5. 在主机A上运行Wireshark程序，执行包俘获操作。
   6. 从主机A向主机B发送Ping包，检查两台主机的ARP表中项目。
   7. 从俘获的分组中找出ARP报文，并分析ARP协议执行的全过程，画出或写出ARP协议报文的交互过程，分析实验结果和现象。

## 4.4 实验结果

### 4.4.1 分析IP协议

### 4.4.1 分析ARP协议

## 4.5 实验中的问题及心得

## 参考文献

1. CiscoPacketTracer网络实验手册

# 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

## 5.5 实验中的问题及心得

## 参考文献

1. CiscoPacketTracer网络实验手册

# 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. 分析TCP序列/应答编号和流量控制
3. 分析应用层内容
4. 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

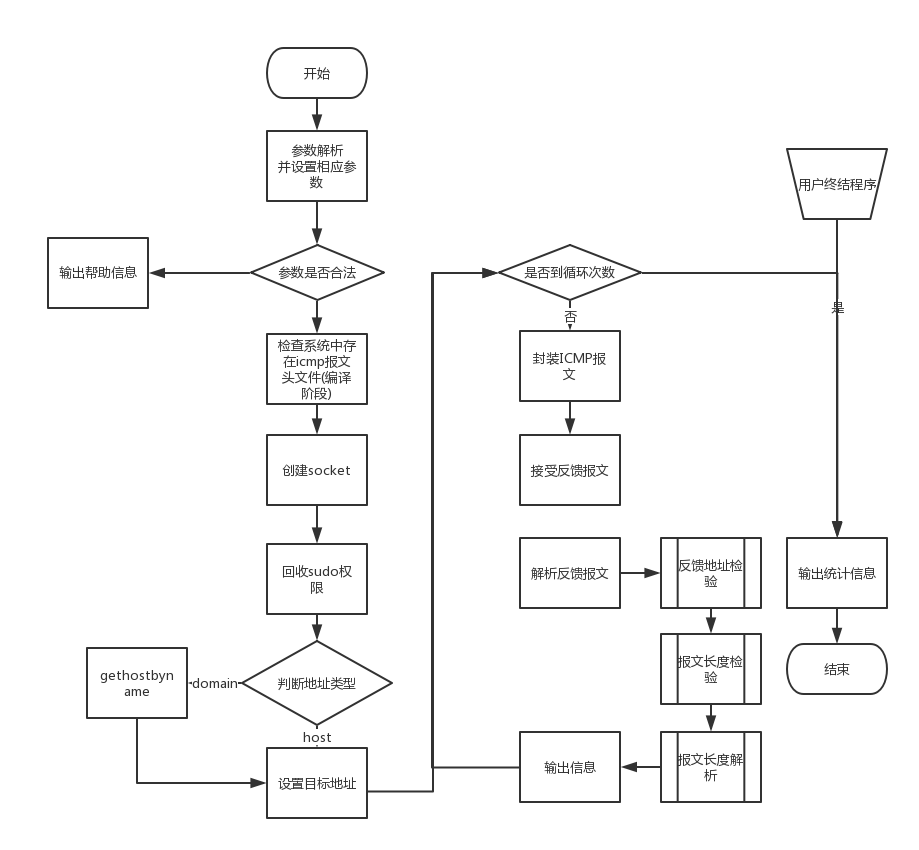


图7-1 myping函数流程图

### 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

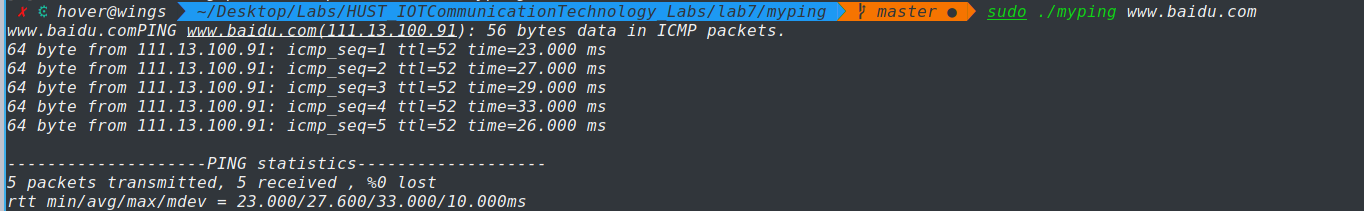


图7-3 myping [www.baidu.com](http://www.baidu.com)

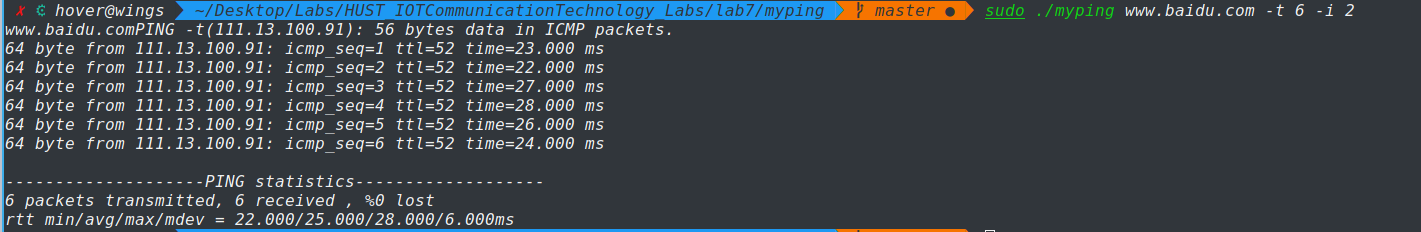


图7-4 myping [www.baidu.com](http://www.baidu.com) 参数解析测试

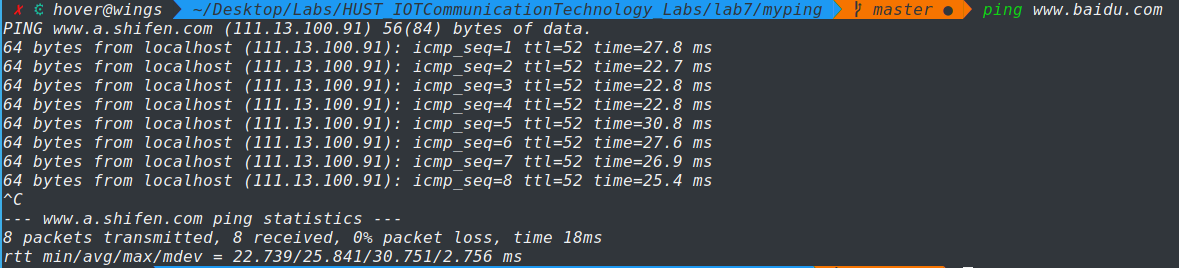


图7-5 ping www.baidu.com对比图

### 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