

用 VB 实现基于 Windows 通信机制的 大文件点对点传输

周化祥, 胡志刚

(中南大学 信息科学与工程学院, 湖南 长沙 410004)

摘要: 从点对点技术介绍入手, 阐述并比较分析了常见的点对点网络传输的技术方法; 在分析了大文件数据在 Windows 通信机制下的处理过程的基础上, 用 VB 语言设计并实现了基于 Windows 通信机制的大文件点对点传输控制方法, 构建了服务器端和客户端的程序, 有效地避免接收客户端的资源被通信程序大量占用。

关键词: 点对点技术; 大文件传输; Windows 通信机制

中图分类号: TP89

文献标识码: A

文章编号: 1673-9833(2008)04-0053-04

Large File Peer-to-Peer Transmission Based on Windows Communication Mechanism with VB

Zhou Huaxiang, Hu zhigang

(School of Information Science and Engineering, Central South University, Changsha 410004, China)

Abstract: In view of introduction of peer-to-peer technology, its methods of internet transmission is then compared and analyzed. On the basis of analyzing the handling process of large file under Windows communication mechanism, the peer-to-peer transmission control method based on Windows communication mechanism is designed and carried out by VB, which built the server software and client server, and can avoid remarkably the occupation on the resources received client by communication programs.

Key words: peer-to-peer technology; large file transmission; Windows communication mechanism

近年来, 随着 Internet 的发展和普及, 人们对网络传输的需求量越来越大, 远程教育、网络会议等众多新兴软件的产生, 对大文件的传输问题提出了挑战。传统的大文件传输通常在客户端建立一个较大的 Buffer 缓冲区, 这样不仅会大量占用系统资源, 而且传输的速度十分缓慢, 不能实时地显示传输进度。由于传输的时间比较长, 期间发生中断或计算机重启等意外的可能性非常高。

针对以上问题, 本文在 B/S 架构的基础上, 利用 Windows 通信机制, 采用 VB Winsock 控件技术实现了一种新的点对点传输方法, 它能很好的满足当前大文

件传输的需要。

1 点对点传输概述

1.1 常见的点对点传输方法比较分析

实现点对点通信的方法大致可以分为 3 类: 一是使用高级语言基于 Socket 套接字编程; 二是基于 MPI (消息传递接口) 通信编程; 三是基于中间件的通信编程。在网络通信中, 通信双方可能是同构的, 也可能是异构的。通信软件基于的操作系统可能是相同的, 也可能是不同的。因此, 在实现点对点通信中, 也要解决异构环境下的编程问题。

收稿日期: 2008-04-18

作者简介: 周化祥 (1969-), 男, 湖南长沙人, 长沙商贸旅游职业技术学院讲师, 中南大学硕士研究生, 主要研究方向为软件工程和数据库。

使用基于 Socket 套接字编程方式是解决异构环境下通信问题的最佳方式,它成本低,不需要额外的系统支持,适合于各种操作系统。同时对传统的套接字通信方法进行改进,使用多线程并发程序设计方法,让每一个数据通信事务构成一个线程,并使其并发执行,以达到提高传输速度和吞吐量^[1]。

1.2 大文件点对点传输方法分析

目前基于点对点传输技术的大文件传输,主要的方法就是首先在接收客户端建立一个较大的 Buffer 缓冲区,以接收大文件数据,但是这样的方法会大量占用接收客户端的系统资源,而且传输的可靠性与无错性无法得到有效的保障^[2]。

目前研究的比较多的传输方法是,首先将大文件分割压缩为若干个小数据文件,在发送和接收双方之间建立通信通道,实现点对点传输,在接收客户端编程实现对接收到的若干个小数据文件进行组合解压缩,从而实现大文件的快速传输;另一方面,能够有效避免接收客户端的资源被通信程序大量占用,实现了多线程的并行运行^[3]。

Microsoft 在 Sockets API 基础上创建了 Winsock 控件,专门用于 Windows 接口,与 Sockets 完全兼容,Winsock 的开发工具可以在 VC、VB 的编译器中找到,因此,采用 VB 实现大文件的点对点传输是可行的。

2 用 VB 实现基于 Windows 通信机制的大文件点对点传输

2.1 Winsock 方法

Winsock 控件可以应用在点对点的文件传送应用程序之中^[4],它的工作原理如下:客户端向服务器端发出连接请求,服务器端则不停地监听客户端的请求,当两者的协议沟通时,客户端和服务器端之间就建立了连接,这时客户端和服务器端就可以实现双向数据传输。实际编程中,必须分别建立 1 个服务器端应用程序和 1 个客户端应用程序,两个应用程序中分别有自己的 Winsock 控件。首先设置 Winsock 控件使用的协议,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 协议适用于需要分别与很多下属通信,或者建立的连接比较多且为时变的情况,特别是在数据量很小的时候。

2.2 Windows 通信机制分析

应用程序的网络通信归根结底是利用相同的通信协议来完成信息的传输,应用程序和 Winsock 都工作在 Windows 的用户模式下,操作系统仅仅通过 Winsock 是不能完成网络间的通信,还需要底层的支持,而套接字仿真器(套接字核心模式驱动程序)和传输驱动程序接口(Transport Driver Interface, TDI)是

负责操作系统核心态环境下的网络通信,起到了 Winsock 和传输协议之间的通信桥梁作用。

Winsock 是网络通信应用程序套接字仿真器间的接口,TDI 是套接字仿真器和传输协议间的接口核心模式,负责核心模式驱动程序与传输协议间的通信^[5]。

当应用程序利用 Winsock 发送和接受数据时,并不是由 Winsock 从网络上发送和接收数据的,而是由核心模式驱动程序 AFDSYS 负责管理发送和接收缓冲区来发送和接收数据,也就是说,当应用程序调用 send 或 WSASend 函数来发送数据时,AFDSYS 把数据复制进他自己的发送缓冲区,然后 send 或 WSASend 函数立即返回,AFDSYS 在后台负责把数据发送出去,

AFDSYS 管理的发送缓冲区 SO-SNDBUF 和接收缓冲区 SO-RCVBUF 在缺省时 2 个缓冲区的大小都为 8 192 字节,但可以根据实际要求由应用程序设定,由于我们传输的对象是大数据量文件,因此需要对系统的发送缓冲区和接收缓冲区作相应的设定,以保障大数据量的文件数据的发送和接收。

2.3 大文件传输的处理过程分析

为了实现网络环境中可靠的文件传输,首先需要设计 2 个程序,即发送端和接收端,发送端指定要发送的文件,接收端可以将需要发送的文件名和路径包含在控制指令中以启动发送端,文件将以 Stream 流的方式进行传输。

文件分割:

由于在网络上一次传输的数据大小不能是任意的,当文件较大时必须将其分割,采用 Stream 流进行数据传输的时候,可利用指针定位的方式来实现文件的分割:1)在发送端将要传输的文件生成 Stream 流,将指针定位于要发送的数据的头部并指定一次要发送的数据块的大小;2)当一次数据块发送完成后,指针移动值的大小为本次发送的数据块的大小;3)当接收到下一个传输控制指令后,开始下一块数据的发送,这样直到数据发完为止。

数据发送:

在数据发送过程中还需要判断数据发送是否结束和处理最后数据块大小不足的情况,具体方法为^[6]:1)开始发送数据之前要设定文件剩余块的大小,初始时剩余块大小即为文件大小;2)发送数据时,首先比较剩余块和设定发送块的大小,若剩余块大,则本次数据发送块即为设定块大小;3)数据发送完毕后,将剩余块大小减去发送块大小。若剩余块小,则说明该块已经为文件最后一块,设定本次数据发送块即为剩余块大小,此时剩余块减去发送块后大小为 0,文件发送完毕。

数据接收:

1)在接收端接收到数据流后,首先将其长度记

下, 并将数据写入缓冲区; 2) 发出下一数据块的传输控制指令, 以启动下一数据块的传输; 3) 将缓冲区的数据追加写入一个打开的文件中。当到达的数据长度比指定的一次发送的数据块小时, 可以确定这是最后一个数据块, 说明数据传输已经完成, 这时可以把打开的文件关闭, 结束一次文件的传输。

可以看出, 传输过程是发送端和接收端一个相互协作的过程, Socket 通信在 Windows 中是以排队的形式由操作系统处理。因此, 网络通信的基本方式是请求和应答。Socket 编程也遵循数据分包传送这一基本规则, 每次发送和接收 1 个包, 以保证数据传输的安全性和稳定性, 同时也不至于过多地占用系统资源。

在接收端只能用文件追加写入的方式来恢复文件, 而不能将所有数据都追加写入流后, 再将流数据存储成文件。经过实践表明, 当传输文件较小时用后一种方式还能胜任, 当发送文件过大时接收端就会产生内存溢出错误, 即使文件较小不溢出, 对传输性能也有影响。

2.4 大文件点对点传输的 VB 实现

2.4.1 服务器端程序

1) 创建 Socket 服务器

向窗体添加 Winsock 控件, 命名为 SS, 在 LOAD 事件中增加以下代码:

```
Sub Form1_load ()
    ss.localPort = 2000 '服务器端口号避免与熟知端口冲突
```

```
    ss.listen '开始侦听
```

```
End Sub
```

2) 接受客户连接请求

客户连接到服务器, 产生 connectionRequest 事件

```
Sub SS_connectionRequest (Byval Requested As Long)
```

```
    ss.close
```

```
    ss.acceptRequestID 'accept 方法接受一个连接请求
```

```
End Sub
```

3) 接收客户发送的数据

客户向服务器发送的数据到达, 产生 Data Arrival 事件

```
Sub SS_DataArrival (Byval bytestotal As Long)
```

```
    Dim s As string
```

```
    ss.GetData s 'GetData 方法用于接收数据
```

```
    text1.text = s
```

```
End Sub
```

4) 向客户发送数据

点击“Send”按键, 产生 Click 事件

```
Sub Send_clkick ()
```

```
    ss.SendData text2.text
```

```
End Sub
```

5) 响应客户关闭连接

客户关闭连接时, 服务器会产生 close 事件

```
Sub SS_close
```

```
    ss.close '关闭当前连接
```

```
    ss.listen '重新开始侦听
```

```
End Sub
```

6) 主动关闭连接

```
Sub Form1_Unload (Cancel as integer)
```

```
    ss.close
```

```
End Sub
```

2.4.2 客户端程序

1) 创建客户连接服务器

向窗体添加 Winsock 控件, 命名为 SC, 点击“CONNECT”按键, 产生 Click 事件

```
Sub CONNECT_Click ()
```

SC.RemoteHost = text1.text '服务器主机名或 IP 地址

```
SC.RemotePort = text2.text '服务器端口号
```

```
End Sub
```

2) 接收服务器发送的数据

服务器向客户发送的数据到达, 产生 DataArrival 事件

```
Sub SC_DataArrival (Byval bytestotal As Long)
```

```
    Dim s As string
```

```
    SC.GetData s '接收数据
```

```
End Sub
```

3) 向服务器发送数据

点击“Send”按钮, 产生 Click 事件

```
Sub Send_clkick ()
```

```
    SC.SendDatatext2.text
```

```
End Sub
```

4) 响应服务器关闭连接

服务器关闭连接时, 客户端会产生 close 事件

```
Sub SC_close ()
```

```
    SC.close '关闭当前连接
```

```
    SC.listen '重新开始侦听
```

```
End Sub
```

5) 主动关闭连接

```
Sub Form1_Unload (Cancel as integer)
```

```
    Sc.close
```

```
End Sub
```

软件程序流程如图 1 所示:

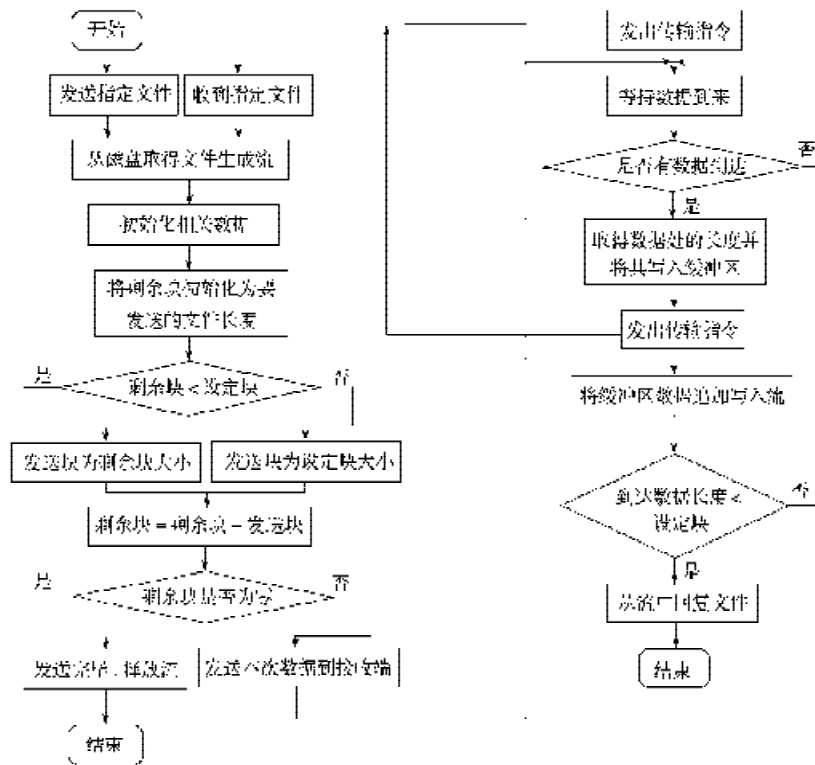


图1 软件程序流程图

Fig. 1 Software routine flow chart

参考文献

3 结语

基于 VB WinSock 控件的文件传输实现灵活, 只需配置 TCP / IP 协议。除了能传输文件以外, 还能进行文本文件的传输从而用于网络聊天程序的实现。由于其传输的实现是通过 buffer 传送的, 因此, 对于大文件分配一个足够大的 buffer 是不现实的, 通过将大文件分割到宿主机后再合并可以较好地实现大文件的传输问题。在文件传输系统中, 利用 Winsock 实现了大文件在 Client 和 Server 间的无差错快速传输, 通过修改发送缓冲区和接收缓冲区的大小以及采用多线程技术, 解决了利用 Winsock 发送和接收大数据量文件的难题, 在此基础上还可以实现一对多或多对多的文件传输。

- [1] 赵双明. C++环境下的Internet网络编程[J]. 计算机应用, 2000, 10(11): 109-111.
- [2] 尹浩琼, 李 剑. TCP/IP 详解 [M]. 2版. 北京: 电子工业出版社, 2003.
- [3] 吴 易, 王 凌. Java 技术在P2P环境下的应用[J]. 微计算机信息, 2005, 21(3): 21-23.
- [4] 张维承, 王 勇, 陈抗生. 原始套接字编程在嵌入式 Internet 通信协议中的应用[J]. 计算机应用研究, 2002, 19(10): 37-38.
- [5] Gary R Wright, W Richard Stevens. TCP/IP Illustrated[J]. The Implementation Pearson Education Asia Ltd., 2002, 1(12): 232-233.
- [6] Douglas E Comer. Computer Networks and Internets[J]. Printice Hall, 1998, 29(30): 202-205.

(责任编辑: 罗立宇)