

基于 H.263 协议的网络视频监控系统研究

The Study of Network Video Monitoring System Based on H.263 Protocol

张利飞 毛文林 王文庆 (西安交通大学电子与信息工程学院 710049)

摘要:为了克服传统监控系统的缺点,本文提出了基于 H.263 协议的网络视频监控系统的实现方案,并对系统的软硬件结构、系统的实现方案、视频流的传输和播放进行了阐述。

关键词:网络 监控 服务器 视频

1 研究背景

传统监控系统在摄影机影像的浏览上藉由低阶的四或十六分割多任务器输入处理或由昂贵的模拟式矩阵进行选择,所有的影像经由制式化的安排,无法弹性变更顺序与画面组合,储存的影像资料无法有效的保存与管理,影像回放仅能限制于控制室或特定地点,投入昂贵,不稳定的单机封闭式架构,分散的管理功能等。传统监视系统无法发展成更高程度整合的产品,更无法整合出大型的监控系统。

为了克服这些缺陷,实现远程的实时监控,影像的存储和随意回放,我们开发出了 NVS (Network Video System) 系统。本系统的研究方向是:资料数字化、系统网络化、功能模块化。

2 系统架构设计

2.1 系统的硬件拓扑结构

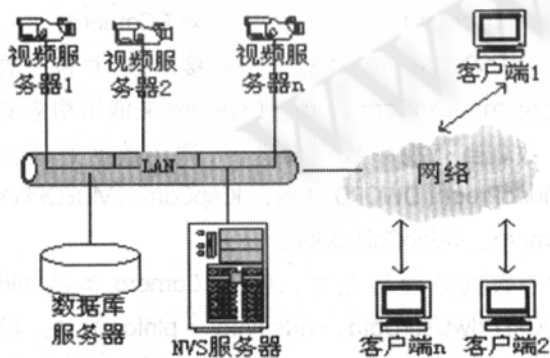


图 1 系统硬件拓扑结构

在网络中,每个摄像机同一个视频服务器相连。视频服务器设有固定 IP,负责视频数据的采集、压缩(基于 H.263)、传送。

视频数据传送到 NVS 服务器,NVS 服务器负责数据的存储、转发等。用户登陆 NVS 客户端,通过与服务器的交互,便可实现对远程场景的实时监控、录像回放、图像设置等功能。

2.2 系统的软件体系结构

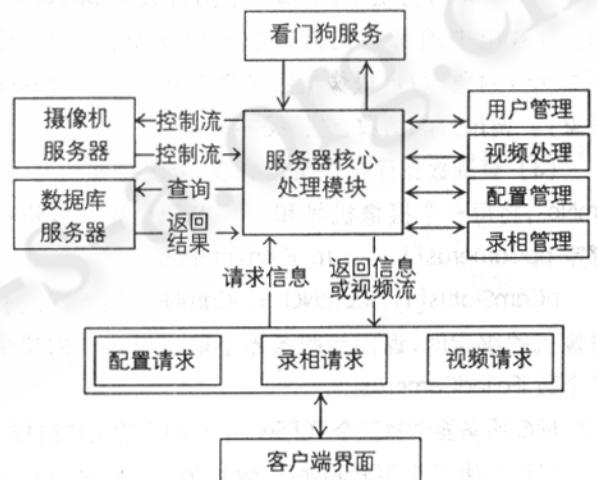


图 2 系统软件体系结构

NVS 服务器端主要由四部分构成:

(1) 目录服务。在目录服务中,应用程序可以建立链接并获得配置资料的集中信息。

(2) 网络视频传输服务。此服务可以使 Internet 上的客户端用户在不使用 VPN 的情况下通过因特网安全的访问,还负责视频的重定向或者通过解决网络

地址转换等相关问题。

(3) 文档存储服务。此服务在 NVS 系统中是至关重要的,因为它是 NVS 系统与摄像机服务器之间的联络纽带。它负责发现动态设备,以及轮检摄像机服务器的状态。所有与摄像机服务器设备的通信都通过这一服务来建立。它也是所有视频与多媒体数据流存储的地方。

(4) 看门狗服务。此服务可以提供对 NVS 系统的监视功能。如果 NVS 某一个服务失败,看门狗负责重新启动服务。

3 系统实现方案探讨

3.1 NVS 服务器启动初始化过程

(1) 初始化数据库,重置一些数据库中的状态信息表,将相关项清零或者设置成初始值,由 `ResetDataBase()` 完成。初始化摄像机状态数组 `CAMSTATUS [MaxCameras]`,将地址赋给 `pCamStatus`,数组中的每一个元素都是一个摄像机状态结构体。

(2) 初始化所有摄像机所需要的系统资源。在 `InitAllCameras()` 方法中,申请一个指针数组 `ppCameras[]`,数组中的每个元素存放一个指针,指向一个 `CCamera` 对象(每台摄像机服务器对应一个 `CCamera` 对象)。调用分配函数,进行缓冲区申请。

(3) 查询数据库摄像机信息表,取出摄像机号 `iCamNo`,将每一个摄像机号和一块摄像机缓冲区进行绑定 `ppCameras[i] -> m_iCamInfoPos = i;`

`pCamStatus[i].iCamNO = iCamNo;` 当完成所有摄像机的绑定时,返回给服务器不能正常启动的摄像机个数 `iFailedCameras`。

3.2 NVS 服务器创建三个 TCPServer 和相应的工作线程

(1) 创建三个 `TCPServer`。`svrVideo` 负责对视频数据的处理;`svrControl` 负责对控制和配置信息的处理;`svrPTZ` 负责对摄像机和云台的控制命令的处理。

(2) 创建一个 `TaskChecker` 线程,检查要执行的录相任务。当添加录相任务时,NVS 服务器会激活摄像机服务器,此时摄像机服务器开始向 NVS 服务器发送视频数据,或者其他类型的数据。

(3) 创建一个 `CameraChecker` 线程,负责定时查询摄像机的连接状态,在固定时间对每天保存的录相进行检查并删除旧的录相。初始化共用文件存放的路

径,然后初始化每个摄像机的录相文件存放的路径,初始化报警录相的存放路径,以及磁盘满时的文件处理策略。

3.3 启动所有的摄像机完成服务器启动

(1) 调用 `StartAllCameras()`,启动所有的摄像机。执行 `ppCameras[i] -> StartUp()`。在 `StartUp()` 方法中首先启动云台控制,然后调用 `ICspCam` 类中的方法连接上摄像机:

```
m_pCspCam -> SetAuthInfo ( m_CamCfg. c8UN,
m_CamCfg. c8PWD)
```

```
m_pCspCam -> Connect( ( char * ) m_pszIP)
```

(2) 生成系统托盘,将托盘最小化。至此服务器的启动完成,此时服务器有三个 `TCPServer` 处于侦听状态,在服务器端使用 `TidThreadMgrPool` 控件,它用来管理三个 `TCPServer` 生成的线程。

3.4 NVS 客户端实现

用户登录后,需要执行不同的功能时,客户端会向服务器申请不同的端口,用来建立相应连接。服务器端会触发相应 `TCP` 控件的处理函数,生成相应的处理线程。每个线程生成后,都要申请自己的处理缓冲区。然后取得客户端发送的信息包的包头放入缓冲中,以包头中的消息类型项 `MsgType` 作为开关语句的条件,针对不同的消息类型作相应的处理,并返回给客户端相应的信息包。

4 基于 H. 263 协议的视频传输和播放

4.1 视频服务器到 NVS 服务器的视频流

NVS 服务器启动后,会为每一个摄像机服务器生成一个 `CCamera` 对象。在构造函数 `CCamera()` 中,生成一个 `CSPCAM_API` 动态库的连接对象,并将它的地址赋给 `m_pCspCam` 由 `m_pCspCam` 来调用相应的处理函数。当有视频数据发送时,调用数据处理函数 `SetVideoProc((DWORD) this, (ICspCam::VIDEOPROC) CCamera::VideoCallback)`。

在数据处理函数中,调用 `CCamera` 类的 `Video(DWORD dwUserData, IVideoInfo * pInfo)` 方法。指针 `pInfo` 指向一个 `IVideoInfo` 类,里面存放所有从摄像机服务器传过来的视频数据信息,包括 I、P 帧标志信息和压缩过的数据等。

在 `Video()` 中将数据写入摄像机缓存中:`m_pp-`

Frames[] 数组指向视频缓冲区的 144 帧, 数组的每一项对应视频缓冲区中的一帧, 压缩过的视频数据放在 `pInfo -> m_pEncodeVideo`, 先写入头信息, 再写入视频数据。

同时, NVS 服务器会判断是否有录相任务, 进行相应的文档存储服务。

4.2 NVS 服务器到客户端的视频流

(1) 请求实时视频。当客户端有实时视频播放请求时, 客户端与服务器建立 `svrVideoTCP` 链接。NVS 服务器自动调用 `svrVideoExecute (TidPeerThread * AThread)`, 生成一个线程进行处理。用户发送请求消息 `NVS_REALTIME_VIDEO_REQUEST` (请求实时视频数据)。服务器首先调用 `CheckRights()` 来验证用户是否有权限。如果验证通过, 服务器处理线程则从对应的摄像机缓存中找出最后的一个 I 帧, 从这一帧开始向客户端发送。

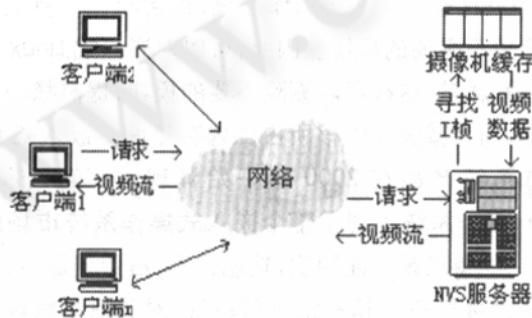


图 3 实时视频请求时的视频流

(2) 请求录相视频。当客户端有录相视频播放请求时, 建立 TCP 链接、生成处理线程同(1)。处理线程根据摄像机号 `nCamNO` 和录相开始时间调用 `GetRecfileName()` 方法。此方法通过数据库服务器可以获得录相文件的存放路径和文件名, 并将值返回给参数 `szFilename`。此时服务器处理线程可以打开该录相文件。

然后执行循环操作: 先读出第一帧的包头信息, 根据包头中的帧长度 `dwDataSize` 将视频数据读出, 将第一帧完整发送给客户端。然后将发送的长度累加起来 `dwSended + = dwReaded + dwHeaderLen`, 判断已经

发送的长度 `dwSended` 是否大于录相文件的长度 `dwLength`。如果 `dwSended < dwLength`, 就接着对第二帧做相同操作, 否则就说明录相文件已发送完毕, 结束循环。

4.3 NVS 客户端视频播放

客户端发送视频请求后, 一旦收到服务器的确认信息, 即开始接收视频数据进行播放。客户端的播放是通过 `PlayForm.ocx` 来实现的。用组件形式便于客户端的移植、扩展和升级。

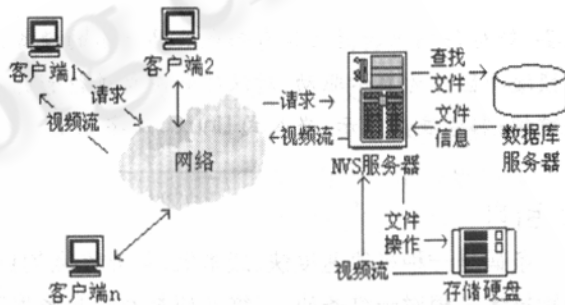


图 4 录相视频请求时的视频流

5 结束语

NVS 系统实现了监控信息的数字化处理, 能够依托普及率极高的以太网为骨干实现异地传输、远程控制; 而组件技术的应用让系统拥有更好的升级性, 当有新的影像技术、压缩技术和储存技术推出或增加新的功能时, 能以较低的成本对系统进行实时升级。该系统已在油田、发电厂、小区监控等多行业投入应用, 取得不错的经济和社会效益。

参考文献

- 1 NVS 系统开发进程报告, 西安交通大学电信学院网络技术课题组, 2004。
- 2 林福宗, 多媒体技术基础, 清华大学出版社, 2002。
- 3 田丰、胡湘渝、吴成勇, C++ Builder 6.0 COM 程序设计, 北京 国防工业出版社, 2003。
- 4 詹青龙, 网络视频技术及应用, 西安电子科技大学出版社, 2004。