一种 H.323 视频会议系统音视频同步方法①

白骋宇 张海峰 (杭州电子科技大学 电子信息学院 浙江 杭州 310018)

摘 要： 针对 H.323 视频会议系统设计了一种基于 RTP 的音视频同步方法，该方法在严格遵守 RTP 协议的前提下，将音视频数据联系起来通过同一个媒体通道传输，从而达到唇音同步的目的。实验表明：该方法在对图像质量影响很小的情况下，很好地实现了音视频的同步播放，并且具有实现简便，不增加系统负担等优点，具有广泛的实用性。

关键词： 音视频；同步；H.323；视频会议；RTP

A Labial Synchronization Method for H.323 Video Conference System

BAI Cheng-Yu, ZHANG Hai-Feng

(School of Electronics Information, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: An RTP-based method of labial synchronization in H.323 video conference system is described. This scheme strictly obeys the RTP protocol, but brings the audio and video data into mutual connection for transmission via the same media channel, to achieve the purpose of labial synchronization. Experiments show that the scheme has not only little influence on the image quality, but also achieves a very good synchronous player of audio and video. This method is easy for operation. It has little burden on the system and can be used widely.

Keywords: audio and video; synchronization; H.323; video conference; RTP

H.323 视频会议系统中，发送端同时采集到的音视频数据能在接收端同时播放，则认为唇音同步。终端采集到的音视频数据肯定是同步的，要保证同时播放，就要保证音视频在采集和播放处理过程中消耗的时间相同。IP 网络的特点决定了通过不同通道的音视频数据传输所消耗的时间不可能完全相同，唇音同步是视频会议系统中的一大难题。如果同时采样的音视频数据播放时间偏差在[-80ms,+80ms]以内，用户基本上感觉不到不同步，一旦超出[-160ms, +160ms]，用户就可以明显感觉到，中间部分是临界范围[1]。

1 引言

1.1 文章安排

本文第 2 节分析了现有的音视频同步方案的缺点。第 3 节详细描述了本文所设计方案的实现过程。第 4 节给出实验数据以及分析结果。第 5 节给出结论。

1.1.1 基本介绍

H.323 视频会议系统中，音视频不同步现象产生的原因除了网络环境外，还有一个是音视频的分开传输。虽然 H.323 建议音视频通过不同道道传输，但是实际传输数据的 RTP[2,3]协议和其底层的 UDP 协议都没有规定一对连接只能传输音频或者视频中的一种，通过同一个通道传输音视频完全可能，而且这样可以最大程度的减少网络原因引起的音视频不同步，本文给出了这一设想的实现方案，并做了验证。

2 现有解决方案

目前最常用的唇音同步方法从思路上可以分为以下两类：

思路一，发送端给每个要发送的 RTP 包打上时戳，记录它们的采样时间。接收端通过增加延时等方式，

① 收稿时间:2009-10-10;收到修改稿时间:2009-11-16

保证同时采样的数据同时播放。这类方法的实现需要一个中立的第三方参考时钟，需要有 RTCP 协议的 SR[2,3]的参与，如果这两个条件不具备，同步就失去了依据。

思路二，唇音不同步本质上是由 H.323 视频会议系统中音视频的分开传输和处理导致的，如果采用某种方法将音视频信息关联起来，就可以有效的避免不同步现象。一种实现方案是，将音频按一定的对应关系嵌入到视频中传输，接收端从视频中提取音频数据并重建，从而达到唇音同步的目的[4]。该方案实现较复杂，而且采用非标准的 RTP 实现方式，会给不同厂商 H.323 产品间的互通带来困难。

3　一种新的音视频同步方法

本方法基本思路是：在音视频数据的采样、编码、打包、发送、网络传输、接收、网络异常处理、拆包、解码、播放这十个处理过程中，采集、编码、打包、拆包和解码的时间基本上固定，不会因为网络环境差异造成时延的差异，而发送、网络传输、接收、网络异常处理四个过程则具有较大的随机性，其处理时间会随着网络性能的不同有较大的差异，进而造成播放时音视频的不同步。因此唇音同步处理的重点就在于保证发送、网络传输、接收、网络异常处理这四个过程中音视频的同步，即图 1 中发送同步到组帧同步之间的部分。

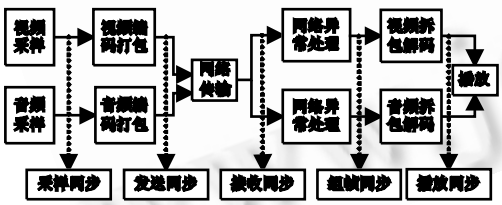

图 1　唇音同步实现全过程

其他处理过程引起的时间差，只要在系统稳定后给音频加上固定的延时即可，因为一般情况下，音频处理所花的时间比视频处理少，具体的差值可多次实验统计得到。

RTP 协议规定每个 RTP 包中所承载的有效载荷类型(PT)是唯一的，但是如果将音视频通过同一个通道传输，并且保证同一时刻采集到的音视频帧顺次交错发送，则既能保证音视频在传输中的同步，又遵守了 RTP 协议。音频数据量较小，一个 RTP 包即能承载一帧，一个视频帧则需要多个 RTP 包承载，帧结束标志采用 RTP 包头中的 Mark 字段，该字段为 1，则说明当前包是一帧的结束包。

依据上述思想，方案具体实现过程设计如下：

(1) 发送端分别独立的对音视频信息进行采样，组帧和打包，然后放到各自的缓冲队列中等待发送

(2) 数据发送模块从发送缓冲中取数据，

1) 从音频缓冲队列中取一个包(一帧)；

2) 从视频缓冲队列中取数据，每取一个包，都判断 RTP 包头的 Mark 字段是否为 1，如果为 1，说明当前视频帧已经取完，转 1)，如果 Mark 字段为 0，说明当前视频帧还未取完，转 2)；

(3) 音视频数据通过同一个通道发送到网络；

(4) 接收端收到数据，根据包头中的 PT 字段区分音视频，放到各自的接收缓冲队列中进行请求丢包重传、乱序重排等网络异常处理[5,6]，然后进入组帧缓冲等待解码器取走数据，进入组帧缓冲的数据没有乱序包和重包，偶有丢包；

(5) 音视频各自拆包组帧，实现过程如图 2 所示；

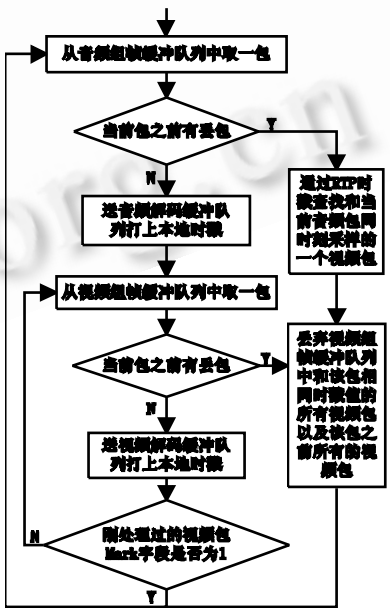

图 2　组帧同步实现原理图

(6) 音视频从各自的解码缓冲队列中按顺序取数据送解码，通过组帧过程中给音视频数据加上的本地时戳来校准后同步播放。

丢包判断实现细节说明：

在终端的可靠性和代码的健壮性得到保证的前提下，发送端是不可能有包序号不连续的，对于接收端，本方案中的丢包，是指经过丢包重传等网络异常处理策略之后依然存在的丢包，必然是及其少量的。本方法中的音频采样、组帧和打包是分开处理的，即音视频 RTP 包号分别连续，所以一般情况，依据各自的包序号即可判断是否有丢包。而对于一个会话中收到的第一个媒体包即丢失的情况，一旦出错，可能导致音视频播放时间整体错位。本文通过发送端所加的 RTP 包头中的时戳来避免这种情况，时间戳的计算公式如下：

Timestamp(0) = (unsigned long) rand();

Timestamp(t)=Timestamp(0)+△T*freq /1000;

△T = T(t) － T(0)，时间差，单位: ms；freq: 采样频率；

H.323 视频会议中，与会各方的编解码协议、采样率、帧率等参数在打开通道后的能力协商阶段即已确定，要改变这些参数，必然要重新能力协商[7]，而任何时候应用层都知道协商的结果。所以只要规定一个会话中发送的头一个音频包和头一个视频包的时戳相同，即可由时戳来建立音视频包的对应关系。实际上，视频数据头一帧的图像分成多个包传输，这几个包具有相同的时戳，同时丢失的可能性很小。而且视频组帧解码过程中，还要分 I 帧、P 帧和 B 帧区别处理，比如每个 GOP 中只要 I 帧丢失，其后的 P 帧和 B 帧都必须丢弃，直到收到下一个 I 帧，这已经超出了本文的研究范围，此处不再详述。

4　理论分析和结果验证

理论上讲，采用本方法后，在网络状态良好时能做到音视频传输中的完全同步。网络状态恶化时，随着丢包率的增加，同步效果会稍微变差，其中随机丢包比周期丢包对同步效果的影响更明显，这是因为随机丢包会引起更多的网络抖动。而在帧率码率和编解码协议不变的情况下。带宽越小，网络越容易拥塞，所以带宽降低时同步效果也会变差。

将本方案应用在开源的 H.323 协议栈 OPENH323 上[7]，实现了一个简单的基于 PC 机的 H.323 桌面终端。两台终端建立会话，通过 IP cloud 在两台终端间模拟各种复杂恶劣的网络环境，然后使用 Wireshark 抓包，可以看到音视频包的发送接收时间以及有关包头信息，进而计算出传输中引起的音视频偏差时间。考虑到算法的复杂度，本方案选择了相对较易实现的 H.261 和 GSM6.10 作为音视频编解码协议。图 3 是呼叫建立后在发送端 10.21.11.121 上截的图。发送端敲击麦克风，接收端看到敲击动作的同时听到敲击声，同步效果良好。

图 3　验证平台——终端互通实现效果图

终端 10.21.11.121 在正常网络环境下，以 512k 的带宽呼叫终端 10.21.11.152，呼叫建立 5 分钟之后用 Wireshark 抓到的音视频数据包如图 4 和图 5 所示：

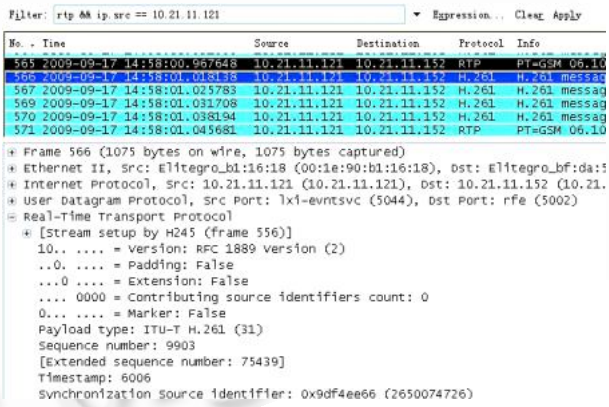

图 4　发送端音视频数据抓包

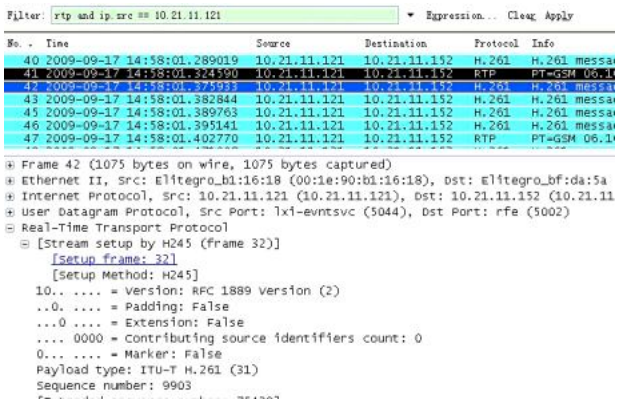

图 5　接收端音视频数据抓包

随机选取了 20 个这样的音视频组合，测得传输

引起的音视频时间差值，求的平均值为 0.000051s，即 51μs。可以认为，在正常情况下，传输阶段不会引起失步。

多次改变呼叫带宽和网络丢包率，反复试验，得到的不同环境下由传输引起的音视频时间差如表 1 所示。

表 1　不同环境下由传输引起的音视频时差(单位：μs)

带宽	周期丢包 0.5%	周期丢包 1%	随机丢包 1%	随机丢包 2%
512k	219	346	398	754
768k	283	352	371	722

由表 1 中的数据可以看出，随着丢包率的增大，音视频失步有所增加。并且相同丢包率下，随机丢包对同步效果的影响更明显，这和理论分析的结果完全吻合。但是即便在播放阶段还有 2%丢包这样恶劣的环境下，传输引起的音视频时间差仍然低于 1000us。即：该方法将[-80ms,+80ms]的同步范围的 159/160 留给音视频处理和组帧解码阶段。

理论上讲，低带宽高丢包环境下，使用该方法后视频质量会有所下降。这是因为，本文的算法增加了视频帧被丢弃的概率。如图 4 所示，每个 CIF 格式的视频帧需要 4 个 H.261 的 RTP 包来传输，其中任意一个包丢失都会使该帧成为无用帧被丢弃。采用了本文的同步策略后，如果该视频帧对应的音频包丢失，该帧也会被丢弃。这一点可以根据系统的实际需求做出取舍，比如用前一个包的重复播放来代替丢掉的音频包，而这样会增加音频播放的滞顿感。这些问题正在进一步研究中。

5　结语

本方法最大的亮点在于很好的实现了音视频同步的同时，最大程度的遵守了 RTP 协议和 H.323 标准。此外，该方法实现简便、可以和现有的唇音同步方案同时使用、并且不会额外增加系统的负担，具有很大的实用价值。

参考文献

1 贾杰,常义林,杨付正, 许延. H.323 同步控制实现研究.通信学报. 2004,25(5):67－74.

2 Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A Transport Protocol for Real-Time Applications. [1996-01-01]. http://www.ietf.org/rfc/rfc1889.txt.

3 Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A Transport Protocol for Real-Time Applications [2003-07-01]. http://www.ietf.org/rfc/rfc3550.txt.

4 严权锋,胡娟,石炎生.一种基于数字水印的音视频同步传输方法.通信技术, 2008,41(6):59－61.

5 杭州华三通信技术有限公司, H3C 视讯会议网络自适应技术白皮书[2009-09-20]. http:// ww w.h3c.com.cn/Products___Technology/Products/IP_Multimedia/IP_Video/Home/Video_Information/Tec_Hot/200801/334149_30003_0.htm

6 Polycom, Inc. Polycom Delivers HD Video Conferencing Media Server For Small And Medium Enterprises And Distributed Video Networks[2008-07-28]. http://www. polycom.com/ company/news_room/press _releases/2008/20080728_1.html.

7 卢政.如何通过 Openh323 开发自己的 H.323 协议栈 [2002-12-18]. http://www.lsdn.org/82803/