学生学号	0121410880117	实验课成绩	

武漢理工大學

学 生 实 验 报 告 书

实验课程名称　　通信原理

开 课 学 院　　计算机科学与技术学院

指导教师姓名　　刘维

学 生 姓 名　　陈鑫宇

学生专业班级　　软件工程 1401

2015 -- 2016 学年 第 2 学期

实验课程名称：_____通信原理_____

实验项目名称	采用 Winsock 在有线局域网上的点 - 点通信			实验成绩	
实 验 者	陈鑫宇	专业班级	软件 **1401**	组 别	
同 组 者	叶旺			实验日期	年 月 日

一部分：实验预习报告（ 包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等 ）

实验目的与要求

1. “采用 Winsock 在有线局域网上的点 - 点通信”
2. 熟悉 VB6.0 的控件和界面设计，进而熟悉 Winsock 的有关控件及编程方法。
3. 在了解所用的两个工作站和服务器的 IP 地址后，采用 VB6.0 的控件和 Winsock 控件编写并调试在有线局域网上的点 - 点通信程序。
4. 具有点对点通信功能，任意客户端之间能够发送消息。

编译语言与环境

1. 编程语言 C/C++等均可；本次实验采用 C++语言版本
2. 安装 vs2015 或更高版本的 Windows系统 pc 机

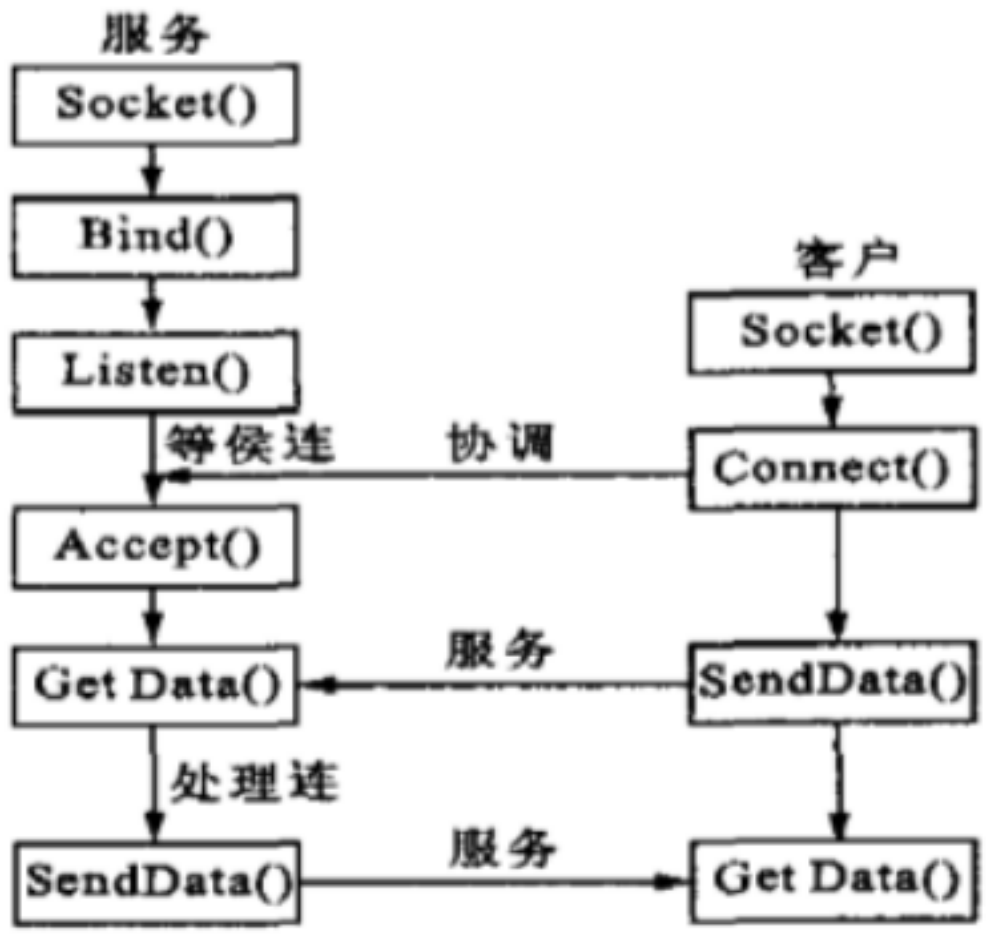


图 1 面向连接的点到点通信流程图

网络编程是通过使用套接字来达到进程间通信目的的编程，Socket 编程是网络编程的主流工具，Socket API 是实现进程间通信的一种编程设施，也是一种为进程间提供底层抽象的机制，提供了访问下层通信协议的大量系统调用和相应的数据结构。具体流程如上图所示。

1、建立连接

1) 服务程序调用 socket 创建一个新的套接字，并在传输层实体中分配表空间，返回一个文件描述符用于以后调用中使用该套接字；调用 bind 将一个地址赋予该套接字，使得远程客户程序能访问该服务程序；调用 listen 分配数据空间，以便存储多个用户的连接建立请求；调用 accept 将服务程序阻塞起来，等待接收客户程序发来的连接请求。当传输层实体接收到建立连接的 TPDU时，新创建一个和原来的套接字相同属性的套接字并返回其文件描述符。服务程序创建一个子进程处理此次连接，然后继续等待发往原来套接字的连接请求。

2) 客户程序调用 socket 创建一个新的套接字，并在传输层实体中分配表空间，返回一个文件描述符用于在以后的调用中使用该套接字；调用 connect 阻塞客户程序，传输层实体开始建立连接，当连接建立完成时，取消阻塞；

2、数据传输

双方使用 send 和 receive 完成数据的全双工发送。

3、释放连接

每一方使用 close 原语单独释放连接。

关键代码示例：

服务器端：

```
//定义服务器端　socket
	sockServer = socket(AF_INET, SOCK_STREAM,0);
	//设置服务器端　socket
	addrServer.sin_addr.S_un.S_addr = htonl(INADDR_ANY);//　本机 IP
	addrServer.sin_family = AF_INET;
	addrServer.sin_port = htons(6000);

	//将服务器　socket 绑定在本地端口
	bind(sockServer, (SOCKADDR *)&addrServer, sizeof(SOCKADDR));
	//Listen 监听端口
	listen(sockServer, 10);//10 为等待连接数目
	printf(" 服务器已启动：　\n 监听中 ...\n");

	len = sizeof(SOCKADDR);

	while (1)
	{
		//accept 会阻塞进程，直到有客户端连接上来为止
		sockClient = accept(sockServer, (SOCKADDR*)&addrClient, &len);
		//当客户端连接上来时，拼接字符串
		sprintf(sendBuf,　" 欢迎　ip:%s　的用户连接，这里是陈鑫宇的服务器，欢迎使用　　\n",
inet_ntoa(addrClient.sin_addr));
		//向客户端发送字符串
		send(sockClient, sendBuf, strlen(sendBuf) + 1, 0);
		//获取客户端返回的数据
		recv(sockClient, recvBuf, 100, 0);
		//打印客户端返回的数据
		printf("%s\n", recvBuf);
		//关闭 socket
		closesocket(sockClient);
	}
```

客户端：

```
// 新建服务器端 scoket
    sockServer = socket(AF_INET, SOCK_STREAM, 0);

    // 定义要连接的服务端地址
    addrServer.sin_addr.S_un.S_addr = inet_addr("10.139.14.180");//  目标 IP (100.64.175.119 是本机地址 )
    addrServer.sin_family = AF_INET; //  协议类型是 INET
    addrServer.sin_port = htons(6000);//  连接端口 1234

// 让 sockClient 连接到 服务端
connect(sockServer, (SOCKADDR *)&addrServer, sizeof(SOCKADDR));

// 从服务端获取数据
recv(sockServer, recvBuf, 100, 0);

// 打印数据
printf(" 服务器端输入： %s\n", recvBuf);

message = "大家好，我是叶旺 ";
printf(" 向服务器发送： %s\n",message);
// 发送数据到服务端
send(sockServer, message, strlen(message) + 1, 0);

// 关闭 socket
closesocket(sockServer);
WSACleanup();
```

第二部分： 实验过程记录 （可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

服务器 ip 地址为 192.168.1.110

客户端发送的信息成功传输到了服务器并显示了出来，

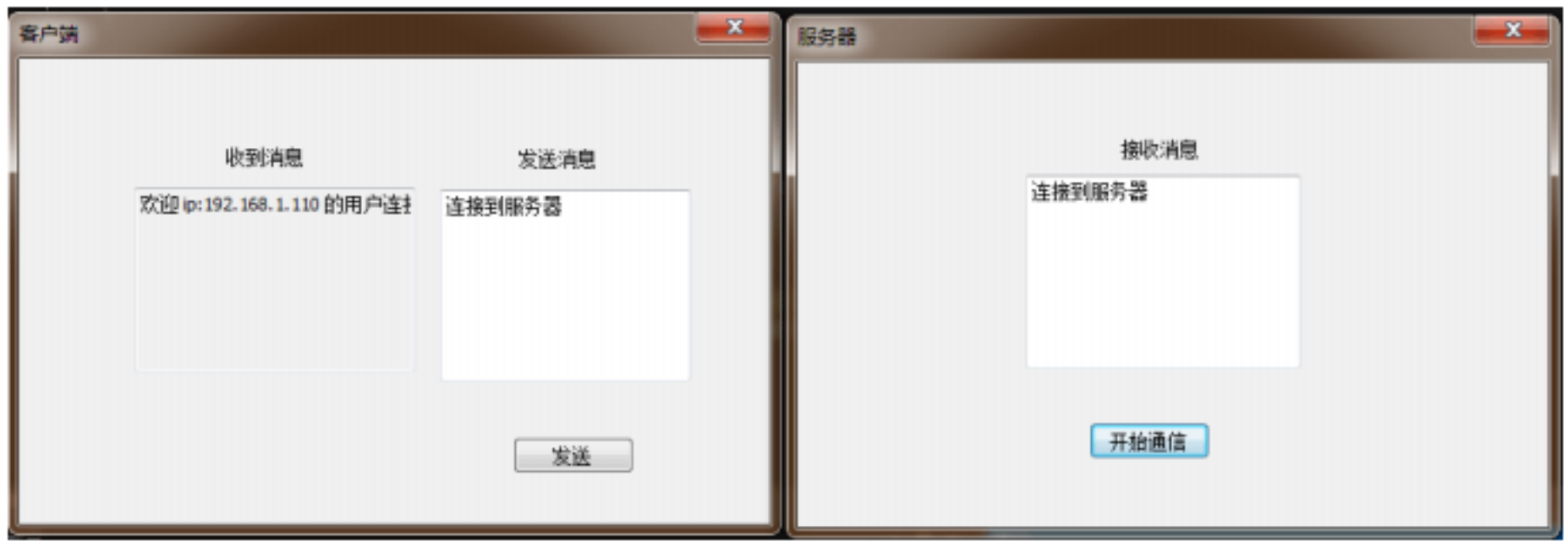

客户端
收到消息
发送消息
发送


服务器
接收消息
开始通信

第三部分　结果与讨论 （可加页）

一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）

打开服务器程序，然后点击开始通信按钮，打开客户端程序在客户端输入数据，数据成功从客户端传输到了服务器端，并成功显示了出来，实验成功。



二、小结、建议及体会

通过本次实验能够感受到网络在实际当中的运用，使我对网络编程有了更深的概念。同时让我发现自己知识结构的欠缺。虽然基本功能已经实现，但还是存在很多不稳定的问题尚待解决。

此次实验是对 Socket 的一个应用，让我更深的了解了 Socket 函数的作用及使用方法。实验过程中遇到了很多的问题，例如：程序只能通信一次。刚开始我以为建立的 socket 套接口只要建立一次并更新连接地址就可以多次使用。当初不知道问题存在的在于这，在程序加入很多错误报告代码，发现再客户端再次 connect 的时候返回负值。在网上查看很多相关代码之后，发现问题所在，把建立 socket 的代码写进客户端的循环体中，每次通信都重新建立 socket 套接口就可以了。

实验项目名称	采用 Winsock 在无线局域网上的点 - 点群发通信			实验成绩	
实 验 者	陈鑫宇	专业班级	软件 **1401**	组 别	
同 组 者	叶旺，杨彪			实验日期	年 月 日

一部分：实验预习报告（ 包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等 ）

实验目的与要求

1.“ 采用 Winsock 在无线局域网上的点 - 点群发通信 ”

2. 修改实验 01 的界面设计，设计为多个接收地址的群发界面（因为群发是一对多的循环点 - 点通信），并修改相应的程序。

3. 在了解所用的 3 个工作站和服务器的 IP 地址后，采用 VB6.0 的控件和 Winsock 控件编写并调试在无线局域网上的点 - 点群发通信程序。

编译语言与环境

1. 本次实验采用 C++编程语言版本

2. 安装 vs2015 的 Windows系统 pc 机

问题分析

点到群的通信就是客户端将消息发送至多个服务器端。 相当于在客户端加入一个循环，依次给不同 ip 的服务器发送消息。

在实验一的基础上， 修改服务器端的消息响应方法可以实现点到群的通信。 实验一中点到点的通信原理是， 客户端将消息发送到服务器端， 然后服务器端反馈客户端相应的信息。 在本实验中可以修改为客户端将消息发送至多个服务器端， 然后接受传送回来的信息。相当于在客户端加入一个循环，依次给不同 ip 的服务端发送消息。

关键代码示例：

服务器端：

```
// server.cpp :        实现文件
//
#define     _CRT_SECURE_NO_WARNINGS

#include    "stdafx.h"
#include    "通信实验 .h"
#include    "server.h"
#include    "afxdialogex.h"
#include  <WinSock2.h>
#include  <stdio.h>
#pragma comment( lib   , "ws2_32.lib"    )
static     int   num = 0;
// server      对话框

IMPLEMENT_DYNAMIC(server , CDialogEx )

server ::server(    CWnd* pParent   /*=NULL*/ )
        :   CDialogEx ( IDD_DIALOG1, pParent )
{
}
server ::~server()
{
}
void   server ::DoDataExchange(  CDataExchange*  pDX)
{
        CDialogEx ::DoDataExchange(  pDX);
}
BEGIN_MESSAGE_MAP(server , CDialogEx )
        ON_BN_CLICKED(IDC_BUTTON1, & server ::OnBnClickedButton1)
        ON_WM_TIMER()
END_MESSAGE_MAP()
// server      消息处理程序
void   server ::OnBnClickedButton1()
{
        // TODO:    在此添加控件通知处理程序代码
        //SetTimer(1, 50000, NULL);
        int   err;   //  错误信息
        int   len;

        char  sendBuf[100];     //  发送至客户端的字符串
        char  recvBuf[100];     //  接受客户端放回的字符串
```

```
    SOCKET sockServer;    // 服务器端 Socket
    SOCKADDR_IN addrServer;    // 服务器端地址
    SOCKET sockClient;    // 客户端 Socket
    SOCKADDR_IN addrClient;    // 客户端地址

    WSADATA wsaData; //winsock 结构体
    WORD wVersinRequested;  //winsock 版本

                                    // 配置 Windows Socket 版本
    wVersinRequested =  MAKEWORD(2,2);
    // 初始化 Windows Socket
    err = WSAStartup(wVersinRequested, &wsaData);

    if (err != 0)
    {
        // 启动错误，程序结束
        return ;
    }
    /*
    确认 WinSock DLL 支持 2.2
    */
    if ( LOBYTE(wsaData.wVersion) != 2 ||       HIBYTE(wsaData.wVersion) != 2)
    {
        // 启动错误
        WSACleanup(); // 终止 WinSock 2 DLL 的使用
        return ;
    }
    // 定义服务器端 socket
    sockServer = socket( AF_INET, SOCK_STREAM, 0);
    // 设置服务器端 socket
    addrServer.sin_addr.S_un.S_addr = htonl( INADDR_ANY); // 本机 IP
    addrServer.sin_family = AF_INET;
    addrServer.sin_port = htons(6000);
    // 将服务器 socket 绑定在本地端口
    bind(sockServer, ( SOCKADDR*)&addrServer,  sizeof ( SOCKADDR));
    //Listen 监听端口
    listen(sockServer, 5);      //5 为等待连接数目
    printf( "服务器已启动： \n 监听中 ...\n" );
    len =  sizeof ( SOCKADDR);
    //accept 会阻塞进程，直到有客户端连接上来为止
    sockClient = accept(sockServer, ( SOCKADDR*)&addrClient, &len);
    // 当客户端连接上来时，拼接字符串
    sprintf(sendBuf,  "欢迎 ip:%s 的用户连接，发送成功，欢迎使用 \n" ,
```

```
inet_ntoa(addrClient.sin_addr));
        // 向客户端发送字符串
        send(sockClient, sendBuf, strlen(sendBuf) + 1, 0);
        // 获取客户端返回的数据
        recv(sockClient, recvBuf, 100, 0);
        // 打印客户端返回的数据
        CString   text;
        text    = recvBuf;
        SetDlgItemText  (IDC_EDIT1, text);
        // 关闭 socket
        closesocket(sockClient);
}
```

客户端：

```
#include    <Winsock2.h>
#include    <stdio.h>
#pragma comment(lib  ,  "ws2_32.lib"    )
#include    "stdafx.h"
#include    "通信实验 .h"
#include    "client.h"
#include    "afxdialogex.h"
// client       对话框
IMPLEMENT_DYNAMIC(client  , CDialogEx )
client   ::client(     CWnd*  pParent   /*=NULL*/ )
        :   CDialogEx ( IDD_DIALOG2,  pParent )
{
}
client   ::~client()
{
}
void   client   ::DoDataExchange(  CDataExchange*   pDX)
{
        CDialogEx ::DoDataExchange(  pDX);
}
```

```
BEGIN_MESSAGE_MAP(client , CDialogEx)
	ON_BN_CLICKED(IDC_BUTTON1, &client::OnBnClickedButton1)
END_MESSAGE_MAP()


// client 消息处理程序


void client::OnBnClickedButton1()
{
	// TODO: 在此添加控件通知处理程序代码
	char * ip1 = "192.168.1.110";
	char * ip2 = "192.168.1.111";
	CString text;
	GetDlgItemText(IDC_EDIT1, text);

	int nLength = text.GetLength();
	int nBytes = WideCharToMultiByte(CP_ACP, 0, text, nLength, NULL, 0, NULL, NULL);
	char * message = new char [nBytes + 1];
	memset(message, 0, nLength + 1);
	WideCharToMultiByte(CP_OEMCP, 0, text, nLength, message, nBytes, NULL, NULL);
	message[nBytes] = 0;
	int err;
	char recvBuf[100];

	SOCKET sockClient;	// 客户端 Scoket
	SOCKADDR_IN addrServer;	// 服务端地址

	WSADATA wsaData;
	WORD wVersionRequested;

	wVersionRequested = MAKEWORD(2, 2);

	err = WSAStartup(wVersionRequested, &wsaData);

	if (err != 0)
	{
		return;
	}

	if ( LOBYTE(wsaData.wVersion) != 2 || HIBYTE(wsaData.wVersion) != 2)
	{
		// 启动错误，程序结束
		WSACleanup();
```

```
            return ;
        }

        //  新建客户端 scoket
        sockClient = socket(  AF_INET,  SOCK_STREAM,0);

        //  定义要连接的服务端地址
        addrServer.sin_addr.S_un.S_addr = inet_addr(ip1);           //  目标 IP (175.0.170.28  是本机地址 )
        addrServer.sin_family =  AF_INET;  //  协议类型是 INET
        addrServer.sin_port = htons(6000);      //  连接端口 1234

                                                //  让 sockClient  连接到 服务端
        connect(sockClient, (  SOCKADDR*)&addrServer,  sizeof (SOCKADDR));
        //  发送数据到服务端
        send(sockClient, message, strlen(message) + 1, 0);

        //  从服务端获取数据
        recv(sockClient, recvBuf, 100, 0);

        //  打印数据
        text  = recvBuf;
        SetDlgItemText (IDC_EDIT2, text);
        //  关闭 socket
        closesocket(sockClient);
        WSACleanup();
        //  定义要连接的服务端地址二
        addrServer.sin_addr.S_un.S_addr = inet_addr(ip2);           //  目标 IP
        addrServer.sin_family =  AF_INET;  //  协议类型是 INET
        addrServer.sin_port = htons(6000);      //  连接端口 1234
                                                //  让 sockClient  连接到 服务端
        connect(sockClient, (  SOCKADDR*)&addrServer,  sizeof (SOCKADDR));
        //  发送数据到服务端
        send(sockClient, message, strlen(message) + 1, 0);

        //  从服务端获取数据
        recv(sockClient, recvBuf, 100, 0);

        //  打印数据
        text  = recvBuf;
        SetDlgItemText (IDC_EDIT2, text);
        //  关闭 socket
        closesocket(sockClient);
        WSACleanup();
}
```

第二部分：实验过程记录 （可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

客户端分别将数据传输到服务器 a 和服务器 b，服务器成功显示客户端发送的信息，实验成功。

实验过程中出现了编译器报错问题，查询资料，修改参数后解决了问题。

第三部分　结果与讨论（可加页）

二、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）

通过修改实验一所定义的客户端的消息发送可以实现点到多个点的群发通信，客户端依次将信息发送到第一个　ip，第二个 ip，第三个 ip

客户端发送消息，计算机　**a** 上服务器接收到消息

计算机　**b** 上服务器接收到消息

二、小结、建议及体会

通过在实验一的基础上继续开发，　我实现了多个接收地址的群发界面，　更加深入了解了 winsock 通信机制，以及利用　mfc 编程实现的方法。

了解了群发是一对多的循环点　- 点通信，熟悉了通信的技术以及实现方法。

上课时认真学习通信原理就能快速正确的完成实验，收获知识。