

**实验指导手册**

**计 算 机 网 络 原 理**



**计算机与软件学院**

**目录**

[实验项目1 局域网设计及仿真 1](#_Toc10363)

[1.1实验目的 1](#_Toc18111)

[1.2实验内容 1](#_Toc18906)

[1.3基本要求 2](#_Toc4740)

[1.4实验步骤 2](#_Toc16263)

[1.4.1 在Packet Tracer构建网络模型 2](#_Toc27820)

[1.4.2 IP地址规划 2](#_Toc11554)

[1.4.3 交换机的工作原理 3](#_Toc9921)

[1.4.4 网络路由 6](#_Toc4303)

[1.4.5思考与解决问题 7](#_Toc30224)

[实验项目2网络应用层协议编程 8](#_Toc9086)

[2.1实验目的 8](#_Toc27992)

[2.2实验内容 8](#_Toc7245)

[2.3基本要求 8](#_Toc8466)

[2.4 实验步骤 8](#_Toc30792)

[2.4.1 TCP编程 8](#_Toc13001)

[2.4.1 UDP编程 11](#_Toc12493)

[实验项目3 网络协议数据捕获及分析 13](#_Toc9306)

[3.1实验目的 13](#_Toc24154)

[3.2实验内容 13](#_Toc6679)

[3.3基本要求 13](#_Toc29334)

[3.4实验步骤 13](#_Toc11043)

[3.4.1 在Packet Tracer构建网络模型 13](#_Toc19408)

[3.4.2 配置路由协议 14](#_Toc6524)

[3.4.3 分析与解决问题 14](#_Toc13644)

[3.4.4使用Wireshark捕获数据包并进行分析 16](#_Toc22417)

**实验报告书写请登录：“http://172.31.36.37:8084/#/login”**

# 实验项目1 局域网设计及仿真

## 1.1实验目的

能够在Packet Tracer模拟软件中进行网络系统模拟仿真。

理解交换机的工作原理，掌握局域网的规划构建的基本方法。

掌握IP地址的结构(网络部分与主机部分的区分)及子网掩码的含义。

理解路由表的作用、基本结构以及路由器转发分组的机制。

## 1.2实验内容

利用网络仿真软件（如Packet Tracer等）进行网络规划设计，包括设计网络拓扑结构、网络设备选择、参数配置、数据分析等。具体实验内容如下：

（1）利用Packet Tracer软件模拟一个局域网，需求为：给定一个24位前缀长度的网络（如192.168.1.0/24），对其进行子网划分，要求能够保证局域网1和局域网2 至少120台机器。另外整个网络通过ISP路由器连接Internet网络（表示有多个网络）如图1-1所示。

（2）合理规划各网络及各设备的IP地址。

（3）验证交换机的工作原理。

（4）给出路由器R1和R2的路由表。

（5）设置路由表，使得局域网1、局域网2能相互连通并能正常访问DNS-serv和Internet网络资源。

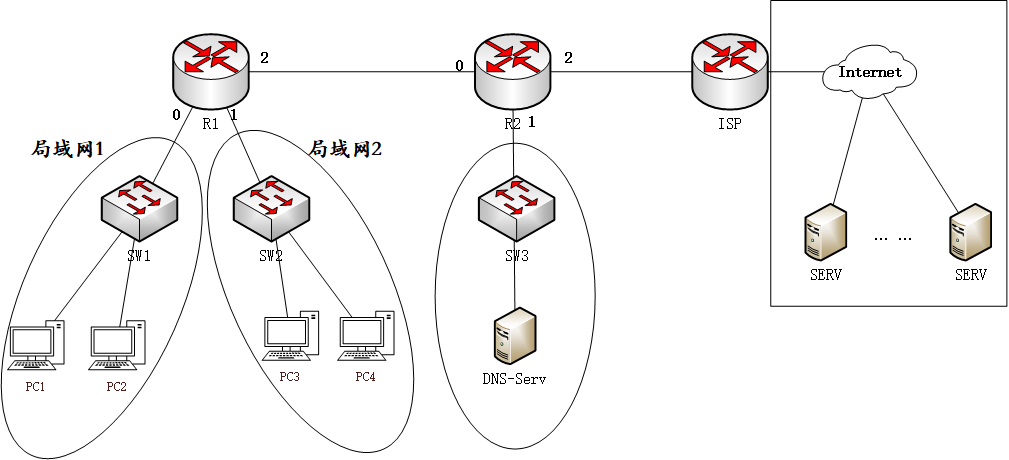


图 1-1

通过本实验使学生学会使用网络仿真软件的一般方法，理解交换机的工作原理，掌握局域网的规划构建的基本方法，能够借助网络仿真软件进行计算机网络系统模拟仿真，满足网络工程项目建设的需要。

## 1.3基本要求

（1）课前准备：仔细阅读实验指导书，详细规划实验过程和步骤，设计实验数据记录表，对相关知识做好预习和准备。

（2）实验过程：按照实验指导书要求，完成实验内容，记录好实验过程、关键数据和实验结果。

（3）实验报告：根据实验情况及时撰写实验报告，实验报告应包含实验目的、实验内容、实验步骤、实验数据及分析，以及实验总结等内容。

## 1.4实验步骤

### 1.4.1 在Packet Tracer构建网络模型

在Packet Tracer软件中构建如图1-2所示拓扑结构图。

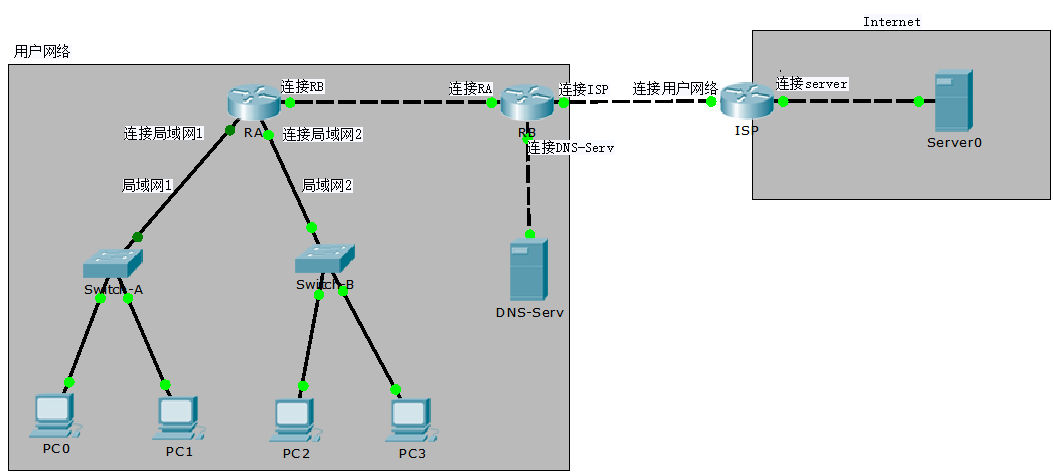


图 1-2 实验拓扑结构图

### 1.4.2 IP地址规划

合理规划各网络及各设备的IP地址，完成表1-1 中各主机及设备接口的信息。。

表1-1 每一台PC 配置IP 地址以及掩码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 设备或接口 | 连接的设备及端口号（例如：RA fa0/0） | IP地址 | 子网掩码 | MAC地址 |
| PC0 |  |  |  |  |
| PC1 |  |  |  |  |
| PC2 |  |  |  |  |
| PC3 |  |  |  |  |
| DNS-Serv |  |  |  |  |
| RA（连接局域网1）的接口 |  |  |  |  |
| RA（连接局域网2）的接口 |  |  |  |  |
| RA（连接RB）的接口 |  |  |  |  |
| RB（连接RA）的接口 |  |  |  |  |
| RB（连接DNS\_Serv）的接口 |  |  |  |  |
| RB（连接ISP）的接口 |  |  |  |  |
| ISP（连接用户网络）的接口 |  |  |  |  |
| ISP（连接server）的接口 |  |  |  |  |

### 1.4.3 交换机的工作原理

测试网络的连通性及交换机MAC地址映射表的变换情况.

切换到模拟器工作模式，使用查看工具( )查看交换机的MAC 表：此时MAC 表是空的，没有任何表项，如图1-3 所示：

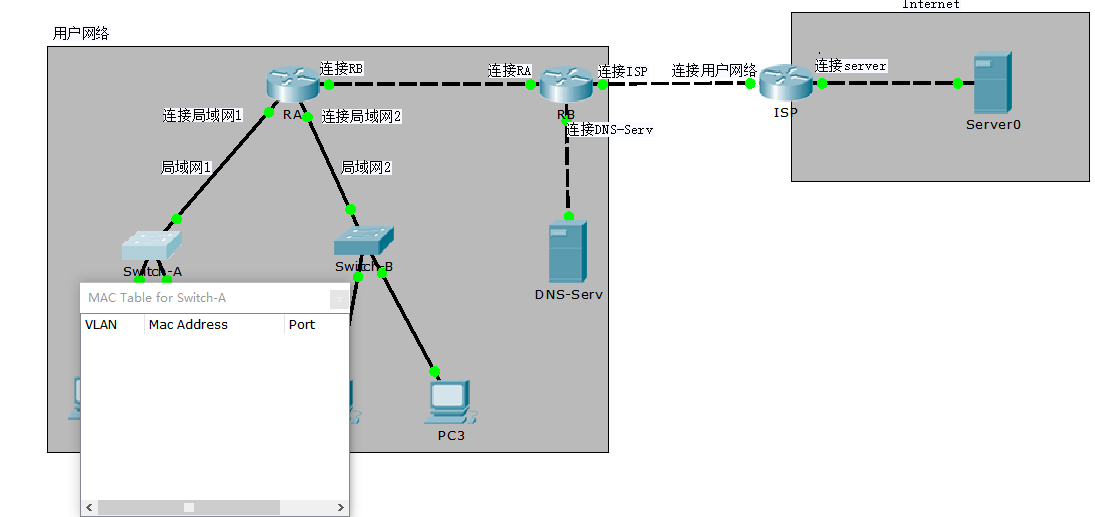


图1-3 初始化时交换机的MAC 地址表

**发送报文并观察MAC 表变化**

按照图1-4选择过滤，仅仅允许ARP、ICMP 协议报文。

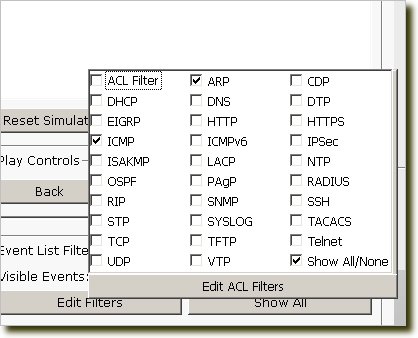


图1-4 设置过滤条件仅允许ICMP、ARP 协议报文

从工具栏中选择“Add Simple PDU”工具( )，然后依次点击PC0、PC1，表示PC0 ping PC1（也可以切换到模拟器/simulation模式下，然后打开PC机的命令行界面，输入ping 目标ip地址），但是此时PC0 还没有PC1 的ARP 表项。ARP 协议在ICMP 报文之前发送，用于学习目的IP地址对应的MAC 地址，只有学习到目的IP 地址对应的MAC 地址之后，协议栈才能正确的在IP 报文之前封装以太网帧头。所以此时PC0 首先发送一个ARP Request 报文。点击“Capture / Forward”按钮，可以一步一步观察ping 命令执行过程。

PC0 发送一个ARP Request 报文给交换机，交换机接收到PC0 的ARP Request 报文之后，将依据“基于接收帧的源MAC 地址学习”原则，添加一条MAC 表项。因为该帧是广播帧，交换机做泛洪处理，路由器RA丢弃该广播帧，如图1-5所示，同时填写表1-2连通性测试表。

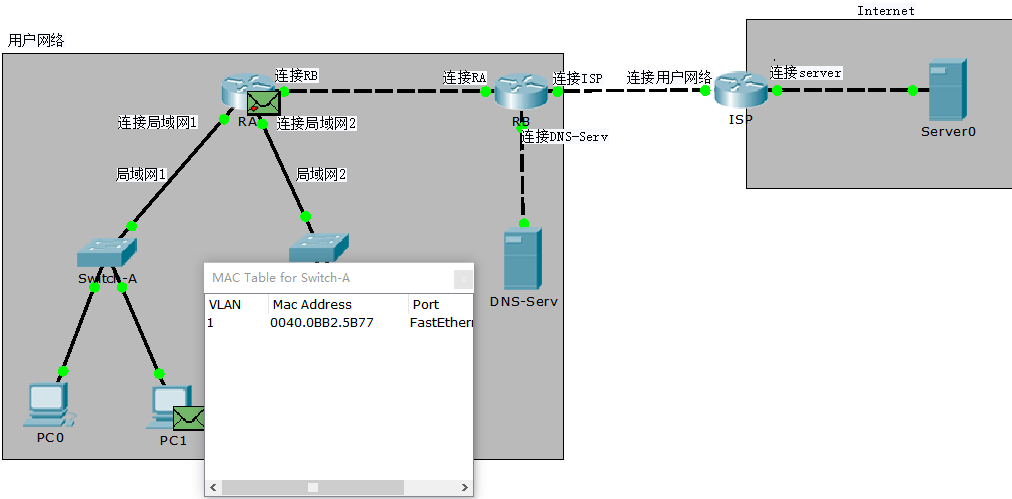


图1-5 PC0 的ARP Request 报文到达交换机，交换机学习并建立MAC 表项，

表 1-2 交换机工作原理及连通性测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **源主机<-->目的主机**  <-->:表示源主机和目的主机之间的通信  -->:表示源主机向目的主机发送数据  <--：表示目的主机向原主机响应（返回）数据 | | **交换机MAC地址映射表变化情况** | **交换机收到数据帧的类型（单播、广播）及转发方式** |
| PC0<-->PC1 | PC0-->PC1 |  |  |

PC1 回复ARP Reply 报文，交换机接收到该帧之后，将学习到PC1 的MAC 地址，并添加到MAC 表，如图1-6所示，**完成表1-3**。

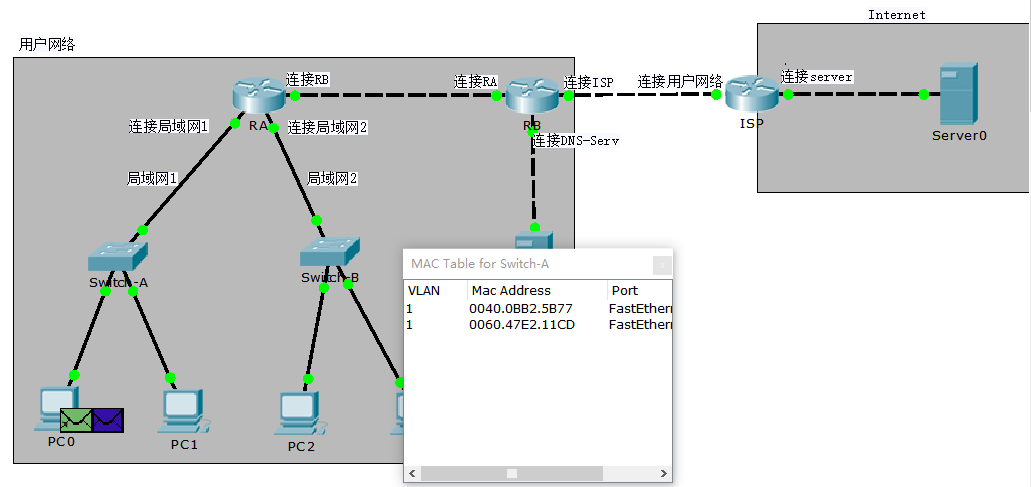


图1-6 交换机接收到PC1 的ARP Reply，并学习到PC1 的MAC 地址

表 1-3 交换机工作原理及连通性测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **源主机<-->目的主机**  <-->:源主机和目的主机之间的通信  -->:源主机向目的主机发送数据  <--：目的主机向原主机响应（返回）数据 | | **交换机MAC地址映射表变化情况** | **交换机收到数据帧的类型（单播、广播）及转发方式** |
| PC0<-->PC1 | PC0<--PC1 |  |  |

完成表1-2、1-3的测试任务后查看交换机的MAC 表：此时MAC 表已有pc0、pc1的MAC地址和交换机对应的端口信息。此时按表1-4再次测试主机之间的连通性，并完成表1-4的内容。

表 1-4 交换机工作原理及连通性测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **源主机<-->目的主机**  <-->:源主机和目的主机之间的通信  -->:源主机向目的主机发送数据  <--：目的主机向原主机响应（返回）数据 | | **交换机MAC地址映射表变化情况** | **交换机收到数据帧的类型（单播、广播）及转发方式** |
| PC0<-->PC1 | PC0-->PC1 |  |  |
| PC0<--PC1 |  |  |

**问题：**通过实验结果可以验证交换机的工作原理是？

### 1.4.4 网络路由

（1）路由表（Routing Table）：存储在路由器的内存中，用于指示路由器如何将IP数据包转发至正确目的地的信息表。路由表的基本结构如表1-5所示。

表1-5 路由表基本结构

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |

* destination ：目的地址（网络、主机）
* mask（掩码）：目的网络中的子网掩码
* nexthop（地址）：下一跳路由器的IP地址
* interface（接口）：该路由的发送接口名

（2）路由表的形成

* 直连网段

配置IP地址，端口UP状态，形成直连路由。

**直连路由：**路由器接口所连接的子网的路由方式称为直连路由。

直连路由是由链路层协议发现的，该路径信息不需要网络管理员维护，也不需要路由器通过某种算法进行计算获得，只要该接口处于活动状态(Active)，路由器就会把通向该网段的路由信息填写到路由表中去，直连路由无法使路由器获取与其不直接相连的路由信息。

* 非直连网段

需要静态路由或动态路由，将网段添加到路由表中。

**静态路由：**由管理员手工配置、维护路由表。

（3）根据需求分析，写出路由器RA、RB、ISP的路由表，使得使得局域网1、局域网2能相互连通并能正常访问DNS-serv和Internet网络资源。

表1-6 路由器RA的路由表

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

表1-7 路由器RB的路由表

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

表1-8 路由器ISP的路由表

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 1.4.5思考与解决问题

（1）按表1-9所示内容进行测试并完成表中内容。

首先清除测试主机的arp缓存表。

arp -a [inet\_addr] 显示地址映射表项，[ ]为可选项。

arp -d [inet\_addr] 删除由inet\_addr所指定的表项。

表1-9 ARP缓存表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **源主机<-->目的主机**  <-->:源主机和目的主机之间的通信  -->:源主机向目的主机发送数据  <--：目的主机向原主机响应（返回）数据 | | 发送主机发送数据之前的ARP表 | 接收主机接收数据之前的ARP表 | 封装数据包时，原IP、MAC地址 | 封装数据包时，目的IP、MAC地址 | 接收/发送端收到ARP请求/响应后ARP表的变化情况 |
| PC0<-->PC1 | PC0-->PC1 |  |  |  |  |  |
| PC0<--PC1 |  |  |  |  |  |
| PC0<-->PC2 | PC0-->PC2 |  |  |  |  |  |
| PC0<--PC2 |  |  |  |  |  |

（2）当PC0 的ARP 表中没有PC1 的IP/MAC 表项时，PC0 ping PC1 时将首先发送什么报文，如果PC0 的ARP 表中有PC1 的IP/MAC 表项，情况又会怎么样？

（3）交换机对ARP 请求报文如何处理？

**实验项目2网络应用层协议编程**

## 2.1实验目的

理解Socket的作业及Socket的通信机制

熟悉Socket的类型及使用场景

掌握Socket的工作流程

能够熟练使用常见编程语言完成Socket网络应用程序的编写和调试

## 2.2实验内容

（1）运用特定的编程开发语言（C、Python、Java等），完成具体的应用层网络应用的开发；

（2）使用TCP或UDP作为传输层的联网协议进行设计，并能够对比两种传输层协议在网络通信及协议开发中的区别；

（3）网络应用层程序能够真正实现服务器的角色，并支持多个客户端同时接入，客户端之间通信互不干扰；

（4）通过最终编译的可执行程序验证最初的实验目标，并总结相关的心得体会。

## 2.3基本要求

（1）课前准备：仔细阅读实验指导书，详细规划实验过程和步骤，设计实验数据记录表，对相关知识做好预习和准备。

（2）实验过程：按照实验指导书要求，完成实验内容，记录好实验过程、关键数据和实验结果。

（3）实验报告：根据实验情况及时撰写实验报告，实验报告应包含实验目的、实验内容、实验步骤、实验数据及分析，以及实验总结等内容。

## 2.4 实验步骤

### 2.4.1 TCP编程

**（1）建立一个TCP服务器连接需要6个步骤（以Python为例）**

创建socket对象。调用socket构造函数：

socket=socket.socket(familly,type)

family的值可以是AF\_UNIX(Unix域，用于同一台机器上的进程间通讯)，也可以是AF\_INET（对于IPV4协议的TCP和 UDP），至于type参数，SOCK\_STREAM（流套接字）或者 SOCK\_DGRAM（数据报文套接字）、SOCK\_RAW（raw套接字），TCP使用SOCK\_STREAM参数。

将socket绑定（指派）到指定地址上，socket.bind(address)

address必须是一个双元素元组,((host,port)),主机名或者ip地址+端口号。如果端口号正在被使用或者保留，或者主机名或ip地址错误，则引发socke.error异常。

绑定后，必须准备好套接字，以便接受连接请求。

socket.listen(backlog)

backlog指定了最多连接数，至少为1，接到连接请求后，这些请求必须排队，如果队列已满，则拒绝请求。

服务器套接字通过socket的accept方法等待客户请求一个连接：

connection,address=socket.accept()

调用accept方法时，socket会进入'waiting'（或阻塞）状态。客户请求连接时，方法建立连接并返回服务器。accept方法返回一个含有两个元素的元组，形如(connection,address)。第一个元素（connection）是新的socket对象，服务器通过它与客户通信；第二个元素（address）是客户的internet地址。

处理阶段，服务器和客户通过send和recv方法通信（传输数据）。

服务器调用send，并采用字符串形式向客户发送信息。send方法返回已发送的字符个数。服务器使用recv方法从客户接受信息。调用recv时，必须指定一个整数来控制本次调用所接受的最大数据量。recv方法在接受数据时会进入'blocket'状态，最后返回一个字符串，用它来表示收到的数据。如果发送的量超过recv所允许，数据会被截断。多余的数据将缓冲于接受端。以后调用recv时，多余的数据会从缓冲区删除。

传输结束，服务器调用socket的close方法以关闭连接。

**（2）建立一个TCP客户端连接则需要4个步骤**

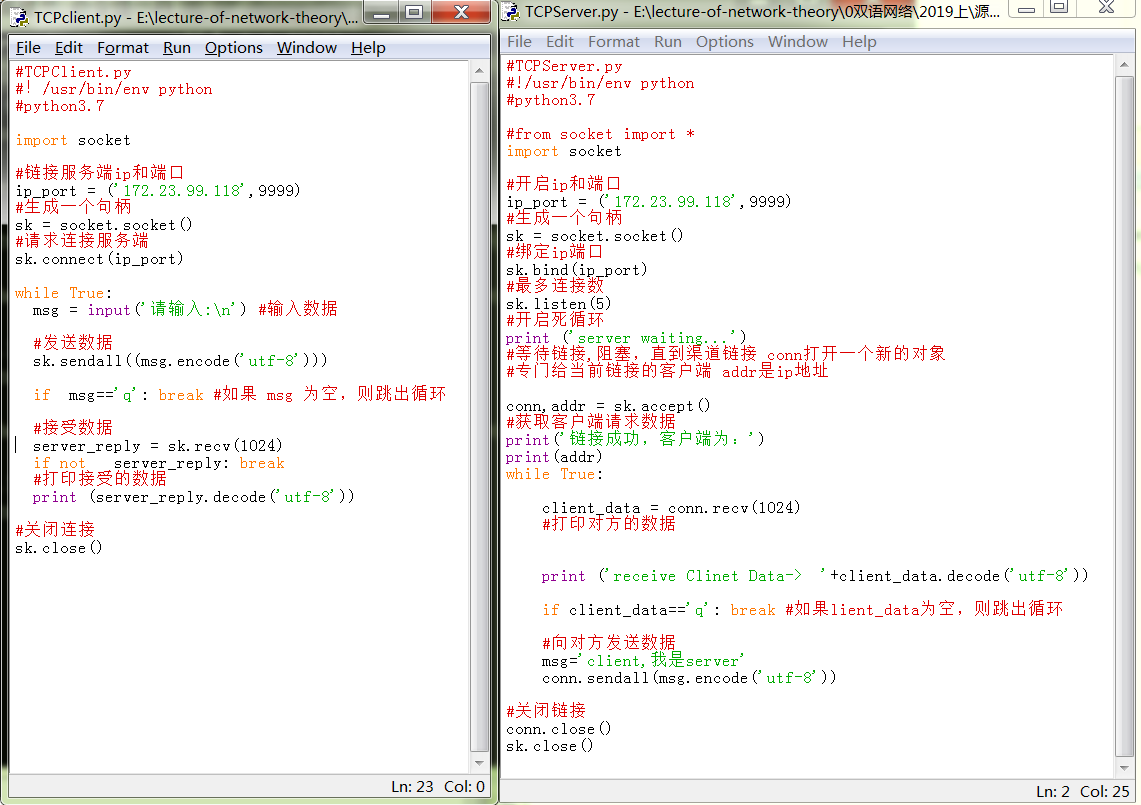
创建一个socket以连接服务器 socket=socket.socket(family,type)

使用socket的connect方法连接服务器 socket.connect((host,port))

客户和服务器通过send和recv方法通信。

结束后，客户通过调用socket的close方法来关闭连接。

**（3）建立TCP服务器与TCP客户端程序**



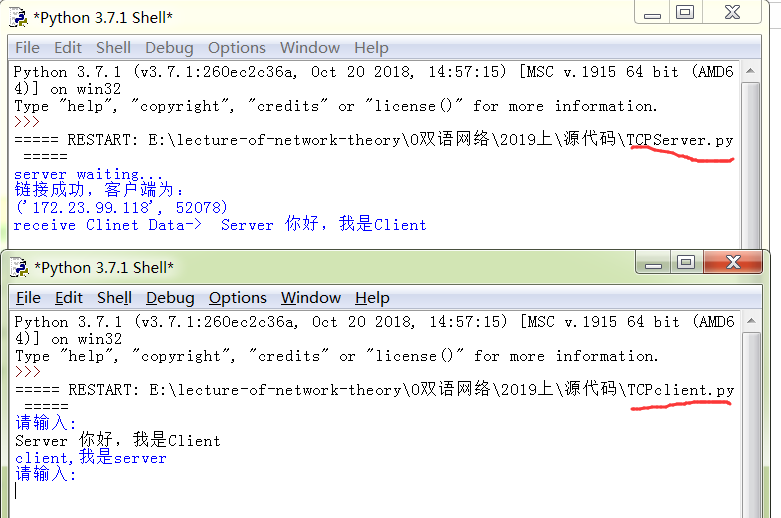
**服务器程序：**

创建套接字，绑定套接字，开始监听，就是一个while循环一直监听有没有消息连接。

**客户端程序：**

创建套接字，连接，等待输入，发送。

同一台电脑结果在两个shell中演示。



### 2.4.1 UDP编程

由于UDP是无连接的，与TCP的区别在服务器端表现为，无需监听函数listen()对客户端的连接进行监听； 在客户端的表现为客户端的socket无需与服务器建立连接就能够进行数据的发送与接收，即无需connect()函数。

**（1）建立一个UDP服务器连接需要4个步骤（以Python为例）**

创建socket对象。调用socket构造函数：

socket=socket.socket(familly,type)

family的值可以是AF\_UNIX(Unix域，用于同一台机器上的进程间通讯)，也可以是AF\_INET（对于IPV4协议的TCP和 UDP），至于type参数，SOCK\_STREAM（流套接字）或者 SOCK\_DGRAM（数据报文套接字）,SOCK\_RAW（raw套接字），UDP 使用SOCK\_DGRAM参数。

将socket绑定（指派）到指定地址上，socket.bind(address)

address必须是一个双元素元组,((host,port)),主机名或者ip地址+端口号。如果端口号正在被使用或者保留，或者主机名或ip地址错误，则引发socke.error异常。

处理阶段，服务器和客户通过sendto()和recvfrom()方法通信（传输数据）。

传输结束，服务器调用socket的close方法以关闭连接。

**建立UDP服务器程序：**

创建套接字，开始监听，就是一个while循环一直监听有没有消息连接。

创建套接字，while循环一直监听有没有消息连接。

**（2）建立一个UDP客户端连接则需要4个步骤**

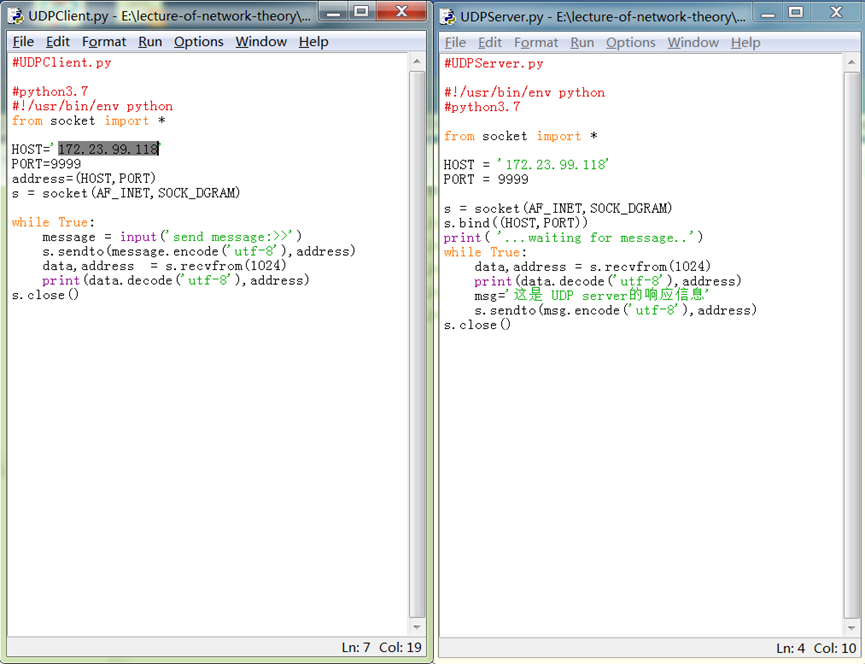
创建一个socket以连接服务器 socket=socket.socket(family,type)

使用socket的connect方法连接服务器 socket.connect((host,port))

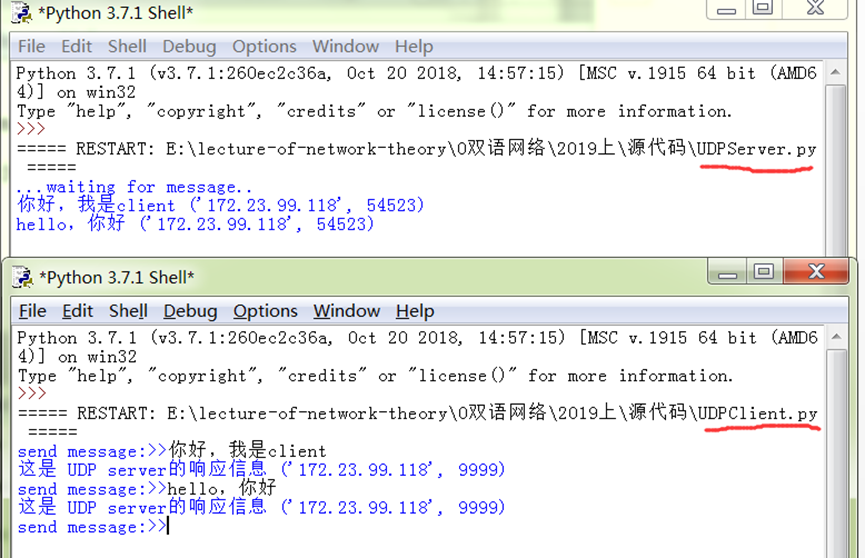
客户和服务器通过sendto()和recvfrom()方法通信。

结束后，客户通过调用socket的close方法来关闭连接。

**（3）编写UDP服务器与客户端程序**



**（4）运行效果**



**实验项目3 网络协议数据捕获及分析**

## 3.1实验目的

掌握以太网帧的格式及各字段的含义

掌握IP包的组成格式及各字段的含义

掌握UDP报文和TCP报文的格式及各字段的含义

理解HTTP协议、DNS协议工作机制

熟悉用户数据在网络各层之间的封装关系

## 3.2实验内容

利用网络仿真软件（如Packet Tracer等）进行网络规划设计，包括设计网络拓扑结构、网络设备选择、参数配置、数据分析等。具体实验内容如下：

（1）利用Packet Tracer仿真软件搭建一个包含DNS、WWW等服务的互联网。

（2）配置测试各种服务并测试连通性。

（3）捕获IP数据报、TCP报文段、HTTP报文、DNS报文，识别和分析IP协议、TCP协议、UDP协议。

（4）使用Wireshark捕获实验项目二网络应用层协议编程两种通讯的数据包，并进行分析，以深入了解TCP、UDP协议的数据传输机制。

通过本实验使学生学会使用网络仿真软件的一般方法，熟悉用户数据在整体网络中的活动和封装情况，能够运用网络协议分析工具获取网络流量，分析网络流量和协议结构解，理解各层协议的功能、作用及相互之间的关系，理解网络协议工作原理。

## 3.3基本要求

（1）课前准备：仔细阅读实验指导书，详细规划实验过程和步骤，设计实验数据记录表，对相关知识做好预习和准备。

（2）实验过程：按照实验指导书要求，完成实验内容，记录好实验过程、关键数据和实验结果。

（3）实验报告：根据实验情况及时撰写实验报告，实验报告应包含实验目的、实验内容、实验步骤、实验数据及分析，以及实验总结等内容。

## 3.4实验步骤

### 3.4.1 在Packet Tracer构建网络模型

建立如图3-1所示网络结构，根据图中标注的主机A、主机B、DNS服务器、WEB服务器和路由器接口（接口根据自己建立的网络拓扑结构的实际情况指定）的IP地址和子网掩码，正确配置各设备/接口的相关参数，完成表3-1内容，MAC地址请根据自己建立的网络拓扑图中每台PC实际的MAC地址如实填写（注意：MAC地址每个人的都不一样）。

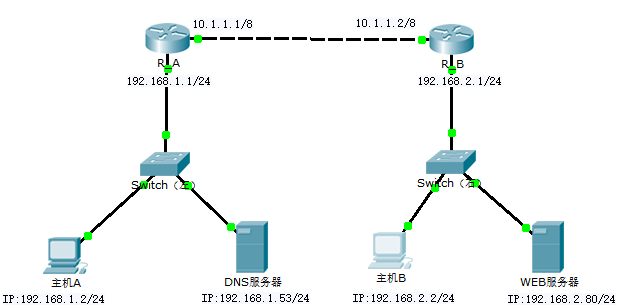


图3-1 网络拓扑结构

表3-1 主机参数表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主机 | IP地址 | 子网掩码 | MAC地址 | 默认网关 | DNS服务器 |
| 主机A | 192.168.1.2 | 255.255.255.0 |  |  |  |
| 主机B | 192.168.2.2 | 255.255.255.0 |  |  |  |

### 3.4.2 配置路由协议

配置路由协议使整个网络连通，并完成路由器A和B的路由表表3-2和表3-3的内容。

表 3-2 路由器A的路由表

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

表 3-3 路由器B的路由表

|  |  |  |  |
| --- | --- | --- | --- |
| Destination（目的地） | Mask（掩码） | Nexthop（下一跳） | Interface（接口） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.4.3 分析与解决问题

（1）主机A发送一个IP数据报到主机B的过程中，各结点要进行路由选择和转发，写出各结点对数据在网络层和数据链路层的封装情况，完成表3-4内容。

表3-4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据链路层封装 | | 网络层封装 | |
| 数据所在结点 | 目的MAC地址 | 源MAC地址 | 目的IP地址 | 原IP地址 |
| 主机A |  |  |  |  |
| 路由器A |  |  |  |  |
| 路由器B |  |  |  |  |

（2）DNS服务器上建立一条资源记录（A），域名为www.123.cn，IP地址为192.168.2.80。WEB服务器上有默认index.html文档，新建一个文档123.html。

a) 用户在主机A的浏览器的地址栏里输入什么地址可以访问到WEB服务器的主页index.html？

b) 用户在主机A的浏览器的地址栏里输入什么地址可以访问到WEB服务器的主页123.html？

c) 用户在主机A的浏览器的地址栏里输入www.123.cn，观察、记录整个通信过程（切换至simulation模式下）。分析传输过程各阶段的作用（分阶段表述），并写出各阶段源数据在不同结点上的各层的封装情况，如表3-5所示。在主机B的浏览器的地址栏里输入www.123.cn，观察、记录整个通信过程并完成表3-6内容。

表 3-5 主机A浏览器地址栏输入“www.123.cn”的数据封装记录表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据传输方向（源-->目的或源<--目的）及数据所在结点 | 数据链路层封装 | | 网络层封装 | | 传输层 | | | 应用层 |
| 目的MAC地址（需描述是谁的MAC） | 源MAC地址 | 目的IP地址 | 原IP地址 | 何种协议 | 目的端口 | 原端口 | 何种服务 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

表 3-6 主机B浏览器地址栏输入“www.123.cn”的数据封装记录表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据传输方向（源-->目的或源<--目的）及数据所在结点 | 数据链路层封装 | | 网络层封装 | | 传输层 | | | 应用层 |
| 目的MAC地址 | 源MAC地址 | 目的IP地址 | 原IP地址 | 何种协议 | 目的端口 | 原端口 | 何种服务 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

### 3.4.4使用Wireshark捕获数据包并进行分析

**（1）监听工具简介**

开发调试过程中需要使用一些抓包工具，Wireshark这个抓包工具比较强大，具体的网上能查到很多，这里只是简单说下一些常用的命令：

ip.addr==需要查的IP地址 and tcp.port==443 ： 查询地址和端口号匹配的记录

ip and (tcp.port==4437 or tcp.port==5775) ：查询协议和端口号匹配的记录

选中要看的那条记录--右击--Follow TCP Stream 就能看到包里的信息了。如果是明文可以直接看到内容，如果加密了看到的就是乱码。

Wireshark虽然强大，但是无法抓取本地回环数据。也就是说：当你本地测试一个socket程序，发送/接收到本地端口的时候，Wireshark就无能为力了。这个时候需要另外一个工具：RawCap 。

RawCap具体命令如下：进入命令行(cmd),运行代码如下：

C:\ RawCap >RawCap.exe 本地IP dumpfile.pcap

或者

C:\ RawCap RawCap.exe 172.23.99.118 dumpfile.pcap

抓好包后，按Ctrl+C，停止抓包。这个时候会在RawCap的同级目录下显示一个dumpfile.pcap文件。通过wireshark打开，就可以看到本地环回的数据包了。

**（2）TCP通讯捕捉与分析**

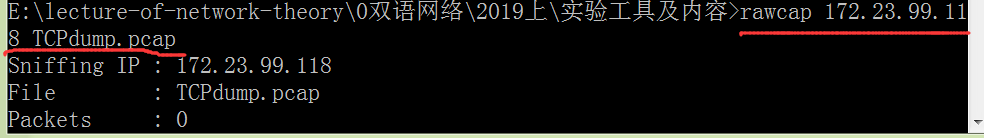
编写、调试好TCP的服务器端与客户端程序。

控制台方式进入RawCap.exe的文件所在路径。

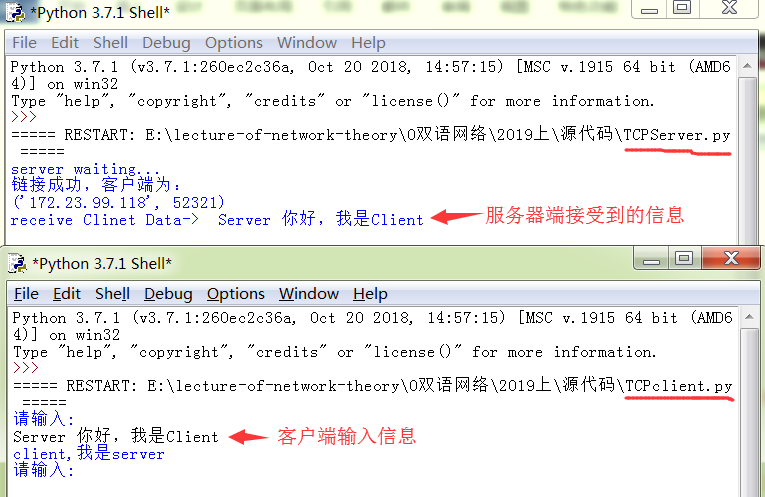
在控制台方式输入命令：

RawCap.exe 本地IP dumpfile.pcap

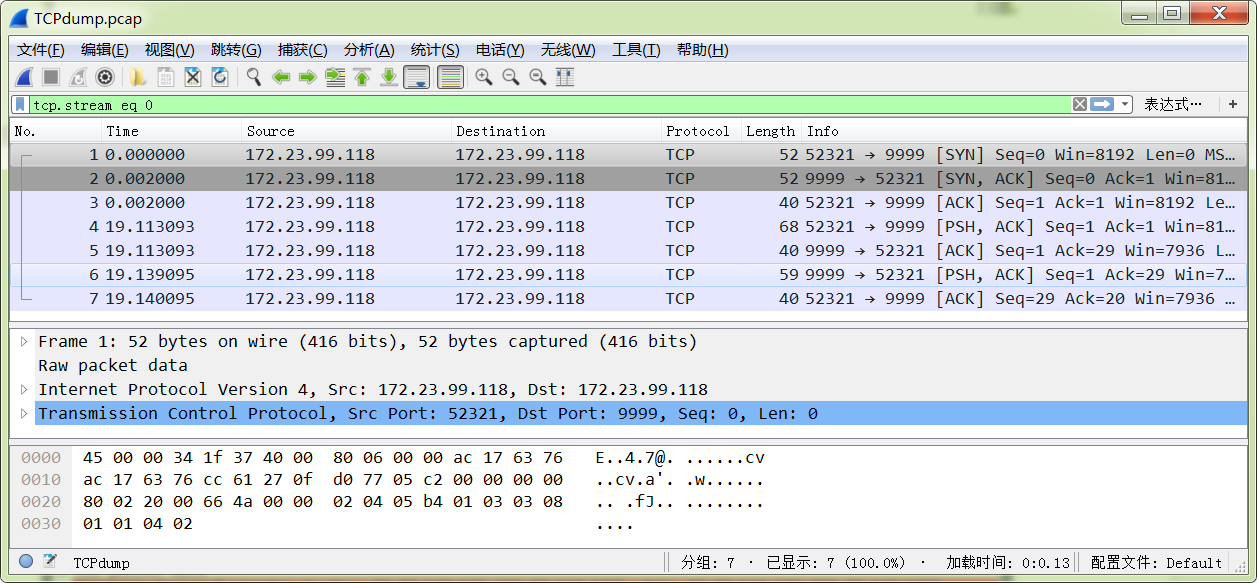
例如：RawCap.exe 172.23.99.118 UDPdump.pcap



在启动RawCap之后，再启动服务器程序，最后启动运行客户机程序，然后进行数据通讯，按这个顺序，RawCap就可以扑捉到client-server的全部通讯数据，包括三次握手。数据通讯如下图：

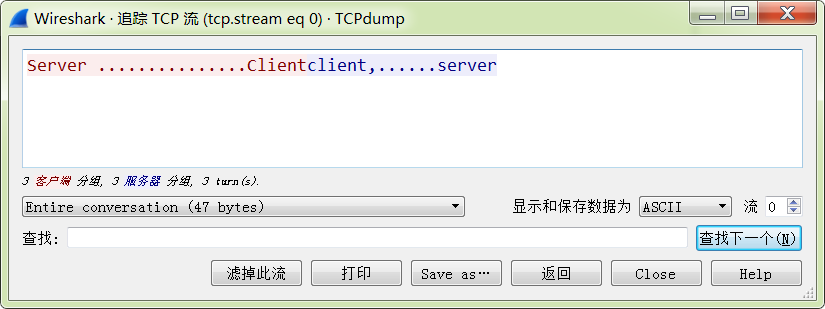


最终用Wireshark打开抓包文件TCPDump.pcap，在Filter栏目中输入TCP,然后点击Apply,如下图所示，显示出客户机与服务器的全部通讯数据。



查看Tcp流中的应用层数据

在Wireshark的Analyze菜单中选择“Follow TCP Stream”功能，就会出现下面的窗口，窗口显示的就是客户机与服务器通讯的内容，内容出现的顺序同他们在网络中出现的顺序一致。从A到B的通信标记为红色，从B到A的通信标记为蓝色。



实验任务

查出通讯三次握手、源端口、目的端口、TTL值、IP版本号、协议类型、IP头长度以及传输的数据内容。

**（3）UDP通讯捕捉与分析**

编写、调试好UDP的服务器端与客户端程序。

注意：程序中的IP地址要设置为本机的IP地址，不要使用回环地址127.0.0.1（数据通讯抓不到）。

控制台方式进入RawCap.exe的文件所在路径。

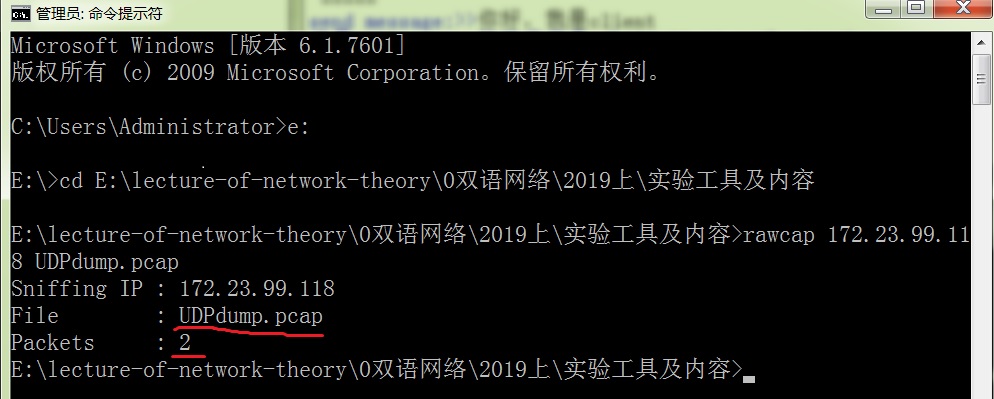
在控制台方式输入命令：

RawCap.exe 本地IP dumpfile.pcap

例如：RawCap.exe 172.23.99.118 UDPdump.pcap

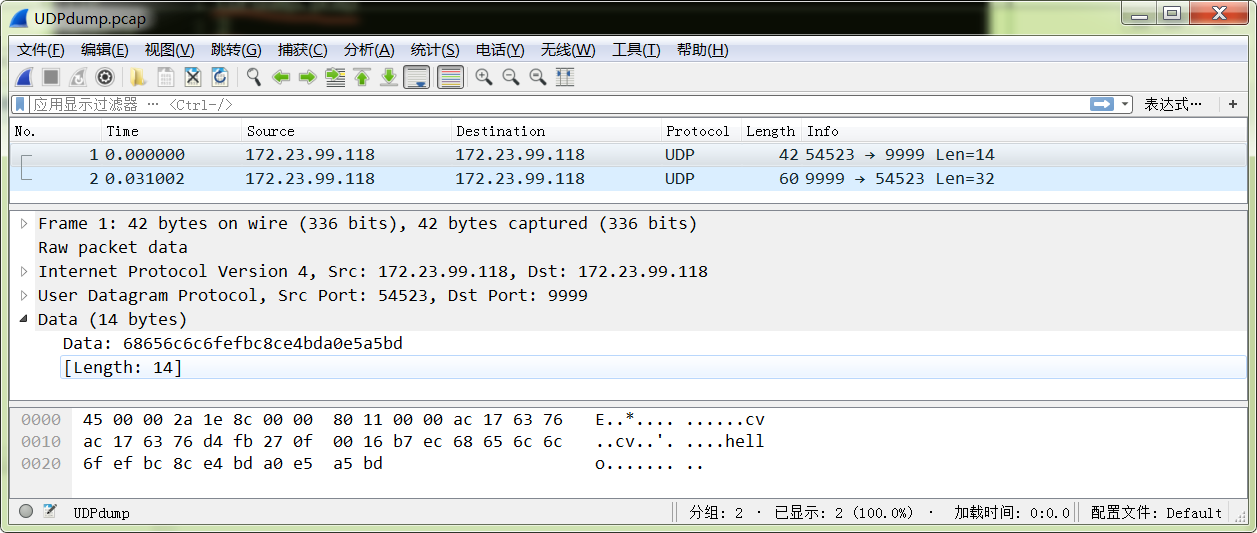
在启动RawCap之后，再启动服务器程序，最后启动运行客户机程序，然后进行数据通讯，按这个顺序，RawCap就可以扑捉到client-server的全部通讯数据。

客户机与服务器通讯结束后，回到控制台，看到RawCap已经捕捉到一些数据包。



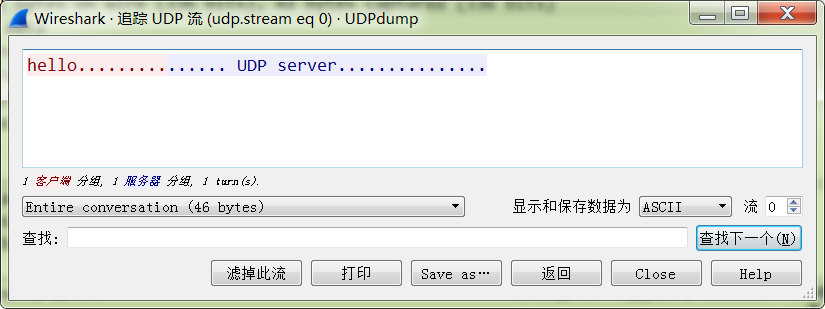
按Ctrl+C，停止抓包。这个时候会在RawCap的同级目录下显示一个UDPdump.pcap文件。

通过wireshark打开，就可以看到本地通讯的数据包了。



查看UDP流中的应用层数据

在Wireshark的Analyze菜单中选择“Follow UDP Stream”功能，就会出现下面的窗口，窗口显示的就是客户机与服务器通讯的内容，内容出现的顺序同他们在网络中出现的顺序一致。从A到B的通信标记为红色，从B到A的通信标记为蓝色。



实验任务：

查出通讯的源端口、目的端口、TTL值、IP版本号、协议类型、IP头长度以及传输的数据内容。