基于 JMF 的电梯实时视频广播系统

Network Video Broadcast System based on JMF

吴建文　陈立定　（华南理工大学自动化学院 广州 510641）

摘　要： 为了实现建筑物的真正智能化和信息化，解决电梯作为建筑物内的“信息孤岛”这一问题，本文构建了一套基于 JMF 的网络视频实时广播系统。本文先概要叙述该系统的实现技术：MPEG－4 编解码算法、RTP/RTCP 实时传输协议，流媒体技术，然后利用 Sun 公司开发的专门应用于网络视频传输与播放的 JMF 包，针对电梯这一特殊环境构建网络视频广播系统。

关键词： MPEG－4　流媒体　RTP/RTCP　JMF

1　引言

随着我国经济的迅速发展和我国人们生活水平的不断提高，人们对信息化的要求也日趋强烈，而人们大部分的时间都是呆在建筑物内，因此提高建筑物的智能化和信息化尤其重要，而电梯作为建筑物内的一种重要的运输工具，却成为建筑物内的信息孤岛，电梯内的人们与外界隔绝。

本文将利用 MPEG－4 编解码算法、流媒体技术和 Sun 公司推出的一种开发流媒体应用的应用程序接口 JMF（Java Media Frame－work），开发实现电梯信息化的网络视频实时广播系统（电梯视频广播系统）。

2　电梯视频广播系统的架构

电梯视频广播系统采取 B/S 模式，包括终端播放器和服务器端两个主要部分，其中服务器端又包括视频服务器、视频管理发布平台。管理员成功登录后，即可看到视频管理发布界面，可以看到各电梯正在播放的视频信息，可以对它进行停止，快进快退，调节音量等操作；也可以从视频服务器中搜索要播放的视频，然后组播到各电梯。另外，服务器端还包括视频采集和存储平台，可以从 VCD/DVD，移动硬盘等工具拷贝视频，并进行一些关于视频信息的记录，以方便搜索。整个系统构成如下图所示。

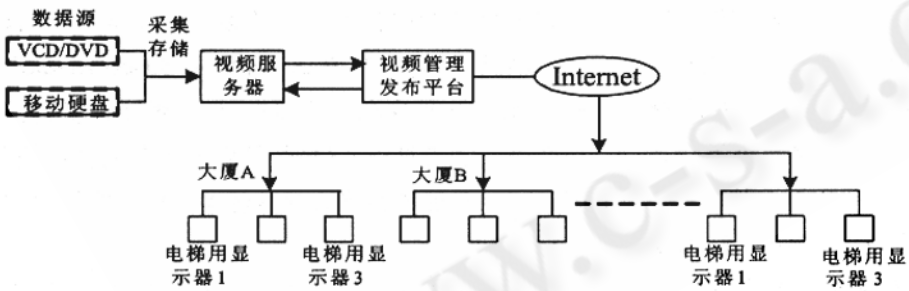

图 1　电梯视频广播系统系统图

3　电梯视频广播系统的实现技术

3.1　MPEG－4 编解码技术

电梯视频广播系统的数据源经过 MPEG－4 算法压缩成 MPEG－4 视频流，然后利用 RTP/RTCP 协议传输到终端播放器进行实时播放。MPEG－4 采用了新一代视频编码技术，它在视频编码发展史上第一次把编码对象从图像帧拓展到具有实际意义的任意形状视频对象，从而实现了从基于像素的传统编码向基于对象和内容的现代编码的转变，因而引领着新一代智能图像编码的发展潮流。

MPEG－4 标准提供在多媒体环境下多媒体数据的有效存储、传输、操作等方面的核心技术。MPEG－4 技术包含两个主要部分：视听对象的编码工具集、对编码对象和编码工具的描述。MPEG－4 在功能上的特

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

点主要有:基于内容的交换性(Content – based Interactivity)、高压缩率(Compression)、灵活多样的存取(Universal Access)等。基于内容的扩展性是 MPEG – 4 中灵活多样性中的一个关键因素,因为它提供了自

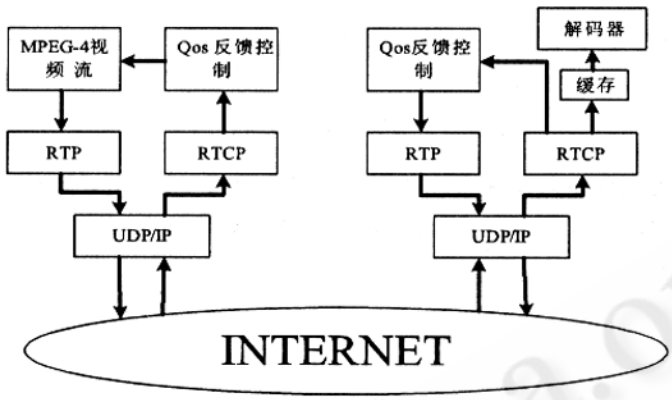

图 2 基于 RTP 的 MPEG – 4 视频传输模型

适应使用资源的能力。例如:它允许制作者规定:对具有最高优先级的对象用可以接受的质量进行显示,第二优先级的对象则以较低的质量显示,而其余内容对象则不予显示。可见,这种方式能最有效地利用有限的资源。

3.2 流媒体技术

流媒体指的是在网络中使用流技术传输的连续时基媒体,其特点是运用可变带宽技术,以流的形式进行数字媒体的传送,在播放前不需要下载整个文件,而是采用边下载边播放的方式,是视频会议、IP 电话等应用场合的技术基础。

流式传输的实现需要相应的网络传输协议,最重要的流媒体协议有两个:即实时传输协议(RTP, Real – Time Transport Protocol)和实时传输控制协议(RTCP, Real – Time Transport Control Protocol),它们是由 IETF(Internet Engineering Task Force)为视音频的实时传输而设计的,是网络多媒体应用的核心协议之一。RTP 应用于一对一或一对多的传输。RTP 本身并不能为按顺序传送数据包提供可靠的传送机制,也不提供流量控制或拥塞控制,它依靠 RTCP 提供这些服务。在 RTP 会话期间,各参与者周期性地传送 RTCP 包。RTCP 包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料,因此,服务器可以利用这些信息动态地改变传输速率,甚至改变有效载荷类型。

图 2 是描述 RTP 协议下的 MPEG – 4 视频传输过程的模型。RTP/RTCP 协议架构在 UDP/IP 之上的系统应用层中。MPEG – 4 视频流数据在发送时按顺序被封装上 RTP 报头、UDP 报头以及 IP 报头,然后封装好的 IP 数据包通过 Internet 发送给接受端,接受端收到 IP 数据报后以相反的顺序将各数据报头取出来,先分析 RTP 报头,判断版本、负载类型等有效性,按照时间戳和包序列号等信息排序 RTP 数据报,重构视频帧,再送入缓冲供解码器解码回放。在接受端利用 Qos 反馈控制分析接收数据报的时延,丢报率等信息,并由此判断网络拥塞情况,RTCP 根据这些信息周期性的向发送端返回 RTCP 控制包,以检验接收数据的真确性,并使发送端可以对输出速率作出自适应控制。

3.3 Java 媒体框架 JMF

Java 作为一种优秀的面向对象的编程语言,具有鲁棒性,安全性、多线程、可移植、分布式等诸多优点,具有强大的网络编程机制,其平台无关性更是实现了"编译一次,到处运行",是一种理想的网络编程语言。

JMF 是一种 Java 语言开发流式媒体应用的 API,它采用统一的结构和消息传递协议,能够捕获、播放、转换包括音频和视频在内的多种媒体格式,可以将音频、视频和其它时基媒体(Time – Based Media)整合到 Java 应用程序和小程序中,为多媒体开发者提供了一个强大的工具箱,它隐藏了底层复杂的实现细节,开发者利用它提供的接口就能够编写出功能强大的多媒体程序。

在 JMF 中定义了三个与 RTP 相关的包,分别是 javax. media. rtp, javax. media. rtp. event, javax. media. rtp. rtcp,通过它们,JMF 可以实现网络传输和接收回放视频流。JMF 主要有输入、处理机和输出三部分组成。一般由一个 MediaLocator 对象来描述一个输入,如 wav 文件或网络媒体流;JMF 处理机由 Processor 接口的实例描述,Processor 接口扩展至 Player 接口,它继承了 Player 接口的所有属性,并增加了 Configuring 和 Configured 两个属性,用于 Processor 从输入流收集信息时的通信;DataSink 用于读取媒体内容并实现在网络中传输。基于 JMF 的 RTP 数据流传输与接收示意图如下所示。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

图 3　基于 JMF 的 RTP 数据流传输与接收

4　RTP 协议的 JMF 实现

4.1　RTP 媒体流的传输

如图 3 所示，在网上传输 RTP 媒体流，必须使用处理器和 RTP 编码的数据源，并构建一个会话管理器来控制传输。处理器的输入可以是当前捕获的数据，也可以是已存储的数据。实现利用 RTP 协议在网络上实时发送媒体数据流的过程如下：

（1）选择要发送的媒体文件，获取数据源，构造媒体定位器；

（2）通过会话管理器（Session Manager）传输 RTP 数据流，具体为：产生一个 JMF 处理器（Processor），为每一种 RTP 格式设置相应的轨迹格式；从处理器获取数据源；会话管理器产生一个发送数据流；开始会话传输；通过监听 ControllerEvent 事件控制会话的过程；停止会话，删除会话管理器；

（3）设计界面，更好的实现交互。

基于以上步骤，将对利用 JMF 实现 RTP 视频传输的源码进行分析：

```
…
    public class MediaTransmitter{
      private MediaLocator mediaLocator = null;//媒体定位
      private String ipAddress; //接收端的 IP 地址
      private int portBase;//传输端口号
      private Processor processor = null;//处理器
      private RTPManager rtpManagers[ ];//RTP 管理器
      private DataSource dataOutput = null;//输出的数据源
…
```

为了获取媒体文件，我们在 MediaTransmitter 类中声明全局变量 mediaLocator，一个 RTP MediaLocator 需符合以下规则：rtp://address:port/content－type

其中，address 和 port 分别是接收端的 IP 地址和端口，传输端和接收端必须使用相同的端口，当以广播的模式传播媒体流到子网中的所有机器时，要以 255 作为最后的子网地址，如 192.168.1.255；content－type 是媒体流的类型，如 audio、MPEG 等。

为了使本地的媒体文件能够在网络上传输，需要创建一个处理机 Processor，它将要传输的媒体文件转换为 RTP 媒体流。创建一个 Processor 对象，需要创建一个为 Processor 描述输入和输出媒体类型的 ProcessorModel 实例。下面的 Processor 的职责是转换 MP3 音频媒体为一个 RTP 数据流：

```
public void setDataSource(DataSource ds) throws Exceptions{
    mediaProcessor = Manager.createRealizedProcessor(
        new ProcessorModel(DataSource, FORMATS,CONTENT_DESCRIPTOR);
…
```

ProcessorModel 包括三个参数：DataSource 是被处理的输入文件；FORMATS 描述输入媒体的格式；CONTENT_DESCRIPTOR 是处理机的输出格式，即 RTP 媒体流。如下所示：

```
private static final Format[ ] FORMATS = new Format[ ]{.
    new AudioFormat(AudioFormat.MPEG_RTP)};
private static final ContentDescriptor CONTENT_DESCRIPTOR =
    new ContentDescriptor(ContentDescriptor.RAW_RTP);
```

有了一个处于 Realized 状态的 Processor，我们需要设置 DataSink 以传播 RTP 媒体流：

```
dataSink = Manager.createDateSink(mediaProcessor.getDateOutput(),mediaLocator);
```

createDataSink( ) 方法获取 Processor 的输出（作为一个 DataSource 参数）和 MediaLocator 对象。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

除此之外，我们还需要创建 DataSource 对象，如下所示：

```
DataSource source = Manager. createDataSource (new MediaLocator( mediaFile. toURL) ) ;
```

我们通过调用 startTransmitting( ) 方法来开始 MediaTransmitter，如下所示：

```
public void startTransmitting ( ) throws IOException{
mediaProcessor. start( ) ;
dataSink. open( ) ;
dataSink. start( ) ;
}
```

首先开启处理机，然后打开并启动 DataSink。在这个调用后，接收机器就可在媒体传送者上监听。

4.2 RTP 数据流的接收

实时接收并播放网络媒体数据流的程序为每一种新接收到的媒体数据流产生一个播放器，一边接收媒体数据，一边进行播放。实现利用 RTP 协议在网络中实时接收并播放媒体数据流的应用程序需要完成四个步骤：

(1) 实现 ReceiveStreamListener 监听接口，监听 NewReceiveStreamEvent 事件；

(2) 当接收到 NewReceiveStreamEvent 事件后，通过事件获取接收媒体数据流(Receive Stream)，然后通过接收媒体数据流获取 RTP 数据源(DataSource)；

(3) 将数据源传给 Manager. createPlayer( ) 产生一个播放器；

(4) 给播放器添加监听器，等到播放器实现后，即可显示播放数据。

基于以上步骤实现的应用程序的部分代码分析如下：…

```
public synchronized void update ( ReceiveStreamEvent evt) {
RTPManager mgr = ( RTPManager) evt. getSource ( ) ;
Participant part = evt. getParticipent( ) ; //获取发送者
ReceiveStream stream = evt. getReceiveStream( ) ; //获取接收到的数据流
if( evt instanceof NewReceiveStreamEvent) { //接收到新的数据流
try{
stream = ( ( NewReceiveStreamEvent) evt) . getReceiveStream( ) ;
//获取新数据流
DataSource ds = stream. getDataSource( ) ; //获取数据源
…
Player player = javax. media. Manager. createPlayer( ds) ; //构造媒体播放器
player. addControllerListener( this) ; //给播放器添加控制器监听
p. realize( ) ; //实现播放器
…
}
…
```

5 结束语

本文使用 JMF 成功开发了基于 RTP/RTCP 协议的网络实时广播系统，该系统具有低丢包率、时延小、容错能力强等优点，尤其适用于低比特率和网络状况变化的环境中，在局域网中测试运行正常。

参考文献

1 汪理虎、刘春生，RTP/RTCP 协议下 MPEG－4 视频刘传输系统应用研究，工业控制计算机，2006 年 19，第 3 期.

2 葛艳红、李文锋、刘旭光，基于 RTP 流媒体实时传输的 Java 实现，计算机与现代化，2007 年第 2 期.

3 谢长勇、陈念年、李波，基于 MPEG－4 网络实时视频传输系统的设计，攀枝花学院学报，2005 年第 22 卷第 4 期.

4 邓丹，嵌入式 Mp3 流媒体网络广播系统的设计及其在智能小区中的应用，四川大学 06 年硕士学位论文.

5 傅嘉模，基于 JMF 平台开发 BS 模式下多媒体计算机远程监控系统的研究与实现，首都师范大学 02 硕士学位论文.

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn