第 21 卷 第 4 期 华侨大学学报(自然科学版) Vol. 21 No. 4
2000 年 10 月 Journal of Huaqiao University (Natural Science) Oct. 2000

文章编号 1000-5013(2000)04-419-06

计算机网络对等安全通信技术的研究

吴 金 龙

(华侨大学计算机科学系, 泉州 362011)

摘要 论述在计算机局域网上实现对等安全通信的 3 种技术, 以及在因特网环境下, 利用 TCP/IP 协议实现以节点之间信息交换为主要特征的对等安全通信模型. 介绍利用不同协议, 实现对等安全通信的设计思想和技术. 就数据传输过程中的安全性等方面的问题进行讨论, 文中给出部分源程序代码.

关键词 计算机网络, 安全通信, 网络协议, 客户机/服务器

中图分类号 TP 393.08 : TN 915 文献标识码 A

在计算机网络通信中, 大多数信息传送是以服务器为主导的中转通信. 为减轻服务器的工作负担和提高通信效率, 有必要利用网络协议及其软件所提供的编程接口, 实现网络节点之间的同步传送和对等安全通信机制, 以保证网络环境下的通信安全.

1 局域网对等安全通信技术

在局域网上实现对等安全通信的技术有多种, 下面主要讨论 3 种.

第一种技术是用户通过计算机, 直接对网卡 NIC(Network Interface Controllor) 的各种寄存器进行编程控制, 完成数据分组的正确发送和接收. 例如 NE2000 是 NOVELL 公司生产的 16 位适合 ISA 总线的网卡. 该网卡按功能可分为接口电路、缓冲 RAM、站地址 PROM、自举 ROM、状态设置跳线器、连接器、DP8390、DP8391 及 DP8392 等 9 个部分. 其中 DP8390 是网络接口控制电路 NIC, 它全面支持 IEEE802.3 标准; 内部包含实现微机对网卡的收发控制、状态检测及收发数据缓冲用的寄存器堆; 另有对网卡上缓冲 RAM 的读写控制逻辑. 用户对 NE2000 网卡通信过程的控制主要是对各种命令寄存器编程控制的. 这种技术不要求任何网络软件的支持, 应用开销小, 实时性好, 但编程针对性强、移植性较差[1].

第二种技术是利用 IBM 的 NETBIOS(网络基本输入输出系统) 或 NETBIOS 仿真提供的数据服务来实现实时对等通信. NETBIOS 是一个网络编程接口, 位于会话层与表示层之间, 接口程序与较低层的各种活动隔离. NETBIOS LAN 适配器在网络上是由一个或多个网络名来区分的, 每一个网络名字由 16 个字符组成. 名字的第 1 个字符不能为 2 进制 0 或星号*,

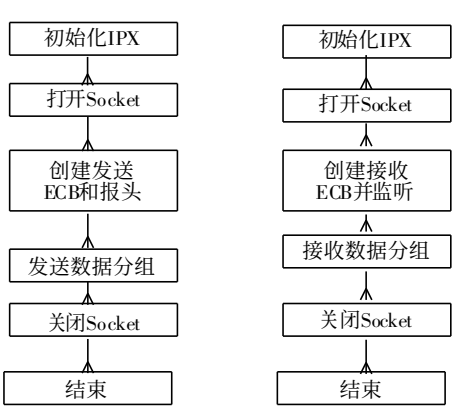
收稿日期 2000-03-31 作者简介 吴金龙(1943-), 男, 副教授

基金项目 福建省自然科学基金资助项目

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved. http://w

第16个字符由系统保留. 名字类型有Name_Claim或者Add_Group_Name_Claim,它表示一个网络名或一个组名. Windows XX中的工作组名与NETBIOS的组名相对应,通过"网上邻居"可见到这些名字. NETBIOS本身无路由功能,但它可利用TCP/IP的路由特性,将NETBIOS名解释成IP地址,实现访问远程机器的目的. 但是,通过NETBIOS连接Internet是不可能的,这由它的名字单一机制所决定. 用NETBIOS进行通信时先由应用程序发出监听命令访问名字表中的某个名字,以便建立虚电路,实现发送和接收数据. 这是面向连接的一种通信方式. 此外,NETBIOS还可以用广播数据报或普通数据报的型式,实现无连接的通信. 当然,这种通信不提供差错控制,效率较低. 由此可见,NETBIOS是一个应用较广泛的标准接口,主要优点是编程方便,可移植性强. 主要缺点是网络繁忙时,错误较多.

第三种技术是利用微机网络提供的支持实时通信的编程接口,建立用户的实时对等安全通信环境[2]. NOVELL网络的操作系统NetWare为用户提供丰富的服务支持. 其中IPX协议提供网络层数据报接口,使应用程序可以调用IPX与网上其它工作站、服务器设备相连接,实现发送和接收数据报的功能. IPX数据报由包头及数据两部分组成. 30字节的包头由检验和,分组长度,传输控制,分组类型和目的网络号、节点地址、套接口号以及源网络号、节点地址、套接口号组成. 数据部分不能超过546个字节. IPX协议地址由网络号、工作站节点地址及应用程序使用的套接口号Socket组成. 套接口是IPX的特有概念,它相当于发送或接收数据的逻辑通信通道. IPX利用事件控制块ECB结构来实现发送、侦听及调度的管理任务. 它可分为发送ECB及接收ECB,结构相同,所填写的内容不同. IPX按FIFO(先入先出)的规则对ECB队列进行管理,完成通信任务. IPX的主动服务,如发送数据包、打开或关闭套接口、取消事件等,由用户程序调用7AH中断的指定功能完成;异步通信中断对接收数据的实时后处理则由事件服务例程ESR的调用来完成. ESR通过ECB中的ESRHandler项挂接在IPX事件上. ESR是应用程序定义的一个例程. 在某一事件发生后由IPX(0BH号中断)或异步事件服务AES(08H号中断)调用. 该事件可以是一次发送完毕,一次侦听结束或一个重新调度自身的IPX事件等. 由此可见,利用IPX在两个工作站之间实现对等通信,其发送和接收工作站的工作过程如图1所示.



(a) 发送方 (b) 接收方

图1 利用IPX协议实现对等通信的过程

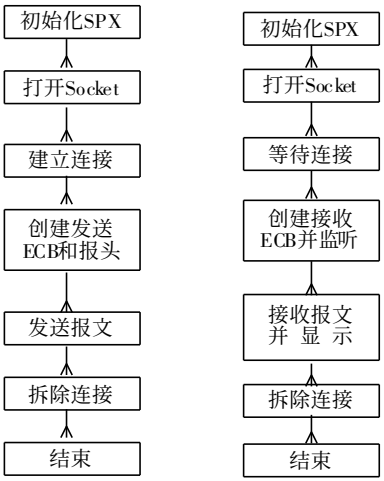
初始化IPX的工作是调用IPX_installed()函数、检查IPX是否存在;返回调用IPX功能和函数的指针IPX_SPX;打开套接口号Socket,发送方获取接收方的网络地址(网络号、节点地址及套接口号). 创建ECB和报头的主要任务是分配空间、初始化ECB、调用功能函数IPXSendPacket和IPXListenForPacket进行发送和接收数据. 最后关闭Socket,完成通信过程. 显然,利用IPX通信协议,无需建立连接拆除,通信速度快、效率高,但不保证按序传输.

与IPX紧密相连的是SPX协议,它是传输层的数据报接口,提供分组排序和重传的功能,保证每个分组都能按序可靠地到达目的地. SPX协议发送和接收数据分组的工作过程如图2

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://w

所示.

由图可见, SPX 与 IPX 协议的工作过程, 除了建立连接、双向传输信息和拆除连接以外, 其它细节都很相似. SPX 包头长为 42 字节, 数据区为 0–534 字节. 其中比 IPX 增加的 12 个字节包头包括连接控制、数据流类型、源连接 ID 号、目的连接 ID 号、顺序号、确认号和分配号等字段, 均围绕按序发送和可靠接收的目的而设置的. 例如, 顺序号指明发送报文的顺序; 确认号则指明接收方的应答信息, 希望接收的分组序号; 分配号指明可用的缓冲区数量, 仅当顺序号等于远程分配号时, SPX 才可能发送数据分组, 起到端对端的流控制作用. SPX 在传送前一定要获得对方的网络地址及 ID 识别号, 而不能象 IPX 对全体工作站进行广播. SPX 只有在两个工作站建立起连接后, 发送和接收才有可能, 而且任一台工作站都能发送或接收, 使传送具有双向性. 当工作站检测到用户按了 Escape 键时, 则将它传送给对方, 使其释放 SendECB, 并进而调用 SPXTerminateConnection( ) 函数拆除连接并结束通信. 为了得到对方的确认和正确发送的保证, SPX 将包排序, 保证按序传送; 对传送中的错误要进行检测和纠正, 对重复的包要丢弃, 对丢失的包要重发. 显然, SPX 放弃了 IPX 的高速度传送性能才获得了可靠安全传送的特性.

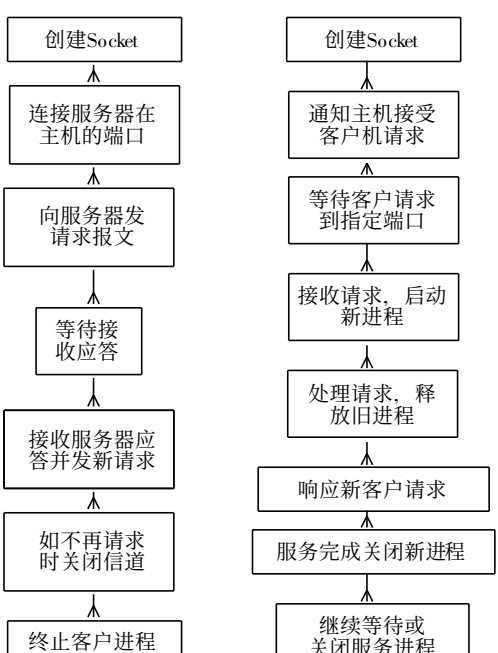


(a) 发送方　　(b) 接收方

图2　利用SPX实现对等通信过程

2　互联网对等安全通信技术

目前, 支持互联网上的 TCP/IP 协议的 WinSock 已集成在 Windows xx 及 Windows NT 中, 利用 Socket 进行通信可以采用两种形式. 第一种是面向连接的流方式(Stream Socket), 两个通信的应用程序先要建立一种虚拟的连接关系, 在数据传输过程中使用连接号, 或者说套接字 Socket 本质上就是连接号组成的管道; 虽然数据分组不带目的地址, 但收发数据不但顺序一致, 而且内容相同. 显然流方式采用 TCP 协议, 它保证通信按序发送, 可靠到达, 对数据采用重发和校验机制, 适合于数据文件的安全传送. 第二种方式是无连接的数据报方式(Datagram Socket). 每个数据分组要带上完整的目的地址, 如同邮件一样, 它不保证数据在传送过程中是否会丢失, 也不保证传输顺序或数据的正确. 显然, 数据报使用 UDP 协议.



(a) 客户机一方　　(b) 服务器一方

图3　客户机/服务器通信进程

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://w

在网络环境下,采用客户机/服务器模式的通信方式,其过程如图3所示.

由此可见,TCP/IP应用中的服务器进程必须先于客户机进程启动,直到对客户机的响应结束或被迫终结.其程序流程图如图4所示.

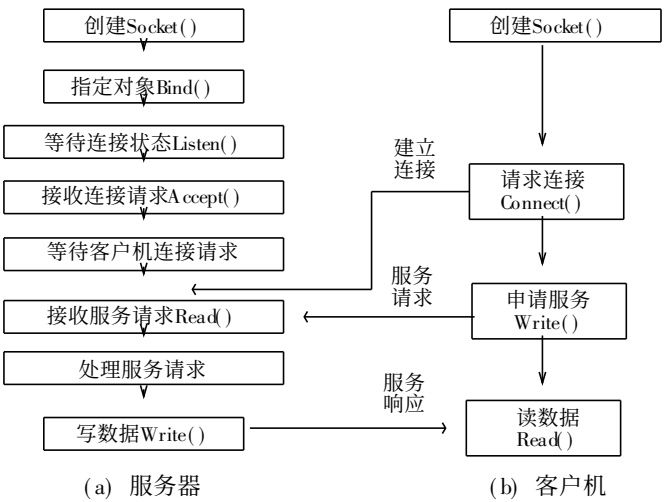


图4 面向连接的程序流程图

WinSock提供的通信函数有100多个,但流程图中只列出主要的几个.有些函数,如寻求连接Connect()和接收连接Accept()等,在执行过程中,可能由于操作系统忙于其它事务或通信信道被别的进程占领而导致无法返回,为使该进程避免休眠态存留在内存中,处于阻塞的函数会不断调用系统函数GetMessage()以保持消息循环的正常进行.为了实现非阻塞通信,WinSock提供异步选择函数WSAAsynSelect(),由该函数注册某些网络事件,如接收缓冲区满,允许发送数据,请求连接等事件发生时,应用程序可收到相应的信息.应用程序在使用Windows Sockets DLL之前必须先调用启动函数WSAStartup();当应用程序终止时,则必须调用WSACleanup()函数将自己从DLL中注销.如果当前线程的操作有误,则调用WSAGetLastError()函数,就可在调用失败后返回错误代码.

基于TCP/IP协议和WinSock编程接口的对等通信程序可以利用互联网上最流行的语言Java进行设计.Java提供了强大的网络支持机制,Java的API(应用程序接口)中以面向对象的类形式提供了两个不同层次的网络支持机制.第一种机制是用URL(统一资源定位器)访问网络资源的类库.第二种机制是利用Socket实现客户机/服务器通信模式的类库.Java通过采用面向对象的方法,为用户提供了与平台无关的使用接口[5].

虽然客户机和服务器之间的通信组件是各种各样的,但它们一般由一个地址加上端口号来标识,每个地址最多可存65535个端口,每个服务器程序都在某个端口上提供服务.当客户机想使用该应用程序时,就要连接到对应的端口号.通常0～1024为网络系统的保留端口,1024以上的端口号可供程序员选用.我们利用4466作为主要通信端口(命令端口)进行编程,当要进行数据传送时就动态地占用4467作为数据传送端口.下面列出我们编程实现的Java to Java Copy的文件传送程序中由CLIENT发送到SERVER的源程序代码.

```
public void sendFile(File filename){
  System.out.println("in sendFile");
  byte[]buffer= new byte[1024];
  int bytes__read;
  try{
    System.out.println("connecting data socket");
    Socket sendSocket=
```

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://w

```
        new Socket(cm1Socket.getInetAddress(), 4467);
        Data OutputStream out=
        new Data OutputStream(sendSocket.getOutputStream());
        FileInputStream fin= new FileInputStream(filename);
        while(true){
          bytes__read= fin. read(buffer);
          System.out.println(bytes__read);
          if(bytes__read< = 0 ‖ bytes__read= = - 1){
          break;
          }
          out.write(buffer, 0, bytes__read);
    }
      out.close();
      fin.close();
    }catch(IOException e){
      System.out.println("Error in SendFile!"+ e);
    }System.out.println("OK");
}
```

同样的道理, 由 CLIENT 中接收从 SERVER 送来的文件的源程序代码与上述模块仅有个别语句的差别. 由于 Java 的多线程机制, 一个进程可以运行多个线程并发地执行多件事件. 因此, SERVER 可用一个线程执行图形用户界面 GUI, 第二个线程进行监测用户连接情况, 第三个线程是第一个用户连接产生的用户服务线程, 第四个线程是下一个有效用户连接生成的用户服务线程…. 例如, 生成一个监听线程并进行监听的模块, 其程序的源代码如下:

```
    public JJCPServer(int port, Server Ser){
        try{
          listen__socket= new ServerSocket(port);
        }
        catch(IOException e){
          System.err.println("Exception creating server socket");
        }
        this.start();
    }
```

由于 TCP/IP 协议在服务器方等待客户方的连接请求时有可能产生阻塞的情况. 因此, 创建该线程的目的就是不断监测请求连接活动, 以保证多用户进行连接的可能性. 当有用户请求连接成功后, 就产生新的 Connection 用户服务线程, 为用户提供服务. 每个用户对应一个 Connection 服务线程, 以便使多用户共享文件的愿望得到实现.

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://w

3 数据安全性

在网络通信过程中,主要有链路加密和端对端加密. 端对端加密要求被保护的数据在中间节点不以明文的形式出现. 若在传输层以下进行加密,则网络层中数据的分组头将是密文,通信子网不能依照密文进行有效传送. 因此,要保证数据传送的安全可靠性,一定要在传输层以上各层中实现. 据此,IPX/SPX 和 TCP/IP 协议中的 IPX 和 IP 协议属于网络层,显然对数据的安全传送无法保证. NETBIOS 是一个位于会话层和表示层之间的编程接口,如果采用面向连接的通信方式,数据传送的安全性是有保证的. SPX 和 TCP 协议同属传输层,它们能保证通信的数据分组按序发送,或保证采用重发和校验的机制进行纠错. 相对而言,这些协议的编程接口是有安全保证的.

但是,网络上的通信是很难保证绝对安全的. 例如,NETBIOS 使用网络卡的永久性节点名表示通信工作站,如果有人"冒充"该名以获取授权,并进行窃听活动,数据的安全性将受到威胁. 为保证安全,支持软件要保证在一个 LAN 中节点的永久名与本地管理 ID 是唯一的.

同样,TCP/IP 的安全性也有问题. 当有人盗用别人的 IP 地址时,不仅数据传送不安全,连计费系统也会受到破坏. 最好的办法是使 IP 地址和网络卡地址保持一致性. 然而随着网络通信的发展,各种安全技术的研究也在不断深入,诸如防火墙技术、路由器加密、网络地址转换、身份验证、数据加密解密、数字签名、数据完整性控制等,这些技术的开发和应用又与网络协议紧密相关. 我们研制的 JJCP 系统,正是基于 TCP/IP 协议,利用 Java 编程的数据通信加密解密集成系统. 经过多次实验表明,这套系统实现工作站间的文件传送,安全性好,操作简单,界面友好,具有广阔的应用前景.

参 考 文 献

1 李 鹏编. 计算机通信技术及其程序设计[M]. 西安:西安电子科技大学出版社,1998. 104～117, 135～149

2 李继敏. NOVELL NetWare 386 深开发环境技术[M]. 北京:学苑出版社, 1993. 47～61

3 吴金龙,吕吉实. 利用 JAVA 编程实现网络安全通信[J]. 华侨大学学报(自然科学版), 1999, 20(3): 312～316

A Study on Techniques for Realizing Peer to Peer Safety Communication on Computer Network

Wu Jinlong

(Dept. of Comput. Sci., Huaqiao Univ., 362011, Quanzhou)

Abstract Three techniques are presented for realizing peer-to-peer safety communication of computer local network. And a model of peer-to-peer safety communication with internodal information exchange as primary characterictic is realized in Internet environment by applying TCP/IP protocol. The design idea and technique for realizing safety communication by applying different protocols are presented. The safety and other problems in data transmission are discussed, and part of source program codes are given.

Keywords computer network, safety communication, network protocol, client/server

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://w