



(12) 发明专利

(10) 授权公告号 CN 109587551 B

(45) 授权公告日 2020.12.25

(21) 申请号 201710909775.4

H04N 21/647 (2011.01)

(22) 申请日 2017.09.29

H04L 29/06 (2006.01)

(65) 同一申请的已公布的文献号
申请公布号 CN 109587551 A

(56) 对比文件

EP 0725506 A2, 1996.08.07

CN 105847977 A, 2016.08.10

(43) 申请公布日 2019.04.05

审查员 姚臣益

(73) 专利权人 北京金山云网络技术有限公司
地址 100085 北京市海淀区小营西路33号
3F02室

专利权人 北京金山云科技有限公司

(72) 发明人 武爱敏 王文博

(74) 专利代理机构 北京柏杉松知识产权代理事
务所(普通合伙) 11413

代理人 马敬 项京

(51) Int. Cl.

H04N 21/439 (2011.01)

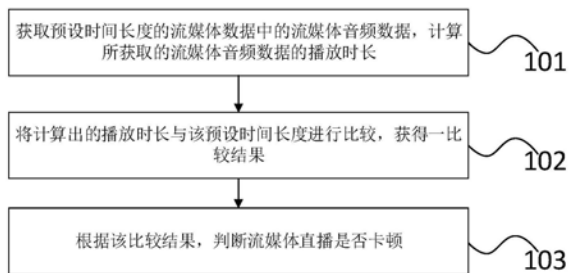
权利要求书3页 说明书18页 附图3页

(54) 发明名称

一种流媒体直播卡顿的判断方法、装置、设备及存储介质

(57) 摘要

本发明实施例提供了一种流媒体直播卡顿的判断方法、装置、设备及存储介质,涉及通信技术领域,解决现有技术存在对流媒体是否出现卡顿判断不准确的问题。其中,该流媒体直播卡顿的判断方法,应用在流媒体直播过程中接收流媒体的设备,所述方法包括:获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;根据所述比较结果,判断流媒体直播是否卡顿,以实现准确地判断流媒体直播是否出现卡顿。



1. 一种流媒体直播卡顿的判断方法,其特征在于,应用在流媒体直播过程中接收流媒体的设备,所述方法包括:

获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;

根据所述比较结果,判断流媒体直播是否卡顿;

其中,所述根据所述比较结果,判断流媒体直播是否卡顿的步骤包括:

在所述比较结果表明计算出的播放时长小于或大于所述预设时间长度时,

判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿,其中,所述预设条件表明所述流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

2. 根据权利要求1所述的方法,其特征在于,所述获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长的步骤包括:

获取所述预设时间长度的流媒体数据中的流媒体音频数据;

记录所述预设时间长度的第一个音频数据包内的第一时间戳;

记录所述预设时间长度的最后一个音频数据包内的第二时间戳;

计算所述最后一个音频数据包内的第二时间戳与所述第一个音频数据包内的第一时间戳之间差值的绝对值,获得所获取的流媒体音频数据的播放时长。

3. 根据权利要求1所述的方法,其特征在于,所述根据所述比较结果,判断流媒体直播是否卡顿的步骤包括:

在所述比较结果表明计算出的播放时长等于所述预设时间长度时,

判定流媒体直播未卡顿。

4. 根据权利要求1所述的方法,其特征在于,

在所述比较结果表明计算出的播放时长大于所述预设时间长度时,

所述判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:判断所述预设时间长度的所获取的流媒体音频数据的数据量是否大于所述数据缓冲区所能容纳的数据量,

若所述预设时间长度的所获取的流媒体音频数据的数据量大于所述数据缓冲区所能容纳的数据量,判定流媒体直播卡顿;

若所述预设时间长度的所获取的流媒体音频数据的数据量小于或等于所述数据缓冲区所能容纳的数据量,判定流媒体直播未卡顿。

5. 根据权利要求4所述的方法,其特征在于,若判断流媒体直播卡顿,所述方法还包括:

确定溢出所述数据缓冲区的数据量,将所溢出的数据量删除。

6. 根据权利要求1所述的方法,其特征在于,

在所述比较结果表明计算出的播放时长小于所述预设时间长度时,

所述判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:判断所述数据缓冲区内缓存的数据量是否为零,且所获取的流媒体音频数据的数据量是否为零,

若所述数据缓冲区内缓存的数据量为零,且所获取的流媒体音频数据的数据量为零,判定流媒体直播卡顿;

若所述数据缓冲区内缓存的数据量大于零,且所获取的流媒体音频数据的数据量大于零,判定流媒体直播未卡顿。

7. 一种流媒体直播卡顿的判断装置,其特征在于,应用在流媒体直播过程中接收流媒体的设备,所述装置包括:

获取模块,用于获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

比较获得模块,用于将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;

判断模块,用于根据所述比较结果,判断流媒体直播是否卡顿;

所述判断模块包括:

第二判断子模块,用于在所述比较结果表明计算出的播放时长小于或大于所述预设时间长度时,判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿,其中,所述预设条件表明所述流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

8. 根据权利要求7所述的装置,其特征在于,所述获取模块包括:

第一获取模块,用于获取所述预设时间长度的流媒体数据中的流媒体音频数据;

第一记录模块,用于记录所述预设时间长度的第一个音频数据包内的第一时间戳;

第二记录模块,用于记录所述预设时间长度的最后一个音频数据包内的第二时间戳;

计算获得模块,用于计算所述最后一个音频数据包内的第二时间戳与所述第一个音频数据包内的第一时间戳之间差值的绝对值,获得所获取的流媒体音频数据的播放时长。

9. 根据权利要求7所述的装置,其特征在于,所述判断模块包括:

第一判断子模块,用于在所述比较结果表明计算出的播放时长等于所述预设时间长度时,判定流媒体直播未卡顿。

10. 根据权利要求7所述的装置,其特征在于,

所述第二判断子模块具体用于:在所述比较结果表明计算出的播放时长大于所述预设时间长度时,判断所述预设时间长度的所获取的流媒体音频数据的数据量是否大于所述数据缓冲区所能容纳的数据量,

若所述预设时间长度的所获取的流媒体音频数据的数据量大于所述数据缓冲区所能容纳的数据量,判定流媒体直播卡顿;

若所述预设时间长度的所获取的流媒体音频数据的数据量小于或等于所述数据缓冲区所能容纳的数据量,判定流媒体直播未卡顿。

11. 根据权利要求10所述的装置,其特征在于,所述装置还包括:

删除模块,用于确定溢出所述数据缓冲区的数据量,将所溢出的数据量删除。

12. 根据权利要求7所述的装置,其特征在于,

所述第二判断子模块具体用于:在所述比较结果表明计算出的播放时长小于所述预设时间长度时,判断所述数据缓冲区内缓存的数据量是否为零,且所获取的流媒体音频数据的数据量是否为零,

若所述数据缓冲区内缓存的数据量为零,且所获取的流媒体音频数据的数据量为零,判定流媒体直播卡顿;

若所述数据缓冲区内缓存的数据量大于零,且所获取的流媒体音频数据的数据量大于零,判定流媒体直播未卡顿。

13.一种在流媒体直播过程中接收流媒体的设备,其特征在于,包括处理器和存储器,所述存储器,用于存放计算机程序;

所述处理器,用于执行所述存储器上所存放的程序时,实现权利要求1-6任一所述的方法步骤。

14.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质内存储有计算机程序,所述计算机程序被处理器执行时实现权利要求1-6任一所述的方法步骤。

一种流媒体直播卡顿的判断方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及通信技术领域,特别是涉及一种流媒体直播卡顿的判断方法、装置、设备及存储介质。

背景技术

[0002] 目前的流媒体是指采用流式传输的方式,在因特网Internet播放的媒体格式。流媒体又叫流式媒体,它是指商家用一个视频传送服务器把节目当成数据包发出,传送到网络上。用户通过解压设备对这些数据进行解压后,节目就会像发送前那样显示出来。

[0003] 一般流媒体直播从信号采集端到解码接收端,会经过许多环节,由于流媒体实时传输的特性对于网络质量有非常高的要求,会经常遇到卡顿等类似的影响观看体验的问题,所以非常必要地通过一些有效的手段或方法对直播卡顿情况进行检验和分析。

[0004] 目前应用比较广泛的卡顿判断方法主要有以下两种:

[0005] 1、通过对直播数据中的视频压缩码率变化情况进行分析;

[0006] 获取流媒体的视频直播数据;

[0007] 通过CBR(Constant bit rate,固定比特率编码)的压缩方式,对该视频直播数据进行压缩;

[0008] 将压缩后的视频直播数据进行传输,对比当前时间点的视频比特率与预设时间点对应的视频比特率之间的差值;如果该差值大于根据用户需求设置的第一预设阈值,说明传输速度不足,则判断视频直播数据出现卡顿;如果该差值小于根据用户需求设置的第一预设阈值,说明传输速度足够,则判断视频直播数据未出现卡顿。前述预设时间点在前述的当前时间之前。

[0009] 2、通过对直播数据中的视频帧率变化情况进行分析:

[0010] 基于视频编码的技术基础,编码器在媒体头中写入设定的理论码率、理论帧率等信息,该信息用来帮助解码器正确识别被压缩的数据;

[0011] 解码器通过读取媒体头中的信息,得到该流媒体的视频直播数据在压缩编码时所设定的理论码率及理论帧率,理论帧率表示每秒钟有多少幅被压缩的视频图像;

[0012] 解码器获取每秒钟实际接收解码的帧率,对比理论帧率的数值与实际接收解码的帧率的差值,在该差值大于根据用户需求进行设置的第二预设阈值时,说明出现数据丢失,则判断视频直播数据出现卡顿;在该差值小于根据用户需求进行设置的第二预设阈值时,说明出现数据足够传输,则判断视频直播数据未出现卡顿,其中,该实际接收解码的帧率为每秒钟实际接收解码的视频图像数目。

[0013] 以上现有技术的两种卡顿判断方法,都是通过流媒体中视频的视频信息比如视频比特率或者帧率,判断视频直播是否出现卡顿。但是,视频直播数据在实际网络传输过程中,网络可能存在不太稳定的状态,网络为了保证视频直播数据能够传输,实际处理会主动丢弃部分视频帧,这样网络主动丢弃的视频,并不是实际的卡顿,但是利用上述两种卡顿判断方法,可能会判断为视频直播数据出现卡顿,因而,现有技术存在对流媒体是否出现卡顿

判断不准确的问题。

发明内容

[0014] 本发明实施例的目的在于提供一种流媒体直播卡顿的判断方法、装置、设备及存储介质,以实现准确地判断流媒体直播是否出现卡顿。具体技术方案如下:

[0015] 第一方面,本发明实施例提供了一种流媒体直播卡顿的判断方法,应用在流媒体直播过程中接收流媒体的设备,上述方法包括:

[0016] 获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

[0017] 将计算出的播放时长与上述预设时间长度进行比较,获得一比较结果;

[0018] 根据上述比较结果,判断流媒体直播是否卡顿。

[0019] 可选的,上述获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长的步骤包括:

[0020] 获取上述预设时间长度的流媒体数据中的流媒体音频数据;

[0021] 记录上述预设时间长度的第一个音频数据包内的第一时间戳;

[0022] 记录上述预设时间长度的最后一个音频数据包内的第二时间戳;

[0023] 计算上述最后一个音频数据包内的第二时间戳与上述第一个音频数据包内的第一时间戳之间差值的绝对值,获得所获取的流媒体音频数据的播放时长。

[0024] 可选的,上述根据上述比较结果,判断流媒体直播是否卡顿的步骤包括:

[0025] 在上述比较结果表明计算出的播放时长等于上述预设时间长度时,

[0026] 判定流媒体直播未卡顿。

[0027] 可选的,上述根据上述比较结果,判断流媒体直播是否卡顿的步骤包括:

[0028] 在上述比较结果表明计算出的播放时长小于或大于上述预设时间长度时,

[0029] 判断上述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿,其中,上述预设条件表明上述流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

[0030] 可选的,在上述比较结果表明计算出的播放时长大于上述预设时间长度时,

[0031] 上述判断上述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:判断上述预设时间长度的所获取的流媒体音频数据的数据量是否大于上述数据缓冲区所能容纳的数据量,

[0032] 若上述预设时间长度的所获取的流媒体音频数据的数据量大于上述数据缓冲区所能容纳的数据量,判定流媒体直播卡顿;

[0033] 若上述预设时间长度的所获取的流媒体音频数据的数据量小于或等于上述数据缓冲区所能容纳的数据量,判定流媒体直播未卡顿。

[0034] 可选的,若判断流媒体直播卡顿,上述方法还包括:

[0035] 确定溢出上述数据缓冲区的数据量,将所溢出的数据量删除。

[0036] 可选的,在上述比较结果表明计算出的播放时长小于上述预设时间长度时,

[0037] 上述判断上述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:判断上述数据缓冲区内缓存的数据量是否为零,且所获取的流媒体音频

数据的数据量是否为零，

[0038] 若上述数据缓冲区内缓存的数据量为零，且所获取的流媒体音频数据的数据量为零，判定流媒体直播卡顿；

[0039] 若上述数据缓冲区内缓存的数据量大于零，且所获取的流媒体音频数据的数据量大于零，判定流媒体直播未卡顿。

[0040] 第二方面，本发明实施例还提供了一种流媒体直播卡顿的判断装置，应用在流媒体直播过程中接收流媒体的设备，上述装置包括：

[0041] 获取模块，用于获取预设时间长度的流媒体数据中的流媒体音频数据，计算所获取的流媒体音频数据的播放时长；

[0042] 比较获得模块，用于将计算出的播放时长与上述预设时间长度进行比较，获得一比较结果；

[0043] 判断模块，用于根据上述比较结果，判断流媒体直播是否卡顿。

[0044] 可选的，第一获取模块，用于获取上述预设时间长度的流媒体数据中的流媒体音频数据；

[0045] 第一记录模块，用于记录上述预设时间长度的第一个音频数据包内的第一时间戳；

[0046] 第二记录模块，用于记录上述预设时间长度的最后一个音频数据包内的第二时间戳；

[0047] 计算获得模块，用于计算上述最后一个音频数据包内的第二时间戳与上述第一个音频数据包内的第一时间戳之间差值的绝对值，获得所获取的流媒体音频数据的播放时长。

[0048] 可选的，第一判断子模块，用于在上述比较结果表明计算出的播放时长等于上述预设时间长度时，判定流媒体直播未卡顿。

[0049] 可选的，第二判断子模块，用于在上述比较结果表明计算出的播放时长小于或大于上述预设时间长度时，判断上述流媒体音频数据的数据量是否满足预设条件，判断流媒体直播是否卡顿，其中，上述预设条件表明上述流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

[0050] 可选的，上述第二判断子模块具体用于：在上述比较结果表明计算出的播放时长大于上述预设时间长度时，判断上述预设时间长度的所获取的流媒体音频数据的数据量是否大于上述数据缓冲区所能容纳的数据量，

[0051] 若上述预设时间长度的所获取的流媒体音频数据的数据量大于上述数据缓冲区所能容纳的数据量，判定流媒体直播卡顿；

[0052] 若上述预设时间长度的所获取的流媒体音频数据的数据量小于或等于上述数据缓冲区所能容纳的数据量，判定流媒体直播未卡顿。

[0053] 可选的，删除模块，用于确定溢出上述数据缓冲区的数据量，将所溢出的数据量删除。

[0054] 可选的，上述第二判断子模块具体用于：在上述比较结果表明计算出的播放时长小于上述预设时间长度时，判断上述数据缓冲区内缓存的数据量是否为零，且所获取的流媒体音频数据的数据量是否为零，

[0055] 若上述数据缓冲区内缓存的数据量为零,且所获取的流媒体音频数据的数据量为零,判定流媒体直播卡顿;

[0056] 若上述数据缓冲区内缓存的数据量大于零,且所获取的流媒体音频数据的数据量大于零,判定流媒体直播未卡顿。

[0057] 第三方面,本发明实施例提供了一种在流媒体直播过程中接收流媒体的设备,包括处理器和存储器,存储器,用于存放计算机程序;

[0058] 处理器,用于执行存储器上所存放的程序时,实现第一方面上述的步骤。

[0059] 第四方面,本发明实施例还提供了一种在流媒体直播过程中接收流媒体的设备,包括处理器、通信接口、存储器和通信总线,其中,处理器,通信接口,存储器通过总线完成相互间的通信,

[0060] 存储器,用于存放计算机程序;

[0061] 处理器,用于执行存储器上所存放的程序时,实现第一方面上述的步骤。

[0062] 第五方面,本发明实施例还提供了一种计算机可读存储介质,上述计算机可读存储介质内存储有计算机程序,上述计算机程序被处理器执行时实现第一方面上述的步骤。

[0063] 第六方面,本发明实施例提供了一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述第一方面上述的步骤。

[0064] 第七方面,本发明实施例提供了一种计算机程序,当其在计算机上运行时,使得计算机执行上述第一方面上述的步骤。

[0065] 本发明实施例提供的流媒体直播卡顿的判断方法、装置、设备及存储介质,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。当然,实施本发明实施例的任一产品或方法必不一定需要同时达到以上所述的所有优点。

附图说明

[0066] 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,下面将对实施例或现有技术描述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。

[0067] 图1为本发明实施例的流媒体直播卡顿的判断方法的第一种流程示意图;

[0068] 图2为本发明实施例图1中步骤101的具体流程示意图;

[0069] 图3为本发明实施例的流媒体直播卡顿的判断方法的第二种流程示意图;

[0070] 图4为本发明实施例的流媒体直播卡顿的判断装置的结构示意图;

[0071] 图5为本发明实施例的在流媒体直播过程中接收流媒体的设备的结构示意图。

具体实施方式

[0072] 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在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明实施例保护的范围。

[0073] 由于现有技术中是通过通过对通过流媒体数据中视频的视频信息比如视频比特率或者帧率,来判断视频直播是否出现卡顿。而且在实际视频直播数据传输过程中,可能由于网络存在不太稳定的状态,网络实际主动会丢弃部分视频,以保证视频直播数据能够传输。

[0074] 现有技术只是流媒体数据中视频的视频信息进行处理,判断视频直播是否出现卡顿。现有技术并不存在对流媒体数据中的流媒体音频数据的处理或者压缩,判断视频直播是否出现卡顿,因此基于上述流媒体音频数据不做任何处理的基础上,发明人还考虑到流媒体数据传输过程中视频数据的卡顿并不影响流媒体音频数据。也就是,在流媒体数据实际传输过程中,视频数据按照自身的播放要求进行播放,流媒体音频数据可以按照自身的播放要求进行播放,只要实现流媒体数据中的视频数据和流媒体音频数据的同步传输即可完成流媒体的传输。其中,该自身的播放要求由网络的实际传输需求及流媒体播放端的播放能力决定。比如,该实际传输要求包括:视频码率;播放能力包括:网络带宽、视频清晰度及网络吞吐量。

[0075] 上述流媒体音频数据在实际播放过程中,一旦出现中断或者错误,则可以准确地判断流媒体直播出现卡顿。基于上述发明人考虑,本发明实施例为了解决现有技术存在流媒体直播卡顿误判的问题,提供了一种流媒体直播卡顿的判断方法、装置、设备及存储介质,利用流媒体数据中的流媒体音频数据进行卡顿的判断,从而实现流媒体直播卡顿的准确判断,对于后期改善卡顿作好基础,提高了流媒体直播的顺畅度,进而提高了用户体验效果。

[0076] 下面首先对本发明实施例所提供的流媒体直播卡顿的判断方法进行介绍。

[0077] 需要说明的是,本发明实施例所提供的流媒体直播卡顿的判断方法也可以应用在流媒体直播过程中接收流媒体的设备。该设备具体可以为:台式计算机、便携式计算机、智能移动终端、移动终端,服务器等。可以应用于在流媒体直播过程中接收流媒体的设备内的客户端,在此不作限定,任何可以实现本发明实施例的设备,均属于本发明实施例的保护范围。

[0078] 如图1所示,本发明实施例所提供的一种流媒体直播卡顿的判断方法,应用在流媒体直播过程中接收流媒体的设备,可以包括如下步骤:

[0079] 这里的应用在流媒体直播过程中发送流媒体的设备,在本发明实施例后续介绍中简称为设备。

[0080] 步骤101,获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长。

[0081] 流媒体数据指将一连串的媒体数据压缩后,经过网络分段发送压缩后的流媒体数据。该流媒体数据在网络上即时传输影音,以供用户观赏。该流媒体数据中流媒体音频数据指人耳能够听到的频率范围内的声音。

[0082] 预设时间长度可以根据用户需求进行设定。该预设时间长度为该设备的操作系统时间。一般该预设时间长度最短为1秒,在此1秒的基础上可以增加时长,本发明实施例中,对该预设时间长度的上限不做限定。在满足带宽需求的情况下设任何预设时间长度,均属

于本发明实施例的保护范围。该带宽需求的情况可以是指传播的流媒体数据小于或者等于的传输带宽。该带宽需求用于保证流媒体数据的正常传输。示例性的,在该预设时间长度设置为1秒时,该1秒可以是指系统时间戳为2017年6月26日星期一14点01分05秒,与系统时间戳为2017年6月26日星期一14点01分06秒之间的1秒。本发明实施例的,预设时间长度越小,则判断流媒体直播是否出现卡顿越准确。

[0083] 上述的系统时间戳,是指在一连串的资料中加入时间信息,如年月日时间,此处的一连串的资料可以为传播的流媒体数据。

[0084] 可选的,在一种具体实现方式中,这里步骤101包括:按照预设周期,获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长。其中,该预设周期大于或等于该预设时间长度,这样保证设备在该预设周期内获取所有流媒体数据中的流媒体音频数据,使得对该流媒体音频数据完成播放时长的计算。

[0085] 该流媒体音频数据可以包括每个流媒体音频数据包的时间信息,每个流媒体音频数据包的数据量,以及流媒体音频数据包的总数据量。该时间信息可以是指时间戳,也可以是指数据包正常的传输时间对应的的时间长度,该数据包正常的传输时间对应的的时间长度可以根据实验得出,也可以根据用户需要进行设置。任何能够获取流媒体音频数据的播放时长的任何时间信息,均属于本发明实施例的保护范围,在此不一一举例。

[0086] 步骤102,将计算出的播放时长与该预设时间长度进行比较,获得一比较结果。

[0087] 步骤103,根据该比较结果,判断流媒体直播是否卡顿。

[0088] 流媒体直播是指经过广播电视网、互联网等媒介播出的节目内容与现场事件同步进行的传输方式。该卡顿是指按照固定频率或速度传输数据时,出现固定频率降低或者速度降低的情况,引发的不流畅地观看体验。

[0089] 本发明实施例提供的流媒体直播卡顿的判断方法,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。

[0090] 为了更加快速地确定出流媒体音频数据的播放时长,如图2所示,可选的,在一种具体实现方式中,上述步骤101包括如下步骤:

[0091] 步骤201,获取该预设时间长度的流媒体数据中的流媒体音频数据。

[0092] 流媒体数据传输过程中都是通过数据包进行数据传输,实现上述流媒体音频数据的获取步骤如下:

[0093] 应用在流媒体直播过程中发送流媒体的设备,通过网络或者本地存储得到流媒体数据,提取出该流媒体数据中的流媒体音频数据和流媒体视频数据。

[0094] 根据用户设定的预设采样率,利用编码器按照预设采样频率对该流媒体音频数据进行采样编码,得到每秒的流媒体音频数据的音频帧率,利用该音频帧率计算出来解码器所需解码的时长,编码器以该时长为打包周期,周期性的打包一个流媒体音频数据包,并在这个流媒体音频数据包中写入时长,以保证在编码器编码数据时,按照音频帧率,对每个流媒体音频数据包进行编码。

[0095] 解码器接收到编码的流媒体数据后,会参考前述时长对流媒体视频数据包进行解

码。正常情况下,每两个连续的流媒体数据包的时间差值应该均在该时长内,不出现大幅度的跳变。

[0096] 应用在流媒体直播过程中接收流媒体的设备,读取该流媒体数据包,能够获取该流媒体数据中的流媒体音频数据包、该预设采样率及该时长。这样可以保证接收到流媒体数据包的设备能够解码所接收到的流媒体数据。

[0097] 示例性的,上述的预设采样率可以利用AAC (Advanced Audio Coding,高级音频编码) 进行编码压缩,并将预设采样率设置为44.1kHz。

[0098] AAC格式在压缩过程中会将采样点进行打包。打包方式为每1024个采样点打一个包。每个包称之为一个AAC音频帧。这样,44100个采样点按照上打包方式进行打包,就会产生43个音频帧(即44100除以1024约等于43),也就是,得到的音频帧率为43fps。利用该音频帧率,可计算出每个音频帧所需要的解码时长约为23毫秒(即1000除以43约等于23)。当然此处仅以AAC的音频编解码技术为例进行说明,其他的MP3音频编解码技术同样适用,任何能够实现本发明实施例的音频编解码技术,均属于本发明实施例的保护范围,在此不一一举例。

[0099] 上述音频的帧率是指声音在从模拟到数字化的采集转换过程中会对固定数量的采样点进行打包,每一秒钟打包音频包的数量,该固定数量比如为上述的1024。

[0100] AAC的格式可以提供较好的音质及较高的压缩率。衡量音频质量最重要的一个参考指标为采样率,即在声音从模拟波形到数字化的过程中,每秒钟对于声波波幅所做的数字化信息采集的样本的数量。人耳能够听到的最高的采样率为20kHz,为了避免压缩、传输等过程造成的声音失真,应用的音频采样率为44.1kHz,即每秒有44100个采样点。

[0101] 步骤202,记录该预设时间长度的第一个音频数据包内的第一时间戳。

[0102] 第一个音频数据包主要由“目的IP (Internet Protocol,互联网协议) 地址”、“源IP地址”、“净载数据”以及标识部分构成。该第一个音频数据包包括包头和包体,包头是固定长度,包体的长度不定,各字段长度固定,双方的请求数据包和应答数据包的包头结构是一致的。

[0103] 上述的源IP地址是指第一个音频数据包的发送设备地址,本发明实施例的第一个音频数据包的发送地址可以为信号采集编码端。上述的目的IP地址是指第一个音频数据包的接收设备地址。上述的净载数据可以为流媒体音频数据,上述的标识部分可以是数据总数据量的标识,也可以是每个数据包的数据量的标识。

[0104] 第一个音频数据包及第一个音频数据包对应的时间戳可以记录在本地硬盘中,也可以是记录在预先建立网络连接的网络虚拟磁盘中,任何能够实现记录该音频数据包及该音频包对应的时间戳的位置,均属于本发明实施例的保护范围。通过记录该时间戳,可以方便后期用户直接使用或者设备在该记录的位置读取该时间戳进行直接使用,不用实时获取该时间戳时占用运行内存,不仅减轻了运行内存及带宽的负担,而且也提高了数据处理的效率。

[0105] 第一个音频数据包及第一个音频数据包对应的时间戳可以记录在临时缓存中,这样可以在预设时间长度后,将使用完成的时间戳进行删除,减少存储空间的压力。任何能够实现记录该时间戳的方式,均属于本发明实施例的保护范围,在此不一一举例说明。

[0106] 将记录第一个音频数据包内的时间戳作为第一个音频数据包内的第一时间戳DS,

该第一个音频数据包内的第一时间戳DS包括:编码器按照预设采样率对该流媒体音频数据进行采样编码,并且在这个流媒体音频数据包中写入上述时长的起始时间戳。

[0107] 步骤203,记录该预设时间长度的最后一个音频数据包内的第二时间戳。

[0108] 最后一个音频数据包主要由“目的IP(Internet Protocol,互联网协议)地址”、“源IP地址”、“净载数据”以及标识部分构成。该最后一个音频数据包包括包头和包体,包头是固定长度,包体的长度不定,各字段长度固定,双方的请求数据包和应答数据包的包头结构是一致的。

[0109] 上述的源IP地址是指最后一个音频数据包的发送设备地址,本发明实施例的最后一个音频数据包的发送地址可以为信号采集编码端。上述的目的IP地址是指最后一个音频数据包的接收设备地址。上述的净载数据可以为流媒体音频数据,上述的标识部分可以是数据总数据量的标识,也可以是每个数据包的数据量的标识。

[0110] 最后一个音频数据包及最后一个音频数据包对应的时间戳也可以记录在本地硬盘中,也可以是记录在预先建立网络连接的网络虚拟磁盘中,还可以记录在临时缓存中。任何能够实现记录该音频数据包及该音频包对应的时间戳的位置,均属于本发明实施例的保护范围。

[0111] 将记录最后一个音频数据包内的时间戳作为最后一个音频数据包内的第二时间戳DE,该最后一个音频数据包内的第二时间戳DE包括:编码器按照预设采样率对该流媒体音频数据进行采样编码,并且在这个流媒体音频数据包中写入上述时长的结束时间戳。

[0112] 步骤204,计算该最后一个音频数据包内的第二时间戳与该第一个音频数据包内的第一时间戳之间差值的绝对值,获得所获取的流媒体音频数据的播放时长。

[0113] 利用所获取的流媒体音频数据的播放时长,进而获得流媒体数据的播放时长。

[0114] 这里计算出的该绝对值作为所获取的流媒体音频数据的播放时长,从而获得流媒体数据的播放时长。在流媒体数据的播放稳定时,也就是处于正常播放情况下,第一时间戳DS与第二时间戳DE之间的差值等于编码器在按照预设采样率对该流媒体音频数据进行采样编码时,在这个流媒体音频数据包中写入的时长,也就是,流媒体数据的播放时长。相较于目前常见的流媒体数据的视频帧率为30fps、25fps、20fps、15fps等,视频帧率越低,连续画面的平滑程度越差,但一般不低于每秒15帧,人眼都会认为是流畅的画面,且在一些应用场景下,也会有主动丢弃视频帧的编码方式或传输方式,对于观众来讲较低的视频帧率仅会带来不太平滑的画面效果。但是只要声音连续,并不会出现直播短暂中断等的卡顿现象及感觉,并且音频帧与视频帧不同,只要有音频帧丢失或异常,观众就能很容易察觉到人物对话或者音乐的停顿、慢放、不连续。

[0115] 应用本发明实施例,在接收到第一个音频数据包及最后一个音频数据包时,分别记录对应的第一时间戳和第二时间戳,能够快速利用两者之差值,确定出所获取的流媒体音频数据的播放时长。这里使用了最基本的对流媒体数据中的音频数据进行音频编码时间戳前后对比,利用音频编码的音频帧率及两者之差,在传输过程中的变化,作为流媒体直播过程中是否出现卡顿的重要判断参考指标,可以准确地得知音频帧的丢失,观众从而更加容易的察觉人物对话或者音乐的停顿、慢放、不连续,设备也方便进行检测卡顿。

[0116] 为了更加准确地判断该播放时长是否正确,该流媒体直播卡顿的判断方法,在上述步骤202之后,所述方法还可以包括:在接收到第一个音频数据包时,记录该第一个音频

数据包对应的操作系统时间戳SS (System_Time_Start, 系统开始时间), 以及

[0117] 在上述步骤203之后, 所述方法还可以包括: 在接收到最后一个音频数据包时, 记录该最后一个音频数据包对应的操作系统时间戳SE (System_Time_End, 系统结束时间);

[0118] 利用最后一个音频数据包对应的操作系统时间戳SE减去第一个音频数据包对应的操作系统时间戳SS所得到的操作系统时间戳差值, 在该操作系统时间戳差值的大小与上述DE-DS差值的绝对值的大小相同时, 则表明该流媒体音频数据包的传输稳定, 且判断为该播放时长正确。说明流媒体音频数据在传输过程中, 没有中断或者延时传输。

[0119] 由于在流媒体播放过程中, 可能出现流媒体直播卡顿的情况, 也可能出现流媒体直播未卡顿的情况, 为了能够准确地判断流媒体直播是否出现卡顿的情况。可选的, 在一种具体实现方式中, 步骤103可以包括: 在该比较结果表明计算出的播放时长等于该预设时间长度时, 判定流媒体直播未卡顿。

[0120] 这里的比较结果表明计算出的播放时长等于该预设时间长度, 可知该预设时间长度, 设备正确接收到了等时长的流媒体音频数据, 即流媒体直播过程中未出现流媒体音频数据丢失的问题, 此时数据缓冲区内缓存的数据量为该预设时间长度流媒体音频数据对应的数据量, 则判定流媒体直播未卡顿。

[0121] 示例性的, 假设获取的流媒体音频数据中第一个音频数据包内的第一时间戳DS为2017年5月26日星期一14点01分07秒, 最后一个音频数据包内的第二时间戳DE为2017年5月26日星期一14点01分09秒, 利用最后一个音频数据包内的第二时间戳DE减去第一个音频数据包内的第一时间戳DS得到差值的绝对值为2秒, 将该2秒作为上述播放时长。

[0122] 假设预设时间长度为2秒, 设备接收到第一个音频数据包内的系统时间戳为2017年5月26日星期一14点01分08秒, 正好在2秒时的系统时间戳为2017年5月26日星期一14点01分10秒, 设备接收到最后一个音频数据包SE, 也就是, 利用最后一个音频数据包内的系统时间戳减去第一个音频数据包内的系统时间戳, 得到差值的绝对值为2秒, 将该2秒作为上述预设时间长度, 这样预设时间长度与播放时长相等, 说明判定流媒体直播未卡顿, 流媒体音视频数据在传输过程中, 没有中断或者延时传输。

[0123] 相较于目前的视频编码技术由编码压缩输出的视频质量较差的CBR (Constants Bit Rate, 固定码率) 技术, 后来变为编码压缩输出的视频质量更优的VBR (Variable Bit Rate, 动态比特率) 技术, 即单位时间内的视频比特率会随着视频内容本身的变化频繁程度及色彩等信息的复杂程度而上下浮动变化。虽然总体平均比特率基本恒定, 但带来的直接问题就是单位时间内视频比特率数值波动非常大, 最高可能出现2、3倍以上的变化范围。此时, 将根据用户需求进行设置的第二预设阈值作为参考值, 并且该第二预设阈值与实际视频比特率比较, 可能存在较大偏差, 由于单位时间内视频比特率数值上下浮动变化很大, 这样不仅很难设置一个较为准确的第二预设阈值, 来作为参考值, 实现对整个流媒体直播过程的视频比特率进行对比分析, 而且由于网络主动丢弃的视频, 并不是实际的卡顿, 该动态比特率算法并不能识别出来该情况, 可能会将该网络主动丢弃的视频, 判定流媒体直播出现卡顿。

[0124] 应用本实施例, 利用流媒体音频数据的播放时长等于该预设时间长度时, 判定流媒体直播未卡顿, 并不需要识别视频帧, 也不需要设置第二预设预置作为参考值, 来对整个直播过程的视频比特率进行对比分析, 更不会出现将网络主动丢弃的视频, 错误判断为流

媒体直播的卡顿的情况,提高了判定流媒体直播未卡顿的准确性。

[0125] 为了能够准确地判断流媒体直播未出现卡顿,在一种具体实现方式中,步骤103可以包括如下步骤:

[0126] 在该比较结果表明计算出的播放时长小于或大于该预设时间长度时,判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿,其中,该预设条件表明流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

[0127] 在该比较结果表明计算出的播放时长小于该预设时间长度时,可知设备在还未到预设时间长度时,已经接收到完成了流媒体音频数据。表明理论和实际使用同样的预设时间长度,设备接收到了比理论值要少的音频帧数据。

[0128] 示例性的,假设获取的流媒体音频数据中第一个音频数据包内的第一时间戳DS为2017年5月26日星期一14点02分07秒,最后一个音频数据包内的第二时间戳DE为2017年5月26日星期一14点02分08秒,利用最后一个音频数据包内的第二时间戳DE减去第一个音频数据包内的第一时间戳DS得到差值的绝对值为1秒,将该1秒作为上述播放时长。

[0129] 假设预设时间长度为2秒,设备接收到第一个音频数据包内的系统时间戳为2017年5月26日星期一14点02分08秒,在系统时间戳为2017年5月26日星期一14点02分09秒,也就是在1秒内设备接收到最后一个音频数据包SE,而预设时间长度为2秒,在1秒内已经接收完音频数据包,在剩下的1秒内并没有接收到音频数据包,即,系统时间戳为2017年5月26日星期一14点02分10秒,设备接收的音频数据包为零。

[0130] 设备通常会设定一个数据缓冲区,数据缓冲区的数据量通常包括:数据缓冲区所能容纳的数据量及数据缓冲区内缓存的数据量,该数据缓冲区所能容纳的数据量可以定义为CL(Cache_Length,数据缓冲区所能容纳的数据量),用来容纳定量的数据。示例性的,该数据缓冲区所能容纳2秒时长的数据量,则 $CL=TI*2$,其中,TI为预设时长。

[0131] 这里数据缓冲区所能容纳的数据量包括数据量的下限值,数据量的上限值,以及能够缓存数据处于上限值与下限值之间的数据合法范围。可选的,一种具体实现方式中,在数据缓冲区内缓存的数据量不足数据量的下限值时,由于流媒体直播需要读取数据进行播放,而此时由于数据缓冲区内的没有缓存数据或者不满足播放数据的数据量,且也没有接收到流媒体音频数据,此时就会判定流媒体直播出现卡顿。

[0132] 可选的,另一种具体实现方式中,在数据缓存区中缓存数据大于数据量的上限值时,由于数据缓存区的数据量得不到及时的播放,超出数据量的上限值的数据可能就会丢失,此时就会出现卡顿。因此可以使用数据缓存区的数据量准确地判断流媒体直播是否卡顿。

[0133] 具体的,示例性的,网络拥塞可能引起流媒体直播的卡顿。数据缓冲区所能容纳的数据量的上限值为3秒时间长度对应的数据量,由于网络拥塞,则可能此时设备接收到流媒体音频数据不足1秒时间长度对应的数据量,原本每次读取播放的2秒时间长度对应的数据量,而此时数据缓冲区内缓存的数据量就不足够,只能读取1秒时间长度对应的数据量,此时设备播放完成1秒时间长度对应的数据量后,流媒体播放就会出现空白,也就是,流媒体直播出现了卡顿。

[0134] 如果在网络拥塞过后,流媒体音频数据的数据量突然增大。比如,流媒体音频数据的数据量为5s时间长度对应的数据量,因为该5s时间长度对应的数据量大于3秒时间长度

对应的数据量,网络就会自行选择压缩,丢弃掉多余的2秒时间长度对应的数据量,因此就没有缓存该2秒时间长度对应的数据,设备就接收不到流媒体数据量,这样流媒体播放出现断点,也就是,流媒体直播出现了卡顿。

[0135] 等到网络恢复以后,流媒体音频数据的数据量按照每次正好等于或者小于原数据缓存区的数据缓冲区所能容纳的数据量的上限值进行播放,流媒体直播才不会出现卡顿。

[0136] 具体的,示例性的,在写代码或者编码器编码时,编码器写入数据包的时间戳出现错误。假设正常时间戳应该是根据解码频率进行设置的,每秒一个数据。比如,第1秒对应一个数据包1,第2秒对应一个数据包2,第3秒对应一个数据包3,第4秒对应一个数据包4,第5秒对应一个数据包5;

[0137] 假设设置的预设时间长度为5秒,也就是设备正好可以在5秒内接收这5个数据包。但是在传输过程出现错误时,第1秒对应一个数据包1,第3秒对应一个数据包2,第5秒对应一个数据包3,第7秒对应一个数据包4,第9秒对应一个数据包5。这样在后期实际传输时,传输数据的时间戳就会变长为7秒。因此,网络实际传输的时间长度就大于数据包正常时间戳传输的时间长度,那么到第5秒时,共发送了数据包1、数据包2及数据包3,但是数据包4和数据包5并未及时发出,这些在正常时间戳传输的时间长度内未及时发出的数据包4和数据包5,在网络恢复后进行追赶重发,那么解码端无法在5秒内解码未及时发出去的数据包4和数据包5,使得流媒体直播会出现卡顿。

[0138] 该预设条件表明流媒体音频数据的数据量与数据缓冲区所能容纳的数据量的比较关系是在数据缓冲区所能容纳的数据量的基础上,判断流媒体音频数据的数据量是否超出数据缓冲区所能容纳的数据量的上限值,也就是,是否存在数据缓冲区所能容纳的数据量的数据溢出的关系;利用这个关系,在该比较结果表明计算出的播放时长小于或大于该预设时间长度时,可以判断在数据缓冲区所能容纳的数据量的数据溢出,判定流媒体直播出现卡顿。或者,

[0139] 判断流媒体音频数据的数据量是否达到数据缓冲区所能容纳的数据量的下限值,是否存在数据缓冲区内没有数据的关系。利用这个关系,在该比较结果表明计算出的播放时长小于或大于该预设时间长度时,且在下限值为零时,在数据缓冲区缓存的数据量为零时,判定流媒体直播出现卡顿。

[0140] 应用本发明实施例,通过数据缓冲区所能容纳的数据量是否数据溢出及流媒体音频数据的数据量的比较关系,或者,通过数据缓冲区缓存的数据量是否为零及流媒体音频数据的数据量的比较关系,来可以准确地判断流媒体直播是否卡顿。

[0141] 可选的,在一种具体实现方式中,在比较结果表明计算出的播放时长大于预设时间长度时,上述判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:

[0142] 判断所述预设时间长度的所获取的流媒体音频数据的数据量是否大于数据缓冲区所能容纳的数据量。

[0143] 播放时长大于该预设时间长度时,可知设备在预设时间长度达到后,依然未接收到完成了流媒体音频数据,可能会造成流媒体直播过程中出现流媒体音频数据追赶或者流媒体数据重发的问题。

[0144] 示例性的,假设获取的流媒体音频数据中第一个音频数据包内的第一时间戳DS为

2017年5月26日星期一14点02分09秒,最后一个音频数据包内的第二时间戳DE为2017年5月26日星期一14点02分12秒,利用最后一个音频数据包内的第二时间戳DE减去第一个音频数据包内的第一时间戳DS得到差值的绝对值为3秒,将该3秒作为上述播放时长。

[0145] 假设预设时间长度为2秒,设备接收到第一个音频数据包内的系统时间戳为2017年5月26日星期一14点02分08秒,在系统时间戳为2017年5月26日星期一14点02分09秒及系统时间戳为2017年5月26日星期一14点02分10秒,设备依然接收到最后一个音频数据包与第一个音频数据包之间的音频数据包,设备没有接收到最后一个音频数据包,可能是网络的流媒体数据过多,设备没有及时接收到最后一个音频数据包,也可能是网络就会自行选择压缩,丢弃掉多余的1秒时间长度的最后一个音频数据包,这样流媒体数据就会重发该最后一个音频数据包。

[0146] 在比较结果表明计算出的播放时长大于预设时间长度时,这时候需要知道预设条件所表明流媒体音频数据的数据量与数据缓冲区所能容纳的数据量的比较关系,设备才能够准确地进行流媒体直播是否卡顿的判断。

[0147] 这里的比较结果表明计算出的播放时长大于预设时间长度,设备接收到了比理论值还要多的流媒体音频数据,可能在此之前出现过卡顿,这些多于理论值的流媒体音频数据是之前网络拥塞未及时发出的数据,而等到网络拥塞恢复正常以后,该多于理论值的流媒体音频数据追赶重发,故出现播放时长大于预设时间长度的情况。

[0148] 这里的是否满足预设条件,说明在多于理论值的流媒体音频数据追赶重发时,需要判断所追赶重发的流媒体音频数据是否处于数据缓冲区所能容纳的数据量内,从而利用该预设条件,能够准确地判断流媒体直播是否出现卡顿。

[0149] 若所述预设时间长度的所获取的流媒体音频数据的数据量大于数据缓冲区所能容纳的数据量,判定流媒体直播卡顿。

[0150] 这里的所获取的流媒体音频数据的数据量大于数据缓冲区所能容纳的数据量,可知流媒体音频数据已经超出了数据缓冲区所能容纳的数据量,会出现媒体音频数据溢出数据缓冲区的情况,并出现溢出的数据被网络丢弃的问题,此时判定流媒体直播卡顿。

[0151] 若所述预设时间长度的所获取的流媒体音频数据的数据量小于或等于数据缓冲区所能容纳的数据量,判定流媒体直播未卡顿。

[0152] 这里的所获取的流媒体音频数据的数据量小于或等于数据缓冲区所能容纳的数据量,可知流媒体音频数据的数据量处于数据缓冲区所能容纳的数据量合法范围内,既未在传输过程中丢失,也未超出缓冲区而被丢弃,判定流媒体直播未卡顿。

[0153] 这里的数据缓冲区所能容纳的数据量包括数据缓冲区的数据量的上限值,或者根据用户需求设置的数据缓冲区所能容纳的数据量。该数据缓冲区所能容纳的数据量可以是根据用户长期试验,得到一个预设数值,将该预设数值作为数据缓冲区所能容纳的数据量,这样就会在流媒体音频数据的数据量大于该预设数值时,判定出现流媒体直播卡顿,更加精细地判定流媒体直播卡顿的准确性。

[0154] 为了能够减轻数据传输对网络造成的压力,可选的,在一种具体实现方式中,若判定流媒体直播卡顿时,该流媒体直播卡顿的判断方法还包括:确定溢出所述数据缓冲区的数据量,将所溢出的数据量删除。

[0155] 相较于目前按照动态画面或者静态画面,进行视频编码帧率的选择,即,除视频编

码帧率也有恒定及动态之分,视频编码帧率可以是VFR(Variable Frame Rate,可变帧率)编码算法,与VBR类似。随着画面动态变化的程度不同,编码器会编码产生不同的帧率,如动态画面下采用较高的视频帧率,静态画面下采用较低的视频帧率,这样对于节省视频编码的文件尺寸有较大帮助,但如果将按照用户需求进行设置一个第二预设阈值作为比较参数,与视频帧率比较,可能会存在小于该第二预设阈值的视频帧率。这个小于该第二预设阈值的视频帧率,会被判断为出现了流媒体直播卡顿。但实际情况是,该小于该第二预设阈值的视频帧率,只是说明出现了较多的静态画面,并不一定会出现流媒体直播卡顿。这样选择的视频编码帧率,并不能识别出来该情况,可能会判定流媒体直播出现卡顿。

[0156] 应用本实施例,在比较结果表明计算出的播放时长大于预设时间长度时,利用流媒体音频数据的数据量与数据缓冲区所能容纳的数据量的比较关系,判断流媒体直播是否卡顿,提高了判断流媒体直播是否卡顿的准确性,并不需要通过画面动态的变化设置比较参数,减小了设置合适的比较参数的难度。

[0157] 为了能够准确地检测流媒体直播是否出现卡顿,在一种具体实现方式中,在比较结果表明计算出的播放时长小于预设时间长度时,判断所述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿的步骤包括:

[0158] 判断所述数据缓冲区内缓存的数据量是否为零,且所获取的流媒体音频数据的数据量是否为零。

[0159] 在比较结果表明计算出的播放时长小于预设时间长度时,这时候需要知道预设条件所表明流媒体音频数据的数据量与数据缓冲区缓存的数据量的比较关系,设备才能够准确地进行流媒体直播是否卡顿的判断。

[0160] 若判断数据缓冲区内缓存的数据量为零,且所获取的流媒体音频数据的数据量为零,判定流媒体直播卡顿。

[0161] 若判断数据缓冲区内缓存的数据量大于零,且所获取的流媒体音频数据的数据量大于零,判定流媒体直播未卡顿。

[0162] 这里的比较结果表明计算出的播放时长小于预设时间长度,可知设备接收到了比理论值还要少的流媒体音频数据,需要再判断数据缓冲区内缓存的数据量是否为零,在数据缓冲区缓存的数据量不为空(即,数据缓冲区内缓存的数据量不为零),说明数据缓冲区会有流媒体音频数据,并不影响解码端解码该流媒体音频数据,因此判定流媒体直播未卡顿;

[0163] 在数据缓冲区的缓存的数据量为空,并且在未接收到流媒体音频数据(即,所获取的流媒体音频数据的数据量为零),因此判定流媒体直播卡顿。

[0164] 与现有技术中利用流媒体数据中的视频数据判断流媒体直播是否卡顿相比,应用本发明实施例,依据本发明实施例的流媒体音频数据,判定流媒体直播卡顿,很好地规避了这种现有的判断所带来的直接干扰的问题,便可以实现更准确地判定流媒体直播的卡顿。

[0165] 如图3所示,应用在流媒体直播过程中接收流媒体的设备可以为流媒体数据接收端。本发明实施例的流媒体直播过程的流程如下:

[0166] 步骤301,流媒体直播开始;

[0167] 步骤302,信号采集编码端采集流媒体数据中的流媒体音频数据;

[0168] 步骤303,信号采集编码端编码压缩流媒体中的流媒体音频数据。该有流媒体音频

数据的流媒体数据中具有时长,其中,该时长包括起始时间戳和结束时间戳,也就是,将第一个音频数据包的第一时间戳DS作为起始时间戳,以及最后一个音频数据包的第二时间戳DE作为结束时间戳;

[0169] 步骤304,信号采集编码端发送编码压缩后的流媒体数据;

[0170] 步骤305,流媒体数据接收端接收编码压缩后的流媒体数据,设置预设时间长度TI及数据缓冲区所能容纳的数据量CL;

[0171] 步骤306,流媒体数据接收端记录第一个音频数据包内的第一时间戳DS;

[0172] 流媒体数据中的流媒体音频数据是通过音频数据包进行传输的。

[0173] 步骤307,流媒体数据接收端记录最后一个音频数据包内的第二时间戳DE及记录音频数据包的总数据量DL;

[0174] 上述音频数据包的总数据量DL可以是记录每个音频数据包的数据量,将所有音频数据包的数据量相加得到的,其中,所有数据包包括:第一个音频数据包、最后一个音频数据包及该第一个音频数据包与该最后一个音频数据包之间对应的所有中间音频包。

[0175] 步骤308,流媒体数据接收端计算 $|DE-DS|=DD$,其中, $|DE-DS|$ 可以是指第二时间戳DE减去第一时间戳DS之差的绝对值,或者第一时间戳DS减去第二时间戳DE之差的绝对值,并判断差值的绝对值DD、预设时间长度TI、数据缓冲区所能容纳的数据量CL、音频数据包的总数据量DL的关系。

[0176] 这里的差值的绝对值DD作为所获取的流媒体音频数据的播放时长。

[0177] 步骤309,流媒体数据接收端判断差值的绝对值DD是否等于预设时间长度TI;如果是,说明 $DD=TI$ 时,则执行步骤310;如果不是,且 $DD<TI$ 时,则执行步骤311,或者,如果不是,且 $DD>TI$ 时,则执行步骤313。

[0178] 步骤310,判定流媒体直播无卡顿,直至执行步骤314。

[0179] 步骤311,判断音频数据包的总数据量DL是否大于零;若是,说明 $DL=0$,执行步骤310;若否,说明 $DL<0$,执行步骤312。

[0180] 步骤312,判定流媒体直播卡顿,执行步骤314。

[0181] 步骤313,判断音频数据包的总数据量DL是否小于或等于数据缓冲区所能容纳的数据量CL,如果是,说明 $DL<CL$,则执行步骤310;如果不是,说明 $DL>CL$ 时,数据缓冲区有溢出,则执行步骤312。

[0182] 步骤314,流媒体直播结束。

[0183] 流媒体数据从信号采集编码端,会经过许多环节传输到流媒体数据接收端,由于流媒体数据具有实时传输的特性,对于网络质量有非常高的要求,会经常遇到流媒体直播卡顿等类似的影响观看体验的问题。目前基于现有的视频编码的技术无法准确地判定流媒体直播卡顿,而采用基于本发明实施例的音频编码技术,可以准确地判断出流媒体直播是否卡顿,恰好地避开上述视频编码的技术所带来问题。所以利用本发明实施例的流媒体直播卡顿的判断方法,通过流媒体数据中的流媒体音频数据对流媒体直播卡顿情况进行检验和分析是非常必要的。

[0184] 本发明实施例提供的流媒体直播卡顿的判断方法,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除

了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。

[0185] 如图4所示,本发明实施例提供了一种流媒体直播卡顿的判断装置,应用在流媒体直播过程中接收流媒体的设备,上述装置包括:

[0186] 获取模块401,用于获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

[0187] 比较获得模块402,用于将计算出的播放时长与上述预设时间长度进行比较,获得一比较结果;

[0188] 判断模块403,用于根据上述比较结果,判断流媒体直播是否卡顿。

[0189] 本发明实施例提供的流媒体直播卡顿的判断装置,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。

[0190] 可选的,第一获取模块,用于获取上述预设时间长度的流媒体数据中的流媒体音频数据;

[0191] 第一记录模块,用于记录上述预设时间长度的第一个音频数据包内的第一时间戳;

[0192] 第二记录模块,用于记录上述预设时间长度的最后一个音频数据包内的第二时间戳;

[0193] 计算获得模块,用于计算上述最后一个音频数据包内的第二时间戳与上述第一个音频数据包内的第一时间戳之间差值的绝对值,获得所获取的流媒体音频数据的播放时长。

[0194] 可选的,第一判断子模块,用于在上述比较结果表明计算出的播放时长等于上述预设时间长度时,判定流媒体直播未卡顿。

[0195] 可选的,第二判断子模块,用于在上述比较结果表明计算出的播放时长小于或大于上述预设时间长度时,判断上述流媒体音频数据的数据量是否满足预设条件,判断流媒体直播是否卡顿,其中,上述预设条件表明上述流媒体音频数据的数据量与数据缓冲区的数据量的比较关系。

[0196] 可选的,上述第二判断子模块具体用于:在上述比较结果表明计算出的播放时长大于上述预设时间长度时,判断上述预设时间长度的所获取的流媒体音频数据的数据量是否大于上述数据缓冲区所能容纳的数据量,

[0197] 若上述预设时间长度的所获取的流媒体音频数据的数据量大于上述数据缓冲区所能容纳的数据量,判定流媒体直播卡顿;

[0198] 若上述预设时间长度的所获取的流媒体音频数据的数据量小于或等于上述数据缓冲区所能容纳的数据量,判定流媒体直播未卡顿。

[0199] 可选的,删除模块,用于确定溢出上述数据缓冲区的数据量,将所溢出的数据量删除。

[0200] 可选的,上述第二判断子模块具体用于:在上述比较结果表明计算出的播放时长

小于上述预设时间长度时,判断上述数据缓冲区内缓存的数据量是否为零,且所获取的流媒体音频数据的数据量是否为零,

[0201] 若上述数据缓冲区内缓存的数据量为零,且所获取的流媒体音频数据的数据量为零,判定流媒体直播卡顿;

[0202] 若上述数据缓冲区内缓存的数据量大于零,且所获取的流媒体音频数据的数据量大于零,判定流媒体直播未卡顿。

[0203] 本发明实施例还提供了一种在流媒体直播过程中接收流媒体的设备,包括处理器和存储器,存储器,用于存放计算机程序;

[0204] 处理器,用于执行存储器上所存放的程序时,实现如下步骤:

[0205] 获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

[0206] 将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;

[0207] 根据所述比较结果,判断流媒体直播是否卡顿。

[0208] 如图5所示,本发明实施例还提供了一种在流媒体直播过程中接收流媒体的设备,包括处理器501、通信接口502、存储器503和通信总线504,其中,处理器,通信接口,存储器通过总线完成相互间的通信,

[0209] 存储器503,用于存放计算机程序;

[0210] 处理器501,用于执行存储器上所存放的程序时,实现如下步骤:

[0211] 获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

[0212] 将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;

[0213] 根据所述比较结果,判断流媒体直播是否卡顿。

[0214] 上述电子设备提到的通信总线可以是外设部件互连标准(Peripheral Component Interconnect,PCI)总线或扩展工业标准结构(Extended Industry Standard Architecture,EISA)总线等。该通信总线可以分为地址总线、数据总线、控制总线等。为便于表示,图中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0215] 通信接口用于上述电子设备与其他设备之间的通信。

[0216] 存储器可以包括随机存取存储器(Random Access Memory,RAM),也可以包括非易失性存储器(Non-Volatile Memory,NVM),例如至少一个磁盘存储器。可选的,存储器还可以是至少一个位于远离前述处理器的存储装置。

[0217] 上述的处理器可以是通用处理器,包括中央处理器(Central Processing Unit,CPU)、网络处理器(Network Processor,NP)等;还可以是数字信号处理器(Digital Signal Processing,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。

[0218] 本发明实施例提供的在流媒体直播过程中接收流媒体的设备,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出

现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。

[0219] 本发明实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质内存储有计算机程序,所述计算机程序被处理器执行时实现如下步骤:

[0220] 获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长;

[0221] 将计算出的播放时长与所述预设时间长度进行比较,获得一比较结果;

[0222] 根据所述比较结果,判断流媒体直播是否卡顿。

[0223] 本发明实施例提供的在流媒体直播过程中接收流媒体的设备,通过获取预设时间长度的流媒体数据中的流媒体音频数据,计算所获取的流媒体音频数据的播放时长,比较流媒体音频数据播放时长与预设时间长度得到一比较结果,从而判断流媒体直播是否卡顿。这样消除了利用流媒体中的视频数据检测卡顿的误判,可以准确地判断流媒体是否出现卡顿。为后续调整视频流畅度作基础,在后期有效的调整视频流畅度后,提高了用户体验效果。

[0224] 在流媒体直播过程中最常出现的影响观众观看体验的问题就是卡顿,但卡顿是发生在业务层的一种人类可以感知到的问题,而对于为流媒体数据传输提供基础支撑的网络、服务器及应用程序来讲,只能感知到流媒体数据在传输过程中的速度、完整性等指标,如果能将传输层面的质量指标与业务层面的质量指标进行准确的关联,即应用程序可以通过分析流媒体数据传输的质量变化来判断是否可能出现卡顿,将非常有助于提高流媒体直播的服务质量。

[0225] 因此,本发明实施例提供一种流媒体直播卡顿的判断方法中,应用在目前复杂的互联网流媒体直播架构中,为多级级联的直播全链路卡顿排查分析,提供了重要且切实可行的判断方法,准备好为后期进一步提升流媒体直播服务质量。

[0226] 上述的多级级联用于流媒体数据从信号采集编码端,经过多个相互连接的节点,达到流媒体数据接收端。流媒体数据接收端对流媒体直播数据进行解码。比如,O,P,Q,R这四个转发节点,每两个相邻的节点中的一个可以作为发送端,另外一个可以作为接收端。具体的,当在转发节点O作为发送端时,转发节点P作为接收端;在转发节点P作为发送端时,转发节点Q作为接收端;在转发节点Q作为发送端时,转发节点R作为接收端,这样每一个发送端到相邻的接收端称为一个环节或者一个级别,那么就存在多个环节,多个级别,该多个级别就称为多级级联。

[0227] 示例性的,北京地区的设备发送流媒体数据给西安地区的设备,北京地区的设备与西安地区的设备未建立直接连接,而是通过分发网络,北京地区的设备发送流媒体数据至,一个与陕西地区的设备和北京地区的设备均建立的中转节点设备,然后将中转节点设备转发的流媒体数据再发送给西安地区的设备,这样不仅避免了北京地区的设备与西安地区的设备建立直接连接所形成的链路存在不稳定的问题,可以提高传输的可靠性,而且可以利用本发明实施例的流媒体直播卡顿的判断方法,排查了流媒体数据在多级级联传输过程中的卡顿。

[0228] 本发明实施例提供了一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述流媒体直播卡顿的判断方法的步骤。

[0229] 本发明实施例提供了一种计算机程序,当其在计算机上运行时,使得计算机执行上述流媒体直播卡顿的判断方法的步骤。

[0230] 对于装置/设备/存储介质实施例/包含指令的计算机程序产品/计算机程序而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0231] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0232] 本说明书中的各个实施例均采用相关的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于装置/设备/存储介质实施例/包含指令的计算机程序产品/计算机程序实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0233] 以上所述仅为本发明的较佳实施例而已,并非用于限定本发明的保护范围。凡在本发明的精神和原则之内所作的任何修改、等同替换、改进等,均包含在本发明的保护范围内。

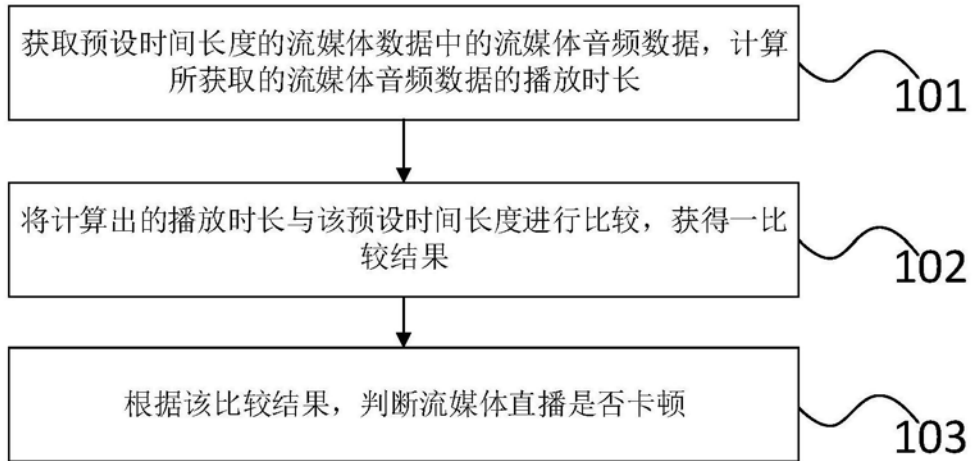


图1

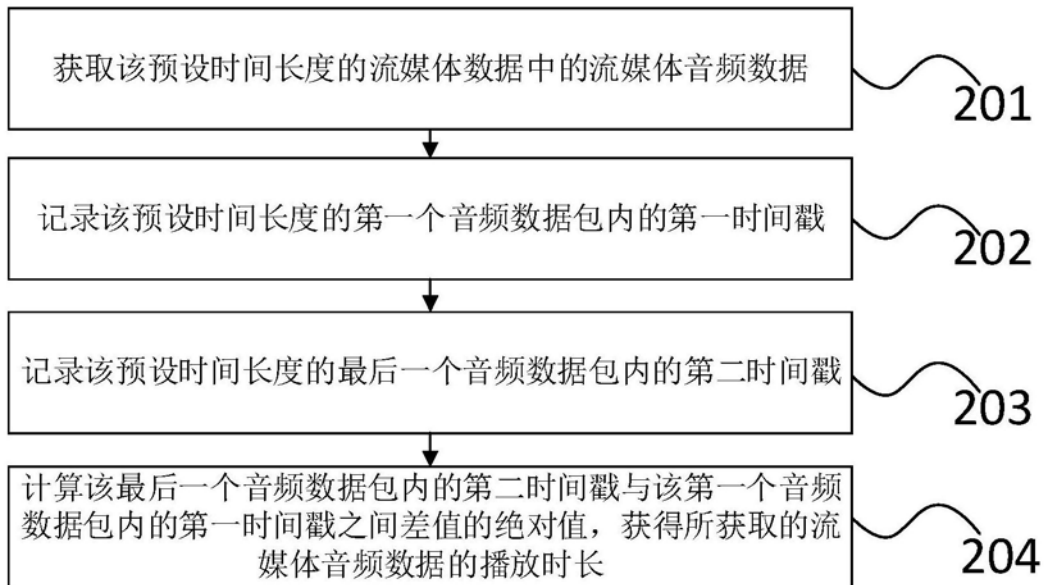


图2

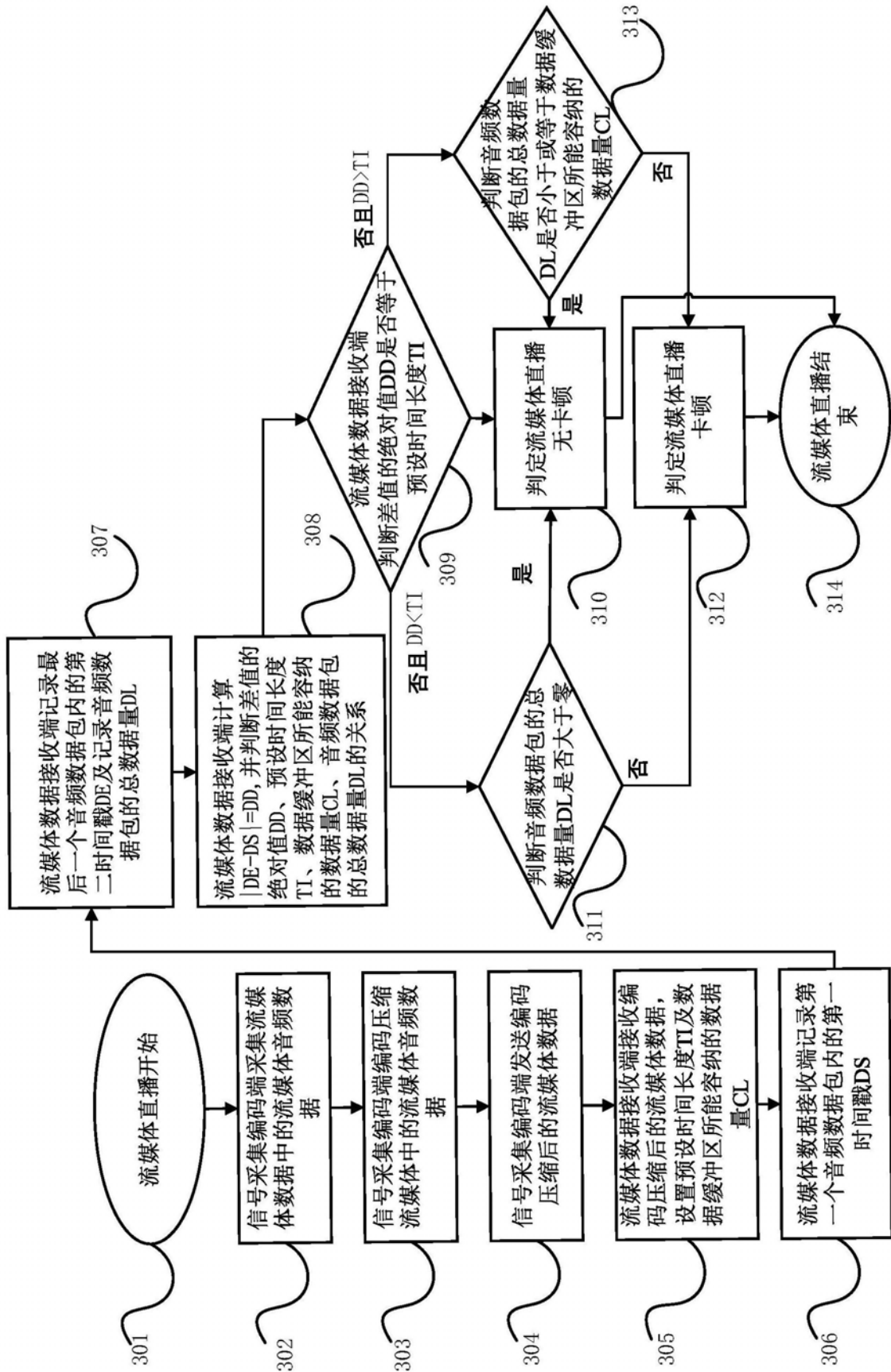


图3

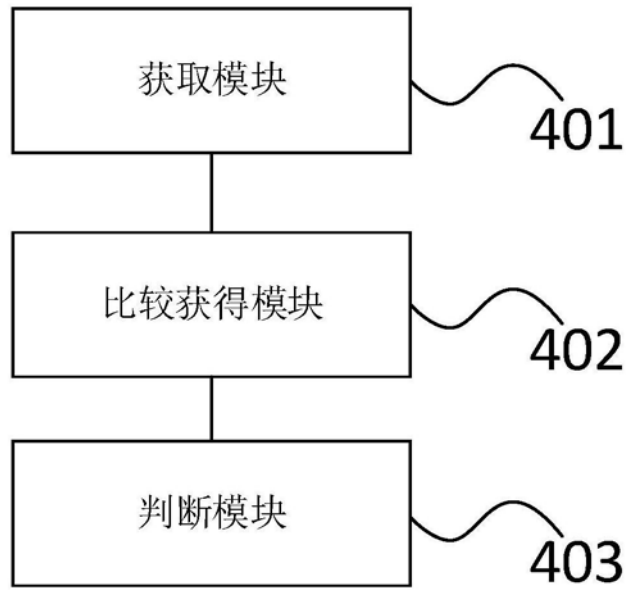


图4

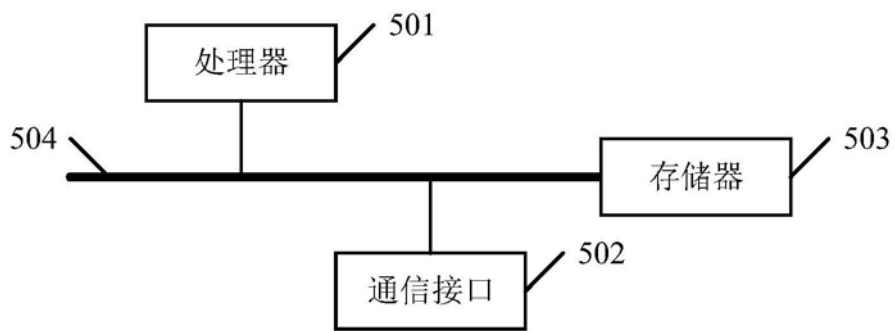


图5