

文章编号: 1672-4291(2004)Sup.-0177-04

局域网上动态视频实时传输技术的研究

景 井, 晁爱农

(空军工程大学 工程学院, 陕西 西安 710038)

摘 要:介绍了一种在局域网上实时传输视频的策略,并给出实现方法,该方法可成功地应用于远程视频集散监控系统中。

关键词:局域网; 动态视频; 实时传输

中图分类号: TP393.1 **文献标识码:** A

Research on real-time transmitting of active video over LAN

JING Jing, CHAO Ai-nong

(Engineering College, Air Force Engineering University, Xi'an 710038, Shaanxi, China)

Abstract: A kind of tactic of real-time transmitting dynamic video over LAN is introduced, and the method of realizing is provided, which is applied on the remote video distributing centre monitoring and controlling system.

Key words: LAN; Dynamic video; Real-time transmitting

随着计算机网络和通信技术的发展,监控系统将朝着网络化、智能化的方向发展。在我们研究开发的远程视频集散监控系统中,二级分控主机(作为服务器)与各现场控制主机服务器之间(作为客户机)是利用 TCP/IP 协议实现视频实时传输。TCP/IP 是使用最广泛的协议族,研究在此协议上的动态视频实时传输具有十分重要的意义。

1 套接字关键技术

1.1 基于 WindowSocket 套接字的种类及选择

首先 Windows 针对网络套接字提供两个类支持,即 CAsyncSocket 和 Csocket。CAsyncSocket(异步套接字)是 Csocket 的基类,提供了数据报和面向流两种方案。异步套接字属于低层套接字,由于它提供了更大的灵活性,在大数据量的多媒体通信中经常采用。Csocket 在 CAsyncSocket 的基础上封装了堵塞的功能。“堵塞”保证了数据在网络上可靠稳定的传输,给一般的编程人员提供了很大的方便。但是在进行堵塞调用的时候,线程将被堵塞,同时其他的工作也不得不停下来等待堵塞调用返回。整个进程不能充分利用 CPU 时间。其次,像视频信息,其特点

是数据量大,监控要求实时性很高,在某些情况下,为了盲目追求数据传输的可靠性而放弃实时性,往往得不偿失。当然,最理想的方法就是在可靠性和实时性之间寻求平衡点,使系统最大限度地使用 CPU,发挥效能。采用的方案是:以 CAsyncSocket 为基准,构造一个派生类,在应用层保证数据的可靠性。

1.2 视频网络传输系统流程

基于 TCP/IP 协议的动态视频传输是在 C/S 结构环境下讨论的,运行于典型的 10 Mbps 以太网之下,此类局域网已经很成熟并且得到普遍应用,当然局域网有更多的结构,它可以很复杂,但对于套接字编程来说,这一切是完全透明的。客户机/服务器数据交换流程,表面上看和经典的面向流的套接字非常相似,但是具体实现截然不同,此处的套接字请求连接、接收连接请求、发送数据、数据应答等都是通过自定义的帧类型来完成。

1.3 套接字的实现方法

1.3.1 客户机/服务器的帧类别、网络状态的定义

我们使用的是面向数据报的套接字,在应用层提供可靠性的保证。考虑在数据报的基础上再提供一

收稿日期: 2004-03-10

作者简介: 景井(1981—),女,黑龙江哈尔滨人,空军工程大学硕士研究生。

层封装,包括帧头标志、帧类别、图像格式、滑动窗口大小、帧号、数据长度(有效数据)、数据区.并定义了几种类型的帧,此外还明确定义了帧结构,如图 1 所示.

```
//帧类别定义
#define frmData 0 //数据帧
#define frmDataAnswer 1 //数据应答帧
#define frmPolling 2 //检测帧
#define frmPIAnswer 3 //检测应答帧
#define frmAbort 4 //异常中止帧
#define frmSysError 5 //系统错误帧
#define frmNormalClose 6 //正常关闭帧
```

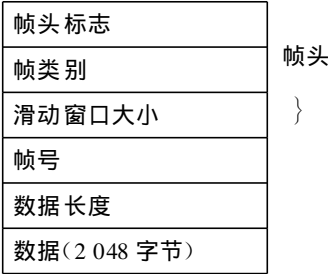


图 1 帧结构

这个帧结构通用于所有的图像类型:128 * 96 格式的图像,每次传输 128 个字节;176 * 144 格式的图像,每次传输 512 个字节;352 * 288 格式的图像,每次传输 2 048 个字节.这样每种视频格式分别定义最大网络传输单元的目的是为了实时的传输和显示.因为每种格式的视频在单位时间内生成的数据量大小是不同的,而且是成倍的关系,如果三种视频格式的最大传输单元大小相同,相对小的图像很难填满缓冲区,这样会失去服务端显示的实时性;而对于相对大的图像,由于数据量很大,网络发送的频率很高,容易引起堵塞和缓冲区填满的错误状态.

同时还定义了网络套接字的 10 个状态,客户端和服务器端的套接字都在这 10 个状态之间转换.其中初始化状态、接收数据状态进行的速度很快,终端程序很难显示出来,但这些状态在程序运行过程中是存在的.各种状态如下所示:

```
#define Vedio-STAT-NULL 0 //断开
#define Vedio-STAT-INIT 1 //初始化状态
#define Vedio-STAT-COMM 2 //通信状态
#define Vedio-STAT-STOP 3 //堵塞状态
#define Vedio-STAT-DATA 4 //发送数据状态
#define Vedio-STAT-REC 5 //接收数据状态
#define Vedio-STAT-POLLING 6 //检测状态
#define Vedio-STAT-ABORT 7 //中断状态(没有
```

```
使用)
#define Vedio-STAT-SYSERROR 8 //系统错误状态
#define Vedio-STAT-CLOSE 9 //正常关闭状态
#define Vedio-STAT-WAITFORDATA 10 //等待接收数据
```

1.3.2 客户端套接字的定义及实现 在客户端要求客户能主动向服务器发送连接请求;在接收到连接请求应答帧之后,保存对方套接字的地址和端口号,系统处于通信状态;如果有数据,并且系统处于通信状态先置客户端套接字为发送数据状态,套接字向网络发送数据帧,然后置客户端套接字为堵塞状态,这时,缓冲区中即使有数据也不能发送,必须等待对方的应答,这样就避免了大量数据淹没网络的发生;如果客户端套接字接收到服务器的数据应答帧,客户端套接字恢复到通信状态如果在 30 s 之内接收不到回应则确定网络中断,关闭套接字重新连接.这样客户端套接字处于一个可控的状态转换之中不会引起系统故障.

1.3.3 服务器端的套接字的定义及实现 在服务器端,创建主应用程序时创建侦听套接字,系统处于等待连接状态;当有客户端连接请求到时,侦听套接字创建接收套接字来准备接收数据,如果侦听套接字创建成功,发送检测应答帧,服务器接收套接字处于等待接收数据状态;当有数据到,服务器接收套接字处于接收数据状态,服务器端接收套接字验证帧头标志、帧长度、检查帧类型.如果是数据帧就将数据存入缓冲区,向客户端套接字发送数据应答帧,服务器接收套接字处于通信状态;如果服务器接受管理人员的关闭命令,服务器端将发送关闭帧,自动关闭套接字.服务端的网络套接字的状态同样处于有限的几个可控的状态之下.

1.4 网络套接字实现的特点

CasyncSocket 类不提供堵塞的功能,其中有利有弊.为了在应用层保证网络套接字双方的可靠性,上面的很多描述都是针对这个目标而进行的.这种方法与面向连接的流套接字有本质的区别,数据报数据发送以后,无法保证对方一定收到,要可靠传送,对方必须对发送信号表示确认,这种功能数据报套接字是无法实现的,必须由应用层自己完成.而面向连接的流套接字在该层保证数据的有序可靠发送时,在调用堵塞函数时系统效率会下降,若实时性要求很高时,往往可以牺牲一些网络传输的可靠性(容许丢失很少的几帧),而且这种要求在面向流的套接字中是无法实现的,通过发送检测帧来代替流套接

字的 Connect 函数, 发送完检测帧后可以做其他任何事, 直到收到服务器端的检测应答帧且中间没有任何的堵塞. 如果使用 Connect 函数, 由于它是一个堵塞函数, 在对方应答之前调用函数的进程将被堵塞, 除非连接超时. 用定时器实现了超时处理. 滑动窗口实现的具体策略是要能准确判断客户端、服务器端连接状态; 要保证数据的正确传输, 以至于服务器端能正确解压出视频信号; 保证视频信号的实时传输. 在能保证解压质量图像的前提下, 使传输延时最小. 在偶尔丢失帧的情况下, 服务器端不要求重发该帧, 而是让客户端发送当前的视频数据, 以保证视频的实时传输.

为此, 经过大量测试, 发现由于压缩算法的特殊性, 在传输过程中, 丢掉的数据在传输总数据三分之一范围以内, 客户端能正确解压出来, 以此确定了传输质量, 确定了数据丢失的最大上限; 如果传输过程中, 数据丢失率在 $1/3$ 之内, 就认为传输链路正常, 数据丢失率等于或大于 $2/3$, 就认为传输链路堵塞. 这样传输链路的可靠性、实时性就会增强.

2 IP 组播技术

2.1 单播、组播、广播技术

组播技术被认为是 WWW 技术推广之后出现的最激动人心的网络技术之一. 组播是一种允许一个或多个发送者发送单一的数据包到多个接收者的网络技术. 组播源把数据包发送到特定组播组, 而只有属于该组播组的地址才能接收到数据包. 组播可以节省网络带宽, 因为无论有多少个目标地址, 在整个网络的任何一条链路上只传送单一的数据包. 基于三种通讯方式的网络结构和数据传递过程的工作方式可以看出.

数据信道: 如果一台主机同时给很少量的接收者传输数据, 一般没有什么问题. 如果有大量主机希望获得数据包的同一份拷贝时却很难实现. 这将导致发送者负担沉重、延迟长、网络拥塞. 为保证一定的服务质量需增加硬件和带宽.

组播 (Multicast) 传输: 它提高了数据传送效率, 减少了主干网出现拥塞的可能性. 组播组中的主机可以在同一个物理网络, 也可以来自不同的物理网络 (如果有组播路由器的支持).

广播 (Broadcast) 传输: 是指在 IP 子网内广播数据包, 所有在子网内部的主机都将收到这些数据包. 广播意味着网络向子网主机都投递一份数据包, 不论这些主机是否乐于接收该数据包. 然而广播的使

用范围非常小, 只在本地子网内有效, 因为路由器会封锁广播通信. 广播传输增加非接收者的开销.

目前, 使用得最为广泛的组播技术是 IP Multicast. IP 组播技术是一种为优化使用网络资源而产生的技术, 通常用于多点工作方式下的应用程序中, 它是标准 IP 网络层协议技术的一个扩展. 从 Steve Deering 于 1989 年提出的 IETF 的 RFC1112 “Host Extension for IP Multicast” 中的定义我们可以得知: IP 组播的核心思想是通过一个 IP 地址向一组主机发送数据 (UDP 包). 发送者仅仅向一个组地址发送信息, 接收者只需加入到这个分组就可以接收信息, 所有的接收者接收的是同一个数据流, 组中成员是动态的, 可以根据自己的意愿随时随意加入或退出. 每一台主机都可以同时加入到多个组中, 每一个组播地址可以在不同的端口或者不同的套接字 (Socket) 上有多个数据流, 同时许多实际应用可以共享一个组地址. IP 组播技术可以有效地避免重复发送可能引起的广播风暴, 并且能够突破路由器的限制, 将数据包传送到其它网段.

2.2 IP 组播技术在多点视频数据传输方面的优势

由于数字视频在网络传输时有很大的数据吞吐量, 如果使用端对端的 IP 单播技术进行数字视频的多点传送, 首先, 视频服务器必须始终保持在侦听状态, 以了解每一个动态加入的客户端的服务请求, 而套接字的侦听非常消耗系统的 CPU 资源, 过于频繁的侦听容易造成系统的不稳定, 还会影响视频传输的实时性, 造成视频在网络中传输时出现频繁抖动, 最终影响视频传输的服务质量 (QoS); 其次, 视频服务器面对不同的客户端的同一视频服务请求, 需要进行重复发送, N 个客户端需要占用 N 倍的网络带宽资源, 极大地浪费了网络带宽资源, 如果控制不力, 还会引起广播风暴, 造成系统全面崩溃. 因此, 在网络带宽环境能够无限满足视频传输需要的前提下, 点对点传送和组播在性能上无本质差异, 但是, 这种理想状态基本上不会出现, 否则除了研究网络带宽以外, 其它的网络技术就失去了研究的基础和意义. 我们设想在 10BASE-T 的局域网环境下, 当只有 2 个或单个客户机提出视频服务请求时, 二者无明显性能差异; 当有 3 个至 5 个客户机提出视频服务请求时, 二者之间的差异就比较显著, 采用点对点传送方式的视频服务器明显已经力不从心, 网络丢包和延迟比较严重, 接收端视频明显滞后、不连续; 当有 5 个以上的客户机提出视频服务请求时, 就造成了广播风暴, 系统处于崩溃的边缘.

由此可见, IP 组播技术在多点视频数据传输方面具有很大的优势, 当某个 IP 站点向网络中的多个 IP 站点发送同一视频数据时, IP 组播技术可以减少不必要的重叠发送, 与多次点对点的单播(Unicast)相比, 减轻了系统和网络的负担, 提高了 CPU 资源和网络带宽的利用率, 极大地改善了视频数据传输的实时性. 参与通信的各主机不论是源站点还是目的站点均使用同一程序, 无客户机和服务器之分, 从而具有对等性.

2.3 IP 组播地址格式

IP 地址方案专门为组播划出一个地址范围, 在 IPv4 中为 D 类地址, 范围是 224.0.0.0 到 239.255.255.255, 并将 D 类地址划分为局部链接组播地址、预留组播地址、管理权限组播地址如下:

局部链接地址: 224.0.0.0~224.0.0.255, 用于局域网, 路由器不转发属于此范围的 IP 包;

预留组播地址: 224.0.1.0~238.255.255.255, 用于全球范围或网络协议;

管理权限地址: 239.0.0.0~239.255.255.255, 组织内部使用, 用于限制组播范围.

IP 组播技术比较符合多点、多网络平台和可扩展性的视频监控要求, 而且可以实现跨楼宇范围的远程视频监控. 因而, IP 组播技术的使用对于网络

视频的多点实时传输、网络多点实时监控具有特别重要的意义.

3 应用实践

根据上文所述, 我们设计实现了数字视频局域网监控系统.

3.1 数字视频局域网监控系统技术特点

(1) 稳定性: 本系统不是纯粹采用数字视频、而是模拟和数字的结合. 因而具有高度的稳定性和可靠性.

(2) 灵活性: 便于实现整个系统的模块化管理, 将视频模块、云台镜头控制模块、局域网传输模块等模块通过计算机集中管理, 大大减少了工作难度.

(3) 信息智能化: 单一传统的监控系统因其通用性差, 不易扩展, 不能网络化等致命缺点, 势必会被淘汰. 而基于局域网的监控系统建立起以计算机为中心的监控平台, 为信息化的管理奠定了坚实的基础. 以计算机为控制中心, 通过系统软件实现控制界面的可视化及控制环境的数字视频化, 可以方便地实现灵活机动的智能化控制.

3.2 系统结构图

系统结构如图 2 所示.

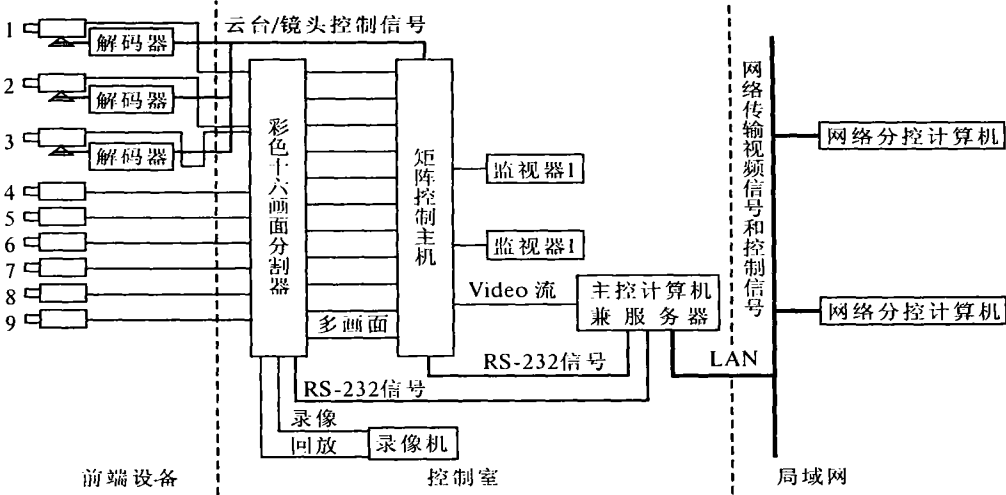


图 2 系统结构图

(下转第 184 页)

解密: $m = cd \pmod{n}$.

利用目前已经掌握的知识和理论, 分解 2 048 bit 的大整数已经超过了 64 位计算机的运算能力, 因此在目前和预见的将来, 它是足够安全的.

3 结 语

数据信息网络安全研究历经了通信保密、数据保护两个阶段. 未来解决网络数据安全问题应从安全体系结构、安全协议、现代密码理论、信息分析和监控以及信息安全系统五个方面开展研究, 各部

分相互协同形成有机整体.

参考文献:

- [1] 白成杰, 白成林, 韩继广, 等. 计算机通信网络及应用 [M]. 北京: 人民邮电出版社, 2003.
- [2] 袁京生, 吴砚农. 计算机网络安全基础 [M]. 北京: 人民邮电出版社, 2002.
- [3] 卢开登. 计算机密码学: 计算机网络中的数据保密与安全 [M]. 北京: 清华大学出版社, 2004.

〔责任编辑 张惠民〕

(上接第 180 页)

4 结 语

在局域网的环境下, 我们成功地完成了视频(压缩之后的视频流)的实时传输, 能够满足视频监控的各项指标. 整个系统(包括服务器端和客户端)的软件用 VC++6.0 编程实现, 同时在校园网上对该软件进行了测试, 效果和在局域网上一致. 如果要投入到因特网上使用该软件, 需要在传输策略上作修改后, 可以达到上述两种环境下的效果.

参考文献:

- [1] 杨磊. 闭路电视监控系统 [M]. 北京: 机械工业出版社, 1999.
- [2] David Bennett. VC++6 开发人员指南 [M]. 徐军等译. 北京: 机械工业出版社, 1999.
- [3] 黎洪松. 数字视频技术及其应用 [M]. 北京: 清华大学出版社, 1997.
- [4] 玉琢. 多媒体技术 [M]. 北京: 清华大学出版社, 1999.

〔责任编辑 张惠民〕