

动态网络环境下的音视频同步技术^①

南春辉, 李 博, 武 颖

(中北大学 电子测试技术国家重点实验室 仪器科学与动态测试教育部重点实验室, 太原 030051)

摘 要: 针对网络传输中由于延迟、抖动、网络传输条件变化等因素引起的音视频不同步的问题, 设计并实现了一种适应不同网络条件的音视频同步方案. 利用音视频编码技术 AMR-WB 和 H.264 具有在复杂网络环境中速率可选择特性, 结合 RTP 时间戳和 RTCP 反馈检测 QOS, 通过控制音视频编码方式, 实现了动态网络环境下的音视频同步方案. 重点介绍了可靠网络环境和动态网络环境下同步算法的设计过程, 并通过实际测试验证了此方案的可行性. 结果表明, 此方案能够保证不同网络环境中的音视频同步.

关键词: 音视频同步; 时间戳; RTP/RTCP

Synchronization of Audio and Video in Dynamic Network

NAN Chun-Hui, LI Bo, WU Ying

(National Key Laboratory for Electronic Measurement Technology, Key Laboratory of Instrumentation Science & Dynamic, North University of China, Taiyuan 030051, China)

Abstract: In view of the problem that audio and video are not synchronized, which is caused by factors such as delay, shaking and the change of network condition in transmission, this paper designs a solution for the synchronization of audio and video which adapts to different network. By use of the property of various rates of the codec AMR-WB and H.264, and RTP time stamp and RTCP feedback detection QOS, this synchronization scheme is realized by controlling the codec of audio and video. In the paper, the design process of synchronization algorithm which works in reliable and dynamic network is emphasized, and the feasibility of the scheme is verified through actual test. The results show that this design can guarantee the synchronization of audio and video in different network environment.

Key words: audio and video synchronization; time stamp; RTP/RTCP

音视频媒体间同步是多媒体系统服务质量(QoS)研究中的一项重要内容. 在网络上传输多媒体数据时, 由于终端对数据的处理方式, 以及网络中的延时、抖动, 会引起音视频流的不同步. 传统的解决方案往往存在实时性差, 时间开销大, 且无法适应动态网络环境等缺陷, 针对此问题, 本文在分析媒体间同步性定义、影响因素等的基础上, 提出了一种基于循环缓冲队列和 RTCP 反馈控制的同步解决方案.

1 媒体间同步性定义

同步是多媒体通信的主要特征, 也是其重要研究内容之一, 同步与否直接影响多媒体通信的质量. 媒体间同步即是要保持音频流和视频流之间的时间

关系^[1]. 为了描述同步, 实现相关的控制机制, 定义了相应的服务质量参数(QoS). 针对音视频, 采用时间差即偏差来表示. 结果表明, 如果偏差限制在一定的范围内, 认为媒体是同步的. 当偏移在 -90ms(音频滞后于视频)到+20ms(音频超前视频)之间时, 人感觉不到试听质量的变化, 这个区域可以认为是同步区域; 当偏移在-185到+90之外时, 音频和视频会出现严重的不同步现象, 此区域认为是不同步区域. 本设计认为偏移在-120ms 到+40ms 之间音视频同步.

1.1 音视频同步的影响因素

在网络环境下, 多媒体信息在传输过程中受到各种因素的影响, 会导致其在接收端不能正确播放, 即

^① 收稿时间:2012-03-08;收到修改稿时间:2012-04-14

音视频不同步. 引起音视频不同步的原因主要有两种: 一种是终端处理数据引起的, 发送端在数据的采集、编码、打包等模块和接收端在了解包、解压、回放等模块时, 由于音频和视频的数据量以及编码算法不同而引起的时间差. 并且发送端没有统一的同步时钟; 另一种是网络传输延时, 网络传输是受到网络的实时传输带宽、传输距离和网络节点的处理速度等因素的影响, 在网络阻塞时, 媒体信息不能保证以连续的“流”数据方式传输, 特别是不能保证数据量大的视频信息的连续传输, 从而引起媒体流内和流间的失步^[2,3].

2 音视频同步系统设计

在音视频同步系统中, 发送端在发送音视频流时, 要给各帧数据打上相对时间戳, 并且音频流和视频流, 一个作为主流, 另一个作为从流. 主流连续播放, 从流的播放由主流的播放状态决定, 从而实现同步. 考虑到人对声音更为敏感, 在本设计中选择音频流作为主流, 视频流作为从流. 发送端通过 AMR-WB 和 H.264 编码模块对 DirectShow 采集到的音视频数据进行编码, 经过同步处理, 最后利用 RTP/RTCP 等协议实现媒体流的传输和控制. 接收端接收到 RTP 传过来的音视频数据包后, 对数据进行解码, 然后同步处理, 最后通过 DirectShow 播放音视频.

3 音视频同步方案设计

考虑到传统的同步方案只是在接收端通过 RTP 时间戳实现同步, 即将具有相同时间戳的音视频数据同时表现出来, 这种方案由于没有从有效控制和适应不同网络环境的角度去实现, 并且读写时间戳的开销太大, 需要全网同步时钟等缺陷, 因此不适应于音视频媒体间同步^[4]. 针对此问题, 这里提出一种结合发送端, 利用 RTP/RTCP 以及可控音视频编码技术, 适用于不同网络条件的同步方案. 主要表现在以下两方面: 1、发送端数据的采集、编码即发送控制; 2、利用 RTCP 的反馈指标, 通过可控速率的音视频编码算法动态适应不同的网络环境.

3.1 RTP 时间戳同步

在网络畅通时, 网络传输时延基本恒定, 抖动很小, 发送端和接收端的音视频帧间隔基本保持一致, 媒体数据基本没有丢失. 由于音视频的 RTP 之间无直

接关联的控制, 所以不能通过关联控制同步. 此时主要利用 RTP 包头的时戳来解决.

在发送端, 同一媒体内的时间戳控制: 针对音频的不同采样速率和视频的不同帧率来动态的控制时间戳的递增速率; 不同媒体间的同步控制: 同一时间采集到的数据打上同样的时间戳, 并在同一线程里交替发送音视频数据包.

在接收端, 当音视频数据到达时, 先对两种数据帧进行解码, 在将其解码数据存入各自的动态循环缓冲区中. 因为音频和视频的每个数据帧解码时间不能准确得到, 为了准确地实现音视频同步回放, 采取先解码再同步处理的方法. 在网络畅通时, 可以把两种数据的解码时间差作为抖动延时的一部分来处理. 但是, 在网络环境不好时, 不采用这种方法处理.

(1) 接收端对音频帧的处理如下:

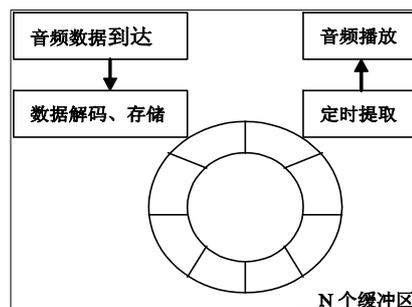


图 1 接收音频帧示意图

如图 1 所示, 为了消除抖动, 接收端采用基于循环缓存区的方法保证音频的连续性. 这种方法有两个优点: 一是可以根据 RTP 数据的接收情况动态的建立缓存空间, 二是可以保证缓存中有足够的音频数据用于播放. 接收端接收到音频帧时, 首先对其解码, 并存入动态的循环缓冲区中, 循环缓存块节点数的门限值为 N , 该值比预计最长抖动时间要大. 开始启动播放音频前, 首先把缓冲区充满, 然后定时提取音频帧播放, 并记录当前播放的时间戳.

(2)接收端对视频帧的处理如下.

如图 2 所示, 视频帧到达时, 接收端对其解码后, 将解码数据存入循环缓冲区. 为了避免高速视频画面出现的块效应, 本系统采用事件驱动的方式来播放视频流. 当缓冲区接收到一个视频数据包时, 把该帧的时间戳 $TVIDEO$ 与当前待播放的音频数据的时间戳 $TAUDIO$ 进行比较. 本设计中规定音视频帧不同步的

容忍度为 $T_{MAX}=120ms$ 。因此对一帧视频数据的处理结果分为以下三种：若 $T_{AUDIO}-T_{MAX}<T_{VIDEO}<T_{AUDIO}+T_{MAX}$ ，就播放该视频帧。

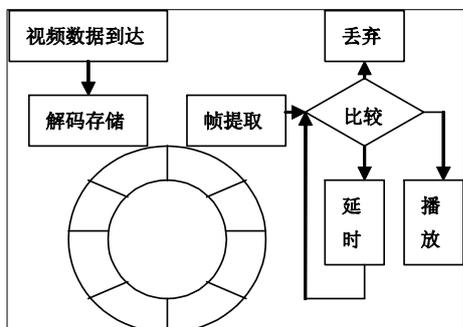


图 2 接收视频帧示意图

若 $T_{VIDEO}<T_{AUDIO}-T_{MAX}$ ，视频帧滞后，就丢弃该帧。

若 $T_{VIDEO}>T_{AUDIO}+T_{MAX}$ ，视频帧超前，等待下一次定时读取音频帧时再处理。

接收端对视频帧进行同步处理的实现代码如下：

```
OnRTPPacket ( RTPPacket *pack,
    const RTPTime &receivetime,
    const RTPAddress *senderaddress )
{
    size_t bufsize=pack->GetPayloadLength();
    memset(m_buf,0,MAX_PACKET_SIZE);
    //接收视频流的 RTP 数据包 memcpy(m_buf,
(void*)pack->GetPayloadData(),bufsize);
    //对视频数据进行同步处理
    m_psynvideo->Issynvideo(TAUDIO,m_buf);
    switch(Issyn)
    {
    case 1: //播放该视频帧
        m_pVideoOut->ReceiveVideo(m_buf,bufsize);
        break;
    case 2:
        delete(m_buf); //视频帧滞后, 丢弃该帧
        break;
    case 3:
        waite(m_buf); //视频帧超前, 等待下次处理
        break;
    }
}
```

}

3.2 RTCP 反馈控制

当网络环境较差，无法为系统提供 RSVP 时，音视频流不能按原定的传输速率传送，否则会出现数据包丢失严重的情况，这时需要采用 RTCP 来进行反馈控制。即利用 RTCP 的发送报告 SR 和接收报告 RR 包监测 QoS^[5]。

接收端将 RR 包发送给源端，该报告包含用来估算分组丢失和分组延迟抖动等必要信息。源端根据这些信息控制媒体数据的发送量，及时有效地解决同步问题。

根据评估 RR 包的参数，得到长时指标丢包率和短时指标间隔抖动。当丢包率和抖动达到一定值时：音频方面，当网络丢包率和抖动达到某一区域时，选择不同的 AMR-WB 传输速率，来降低音频传输码率，提高传输效率和系统容量，为视频传输减少了带宽负担。

视频方面，根据不同值调整视频数据的发送量，即在发送端对视频的空域和时域性能进行平衡，选择丢帧：

(1)当丢包率和抖动很高，即信道速率很低时，通过降低视频帧率，使每一帧能够具有较好的空域质量，使用户在较低的速率条件下，任然可以得到较好的图像质量。

(2)当丢包率和抖动保持在中等水平，即信道速率中速时，在保持一定的空域质量条件下，应优先考虑时域质量，增强视频的连续性。

(3)当丢包率和抖动回到较好的水平，即信道速率较高时，在空域质量达到一定程度后，继续提高空域质量，效率不会太高，反而是图像连续性的提高对视频质量的改善更明显。

4 结论

本文设计实现了一种适应不同网络环境的音视频同步方案。设计中利用 RTP 时间戳及循环缓冲区在可靠网络环境下对音视频进行同步，以及在动态网络环境下，利用 RTCP 反馈控制来动态改变音视频编码方式的同步方案。此方案已经成功应用于作者开发的网络多媒体终端上，保持了较低的丢包率，保证了终端之间多媒体信息的传输质量。

(下转第 178 页)

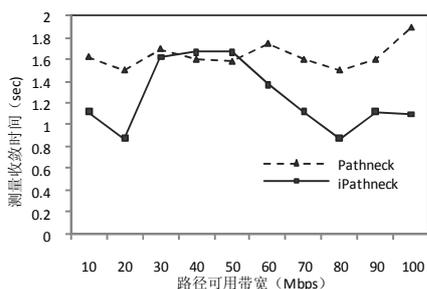


图8 算法时间收敛性比较

4 结语

iPathneck 融合了递归探测包列模型和速率模型, 集成了可用带宽测量和瓶颈定位功能。采用速率模型提高了可用带宽的测量精度, 并提高了瓶颈位置准确的准确性。引入辅助探测包协助递归探测包列采用三元搜索算法大大缩短了测量时间。该工具通过任务归并, 减小测量开销, 达到了测量精度与测量开销之间的平衡。

但是 iPathneck 对于与瓶颈链路可用带宽相近的链路仍然无法识别, 下一步工作将致力于改进这个缺陷。

参考文献

- HU N, Steenkiste P. Evaluation and characterization of available bandwidth probing techniques. *IEEE JSAC*, 2003, 21(6): 879–864.
- Salehin KM, Rojas-Cessa R. Schemes to Measure Available Bandwidth and Link Capacity with Ternary Search and Compound Probe for Packet Networks. *IET Commun*, 2010, 4(2):240–252.
- Antoniades D, Athanatos M, Papadogiannakis A, et al. Available bandwidth measurement as simple as running wget. *Proc PAM*, Adelaide, Australia, 2006.
- Sommers J, Barford P, Willinger W. Laboratory-based calibration of available bandwidth estimation tools. *Micro-processors and Microsystems*, 2007,31(4).
- Hu NN, Li ZQ, Mao M. Locating Internet Bottlenecks: Algorithms, Measurements, and Implications. *SIGCOMM 04*, Aug. 30-Sept.3, 2004,Portland, Oregon, USA.
- Goldenberg D, Qiu L, Xie H, Yang YR, Zhang Y. Optimizing Cost and Performance for Multihoming. *Proc. ACM SIGCOMM*, August 2004.

(上接第 105 页)

ication system. Gateway to the 21st Century 2nd International Conference on Personal Communications. Ottawa, Ont. 12-15 Oct.1993:975–979.

- Wong CY, Cheng RS, Ietaief KB, et al. Multicarrier of DM with adaptive subcarrier, bit, and power Allocation. *IEEE J. Select. Areas Commun*, 1999,17(10):1747–1758.
- Kim I, Lee HL, Kim B, Lee YH. On the use of Linear programming for dynamic subchannel and bit allocation in multiuser OFDM. *IEEE Globecom*,2001,6:3648–3652.
- Yin H, Liu H. An efficient multiuser loading algorithm for OFDM based broadband wireless systems. *IEEE Globecom*,

2000,1:103–107.

- Shen ZK, Andrews JG, Brian L, Evans S. Adaptive Resource Allocation in Multiuser OFDM Systems With Proportional Rate Constraints. *IEEE Trans. on Wireless*, 2005,(4):2726–2737.
- Mao ZW, Wang XM. Efficient Optimal and Suboptimal Radio Resource Allocation in OFDMA System. *IEEE Trans. on Wireless Communications*. 2008,7(2): 440–445.
- 马莉. MATLAB 语言实用教程.北京:清华大学出版社, 2010. 141–176.

(上接第 122 页)

参考文献

- 柴若楠,曾文献.音视频同步技术综述.计算机系统应用, 2011,20:223–226.
- Kuo CC, Chen MS, Chen JC. An adaptive transmission scheme for audio and video synchronization based on real-time transport protocol. 2001 IEEE Int. Conference on Multimedia and Expo(ICME2001), Tokyo, 22-25 Aug. 2001:403–406.

- Palacharla S, Karmouch A, Mahmoud SA. Design and Implementation of a Real-time Multimedia Presentation System using RTP. *Computer Software and Applications Conference, (COMPSAC'97)*. 1997: 376–381.
- 董春兵.音视频同步的研究与实现[硕士学位论文].长春:吉林大学,2007.
- 潘鹏,杜旭,叶婷.RTP/RTCP 实时传输协议的研究与 Linux 实现.计算机工程与应用,2005,41(24):105–107.