

基于 gst-rtsp-server 的网络摄像机设计与实现

朱洪飞, 马跃

(北京邮电大学计算机学院, 北京 100876)

摘要: 在现代社会中, 安防工作无处不在。传统的视频监控系统布线复杂, 成本较高, 越来越不能满足人们的需求。为了利用现有的以太网, 本文提出了一种基于 gst-rtsp-server 流媒体服务器的网络摄像机解决方案。该系统利用 GStreamer 多媒体框架完成视频、音频采集, 编码以及本地存储。用户可通过 gst-rtsp-server 流媒体服务器进行实时监控或录像文件点播。此外, 在流媒体服务器中引入了 FEC 编码, 可适应较差网络环境, 完成一定程度上的数据包修复工作。

关键词: 网络摄像机; GStreamer; RTSP; FEC

中图分类号: TN919.82

Design and Implementation of Network Camera Based on gst-rtsp-server

ZHU Hongfei¹, MA Yue²

(1. Beijing University of Posts and Telecommunications, Beijing 100876;

2. Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: In modern society, the security work is everywhere. The traditional video monitoring system, which is complicated to be connected and relatively high cost, can hardly satisfy people's needs. In order to use the existing Ethernet network, this paper put forward a network camera solution which is based on gst-rtsp-server streaming media server. The system uses GStreamer multimedia framework to do video, audio capture, coding and local storage. Users can watch real-time monitoring or video file on demand through the streaming media server. In addition, in streaming media server introduced into the FEC coding, which make this system can adapt to a poor network environment and complete a degree of packet repair work.

Key words: Network Camera; GStreamer; RTSP; FEC

0 引言

世界上第一台网络摄像机诞生于 1996 年, 至今已经有十几年的历史。在网络摄像机出现的早期, 它们的技术水平与专业的模拟摄像机无法相提并论, 通常被当作普通的 Web 摄像头使用, 用来通过局域网或 Internet 观看物体或事件。由于网络摄像机最初的设计目标就是为了利用数字图像、网络 and Internet 所具有的优势, 来拓展基于网络的新应用, 因此在最开始的时候, 在视频监控应用被模拟技术所统治的时候, 从来没有人想过会把它应用到专业的视频监视应用当中。这一情况显然已经发生了变化: 近几年间, 网络摄像机的技术水平已经赶上了模拟摄像机, 能够完全满足专业视频监控的实际需求和技术规范, 在一些与视频监控相关的重要因素方面, 网络摄像机的性能甚至已经超过了模拟摄像机。因此, 用户有足够的理由在专业视频监控应用中选择网络摄像机。

作者简介: 朱洪飞 (1987-), 男, 硕士研究生, 计算机网络与多媒体通信

通信联系人: 马跃, 男, 副教授, 计算机网络与多媒体通信. E-mail: mayue@bupt.edu.cn

1 系统介绍

本系统采用的传输方式是将实时视频/音频流或多媒体文件经过特殊的压缩方式分成一个个压缩包，通过流媒体服务器^[1]向用户计算机连续、实时传送。在采用流式传输方式的系统中，用户不必像采用下载方式那样等到整个文件全部下载完毕，而是在本地计算机上对网络实时传输过来的视频/音频编码数据直接进行解压播放。

RTP 协议^[2]和 RTSP 协议^[3]作为流式传输协议，具有实时性、低带宽以及支持面向无连接的特点，非常适合于进行流媒体数据的传输，图 1 所示是 RTP 和 RTSP 协议在本视频传输系统中的工作机理^[4]。

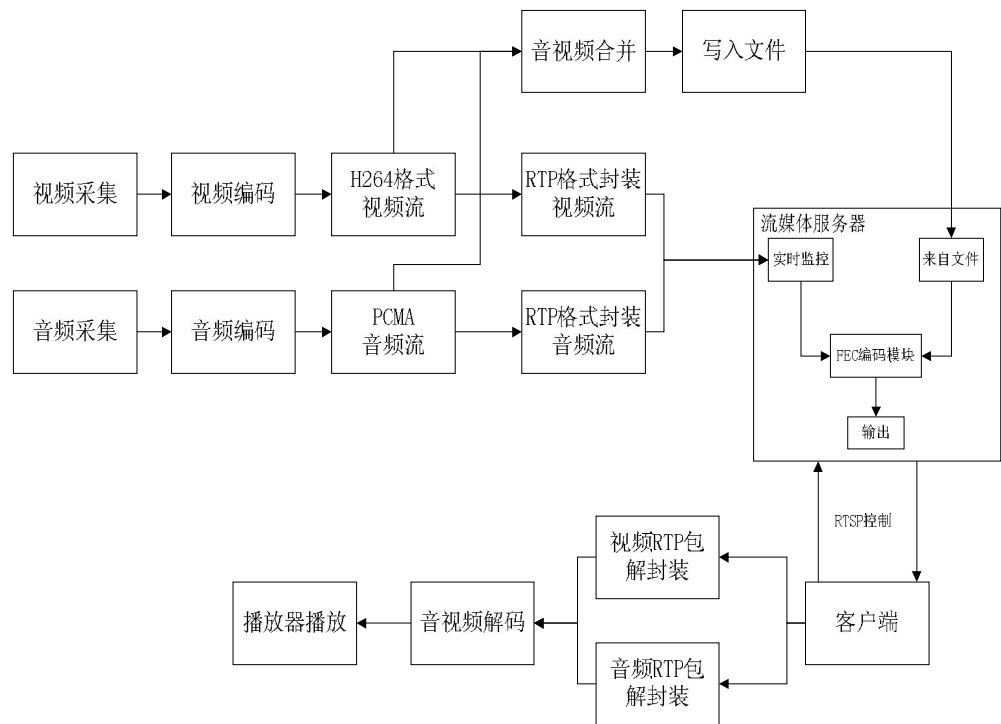


图 1 视频传输工作机理
Fig.1 Video Transmission Mechanism

2 系统实现

2.1 GStreamer

在本系统当中，前端音视频采集、编码及 RTP 数据流封装通过 GStreamer^[5-6]多媒体框架实现。GStreamer 是一个开源的多媒体框架库，其目标是要简化视频应用程序的开发，目前已经能够被用来处理多种格式的多媒体数据。它具有良好的跨平台性，可移植到很多操作系统、处理器和编译器上；具有强大的内核库，具有良好的面向对象设计和继承关系；采用了轻量化的数据流，拥有高性能、低消耗的优点；采用时钟机制，保证内部数据流同步。

GStreamer 采用了基于插件^[7]和管道的体系结构，框架中的所有的功能模块都被实现成可以插拔的组件，并且在需要的时候能够很方便地安装到任意一个管道上，所有插件都通过管道机制进行统一的数据交换。在本系统中，使用了相应的插件来实现系统前端功能，与图 1 对应，其结构如图 2 所示。

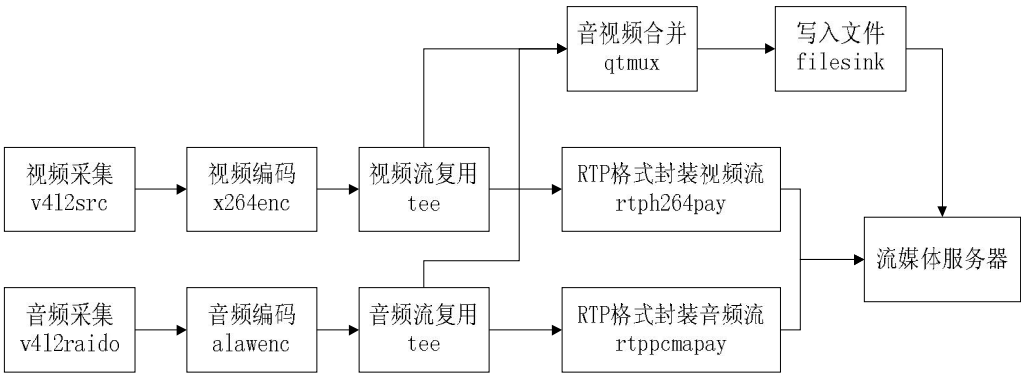


图 2 GStreamer 插件连接
Fig.2 GStreamer Element Connection

65 2.2 流媒体服务器

本系统采用了开源的流媒体服务器 `gst-rtsp-server`，它工作于 `GStreamer` 之上，能够获取管道当中的 `RTP` 数据流，通过 `RTSP` 协议控制与客户端之间的交互。此外，本系统在流媒体服务器中引入了 `FEC` 编码^[8]模块，可以完成一定限度内的数据包恢复工作，从而适应较差网络环境。

70 2.2.1 FEC 编码

`FEC`（Forward Error Correction），是一种数据编码技术，传输中检错由接收方进行验证，在 `FEC` 方式中，接收端不但能发现差错，而且能确定二进制码元发生错误的位置，从而加以纠正，发现错误无须通知发送方重发。本系统所使用的 `FEC` 编码与此有一定的区别，不是在数据包内部编码，而是数据包级编码，构造冗余的 `FEC` 数据包，原理如图 3 所示。

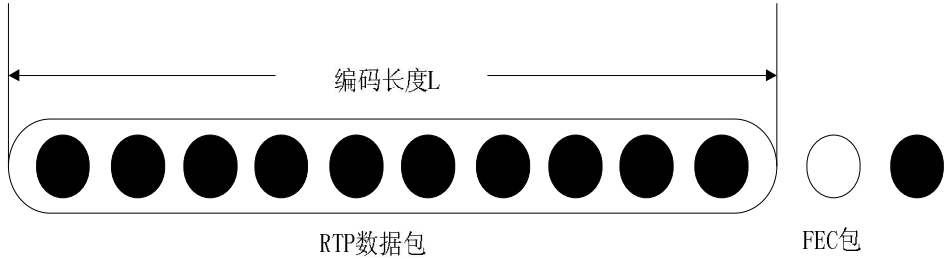


图 3 FEC 编码原理
Fig.3 FEC Encoding Principle

`FEC` 编码模块对于收到的 `RTP` 数据流，根据约定的编码长度 `L`，每隔 `L` 个 `RTP` 包插入一个 `FEC` 包。`FEC` 包结构如图 4 所示，与 `RTP` 数据包相区别，`FEC` 包当中增加了 `FEC` 头，包含编码起始 `RTP` 数据包序列号、数据包间隔等信息。`FEC` 包的数据部分由 `L` 个 `RTP` 包采用相应的算法生成，本系统采用 `L` 个 `RTP` 包数据部分相应位异或的方法。当客户端收到整个数据流后，会根据 `FEC` 包头部包含的起始 `RTP` 数据包序列号检测丢失情况，如果在这 `L` 个 `RTP` 包中有一个丢失，则使用 `FEC` 包数据部分与其余 `L-1` 个 `RTP` 包数据部分相应位异或可还原丢失的 `RTP` 包。如果有多个包丢失，则不能恢复。也就是说，本系统采用的 `FEC` 编码的有效恢复条件为丢包长度大于等于编码长度 `L`。

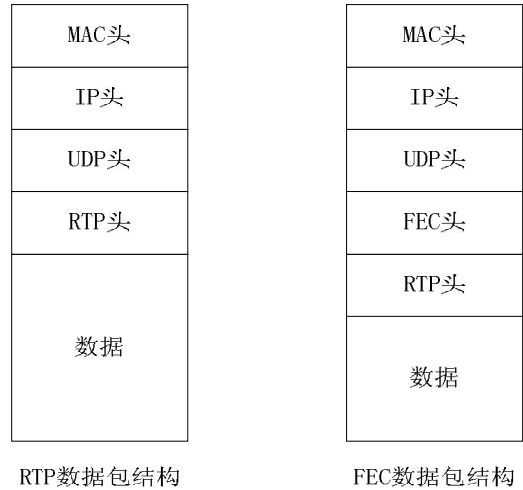


图 4 RTP、FEC 包结构
Fig.4 RTP、FEC Packet Structure

2.2.2 通信协议选择

90 gst-rtsp-server 基于 RTSP 实时流媒体传输协议，定义了整个流媒体客户端和服务器的会话过程。RTSP 协议本身只对传输流进行控制，数据流的实际传输使用的是 RTP 协议。RTSP 协议中，最基本的功能包括 OPTION，DESCRIBE，SETUP，PLAY，PAUSE，TEARDOWN 和 ANNOUNCE 这几种。可扩展的功能还包括 RECORD，REDIRECT，GET_PARAMETER，SET_PARAMETER 等。一个简单的 RTSP 交互过程如图 5 所示。

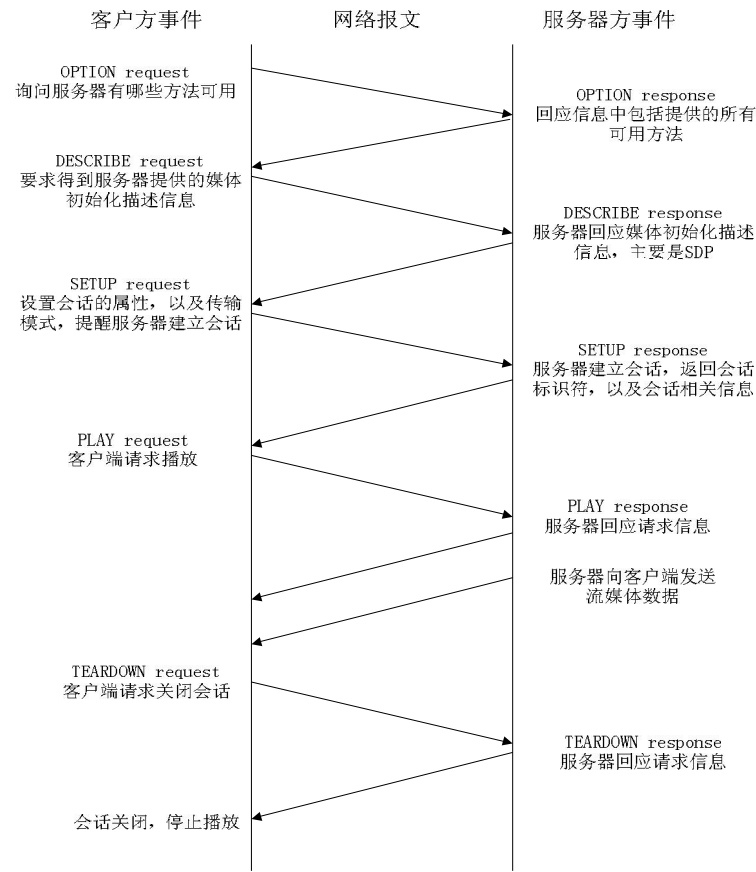


图 5 RTSP 交互过程
Fig.5 RTSP interaction process

上述的过程是标准的、友好的 RTSP 流程，但实际的需求中并不一定按部就班来。其中 SETUP 和 PLAY 是必需的。如果服务器客户端约定好，有哪些方法可用，则 OPTION 请求可以省略。如果有其他途径得到媒体初始化描述信息（比如 http 请求等），则也不需要通过 RTSP 中的 DESCRIBE 请求来完成。对于 TEARDOWN 请求，可以根据系统需求的设计来决定是否使用。

3 系统功能测试及结果

3.1 测试项目

- (1) 系统整体功能测试，能否得到实时监控视频；
- (2) FEC 功能测试，能否恢复丢失数据包。

3.2 测试结果

- (1) 系统整体正常工作，通过抓包分析，RTSP 交互流程与预期一致，能够正常观看实时监控视频与录像文件；
- (2) 客户端模拟丢包，当丢包间隔大于等于编码长度时，能够完全恢复丢失数据包。当丢包间隔小于编码长度时，不能恢复丢失数据包，与预期一致。

4 结论

本文提出了一种基于 gst-rtsp-server 流媒体服务器的网络摄像机解决方案，通过系统测试，验证了整个系统设计的正确性、可行性，为实时监控系统的方案选择提供了很好的参考价值，具有一定的现实意义。

[参考文献] (References)

- [1] 廖勇. 流媒体技术入门与提高[M]. 北京：国防工业出版社，2006.
- [2] IETF. RTP: A Transport Protocol for Real-Time Applications[S]. RFC3550.
- [3] IETF. RTSP:Real Time Streaming Protocol[S]. RFC2326.
- [4] 韩慧英，潘娅. 基于 Davinci 的嵌入式 Web 视频监控系统[J]. 兵工自动化，2010,29（4）：75-77.
- [5] Wim Taymans , Steve Baker , Andy Wingo. GStreamer Application Development Manual[OL]. <http://gstreamer.freedesktop.org/data/doc/gstreamer/head/manual/html/index.html>
- [6] 肖文鹏 . 用 GStreamer 简化 Linux 多媒体开发 [OL]. [2004-6-1]. <http://www.ibm.com/developerworks/cn/linux/l-gstreamer/>
- [7] GStreamer Team. Overview of available plug-ins. <http://gstreamer.freedesktop.org/documentation/plugins.html>
- [8] 杨天扬，许卫东，黄建华. 802.16e 标准中 FEC 编码技术研究[J]. 铁路计算机应用，2008，8：45-49.