

基于 UPnP 和 UPnP AV 的多媒体内容同步和回放

张 莉

(南京大学软件新技术国家重点实验室 南京 210008)

(南京大学计算机科学与技术系 南京 210008)

摘 要 简要描述了基于 UPnP 技术架构和 UPnP AV 规范的家庭娱乐系统(HES)的实现方案,在 HES 中提出了设备选择算法、轻量级可切换播放服务和快速同步算法。设备选择算法实现了用户零干预条件下的自动切换,轻量级可切换播放服务提供了一个简单高效的服务实现,而快速同步算法将媒体播放切换动作前的同步时间降低到秒数量级,与传统内容同步方法分钟级的延时相比具有重要的使用价值。

关键词 UPnP, UPnP AV, 内容同步, 多媒体回放

Multimedia Synchronization and Playback Based on UPnP and UPnP AV

ZHANG Li

(The State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210008, China)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210008, China)

Abstract A solution for Home Entertainment System (HES) based on UPnP architecture and UPnP AV specification is described in this paper. In this solution, we create and apply a device selection algorithm, a lightweight switchable playing service and an instant content synchronization algorithm. The device selection algorithm is designed to switch automatically without any user's manual operation while the lightweight switchable playing service provides a simplified and efficient service implementation. The instant content synchronization algorithm for background transfer is developed to reduce the switching delay to within a few seconds. This algorithm demonstrates its practical value when compared with traditional content synchronization, which usually take up to several seconds.

Keywords UPnP, UPnP AV, Content synchronization, Multimedia playback

1 引言

随着科技的发展,越来越多的消费电子设备进入人们的日常生活。其中娱乐电子设备占了很大的比重,便携式音乐播放器、掌上电脑(PDA)和音乐手机等都给用户带来了随身的娱乐体验。另一方面,电视机、家庭影院和家庭影院个人电脑(Home Theater Personal Computer,简称 HTPC)等设备在回放质量、存储能力和处理能力上都有较大的优势。

但是媒体文件在这些存储设备上互相之间是独立的,不同设备之间很难共享或者根本无法共享。为了让相同的文件可以在不同设备之间共享,用户必须手动将文件复制到不同设备上,而这对于用户来说是一个低效、繁琐的过程。另外一个重要的方面,在设备之间切换也需要手动进行。在我们设计开发的家庭娱乐系统(Home Entertainment System,简称 HES)中,正是要为用户提供一个无缝的多媒体娱乐体验,媒体文件只需要在一个联网设备中存在,文件可以自动进行复制和共享,媒体播放进程也可以自动在设备间切换。

2 基于 UPnP 架构的 HES 系统原理

2.1 UPnP 介绍

为解决家庭消费电子设备(CE)之间的互连问题,1994 年 UPnP Forum 提出了 UPnP 架构。它并非对即插即用简单的扩展,而是旨在为种类繁多的设备之间提供零配置、自动组网

能力和自动发现能力。同时,UPnP 也定义了设备间的控制方式和事件通知方式。正如 UPnP 架构的名字所指出的,UPnP 是通用的,这表现在它的设计独立于网络连接方式、操作系统和编程结构等各方面。

为以最小代价获得设备间的最大互连能力,UPnP 架构充分利用了现有的广泛使用的流行协议或技术,例如 IP, TCP, UDP, HTTP 和 XML 等。这使得扩充 UPnP 能力和实现互操作性非常容易。如果引入适当的桥接技术,UPnP 也完全可以工作在非 IP 的设备上,例如 bluetooth, IrDA 或者直接 USB 连接上。

UPnP 架构对以下几个方面进行了定义:

- ①地址获取,包括 DHCP、指定 IP 地址和自动 IP 地址。
- ②设备发现,通过 UDP 协议发现网络上的其他设备。
- ③设备描述,对特定的设备进行查询,以获得设备以及设备所提供服务的描述信息。
- ④设备控制,按照描述信息的规定对设备进行控制,完成需要的动作。
- ⑤事件通知,在自身状态改变的时候通知监控这些状态的设备。

对于 HES 系统来说,UPnP 架构所定义的这些方面非常好地适应了系统功能的需要,地址获取和设备发现等作为互连基础的重要性不言自明,设备描述可以准确地刻画当前系统的播放能力和优先选择。通过设备控制,不同设备之间可

以通过命令协作进行媒体的同步和播放,设备之间状态信息的共享则可以借助事件通知完成。

2.2 媒体播放设备间的发现

媒体播放设备通过向 UPnP 协议规定的组播地址 239.255.255.250:1900 发送以下格式的数据包进行搜索

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: 3
ST: ssdp:all
```

收到搜索命令的设备通过 HTTP 单播发回搜索响应,例如:

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 18 Dec 2006 20:30:44 +0800 (CST)
EXT:
LOCATION: http://192.168.1.55/description? n=root
SERVER: Linux SPS/1.09 UPnP/1.0 MP3Player(Linux)/0.99
ST: ssdp:all
USN: uuid:016389DB-F894-4BCC-9D8C-9EB6F7278782::upnp:rootdevice
```

在一个预先定义的等待时间内,系统接受来自网络上的其他设备的响应并记录,而超过等待时间的响应被直接丢弃。在 HES 系统中,对于所有发出应答的设备,我们提出控制点(Control Point, 简称 CP)可以按照以下设备选择算法自动决定适合播放转移的最佳目标。

①最近最多使用的设备

在所有设备保存的转移历史中找到转移次数最多的设备 D_1 , 启动和设备 D_1 的转移过程。但如果用户在 20s 内又从设备 D_1 转移到设备 D_2 , 设备 D_1 不再将这次转移动作计数。

②优先级最高的设备

这种算法适合设备较少而且用户对于设备的偏好比较固定的情况,例如总是倾向于使用客厅的数字式音响播放音乐文件。当前的播放设备会根据相应设备的描述文件直接选出优先级最高的文件进行播放。

③最先应答设备

处理器负荷最轻的设备或者网络响应速度最快的设备非常可能在最短的时间内做出响应。根据这种算法,直接选择最先响应的设备,其他后续的响应将被直接忽略。

2.3 媒体播放设备的描述文件

当一个控制点发现一个设备的时候,控制点只获得了有限的设备信息。为了让控制点获取更多的设备信息及其能力,或者和该设备进行交互,UPnP 架构定义了设备描述文件和服务描述文件的细节。描述文件以 XML 语言组织,遵从 UPnP Forum 所定义的 UPnP 服务模板。同时,制造商可以按照规定自行定义扩展。通过这种方式,设备或者服务的能力以及控制命令和数据的定义以一种不同设备之间可以解释的共同语言定义。

通常一个 UPnP 设备可能会嵌套多个服务。例如,HTPC 就可以既包含一个媒体存储服务,用来提供一个容量的存储空间,也可以包含一个媒体播放服务,利用安装在其中的应用软件播放媒体文件。设备本身也是可以嵌套的。同样以 HTPC 为例,如果 HTPC 本身安装了打印机或者刻录机,就可以在根设备中嵌套虚拟打印机设备或者虚拟刻录机设备。在 HES 系统中,媒体播放设备基于单一设备模型设计,在系统功能分解的基础上,将功能映射为设备所包含的服务。

3 UPnP AV

UPnP AV 是基于 UPnP 架构之上提出的一个 UPnP CP 和 UPnP AV 设备之间交互的通用性定义,同样是独立于操

作系统和传输协议,并且支持丰富的设备类型,例如电视机、DVD 播放机、机顶盒、MP3 播放器和其他多种数字多媒体播放设备。这个定义主要包括了两类设备,其中提供媒体文件浏览、传输和查找的设备称为 Media Server,而完成渲染(rendering)和控制的设备称为 Media Render。需要注意的是,UPnP AV 中设备的概念是泛化的,可以是一个具体存在的设备,也可以只是某个物理设备中的一部分,两类设备甚至完全可以对应相同的一个物理设备。UPnP AV 定义的服务包括内容目录服务(Content Directory Service,简称 CDS)、连接管理服务(Connection Manager Service,简称 CMS)、AV 传输服务(AV Transport Service,简称 AVTS)和渲染控制服务(Rendering Control Service,简称 RCS),这些服务不一定存在每一类设备中,它们的对应关系如图 1 所示(其中√表示必须支持,○表示可选支持)。

	CDS	CMS	AVTS	RCS
Media Server	√	√	○	
Media Render			○	√

图 1 UPnP AV 定义中设备和服务的对应关系

在 HES 系统中,为了实现播放控制、媒体搜索和内容同步等系统功能,我们选择实现了 UPnP AV v2 规定的 AVTransport, ContentDircorey 和 ConnectionManager 服务。RenderingControl 服务的主要目的在于实现对于声音和图像的控制,为了适应包括各种家用消费电子(Consumer Electronic)设备在内的需求,定义非常复杂,并不完全适合本系统的自动切换媒体播放的目的。为了提供一个简化、高效的播放服务,我们定义了一种轻量级可切换播放服务(Switchable Playing Service,简称 SPS)。各种服务间的逻辑关系如图 2 所示。

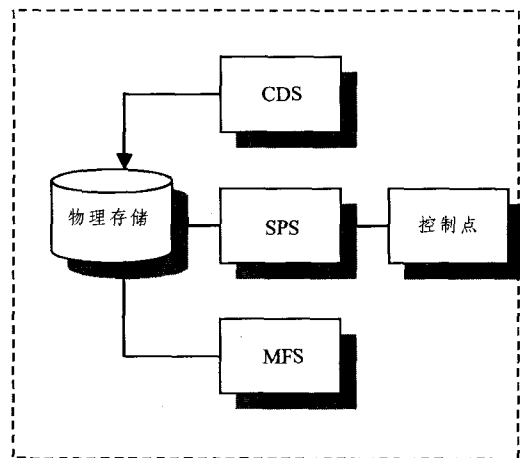


图 2 HES 系统中各种服务的组成及其关系

3.1 可切换播放服务 SPS 的自定义命令

3.1.1 播放列表创建命令(CreatePlayList)

播放列表创建命令在对方的 media player 中建立指定内容的播放列表,其输入参数由一个 base 64 编码的元数据(meta data)组成。HES 系统对元数据的 BNF 定义如下:

```
playListData ::= base64(raw_meta_data_set)
raw_meta_data_set = 1 * ( item_meta_data "0x0d" "0x0a" )
item_meta_data = ( index ";" artist ";" album ";" track ";" title )
artist = <the artist information defined in the dublin core>
```

album=<the album information defined in the dublin core>
track=<the track number>
title=<the title of the audio item>

在 SPS 设备成功创建对应的播放列表后,作为对请求设备的响应,播放列表的实例标识符将被返回给请求设备。

3.1.2 播放恢复命令(ResumePlaying)

播放恢复命令启动对方设备的播放过程,并从当前设备中断的点开始继续。输入参数包括在上一步骤中返回的实例标识符、断点在播放列表中的位置索引及所处的时间位置。例如在当前列表的第 5 个曲目,第 2 分 30 秒继续播放动作将被表示成:

ResumePlaying(current_play_list, 5, "2:30")

3.1.3 获取播放命令(GetPlaylist)

获取播放命令用来取回在对方设备上设置的播放列表数据,以便将播放切换回本设备。动作的参数仅仅包含一个在步骤 1 中返回的实例标识符,返回的数据格式定义和播放列表创建命令中的定义是一致的。

SPS 服务提供内部状态的通知,通过定义当前播放列表的实例标识符、播放文件的曲目编号和曲目的元数据信息。详细定义可参见图 3。

变量名	描述	类型
PlaylistID	播放列表在该设备的唯一标识符	ui4
Current-Item	正在播放文件在该播放列表中的位置索引	ui4
Artist	正在播放文件的艺术家	string
Album	正在播放文件的专辑名称	string
Title	正在播放文件的名称	string
Track	正在播放文件的曲目号码	ui4

图 3 SPS 服务的状态变量列表

3.2 后台内容同步服务

播放切换同时需要解决的另一个问题是多媒体内容的同步。理论上,不同设备之间必须维护相同的多媒体文件集合,以保证当前媒体文件的播放可以从一个设备自由地切换到另外一个设备。然而实际情况常常是发生切换的两个设备属于完全不同的两类范畴,处理能力和存储容量都有极大差异。这种差异决定了两个设备的媒体集合在绝大多数情况下是不同的,即 $S(D_1) \neq S(D_2)$ 。为解决这个问题,我们提出了一种快速同步方法,以保证经过短时间的延时,达到播放列表所包含的文件集合是设备 D_1 和设备 D_2 的文件交集的子集,即 $S(Pl) \in S(D_1) \cap S(D_2)$ 。

快速同步方法利用了媒体播放设备在 SPS 播放过程中空闲的传输服务。如果播放设备的资源情况能够满足两个服务同时运行的要求,那么就可以利用多任务机制在当前文件的播放过程传输其他所需的文件。如果用 $R(item)$ 表示文件 $item$ 的剩余播放时间, $T(item)$ 表示文件 $item$ 的总计播放时间, $size(item)$ 表示文件 $item$ 的大小, Th 表示网络的吞吐能力, $current$ 表示当前文件在播放列表中的位置索引,那么通过选择合适的 n 满足以下条件:

$$R(current) + \sum_{i=current+1}^n T(i) > \frac{size(n+1)}{Th}$$

位置索引在区间 $(current, n)$ 的文件如果在 D_2 上不存在,需要在媒体播放的切换动作之前进行同步传输。在这些文件传输结束之后可以立即切换到对方设备,并从中断点开始播放。之后,可以继续同步位置索引大于 n 的文件。由于 AVTransport 服务中使用带外传输协议(out-of-band proto-

col),通过选择高效率的传输协议(例如 HTTP),可以充分利用网络提供的最大的传输能力。图 4 表示了一个典型的传输切换过程,只有文件 i 需要在切换前同步,其余文件在切换开始后进行。

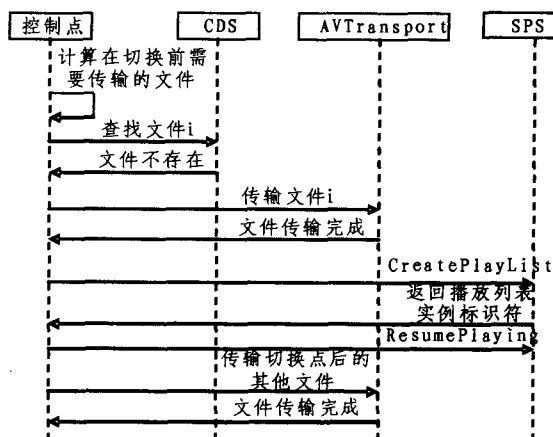


图 4 一个典型的 HES 系统后台同步过程

3.3 实验结果

HES 目前已经在两台 Linux 系统上完成了参考实现。在 HES 系统的性能测试中,测试数据为 20 个 mp3 文件组成的播放列表,平均文件大小为 6M。以此为基础,我们分别在 802.11b 和 802.11g 网络环境下对比测试了传统的串行同步切换和本文所提出的快速同步算法,图 5 所示实验结果表明快速同步算法可以极大地降低传输延时。

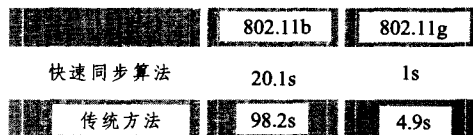


图 5 一个典型的 HES 系统后台同步过程

结束语 基于设备选择算法、轻量级可切换播放服务和快速同步算法的 HES 系统具备了快速内容同步和智能的设备选择功能,从而明显改善用户体验。另一方面, HES 后台运行的文件同步和前台的媒体播放互相并无干扰。下一步计划将 HES 的参考实现移植到安装有嵌入式 Linux 的便携式播放器和机顶盒中。从协议角度来看, UPnP 协议正在逐渐发展中,媒体同步将在 UPnP AV 中进一步得到加强,在进一步跟踪新协议发展的基础上, HES 的多媒体文件同步将会采用标准协议,以获得和 CE 厂商的设备之间更好的互操作性。

参考文献

- [1] The UPnP™ Forum. UPnP™ Device Architecture 1.0, version 1.0.1. <http://www.upnp.org/resources/documents.asp>, December 2003
- [2] Ritchie J, Kuehnel T. UPnP AV Architecture: 1, version 1.00. <http://www.upnp.org/resources/documents.asp>, June 2002
- [3] Ritchie J, Knapen G. ContentDirectory: 2 Service Template Version 1.01. <http://www.upnp.org/resources/documents.asp>, May 2006
- [4] Presser A, et al. AVTransport: 2 Service Template Version 1.01. <http://www.upnp.org/resources/documents.asp>, May 2006
- [5] Fielding R, et al. Hypertext Transfer Protocol - HTTP/1.1. <http://www.ietf.org/rfc/rfc2616.txt>, June 1999
- [6] Goland Y Y, Schlimmer J C. Multicast and Unicast UDP HTTP Messages. <http://tools.ietf.org/html/draft-goland-http-udp-01>, October 2000
- [7] Goland Y Y, et al. Simple Service Discovery Protocol/1.0. <http://tools.ietf.org/id/draft-cai-ssdp-v1-03.txt>, October 1999