

(12) 发明专利

(10) 授权公告号 CN 102291244 B

(45) 授权公告日 2013. 10. 23

(21) 申请号 201110229450. 4

(22) 申请日 2011. 08. 11

(73) 专利权人 湖南康通电子科技有限公司

地址 410003 湖南省长沙市开福区德雅路  
282 号

(72) 发明人 刘雅浪

(74) 专利代理机构 长沙市融智专利事务所

43114

代理人 黄美成

(51) Int. Cl.

H04L 12/18(2006. 01)

H04L 29/08(2006. 01)

H04L 29/06(2006. 01)

审查员 匡仁炳

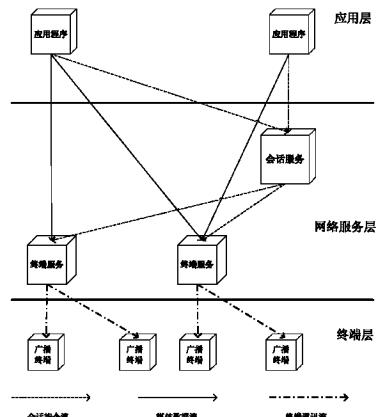
权利要求书2页 说明书12页 附图1页

(54) 发明名称

一种云广播系统及方法

(57) 摘要

本发明公开了一种云广播系统及方法，首先从架构上设置应用层、网络服务层和终端层以构成一个IP广播系统；应用层为提供给用户使用的各种音频应用程序，这些应用程序用于将各种来源的音频流发送到广播终端，或者接收来自广播终端现场采集编码的音频流进行处理；网络服务层用于管理网络中所有会话对象和广播终端；在网络服务层设有会话服务器和终端服务器；终端层由多个广播终端组成，广播终端与终端服务器连接，最终完成声音信号的数模转换。该云广播系统及方法突破了单台服务器性能瓶颈，能支持大规模IP广播网络。



1. 一种云广播方法,其特征在于,从架构上设置应用层、网络服务层和终端层以构成一个 IP 广播系统;

应用层为提供给用户使用的各种音频应用程序,这些应用程序用于将各种来源的音频流发送到广播终端,或者接收来自广播终端现场采集编码的音频流进行处理;

网络服务层用于管理网络中所有会话对象和广播终端;在网络服务层设有会话服务器和终端服务器;

终端层由多个广播终端组成,广播终端与终端服务器连接,最终完成声音信号的数模转换;

会话服务器和终端服务器均为 1 个或多个,应用程序通过会话服务器建立和维护会话,一个应用程序只对应一个会话服务器,一个会话服务器能服务 1 个或多个应用程序;

终端服务器维护广播终端的工作状态,一个广播终端只对应一个终端服务器,一个终端服务器管理一个或多个广播终端;

所述云广播方法中涉及的数据流包括:会话指令流、媒体数据流和终端通讯流:

1) 会话指令流:建立与维护广播会话关系的通讯指令,发生在应用程序与会话服务器、会话服务器与终端服务器之间,应用程序与终端服务器亦从会话指令中获取到媒体流的发送方向;

2) 媒体数据流:承载音频数据,在应用程序、终端服务器和广播终端之间传送;

3) 终端通讯流:终端控制指令,传递终端服务器和广播终端之间的控制信息;

会话服务器通过查询系统数据库,建立起终端服务器与其管理的广播终端之间的连接关系视图,如果一个会话包含多个广播终端,这些广播终端分别与不同的终端服务器连接,会话服务器将会话操作请求分发到相关的终端服务器;

会话服务器定位到网络中所有的广播终端,将应用程序对广播终端的控制请求转发到对应的终端服务器处理;通过会话服务器和终端服务器使得 IP 广播系统获得了可以扩展的能力:通过扩展会话服务器数量,以增加可支持的应用程序的数量;通过扩展终端服务器的数量增加广播终端的数量;

与会话相关的过程包括会话建立过程和会话撤销过程:

会话建立过程为:

步骤 1:应用程序发送邀请消息即 INVITE 消息到与该应用程序连接的会话服务器,在 INVITE 消息中,将音频流发送的目标终端 ID 通知给会话服务器;

步骤 2:会话服务器收到 INVITE 消息后,解析目标终端 ID,通过查询全局配置表,获取到这些终端所连接的终端服务器主机地址与端口;

步骤 3:会话服务器将 INVITE 消息转发给步骤 2 所查询到终端服务器服务主机;

步骤 4:终端服务器收到 INVITE 消息后,修改目标终端会话状态信息,准备好将数据流发送到目标终端,并打开接收音频流的端口;

步骤 5:各个终端服务器发送加入消息即 JOIN 消息给应用程序,通知应用程序接收音频流的地址与端口,报告已经加入到会话中的终端;

步骤 6:应用程序根据配置的策略,在收到一定数量的 JOIN 消息后,启动数据流的传输,将音频数据流直接发送到各个终端服务器上报的媒体流接收端口;

在数据流的传输过程即会话生存期中,应用程序将周期性的发送心跳消息即 HBEAT 消

息到会话服务器，报告会话生存，会话服务器在超过设定时间未收到会话的 HBEAT 消息，将通知终端服务器撤销会话；

会话撤销过程为：

- 1) 应用程序在数据流传输完毕后，发送结束指令即 BYE 指令到会话服务器；
- 2) 会话服务器将 BYE 指令转发至与当前会话关联的终端服务器；
- 3) 终端服务器在收到 BYE 指令后，撤销当前会话；

会话生存期间，终端服务器选择性地主动发送离开指令即 LEAVE 指令给应用程序，通知自身管理的某些广播终端离开会话。

2. 一种云广播系统，其特征在于，包括应用层设备、网络服务层设备和终端层设备；

所述的应用层设备包括音频播放设备；

网络服务层设备包括会话服务器和终端服务器；

终端层设备由多个广播终端组成；

广播终端、会话服务器以及应用层设备均与终端服务服务器连接；会话服务服务器与应用层设备连接；

所述的云广播系统采用权利要求 1 所述的云广播方法进行实现会话和音频数据传输。

3. 根据权利要求 2 所述的云广播系统，其特征在于，应用层设备还包括音频采集设备。

## 一种云广播系统及方法

### 技术领域

[0001] 本发明涉及一种云广播系统及方法。云广播即支持分布式扩展的 IP 网络广播。

### 背景技术

[0002] 广播系统在社会的生产和生活中有着非常广泛的应用,可以快捷方便的以声音的形式将信息传播出去,或者用来作为背景音乐或者教学节目的播放设施,在车站、厂矿、学校、小区、酒店等场所都有着非常广泛的应用。

[0003] 广播系统主要解决音频信息的传输问题。最初的广播系统采用定压功放实现,声音信号直接以模拟电信号传输,音频线上可以驳接多个扬声器。直接传输模拟信号最大的问题在于无法解决传输过程中信号衰减和噪声干扰的问题,传输的距离、接入的广播点数目都受到限制。数码多址广播是在基础定压功放广播系统之上增加频道选择功能的演进产品,本质上就是同时铺设多个音频线,另外外加一路控制信号线,可以控制扬声器选择哪一路音频进行扩音,即增加多路传输特性,但传输的音频信号本身还是模拟信号,无法克服定压功放广播自身固有的缺陷。

[0004] 随着计算机网络技术的普及,计算机网络技术也应用到广播领域。采用 TCP/IP 网络协议传输声音信息的广播系统称之为 IP 广播。在 IP 广播系统传输的声音信号需要先经过采样编码为数字信号,传输的是纯数字信号,数字信号在传输过程中不会失真,信号传输距离不受限制,因此从根本上解决了定压功放在信号传输上的缺陷。IP 广播主要由服务器软件系统、IP 终端设备以及中间的网络传输设备构成,服务器软件将音频流发送到指定终端,终端将接收到的音频流解码还原为模拟信号,驱动喇叭播放声音。

[0005] 随着 IP 广播系统越来越多的应用,人们希望采用 IP 广播技术构建更大规模的广播网络,譬如全省、全地区统一建立的农村广播,连锁机构广播系统,集团公司广播系统等等。现有 IP 广播技术还是基于单服务器架构模式,服务器硬件性能、服务器网络接口带宽限制了网络节点数目,一般单台服务器能够支持的终端节点数目为 200 ~ 500 个。因而,现有的 IP 广播系统扩展性能差。

[0006] 现有 IP 广播系统一般由如下设备构成:

[0007] 1. 一台计算机作为广播服务器,运行系统服务软件,负责处理与系统中所有终端设备的通讯;

[0008] 2. 多个 IP 广播终端设备,接收来自广播服务器的音频流,将其解码还原为模拟音频信号,驱动扬声器输出声音,或者将现场声音进行采集编码,通过网络接口发送到广播服务器;

[0009] 3. 一些客户端应用程序,连接广播服务器,通过服务将音频流发送到指定终端,或者接收来自指定终端的音频流;

[0010] 现有技术的缺点

[0011] 现有 IP 广播技术的问题在于存在服务器性能瓶颈,网络中所有终端都必须由一台服务器进行管理,不能通过扩展服务器数量增加网络容量(服务器之间没有任何关联,

连接不同服务器的客户端和应用程序不能通讯,等同于隔离的网络),当终端数量达到一定量时,服务器不能保证及时处理与所有终端的通讯,声音播放不再流畅。

## 发明内容

[0012] 本发明所要解决的技术问题是提供一种云广播系统及方法,该云广播系统及方法可以方便灵活地扩展 IP 广播系统的规模。

[0013] 本发明为解决上述技术问题所采用的技术方案是:

[0014] 一种云广播方法,其特征在于,从架构上设置应用层、网络服务层和终端层以构成一个 IP 广播系统;

[0015] 应用层为提供给用户使用的各种音频应用程序,这些应用程序用于将各种来源的音频流发送到广播终端,或者接收来自广播终端现场采集编码的音频流进行处理;

[0016] 网络服务层用于管理网络中所有会话对象和广播终端;在网络服务层设有会话服务器和终端服务器;

[0017] 终端层由多个广播终端组成,广播终端与终端服务器连接,最终完成声音信号的数模转换。

[0018] 会话服务器和终端服务器均为 1 个或多个,应用程序通过会话服务器建立和维护会话,一个应用程序只对应一个会话服务器,一个会话服务器能服务 1 个或多个应用程序;

[0019] 终端服务器维护终端设备及广播终端的工作状态,一个终端设备只对应一个终端服务器,一个终端服务器管理一个或多个终端设备;

[0020] 所述 IP 广播方法中涉及的数据流包括:会话指令流、媒体数据流和终端通讯流:

[0021] 1) 会话指令流:建立与维护广播会话关系的通讯指令,发生在应用程序与会话服务器、会话服务器与终端服务器之间,应用程序与终端服务器亦从会话指令中获取到媒体流的发送方向;

[0022] 2) 媒体数据流:承载音频数据,在应用程序、终端服务器和广播终端之间传送;

[0023] 3) 终端通讯流:终端控制指令,传递终端服务器和广播终端之间的控制信息;

[0024] 会话服务器通过查询系统数据库【系统数据库一般设置在网络层,它是系统中各个模块相互交流的数据中心,存储数据包括:终端配置数据、会话状态数据、终端服务配置数据、会话服务配置数据、用户管理数据;】建立起终端服务器与其管理的终端设备之间的连接关系视图【附图 1 中显示了终端服务器与其管理的终端设备之间的连接关系视图】,如果一个会话包含多个广播终端,这些广播终端分别与不同的终端服务器连接,会话服务器将会话操作请求分发到相关的终端服务器。

[0025] 会话服务器定位到网络中所有的广播终端,将应用程序对广播终端的控制请求转发到对应的终端服务器处理;通过会话服务器和终端服务器使得 IP 广播系统获得了可以扩展的能力:通过扩展会话服务器数量,以增加可支持的应用程序的数量;通过扩展终端服务器的数量增加终端设备的数量。

[0026] 与会话相关的过程包括会话建立过程和会话撤销过程:

[0027] 会话建立过程为:

[0028] 步骤 1:应用程序发送邀请消息即 INVITE 消息到与该应用程序连接的会话服务器,在 INVITE 消息中,将音频流发送的目标终端 ID 通知给会话服务器;

- [0029] 步骤 2 :会话服务器收到 INVITE 消息后,解析目标终端 ID,通过查询全局配置表,获取到这些终端所连接的终端服务器主机地址与端口 ;
- [0030] 步骤 3 :会话服务器将 INVITE 消息转发给步骤 2 所查询到终端服务器服务主机 ;
- [0031] 步骤 4 :终端服务器收到 INVITE 消息后,修改目标终端会话状态信息,准备好将数据流发送到目标终端,并打开接收音频流的端口 ;
- [0032] 步骤 5 :各个终端服务器发送加入消息即 JOIN 消息给应用程序,通知应用程序接收音频流的地址与端口,报告已经加入到会话中的终端 ;
- [0033] 步骤 6 :应用程序根据配置的策略,在收到一定数量【一般取目标终端总数的百分之三十】的 JOIN 消息后,启动数据流的传输,将音频数据流直接发送到各个终端服务器上报的媒体流接收端口 ;
- [0034] 在数据流的传输过程即会话生存期中,应用程序将周期性的发送心跳消息即 HBEAT 消息到会话服务器,报告会话生存,会话服务器在超过设定时间未收到会话的 HBEAT 消息,将通知终端服务器撤销会话 ;
- [0035] 会话撤销过程为 :
- [0036] 1) 应用程序在数据流传输完毕后,发送结束指令即 BYE 指令到会话服务器 ;
- [0037] 2) 会话服务器将 BYE 指令转发至与当前会话关联的终端服务器 ;
- [0038] 3) 终端服务器在收到 BYE 指令后,撤销当前会话 ;
- [0039] 会话生存期间,终端服务器选择性地主动发送离开指令即 LEAVE 指令给应用程序,通知自身管理的某些广播终端离开会话。
- [0040] 一种云广播系统,包括应用层设备、网络服务层设备和终端层设备 ;
- [0041] 所述的应用层设备包括音频播放设备 ;
- [0042] 网络服务层设备包括会话服务器和终端服务器 ;
- [0043] 终端层设备由多个广播终端组成 ;
- [0044] 广播终端、会话服务器以及应用层设备均与终端服务服务器连接 ;会话服务服务器与应用层设备连接 ;
- [0045] 所述的云广播系统采用前述的云广播方法进行实现会话和音频数据传输。
- [0046] 应用层设备还包括音频采集设备。
- [0047] 广播属于一对多的通信模式,和电话通信有所区别,电话通信属于一对一的通信,必须确保双方都能接收到对方的语音流,广播通信存在多个数据接收者 (SA),不必保证每个接收者都能接收到的数据流,接收者也可以选择何时开始接收数据流,将会话建立分为“邀请”和“加入”两个阶段,这样便将各个 SA 的行为独立开来,可以比较灵活的处理多方通信的场景。名词说明
- [0048] Session :会话,如无特殊说明,本专利中专指广播会话,广播会话表示一个广播事务,即多个通信节点之间进行一次信息传递所包含的从发起、执行到结束的过程由某个发起者发起的 ;
- [0049] Session Server :SessionSvr,会话服务程序,系统中管理会话信息的服务程序 ;负责将会话请求发送到被邀请的网络节点 ;
- [0050] Terminal Server :TRMSvr,终端服务程序,系统中管理终端设备工作状态的服务程序 ;

[0051] Session Protocol :会话通讯协议,会话协议定义会话建立和运行期间参与者之间消息报文格式以及邀请、关闭、加入、离开等会话事务的执行过程；

[0052] 有益效果：

[0053] 本发明的云广播系统及方法,通过部署多个服务器支持大规模 IP 广播网络,使系统可以平行扩展。本发明建立分布式架构,将管理终端的负载分担多个服务上,保障与每个终端的通讯能得到及时处理。

[0054] SessionSvr 为 App 屏蔽了网络部署结构细节,应用不需要知道各个终端的具体分布,以及各个 TRMSvr 与终端之间的对应关系,只需将会话请求发送给 SessionSvr 处理即可,便于用户使用。

[0055] 本发明的优点如下：

[0056] 1. 良好的可扩展性：由分层设计、会话状态与终端状态分布存储与维护、标准会话管理协议设计使得系统具备良好的可扩展性。每一个终端,每一次会话,在服务程序都需要分配相应的 CPU 和内存资源对其进行逻辑运算和状态保存,因此计算机硬件性能 (CPU 计算能力,内存大小,出口带宽等) 限制了单台服务最大能支持的终端数量和会话数量。虽然当前 IP 广播系统通过增加服务器,可以增加支持终端数量,但连接不同服务器的终端之间,不同的服务器之间是彼此完全独立的,不能构成一个互联互通的网络,因此只能是彼此毫无关系的小规模广播系统。“云广播”系统 (即本发明的 IP 广播系统) 将终端状态和会话状态数据分布到不同的服务主机保存和处理,各个服务主机之间通过会话协议进行通讯,构成了互联互通的网络系统,通过扩展服务器数量,便可以扩大网络规模。

[0057] 2. 会话管理协议比较好的描述了广播会话各个阶段的操作,依据标准协议,可以比较方便的在核心服务系统之上建立各种应用。

## 附图说明

[0058] 图 1 是本发明的架构图。

## 具体实施方式

[0059] 下面结合附图和具体实施方式对本发明作进一步说明。

[0060] 实施例 1：

[0061] 如图 1,整个系统划分为三个层次,从上至下依次为 :应用层、网络服务层,终端层。其中网络服务层为分布式 IP 广播系统核心组件,为上层应用屏蔽了终端设备的分布细节。

[0062] 系统分层说明

[0063] 应用层 :应用层为提供给用户使用的各种音频应用程序,这些应用程序完成将各种来源的音频流发送到终端设备,或者接收来自终端设备现场采集编码的音频流进行处理。这些应用程序包括音频文件播放、电脑实时采集播放、报警广播、内部对讲、定时广播等；

[0064] 网络服务层 :网络服务层主要包括会话服务 (SessionSvr) 和终端服务 (TRMSvr) 两个模块,这两个模块起到管理网络中所有会话对象和终端设备的作用,是构建分布式 IP 广播系统的核心组件；

[0065] 终端层 :这一层为终端设备,与上层的终端服务连接,最终完成声音信号的数模转换 ;

[0066] 系统中的实体主要包括应用 (App) 【这里的应用就是和广播系统服务 (SessionSvr, TRMSvr) 有关联的应用程序,一个应用就是一套程序,但可以部署到多个电脑上运行。】、会话服务 (SessionSvr)、终端服务 (TRMSvr)、终端 (Terminal),它们之间的关系如下 :【会话服务 (SessionSvr)、终端服务 (TRMSvr) 对应的实体或者硬件设备就是服务器】

[0067] 这里的终端指网络音频编解码设备,是类似电视机顶盒的多媒体网络设备,主要构成为 MCU( 微处理器 )、网卡芯片等硬件和运行在上面的嵌入式软件构成。

[0068] 会话服务和终端服务都可以部署多个。每个终端服务管理一定数量的终端,每个会话服务负责处理一定数量的应用接入请求,根据全局的配置信息,会话服务可以定位到网络中所有的终端服务,将应用对终端的控制请求转发到对应的终端服务处理。

[0069] 4) App 通过 SessionSvr 建立和维护会话,一个 App 只对应一个 SessionSvr,一个 SessionSvr 可以服务多个 App ;

[0070] 5) TRMSvr 维护终端设备的工作状态,一个终端设备只对应一个 TRMSvr,一个 TRMSvr 可以管理多个终端 ;

[0071] 6) 系统中的数据流主要包括 :会话指令流,媒体数据流,终端通讯流。

[0072] a) 会话指令流 :建立与维护广播会话关系的通讯指令,发生在 App 与 SessionSvr、SessionSvr 与 TRMSvr 之间,App 与 TRMSvr 亦从会话指令中获取到媒体流的发送方向 ;

[0073] b) 媒体数据流 :承载音频数据,在 App, TRMSvr, Terminal 之间传送 ;

[0074] c) 终端通讯流 :终端控制指令,传递 TRMSvr 和 Terminal 之间的控制信息 ;

[0075] 7) SessionSvr 通过查询系统数据库,建立起 TRMSvr 与其管理终端之间的连接关系视图,如果一个会话包含多个终端,这些终端分别与不同的 TRMSvr 连接,SessionSvr 将会话操作请求分发到相关的 TRMSvr。通过 SessionSvr 和 TRMSvr,系统获得了可以扩展的能力 :通过扩展 SessionSvr 服务器数量,系统可以增加可支持应用的数量 ;通过扩展 TRMSvr 服务器数量,可以增加终端设备的数量,一般的,终端设备数量是网络规模的主要指标 ;

[0076] i. 会话协议 (Session Protocol, SP)

[0077] 会话协议是建立分布式广播会话的表达方式,是“云广播”系统的关键技术之一。会话协议参考在互联网服务中广泛使用的 http 协议,是一种文本协议,具备良好的可读性和可扩展性,包含请求和应答两种类型的报文。

[0078] 协议报文格式

[0079] 1. 请求格式 :

[0080] Method SP Request-Uri SP Version CRLF

[0081] message-header\*

[0082] CRLF

[0083] [message-body]

[0084] 其中 SP 为字段之间的间隔符,使用一个或者多个空格, CRLF 为 “\r\n”;

[0085] Method : 请求方法, 表示是何种请求, 可以是 “INVITE”, “CANCEL”, “JOIN”, “LEAVE”, “BYE”, 各个方法的具体意义下面章节再详细说明 ;

[0086] Request-Uri : 请求发送目标节点 URI, 格式为 sp:node-id, 譬如 :sp:ss.domain.

com

[0087] Version : 协议版本, 当前为 “bsp/1.0”

[0088] message-header : 消息头部, 每行为一个字段, 可以有零项或者多项, 承载请求的参数, 字段格式后面说明 ;

[0089] message-body : 消息正文, 保存消息扩展内容, 保存体积较大, 格式复杂, 不适合在头部承载的参数。Message-body 的长度由 Header 中 “Content-Length” 字段说明, 类型由 “Content-Type” 字段说明 ;

[0090] 字段格式 : FieldName COLON SP\*FieldValue CRLF, 其中 COLON 为 ‘:’, COLON 和字段值之间可以有零个或者多个空格 ;

[0091] 2. 应答格式 :

[0092] Version SP Status-code SP Reason-Phrase CRLF

[0093] message-header\*

[0094] CRLF

[0095] [message-body]

[0096] Respond 中除第一行外, 其它部分格式与 Request 相同, 这里只对第一行进行说明。

[0097] Version : 协议版本, 当前为 “sp/1.0”

[0098] Status-code : 请求处理返回状态码, 表示接收者对请求的处理情况, 具体编码后面章节说明 ;

[0099] Reason-Phrase : 处理状态描述信息 ;

[0100] 会话协议中的参与者

[0101] 1. Session Initiator (SI), 会话发起者 : 发起建立会话, 并负责维护整个会话的生命周期, 在本系统中, 各种应用即承担了 SI 的角色 ;

[0102] 2. Session Server (SS), 会话服务 : 在会话服务上, 保存了当前有效的会话状态数据, 在本系统中, 各个 SessionSvr 实例即为 SS ;

[0103] 3. Session Agent (SA), 会话代理 : 会话代理代理其它不支持 SP 协议的设备或者应用执行各种会话操作, 并将会话请求转化为被代理对象能够处理的操作, 在本系统中, TRMSvr 即为 SA, 代理终端设备处理会话请求 ;

[0104] 会话协议主要内容

[0105] 会话协议包括如下命令 : INVITE, BYE, JOIN, LEAVE, HBEAT。其中 INVITE 建立会话或者邀请新的终端设备加入会话, BYE 撤销会话, JOIN 表示 SA 请求加入会话, LEAVE 表示 SA 通知离开会话, HBEAT 由 SI 周期性发给 SS, 表示会话处于活动状态。

[0106] INVITE 命令内容

[0107] SI (或者 SI 的代理者) 通过 INVITE 消息对 SA 节点发出会话邀请, 建立起会话。会话相关请求主要包括 INVITE, BYE, JOIN, LEAVE 这 4 个, 下面将详细说明其请求和应答格式, 以及处理流程。

[0108] INVITE 可以邀请新的终端加入已经建立的会话 ;

[0109] 请求 :

[0110] Request :

- [0111] -----
- [0112] INVITE SS-URI sp/1.0
- [0113] From :SIUri
- [0114] To :TargetID1, TargetID2, ...
- [0115] CSeq :nnnn INVITE
- [0116] SessionId :{uuid of session}
- [0117] Appdata :
- [0118] 头部字段说明：
- [0119] 1) From :会话发起用户 ID
- [0120] 2) To :会话目标节点 ID, 各个目标 ID 之间用 ‘,’ 隔开；
- [0121] 3) CSeq :请求序列号, 32 位无符号整数, 在一个会话内, 在同一方向上的请求序列号单调递增；
- [0122] 4) SessionId :会话 ID, 必须保证全局唯一；
- [0123] 5) Appdata :应用自定义数据, 交由 SS 统一维护；
- [0124] 应答：
- [0125] INVITE 请求的应答有两种, 一是 SI 发送 INVITE 请求给 SS 后, SS 给 SI 的应答 (Respond1); 另外一个是 SS 解析完会话目标 SA 集合后, 各个 SA 发送给 SS 的应答 (Respond2)。
- [0126] Respond1 :
- [0127] Respond1 :SS → SI
- [0128] -----
- [0129] sp/1.0 Status-Code Reason-Phrase
- [0130] [Via :xxxxx]
- [0131] CSeq :nnn INVITE
- [0132] SessionId :
- [0133] Agents :
- [0134] 字段说明：
- [0135] ■ CSeq :与请求中 CSeq 字段相同；
- [0136] ■ Agents :目标 SA ID 清单 (即目标终端所属区号清单)；
- [0137] Respond2 :
- [0138] Respond2 :SA → SS
- [0139] -----
- [0140] sp/1.0 Status-Code Reason-Phrase
- [0141] CSeq :nnn INVITE
- [0142] SessionId :
- [0143] From :SAUri

- [0144] 字段说明 :
- [0145] ■ CSeq :与请求中 CSeq 字段相同 ;
- [0146] ■ AgentId :SA 节点 ID( 区号 )
- [0147] BYE 命令内容
- [0148] SI 发送 BYE 请求,通知参与者结束会话。SS 和 SA 接收到 BYE 请求时,撤销本地存储的会话数据 ;
- [0149] 请求
- [0150] Request :
- [0151] -----  
-----
- [0152] BYE SS-URI sp/1.0
- [0153] [Via :transport-of-proxy
- [0154] Via :... ]
- [0155] From :SIUriCSeq:nnnn BYE
- [0156] SessionId :
- [0157] [To :trmid, trmid, ... ]
- [0158] 请求发送路径 :SI → SS → SA1, SA2, ...
- [0159] 应答
- [0160] Respond1 :SS → SI
- [0161] -----  
-----
- [0162] sp/1.0 Status-Code Reason-Phrase
- [0163] [Via... ]
- [0164] CSeq :nnn BYE
- [0165] SessionId :
- [0166] Respond2 :SA → SS
- [0167] -----  
-----
- [0168] sp/1.0 Status-Code Reason-Phrase
- [0169] CSeq :nnn BYE
- [0170] SessionId :
- [0171] From :SAUri
- [0172] JOIN 命令内容
- [0173] SA( 或者 SA 的代理者 ) 发送 JOIN 请求给 SI, 报告加入已存在的会话, 并告知自己媒体数据接收设置。SA 可以在收到 SI 的 INVITE 请求后加入到会话中, 也可以直接向 SI 发送 JOIN 请求( 在获取到 SessionId 的情况下 )。
- [0174] 请求
- [0175] Request :SA → SS → SI
- [0176] -----

---

[0177] JOIN SS-URI sp/1.0  
[0178] [Via :transport-of-proxy  
[0179] Via :... ]  
[0180] From :AgentUri  
[0181] CSeq :nnnn JOIN  
[0182] SessionId :  
[0183] Receive-Port :recv-address:recv-port  
[0184] Targets :  
[0185] 头部字段说明：  
[0186] ■ Receive-Port :RTP 数据流接收端口；  
[0187] ■ Targets :加入会话的目标终端号码清单，如果为 AAA0000，则表示包括全区所有终端，多个号码之间用‘,’隔开；  
[0188] 应答  
[0189] Respond :SI → SS → SA  
[0190] -----  
-----

[0191] sp/1.0 Status-Code Reason-Phrase  
[0192] [Via :transport-of-proxy  
[0193] Via :... ]  
[0194] From :AgentUri  
[0195] CSeq :nnn JOIN  
[0196] SessionId :  
[0197] Owner :  
[0198] SessionAttr :  
[0199] Media :  
[0200] Targets :  
[0201] 头部字段说明：  
[0202] ■ Owner :Uri of SI  
[0203] ■ SessionAttr :Session Attr  
[0204] ■ Targets :新加入会话的终端 ID 清单；  
[0205] LEAVE 命令内容  
[0206] SA(或者 SA 代理者) 向 SI 发送 LEAVE 请求，报告离开会话的终端清单。  
[0207] 请求  
[0208] Request :SA → SS → SI  
[0209] -----  
-----

[0210] JOIN SS-URI sp/1.0  
[0211] CSeq :nnnn LEAVE

- [0212] SessionId :
- [0213] From :AgentUri
- [0214] Targets :
- [0215] 头部字段说明 :
- [0216] ■ Target-num :当前还存在于会话中的终端数量 ;
- [0217] ■ Targets :离开会话的目标终端号码清单 ;
- [0218] 应答
- [0219] Respond :SI → SS → SA
- [0220] -----
- [0221] -----
- [0221] sp/1.0 Status-Code Reason-Phrase
- [0222] CSeq :nnn LEAVE
- [0223] SessionId :
- [0224] From :AgentUri
- [0225] Targets :
- [0226] ■ Targets :新退出会话的终端 ID 清单。
- [0227] HBEAT 命令内容
- [0228] SI 定期向 SS 发送的心跳请求,报告自己的状态以及当前发起的会话 ID 清单,如果 SS 长时间没有收到 SI 的 HBEAT 请求,则认为 SI 状态异常,由该 SI 发起的会话已失效。
- [0229] 请求
- [0230] Request :SI → SS
- [0231] -----
- [0232] -----
- [0232] HBEAT SS-URI sp/1.0
- [0233] Via :transport-of-SI
- [0234] From :Session-Owner-UserID
- [0235] CSeq :nnnn HBEAT
- [0236] Sessions :session-id1, session-id2
- [0237] 头部字段说明 :
- [0238] ■ Sessions :SI 当前拥有会话 ID 清单 ;SI 如果通过 NAT 设备与 SS 连接,可以先设置 Sessions 为空串,发送 HBEAT 请求至 SS,获得自己今后 NAT 映射后的出口地址 ;
- [0239] 应答
- [0240] Respond :SS → SI
- [0241] -----
- [0242] -----
- [0242] sp/1.0 Status-Code Reason-Phrase
- [0243] CSeq :nnn HBEAT
- [0244] {NATAddr :xxx. xxx. xxx. xxx:nnnn}
- [0245] 头部字段说明 :

- [0246] NATAddr :SS 将实际收到 SI 请求的数据源地址、端口发回到 SI, 以便 SI 获取到自己的 NAT 映射端口, 可选 ;
- [0247] 会话建立与撤销过程
- [0248] 1. SI 发出 INVITE 消息到 SS, INVITE 消息中包含需要加入到会话中的终端编号清单 ;
- [0249] 2. SS 接收到 SI 的 INVITE 消息后, 解析会话所包含的终端 ID, 通过全局配置信息, 获取到管理这些终端的 TRMSvr 服务地址与端口, 将 INVITE 消息转发到这些 TRMSvr (SA) ;
- [0250] 3. TRMSvr 接收到 INVITE 消息后, 根据各自现实情形与策略配置, 决定是否加入会话, 以及决定自己所哪些终端可以加入到会话中。如果存在需要加入会话的终端, 则发送 JOIN 消息到 SS, 由 SS 转发至 SI ;
- [0251] 4. SI 根据接收到的 JOIN 请求, 可以获知哪些受邀对象已经加入到会话中, 如果加入到会话中的终端设备数量满足策略需求, 则启动媒体流的传输 ;
- [0252] 5. 在会话进行过程中, SA 可以通过主动发送 JOIN, LEAVE 加入或者离开会话, SI 也可以通过主动向其它 SA 发送 INVITE 请求, 将该 SA 管理的设备加入到会话中 ;
- [0253] 6. 在会话进行过程中, SI 周期性发送心跳请求 (HBEAT 命令) 到 SS, SS 如超过规定时间未收到 SI 的心跳, 则认为会话已异常终止, 发送 BYE 命令到各个 SA, 请它们退出会话 ;
- [0254] 7. 当媒体流传输完毕, SI 发送 BYE 命令到 SS, 通知会话结束, SS 负责转发 BYE 命令到其它会话参与者 ;
- [0255] 会话建立过程 :
- [0256] 1. App 发送 INVITE 消息到与之连接的 SessionSvr ( 每个应用可以配置不同的 SessionSvr 为之服务 ), 在 INVITE 消息中, 将音频流发送的目标终端 ID 通知给 SessionSvr ;
- [0257] 2. SessionSvr 收到 INVITE 消息后, 解析目标终端 ID, 通过查询全局配置表, 获取到这些终端所连接的 TRMSvr 主机地址与端口 ;
- [0258] 3. SessionSvr 将 INVITE 消息转发给步骤 2 所查询到 TRMSvr 服务主机 ;
- [0259] 4. TRMSvr 收到 INVITE 消息后, 修改目标终端会话状态信息, 准备好将数据流发送到目标终端, 并打开接收音频流的端口 ;
- [0260] 5. 各个 TRMSvr 发送 JOIN 消息给 App, 通知 App 接收音频流的地址与端口, 报告已经加入到会话中的终端 ;
- [0261] 6. App 根据配置的策略, 在收到一定数量的 JOIN 消息后, 启动数据流的传输, 将音频数据流直接发送到各个 TRMSvr 上报的媒体流接收端口 ;
- [0262] 7. 在数据流的传输过程中 ( 会话生存期 ), App 将周期性的发送 HBEAT 消息到 SessionSvr, 报告会话生存, SessionSvr 在超过设定时间未收到会话的 HBEAT 消息, 将通知 TRMSvr 撤销会话 ;
- [0263] 会话撤销过程
- [0264] 8) App 在数据流传输完毕后, 发送 BYE 指令到 SessionSvr ;
- [0265] 9) SessionSvr 将 BYE 指令转发至与当前会话关联的 TRMSvr ;
- [0266] 10) TRMSvr 在收到 BYE 指令后, 撤销当前会话, 终端可能切换到其它会话 ( 发送其

它会话音频流到终端播放)或者空闲;

[0267] 11) 会话生存期间, TRMSvr 也可以主动发送 LEAVE 指令给 App, 通知自身管理的某些终端离开会话(可能需要加入到其它会话中, 或者在某些终端需要临时屏蔽当前会话音频流);

[0268] 见会话建立过程说明, 为提高网络传输效率, 避免出现网络瓶颈, 音频流由应用直接发送给 TRMSvr, 再由 TRMSvr 根据终端所在网络的情况, 通过单播或者组播方式将数据流传输到终端。

[0269] 这里说一下“云广播”系统即云广播系统可能的一种应用场景。譬如农村广播, 假定某市建立了统一的农村 IP 广播系统, 每个县分配一个 TRMSvr, 管理全县的终端设备, 全市建立 2 台 SessionSvr, 每 10 个县分配一台 SessionSvr(假定共有 20 个县), 县和市都有广播站, 广播站应用程序具备音乐播放和实时广播讲话的功能。这些功能都依托“云广播”系统来实现, 通过系统权限管理, 各县广播站应用只允许访问本县终端, 市广播站应用允许访问全市终端设备。广播站应用程序读取所运行电脑上存储的音乐文件, 或者通过电脑 MIC 实时采集编码音频流, 通过 SessionSvr 建立会话, 将音频流发送到 TRMSvr 通过 JOIN 消息上报的数据流接收端口, TRMSvr 再将音频流转发至目标终端设备, 终端将音频流解码还原为声音进行播放。

[0270] 本发明的协议参考了以下资料:《RFC :3261》, SIP 协议标准。

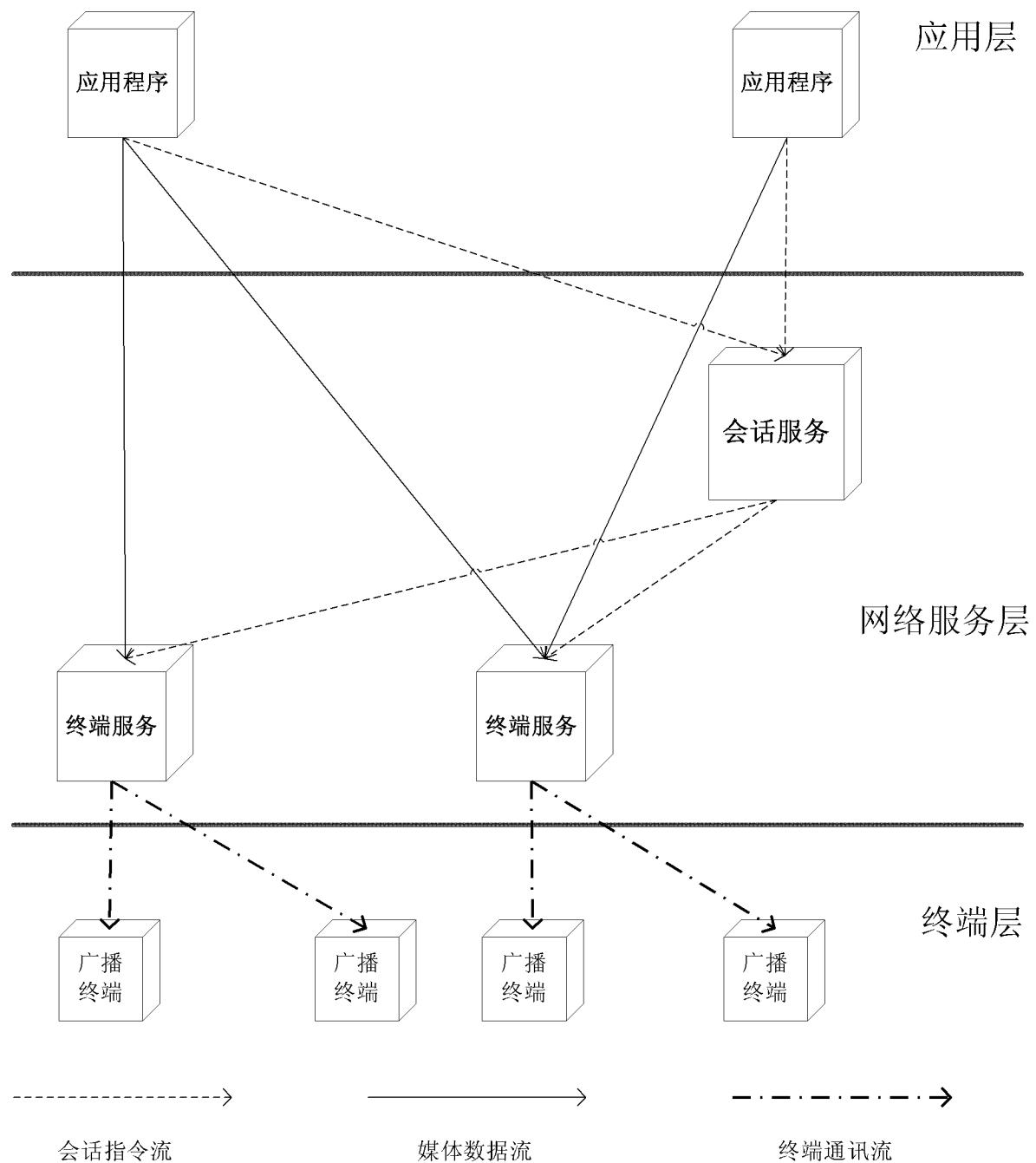


图 1