**计算机网络课程**

**设计报告**

1. 年 12 月

目录

[一、网络聊天程序的设计与实现 2](#_Toc27925343)

[1、内容 2](#_Toc27925344)

[2、设计步骤 2](#_Toc27925345)

[3、 调式过程 3](#_Toc27925346)

[4、结果及结果分析 3](#_Toc27925347)

[5、心得体会 4](#_Toc27925348)

[二、Tracert 与 Ping 程序设计与实现 4](#_Toc27925349)

[1、内容 4](#_Toc27925350)

[2、设计步骤 4](#_Toc27925351)

[3、 调式过程 4](#_Toc27925352)

[4、结果及结果分析 5](#_Toc27925353)

[5、心得体会 5](#_Toc27925354)

[三、基于 IP 多播的网络会议程序 6](#_Toc27925355)

[1、内容 6](#_Toc27925356)

[2、设计步骤 6](#_Toc27925357)

[3、 调式过程 6](#_Toc27925358)

[4、结果及结果分析 7](#_Toc27925359)

[5、心得体会 7](#_Toc27925360)

[四、网络嗅探器的设计与实现 7](#_Toc27925361)

[1、内容 7](#_Toc27925362)

[2、设计步骤 7](#_Toc27925363)

[3、 调式过程 8](#_Toc27925364)

[4、结果及结果分析 8](#_Toc27925365)

[5、心得体会 8](#_Toc27925366)

[五、电子邮件客户端程序设计与实现 9](#_Toc27925367)

[1、内容 9](#_Toc27925368)

[2、设计步骤 9](#_Toc27925369)

[3、 调式过程 10](#_Toc27925370)

[4、结果及结果分析 10](#_Toc27925371)

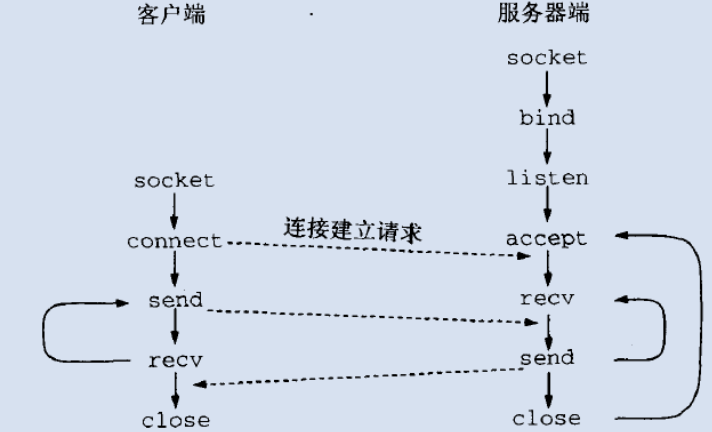
[5、心得体会 11](#_Toc27925372)

# 一、网络聊天程序的设计与实现

## 1、内容

了解 Socket 通信的原理，在此基础上编写一个聊天程序

## 2、设计步骤



服务器端编程的步骤：

1、加载套接字库，创建套接字(WSAStartup()/socket())；  
2、绑定套接字到一个 IP 地址和一个端口上(bind())；  
3、将套接字设置为监听模式等待连接请求(listen())；  
4、请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字(accept())；  
5、用返回的套接字和客户端进行通信(send()/recv())；  
6、返回，等待另一连接请求；  
7、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())

客户端编程的步骤：

1、加载套接字库，创建套接字(WSAStartup()/socket())；  
2、向服务器发出连接请求(connect())；  
3、和服务器端进行通信(send()/recv())；  
4、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

## 3、 调式过程

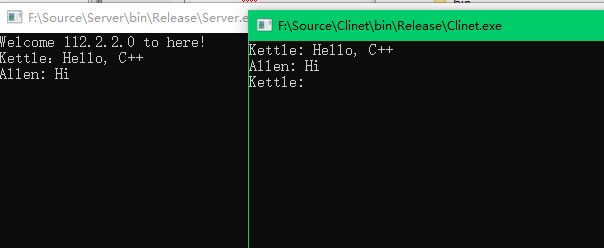
1）先启动服务端server,就是先启动启动监听

2）启动客户端程序运行

3）这样就可以进行将客户端向服务器端进行发送数据

4）这个过程只是实现了客户端与服务端的收发信息，而且只是发一条后服务端会一条，因为这里并没有使用线程实现，也没有实现信息转发

## 4、结果及结果分析



客户端发送一条Hello,C++的信息，然后服务器将收到这条信息，这时服务器必须先输入一条信息返回才能继续收发，上面发回去的是Hi。

* 如果要实现 客户端和服务端进行真正的聊天，就需要把接收数据的部分放入线程，让其不断的运行使的可以随时收到数据。如果需要更好的实现聊天还是使用UDP发送，因为发送的数据并不会太多，当然了里面如果涉及文件传输时，这个文件传输还是需要使用TCP进行发送等待
* 如果要实现多个客户端进行聊天，那么可以通过服务端进行转发，并且在发送的数据中加入不同客户端的标志，如用户名或账号等，然后根据这些标志进行转发
* 如果要实现离线的用户可以在登录后查看离线这段时间收到的数据，使用数据库暂时将数据存入数据库，当客户登录时再讲数据发送给客户端，如果某个客户未开启聊天数据漫游，当客户端接收倒数据后就将这数据从数据库中删除

## 5、心得体会

在这个实验中，可以了解到C++在windows下如何使用Socket通信，不过在此之前我已经了解过Java 如何实现Socket通信。与Java实现的Socket通信相比，C++更加偏向底层，当然了，里面也没有说明计算机程序运行时如何将数据通过网卡发送到网络中。所以，我认为无论是C++还是Java实现Socket通信都只是通过调用现成API进行收发消息，想要了解这里面的真正原理，我觉得并不应该局限于这里

# 二、Tracert 与 Ping 程序设计与实现

## 1、内容

了解 Tracert 程序的实现原理，并调试通过。 然后参考 Tracert 程序和教材 4.4.2 节，编写一个 Ping 程序，并能测试本局域网的所有机器是否在线，运行界面如图 1 所示的 QuickPing 程序

Tracert实现原理： Tracert 程序关键是对 IP 头部生存时间(time to live)TTL 字段的使用,程序实现时是向目地主机发送一个 ICMP 回显请求消息，初始时 TTL 等于 1，这样当该数据报抵达途中的第一个路由器时， TTL 的值就被减为 0，导致发生超时错误，因此该路由生成一份 ICMP 超时差错报文返回给源主机。随后，主机将数据报的 TTL 值递增 1，以便 IP 报能传送到下一个路由器，并由下一个路由器生成ICMP 超时差错报文返回给源主机。不断重复这个过程，直到数据报达到最终的目地主机，此时目地主机将返回 ICMP 回显应答消息。这样，源主机只需对返回的每一份 ICMP 报文进行解析处理，就可以掌握数据报从源主机到达目地主机途中所经过的路由信息。

## 2、设计步骤

QuickPing程序的通过发送ICMP报文来测试是否能将数据包发送到一台主机，而QuickPing这通过遍历输入区间内的ip,如果可以收到正确的ICMP返回报文，则认为该子网内对应ip的主机在线

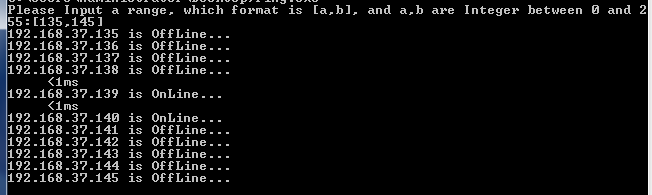
## 3、 调式过程

1）输入IP主机号区间

2）然后运行Ping函数，Ping函数会将数据包进行解析，根据解析进行判断数据是否到达

3）根据Ping的返回值判断某个IP对应的主机是否在线

## 4、结果及结果分析



在程序中已将网络号（也就是现在的网络前缀）192.168.37.固定在代码中，这样输入后面的主机区间[35,45]就可以查找区间内在线主机

上面的在线主机有

|  |
| --- |
| 192.168.37.139  192.168.37.140 |

当要实现动态输入网络号时，只需要添加几行代码即可

## 5、心得体会

在本实验中，对Trancert的具体实现有了更加深入的了解，以及在发送/解析IP数据报时如何进行校验码的计算，对一些问题有了答案：

1. 校验码在计算的时候如何处理超过16位是的进位？

|  |
| --- |
| /\*\*  \* 计算网际校验和函数  \* 计算检验和的方法有两个,加运算时，需循环进位  \* - 每16位取反求和再取反  \* - 先对每16求和再取反  \* 以下为第二种  \*/  USHORT checksum(USHORT \*pBuf, int iSize){  unsigned long cksum=0;  while(iSize>1){  cksum+=\*pBuf++;  iSize-=sizeof(USHORT);  }  if(iSize){  cksum+=\*(UCHAR\*)pBuf;  }  cksum=(cksum>>16)+(cksum&0xffff);  cksum+=(cksum>>16);  return (USHORT)(~cksum);  } |

1. ICMP数据包如何进行解析？

# 三、基于 IP 多播的网络会议程序

## 1、内容

设计一个图形界面的网络会议程序（实现文本多播方式即可）

## 2、设计步骤

Receiver 项目实现步骤：

(1)、创建一个 SOCK\_DGRAM 类型的 Socket。  
(2)、将此 Socket 绑定到本地的一个端口上，为了接收服务器端发送的多播数据。  
(3)、加入多播组。

(4)、接收多播数据。

Sender 实现步骤：

(1)、创建一个 SOCK\_DGRAM 类型的 Socket。  
(2)、加入多播组。

(3)、发送多播数据。

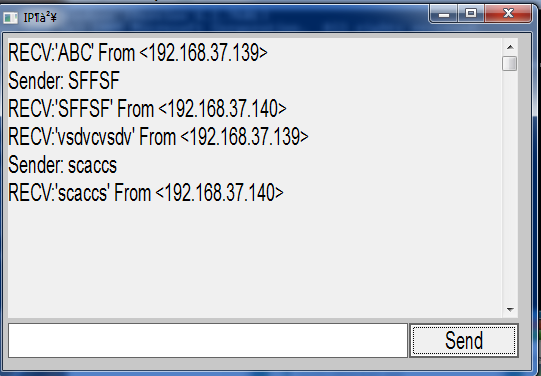
结合后，将接收的的过程放到线程里面去，这样就可以使用一个程序进行IP多播进行发送接收数据

## 3、 调式过程

1）将IP多播编译后的exe文件放到两台NAT网络模式下的两台虚拟机下（用于模拟同一局域网下的两台主机）

2）运行，任意一方发送消息，另一方都能接收到数据，当然，如果有多台主机，任意一个发送，其他的都能接收到

## 4、结果及结果分析



192.168.37.139是另一台主机发送的消息，

192.168.37.140是本主机发送的消息

## 5、心得体会

在该实验中，了解了如何使用多播进行发送接收信息，以及IP多播地址的使用，如IP多播智能用于目的地址。

# 四、网络嗅探器的设计与实现

## 1、内容

设计一个可以监视网络的状态、数据流动情况以及网络上传输的信息的网络嗅探器

## 2、设计步骤

1）得到数据包，先将其转存到内存里，以备以后再用。

2）分析当前的数据包，分析过程如下：

①数据包的前14个字节(Byte)代表数据链路层的报文头,其报文格式是前6Byte为目的MAC地址，随后的6个Byte为源Mac地址，最后的2Byte代表上层协议类型，之后就是IP数据报4Byte为源IP,4Byte目标IP，最后如果是UPD或TCP时，解析除端口号。

②根据1所分析到的协议类型进行类似1的迭代分析。这样就可以得到各层中的报文头信息和数据信息。

## 3、 调式过程

1）在Linux使用gcc编译源程序

2）执行编译后的文件

3）程序会将截获数据包的源MAC地址、目标MAC地址、源IP、目标IP、 源端口以及目标端口

4）在Linux按Ctrl+C终止程序运行

## 4、结果及结果分析

运行结果：

|  |
| --- |
| MAC: EE:FF:FF:FF:FF:FF==>00:16:3E:0E:17:EC  IP: 111.8.72.147 => 172.19.67.75  Protocol: TCP,source port: 40829,dest port: 22  MAC: EE:FF:FF:FF:FF:FF==>00:16:3E:0E:17:EC  IP: 111.8.72.147 => 172.19.67.75  Protocol: TCP,source port: 40829,dest port: 22  MAC: EE:FF:FF:FF:FF:FF==>00:16:3E:0E:17:EC  IP: 111.8.72.147 => 172.19.67.75  Protocol: TCP,source port: 40829,dest port: 22 |

这里面的具体含义：

|  |
| --- |
| 源MAC地址=>目标MAC地址  源IP地址=>目标IP地址  协议 源端口 目标端口 |

打印这个结果的原因是，我在本机中石油Xshell连接远程服务器，40829就是本机打开的端口与服务器的22端口连接，其中源地址是本地计算机所连接网络所在的公网IP地址，而里面的目标地址是对应目的局域网IP地址

## 5、心得体会

这个实验的学习，我可以对TCP、UDP、IP数据报以及MAC数据帧的组成更加了解，了解这些之后就可以将截获的数据报进行解析。这也就是网络嗅探器，的工作，它实时的监控着网络的数据传输，通过网络嗅探器可以及时发现网络故障，并是网络得到及时的修复。

# 五、电子邮件客户端程序设计与实现

## 1、内容

参照教材 6.5 节原理，设计一个电子邮件客户端程序

## 2、设计步骤

这里只涉及到了邮件的发送和接收，其实这个过程及时使用TCP跟相应的服务器建立连接，发送邮件时连接smtp.163.com（这里我使用的163邮箱）的25端口，接收邮件时连接pop3.163.com 的110端口，在这里的收发邮件的过程跟telnet发送接收邮件没有多大的差别

在建立连接后的发送的相应的命令如下

|  |  |
| --- | --- |
| 发送邮件 | 发送ehlo报文  ehlo 163.com  输入登录命令，依次输入登录命令，base64编码后的账号和客户端授权码  auth login  bHloY2NfdGVzdEAxNjMuY29t  cWF6d3N4MTI=  设置邮件发送方和邮件接收方  mail from: <lyhcc\_test@163.com>  rcpt to: <lyhcc\_email@163.com>  使用DATA发送数据，发送数据需要以相应的格式，并使用Cc拷贝一份给自己才能正常发送，不然会被认为是垃圾邮件而发送失败  data  from:lyhcc\_test<lyhcc\_test@163.com>  to:from:lyhcc\_email<lyhcc\_email@163.com>  Cc:lyhcc\_email<lyhcc\_email@163.com>  subject:Email  This is a spam mail  . |
| 接收邮件 | 账号和授权码  user [lyhcc\_test@163.com](mailto:lyhcc_test@163.com)  passwd qazwsx12  列出邮箱邮件  list  获取邮件,  retr 1 |

使用上面的命令读取的邮件处理邮件内容，还其他信息，还有就是那些有样式的邮件直接这样读取相应的数据会出现乱码

## 3、 调式过程

1）执行程序，输入指令

2）发送或者接受邮件

3）查看是否收到邮件或是否正确接收邮箱

## 4、结果及结果分析

发送邮件结果分析



读取邮件



## 5、心得体会

在这次实验中，我了解到了如何使用telnet进行发送邮件，读取邮件，以及如何使用在程序中使用socket进行发送/接收邮件。还有就是对于SMTP和POP协议的应用有了更好的了解