

基于 ACL 的蓝牙音频应用模型及算法研究

郭昌建¹, 吴永忠²

(1. 合肥学院 计算机科学与技术系, 安徽 合肥 230022;

2. 合肥工业大学 计算机学院, 安徽 合肥 230009)

摘要:论述了蓝牙协议的体系结构、核心协议、蓝牙传输机制、硬件模块的组成和部分多媒体音频格式。针对基于同步 SCO 链路的典型蓝牙音频应用模型所存在的一些固有缺陷, 提出了一种基于 ACL 异步链路的新型 BRTAAM 蓝牙音频应用模型及相应的 BRTATP 算法, 详细地阐述了该算法的同步原理及数据分组格式, 最后指出了进一步的研究目标。

关键词:蓝牙; 音频应用模型; 算法; 异步链路

中图分类号: TN929.5

文献标识码: A

文章编号: 1673-629X(2008)09-0068-04

Research of Bluetooth Audio Application Model and Arithmetic Based on ACL

GUO Chang-jian¹, WU Yong-zhong²

(1. Dept. of Computer Science and Technology of Hefei University, Hefei 230022, China;

2. Computer and Information Dept. of Hefei University of Technology, Hefei 230009, China)

Abstract: Briefly discuss bluetooth protocol architecture, protocol core, transfer mode, IC model and multi-media audio formats. Discusses the existing defects of bluetooth audio application model based on SCO link, put forwards a new BRTAAM bluetooth audio application model based on ACL link and its BRTATP arithmetic, addresses the principle of arithmetic's synchronization and its packet format in detail, points out the future research target.

Key words: bluetooth; audio application model; arithmetic; ACL

0 引言

典型的蓝牙音频应用模型(如图 1 所示)是蓝牙组织最早提出的应用模型之一, 它使用了蓝牙基带的同步定向 SCO 链路以及速率为 64kb/s 的对数 PCM、CVSD 编解码技术, 合适的音频编码将在链路管理器之间协商后进行选择, 音频数据不需要流经 L2CAP 层而直接到达基带层。具有协议模型层次少、算法和线路简单等突出优点。但是, 这个模型也存在一些致命的缺陷, 主要表现在两个方面:

(1) 由于音频数据分组采用的是同步 SCO 链路, 它不包括循环冗余检测(CRC), 且不允许重传, 因此, 它传输的音频难以得到 QoS 保证;

(2) 64kb/s 传输速率较低, 只适合语音传输, 完全不适用于如 CD、AC-3 等高质量的音频传输应用。

蓝牙技术除了提供点对点的同步定向传输方式外, 还提供了基于异步链路 ACL 传输方式, 该方式可支持最大不对称 723kb/s(回程为 57.6kb/s), 或对称 433.9kb/s 的传输速率。从应用范围来看, 采用异步传输技术来传输高速、高质量的音频数据将是蓝牙应用的重要发展方向之一。

鉴于上述原因, 提出了利用异步链路 ACL 分组技术实现的新型蓝牙音频应用模型(Bluetooth Real-Time Audio Application Model, BRTAAM)及相应的蓝牙实时音频传输算法(Bluetooth Real-Time Audio Transfer Protocol, BRTATP), 它可以在异步传输机制上实时地实现高质量的双声道、5.1 声道环绕立体声等音频媒体数据传输, 并提供了可靠的 QoS 保证。

1 蓝牙及多音频媒体

1.1 蓝牙协议的体系结构

蓝牙体系结构中的协议自底而上, 分为四层, 即:

(1) 核心协议: 基带、LMP、L2CAP、SDP;

(2) 电缆替代协议: RFCOMM;

收稿日期: 2007-12-24

基金项目: 国家“十五”科技公关计划资助项目(2001BA104C); 安徽省“十五”科技公关计划资助项目(01012012)

作者简介: 郭昌建(1965-), 男, 硕士, 讲师, 研究方向为计算机网络等; 吴永忠, 副教授, 博士, 研究方向为信息家电、矿用监测系统。

(3)电话传送控制协议:TCS 二进制、AT 命令集;

(4)可选协议:PPP、UDP/TCP/IP、OBEX、WAP、vCard、vCal、IrMC、WAE。

除了上述协议外,蓝牙规范还定义了主机控制接口(HCI),它可位于 L2CAP 上层,也可位于其下层^[1]。

蓝牙核心协议及无线单元是蓝牙技术实现的关键,绝大多数应用都需要它,其它协议则可根据应用需要而定。典型的音频应用一般只需要蓝牙核心协议(如图 1 所示)。

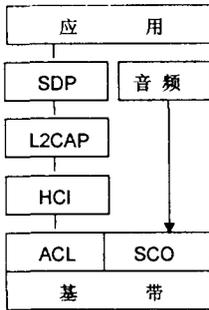


图 1 典型应用模型

1.2 蓝牙数据传输原理及数据格式

蓝牙协议采用电路交换和分组交换的混合方式,以全双工时分(TDD)传输方案实现全双工传输,其符号速率为 1ms/s,信道被划分为宽度为 $625\mu\text{s}$ 的时隙。保留时隙用于同步分组。可支持一条异步数据信道和多达 3 路同步语音信道,或者一条同时支持异步数据和同步语音的信道。每一条语音信道支持 64kb/s 的同步语音连接,异步信道可支持最大 723.3kb/s(回程为 57.6kb/s)的不对称数据传输,或者 433.9kb/s 的对称数据传输。

蓝牙工作在 2.4GHz 的 ISM 频段,在欧美使用带宽为 83.5MHz,设置了 79 个射频跳频点,间隔为 1MHz;在日本、西班牙、法国压缩了带宽,设置了 23 个跳频点,间隔仍是 1MHz。理论跳频速率为 1600 跳/s^[1,2]。

蓝牙基带支持二种连接方式:同步定向连接(SCO)方式:主要支持 64kb/s 的语音传输;异步连接方式(ACL):主要用于分组数据。

基带控制器采用三种纠错方式:1/3 前向纠错编码(FEC)、2/3 前向纠错编码、自动请求重传(ARQ)。为了确定有效载荷正确与否,CRC 应加载于有效载荷中,ARQ 方案只工作于分组的有效载荷上。分组头和同步语音有效载荷不受 ARQ 保护。

分组数据遵循 Little Endian 格式,其标准分组格式为:

识别码(72 位),头(54 位),有效载荷(0~2754 位)。

1.3 多媒体音频格式

对于 CD,当采样频率为 44.1kHz 时,双声道 16bit 的速率为

$$44.1 \times 1000 \times 2 \times 16 = 1411200 \text{ bits/s} = 1.4112 \text{ Mb/s}$$

对于 MPEG-1 的层 1、层 2、层 3,数据速率分别为:384kb/s、192kb/s 和 128kb/s。

对于杜比 AC-3,码率范围为:32~640kb/s,其典型应用是 5.1 声道消费格式 384kb/s 和双声道 192kb/s^[3,4]。

由此可见,采用基于 64kb/s 速率的 SCO 分组技术的蓝牙典型音频应用模型是无法传输高质量的音频数据的。

1.4 蓝牙硬件模块描述

蓝牙硬件主要由四个部分组成,即:无线射频单元、链路控制器(LC)、DSP 微处理器和主机接口。蓝牙规范规定主机接口至少要支持 USB、PC 卡。

2 模型与算法研究

2.1 协议栈模型

音频协议模型如图 2 所示,由蓝牙核心协议和应用组成,OSI 自底而上共分 5 层,即基带、HCI、L2CAP、SDP 和 BRTATP、应用层。

BRTATP 算法模型如图 3 所示,其主要功能二个:一是通过同步控制器来管理和调度异步流数据,确保 ACL 异步链路能够连续发送实时流媒体数据;二是将媒体数据切割为适当的媒体对象粒度,以满足流式数据的传输^[5]。

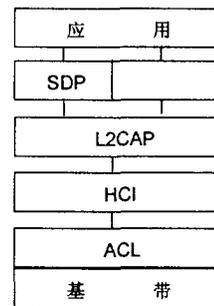


图 2 BRTATP 模型

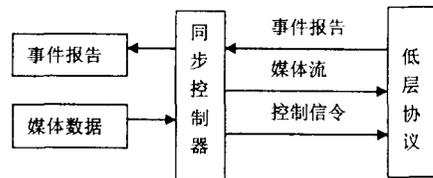


图 3 BRTATP 算法模型

2.2 QoS 与同步

仅从传输速率角度来看,用 ACL 高速传输模式可传输目前许多高质量的音频数据。但是,音频数据的

传输要求很高的连续性和实时性,而 ACL 链路是基于无链接异步数据分组的传输,当链路质量不高时,为了确保服务质量,它对不可靠数据采用了重传机制,重传次数需视链路质量而定。因此,链路的时延和抖动具有不确定性,若媒体对象回放时间 \leq 传输时延,可正常表现;反之,则造成播放的不连续。音频端到端延时最大为 500ms,抖动应小于 10ms。超出这个限度,听众就会有明显的不适感觉。因此,采用 ACL 链路的蓝牙音频应用模型,其关键技术之一在于解决传输链路的时延问题,即媒体流的同步^[4]。

(1) QoS 定义与控制信令格式。

同步要求可以用服务质量(QoS)来表达, QoS 定义为一个六元组参数集:(Sever_Type, Com_Rate, Token_rate, Peak_Bandwith, Latency, Delay_Variation), 其中 Sever_Type: 服务类型,表示所需服务的层次; Com_Rate: 通讯速率,表示一个应用按照该速率连续发送数据;Token_rate: 最大令牌长度,如果达到最大令牌长度,应用就必须等待,或者将数据丢弃;Peak_Bandwith: 带宽峰值,来自应用的分组连续传输速率; Latency: 时延,指源端传输一位和首次无线传输之间的最大可接受延迟;Jitter: 抖动,分组最大延迟时间和最小延迟时间之差,应用采用该值来确定目标端所需缓冲区的大小。其中 Bandwith = DH5 分组速率, Latency = 500ms, Jitter = 10ms。 QoS 控制信令格式如图 4 所示。

类型=0x003	长度(22 字节)	标志(1 字节)	服务类型(1 字节)
通讯速率(4 字节)		最大令牌长度(4 字节)	
峰值带宽(4 字节)	时延(4 字节)	抖动(4 字节)	

图 4 服务质量(QoS)控制信令数据格式

(2) 同步控制算法。

为方便叙述,定义如下: T_m 为一个媒体对象表现时间, T_p 为一个媒体对象传输时间, T_{ack} 为一个媒体对象传送应答时间, T 为一个蓝牙分组标准传送时间, $T = T_p + T_{ack}$, T_l 为最大时延时间, ΔT_i 为 i 个媒体对象的时延, M_o 为媒体对象粒度, Z 为一帧音频数据大小, P 为一个蓝牙分组大小, MTU 为蓝牙最大传输单位。

为确保媒体对象表现的连续性,必须同时满足下述规则。

规则 1: 假设媒体对象粒度 M_o , 则 M_o 必须同时满足:

$$\begin{cases} M_o = n \times Z \\ M_o = m \times MTU \\ n = \text{MOD}(P/Z) \\ m = 1, 2, 3 \dots \end{cases}$$

规则 2: 为了能保证目标节点连续地表现或回放媒体对象,一个媒体对象的表现时间必须不小于媒体对象一次传送时间与应答时间之和,即: $T_m \geq T = T_p + T_{ack}$ 。

规则 3: 由于蓝牙 ACL 链路重传次数具有不确定性,因而它导致了传输时延的不确定性,为了消除媒体对象表现时间与传输时间不匹配时可能造成的表现不连续,目标节点必须预取一定数量的媒体对象粒度 B_{min} , B_{min} 必须满足: $B_{min} \geq T_l/T$ 。

规则 4: 为了能保证目标节点连续地表现或回放媒体对象,源节点第 k 个媒体对象最迟发送时间 t_k 不能迟于 $(k-1)T_m + (B-1) \times T_m -$ 链路总时延,即:

$$T_k \leq (k-1)T_m + (B-1) \times T_m - \sum_{i=1}^{k-1} \Delta T_i$$

规则 5: 为了确保目标节点媒体对象的顺利表现以及源节点不死锁,必须确保单个媒体对象传输最大时间 T_{max} 是有限的,即 $T_{max} \leq T_l$ 。

规则 1 ~ 5 证明略。

同步控制算法概要描述:

Step①: 初始化媒体对象粒度 M_o , M_o 必须满足规则 1 条件;

Step②: 根据链路质量预设缓冲队列 Buffer, Buffer 满足: $\text{Buffer} \geq B_{min} \times M_o$;

Step③: 发送第一个蓝牙分组,同步蓝牙时钟;

Step④: 发送 m 个分组预置缓冲队列, $m = \text{Buffer}/M_o$;

Step⑤: 目标节点按 FIFO 方式连续读取缓冲队列分组,解包;

Step⑥: Read 源节点检测目标节点缓冲队列 Buffer, 准备发送数据分组 Packet_k , Packet_k 为后续分组 $k, k = 1, 2, \dots$ 。

Step⑦: 发送与调整

```
do case (Buffer)
case: Overflow //缓冲队列溢出;
    Wait; //缓冲队列满,则等待;
case: empty //缓冲队列空;
    Send  $\text{Packet}_k$ ; //发送  $\text{Packet}_k$  为后续分组  $k, k = 1, 2, \dots$ 。
    计算  $T_{max}, T_k, T_k = m \times (T_p + T_{ack})$ 。
    If ( $T_{max} \leq T_l$ , and.  $T_k \leq ((k-1)T_m + (B-1) \times T_m - \sum_{i=1}^{k-1} \Delta T_i)$ )
        Send  $\text{Packet}_k$ ; //发送  $\text{Packet}_k, \text{Packet}_k$  为后续分组  $k, k = 1, 2, \dots$ 。
    else if ( $T_{max} \geq T_l$ , and.  $T_k \leq ((k-1)T_m + (B-1) \times T_m - \sum_{i=1}^{k-1} \Delta T_i)$ )
        Abandon  $\text{Packet}_k$ ; //放弃发送分组;
        K = k + 1; //准备下一分组;
```

```

    Return Step⑥; //返回第六步;
else
    return Step④; //返回第四步;
endif
endcase

```

Step ⑧:若媒体传送完毕,则结束;反之,返回第六步。

2.3 数据分组格式

(1) BRTATP 层数据格式。

BRTATP 层数据格式如下:

类型(8),长度(16),媒体数据(0~65509 位),尾(2 位)。

定义:

- ①类型字段:字长 8 位,音频格式说明;
- ②长度字段:字长 12 位,音频有效载荷长度说明;
- ③媒体数据:长度为 0~65509,是音频数据主体;
- ④尾字段:长度 2 位,00 表示媒体数据起始帧,10 表示媒体数据后续帧,11 表示媒体数据结束帧。

表示媒体数据后续帧,11 表示媒体数据结束帧。

(2)L2CAP 数据格式。

L2CAP 基于分组,允许高层协议和应用传输接收长达 64kb 的 L2CAP 数据分组。它遵循基于信道的通讯模型,一条信道代表远程设备上二 L2CAP 实体间的一数据流。信道可以是面向连接的,也可以是无连接的,本算法将采用前者。数据格式如下:

* 长度(2 字节),信道 ID(2 字节),有效载荷(0~65535 字节)。

* 长度:指示了有效载荷(来自上层协议的分组数据)的大小;

* 信道 ID:用于标识分组的目标信道终端。信道 ID 取值与接收分组的设备有关。

* 有效载荷:来自上层协议或者发送到上层协议。用于面向连接的分组 MTU 的最小值将在信道设置期间协商,用于信令分组的 MTU 的最小值为 48 字节^[6]。

(3) HCI 数据格式。

与其它有效物理介质相比,基带协议定义的分组在大小上受到限制,所以大的 L2CAP 分组必须在无线传输之前由 HCI 分段成为多个小基带分组。同样,收到的小基带分组也可以重新组装成为大的单一的 L2CAP 分组。这个过程分为分段与重组(SAR)。

在发端,HCI 将 L2CAP 分组分段成为许多小 ACL 数据格式,对于 L2CAP 分组的第一段,L_CH=10,其余段的 L_CH=01。在收段,HCI 将来自 ACL 层的分组送往 L2CAP 层,L2CAP 层应用则根据分组头中的长度段信息将数据重组成为完整的 L2CAP 分组。

HCI 数据格式如图 5 所示。

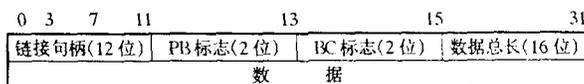


图 5 HCI ACL 数据分组

(4) ACL 层数据格式。

传输高质量的音频信息需要链路高速发送数据,使用 DM5 分组,其数据格式如下:

有效载荷头(2 字节),信息位(最多 339 字节),CRC(2 字节)。

其中有效载荷头数据格式如下:

L_CH(2 位),FLOW(1 位),长度(9 位),未定义(4 位)。

对于 L2CAP 分组起始位,L_CH=10;对于分组延续,L_CH=01。

2.4 数据分组的解封包过程

数据分组解封包过程如图 6 所示,其中数据封装过程为:高层音频数据经过分段打包形成 BRTATP 格式,BRTATP 数据向下进入 L2CAP 层有效载荷字段,L2CAP 层数据经封装后进入 HCI 层数据字段,HCI 层数据经分段,最后进入基带的 ACL 有效载荷字段。数据解封过程与此相逆,ACL 层数据经重组后,向上直至最高层完成的应用层数据的重组^[4]。

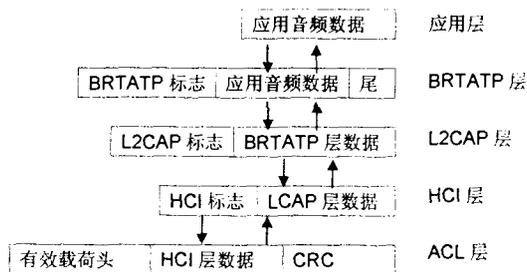


图 6 数据分组解封包过程

3 结束语

文中所提出的模型与算法具有较高的 QoS,音频的表现在一定的范围内有很高的连续性,在“信息家电构架与家居服务器的研制”项目中得以实现,顺利地完成了如 MP3、AC-3 的 5.1 声道和双声道等高质量的音频信号的传输。对于 CD 音频,由于它的速率达 1.41Mb/s,在项目中对它进行了 1:3 的无损固定压缩,因而也得以顺利传输。该模型及其算法主要是利用一块蓝牙模块传输未解包的音频数据,其缺点是目标节点的软件开销较大。因此,将进一步研究已解包音频数据的发送,使之更加适用于低成本、移动式设备的应用。

(下转第 75 页)

删除 NODE 接点,同时调整 NODE 链和 Free 链。

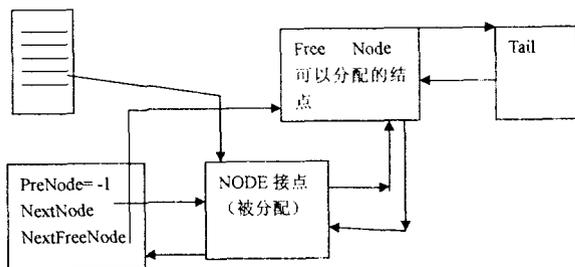


图6 管理器的链式表示状态

大致算法如下:

输入: 一个对象指针。

输出: 对 NODE 链、Free 链的修改。

算法:

(1)把这个对象指针对应的 NODE 接点设置为 Free。

(2)判断此 NODE 的前驱接点是否为 Free,是则合并这两个 NODE 接点。

(3)判断此 NODE 的后继接点是否为 Free,如果是合并这两个接点。

(4)向下调整 freeNode 链,从此接点开始依 NODE 链向下(即每次按 nextNode 往下查)查找第一个 Free 接点,找到则把它设置为此接点的后继 Free 接点。

(5)向上调整 freeNode 链,从 pre 接点开始依 NODE 链向上(即每次按 preNode 往上查)找第一个 Free 接点,找到则把此接点设置为其的后继 Free 接点。

(6)调整 Hash 数组和 Free 数组。

2.4 管理器中查找一对象

在管理器中查找一对象相对较简单,因为不存在对 NODE 链表和 freeNode 链表调整,也不用对 Hash 数组和 DEL 数组进行修改。

输入: 对象描述信息。

输出: 指向此对象的指针,否则为 NULL。

算法:

(1)对对象描述信息进行散列计算出输入值在 Hash 数组中对应的下标,根据这个下标找到图片的存放位置,然后找到一个对象。

(2)如果此对象为空则没找到,否则比较这个对象描述信息与输入的对象描述信息是否相等。

(3)如相等则找到了,返回这个对象,且对这个对象的引用加一。

(4)如果没找到继续执行 Hash(),计算出新的数组下标,重复上面的步骤直到找到要找的图片的地址或返回一个空的。

3 结束语

本内存管理器设计模式已经在一平台无关的播放器库函数中使用,播放器中的一些常用图片,如图标等皆放置管理器中。播放器移植时无需改动播放器本身代码,只需利用本设计模式重新实现本内存管理器。在同一款手机上测试时发现,用本内存管理器能部分提高性能。本模式可以应用到整个系统中,它可以有效的减少系统设计的复杂度和提高整个系统的效率。

文中创新点:提出一种内存管理中间件设计模式,能有效地屏蔽操作系统内存管理的差异,使应用软件可以更方便地在不同操作系统之间移植,同时能方便应用开发中的设计。

参考文献:

- [1] Brooks F P. No Silver Bullet: Essence and Accidents of Software Engineering[J]. IEEE Computer, 1987, 20(4): 10 - 19.
- [2] 杨雷,吴珏,陈汶滨.实时系统中动静结合的内存管理实现[J].微计算机信息,2005(2):15 - 16.
- [3] Johnson R E. Documenting Frameworks using Patterns[C]// Proceedings OOPSLA '92, ACM SIGPLAN Notices. [s. l.]: [s. n.], 1992:63 - 76.
- [4] Gruijs D. A framework of Concepts for Representing Object - Oriented Design and Design Patterns[R]. [s. l.]: Utrecht University, 1997.
- [5] Barr M. C/C++ 嵌入式系统编程[M].于志宏译.北京:中国电力出版社,2001.
- [6] Shull F, Melo W L, Basili V R. An Inductive Method for Discovering Design Patterns from Object - Oriented Software Systems[R]. [s. l.]: University of Maryland, 1996.

(上接第71页)

参考文献:

- [1] 金纯,许光辰,孙睿.蓝牙技术[M].北京:电子工业出版社,2001.
- [2] 吴永忠,韩江洪.蓝牙技术综述[J].微型机与应用,2001(6):60 - 63.
- [3] 吴永忠,韩江洪.蓝牙基带层包格式简析及相关芯片简介[J].计算机应用,2001(10):68 - 70.

- [4] 宋军,顾冠群.多媒体同步[J].中国图象图形学报,1997(6):52 - 55.
- [5] 宣善立,吴永忠,韩江洪.蓝牙技术安全研究[J].计算机工程,2002(9):42 - 45.
- [6] 吴永忠,韩江洪.蓝牙基带层数据包类型简析[C]//全国第十四届计算机科学及其在仪器仪表中的应用学术交流会论文集.合肥:中国科学技术大学出版社,2001.