

基于 VB 的网络通信

郭招娣

(杭州应用工程技术学院 现教中心 杭州 310012)

摘 要 介绍了 VB 中包含的 winsock 控件,以及用该控件实现网络点到点通信的方法。

关键词 VB winsock 控件 TCP 和 UDP 协议 网络通信

中图分类号 TP919.2

网络已成为人们生活中不可或缺的一部分.编写网络程序却是编程中较难的部分.不过,Visual Basic(简称 VB)正在使网络编程变得越来越简单. Visual Basic 专业版和企业版中包括了一些特别设计的控件,这些控件封装了与 Internet 有关的技术,它们提供了和 Internet 有关的功能.这些控件包括 Internet Transfer 控件、WebBrowser 控件,以及 Winsock 控件.其中 Winsock 控件用于与远程计算机建立通信,通过传输控制协议 TCP(Transmission Control Protocol)或者用户数据报文协议 UDP(User Datagram Protocol)进行数据交换.

1 Winsock 控件^[1]

Winsock 控件是 VB5.0 后新增功能,它解决了以往应用 VB 编程时网络中应用程序之间无法实现点对点通信的难题. Winsock 使用 TCP 协议和 UDP 协议允许建立并保持一个到远程计算机上的连接,且可以在连接结束之前实时地进行数据交换.用户仅通过设置属性并借助事件处理就能够轻而易举地连接到一个远程的计算机上,而且只用几个方法就可以实现数据通信. Winsock 控件的主要属性、事件和方法如下:

LocalPort 属性:设置本地计算机使用的端口;

LocalHostName 属性、LocalIP 属性:分别返回本地机器名和本地机器的 IP 地址;

Protocol 属性:设置 Winsock 控件所使用的协议——TCP 或者 UDP(一般设置为 0);

RemoteHost 属性、RemoteHostIP 属性、RemotePort 属性:分别返回或设置远程计算机主机名或点格式下的 IP 地址字符串以及要连接的远程计算机端口号;

Accept 方法:在处理 ConnectionRequest 事件时用这个方法接受新连接;

Bind 方法:指定用于连接的 LocalPort 和 LocalIP;

Close 方法:关闭连接;

Connect 方法:要求连接到远程计算机;

GetData 方法:获取当前的数据块并将其存储在变体类型的变量中;

Listen 方法:创建套接字并将其设置为侦听模式;

SendData 方法:将数据发送给远程计算机;

Close 事件:当远程计算机关闭连接时出现;

Connect 事件:该事件是在到服务器的连接建立之后发生;

ConnectionRequest 事件:当远程计算机请求连接时出现;

DataArrival 事件:当新数据到达时出现。

2 TCP 协议和 UDP 协议^[2]

协议是一组程序或标准,通过它们,不同类型的计算机得以通信.不同类型的计算机就好像持不同语言的各个国家的人一样,根本无法交流.例如,我们知道数据在计算机存储中无非是 0 和 1.同样一个双字节的字符串,有的计算机可能要把高位字节放在首位,有的会放在后面.这样就会造成表达上的混乱.为了解决这种问题,我们必须制订一种规范,只要按照这个规范向网络读写数据,就可以保证互相理解.这个规范就是协议.

2.1 TCP 协议

TCP 协议(Transmission Control Protocol)即传输控制协议,是一个基于连接的协议,是专门用于设计在不可靠的因特网上提供可靠的、点到点通信的协议,可以将它同电话系统相比.在开始数据传输之前,用户必须先建立连接.

2.2 UDP 协议

UDP 协议(User Datagram Protocol)即用户数据报文协议,是一种无连接协议,两台计算机之间的传递类似于传递邮件:消息从一台计算机发送到另一台计算机,但是两者之间没有明确的连接.很多有一个请求和一个响应的客户-服务器应用程序采用 UDP 协议,这样可以避免建立和释放连接的麻烦.

如何理解 TCP 协议和 UDP 协议之间的区别?前面我们已把 TCP 和 UDP 看作打电话和写信.打电话时,拨通电话号码,当对方拿起听筒时,这时在我和对方之间建立了一个连接.只要这个连接维持,也就是电话线路始终维持连通,我们就可以继续通话.只需要在第一次建立连接,就可以持续不断的交换信息,这就是基于连接的 TCP 的特点.

与此相对的是寄信.在寄信时,发信方和收信方之间没有一个明确的、物理上的连接,但是只要在信封上写明了收件人的姓名地址,以及发信人的收件地址,我们可以肯定在正常情况下,对方一定可以收到这封信.为了维持信息的连接不断,就必须不断地发信、收信.但是必须在每一封信上都要写明了收件人的姓名地址,以及发信人的收件地址.这就是基于无连接的 UDP 的运行模式.

3 用 Winsock 控件建立一个简单的聊天程序

使用 Winsock 控件时,通信的双方需要选定相同的协议.TCP 协议适用于传送大容量、需要安全性保证的数据文件;而 UDP 协议适用于有一个请求和一个响应的客户-服务器应用程序,这样可以避免建立和释放连接的麻烦,或者通信的数据量很小的时候.设计时可以使用 Winsock1.Protocol = sockTCPProtocol 方法选定协议,或者直接在属性窗口中选定协议,另外要找到你的计算机的名称,并把它添入 Winsock 的 LocalHost 属性中.

使用 TCP 协议时,如果需要创建一个客户应用程序,就必须识别服务器的名称或 IP 地址,客户

应用程序发出连接请求.服务器应用程序的通信端口随时都将仔细监听对方发出的消息,这是系统进行可靠连接的保证.一旦连接发生,任何一方都可以通过 SendData 发送数据,并借助 GetData 把自己的数据分离出来.传送数据时,需要先设定服务器的 LocalPort 属性,客户机则只需要把 RemoteHost 属性设定为服务器以太网的地址,并设定与服务器 LocalPort 属性相同的端口地址,借助 SendData 方法开始发送消息.服务器和客户机都在 DataArrival 事件中用 GetData 方法获得发送的信息.

创建一个应用程序时,首先要确定你建立的是客户方应用还是服务器服务,只有建立的服务器应用开始工作,并进入监听状态时,客户应用程序才开始建立连接,进入正常的通信状态.笔者建立了一个应用程序,它的功能是一个简单的聊天程序,两个人可以通过它进行实时的交谈.下面是建立服务器应用程序的方法:

(1) 建立一个标准 EXE 工程;

(2) 在一个窗体 Frmserver 上放置两个文本框、两个标签框、一个命令按钮(Cmdsend)和一个 Winsock(名为 tcpserver)控件,程序运行后窗体界面如图 1 所示:

(3) 在 Frmserver 代码编辑器窗口中编写如下程序代码:

```
private Sub Form_Load()
    '将 LocalPort 属性设置为一个整数.
    '然后调用 Listen 方法.
    tcpserver.LocalPort = 1001
    tcpserver.Listen
End Sub

Private Sub Cmdsend_Click()
    '当用户单击“发送”命令按钮时,使用 SendData 方法发送输入的字符串.
    tcpserver.SendData Txtsenddata.Text
End Sub

Private Sub tcpServer_ConnectionRequest(ByVal requestID As Long)
    '检查控件的 State 属性是否为关闭的.如果不是,在接受新的连接之前先关闭此连接.
    If tcpserver.Stater < > sckClosed Then tcpserver.Close
    tcpserver.Accept requestID '接受具有 requestID 参数的连接.
End Sub

Private Sub tcpServer_DataArrival(ByVal bytes Total As Long)
    Dim strData As String '为进入的数据声明一个变量.
    tcpserver.GetData strData '调用 GetData 方法获取数据
    Txtoutput.Text = strData '并将数据赋予给接收消息文本框
End Sub
```

建立客户应用程序的方法为

(1) 建立一个标准 EXE 工程;

(2) 在一个窗体 Frmclint 上放置两个文本框、两个标签框、两个命令按钮和一个 Winsock(名为 tcpclient)控件,程序运行后窗体界面如图 2 所示:

(3) 在 Frmclint 代码编辑器窗口中编写如下程序代码:

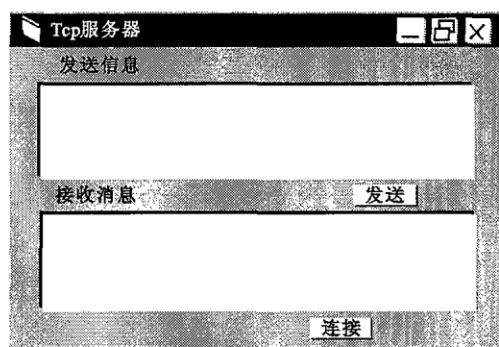


图 1 服务器应用程序界面

```
Private Sub Form_Load()
    'Winsock 控件的名字为 tcpClient.
    '注意:要指定远程主机,可以使用 IP 地址(例如
    "121.111.1.1"),也可以使用.'计算机的"好听的名字"
    如下所示.
```

```
    tcpclient.RemoteHost = "guo"
    tcpclient.RemotePort = "1001"
```

```
End Sub
```

```
Private Sub cmdConnect_Click()
```

```
    '调用 Connect 方法,初始化连接.
```

```
    tcpclient.Connect
```

```
End Sub
```

```
Private Sub tcpClient_DataArrival(Byval bytes Total As Long)
```

```
    Dim strData As String
```

```
    tcpclient.GetData strData
```

```
    Txtoutput.Text = strData
```

```
End Sub
```

```
Private Sub Cmdsend_Click()
```

```
    tcpclient.SendData Txtsend.Text
```

```
End Sub
```

使用 UDP 协议建立对等通信和通过 TCP 建立客户/服务器通信的方法略有不同,因为 UDP 协议不需要显式的连接.在上面的 TCP 应用程序中,一个 Winsock 控件必须显式地进行“监听”,另一个必须使用 Connect 方法初始化连接.使用 UDP 协议时服务器不需要“监听”,客户机也不需要连接,建立客户机和服务器应用程序没有明显的差异,建立的是对等通信.此过程通过以下几步实现:

- (1) 设定 Winsock 的 RemoteHost 属性为一个通信的计算机名称;
- (2) 设定 Remoteport 为一个接口号;
- (3) 调用 Winsock 的 Bind 事件绑定本地的接口号.具体设定方法为

```
Private Sub Form_Load()
```

```
With Winsock1
```

```
    .RemoteHost = "guo"
```

```
    .RemotePort = 1001 '远程连接号
```

```
    .Bind 1002 '绑定的本地号
```

```
End With
```

```
End Sub
```

程序的其它部分与 TCP 方法类似,即通过 SendData 和 GetData 方法发送或提取数据.UDP 和 TCP 协议在使用中各有特点,如果灵活使用,可以得到很好的效果.

4 结束语

以上例程实现的是一个非常简单的点对点网络通信,在此基础上略加改造,可以形成功能复杂的实时计算机网络交互通信系统,用于控制、图形仿真等. Winsock 给我们提供了一种简便的数据传

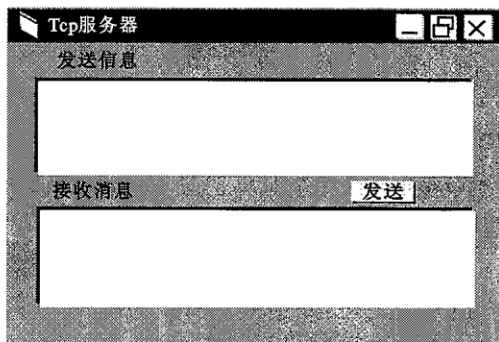


图 2 客户机应用程序界面

送方法,使我们得以轻松地实现网络点对点通信.

参 考 文 献

- 1 Microsoft corporation. 中文 Visual Basic 6.0 程序员指南. 微软中国有限公司译. 北京: 科学出版社, 1998
- 2 Andndrew S Tanenbaum. 计算机网络(第三版). 北京: 清华大学出版社, 1999

Network communication based on visual basic

Guo Zhaodi

(Hangzhou Institute of Applied Engineering, Hangzhou 310012)

Abstract This paper introduces the winsock control in VB, and intrduces the way of point-to-point communication with the winsock control.

Key words visual basic winsock control TCP and UDP protocol network communication