

文章编号:1005-8456(2004)08-0029-03

驼峰信号微机监测系统远程监控功能软件设计

廖文江,蒋大明

北京交通大学 电子信息工程学院,北京 100044

摘要:设计驼峰信号微机监测远程监制系统的体系结构,简要阐明远程控制软件工作模式与原理。根据实际要求设计出远程操作和控制实现方法,同时给出了确保系统有效安全策略及实现方法。

关键词:远程监控;套接字;客户/服务器模型;驼峰

中图分类号 U284.64

文献标识码 B

Design of remote monitoring program in Hump Signal

Microcomputer Detecting System

LIAO Wen-jiang, JIANG Da-ming

(School of Electronics and Information Engineering of Beijing Jiaotong University, Beijing 100044, China)

Abstract: It was designed remote monitoring and control system in Hump Signal Microcomputer Detecting System. and control. The work model and principles of remote operation system were simply described. The methods of remote operation and control were designed to satisfy with the practical situation and the security strategy. Meanwhile, the effective security strategies were given to ensure the system safety.

Key words: remote monitoring control; socket; C/S model; hump

网络技术正在铁路系统中发挥更大的作用。在现阶段铁路交通运输状况下,铁路工作人员需要了解距离很远的某一站场各项电气设备的情况,从而能够掌握此站场的动态。这一问题的解决主要靠网络监控来实现,但是通常网络监控设计实现大都是基于代理服务器的软件开发,它仅能完成一些简单的功能而且系统消耗较大,安全性较低。根据铁路站场驼峰信号微机监测系统的远程监控功能的实际需要,我们采用VC6.0开发,能够方便、实时和动态的对网络中各站场的现场测控计算机进行屏幕监视、控制与管理,从而使铁路的工作人员能够及时地了解到铁路运输网络中的各个站场的情况,更好地完成铁路运输工作。

1 远程监控系统结构设计

铁路驼峰调车场电气设备当前运行状态数据是由下位机采集到的模拟开关量,再通过串口传送到上位机(现场测控计算机)。现场测控计算机(服务器端)能够监测到站场各项设备的运行情况。以太网另一端的铁路工作人员(远程客户端)通过远

程监控软件也能够监测到此站场的各项设备运行情况。下面简要说明远程监控系统的各个组成部分及其功能:

1.1 现场测控计算机

现场测控计算机主要有2个功能:

1) 站场状态的实时监测,使操作人员直观明了地观察站场的状态;

2) 作为驼峰信号微机监测系统远程监控的服务器端,接收远程客户的操作指令并对指令进行译执行,将结果反馈到远程客户端。



图1 远程监控系统服务器端模型

1.2 数据库服务器

主要用作为微机监测系统采集数据的存储数据库服务器,有记录远程操作日志、系统安全数据备份等功能。

1.3 远程客户计算机

通过远程监控软件,对远端现场测控计算机的屏幕监控和对该站场的驼峰信号微机监测系统进行

远程动态调整。有以下几个功能：

- 1) 用户登录管理，以确保网络监控安全性；
- 2) 对站场现场测控计算机屏幕监视；
- 3) 远程控制，可以在客户端的计算机上直接操控某站场现场测控计算机；
- 4) 实时消息发布，可以和现场测控计算机端的操作者进行信息交流；
- 5) 操作日志记录及审查。

2 远程监控功能实现的关键技术

2.1 远程控制原理

远程控制通常采用 Client/Server 模式。客户端安装远程监控软件，发送用户身份信息到服务器端请求连接。服务器端程序进行客户身份验证。如果通过，便与客户端建立网络通信连接。客户端向服务器端计算机发送各种操控指令和消息。服务器端接收到客户端指令后，在服务器端执行相应的操作，并将执行后的结果回传到客户端计算机。

2.2 基于WinSock的网间通信模型

WinSock 是当前 Windows 平台上网络应用编程的标准接口。驼峰信号微机监测远程监控系统的客户端和服务器端间的网络通信就是建立在 Socket 的基础之上的。采用可靠的、面向连接的流式套接字，可以很容易地实现客户和服务器间的网络通信。

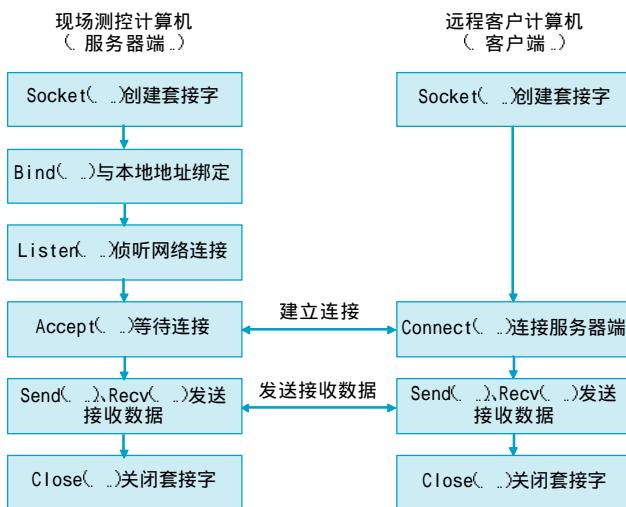


图 2 流式套接字流程图

2.3 屏幕监控和鼠标键盘模拟技术

客户端设置 0.5 s 定时中断，由客户端的定时中

断处理程序发出服务器抓屏指令，服务器端在接收到抓屏指令后获取当前屏幕数据，再经压缩后传递到客户端。在客户端先解压缩文件，再显示于监控程序相应的控件里，这样远程客户端用户就能通过监看驼峰现场测控计算机当前的屏幕内容来了解驼峰站场的各项设备的运行情况。此外，有时需要在客户端对某一站场的驼峰微机监测系统进行远程动态调整，这就要求服务器端能够实时模拟客户端的鼠标键盘操作。在客户端获得鼠标键盘的消息等数据后发送到服务器端。在服务器端，根据数据报文区分各种鼠标键盘事件，在相应的位置产生鼠标键盘事件，从而能够模拟客户端的远程操作，实现对监测系统的远程动态调整。

2.4 远程监控功能实现流程

下面具体分析一下如何在客户端实现对服务器端屏幕监视和服务器端的鼠标键盘模拟功能。

2.4.1 服务器端程序

- 1) 驼峰微机监测系统启动服务器端远程服务，设置客户端访问端口为 4086。

```

ServerSock = socket(AF_INET, SOCK_STREAM, 0);
dstclient_addr.sin_family = PF_INET;
dstclient_addr.sin_port = htons(4086);
dstclient_addr.sin_addr.s_addr = INADDR_ANY; // 可接受任何远程套接字

```

```

bind(ServerSock, (struct sockaddr far *)&dstclient_addr,
sizeof(dstclient_addr));

```

- 2) 当有客户端请求连接到来时，接受请求连接为 NewSock。

```

NewSock = accept(ServerSock, (struct sockaddr far *)
&dstclient_addr, &len);

```

- 3) 向客户端发送通信握手标志 Flag 和接收客户端消息 iMsg，根据不同消息进行相应处理。

```

send(NewSock, (char *)&Flag, sizeof(Flag)+1,
MSG_OOB);

```

```

recv(NewSock, (char *)&iMsg, sizeof(iMsg)+1, 0) switch
(iMsg)

```

```

{ case US_DESKTOPBIT: // 抓屏处理
    SnapScreen(); // 抓取服务器当前屏幕
    SendScreen(); // 发送屏幕数据到客户端
    break;
    case US_MOUSEBTNDOWN: // 鼠标事件处理
    MouseEvent(point); // 产生鼠标事件
    break;
}

```

4)若接收到鼠标事件消息则同时接收鼠标点击位置数据报文。

```
recv(NewSock,(char*)&x,sizeof(x)+1,0);
recv(NewSock,(char*)&y,sizeof(y)+1,0);
point.x=x; point.y=y;
```

5)关闭远程监控服务，服务器端关闭本地套接字NewSock。

2.4.2 客户端程序

1)启动客户端的远程监控程序，指定登陆服务器的IP地址和访问端口4086，调用SetTimer()函数产生0.5 s定时中断，中断触发抓屏消息DibEvent(hWnd)。

```
dwIPAddr=inet_addr(m_csIP); //服务器IP地址
m_hSocket=socket(PF_INET,SOCK_STREAM,0);
sockAddr.sin_family=AF_INET;
sockAddr.sin_port=htons(4086);
sockAddr.sin_addr.S_un.S_addr=dwIPAddr.
```

2)调用函数connect()发送连接请求。

```
connect(m_hSocket,(LPSOCKADDR)&sockAddr,
sizeof(sockAddr)).
```

3)监控窗口捕获消息，进行相应消息处理。

```
switch(message)
{...}
```

case US_MOUSEBTNDOWN:// 鼠标右键按下抬起消息

```
if (MouseEventFlag)//是否可以处理鼠标事件
MouseEvent(US_MOUSEBTNDOWN,hWnd);}
break;
```

case WM_TIMER:// 定时中断触发实时抓屏消息
DibEvent(hWnd); // 实时抓屏消息

4)接收通信握手标志Flag，发送鼠标键盘或抓屏消息。若此时正在处理抓屏程序则鼠标消息阻塞，抓屏程序处理结束，方可发送鼠标键盘消息到服务器端。

```
recv(kSocket,(char*)&Flag,sizeof(Flag)+1,0)
if (Flag == US_FLAG)
```

```
{send(kSocket,(char*)&MouseMsg,sizeof(MouseMsg)+1,MSG_OOB);
SendMousePoint(hWnd);}
```

5)接收服务器端屏幕数据，并显示于客户端窗口中。

```
recv(kSocket,(char*)&Flag,sizeof(Flag)+1,0)
if (Flag == US_FLAG)
{MouseEventFlag = false; //阻塞鼠标事件消息
```

```
iMsg = US_DESKTOPBIT;
send(kSocket,(char*)&iMsg,sizeof(iMsg)+1,
MSG_OOB);
GetScreen(hWnd); // 获得服务器屏幕数据
DisplayScreen(); //显示服务器屏幕内容
MouseEventFlag = true;} //鼠标事件消息可用
6)客户端退出监控程序，关闭kSocket。
```

3 远程监控的安全问题

在驼峰信号微机监测系统远程监控系统的网络安全是以现场测控信息和远程操作信息在网络上传输时受到机密性、完整性和真实性的保护为目的，同时避免他人利用窃听、冒充、篡改、抵赖等手段侵犯操作人员的利益。从以下几个方面来实现系统远程控制的安全：

1)身份认证及访问权限控制，只有身份确认后，方可进行远程操控。同时，对不同类型的用户设置不同的操作权限；

2)采用DES加密算法对网络传输中的数据进行加密和解密，确保信息安全；

3)操作日志记录客户计算机远程操作信息；

4)系统恢复功能，能够做到根据日志记录和系统数据备份恢复到远程操作前的系统状态。

4 结束语

文中设计的微机监测的远程监控系统，在WindowsNT4.0下采用Visual C++6.0开发完成，实现了实时监控网络中任一现场测控计算机的运行情况，并且能根据实际需要对现场测控计算机进行状态调整。已用于现场效果明显，具有一定的参考价值。

参考文献：

- [1] Peter Norton.Rob McGregor. MFC开发Windows95 /NT4应用程序[M].孙凤英.北京：清华大学出版社，1998.
- [2] 汪蒲阳.因特网应用编程[M].北京：清华大学出版社，2000.
- [3] 蒋东兴，林鄂华，陈棋德，等. Windows Sockets 网络程序设计大全[M].北京：清华大学出版社，1999.
- [4] 龚 俭.陆 晟.王 倩. 计算机网络安全导论[M].南京：东南大学出版社，2000.