



(12)发明专利

(10)授权公告号 CN 102857478 B

(45)授权公告日 2016.09.28

(21)申请号 201110182112.X

(56)对比文件

(22)申请日 2011.06.30

CN 101236567 A, 2008.08.06,

(65)同一申请的已公布的文献号

CN 1762155 A, 2006.04.19,

申请公布号 CN 102857478 A

US 7055169 B2, 2006.05.30,

(43)申请公布日 2013.01.02

US 7895350 B1, 2011.02.22,

(73)专利权人 华为技术有限公司

CN 101399684 A, 2009.04.01,

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

审查员 陈莹

(72)发明人 张园园 乐培玉 石腾 惠宇

(74)专利代理机构 深圳市深佳知识产权代理事务所(普通合伙) 44285

代理人 彭愿洁 李文红

(51)Int.Cl.

H04L 29/06(2006.01)

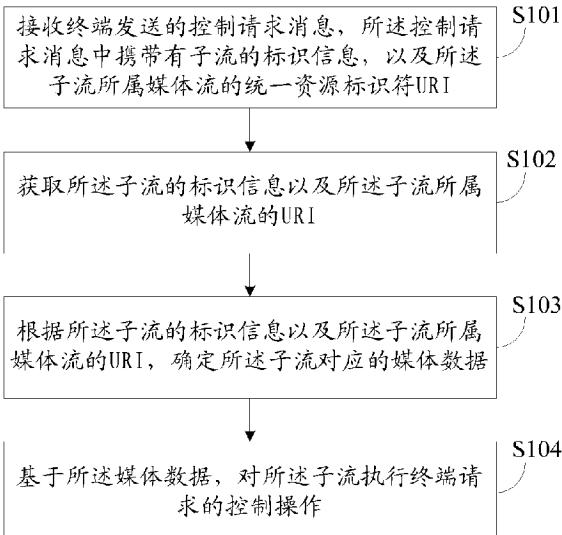
权利要求书4页 说明书21页 附图5页

(54)发明名称

媒体数据控制方法及装置

(57)摘要

本发明公开了媒体数据控制方法及装置，其中一种方法包括：接收终端发送的控制请求消息，所述控制请求消息中携带有子流的标识信息，以及所述子流所属媒体流的统一资源标识符URI；获取所述子流的标识信息以及所述子流所属媒体流的URI；根据所述子流的标识信息以及所述子流所属媒体流的URI，确定所述子流对应的媒体数据；基于所述媒体数据，对所述子流执行终端请求的控制操作。通过本发明，能够在将同一种类型下不同特性的数据通过同一个媒体流进行传输的情况下，实现对不同特性数据的独立控制。



1. 一种媒体数据控制方法,其特征在于,包括:

接收终端发送的控制请求消息,所述控制请求消息中携带有子流的标识信息,以及所述子流所属媒体流的统一资源标识符URI;获取所述子流的标识信息以及所述子流所属媒体流的URI;

根据所述子流的标识信息以及所述子流所属媒体流的URI,确定所述子流对应的媒体数据;

基于所述媒体数据,对所述子流执行终端请求的控制操作;

其中,所述媒体流包括基本层和至少一个增强层;所述子流为所述媒体流的一部分且可以独立解码,包括所述基本层或所述基本层和至少一个所述增强层。

2. 根据权利要求1所述的方法,其特征在于,当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息通过所述控制请求消息的头域携带,所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于多个媒体流的合控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的头域携带。

3. 根据权利要求1或2所述的方法,其特征在于,所述接收终端发送的控制请求消息之前还包括:

接收终端发送的描述信息请求消息;

向所述终端返回携带有子流声明信息的响应消息,以便所述终端通过所述子流声明信息获取所述子流的标识信息。

4. 根据权利要求1或2所述的方法,其特征在于,所述获取所述子流的标识信息之后,该方法还包括:

确定所述控制请求消息中携带的子流的标识信息错误,向所述终端返回携带有子流声明信息的响应消息,以便所述终端根据所述响应消息中携带的子流声明信息重新获取子流的标识信息,并重新发送控制请求消息。

5. 根据权利要求1或2所述的方法,其特征在于,所述控制请求消息中还携带有子流控制特征标签,所述方法还包括:

获取所述子流控制特征标签;

如果能正确识别所述子流控制特征标签,则继续执行所述获取所述子流的标识信息以及所述子流所属媒体流的URI及后续操作,否则拒绝所述控制请求,向终端返回携带有不支持子流控制的信息的响应消息。

6. 根据权利要求1或2所述的方法,其特征在于,所述接收所述终端发送的控制请求消息之前还包括:

接收所述终端发送的RTSP SETUP请求消息,所述RTSP SETUP请求消息中携带有子流控制特征标签;

获取所述子流控制特征标签;

如果能正确识别所述子流控制特征标签,则向终端返回携带有支持子流控制的信息的响应消息,以便所述终端发起子流控制;否则,向终端返回携带有不支持子流控制的信息的

响应消息。

7. 根据权利要求1或2所述的方法，其特征在于，所述控制请求消息中还携带有子流的编码类型信息，还包括：

根据所述子流的编码类型信息获取所述子流的编码类型并根据所述编码类型确定所述子流标识信息对应的子流。

8. 一种媒体数据控制装置，其特征在于，包括：

第一消息接收单元，用于接收终端发送的控制请求消息，所述控制请求消息中携带有子流的标识信息，以及所述子流所属媒体流的统一资源标识符URI；信息获取单元，用于获取所述子流的标识信息以及所述子流所属媒体流的URI；

数据确定单元，用于根据所述子流的标识信息以及所述子流所属媒体流的URI，确定所述子流对应的媒体数据；

媒体控制单元，用于基于所述媒体数据，对所述子流执行终端请求的控制操作；

其中，所述媒体流包括基本层和至少一个增强层；所述子流为所述媒体流的一部分且可以独立解码，包括所述基本层或所述基本层和至少一个所述增强层。

9. 根据权利要求8所述的装置，其特征在于，还包括：

第二消息接收单元，用于接收终端发送的描述信息请求消息；

第一响应单元，用于向所述终端返回携带有子流声明信息的响应消息，以便所述终端通过所述子流声明信息获取所述子流的标识信息。

10. 根据权利要求8所述的装置，其特征在于，还包括：

错误控制单元，用于确定所述控制请求消息中携带的子流的标识信息错误，向所述终端返回携带有子流声明信息的响应消息，以便所述终端根据所述响应消息中携带的子流声明信息重新获取子流的标识信息，并重新发送控制请求消息。

11. 根据权利要求8所述的装置，其特征在于，所述控制请求消息中还携带有子流控制特征标签；所述装置还包括：

第一子流控制特征标签获取单元，用于获取所述子流控制特征标签；

第一控制单元，用于如果能正确识别所述子流控制特征标签，则触发所述消息解析单元继续执行所述获取所述子流的标识信息以及所述子流所属媒体流的URI及后续操作，否则拒绝所述控制请求，向终端返回携带有不支持子流控制的信息的响应消息。

12. 根据权利要求8所述的装置，其特征在于，所述装置还包括：

第二子流控制特征标签获取单元，用于接收所述终端发送的RTSP SETUP请求消息，所述RTSP SETUP请求消息中携带有子流控制特征标签；获取所述子流控制特征标签；

第二控制单元，用于如果能正确识别所述子流控制特征标签，则向终端返回携带有支持子流控制的信息的响应消息，以便所述终端发起子流控制；否则，向终端返回携带有不支持子流控制的信息的响应消息。

13. 根据权利要求8所述的装置，其特征在于，所述控制请求消息中还携带有子流的编码类型信息，还包括：

编码类型获取单元，用于根据所述子流的编码类型信息获取所述子流的编码类型，以便根据所述编码类型确定所述子流标识信息对应的子流。

14. 一种媒体数据控制方法，其特征在于，包括：

获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI；

向服务器发送携带有所述子流的标识信息，以及所述子流所属媒体流的URI的控制请求消息；

接收到服务器返回的响应消息后，对所述子流进行相应的控制操作；

其中，所述媒体流包括基本层和至少一个增强层；所述子流为所述媒体流的一部分且可以独立解码，包括所述基本层或所述基本层和至少一个所述增强层。

15. 根据权利要求14所述的方法，其特征在于，当所述控制请求消息为基于单个媒体流的单独控制请求消息时，所述子流的标识信息通过所述控制请求消息的头域携带，所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带；或当所述控制请求消息为基于单个媒体流的单独控制请求消息时，所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的request-uri字段携带；或当所述控制请求消息为基于多个媒体流的合控制请求消息时，所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的头域携带。

16. 根据权利要求14或15所述的方法，其特征在于，所述获取子流的标识信息以及所述子流所属媒体流的URI包括：

向所述服务器发送描述信息请求消息；

从所述服务器返回的携带有子流声明信息的响应消息中，获取所述子流的标识信息以及所述子流所属媒体流的URI。

17. 根据权利要求14或15所述的方法，其特征在于，还包括：

如果在发送所述控制请求消息之后，接收到服务器返回的携带有子流声明信息的响应消息，则根据所述响应消息中携带的子流声明信息重新获取子流的标识信息，并重新发送控制请求消息。

18. 根据权利要求14或15所述的方法，其特征在于，所述控制请求消息中还携带有子流控制特征标签，以便当所述服务器不支持子流控制时，向终端返回携带有不支持子流控制的信息的响应消息。

19. 根据权利要求14或15所述的方法，其特征在于，向服务器发送携带有所述子流的标识信息，以及所述子流所属媒体流的URI的控制请求消息之前，还包括：

向所述服务器发送RTSP SETUP请求消息，并在所述RTSP SETUP请求消息中携带子流控制特征标签，以便当所述服务器不支持子流控制时，向终端返回携带有不支持子流控制的信息的响应消息。

20. 根据权利要求14或15所述的方法，其特征在于，所述控制请求消息中还携带有子流的编码类型信息，以便所述服务器根据所述子流的编码类型信息获取所述子流的编码类型，根据所述编码类型确定所述子流标识信息对应的子流。

21. 一种媒体数据控制装置，其特征在于，包括：

子流信息获取单元，用于获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI；

消息发送单元，用于向服务器发送携带有所述子流的标识信息，以及所述子流所属媒体流的URI的控制请求消息；

操作执行单元，用于接收到服务器返回的响应消息后，对所述子流进行相应的控制操

作；

其中，所述媒体流包括基本层和至少一个增强层；所述子流为所述媒体流的一部分且可以独立解码，包括所述基本层或所述基本层和至少一个所述增强层。

22.根据权利要求21所述的装置，其特征在于，当所述控制请求消息为基于单个媒体流的单独控制请求消息时，所述子流的标识信息通过所述控制请求消息的头域携带，所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带；或当所述控制请求消息为基于单个媒体流的单独控制请求消息时，所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的request-uri字段携带；或当所述控制请求消息为基于多个媒体流的合控制请求消息时，所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的头域携带。

23.根据权利要求21或22所述的装置，其特征在于，所述子流信息获取单元包括：

描述信息请求消息发送子单元，用于向所述服务器发送描述信息请求消息；

获取子单元，用于从所述服务器返回的携带有子流声明信息的响应消息中，获取所述子流的标识信息以及所述子流所属媒体流的URI。

24.根据权利要求21或22所述的装置，其特征在于，还包括：

重新发送单元，用于如果在发送所述控制请求消息之后，接收到服务器返回的携带有子流声明信息的响应消息，则根据所述响应消息中携带的子流声明信息重新获取子流的标识信息，并重新发送控制请求消息。

25.根据权利要求21或22所述的装置，其特征在于，所述控制请求消息中还携带有子流控制特征标签，以便当所述服务器不支持子流控制时，向终端返回携带有不支持子流控制的信息的响应消息。

26.根据权利要求21或22所述的装置，其特征在于，还包括：

RTSP SETUP请求消息发送单元，用于向所述服务器发送RTSP SETUP请求消息，并在所述RTSP SETUP请求消息中携带子流控制特征标签，以便当所述服务器不支持子流控制时，向终端返回携带有不支持子流控制的信息的响应消息。

27.根据权利要求21或22所述的装置，其特征在于，所述控制请求消息中还携带有子流的编码类型信息，以便所述服务器根据所述子流的编码类型信息获取所述子流的编码类型，根据所述编码类型确定所述子流标识信息对应的子流。

媒体数据控制方法及装置

技术领域

[0001] 本发明涉及多媒体数据处理领域,特别是涉及媒体数据控制方法及装置。

背景技术

[0002] NAT(Network Address Translation,网络地址转换)是一种将私网地址转换成公网地址的技术,需要将“私网IP地址+端口号”转换成“公网IP地址+端口号”。通过NAT技术,能够很好地解决IP地址不足的问题。

[0003] 在多媒体应用中,一个媒体内容可能包含多种数据类型(例如音频、视频及字幕等等),传输时,通常采用地址复用的方式,即不同类型的数据公用一个IP地址,但是使用不同的UDP端口,以此来区分不同的数据类型。这样,当在私网与公网之间进行通信时,需要进行NAT转换的“私网IP地址+端口号”的数目,就与媒体内容中包含的数据类型的数目相同。

[0004] 随着SVC(Scalable Video Coding,可扩展视频编码)及MVC(Multi-view Video Coding,多视视频编码)等技术的出现,一种数据类型下又可能包含多种特性的数据(例如,在基于SVC的应用中,同样是视频数据,可能从不同帧率、不同分辨率、不同质量等方面又分为不同特性的视频数据,等等)。这就需要对各种数据类型的各种特性的数据进行区分。一种简单的区分方式仍然可以通过不同的端口号进行区分,但是,只要各种不同特性的数据对应的端口号不同,就要分别进行NAT转换,因此会造成需要进行NAT转换的“私网IP地址+端口号”数目会非常多,也即NAT的开销会比较大。

[0005] 因此,在另一种方式下,是一种类型下的不同特性的数据对应相同的IP地址及UDP端口号,从语法表示上对不同特性的数据加以区分。这就意味着,同一种类型下不同特性的数据通过同一个媒体流进行传输。

[0006] 然而,现有技术中,只能实现以媒体流为单位的控制。因此,在将同一种类型下不同特性的数据通过同一个媒体流进行传输的情况下,如果在实际应用中需要对不同特性数据进行独立控制,则现有技术无法实现。

发明内容

[0007] 本发明提供媒体数据控制方法及装置,能够在将同一种类型下不同特性的数据通过同一个媒体流进行传输的情况下,实现对不同特性数据的独立控制。

[0008] 本发明一方面提供了一种媒体数据控制方法,包括:接收终端发送的控制请求消息,所述控制请求消息中携带有子流的标识信息,以及所述子流所属媒体流的统一资源标识符URI;获取所述子流的标识信息以及所述子流所属媒体流的URI;根据所述子流的标识信息以及所述子流所属媒体流的URI,确定所述子流对应的媒体数据;基于所述媒体数据,对所述子流执行终端请求的控制操作。

[0009] 本发明另一方面提供了一种媒体数据控制装置,包括:第一消息接收单元,用于接收终端发送的控制请求消息,所述控制请求消息中携带有子流的标识信息,以及所述子流所属媒体流的统一资源标识符URI;信息获取单元,用于获取所述子流的标识信息以及所述

子流所属媒体流的URI；数据确定单元，用于根据所述子流的标识信息以及所述子流所属媒体流的URI，确定所述子流对应的媒体数据；媒体控制单元，用于基于所述媒体数据，对所述子流执行终端请求的控制操作。

[0010] 本发明再一方面提供了一种媒体数据控制方法，包括：获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI；向服务器发送携带有所述子流的标识信息，以及所述子流所属媒体流的URI的控制请求消息；接收到服务器返回的响应消息后，对所述子流进行相应的控制操作。

[0011] 本发明又一方面提供了一种媒体数据控制装置，包括：子流信息获取单元，用于获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI；消息发送单元，用于向服务器发送携带有所述子流的标识信息，以及所述子流所属媒体流的URI的控制请求消息；操作执行单元，用于接收到服务器返回的响应消息后，对所述子流进行相应的控制操作。

[0012] 根据本发明提供的具体实施例，本发明公开了以下技术效果：

[0013] 本发明实施例中，通过在终端发送的控制请求消息中携带子流的标识信息以及子流所属媒体流的URI，使得服务器在收到终端的控制请求消息时，能够获取到这两个信息，进而就可以根据这两个信息获取到终端请求的子流对应的媒体数据，然后就可以基于该媒体数据，对终端请求的子流进行相应的控制操作，例如，包括播放、暂停等等。因此，通过本发明实施例，可以使得即使在将同一种类型下不同特性的数据通过同一个媒体流进行传输的情况下，也能够实现对不同特性数据的独立控制。

附图说明

[0014] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0015] 图1是本发明实施例提供的方法的流程图；

[0016] 图2是本发明实施例提供的方法的第一示意图；

[0017] 图3是本发明实施例提供的方法的第二示意图；

[0018] 图4是本发明实施例提供的方法的第三示意图；

[0019] 图5是本发明实施例提供的装置的示意图；

[0020] 图6是本发明实施例提供的另一方法的流程图；

[0021] 图7是本发明实施例提供的另一装置的示意图。

具体实施方式

[0022] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员所获得的所有其他实施例，都属于本发明保护的范围。

[0023] 参见图1，本发明实施例提供的媒体控制方法包括以下步骤：

[0024] S101：接收终端发送的控制请求消息，所述控制请求消息中携带有子流的标识信

息,以及所述子流所属媒体流的URI(Uniform Resource Identifier,统一资源标识符);

[0025] 需要说明的是,本发明实施例可以与RTSP协议相结合使用,其中的控制请求消息可以是RTSP PLAY请求消息或RTSP PAUSE请求消息等等。

[0026] S102:获取所述子流的标识信息以及所述子流所属媒体流的URI;

[0027] S103:根据所述子流的标识信息以及所述子流所属媒体流的URI,确定所述子流对应的媒体数据;

[0028] 子流是媒体流的一部分,并且子流可以独立解码。不同子流具备不同的媒体特征(媒体特征指:帧率、分辨率、码率、或者视角等),因此可以提供不同的观看效果。

[0029] 例如,媒体内容中的视频可以采用SVC进行编码。从时间、空间、质量等角度上进行划分,编码后的码流包含多层(分为基本层和增强层,基本层只有一个,增强层可以是多个)。其中,基本层可以独立解码,解码后获得的视频可能帧率较低、分辨率较低、或者质量较低,可以用于在带宽受限或者带宽不稳定的应用环境中,满足基本的观看需求。增强层不能独立解码,必须与和基本层及它之前的增强层(之前的增强层通常是指编码时依赖的层,相应的,解码时也需要依赖该层才能进行解码)联合在一起解码,用于提高观看效果,具体可以是从时间、空间、或者质量上对基本层提供的观看效果进行提高。

[0030] SVC编码后输出的码流可以通过一个媒体流传输,即通过一个媒体流传输多层码流。其中,子流由一个或者多个层组成并且可以独立解码,即单独的基本层,或者基本层与一个或多个增强层的组合,都可以构成该媒体流中的子流。当然,需要说明的是,基本层并不是与任意的增强层组合都可以构成一个子流,只有能够独立解码的组合才可以构成子流。例如,SVC码流包括基本层、增强层A和增强层B,对于增强层A和增强层B,基本层+增强层A+增强层B能够独立解码,则该组合可以构成一个子流;而基本层+增强层B不能够独立解码,则该组合不可以构成一个子流。

[0031] 又如,如果媒体内容中的视频是多视视频(多视视频是指由所处几何位置不同的多个摄像机输出的多路视频,每一路视频简称为一个视,这里的摄像机可以是虚拟的摄像机),可以采用MVC对多视视频进行编码得到码流,编码过程中可以选择其中一路视频作为基本视,对基本视编码时不需要进行视间预测,因此可以独立解码。其它视用于提供视角上的可扩展性,编码时需要进行视间预测,因此不能独立解码,只能和基本视及它编码时所依赖的视联合在一起解码。

[0032] MVC编码后输出的码流可以通过一个媒体流传输,即通过一个媒体流传输多视。其中,子流由一个或者多个视组成并且可以独立解码,即单独的基本视,或者基本视与其他视中的一个或多个的组合,都可以构成该媒体流中的子流。当然,需要说明的是,基本视并不是与任意的其他视组合都可以构成子流,只有能够独立解码的组合才可以构成子流。例如,MVC码流包括基本视、其他视A和其他视B。对于其他视A和其他视B,基本视+其他视A+其他视B能够独立解码,则该组合可以构成一个子流;而基本视+其他视B不能够独立解码,则该组合不可以构成一个子流。

[0033] 另外,MVC还支持时间可伸缩性,即还可以从时间角度上进行划分,将每个视的码流分为多层。此时子流的构成还需要考虑时间可伸缩性引入的解码依赖关系。即某个其它视的某个时间级别的层(简称时间层),要和基本视及它编码时所依赖的视中时间级别相同或者更低的层联合在一起解码。

[0034] 无论是SVC还是MVC,不同的子流提供不同的观看效果,同时,对带宽等通信环境的要求、对终端能力的要求也会不同。因此,如果能够实现基于子流的单独控制,则用户或者终端可以根据当前的带宽条件、终端能力等因素选择性地接收子流,从而有利于在播放效果与播放流畅度之间达到均衡。

[0035] 在本发明实施例中,为了实现基于子流的控制,需要终端在该控制请求中携带子流的标识信息,以及所述子流所属媒体流的URI。其中,子流的标识信息是指,在子流对应的媒体流中能够唯一标识一个子流的信息,下面对此进行介绍。

[0036] 例如,在SVC中,可以采用dependency_id(D)、temporal_id(T)、quality_id(Q)来标识不同的层。其中,dependency_id为依赖标识,temporal_id为时间标识,quality_id为质量标识。因此,终端就可以在控制请求消息中携带具体的(D,T,Q)的值作为子流的标识信息,相应的,服务器在根据媒体流URI找到媒体流对应的数据之后,就可以进一步根据具体的(D,T,Q)的值,找到子流对应的数据。

[0037] 在MVC中,子流可以采用temporal_id、view_id标识不同视的不同时间层。其中, temporal_id为时间标识,view_id为视标识。因此,终端就可以在控制请求消息中携带具体的temporal_id和view_id的值作为子流的标识信息,相应的,服务器在根据媒体流URI找到媒体流对应的数据之后,就可以进一步根据解析出的temporal_id和view_id的值,找到子流对应的数据。

[0038] 总之,无论在SVC还是MVC下,终端都可以通过在控制消息中携带子流的标识信息及子流所属媒体流的URI,以便使得服务器获知终端想要控制的是哪个媒体流的哪个子流。

[0039] 其中,关于子流的标识信息及子流所属的媒体流的URI,终端可以首先获取媒体内容的描述信息,从描述信息中获取。其中,终端获取媒体内容的描述信息的途径可以有多种。例如,可以预先通过HTTP协议向存储有媒体内容的描述信息的服务器发起获取媒体描述信息的HTTP请求,从服务器获取该描述信息、或者预先接收携带有媒体内容的描述信息的电子邮件,从电子邮件中获取该描述信息,当然,还可以向服务器发起获取媒体描述信息的RTSP请求,从服务器获取该描述信息。例如,在终端发起控制请求之前,可以首先向服务器发送描述信息请求消息,所述描述信息请求消息为RTSP Describe请求消息,服务器在收到该消息之后,就可以向终端返回媒体内容的描述信息,该描述信息中就可以包括媒体流的个数、各媒体流的URI、各媒体流递送采用的协议、传输协议参数、媒体编码信息等等,此外还可以包括媒体流中所有子流的声明信息。因此,通过解析该服务器返回的该响应消息,终端就可以获取到媒体流的URI,以及媒体流的子流的声明信息,进而从子流的声明信息中就可以获取到各子流的标识信息。

[0040] 具体实现时,可以将子流的声明信息作为媒体描述信息的一部分,并将媒体描述信息制作成SDP文件。在SDP文件中,为了让终端知道SVC码流中包含哪些层,服务器可以通过SDP文件中的参数向终端声明SVC码流中各个层的(D,T,Q)值(或者layer_id值,layer_id为层标识),并指明哪些层可以组合为子流,这样终端就可以直接将需要的子流对应的各个层的(D,T,Q)值或者layer_id值作为子流标识信息发送给服务器,相应的,服务器解析出的子流对应的各个层的(D,T,Q)值或者layer_id值。在解析出(D,T,Q)值或者layer_id值之后,服务器分别按照各层的(D,T,Q)值或者layer_id值去获取各层对应的数据即可。

[0041] 此外,在实际应用中,还可以采用另一种方式实现,具体可以是:在SDP文件中的参

数中,通过属性行a=fmtp中的参数sprop-operat ion-point-info携带一组操作点描述矢量,一个操作点描述矢量用来声明一个操作点。操作点描述矢量的格式可以为:<layer-ID, temporal-ID,dependency-ID,quality-ID,profile-level-ID,avg-framerate,width,height,avg-bitrate,max-bitrate>。其中layer-ID为操作点的层标识,temporal-ID为操作点的时间标识,dependency-ID为操作点的依赖标识,quality-ID为操作点的质量标识,profile-level-ID为操作点的层级标识,avg-framerate为操作点的平均帧率,width为操作点对应的视频帧的宽度,height为操作点对应的视频帧的高度,avg-bitrate为操作点的平均码率,max-bitrate为操作点的最大码率。

[0042] 其中,操作点的layer-ID、temporal-ID、dependency-ID、quality-ID的值分别等于操作点对应的解码依赖性最高的层的layer_id、temporal_id、dependency_id、quality_id的值。可见,操作点既可以用dependency_id(D)、temporal_id(T)、quality_id(Q)的组合进行标识,也可以用layer_id进行标识,并且,一个layer_id对应着一个D、T、Q的组合。操作点对应的是相应的层和它之前的所有层,也即所有(D,T,Q)的值分别小于等于该操作点的(D,T,Q)的值的NAL(Network Abstraction Layer,网络提取层)包组成的码流。因此,一个操作点对应着一个能够独立解码并且具备特定媒体特征的码流,也就是说,一个操作点对应着一个子流。这样,如果终端需要对某子流进行控制,则直接将对应的操作点的(D,T,Q)值或者layer-ID值作为子流的标识信息发送给服务器即可。相应的,服务器解析出的是操作点的(D,T,Q)值或者layer-ID值。其中,如果解析出的是操作点的(D,T,Q)值,则服务器就可以直接将(D,T,Q)的取值分别小于等于服务器解析出的(D,T,Q)的取值的NAL包构成子流对应的媒体数据;例如,服务器从终端的控制请求消息中解析出的(D,T,Q)值是(1,1,0),则服务器可以将(D,T,Q)值为(1,1,0)、(1,0,0)、(0,1,0)、(0,0,0)的NAL包取出,构成子流的媒体数据。或者,如果服务器解析出的是操作点的layer-ID值,则可以首先将layer-ID的取值转换为(D,T,Q)值,然后再将(D,T,Q)的取值小于等于服务器解析出的(D,T,Q)的取值的NAL包构成子流对应的媒体数据。服务器通过查询layer-ID与(D,T,Q)之间的对应关系,将layer-ID的取值转换为(D,T,Q)值。该layer-ID与(D,T,Q)的对应关系可以是预先存储在服务器上的,例如,可以存放在上述SDP文件中,也可以存放在SVC码流的scalability information SEI message中。

[0043] 对于MVC而言,为了向终端声明MVC码流中包含哪些子流,服务器同样可以通过SDP文件中的参数向终端声明MVC码流中包含的视的view_id值以及每个视的时间层的temporal_id值,并指明哪些视的哪些时间层可以组合为子流。这样终端就可以直接将需要的子流对应的各个视的各个时间层的temporal_id和view_id值作为子流标识信息发送给服务器,相应的,服务器也可以在解析出temporal_id和view_id之后,分别去获取各视的各个时间层对应的数据即可。

[0044] 在实际应用中,也可以通过SDP文件中的参数向终端声明MVC码流中包含哪些操作点。具体方式可以是:通过属性行a=fmtp中的参数sprop-mvc-operation-point-info携带一组操作点描述矢量,一个操作点描述矢量用来声明一个操作点。操作点描述矢量的格式为:<operation-point-id,temporal-id,num-target-output-views,1*target-output-view-id,profile-level-id,avg-framerate,avg-bitrate,max-bitrate>。其中operation-point-id为操作点的标识,temporal-ID为操作点的时间标识,num-target-

output-views为操作点的目标输出视的个数,view-id为操作点的目标输出视的标识,profile-level-ID为操作点的层级标识,avg-framerate为操作点的平均帧率,width为操作点对应的视频帧的宽度,height为操作点对应的视频帧的高度,avg-bitrate为操作点的平均码率,max-bitrate为操作点的最大码率。

[0045] 其中,操作点的temporal-id、target-output-view-id分别等于该操作点对应的所有视的所有时间层中时间级别最高的层的temporal_id、该操作点对应的目标输出视的view_id。可见,操作点既可以用temporal_id和一组view_id(target output view的view_id)进行标识,也可以用operation_point_id进行标识。但操作点对应的是所有temporal_id的取值(对应着帧率)小于等于该temporal_id值,并且view_id的取值等于该组view_id值中的任一个,或者该组view_id值对应的任一view解码所依赖(直接依赖或者间接依赖)的所有view的view_id值中的任一个的NAL包组成的码流,因此,一个操作点对应着一个能够独立解码并且具备特定媒体特征的码流,也就是说,一个操作点对应的一个子流,因此,终端也可以将子流对应的操作点的temporal_id和一组view_id(target output view的view_id)或者operation-point-id作为子流标识信息携带在控制请求消息中。当然,由于服务器最终还是要根据具体的temporal-id及view_id值来查找子流对应的具体数据,因此,当终端将operation_point_id作为子流标识信息时,服务器进一步根据operation_point_id与temporal_id和target_output_view_id组合的对应关系确定操作点对应的temporal_id与一组target output view的view_id。这种对应关系可以有多种获取方法。如,可以从对应的SDP文件中的子流声明信息中获取,也可以通过服务器上存放的MVC码流中的view scalability info SEI message中获取。为了确定子流对应的媒体数据,服务器还需要进一步根据view之间的解码依赖关系,确定该组target output views解码依赖(直接依赖或者间接依赖)的view的view_id。解码依赖关系可以根据MVC文件中的元数据(MVC码流以文件的形式存放在服务器上,文件中除了MVC码流外,还有用于描述MVC文件的元数据)获取,如根据ViewIdentifierBox获取。相应媒体资源中所有包头中的temporal_id的取值小于等于获得的temporal_id的值,且view_id的取值等于服务器获得的一组view_id的值(target output view的view_id的值、target output view解码依赖的view的view_id的值)中的某个取值的NAL包构成子流对应的媒体数据。

[0046] 当然,如果不是以SDP文件格式发送描述信息,则可以为服务器生成一个携带媒体流中的子流声明信息的头域,并在对描述信息请求消息的响应消息中携带该头域,这样,终端可以通过解析该响应消息的头域,获知子流声明信息。

[0047] 具体的,所述描述信息请求消息为RTSP Describe请求消息,所述对描述信息请求消息的响应消息为RTSP Describe的成功响应消息。服务器通过RTSPDescribe的成功响应消息中的头域携带子流声明信息时,对于一个包含子流的媒体流,其子流声明信息由可以包括子流所属的媒体流的URI和一组操作点描述矢量(可以与前文所述的操作点描述矢量一致),每个操作点描述矢量都声明一个操作点(每个操作点对应一个子流)。若有多个媒体流中包含子流时,该头域中携带的是一组子流声明信息,分别对应不同的媒体流。不同媒体流的子流声明信息之间采用特殊字符进行分隔,以便于服务器区分,这可以通过语法上的定义来实现。在互联网标准中通常采用ABNF来描述语法定义,具体通过ABNF描述携带子流声明信息的头域的语法定义如下所述。其中,通过“分号”分隔不同媒体流的子流声明信息,

这样,当终端解析该头域时,可以根据“分号”来区分出一个个的媒体流对应的子流声明信息。

[0048]

```
substream-info = "substream-info" HCOLON [substream-info-spec
    * (SEMI substream-spec)]  
  
substream-info-spec = stream-url substream-type 1*descriptor-vector  
stream-url = <as defined in draft-ietf-mmmusic-rfc2326bis-27>  
substream-type = "type" EQUAL substream-type-value  
substream-type-value = "SVC" / "MVC" / substream-type-value-ext  
substream-type-value-ext = token  
descriptor-vector = RAQUOT layer_id_value COMMA temporal_id_value  
                    COMMA dependency_id_value COMMA quality_id_value  
                    COMMA profile_level_id_value  
                    COMMA avg_framerate_value  
                    COMMA width_value COMMA height_value COMMA
```

[0049]

```
avg_bitrate_value COMMA max_bitrate_value LAQUAT
/ RAQUOT operation_point_id_value COMMA
temporal_id_value
COMMA num_target_output_views_value
1* (COMMA target_output_view_id_value)

COMMA profile_level_id_value
COMMA avg_framerate_value
COMMA avg_bitrate_value COMMA max_bitrate_value
LAQUAT
/ descriptor-vector-ext

layer_id_value = 1*4DIGIT; 0~2047
dependency_id_value = DIGIT ; 0~7
temporal_id_value = DIGIT ; 0~7
quality_id_value = 1*2DIGIT; 0~15
profile_level_id_value = *HEX
avg_frame_rate_vale = *DIGIT
width_value = *DIGIT
height_value = *DIGIT
avg_bitrate_value = *DIGIT
max_bitrate_value = *DIGIT
operation_point_id_value = 1*5DIGIT; 0~65535
num_target_output_views_value = 1*4DIGIT; 0~1023
target_output_view_id_value = 1*4DIGIT; 0~1024
descriptor-vector-ext = token

HCOLON = *( SP / HT ) ":" SWS
SWS      = [LWS] ; Separating White Space
LWS      = [CRLF] 1*( SP / HT ) ; Line-breaking White Space
```

[0050]

```

DIGIT      =  %x30-39 ; any US-ASCII digit "0".."9"
HT         =  %x09   ; US-ASCII HT, horizontal-tab (9)
SP         =  %x20   ; US-ASCII SP, space (32)
COMMA     =  SWS "," SWS ; comma
EQUAL     =  SWS "=" SWS ; equal
DQ         =  %x22   ; US-ASCII double-quote mark (34)
RAQUOT    =  ">" SWS ; right angle quote
LAQUOT    =  SWS "<" ; left angle quote
SEMI       =  SWS ";" SWS ; semicolon
HEX        =  DIGIT / "A" / "B" / "C" / "D" / "E" / "F" / "a" / "b" / "c" / "d" / "e" /
              "f"
token      =  1*(%x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39
              / %x41-5A / %x5E-7A / %x7C / %x7E)
              ; 1*<any CHAR except CTLs or tspecials>

```

[0051] 终端接收到RTSP Describe的成功响应消息后,解析携带子流声明信息的头域获取包含子流的媒体流的URI和对应的子流标识信息。在获取到媒体流的URI以及子流的标识信息之后,就可以将其携带于控制请求消息中,发送给服务器。具体实现时,针对不同的应用场景,媒体流的URI以及子流的标识信息在控制请求消息中的位置可以不同。下面对此进行详细地介绍。

[0052] 在已有的基于媒体流的控制方式中,在一种方式下,终端可以基于单个媒体流进行单独控制,例如,对于某视频节目,可以单独对视频流及音频流进行控制。在这种单独控制的情况下,可能存在以下缺点:交互次数多,并且,在单独进行控制时,会使得控制请求先后到达服务器,服务器返回的响应可能也会有先后之分,为了保证音视频播放的同步,终端只能等收到所有媒体流的响应之后才能执行相应的操作,因此,会产生比较长的等待延迟。为此,在另一些应用场景中,还也可以将多个媒体流一起进行合控制(也即通过一个控制请求消息控制多条媒体流)。终端发送控制请求消息时,都会包含request-uri字段,其中,关于单独控制的情况,该字段携带的是媒体流的URI,而对于合控制的情况,该字段携带的通常是合控制的URI。需要说明的是,关于媒体流的URI,如前文所述,终端可以从服务器返回的描述信息请求消息的响应消息中获得。而对于合控制的URI,如果服务器支持合控制,则可以在向终端返回描述信息请求消息的响应消息时,在SDP文件中携带合控制的URI,并且该合控制的URI在SDP文件中的位置与媒体流的URI通常是不同的,例如,通常是位于所有媒体流的URI的最上面,因此,终端同样可以从SDP文件中获取到合控制的URI。

[0053] 因此,如果是进行基于媒体流的单独控制,则仍然可以在控制请求消息的request-uri字段携带子流所属媒体流的URI。而对于子流的标识信息,可以生成一个携带子流标识信息的头域(如,substream头域)并携带在控制请求消息中。在互联网标准中通常

采用ABNF来描述语法定义,因此下面通过ABNF描述携带子流标识信息的头域的语法定义:

substream = “substream”HCOLON [substream-spec
* (COMMA substream-spec)]
substream-spec = substream-type COMMA substream-id
substream-type = “type” EQUAL substream-type-value
substream-type-value = “SVC” / “MVC” / substream-type-value-ext
substream-type-value-ext = token
substream_id = layer-id
/ dependency-id temporal-id quality-id
[0054] / mvc-operation-point-id
/ temporal-id 1*target-output-view-id
/ substream-id-ext
layer-id = “layer_id” EQUAL layer_id_value
layer_id_value = 1*4DIGIT; 0~2047
dependency-id = “dependency_id” EQUAL dependency_id_value
dependency_id_value = DIGIT ; 0~7
temporal-id = “temporal_id” EQUAL temporal_id_value

[0055]

```

temporal_id_value = DIGIT ; 0~7
quality-id = "quality_id" EQUAL quality_id_value
quality_id_value = 1*2DIGIT; 0~15
mvc-operation-point-id = "operation_point_id" EQUAL operation_point_id_value
operation_point_id_value = 1*5DIGIT; 0~65535
target-output-view-id = "view_id" EQUAL view_id_value
view_id_value = 1*4DIGIT; 0~1024
substream-id-ext = token

HCOLON = *( SP / HT ) ":" SWS
SWS      = [LWS] ; Separating White Space
LWS      = [CRLF] 1*( SP / HT ) ; Line-breaking White Space
DIGIT    = %x30-39 ; any US-ASCII digit "0".."9"
HT       = %x09 ; US-ASCII HT, horizontal-tab (9)
SP       = %x20 ; US-ASCII SP, space (32)
COMMA   = SWS "," SWS ; comma
EQUAL   = SWS "=" SWS ; equal
DQ      = %x22 ; US-ASCII double-quote mark (34)
token   = 1*(%x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39
         / %x41-5A / %x5E-7A / %x7C / %x7E)
         ; 1*<any CHAR except CTLs or tspecials>

```

[0056] 例如,一个实例为substream:type=svc;layer_id=1,如果控制请求消息中携带了该头域,则表示请求对携带SVC码流的媒体流中layer_id为1的子流进行播控。相应的,在进行基于媒体流的单独控制的情况下,服务器在收到终端的控制请求消息后,就可以通过解析该控制请求消息的request-uri字段获取子流所属媒体流的URI,通过解析该控制请求消息的头域来获取子流的标识信息。

[0057] 当然,在进行基于媒体流的单独控制的情况下,也可以将子流的标识信息也携带在控制请求消息的request-uri字段中,这样,服务器通过解析控制请求消息的request-uri字段,就可以获取到子流所属媒体流的URI以及子流的标识信息。

[0058] 对合控制而言,服务器仍然需要知道子流所属的媒体流的URI,以及子流的标识信息,才能确定子流对应的数据,而在基于多个媒体流的合控制的情况下,控制请求消息的请求行中的“request-uri”域是合控制的RTSP URI,因此,在合控制的情况下,可以生成一个同时携带子流所属媒体流的URI以及子流的标识信息的头域,并携带在控制请求消息中。这

样,服务器可以通过解析控制请求消息的头域,获取子流所属媒体流的URI以及子流的标识信息,进而就可以确定子流对应的数据。

[0059] 需要说明的是,在合控制的情况下,一个控制请求中会涉及多个媒体流,具体实现时,可以对涉及到的各个媒体流都进行子流控制,也可以仅对其中的一个或几个媒体流进行子流控制。其中,当在一个控制请求中需要对多个媒体流都进行子流控制时,在控制请求消息的头域中携带子流标识信息以及子流对应的媒体流的URI时,这两个信息可以成组出现(一个媒体流对应一组),每组之间可以采用特殊字符进行分隔,以便于服务器区分,这可以通过语法上的定义来实现。例如,头域的ABNF语法可以定位为:

[0060] `substream = "substream" HCOLON [stream-uri] [substream-spec* (COMMA substream-spec)]* (SEMI [stream-uri] [substream-spec* (COMMA substream-spec)])`

[0061] `SEMI = SWS";" SWS; semicolon`

[0062] 其中,SEMI中的[stream-uri]对应媒体流的URI,[substream-spec*(COMMA substream-spec)]对应子流的标识信息,当需要对两个媒体流(stream1和stream2)都进行子流控制时,可以如下表示:[stream1-uri][substream-spec*(COMMA substream-spec)];[stream2-uri][substream-spec*(COMMA substream-spec)],可见,不同媒体流之间可以用“分号”分隔,这样,当服务器解析该头域时,可以根据“分号”来区分出一个个的媒体流。

[0063] 另外需要说明的是,在实际应用中,系统中可能是多种编码类型的媒体流共存,例如,有些媒体流可能是SVC编码,有些媒体流可能是MVC编码,等等。而媒体流采用不同的编码类型时,子流的标识信息通常会采用不同的表示方式,为了简化服务器的识别过程,还可以在控制请求消息中携带子流的编码类型信息。具体实现时,可以采用在用于携带子流标识信息的头域中增加“substream_id”字段,在该字段携带具体的类型值。

[0064] S104:基于所述媒体数据,对所述子流执行终端请求的控制操作。

[0065] 其中,对于视频节目等媒体数据而言,控制请求消息通常也称为控制请求消息,这种控制具体可以包括播放控制、暂停控制、快进、快退控制等等。

[0066] 总之,通过本发明实施例,可以使得即使在将同一种类型下不同特性的数据通过同一个媒体流进行传输的情况下,也能够实现对不同特性数据的独立控制。

[0067] 在实际应用中,终端发送的控制请求消息中携带的子流标识信息,可能存在不正确的情况。发生这种现象的原因可能有多种,例如,其中一种可能是:如前文所述,终端在获取媒体内容的描述信息时,存在多种途径,而当终端是预先通过某种方式(包括HTTP协议的传输或电子邮件等方式)获取的描述信息,并从中获取子流的标识信息时,由于信息获取的时间与实际发送控制请求的时间可能会相隔比较长,一旦服务器在此期间对子流标识信息等进行了更新,则终端依据之前获取到的信息发送请求时,携带的信息可能就是不正确的。

[0068] 为了应对这种情况,在本发明实施例中,服务器在从控制请求消息中解析出子流的标识信息之后,如果发现子流的标识信息不正确(例如,发现解析出的子流标识信息在服务器的数据库中并不存在,则证明终端发送过来的子流标识信息不正确),则可以向终端返回错误响应,并在错误响应中携带各个子流的正确的声明信息(同样可以采用在响应消息的消息体或头域中携带的方式,具体的方式可以与在控制请求消息中携带子流标识等信息时类似,这里不再赘述)。这样,终端就可以从响应消息中解析出正确的子流声明信息,并从

中获取需要进行子流控制的子流的标识信息,然后重新发送控制请求消息,并在该消息中携带正确的子流标识信息以及子流所属媒体流的URI即可。

[0069] 此外,在实际应用中,有些服务器可能不支持基于子流的媒体控制,如果将携带有子流标识信息的控制请求消息发送这种服务器,这种服务器会无法识别出携带有子流标识信息的头域,以至于最终可能又变成基于整个媒体流的控制(因为控制请求消息中同时还携带有媒体流的URI等信息)。

[0070] 为避免上述情况的发生,本发明实施例还提供了相应的解决措施。例如,在一种实现方式下,终端可以在发送控制请求消息时,在控制请求消息中携带子流控制特征标签,服务器能够从控制请求消息中得到该子流控制特征标签。当服务器获得该子流控制特征标签之后,如果服务器支持子流控制,就能够正确识别控制请求消息中新生成的头域(携带有子流标识信息的头域),并按照前文所述的流程进行基于子流的媒体控制。当然,如果服务器本身并不支持子流控制,则在获取到子流控制特征标签后,无法正确识别,因此,可以拒绝终端的控制请求,并向终端返回响应消息,在该响应消息中携带不支持子流控制的信息,而不会再进行基于整个媒体流的控制。

[0071] 具体实现时,可以在控制请求消息的require头域中携带子流控制特征标签。其中,require头域是已有协议中定义好的头域。所有服务器,无论是支持还是不支持子流控制,都能够解析该头域。通过解析该头域,就可以获取到子流控制特征标签。如果该服务器支持子流控制,则能够正确识别其中携带的子流控制特征标签,进而,可以对携带有子流标识等信息的头域进行解析,并从中获取子流的标识等信息,并按照前文所述的流程进行基于子流的媒体控制。如果该服务器不支持子流控制,则不能正确识别解析出的子流控制特征标签,进而,可以拒绝终端的控制请求,并向终端返回响应消息,在该响应消息中包含unsupported头域,其中携带服务器不能识别的子流控制特征标签。终端收到响应消息后,解析其中的unsupported头域,获取子流控制特征标签,以此获知服务器不支持子流控制。当然子流控制特征标签也可以携带在其他已定义的头域中

[0072] 在另一种实现方式下,由于终端在发送控制请求消息之前,通常还需要向服务器发送RTSP SETUP请求消息,要求为需要控制的子流确定传输机制,建立RTSP会话或者将媒体流加入到已有RTSP会话中,终端收到RTSP SETUP的成功响应消息之后,才能进行后续的控制操作。因此,在本发明实施例中,也可以在RTSP SETUP请求消息中携带子流控制特征标签,这样,服务器在接收到终端的RTSP SETUP请求之后,就可以通过解析该RTSP SETUP请求消息,获得子流控制特征标签。之后,如果服务器支持子流控制,则可以将其支持子流控制的信息携带在响应消息中返回给终端。这样,可以通知终端该服务器是支持子流控制的,终端后续可以向该服务器发送子流控制请求。当然,如果服务器不支持子流控制,则在获得子流控制特征标签之后,可以向终端返回响应,并且可以携带不支持子流控制的信息,这样,终端在向该服务器发起控制请求时,就不会发起子流控制请求了。

[0073] 在这种通过RTSP SETUP请求消息携带子流控制特征标签的方式,在具体实现时,可以将子流控制特征标签携带在RTSP SETUP请求的support头域中。support头域也是一个已有协议中定义好的头域,服务器都能解析该头域,将子流控制特征标签通过该头域携带,可以保证服务器能够获取到该子流控制特征标签。这样,服务器解析该头域,就可以获取到子流控制特征标签,如果该服务器支持子流控制,则能够正确识别其中携带的子流控制特

征标签,进而,在RTSP SETUP请求的响应消息中也包含一个携带子流控制特征标签的support头域,通知终端该服务器是支持子流控制的。如果该服务器不支持子流控制,服务器不能正确识别解析出的子流控制特征标签,进而,则可以在RTSP SETUP请求的响应消息中包含一个unsupported头域,其中携带服务器不能识别的子流控制特征标签。终端收到响应消息后,解析其中的unsupported头域,获取子流控制特征标签,以此获知服务器不支持子流控制。当然也可以携带在其他已定义的头域中。

[0074] 以上所述对本发明实施例提供的媒体数据控制方法进行了介绍。需要说明的是,本发明实施例并不仅限于SVC、MVC中的子流控制,其他涉及到子流控制的应用中也可以应用。例如,媒体数据中包含多个子流,不同子流的负载格式不同,此时,可以将负载格式类型作为子流标识信息;又如,媒体数据中包含多个子流,不同子流的源SSRC(Synchronization Source,同步源)不同,此时,可以将源SSRC作为子流的标识信息,等等。为了更好地理解本发明实施例,下面通过具体的例子,对本发明实施例提供的媒体数据控制方法进行更为详细地介绍。

[0075] 首先需要说明的是,在该例子中,假设媒体内容为某视频节目,控制请求消息另外,假设是基于单个媒体流进行控制,则参见图2,该方法可以包括如下步骤:

[0076] S201:终端向服务器发送描述信息(Describe)请求消息,请求从服务器获取媒体内容的描述信息。Describe请求消息中的request-uri字段为媒体内容的URI。

[0077] S202:服务器向终端发送200OK响应消息,响应消息中包含媒体内容的描述信息。其中,媒体内容的描述信息可以包括:媒体流的个数、各媒体流对应的RTSP URI、各媒体流递送采用的协议、传输协议参数、媒体编码信息等。

[0078] 如果描述信息是以SDP文件格式发送,则还可以直接在文件包括媒体流中子流的声明信息。否则,还可以为该响应消息生成新的头域,在头域中携带子流的声明信息。此外,终端也可以通过其他方式获取媒体内容的描述信息。

[0079] S203:终端根据媒体内容的描述信息向服务器发送的RTSP SETUP请求消息,请求确定媒体流的传输机制,建立RTSP会话。RTSP SETUP请求消息的request-uri为要建立和控制的媒体流的RTSP URI,RTSP SETUP请求消息中还包含该媒体流的传输参数。

[0080] S204:服务器确定RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。建立RTSP会话,生成RTSP会话标识,向终端发送200OK响应消息并在200OK响应消息中包含上述会话标识。

[0081] S205:终端向服务器发送控制请求消息。控制请求消息中包括携带有子流标识信息的头域。该子流标识信息用于确定子流对应的媒体数据。不同场景中,子流标识信息可能不同。例如,如果媒体流携带的是SVC码流,子流标识信息可以是layer-id,或者是dependency-id,temporal-id,quality-id的组合;如果媒体流携带的是MVC码流,子流标识信息可以是operation-point-id,或者temporal-id,view-id的组合。控制请求消息的request-uri为要操控的子流对应的媒体流的RTSP URI。此外,控制请求消息中还携带有S204中获得的会话标识,用于标识请求用于哪个RTSP会话。该控制请求消息可以是播放(RTSP PLAY)请求消息或者暂停(RTSP PAUSE)请求消息,等等。

[0082] S206:服务器处理控制请求消息,解析控制请求消息中的request-uri,获取媒体流的RTSP URI,解析携带有子流标识信息的头域,获取子流标识信息。

[0083] S207:如果服务器接受子流操控请求,向终端发送2000K响应消息。根据S206获得的媒体流的RTSP URI确定媒体资源,根据子流标识信息从上述媒体资源中确定子流对应的媒体数据,对子流对应的媒体数据执行请求的操控操作。例如,如果操控请求为RTSP PLAY请求消息,则向终端发送子流对应的媒体数据;如果操控请求为RTSP PAUSE请求消息,则停止向终端发送子流对应的媒体数据。

[0084] S208:如果服务器不接受子流操控请求,向终端发送错误响应消息。

[0085] 下面针对合控制的情况,通过以下应用中的例子进行详细的介绍。总的来说,与基于单个媒体流的控制不同的是,在合控制的情况下,在控制请求消息中的substream头域中,除携带子流标识信息外,还携带子流对应的媒体流的RTSP URI。具体而言,根据建立RTSP会话的方式不同,合控制中子流控制的流程也会有所不同,在其中一种建立RTSP会话的方式下,参见图3,主要可以包括以下步骤(仍以视频节目为例):

[0086] S301:终端向服务器发送Describe请求消息,请求从服务器获取媒体内容的描述信息。Describe请求消息中的request-uri为媒体内容的URI。

[0087] S302:服务器向终端发送2000K响应消息,响应消息中包含媒体内容的描述信息。媒体内容的描述信息包括:合控制URI,媒体流的个数,各媒体流对应的RTSP URI,各媒体流递送采用的协议,传输协议参数,媒体编码信息等。此外,还可以包括媒体流中子流的声明信息。

[0088] S303:终端根据媒体内容的描述信息向服务器发送RTSP SETUP请求消息,请求确定媒体流1的传输机制,建立RTSP会话。RTSP SETUP请求消息的request-uri为媒体流1的RTSP URI,RTSP SETUP请求消息中还包含该媒体流的传输参数。

[0089] S304:服务器确定RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。建立RTSP会话,生成RTSP会话标识,向终端发送2000K响应消息并在2000K响应消息中包含上述会话标识。

[0090] S305:终端根据媒体内容的描述信息向服务器发送RTSP SETUP请求消息,请求确定媒体流2的传输机制。RTSP SETUP请求消息的request-uri为媒体流2的RTSP URI,RTSP SETUP请求消息中还包含该媒体流的传输参数。RTSP SETUP请求消息中包含S304中获得的会话标识,表示将媒体流2加到一个合会话中,与会话中已有的媒体流(媒体流1)进行共同控制。

[0091] S306:服务器确定RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。将相应媒体流加到合会话中,向终端发送2000K响应消息。

[0092] 如果媒体内容还包含其他媒体流,重复S305和S306,直至处理完所有媒体流。其中request-uri换成其他媒体流的RTSP URI,相应的传输参数换成该媒体流的传输参数。

[0093] S307:终端向服务器发送控制请求消息。控制请求消息的request-uri为合控制URI。控制请求消息中包括携带有子流对应的媒体流的URI和子流标识信息的头域。还包含S304中获得的会话标识,用于标识请求用于哪个RTSP会话。所述控制请求消息可以是PLAY请求消息或者PAUSE请求消息。

[0094] S308:服务器处理控制请求消息,解析携带有子流对应的媒体流的URI和子流标识信息的头域,获取媒体流的RTSP URI和子流标识信息。

[0095] S309:如果服务器接受子流操控请求,向终端发送2000K响应消息。根据S308获得

的媒体流的RTSP URI确定媒体资源,根据子流标识信息从上述媒体资源中确定子流对应的媒体数据,对子流对应的媒体数据执行请求的播控操作。如果播控请求为PLAY请求消息,向终端发送子流对应的媒体数据;如果播控请求为PAUSE请求消息,停止向终端发送子流对应的媒体数据。

[0096] S310:如果服务器不接受子流播控请求,向终端发送错误响应消息。

[0097] 在另一种建立RTSP会话的方式下,服务器处理RTSP SETUP请求的方式有所不同,相应的,基于子流的控制流程也会略有不同,参见图4,主要可以包括以下步骤:

[0098] S401:终端向服务器发送Describe请求消息,请求从服务器获取媒体内容的描述信息。Describe请求消息中的request-uri为感兴趣的媒体内容的URI。

[0099] S402:服务器向终端发送2000K响应消息,响应消息中包含媒体内容的描述信息。媒体内容的描述信息包括:合控制URI,媒体流的个数,各媒体流对应的RTSP URI,各媒体流递送采用的协议,传输协议参数,媒体编码信息等。还包括媒体流中子流的声明信息。

[0100] S403:终端根据媒体内容的描述信息向服务器发送RTSP SETUP请求消息,请求确定媒体流1的传输机制,建RTSP会话。RTSP SETUP请求消息的request-uri为媒体流1的RTSP URI,RTSP SETUP请求消息中还包含该媒体流的传输参数,以及pipelined-requests头域,该头域用于携带唯一标示一组流水线式请求消息。其中,流水线式请求消息是指,所有的请求消息可以进行顺序发送,而不需要等到收到上一个请求消息的响应之后再发送下一个请求消息。

[0101] S404:终端根据媒体内容的描述信息向服务器发送的RTSP SETUP请求消息,请求确定媒体流2的传输机制。RTSP SETUP请求消息的request-uri为媒体流2的RTSP URI,RTSP SETUP请求消息中还包含该媒体流的传输参数。RTSP SETUP请求消息中还包含pipelined-requests头域,取值与S403中的pipelined-requests头域的取值相同。

[0102] 如果媒体内容还包含其他媒体流,重复S404,直至处理完所有媒体流。其中request-uri换成其他媒体流的RTSP URI,相应的传输参数换成该媒体流的传输参数。但都包含pipelined-requests头域,取值与S403中的pipelined-requests头域的取值相同。

[0103] S405:终端向服务器发送控制请求消息。控制请求消息的request-uri为合控制URI。控制请求消息中包括携带有子流对应的媒体流的URI和子流标识信息的头域。还包含S404中获得的会话标识,用于标识请求用于哪个RTSP会话。所述控制请求消息可以是RTSP PLAY请求消息或者RTSP PAUSE请求消息。控制请求消息中还包含pipelined-requests头域,取值与S403中的pipelined-requests头域的取值相同。

[0104] S406:服务器确定该组流水线式请求消息中的第一个RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。建立RTSP会话,生成RTSP会话标识。服务器确定该组流水线式请求消息中的第二个RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。将相应媒体流加到合会话中。如果媒体内容还包含其他媒体流,也即该组流水线式请求消息中还包括其他RTSP SETUP请求消息,确定其他RTSP SETUP请求消息中的RTSP URI对应的媒体资源是否可用、传输参数是否可以接受等等。将相应媒体流加到合会话中。直至处理完所有媒体流,也即该组流水线式请求消息中所有RTSP SETUP请求消息。

[0105] S407:服务器处理控制请求消息,解析携带有子流对应的媒体流的URI和子流标识

信息的头域,获取媒体流的RTSP URI和子流标识信息。

[0106] S408:如果服务器接受子流操控请求,依次向终端发送对应该组流水线式请求消息中每个RTSP SETUP请求消息的200OK响应消息,其中包括S406中生成的会话标识。向终端发送对应该组流水线式请求消息中控制请求消息的200OK响应消息。根据S407获得的媒体流的RTSP URI确定媒体资源,根据子流标识信息从上述媒体资源中确定子流对应的媒体数据,对子流对应的媒体数据执行请求的操控操作。如果操控请求为RTSP PLAY请求消息,向终端发送子流对应的媒体数据;如果操控请求为RTSP PAUSE请求消息,停止向终端发送子流对应的媒体数据。

[0107] S409:如果服务器不接受子流操控请求,依次向终端发送对应该组流水线式请求消息中每个RTSP SETUP请求消息的200OK响应消息,其中包括步骤6中生成的会话标识。向终端发送对应该组流水线式请求消息中控制请求消息的错误响应消息。

[0108] 与本发明实施例提供的媒体数据控制方法相对应,本发明实施例还提供了一种媒体数据控制装置,该装置位于服务器端,参见图5,该装置包括:

[0109] 第一消息接收单元501,用于接收终端发送的控制请求消息,所述控制请求消息中携带有子流的标识信息,以及所述子流所属媒体流的统一资源标识符URI;

[0110] 其中,所述控制请求消息包括实时流协议RTSP PLAY请求消息或RTSP PAUSE请求消息

[0111] 信息获取单元502,用于获取所述子流的标识信息以及所述子流所属媒体流的URI;

[0112] 数据确定单元503,用于根据所述子流的标识信息以及所述子流所属媒体流的URI,确定所述子流对应的媒体数据;

[0113] 媒体控制单元504,用于基于所述媒体数据,对所述子流执行终端请求的控制操作。

[0114] 其中,当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息通过所述控制请求消息的头域携带,所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带;此时,信息获取单元502具体可以通过解析所述控制请求消息的头域,获取所述子流的标识信息;通过解析所述控制请求消息的request-uri字段,获取所述子流所属媒体流的URI。

[0115] 或者,当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI也可以均通过所述控制请求消息的request-uri字段携带;此时,信息获取单元502具体可以通过解析所述控制请求消息的request-uri字段,获取所述子流的标识信息以及所述子流所属媒体流的URI。

[0116] 当所述控制请求消息为基于多个媒体流的合控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均可以通过所述控制请求消息的头域携带;此时,信息获取单元502具体可以通过解析所述控制请求消息的头域,获取所述子流的标识信息以及所述子流所属媒体流的URI。

[0117] 为了使得终端能够获取到子流的标识信息,可以在接收到终端发送的描述信息请求消息时,在返回响应消息时,将子流声明信息携带在响应消息中。此时,该装置还可以包括:

- [0118] 第二消息接收单元,用于接收终端发送的描述信息请求消息;
- [0119] 第一响应单元,用于向所述终端返回携带有子流声明信息的响应消息,以便所述终端通过所述子流声明信息获取所述子流的标识信息。
- [0120] 其中,子流声明信息可以在所述响应消息的消息体或头域中携带。也就是说,可以将子流的声明信息作为媒体描述信息的一部分,并将媒体描述信息制作成SDP文件。或者,也可以为响应消息生成一个头域,在该头域中携带子流的声明信息。
- [0121] 由于终端在发送控制请求消息时,其中携带的子流标识信息可能不正确,因此,为了保证能够正常进行子流控制,该装置还可以包括:
- [0122] 错误控制单元,用于确定所述控制请求消息中携带的子流的标识信息错误,向所述终端返回携带有子流声明信息的响应消息,以便所述终端根据所述响应消息中携带的子流声明信息重新获取子流的标识信息,并重新发送控制请求消息。
- [0123] 在实际应用中,有些服务器可能不支持子流控制,为了正确实现子流控制,在一种方式下,可以在控制请求消息中还携带有流控制特征标签,此时,该装置还可以包括:
- [0124] 第一子流控制特征标签获取单元,用于获取所述子流控制特征标签;
- [0125] 第一控制单元,用于如果能正确识别所述子流控制特征标签,则触发所述消息解析单元继续执行所述获取所述子流的标识信息以及所述子流所属媒体流的URI及后续操作,否则拒绝所述控制请求,向终端返回携带有不支持子流控制的信息的响应消息。
- [0126] 其中,所述子流控制特征标签可以在所述控制请求消息的require头域中携带;
- [0127] 相应的,第一子流控制特征标签获取单元具体可以用于:解析所述控制请求消息的require头域,获取所述子流控制特征标签。
- [0128] 在另一种实现方式下,也可以在RTSP SETUP请求消息中携带子流控制特征标签,该装置还可以包括:
- [0129] 第二子流控制特征标签获取单元,用于接收所述终端发送的RTSP SETUP请求消息,所述RTSP SETUP请求消息中携带有子流控制特征标签;获取所述子流控制特征标签;
- [0130] 第二控制单元,用于如果能正确识别所述子流控制特征标签,则向终端返回携带有支持子流控制的信息的响应消息,以便所述终端发起子流控制;否则,向终端返回携带有不支持子流控制的信息的响应消息。
- [0131] 具体实现时,所述子流控制特征标签在所述RTSP SETUP请求消息的support头域中携带;
- [0132] 相应的,所述第二子流控制特征标签获取单元具体可以用于:解析所述RTSP SETUP请求消息的support头域,获取所述子流控制特征标签。
- [0133] 在多种编码类型并存的情况下,为了简化服务器的识别过程,终端在发送控制请求消息时,还可以携带有子流的编码类型信息,此时,该装置还可以包括:
- [0134] 编码类型获取单元,用于根据所述子流的编码类型信息获取所述子流的编码类型,以便根据所述编码类型确定所述子流标识信息对应的子流。上述装置的实施例均是在前述方法的实施例基础上进行的介绍,其中未详述部分,可以参见方法实施例中的介绍,这里不再赘述。
- [0135] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分步骤是可以通过程序来指令相关的硬件来完成,所述的程序可以存储于一计算机可读取存储介质中,

该程序在执行时,包括如下步骤:接收终端发送的控制请求消息,所述控制请求消息中携带有子流的标识信息,以及所述子流所属媒体流的统一资源标识符URI;获取所述子流的标识信息以及所述子流所属媒体流的URI;根据所述子流的标识信息以及所述子流所属媒体流的URI,确定所述子流对应的媒体数据;基于所述媒体数据,对所述子流执行终端请求的控制操作。所述的存储介质,如:ROM/RAM、磁碟、光盘等。

[0136] 以上主要是从服务器角度对本发明实施例进行的描述,下面从终端角度对本发明实施例提供的媒体控制方法进行介绍。参见图6,该方法包括以下步骤:

[0137] S601:获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI;

[0138] S602:向服务器发送携带有所述子流的标识信息,以及所述子流所属媒体流的URI的控制请求消息;

[0139] S603:接收到服务器返回的响应消息后,对所述子流进行相应的控制操作。

[0140] 其中,当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息通过所述控制请求消息的头域携带,所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于多个媒体流的合控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的头域携带。

[0141] 其中,具体在需要获取子流的标识信息以及所述子流所属媒体流的URI时,可以向所述服务器发送描述信息请求消息,从所述服务器返回的携带有子流声明信息的响应消息中,获取所述子流的标识信息以及所述子流所属媒体流的URI。其中,子流声明信息可以携带在该响应消息的消息体或头域中,因此,终端可以通过解析该响应消息的消息体或头域,来获取子流声明信息。

[0142] 在实际应用中,由于终端也有可能通过其他途径来获取子流标识信息,因此,发送控制请求消息时携带的子流标识信息有可能是错误的;服务器在发现该错误之后,可以向终端返回携带所有子流正确标识信息的响应消息,因此,终端如果在发送所述控制请求消息之后,接收到服务器返回的携带有子流声明信息的响应消息,则可以根据所述响应消息中携带的子流声明信息重新获取子流的标识信息,并重新发送控制请求消息。

[0143] 另外,在实际应用中,有些服务器可能不支持子流控制,因此,如果不做任何处理,则如果服务器在接收到终端的控制请求消息之后,即使其中携带了子流的标识信息等信息,也可能会进行基于整个媒体流的处理。为了避免这种现象的发生,本发明实施例中,终端在发送控制请求消息时,还可以在控制请求消息中携带有子流控制特征标签,这样,当在服务器接收到控制请求消息之后,如果不支持子流控制,则无法正确识别该子流控制特征标签,进而向终端返回携带有不支持子流控制的信息的响应消息,而不会进行基于整个媒体流的控制。

[0144] 在另一种实现方式下,还可以在向所述服务器发送RTSP SETUP请求消息时,在RTSP SETUP请求消息中携带子流控制特征标签,这样,当所述服务器不支持子流控制时,向终端返回携带有不支持子流控制的信息的响应消息,当然,如果支持子流控制,则可以向终端返回支持子流控制的响应消息。相应的,如果终端从服务器的响应消息中解析出服务器支持子流控制,则可以向该服务器发起基于子流的控制请求;否则,如果解析出服务器不支

持子流控制,就不会向该服务器发起基于子流的控制请求。

[0145] 在多种编码类型并存的情况下,为了简化服务器的识别过程,终端还可以控制请求消息中携带流的编码类型信息,以便所述服务器根据所述子流的编码类型信息获取所述子流的编码类型,根据所述编码类型确定所述子流标识信息对应的子流。

[0146] 与上述方法相对应,本发明实施例还提供了一种媒体数据控制装置,参见图7,该装置包括:

[0147] 子流信息获取单元701,用于获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI;

[0148] 消息发送单元702,用于向服务器发送携带有所述子流的标识信息,以及所述子流所属媒体流的URI的控制请求消息;

[0149] 操作执行单元703,用于接收到服务器返回的响应消息后,对所述子流进行相应的控制操作。

[0150] 其中,当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息通过所述控制请求消息的头域携带,所述子流所属媒体流的URI通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于单个媒体流的单独控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的request-uri字段携带;或当所述控制请求消息为基于多个媒体流的合控制请求消息时,所述子流的标识信息及所述子流所属媒体流的URI均通过所述控制请求消息的头域携带。

[0151] 在一种实现方式下,子流信息获取单元701可以包括:

[0152] 描述信息请求消息发送子单元,用于向所述服务器发送描述信息请求消息;

[0153] 获取子单元,用于从所述服务器返回的携带有子流声明信息的响应消息中,获取所述子流的标识信息以及所述子流所属媒体流的URI。

[0154] 由于终端也有可能通过其他途径来获取子流标识信息,因此,发送控制请求消息时携带的子流标识信息有可能是错误的;服务器在发现该错误之后,可以向终端返回携带有所有子流正确标识信息的响应消息,相应的,该装置还可以包括:

[0155] 重新发送单元,用于如果在发送所述控制请求消息之后,接收到服务器返回的携带有子流声明信息的响应消息,则根据所述响应消息中携带的子流声明信息重新获取子流的标识信息,并重新发送控制请求消息。

[0156] 另外,在实际应用中,有些服务器可能不支持子流控制,因此,如果不做任何处理,则如果服务器在接收到终端的控制请求消息之后,即使其中携带了子流的标识信息等信息,也可能会进行基于整个媒体流的处理。为了避免这种现象的发生,所述控制请求消息中还携带有子流控制特征标签,以便当所述服务器不支持子流控制时,向终端返回携带有不支持子流控制的信息的响应消息。

[0157] 或者,该装置还可以包括:

[0158] RTSP SETUP请求消息发送单元,用于向所述服务器发送RTSP SETUP请求消息,并在所述RTSP SETUP请求消息中携带子流控制特征标签,以便当所述服务器不支持子流控制时,向终端返回携带有不支持子流控制的信息的响应消息。

[0159] 在多种编码类型并存的情况下,为了简化服务器的识别过程,所述控制请求消息中还携带有子流的编码类型信息,以便所述服务器根据所述子流的编码类型信息获取所述

子流的编码类型,根据所述编码类型确定所述子流标识信息对应的子流。

[0160] 需要说明的是,从终端角度描述的媒体数据控制方法及装置,是与前述从服务器角度描述的媒体数据控制方法及装置相对应的,因此,其中未详述部分,可以参见前文的介绍,这里不再赘述。

[0161] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分步骤是可以通过程序来指令相关的硬件来完成,所述的程序可以存储于一计算机可读取存储介质中,该程序在执行时,包括如下步骤:获取子流的标识信息以及所述子流所属媒体流的统一资源标识符URI;向服务器发送携带有所述子流的标识信息,以及所述子流所属媒体流的URI的控制请求消息;接收到服务器返回的响应消息后,对所述子流进行相应的控制操作。所述的存储介质,如:ROM/RAM、磁碟、光盘等。

[0162] 需要说明的是,本发明实施例所述的终端可以是手机、PDA、笔记本、计算机等等;服务器可以是基站、媒体服务器等等。此外上述所有实施例步骤的执行都可以是服务器或终端的处理器。

[0163] 以上对本发明所提供的媒体数据控制方法及装置,进行了详细介绍,本文中应用了具体个例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处。综上所述,本说明书内容不应理解为对本发明的限制。

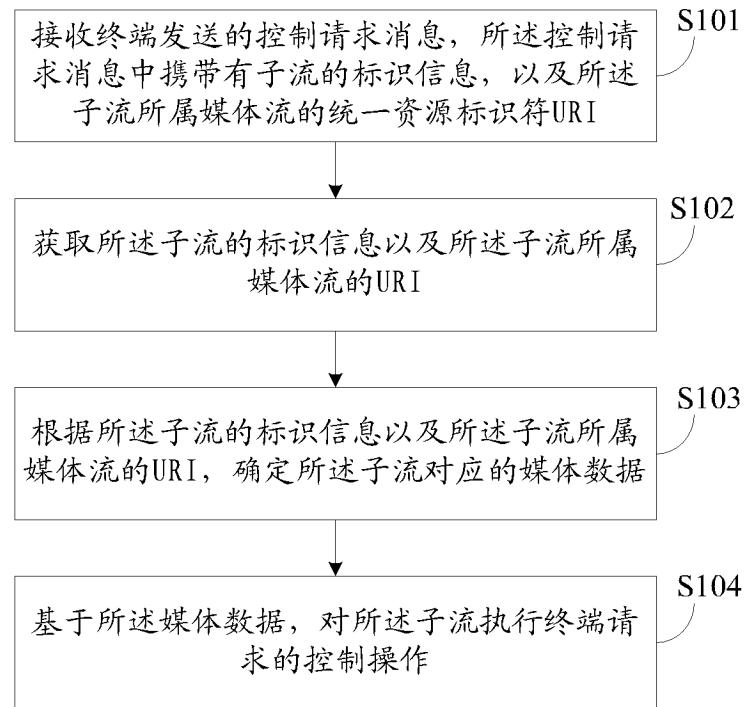


图1

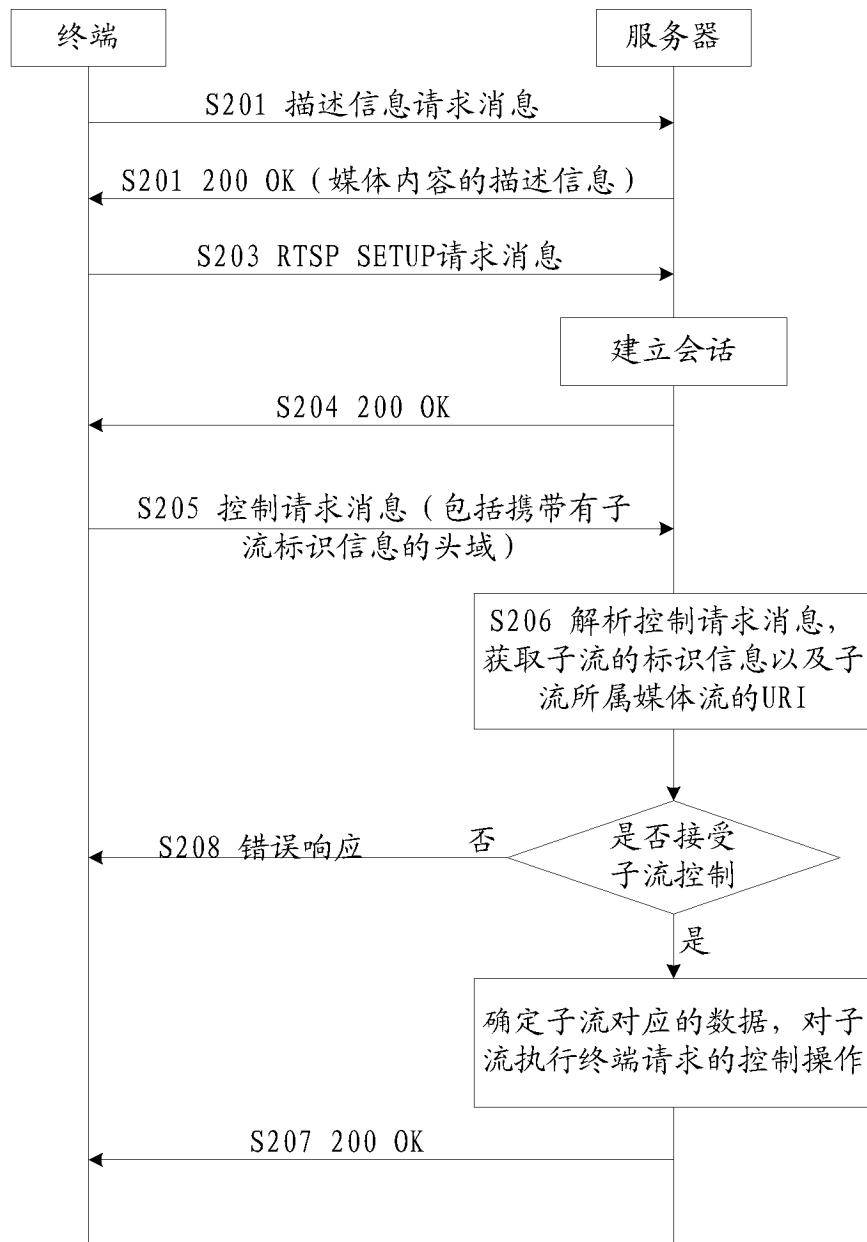


图2

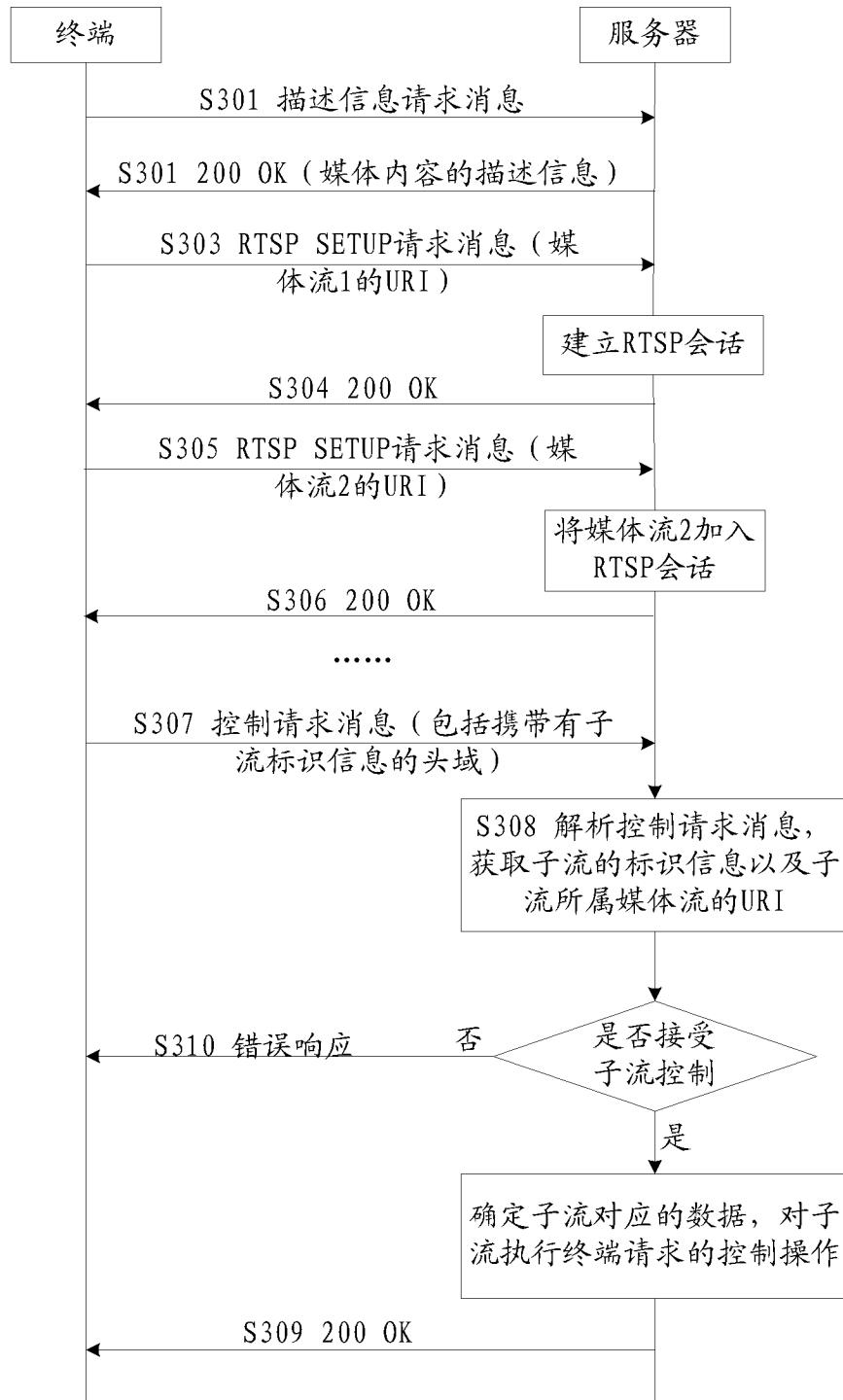


图3

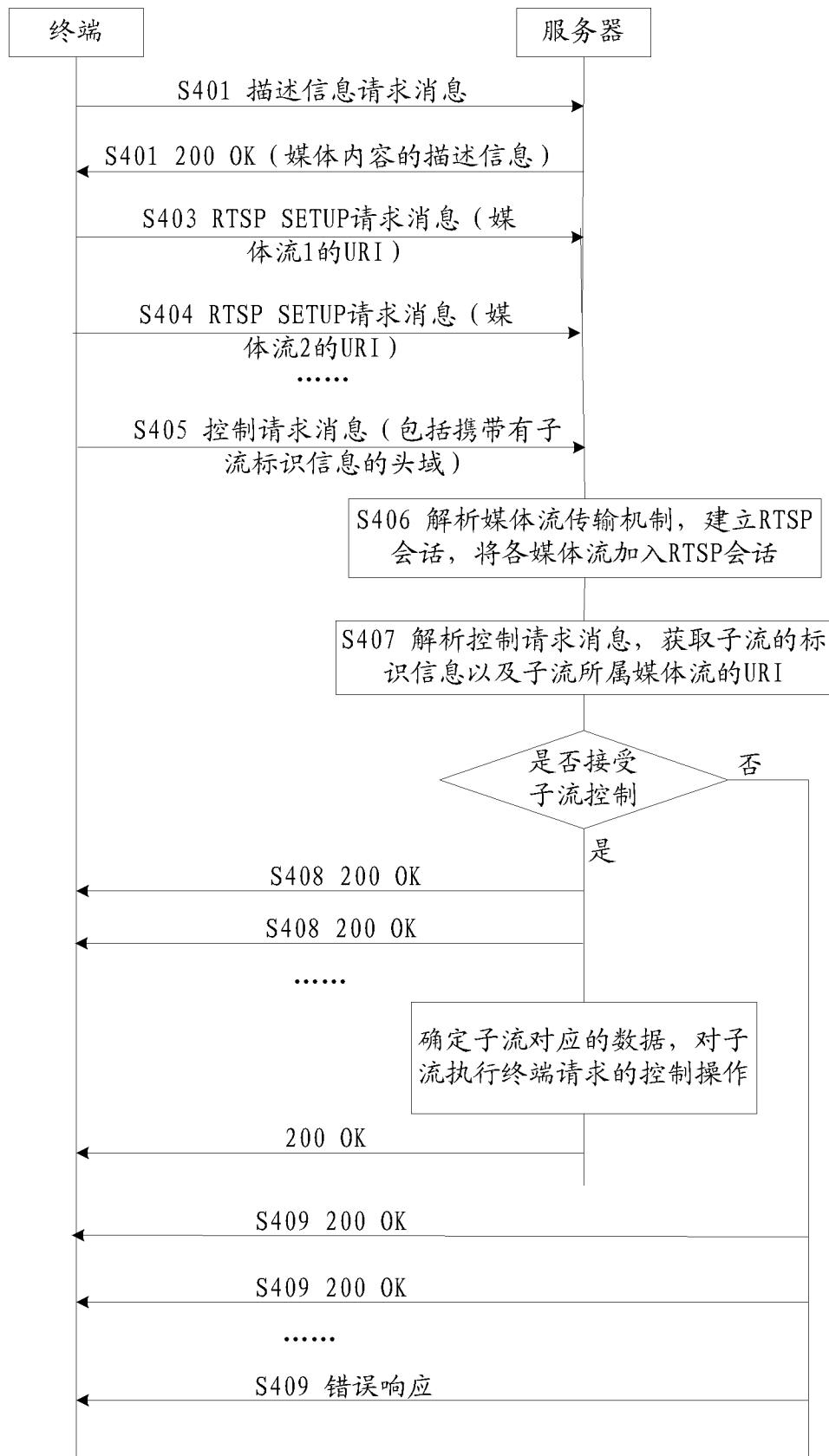


图4

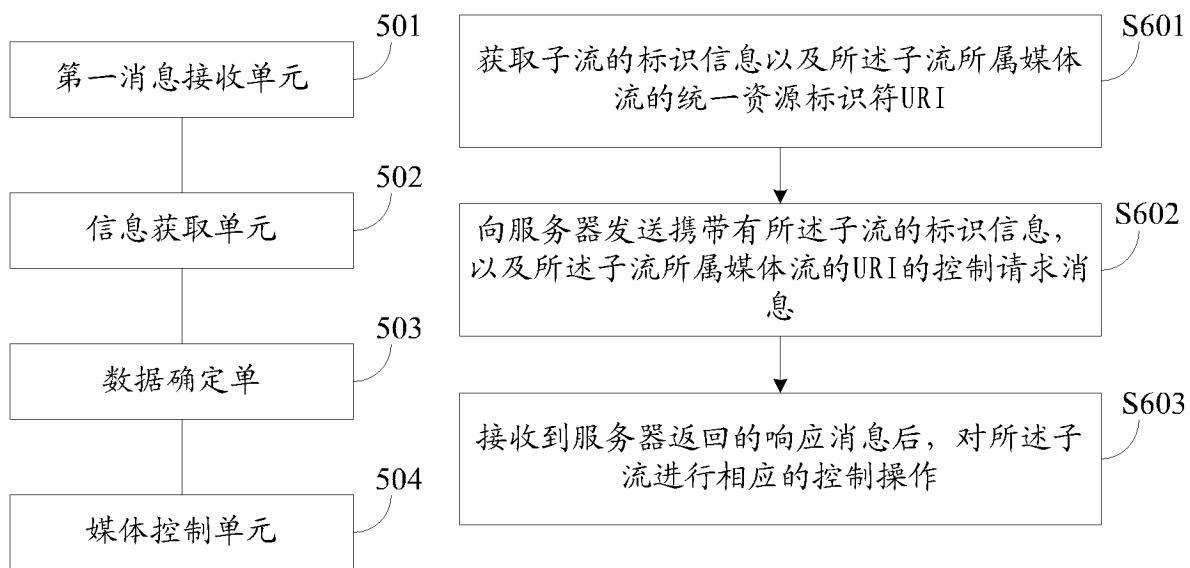


图5

图6

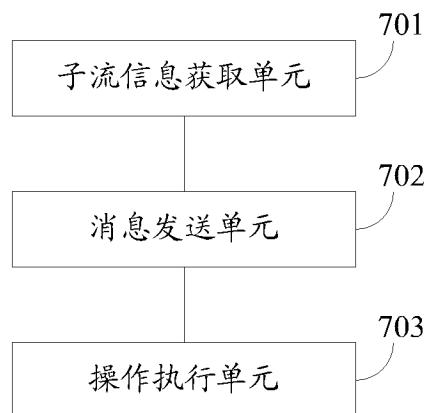


图7