



(12) 发明专利申请

(10) 申请公布号 CN 105393516 A

(43) 申请公布日 2016. 03. 09

(21) 申请号 201480040560. 8

(74) 专利代理机构 北京东方亿思知识产权代理
有限责任公司 11258

(22) 申请日 2014. 07. 15

代理人 李晓冬

(30) 优先权数据

13/943, 445 2013. 07. 16 US

(51) Int. Cl.

H04L 29/06(2006. 01)

(85) PCT国际申请进入国家阶段日

2016. 01. 15

(86) PCT国际申请的申请数据

PCT/US2014/046633 2014. 07. 15

(87) PCT国际申请的公布数据

W02015/009684 EN 2015. 01. 22

(71) 申请人 思科技术公司

地址 美国加利福尼亚州

(72) 发明人 李智 艾丽·C·贝根

乔舒亚·B·盖哈姆

布鲁斯·P·奥斯勒 大卫·R·奥兰

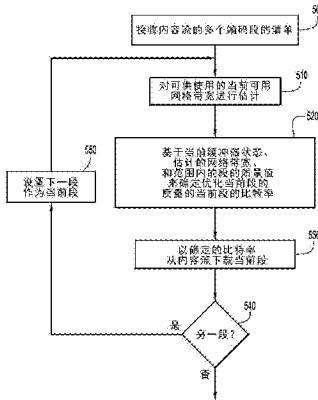
权利要求书3页 说明书14页 附图6页

(54) 发明名称

在自适应流送中用缓冲器和范围约束来进行
质量优化

(57) 摘要

提供了用于在自适应流送中优化质量的技术，其中，关于内容流的内容段的编码比特率和内容质量评分的信息被获得。内容流的内容段由内容服务器提供并且由计算设备下载。针对每一内容段，编码比特率被确定以用于从内容服务器下载内容段，其中，确定当前内容段的编码比特率是基于估计的当前网络带宽、下载的内容段被存储到其中的缓冲器的约束、并且还基于与选定范围内的内容段相关联的质量评分，该选定范围从当前内容段延伸到在范围端点处的内容流中的将来内容段）。每一个内容段以针对该内容段确定的比特率被从内容服务器下载。



1. 一种方法，包括：

获得关于内容流的内容段的编码比特率和内容质量评分的信息，其中，所述内容流的内容段由内容服务器提供并且由计算设备下载；

针对每一内容段，确定所述内容段将被从所述内容服务器下载的编码比特率，其中，确定当前内容段的编码比特率是基于以下各项的：估计的当前网络带宽、下载的内容段被存储到其中的缓冲器的约束、以及与选定范围内的内容段相关联的质量评分，所述选定范围从所述当前内容段延伸到在所述范围端点处的所述内容流中的将来内容段；以及

以针对每一内容段确定的比特率来从所述内容服务器下载所述内容段。

2. 如权利要求 1 所述的方法，其中，所述确定所述当前内容段的编码比特率还包括：

应用动态编程算法，所述动态编程算法使用与所述选定范围内的内容段相关联的质量评分和缓冲器约束信息来确定所述当前内容段的最优质量评分值；以及

应用在线算法，所述在线算法迭代地计算所述选定范围内的段的编码比特率，并且确定与所述当前内容段的最优质量评分值相关联的编码比特率。

3. 如权利要求 2 所述的方法，其中，所述应用动态编程算法包括将与所述选定范围内的内容段相关联的所述质量评分的平均评分最大化。

4. 如权利要求 2 所述的方法，其中，所述应用动态编程算法包括将与所述选定范围内的内容段相关联的所述质量评分的最差情况的质量评分最大化。

5. 如权利要求 2 所述的方法，其中，所述应用动态编程算法还包括基于所述当前内容段的最优质量评分值和先前内容段的最优质量评分值之间的差值来将所述当前内容段的最优质量评分值调整为经修改的质量评分值。

6. 如权利要求 2 所述的方法，其中，所述应用动态编程算法还包括通过幂函数调整所述当前内容段的最优质量评分值，其中，所述当前内容段的最优质量评分值由指数 b 修改，其中 $b \geq 1$ 。

7. 如权利要求 2 所述的方法，其中，与当前内容段相关联的所述缓冲器约束信息包括：缓冲器下限值、缓冲器上限值、与所述当前内容段之前的内容段相关联的缓冲器大小、和与所述计算设备相关联的网络带宽值。

8. 如权利要求 2 所述的方法，其中，关于内容质量评分的所述信息还包括多个编码比特率、多个段持续时间、和与每一内容段的编码比特率相对应的多个内容质量评分。

9. 如权利要求 1 所述的方法，其中，关于所述内容流的内容段的编码比特率和内容质量评分的所述信息从由所述内容服务器提供的清单获得的。

10. 一种装置，包括：

包括质量优化引擎的一个或多个存储器，所述质量优化引擎包括一个或多个应用以及缓冲器，其中所述一个或多个应用对从内容服务器下载的内容流的内容段的质量进行优化，所述缓冲器存储所下载的所述内容流的内容段；

通信接口设备，所述通信接口设备被配置为使能网络上的通信；和

处理器，所述处理器被配置为执行和控制所述质量优化引擎的操作，以执行以下操作：

获得关于所述内容流的内容段的比特率和内容质量评分的信息，其中所述内容流的内容段由内容服务器来提供以供计算设备使用；

针对每一内容段,确定所述内容段将从所述内容服务器被下载的编码比特率,其中确定当前内容段的所述编码比特率是基于以下各项的:估计的当前网络带宽、下载的内容段被存储到其中的缓冲器的约束、以及与选定范围内的内容段相关联的质量评分,所述选定范围从所述当前内容段延伸到在所述范围端点处的所述内容流中的将来内容段;以及以针对每一内容段确定的比特率来从所述内容服务器下载所述内容段。

11. 如权利要求 10 所述的装置,其中,所述处理器被配置为通过以下方式来确定所述当前内容段的编码比特率:

应用动态编程算法,所述动态编程算法使用与所述选定范围内的内容段相关联的质量评分和缓冲器约束信息来确定所述当前内容段的最优质量评分值;以及

应用在线算法,所述在线算法迭代地计算所述选定范围内的段的编码比特率,并且确定与所述当前内容段的最优质量评分值相关联的编码比特率。

12. 如权利要求 11 所述的装置,其中,所述处理器被配置为通过将与所述选定范围内的内容段相关联的所述质量评分的平均评分最大化来应用所述动态编程算法。

13. 如权利要求 11 所述的装置,其中,所述处理器被配置为通过将与所述选定范围内的内容段相关联的所述质量评分的最差情况的质量评分最大化来应用所述动态编程算法。

14. 如权利要求 11 所述的装置,其中,与当前内容段相关联的所述缓冲器约束信息包括:缓冲器下限值、缓冲器上限值、与所述当前内容段之前的内容段相关联的缓冲器大小、和与所述计算设备相关联的网络带宽值。

15. 如权利要求 11 所述的装置,其中,关于内容质量评分的所述信息还包括多个编码比特率、和于每一内容段的编码比特率相对应的多个内容质量评分。

16. 如权利要求 10 所述的装置,其中,所述处理器被配置为从由所述内容服务器提供的清单获得关于所述内容流的内容段的编码比特率和内容质量评分的信息。

17. 一种或多种计算机可读存储介质,所述计算机可读存储介质编码有包括计算机可执行指令的软件,并且当所述软件被执行时能操作来:

获得关于内容流的内容段的编码比特率和内容质量评分的信息,其中,所述内容流的内容段由内容服务器提供并且由计算设备下载;

针对每一内容段,确定所述内容段将被从所述内容服务器下载的编码比特率,其中,确定当前内容段的编码比特率是基于以下各项的:估计的当前网络带宽、下载的内容段被存储到其中的缓冲器的约束、以及与选定范围内的内容段相关联的质量评分,所述选定范围从所述当前内容段延伸到在所述范围端点处的所述内容流中的将来内容段;以及

以针对每一内容段确定的比特率来从所述内容服务器下载所述内容段。

18. 如权利要求 17 所述的计算机可读存储介质,其中,所述指令能操作来通过以下方式确定所述当前内容段的编码比特率:

应用动态编程算法,所述动态编程算法使用与所述选定范围内的内容段相关联的质量评分和缓冲器约束信息来确定所述当前内容段的最优质量评分值;以及

应用在线算法,所述在线算法迭代地计算所述选定范围内的段的编码比特率,并且确定与所述当前内容段的最优质量评分值相关联的编码比特率。

19. 如权利要求 18 所述的计算机可读存储介质,其中,所述指令能操作来应用所述动态编程算法,包括将与所述选定范围内的内容段相关联的所述质量评分的平均评分最大

化。

20. 如权利要求 18 所述的计算机可读存储介质, 其中, 所述指令能操作来应用所述动态编程算法, 包括将与所述选定范围内的内容段相关联的所述质量评分的最差情况的质量评分最大化。

21. 如权利要求 18 所述的计算机可读存储介质, 其中, 与当前内容段相关联的所述缓冲器约束信息包括: 缓冲器下限值、缓冲器上限值、与所述当前内容段之前的内容段相关联的缓冲器大小、和与所述计算设备相关联的网络带宽值。

22. 如权利要求 18 所述的计算机可读存储介质, 其中, 关于内容质量评分的所述信息还包括多个编码比特率、和与每一内容段的编码比特率相对应的多个内容质量评分。

23. 如权利要求 17 所述的计算机可读存储介质, 其中, 所述指令能操作来从由所述内容服务器提供的清单获得所述内容流的内容段的编码比特率和内容质量评分。

在自适应流送中用缓冲器和范围约束来进行质量优化

技术领域

[0001] 本公开涉及计算机网络中的自适应流送。

背景技术

[0002] 在传统的自适应流送中，视频流被分成各自为几秒的短段，其中，以若干不同的比特率从主要的高质量源来对这些段进行编码（或转码），并且输出被存储在客户端从其获取片段的服务器处。通常的做法是编码器 / 转码器使用恒定比特率 (CBR)，来产生视频输出的层 (tier) 或“等级 (level)”的集合。客户端应用使用 HTTP GET 请求来从服务器（经常顺序地）下载段、通过对下载性能进行测量来估计可用的带宽、以及在之前的段完成时选择要获取的下一段的视频等级。

[0003] 通常，几十秒所下载的视频段在客户端处被缓冲以适应带宽波动。可行的速率自适应算法实现了高的平均视频质量、低变化的视频质量和低概率的由缓冲器欠载运行 (underrun) 所引起的视频播出停止。

附图说明

[0004] 图 1 是示例系统的示意图，在该示例系统中自适应流送被用于通过网络向计算设备提供视频内容。

[0005] 图 2 是内容服务器的示例实施例的示意图，该内容服务器以不同比特率和质量等级来向图 1 的系统的计算设备提供内容流的段。

[0006] 图 3 是计算设备的示例实施例的示意框图，该计算设备接收视频流段以用于在图 1 的系统中的设备处显示并进行观看。

[0007] 图 4 是描述使用图 1 的系统以不同的比特率和相应的质量等级来对内容流的段进行编码的示例处理的流程图。

[0008] 图 5 是描述使用图 1 的系统基于缓冲器和范围约束来对内容流的段的质量进行优化的示例处理的流程图。

[0009] 图 6 是描述计算设备的示意图，该计算设备根据本文中所描述的技术以选定比特率来从内容服务器接收视频段。

具体实施方式

[0010] 概述

[0011] 本文中描述了用于优化自适应流送的质量的技术，其中，关于内容流的视频段的编码比特率和视频质量评分的信息被获得。内容流的视频段由内容服务器提供并且由计算设备下载。针对每一视频段，编码比特率被确定以用于从内容服务器下载视频段，其中，确定当前视频段的编码比特率是基于估计的当前可用网络带宽、下载的视频段被存储于其中的缓冲器的约束、并且还基于在所选择的范围（该所选择的范围从当前视频段延伸到在范围端点处的内容流中的将来视频段）内与视频段相关联的质量评分。每一内容段以针对该

内容段的所确定的比特率被从内容服务器下载。

[0012] 示例实施例

[0013] 本文中描述了用于通过应用动态速率自适应分析来提高自适应流送中的内容的质量的技术,该动态速率自适应分析基于视频段的给定的时间范围来对从内容服务器获取或下载的每一视频段的质量等级进行优化。分析将与显示内容的计算设备相关联的范围约束和存储缓冲器考虑在内,其中,视频质量优化基于如下约束而被确定:客户端缓冲器被限制在最高等级(例如,针对低延迟的实时流送)和最低等级(例如,针对最小化缓冲欠载运行的可能性的点播流送)之间。与优化的质量等级相关联的编码比特率被选择以用于每一视频段,其中,优化的质量等级只基于针对该内容的视频段的最近时间流中的选定范围内的视频段。

[0014] 不论来自内容服务器的视频内容是恒定比特率(CBR)编码的还是可变比特率(VBR)编码的都可以使用本文中所描述的技术。本文中所描述的技术还避免了链路带宽变化的任意统计模型,但替代地被配置为对任意可能情境的视频质量进行优化(由于每一段的视频质量基于当前缓冲器约束和视频段的有限或选定将来范围而被确定)。可以被实现以用于本文所描述的优化视频段的质量的技术的视频质量评分的一些非限制性示例包括PSNR(峰值信噪比)和负MSE(均方误差或-MSE)、MS-SSIM(多尺度结构相似性)与VQM(视频质量度量)。

[0015] 本文中所描述的自适应流送系统和方法的实施例使用恒定质量、或近乎恒定质量、段(还被称为区块或片段)来甚至在自适应流送的上移和下移期间提供更好的观看体验。在所描述的实施例中,计算设备可以通过选择恒定质量段来直接追踪和优化质量,以使得计算设备在让带宽波动期间维持平稳的质量,因此为其他客户端释放了可能以其他方式来拉取比当前质量等级需要的数据更多的数据而被浪费的带宽。计算设备通过以下方式来实现这一点:(a)接收可用带宽内的最高质量表示;(b)当带宽可用时,上移到更高的质量表示(而不是上移到更高的比特率,即使质量可能不被提高);和/或(c)当带宽缺乏时,下移到更低的质量。

[0016] 自适应流送系统一般地落入基于推送的(push-based)系统或基于拉取的(pull-based)系统。尽管自适应流送系统的某些实施例可以为了基于推送的系统(或者基于推送的系统和基于拉取的系统的组合)的益处而被使用,但本公开的重点在于基于拉取的系统。在基于拉取的自适应流送中,客户端决定接下来要从源(例如,源服务器或内容服务器、缓存服务器等)请求哪个段。该决定可以基于内容段和观测值的各种属性,比如,当前(所观测的或可用的)带宽和客户端缓冲器中的数据量。在当前系统中,适应可以采取请求以不同比特率被编码的段的形式。例如,客户端可以上移或下移(比如,切换到不同的比特率)或者保持在相同的比特率以提供连续的播放。比特率切换决定基于预定调度进行修订,并且然后新的段以确定的比特率被请求。各种目标函数是可能的,并且客户端可以在算法方面存在不同,该算法被用于尝试最大化在可用带宽约束内递送的视频质量。然而,传统系统间的共同特质是使用段编码比特率来作为视频质量的替代变量。

[0017] 在本文所描述的实施例中,不同于使用以各种目标比特率对内容进行编码的编码器,所公开的自适应流送系统的编码器以各种质量等级来对内容进行编码,并且然后基于时距(temporal duration)来将所编码的流划分成(或者相似地,分段成)段,段的大小以

字节计并且取决于编码速率,其进而对应于网络上所需要的传输带宽。质量变化根据预定质量窗来使能对比特率变化的约束。

[0018] 例如,编码器(或者在一些实施例中,诸如服务器之类的另一网络设备)提供由质量等级直接组织的(例如,明确指示的)内容矩阵。换言之,每一表示对应于(单个)质量等级而不是标称比特率或平均比特率。每一段的元数据包括质量等级、以及时距、和以字节计的段的尺寸或段平均比特率中的任一者(其中最后两个是共轭变量——一个共轭变量可以根据另一已知的时距而被计算得出)。此外,计算设备根据自适应流送系统的某些实施例而被配置为通过基于本文中所描述的技术选择下一段来在给定的带宽约束内实现平稳、非递减的质量。

[0019] 在一些实施例中,该基于质量的自适应流送方案可以被嵌入到元数据中以使得现有的计算设备(在不对软件和/或硬件进行修改的情况下)能够发送请求(例如,按照常规所做地,基于时间戳、比特率等),并且网络设备(例如,源服务器或缓存服务器)基于本文中所描述的技术来对具有最高质量等级的段进行访问。

[0020] 这些实施例和其他实施例和/或其他特征在下文中在示例计算机网络环境的上下文中被描述,其中内容出于协助理解本公开的目的而被具体化为视频。应该认识到,代替或者附加到视频的其他内容(例如,音频、数据、图形、图像等),也被认为在实施例的范围内。此外,自适应流送系统的某些实施例下面在使用超文本传输协议(HTTP)的环境中的自适应流送的上下文中被描述。HTTP是针对客户端-服务器计算的请求-响应标准。在HTTP中,Web浏览器作为客户端,而在托管网站或内容的计算机上运行的应用作为服务器。HTTP使用传输控制协议(TCP),该传输控制协议是被用于互联网和其他相似网络的通信协议的互联网协议(IP)系列的一部分。本领域的技术人员应该理解,对多媒体(例如,视频、图形、音频、和/或数据、或者在本文中还以其他方式单独或共同被称为介质内容、或者简单地被称为内容的多媒体)进行分发的其他网络也可以受益于自适应流送系统和方法的某些实施例,并且因此被认为是在本公开的范围内。本领域普通技术人员应该理解,尽管本文公开了一个或多个实施例的细节,但所描述的细节不必是每个实施例的一部分。

[0021] 图1是示例环境(计算机网络100)的示意图,在该示例环境中自适应流送系统和方法的某些实施例可以被实现。本领域普通技术人员应该知道,在本公开的上下文中,图1中所示的计算机网络100只是示意性的,并且不应该被解释为暗指对本公开的范围的任意限制。计算机网络100可以包括多个个人网络,比如,无线网络和/或有线网络。图1中所示的示例计算机网络100包括多个计算设备102、104、105和106(例如,有线和/或无线的设备,比如蜂窝电话、个人数字助手(PDA)、计算机设备或系统(比如膝上型计算机、个人计算机、机顶终端、具有通信能力的电视、DVD/CD记录器等)),该多个计算设备102、104、105和106与一个或多个服务器设备(比如,边缘或缓存服务器设备108)和/或一个或多个上行设备(比如,内容服务器设备114)进行通信。内容服务器设备114被以通信的方式耦合到一个或多个编码设备,比如,编码或转码设备116。在计算机网络100中可以采用另外的服务器。还在一些示例实施例中,内容服务器设备114和编码设备116可以被集成为单个服务器设备。缓存服务器设备108被耦合到上行网络(比如,广域网(WAN)112),在一个实施例中该上行网络包括互联网。在一些实施例中,缓存服务器设备108可以通过诸如局域网(LAN)之类的网络被耦合到计算设备102、104、105和106。其他网络(包括使用与其他传

输协议或标准进行合并的分组的网络)被认为是在本公开的范围内。缓存服务器设备 108 和内容服务器设备 114 还可以包括一个或多个存储设备(例如,与一个或多个存储设备集成)或者与一个或多个存储设备进行本地通信。

[0022] 缓存服务器设备 108 和计算设备 102、104、105 和 106 之间的通信可以是经由无线连接或有线连接,以非限制性示例的方式包括以太网、令牌环、私人或专有网络、及其他。缓存服务器设备 108 和 / 或内容服务器设备 114 可以包括互联网服务提供商 (ISP) 设施中的服务器、私有服务器、网关、和 / 或被用于传输流视频文件(以及其他内容,比如数据、音频等)的其他设备或设施。本领域普通技术人员应该理解,另外的计算设备、缓存服务器设备、上行服务器设备和诸如路由器、网桥等之类的其他设备可以被用于计算机网络 100。计算设备 102、104、105 和 106、缓存服务器设备 108 以及内容服务器设备 114 之间的互联网协议 (IP) 分组的通信可以根据多个不同的协议((除此以外)包括传输控制协议 (TCP)/IP)中的一个或多个而被实现。

[0023] 编码设备 116 被配置有处理器 118、存储器 120(其被配置有(可执行指令或代码的)编码逻辑 122)和通信(或网络)接口(I/O)126,以接收用于编码和分段(还被称为分块或划分)的内容流并且以不同质量等级的段的形式来将经编码的流发送到内容服务器 114。在一个实施例中,通信接口 126 接收数字化的内容流(例如,电影、电视节目或者体育或娱乐事件等的现场直播),并且处理器 118 被编码逻辑 122 配置为以不同质量等级的多个编码表示来对内容流进行编码。例如,一个表示可以包括以 40 分贝(dB)的(例如,以 PSNR--峰值信噪比所测量的)质量等级被接收的内容流的编码版本,第二表示可以包括以 35dB 的质量等级被接收的内容流的编码版本,并且第三表示可以包括以 30dB 的质量等级被接收的内容流的编码版本。处理器 118 还被编码逻辑 122 配置为将多个表示中的每一个划分成不同质量等级的段。给定质量等级(例如,40dB)的所有段的元数据(例如,诸如 URL 之类的位置、或者每一段的标识、段大小、质量等级、比特率、时间戳信息等)以各自的表示被列于清单(MFST)124 中。换言之,每个表示对应于不同的质量等级。例如,质量等级 40dB 的段的元数据以一个表示被列于清单中,质量等级 35dB 的段的元数据以第二表示被列出,等等。具体地,清单包括关于每一可用段比特率的视频质量评分的信息(VQ/BR 信息 125)。每一质量等级被明确地指示(例如,与从比特率推断的相反),其中明确的指示对应于质量度量(比如,分贝、峰值信噪比(PSNR)或其他众所周知的质量度量(例如,-MSE、MS-SSIM 或 VQM))。编码设备 116 通过通信接口 126 来向内容服务器设备 114 提供清单 124(或者在一些实施例中,用于在别处组装出清单的信息)和针对各种表示的段。

[0024] 尽管在存储于存储器 120 中的编码软件或固件逻辑 122 的上下文中进行描述,但本领域普通技术人员应该理解,编码 / 分段功能可以被分离在多个模块间,或者在一些实施例中使用硬件或硬件和软件 / 固件的组合而被实现。

[0025] 图 2 是示例内容服务器设备 114 的实施例的框图。本领域普通技术人员应该理解,在本公开的上下文中,图 2 中所示的内容服务器设备 114 只是示意性的,并且不应该被解释为暗指对本公开范围的任何限制。一般地,根据硬件架构,内容服务器设备 114 包括处理设备 202、存储器 204、输入 / 输出(I/O)设备 206 和网络接口 208,其中的每个通过本地接口 210 来以通信的方式进行耦合。处理设备 202 是用于执行软件的硬件设备,具体地其被存储于存储器 204 中。处理设备 202 可以是任意定制的处理器或市场上可买到的处理器、中央

处理单元 (CPU)、与内容服务器设备 114 相关联的若干处理器间的辅助处理器、基于半导体的微处理器 (以微芯片或芯片集的形式)、微处理器或一般地用于执行软件指令的任意设备。

[0026] 网络接口 208 包括传输输入和输出的一个或多个设备,例如但不限于,调制器 / 解调器 (用于访问另一设备、系统或网络的调制解调器)、无线电频率 (RF) 收发机或其他类型的收发机、电话接口、网桥、路由器等。

[0027] I/O 设备 206 可以包括输入设备,例如但不限于 : 键盘、鼠标、扫描仪、麦克风等。此外, I/O 设备 206 还可以包括输出设备,例如但不限于 : 打印机、显示器等。

[0028] 本地接口 210 可以是,例如但不限于本领域已知的一个或多个总线或其他有线或无线连接。本地接口 210 可以具有为了简单起见而被省略的另外的元件 (比如,控制器、缓冲器 (缓存)、驱动器、中继器和接收机) 以使能通信。此外,本地接口 210 可以包括地址、控制和 / 或数据连接以使能上述组件间适当的通信。

[0029] 在一个实施例中,内容服务器设备 114 还被配置有耦合到本地接口 210 的集成存储设备 212,该存储设备 212 被配置为存储从编码设备 116 接收的多个内容段。在一些实施例中,存储设备 212 可以外部地被耦合到内容服务器设备 114。

[0030] 存储器 204 包括合适的操作系统 (O/S) 214 和段选择逻辑 216。操作系统 214 实质上控制其他计算机程序 (诸如,段选择逻辑 216) 的执行,并提供调度、输入输出控制、文件和数据管理、存储器管理以及通信协议和相关的服务。段选择逻辑 216 包括可执行代码 (其对针对段的计算设备请求 (例如, HTTP 功能) 进行响应) 和对应于完全或部分 (还持续地进行更新的) 已完成清单 218 的数据结构,该清单 218 对应于由编码设备 116 发送的清单 124(或本文中的信息)。在一些实现方式中,清单 218 可以是不完整的、还在根据编码设备 116 持续地进行组装和更新的 (例如,针对现场直播)。

[0031] 存储设备 212 以不同质量等级的段的形式来存储内容的不同表示。例如,以电影为例,存储设备 212 以第一质量等级 (例如,恒定质量 (CQ) A#1) 存储电影 A 的第一表示,以第二质量等级 (比如, CQA#2) 将电影 A 的第二表示,等等。每一表示是以给定质量等级的多个段的形式,如由图 2 中针对表示 CQA#N (并且类似地适用于其他表示) 的个体、较短块所部分示出的。这些段中的每个通常是可独立解码的 (例如,每一段以一个或多个图片组 (GOP) 的形式),并且一般地持续 2-10 秒,但不限于该持续时间。针对这些不同的质量等级段中的每一个,对于每一段的相关联的元数据的对应的表示在存储于存储器 204 中的清单 218 中被维持。清单 218 (与清单 124 相似) 还包括关于针对每一电影或其他视频内容的每一可用的段比特率的视频质量评分的信息 (VQ/BR 信息 125)。

[0032] 段选择逻辑 216 被配置为对来自计算设备 102、104、105 和 / 或 106 (和 / 或到缓存服务器 108) 的对清单 218 的请求进行回复,或者选择其某些部分。这些下载或获取请求可以通过相同或不同的服务器连接而被作为用于获取或下载内容段的请求。在一个实施例中,使用具体化为机顶盒的计算设备 105 作为下文中的示例计算设备 (应理解类似的原则可应用于其他计算设备),计算设备 105 基于从内容服务器设备 114 获取的清单 218 的列表来请求段。计算设备 105 通过质量等级 (不同于比特率) 有效地请求段,带来了更平滑的、恒定质量的观看体验。在一些实施例中,清单和所参考的段被从缓存服务器设备 108 获取,由计算机网络 100 中的其他计算设备协助对请求的快速响应和 / 或由请求计算设备 105 协

助特技模式。

[0033] 在一些实施例中,如以上所解释的,计算设备 105(或者在一些实施例中,传统的计算设备)可以发送传统的请求(不具有对质量等级的明确指示,并且替代地使用比特率和带宽约束,以及其他可能的参数),并且段选择逻辑 216 通过参考清单 218 来确定符合客户端传输带宽约束的最优质量等级、并进行选择、然后发送于所确定的质量等级相对应的段来对请求进行响应。

[0034] 存储器 204 可以包括易失性存储器元件(例如,随机存取存储器(RAM,诸如 DRAM、SRAM、SDRAM 等))和非易失性存储器元件(例如,ROM、硬盘驱动器、磁带、CDROM 等)中的任意一个或组合。此外,存储器 204 可以包括电的、磁的、光学的、半导体的和 / 或其他类型的存储介质。注意到,存储器 204 可以具有分布式架构,其中各种组件彼此间相距较远,这些组件可以由处理设备 202 进行访问。因此,一般地,存储器 204 可以包括一个或多个有形的(非暂态的)计算机可读存储介质(例如,存储器设备),该计算机可读存储介质编码有包括计算机可执行指令的软件,并且当软件被(处理设备 202)执行时可操作来执行本文中所描述的操作。

[0035] 存储器 204 中的软件可以包括一个或多个单独的程序,其中每个包括用于实现逻辑功能的可执行指令的有序列表。在图 2 中所示的实施例中,存储器 210 中的软件包括操作系统 214 和段选择逻辑 216,如上所述。段选择逻辑 216 的功能可以使用单个模块被实现或者被分布在多个模块中。

[0036] 当段选择逻辑 216 进行操作时,处理设备 202 被配置为执行存储在存储器 204 内的段选择逻辑 216、被配置为将数据传输到存储器 204 和从存储器 204 传输数据、以及一般地被配置为对段选择逻辑 216 的操作进行控制。段选择逻辑 216 和 O/S 214,整体地或部分地(但通常是后者),被处理设备 202 读取,并且然后被执行。

[0037] 图 3 是示出示例计算设备 105 的实施例的框图。本领域的技术人员应该理解,在本公开的上下文中,图 3 中所示的计算设备 105 只是示意性的,并且不应该被解释为暗指对本公开的范围的任意限制。相似的功能可以在计算设备 102、104 和 106 中被发现。计算设备 105 包括通信接口 302,该通信接口 302 适用于使能与计算机网络 100 的设备的通信(例如,TCP/IP),并且适用于接收清单(或本文中的信息)以及被请求的段。例如,通信接口 302 可以是适用于通过 IP 网络、同轴电缆网络、HFC 网络和 / 或无线网络及其他进行通信的类型的通信接口。通信接口 302 被耦合到多路输出选择器(本文中,还称作多路分配器)304。多路分配器 304 被配置为对(例如,在分组报头中的)段的元数据和体数据或有效负载数据(诸如视频和 / 或音频帧序列之类的内容)进行解析。元数据可以包括时间戳信息、分组标识符、程序号、质量等级和 / 或对所接收的段进行编码必要的其他信息。段和元数据信息如下面所进一步解释的那样被转发到媒体引擎 306 或者以其他方式被媒体引擎 306 接收。

[0038] 尽管计算设备 105 在诸如 IPTV 和 VoD 之类的各种互联网视频流送实现方式的上下文中被描述,但本领域普通技术人员应该认识到,在一些实施例中计算设备 105 可以包括额外的和 / 或不同的组件。例如,计算设备 105 的一些实施例可以包括被耦合到通信接口 302 的调谐器系统(例如,射频调谐(未示出)),该调谐器系统包括一个或多个调谐器以用于接收通过通信接口 302 接收的传输流。此外,在一些实施例中,解调器可以例如被用于

对所接收的载波信号进行解调,其中,多路分配器 304 被配置为对一个或多个定义的载波频率的传输流分组进行解析。

[0039] 多路分配器 304 被耦合到总线 305 和媒体引擎 306(还被称为音频 / 视频 (A/V) 处理或解码设备)。在一个实施例中,媒体引擎 306 包括解码逻辑,该解码逻辑包括响应的音频解码器 308 和视频解码器 310 中的一个或多个,解码器其对由通信接口 302 接收的段进行解码,例如,解压缩和重建。当段被回放时,原内容流以无缝的方式被重建。计算设备 105 动态地请求和接收具有恰当质量等级的段。计算设备 102 可以根据可用的接收带宽来适配其内容消耗速率。

[0040] 媒体引擎 306 还包括缓冲器管理功能以基于本文中所描述的技术来协助(例如,与 HTTP 逻辑 325 合作)做出关于需要下移或上移到不同的质量等级的确定。媒体引擎 306 还被耦合到总线 305 和媒体存储器 312,在一个实施例中该媒体存储器 312 包括一个或多个缓冲器 313 以用于临时存储被压缩的和 / 或被重建的图片和 / 或音频帧。在一些实施例中,媒体存储器 312 的缓冲器和 / 或其他缓冲器(例如,网络缓冲器)可以驻留于其他存储器(例如,以下所描述的存储器 322)中,或者被分布在媒体存储器 312 和存储器 322 间。此外,媒体引擎 306 包括质量优化 (QO) 引擎 309,其被用于根据本文中所描述的技术基于对有限数目或选定将来段的分析来确定每段的优化质量。

[0041] 计算设备 105 包括被耦合到总线 305 的额外的组件。例如,计算设备 105 还包括接收机 314(其被配置为(例如,通过直接物理的或无线连接,经由键盘、远程控制、语音激活等来)接收用户输入,以传达用户的请求或命令(例如,用于程序选择、诸如快进、倒带、暂停、频道切换之类的流操纵))、一个或多个处理器 316(示出了一个)(其用于控制计算设备 105 的操作等))、和时钟电路 318,时钟电路 318 包括将在音频、视频、或 A/V 流中接收的系统时钟信息锁定的相位和 / 或频率锁定环电路(或软件、或硬件和软件的组合)以协助解码操作并且为所重建的视听内容的输出计时。

[0042] 在一个实施例中,计算设备 105 还包括存储设备 320(和相关联的控制逻辑)以临时存储内容(例如,连同媒体存储器 312 的(一个或多个)缓冲器 313 所缓冲的内容)和 / 或更永久地存储所记录的内容。计算设备 105 中的存储器 322 包括易失性和 / 或非易失性存储器,并且被配置为存储与操作系统 (O/S) 324 相关联的可执行指令或代码、以及一个或多个应用 326(例如,除了诸如按次计费、音乐、个人视频录像 (PVR)、驱动器软件等之类的其他应用之外还包括交互式编程指南 (IPG)、点播视频 (VoD)、(与广播网络 TV 相关联的) WatchTV、HTTP 逻辑 325)。HTTP 逻辑 325 包括具有 HTTP 客户端功能的一个或多个处理器,并且可以,例如基于来自清单 218 的信息(例如,包括 VQ/BR 信息 125)来生成对段的请求。由 HTTP 逻辑 325 决定上移或下移(或维持)当前的质量等级可以基于来自媒体引擎 306 的反馈(例如,以上所描述的缓冲器等级)和被监控的带宽值。具体地,HTTP 逻辑 325 基于从 QO 引擎 309 接收的信息来生成针对段的请求,其利用本文中所描述的技术来基于对选定范围的将来段的分析来确定段的质量等级。HTTP 逻辑 325 存储所接收的清单(或其中所包括的信息),并且由此获知存储在内容服务器设备 114 处的段的可用的质量等级。因此,在不同表示间的切换(例如,到不同质量等级)通过使用清单中的信息(例如,基于 VQ/BR 信息 125)而被使能。尽管 HTTP 逻辑 325 的功能被描述为软件,但在一些实施例中其可以被配置于硬件或者硬件和软件的组合中。

[0043] 如以上所指示的,计算设备 105 还被配置有显示和输出逻辑 330,显示和输出逻辑 330 除本领域已知的其他电路之外还包括图形和视频处理管线(pipeline)以处理经解码的图片和相关联的音频,并且在显示设备或其他媒体设备上提供表示(例如,显示)或者提供与显示设备或其他媒体设备相关联的表示(例如,显示)。通信端口 332(或多个通信端口)还被包括在计算设备 105 中以用于从其他设备接收信息以及将信息发送到其他设备。例如,通信端口 332 可以描绘 USB(通用串行总线)、以太网、IEEE-1394、串行和 / 或并行的端口等。此外,通信端口 332 可以被配置用于家庭网络(例如,HPNA/MoCA 等)。计算设备 105 还可以包括用于接收模拟视频信号的模拟视频输入端口。

[0044] 本领域普通技术人员应该理解,在本公开的上下文中,计算设备 105 可以包括未示出的其他组件,包括压缩引擎、存储器、解密器、采样器、数字转换器(例如,模数转换器)、多路复用器、条件访问处理器和 / 或应用软件、驱动器软件、互联网浏览器、及其他。此外,尽管 HTTP 逻辑 325 被示为驻留于存储器 322 中,但应该理解在一些实施例中,HTTP 逻辑 325 可以被合并在媒体引擎 306 中或别处(比如,O/S 324 中,及其他位置中)或者与计算设备 105 的其他逻辑相结合。相似地,在一些实施例中,在图 3 中被示出的或者与图 3 相关联地进行描述的组件中的一个或多个的功能可以与另一组件一起被组合为单个集成组件或设备。

[0045] 自适应流送系统的某些实施例可以包括以下各项中的一个或组合的全部或一部分:编码设备 116、缓存服务器 108、内容服务器 114 或者一个或多个计算设备 102、104、105 和 / 或 106。

[0046] 本公开的某些实施例中的 HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 可以在硬件、软件、固件或其组合中实现。在 HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 在软件或固件(统称为,软件代码或软件逻辑、或者简单地逻辑)中实现的实施例中,这些实施例被存储在存储器中,并且由合适的指令执行系统使用计算设备的一个或多个处理器来执行。当 HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 在软件中被实现时,应该注意到,HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 可以被存储在任意计算机可读存储介质上,以供任何计算机相关的系统或方法使用或与之相结合。因此一般地,HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 可以包括一个或多个有形的(非暂态的)计算机可读存储介质(例如,存储器设备),该计算机可读存储介质编码有包括计算机可执行指令的软件,并且当软件被执行时可操作来执行本文中所描述的操作。在本文档的上下文中,计算机可读存储介质是电的、磁的、光学的、半导体的或者其他有形的、物理的设备或装置,该设备或装置可以包含或存储被计算机相关的系统或方法使用的或者与其相关的计算机程序。HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 可以体现于非暂态的计算机可读介质中,以供指令执行系统、装置或设备(比如,基于计算机的系统、包含处理器的系统或者可以从指令执行系统、装置或设备获取指令并执行指令的其他系统)使用或者与之相结合。

[0047] 在 HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 的功能的全部或一部分在硬件中实现的实施例中,HTTP 逻辑 325、媒体引擎 306、编码逻辑 122 和 / 或段选择逻辑 216 的这样的功能可以用本领域周知的以下技术中的任意或组合来实现:(一

个或多个)离散的逻辑电路(其具有用于实现基于数据信号的逻辑功能的逻辑门)、具有合适的组合逻辑门的专用集成电路(ASIC)、(一个或多个)可编程门阵列(PGA)、现场可编程门阵列(FPGA)等。

[0048] 现在参考图4和图5的流程图来描述描绘的系统来针对使用本文中所描述的并且在图1-3中描绘的系统来优化内容流的段的质量的示例方法。参考图4,在400,编码设备116生成内容流的清单124(其还可以被修改或补充,如同存储在内容服务器114的存储器204内的清单218),其中清单包括针对将被一个或多个计算设备102、104、105、106、108下载的编码的视频段的多个比特率和对应的质量评分。如先前所指出的,内容流可以被例如从存储设备212获得,该存储设备212将内容(例如,电影或其他视频表示)存储于不同质量等级处(例如,将电影A存储于CQA#1、CQA#2…CQA#N处、将电影B存储于CQB#1处等)。在410,响应于计算设备(例如,计算设备105)请求某一内容,内容服务器114向该计算设备提供与所请求内容相关联的清单,以协助计算设备对编码比特率的确定,以该编码比特率、使用本文中所描述的技术、基于质量优化来接收内容流的段。在420,内容流的段基于质量优化、以由计算设备为每个段选定的比特率被提供给计算设备。

[0049] 参考图5,在500,当计算设备处的用户想要观看从内容服务器114可得的内容时,内容流的清单被提供给该计算设备。本文中描述了关于(例如,图3中所描述的)计算设备105的技术。然而,如先前所指出的,具有相似硬件和/或软件组件(包括QO引擎)的任意其他合适的计算设备可以被用于执行本文中所描述的技术。清单包括关于内容流的每一经编码段的每一可用段比特率的视频质量评分的信息(VQ/BR信息125)。在510,对将被计算设备105使用的当前可用的网络带宽做出估计。所估计的网络带宽可以以任意合适的方式(例如,基于任意传统或其他合适的测量或估计方法)由计算设备确定。在520,计算设备105的QO引擎309基于当前缓冲器状态、所估计的网络带宽和选定范围中的内容流的段的质量值(基于如本文所述的那样起作用的在线算法和一个或多个动态编程的算法)来确定优化当前段的质量用于当前段的编码比特率。在530,以确定的优化质量的编码比特率来从内容服务器114下载当前段。在存在针对(在540确定的)流的另一段的情况下,下一段在550被设置为当前段,并且处理在510被重复。

[0050] 注意到,尽管图5中所描述的流程图提供了其中清单最初在计算设备处被接收的步骤,但处理可以被容易地修改,以使得清单在很多时候可以在处理期间由计算设备进行接收。例如,在某些情境下,清单可以定期地被更新,以使得经修改的清单(例如,带有经修改的VQ/BR信息)被提供给计算设备。

[0051] 媒体引擎306的QO引擎309,在520处做出关于当前段的编码比特率时,使用在线算法和一个或多个动态编程算法来进行视频段的比特率选择,其中,基于在选定将来时间窗口或范围上的质量优化分析并且将计算设备105的当前缓冲器约束考虑在内来为每一段选择合适的比特率。在每一速率选择步骤,QO引擎309使用最新的可用信息、解决选定范围内的视频段的优化问题、并且计算可用比特率中用于最紧急的视频段的最优比特率(其中,过程针对每下一段而被重复进行)。在每一步骤中,引擎可用的信息包括(例如,在510基于任意传统的或其他合适的测量或估计方法而确定的)计算设备105的当前估计的链路带宽和(在清单124内提供的)所定义的范围内的视频段的比特率和质量评分。

[0052] QO引擎309在考虑到当前条件(例如,网络带宽条件、缓冲器条件和来自当前段的

有限数量或范围内的段的质量等级) 来为内容流中的每段确定产生优化的质量的比特率时, 采用一系列速率选择步骤 n(其中, $n = 1, 2, 3 \dots$)。用于下载当前视频段的比特率(定义为 $R(n)$) 在每一步骤 n 被确定。针对每一视频段存在有限数量的可用比特率。因此, 当前视频段的可用比特率的数量被定义为 $R(n, 1)$, 其中 $1 = 1, 2, 3 \dots L$, 其中 L 是可用比特率的数量。每一比特率 $R(n, 1)$ 还与视频质量评分 $Q(n, 1)$ 相关联。与当前段的选定比特率 $R(n)$ 相关联的质量评分被定义为 $Q(n)$ 。如先前所指出的, 视频段的质量评分可以使用任意适当的技术来确定, 其中一些非限制性的示例包括 PSNR 和 -MSE。

[0053] QO 引擎 309 所使用的算法是基于将范围上的内容流的视频质量的一些综合测量最大化的。本文中所描述的技术使用两个目标函数来将质量最大化。第一个目标是将平均质量评分最大化, 如下式所定义的 :

$$[0054] \max \frac{1}{N} \sum_n Q(n) \quad (\text{公式 } 1)$$

[0055] 其中, N 是段的数目(例如, 范围内段的数目)。

[0056] 第二个目标是将针对段 N 的数目的最差情况下的质量评分最大化, 如下 :

$$[0057] \max \min_n Q(n) \quad (\text{公式 } 2)$$

[0058] 对公式 1 和公式 2 的扩展将通过以下方式来精确地考虑质量变化 : 在每一步骤 n 包含乘法器 $a(n-1, n)$ 以使得公式 1 和公式 2 中的 $Q(n)$ 被替换为 :

$$[0059] Q'(n) = a(n-1, n)Q(n), \quad (\text{公式 } 2b)$$

[0060] 其中, $Q(n)$ 对应于初始的 $Q(n)$ 。当质量已经从一个等级转移到另一等级时, 该乘法器 $a(n-1, n)$ 将整体评分打折扣。例如, 如果在步骤 $n-1$ 和步骤 n 的视频比特率是从相同质量等级选出的, 则设置 $a(n-1, n) = 1$, 并且如果在步骤 $n-1$ 和步骤 n 的视频比特率是从不同质量等级选出的, 则设置 $a(n-1, n) = 0.8$ 。这种方式下, 目标函数考虑平均质量和质量变化。

[0061] 对以上目标(公式 1、公式 2 和公式 2b) 的第二个扩展将根据幂函数, 采取任意的质量评分并将其提高到某一幂次 b, 也就是, 由下式来替换这些公式中的 $Q(n)$:

$$[0062] Q''(n) = Q(n)^b \quad (\text{公式 } 2c)$$

[0063] 其中, $Q(n)$ 对应于初始的 $Q(n)$, 并且 $b \geq 1$ 。

[0064] 注意到, 先前所描述的算法中的任意一个或多个可以被应用于本文中所描述的目标中的任意一个。视频质量的优化还基于介质存储缓冲器 313 的约束以及在速率步骤期间进行分析的当前段的范围大小 $H(n)$ 。缓冲器包括视频段的队列。缓冲器的大小随着在每步新的视频段队列到缓冲器中和旧的视频段以具体的视频播放速率出列随着时间演化。如本文中所描述的, $B(0)$ 表示最初的缓冲器大小, 并且 $B(n)$ 是步骤 n 结束时的客户端缓冲器大小。在视频播放开始之前(即, 在缓冲器的任意损耗之前), 缓冲器大小被定义为 :

$$[0065] B(n) = B(n-1) + 1 \quad (\text{公式 } 3)$$

[0066] 在视频播放开始之后, 缓冲器大小被定义为 :

$$[0067] B(n) = B(n-1) + R(n)/W(n) \quad (\text{公式 } 4)$$

[0068] 其中, $W(n)$ 是步骤 n 的链路带宽, 并且 $R(n)/W(n)$ 是段下载的持续时间。因此, 缓冲器的补充或入队是每步一段, 而缓冲器的消耗或出队在步骤 n 结束时的 $R(n)/W(n)$ 。

[0069] 缓冲器的下限值 (B_L) 和缓冲器的上限值 (B_H) 由引擎 309 定义。除了初始状态 (其中 $B(n) < B_L$) 之外, 当处理当前段时, 引擎 309 所使用的算法维护缓冲器大小, 以使得 $B_L \leq B(n) \leq B_H$ 。引擎 309 还尝试收敛到值 $B(n) = B_{ref}$, 其中 $B_L \leq B_{ref} \leq B_H$ 。

[0070] 时间范围大小被 QO 引擎 309 定义为 $H(n)$, 其中 $H(n)$ 可以在 $H(n) \geq 1$ 的条件下改变。例如, 时间范围大小对于内容流中的某一数目的段可以是恒定的, 但是随着段接近流的终点, $H(n)$ 的值被减少以便不延伸到内容流之外的范围。根据本文中所描述的技术, 从当前段到当前段之前的 $H(n)-1$ 步骤, 只有段的视频比特率和质量信息被分析以确定对应于当前段的优化的质量的比特率。换言之, 只有信息 $R(m, l), Q(m, l)$ (其中, $m = n, n+1, \dots, n+H(n)-1$, 并且 $l = 1, 2, \dots, L$) 对由引擎 309 用来确定当前段的 $R(n)$ 的算法是可用的。

[0071] 为了应用算法来确定当前段的 $R(n)$, QO 引擎 309 使用优化质量在线 (OptimizeQualityOnline) 算法来通过多个步骤进一步确定比特率。优化质量在线算法如本文中所描述的那样在每一步骤应用进一步的算法, 并且被称为范围内最优质量 (OptimizeQualityWithinHorizon)。以下输入被提供给优化质量在线算法:

[0072] 全局输入:

[0073] -- B_L =缓冲器下限

[0074] -- B_H =缓冲器上限

[0075] -- B_{ref} =缓冲器参考等级

[0076] -- κ_B =缓冲器收敛于 B_{ref} 的速度

[0077] 在步骤 n 可用的输入:

[0078] -- $W(n)$ =链路带宽 (例如, 在步骤 n 估计的带宽)

[0079] -- $B(n-1)$ =在之前步骤结束时的缓冲器大小

[0080] -- $H(n)$ =范围大小, $H(n) \geq 1$

[0081] -- $R(m, l), Q(m, l)$, 其中 $m = n, n+1, \dots, n+H(n)-1$, 并且 $l = 1, 2, \dots, L$,

[0082] (针对从当前段到当前段之前的 $H(n)-1$ 步骤处的段、或者到有限范围端点处的段的段的 (从清单可得的) 比特率和质量信息)

[0083] 算法使用输入信息来确定当前视频段 n 的比特率 $R(n)$ 。

[0084] 在每一速率选择步骤 n 处, 给定之前提到的输入, 引擎 309 使用在本文中被称为范围内优化质量的一个或多个算法来反复地解决优化问题以在范围内的段间最优地分配比特率。(一个或多个) 范围内优化质量算法使用以下输入:

[0085] 输入:

[0086] -- B_{init} =最初的缓冲器大小

[0087] -- B_{final} =最终的缓冲器大小

[0088] -- B_L =缓冲器下限

[0089] -- B_H =缓冲器上限

[0090] -- W =当前估计的带宽

[0091] -- H =当前范围大小

[0092] -- $R(m, l), Q(m, l)$, 其中 $m = 1, 2, \dots, H$, 并且 $l = 1, 2, \dots, L$

[0093] (针对从步骤 1 到步骤 H 的段的比特率和质量信息)

[0094] 使用范围内优化质量, 从当前段到当前段之前的 $H-1$ 步骤处的段的段的比特率输

出被确定（即， $R(m)$ 的值，其中 $m = 1, \dots, H$ ）。（一个或多个）范围内优化质量算法可以产生情境中可接受的结果，在这些情境中某些条件（例如，链路带宽和视频源特性）不改变。

[0095] 然而，实际上，带宽和视频源特性将很可能针对内容流进行改变。因此，根据本文中所描述的技术，引擎 309 只采用当前段的比特率，在当前段中算法被采用，并且在每一之后的步骤处重复相同的算法，以使得每次一个步骤确定段比特率。使用该方法，优化质量在线算法可以如下被应用：

[0096] $--B_{init} = B(n-1)$

[0097] $--B_{final} = B(n-1) + \kappa_B * H(n) * (B_{ref} - B(n-1))$

[0098] $--B_L^{local} = \min(B_L, B(n-1))$

[0099] $--R^{local}(m)$ (其中 $m = 1, 2, \dots, H$) = 范围内优化质量 ($B_{init}, B_{final}, B_L^{local}, B_H, W(n), H(n), R(m, 1), Q(m, 1)$)，其中， $m = n, n+1, \dots, H(n)-1$ ，并且 $l = 1, 2, \dots, L$)。

[0100] 为当前段选择的比特率 $R(n)$ 然后被输出为 $R^{local}(1)$ (也就是，被选择用于 H 个段的群组内的第一段的比特率)。针对下一段 (段 $n+1$)，相同的优化质量在线算法被再次应用以获得该段的比特率、之后段的比特率等等。

[0101] 注意到，缓冲器值 B_L^{local} 是 B_{init} 的上限，并且因此可以不同于 B_L ，并且范围内最优质量函数要求 $B_L \leq B_{init} \leq B_H$ 。每一步骤缓冲器向 B_{ref} 的收敛速度由参数 κ_B 来控制。

[0102] 已经定义了针对范围内优化质量函数的输入，在每一步骤 n 为段找到合适的比特率的质量优化是由引擎 309 使用动态编程算法来确定的，以便通过将其分解成可以被反复地解决的更小的子问题来解决问题。如先前所指出的，在步骤 n ，段的每一比特率 $R(n)$ 具有如以下所定义的相应的质量评分：

[0103] $Q(n) = Q(R(n))$ (公式 5)

[0104] 公式 5 可以基于缓冲器而被改写为如下：

[0105] $Q((n-1, B(n-1)) \rightarrow (n, B(n))) = Q(R(n))$

[0106] 换言之，通过将缓冲器从步骤 $n-1$ 处的 $B(n-1)$ 移动到步骤 n 处的 $B(n)$ ，所得到的质量是 $Q(R(n))$ 。 $Q(R(n))$ 的值然后可以使用公式 3 至公式 5 进行估计。

[0107] 相似的质量评分 $Q((m, B(m)) \rightarrow (n, B(n)))$ 可以被定义，其中 $n-m > 1$ 。然而，针对该情况，可以存在从 $(m, B(m))$ 到 $(n, B(n))$ 的多个可能的路径，其产生不同的质量值。最优的质量评分可以通过所有的质量值而被获得 (如 $Q*((m, B(m)) \rightarrow (n, B(n)))$)，其中优化可以以任意合适的方式被定义，比如在平均意义上 (例如，将平均质量评分最大化，如在公式 1 中) 或者在最差情况意义上 (例如，将最差情况的质量评分最大化，如在公式 2 中)。

[0108] 在多项式时间动态编程算法中，问题可以被拆分成更小的子问题以实现更有效的解决方案。在平均意义上使用最优的多项式时间动态编程公式，如下：

$$\begin{aligned} [0109] Q^*((m, B(m)) \rightarrow (n, B(n))) \\ = \max_{B(s), B_L \leq B(s) \leq B_H} Q^*((m, B(m)) \rightarrow (s, B(s))) + Q^*((s, B(s)) \rightarrow (n, B(n))) \end{aligned}$$

[0110] 其中所有的 $m < s < n$ 。

[0111] 针对该公式，从 $(m, B(m))$ 进行到 $(n, B(n))$ 需要步骤 s 进行处理。最大的质量一定是子问题的质量的总和，其中所有可能的 $B_L \leq B(s) \leq B_H$ 。为解决质量优化，更小的子问题可以被解决，然后解决甚至更小的子问题，等等。最后，公式减少到 $Q((n-1, B(n-1)) \rightarrow (n,$

$B(n))$ 。

[0112] 相似地, 针对最差情况意义下的优化, 情形如下:

[0113]

$$Q^*((m, B(m)) \rightarrow (n, B(n)))$$

$$= \max_{B(s); B_L \leq B(s) \leq B_H} \min(Q^*((m, B(m)) \rightarrow (s, B(s))), Q^*((s, B(s)) \rightarrow (n, B(n))))$$

[0114] 其中, 所有的 $m < s < n$ 。

[0115] 比特率解决方案可以以前向或后向递归的方式被实现。现在描述用于在每一步骤 n 使用动态编程算法确定段的比特率的以下两个示例算法: 范围内最优质量平均质量 (OptimizeQualityWithinHorizonAverageQuality) 和范围内最优质量最差情况质量 (OptimizeQualityWithinHorizonWorstCaseQuality)。

[0116] 范围内最优质量平均质量

[0117] 在每一步骤 n , 针对所有的 $R(1, 1)$, 设置 $B(0) = B_{init}$, 以及计算并存储, 其中 $1 = 1, \dots, L$:

[0118] $Q*((0, B(0)) \rightarrow (1, B(1))) = Q((0, B(0)) \rightarrow (1, B(1)))$

[0119] 针对 $m = 2, 3, \dots, H$, 计算并存储:

$$Q^*((0, B(0)) \rightarrow (m, B(m)))$$

$$\begin{aligned} [0120] &= \max_{B(m-1); B_L \leq B(m-1) \leq B_H} Q^*((0, B(0)) \rightarrow (m-1, B(m-1))) \\ &\quad + Q^*((m-1, B(m-1)) \rightarrow (m, B(m))) \end{aligned}$$

[0121] 针对所有的 $B_L \leq B(m) \leq B_H$, 其中 $B(m)$ 的值被量化成离散本 (bin)。

[0122] 优化比特率 $R*(m)$ ($m = 1, \dots, H$) 基于 $Q*((0, B(0)) \rightarrow (m, B(m)))$ 的所计算的值而被确定。例如, 清单 124 可以被查询以确定哪些比特率符合所计算的质量评分值。当前段的比特率是 $R*(1)$ 。

[0123] 范围内最优质量最差情况质量

[0124] 在每一步骤 n 处, 针对所有的 $R(1, 1)$, 设置 $B(0) = B_{init}$, 以及计算并存储, 其中 $1 = 1, \dots, L$:

[0125] $Q*((0, B(0)) \rightarrow (1, B(1))) = Q((0, B(0)) \rightarrow (1, B(1)))$

[0126] 针对 $m = 2, 3, \dots, H$, 计算并存储:

$$Q^*((0, B(0)) \rightarrow (m, B(m)))$$

$$\begin{aligned} [0127] &= \max_{B(m-1); B_L \leq B(m-1) \leq B_H} \min(Q^*((0, B(0)) \rightarrow (m-1, B(m-1))), \\ &\quad Q^*((m-1, B(m-1)) \rightarrow (m, B(m)))) \end{aligned}$$

[0128] 针对所有的 $B_L \leq B(m) \leq B_H$, 其中 $B(m)$ 的值被量化成离散本 (bin)。

[0129] 最优比特率 $R*(m)$ ($m = 1, \dots, H$) 基于 $Q*((0, B(0)) \rightarrow (m, B(m)))$ 的所计算的值而被确定。当前段的比特率是 $R*(1)$ 。

[0130] 如图 5 中所描述的,响应于在 520 确定当前段的比特率,在 530 当前段以所确定的比特率被下载,并且然后处理针对每一后续段被重复(即,重复步骤 510-530,包括使用优化质量线上算法和范围内最优质量算法以确定下一视频段的比特率),直到不再需要视频段。

[0131] 图 6 的示例实施例示意地描绘了由计算设备(例如,图 1 中所描述的计算设备中的任意一个,比如,膝上型计算机 102)以选定编码比特率从内容流的视频段的集合 600 中选择每一视频段,其中集合 600 包括以不同比特率编码的段和相应的质量等级(例如,由内容服务器 114 从存储设备 212 获取的,比如,电影 A)。每一视频段的编码比特率基于本文中所描述的技术进行选择以便当由媒体播放器 606(例如,膝上型计算机 102 的媒体播放器,其以相似的方式被配置并且被配置有与计算设备 105 的媒体引擎 306(包括媒体播放器 606 的 QO 引擎)相似的功能)进行播放时对被显示的视频内容的质量进行优化。每一段以所选择的编码比特率被下载,并在其使用媒体播放器 606 以时间序列被播放之前,被临时存储在存储缓冲器 613 中。

[0132] 传统的算法只基于视频比特率信息,并且不直接基于段的视频质量来选择下一段的比特率。然而,给定固定的比特率,视频的质量将根据视频内容(例如,高运动或高复杂度的场景(比如,具有复杂图形的动作场景)对比低运动或低复杂度的场景(比如,静止场景))变化。本文中所呈现的技术根据这样的传统的算法进行改进以进一步将段的视频质量考虑在内。

[0133] 尽管本文中所描述的技术是关于在从服务器下载视频内容期间对视频质量进行优化,但注意到,针对优化质量的技术适用于被下载以供在计算设备处使用的任意类型的媒体内容,例如,音频、视频和 / 或其他类型的内容。

[0134] 因此,本文中所描述的技术提供线上算法和一个或多个动态编程算法,以用于当解决质量优化问题时,通过考虑缓冲器和范围约束来优化自适应流送中的视频(和 / 或其他内容)的质量。技术适用于 CBR 和 VBR 编码的视频内容,并且还可以与被分配给视频段的任意线性视频质量评分值一起来使用(即,技术结合各种类型的视频质量测量)。

[0135] 用于诸如本文中所描述的系统的实现方式通过在计算设备处播放内容流期间提供更高的平均质量或减少质量变化来提高视觉质量。这些技术可以实现将平均视频质量最大化以及将最差情况的质量最大化的算法。此外,任意在线流送视频系统(包括提供点播视频和实时流送服务的系统)可以使用本文中所描述的技术。

[0136] 以上描述只意图是以示例的方式。

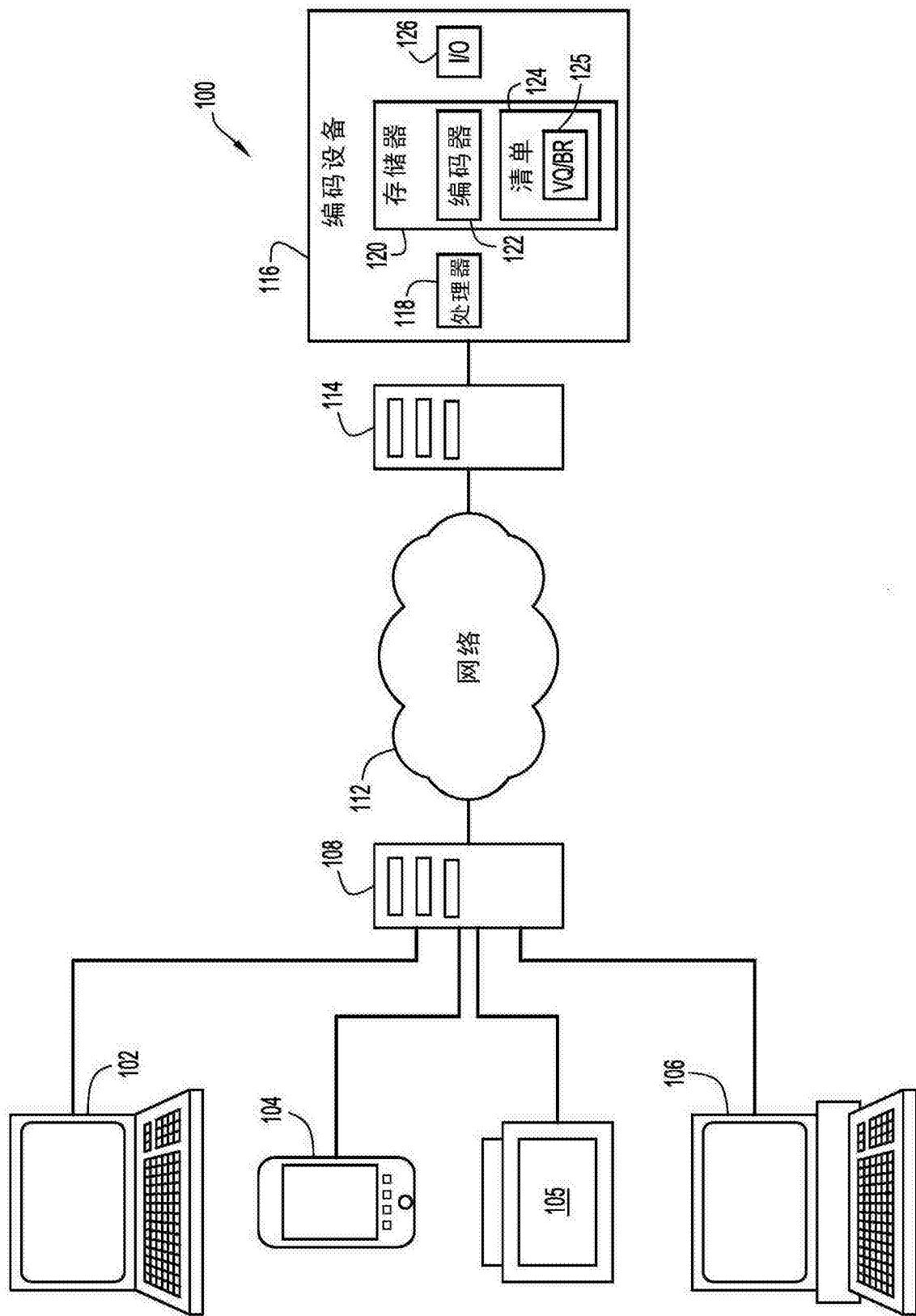


图 1

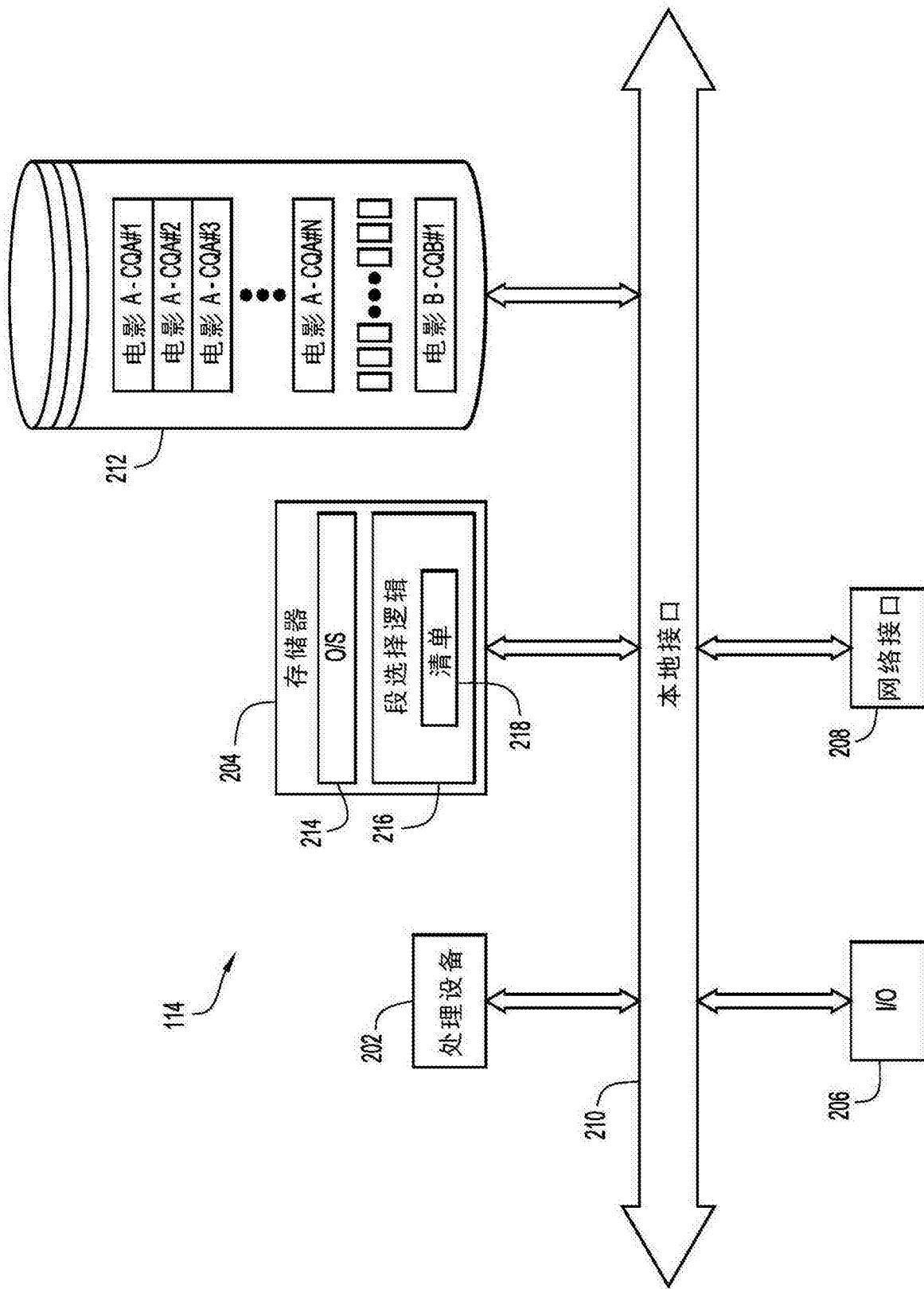


图 2

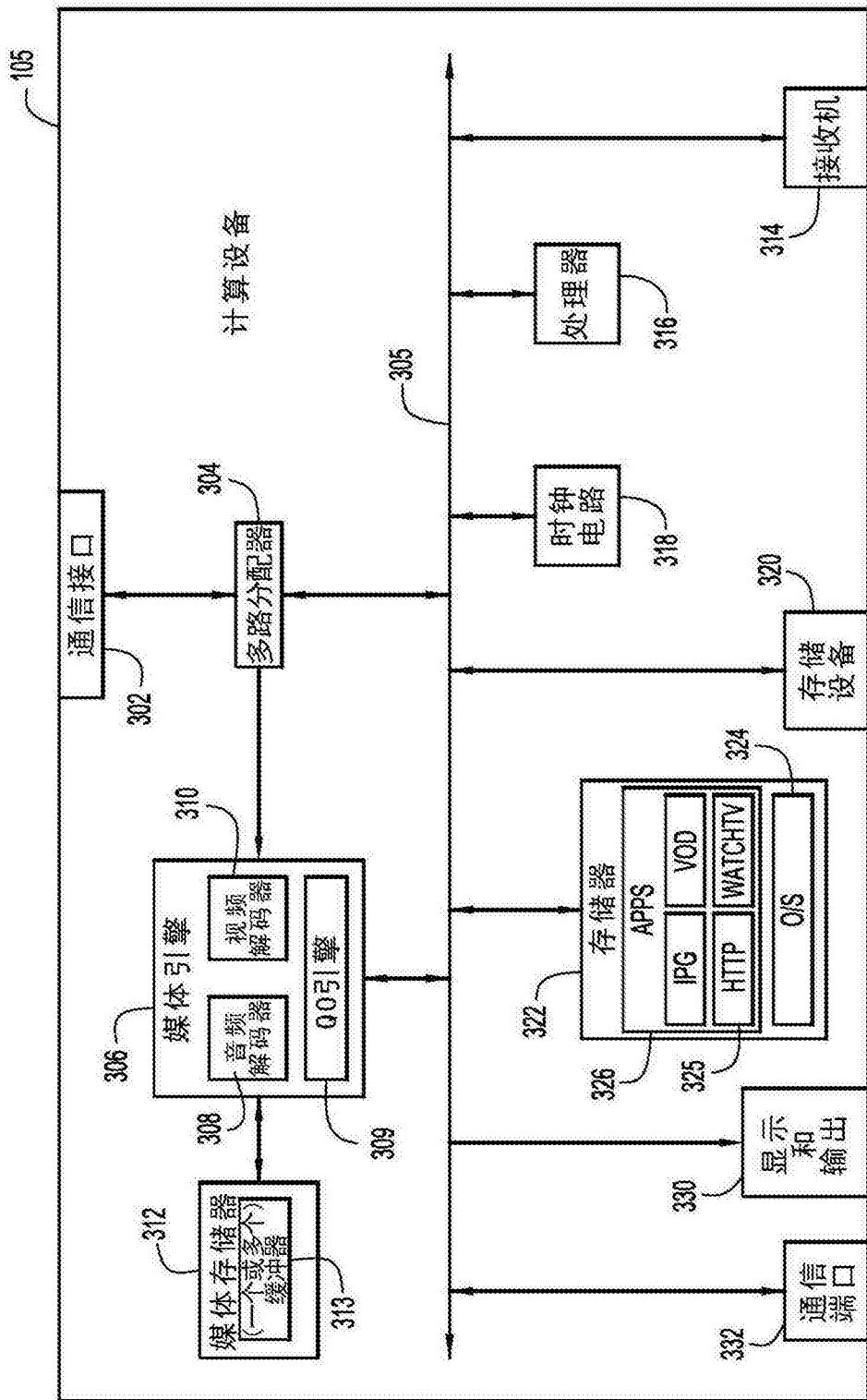


图 3

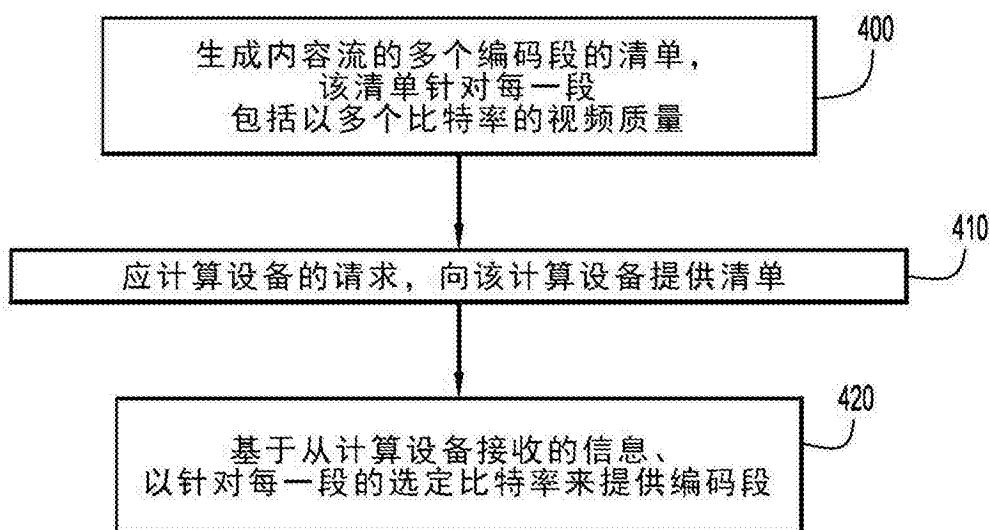


图 4

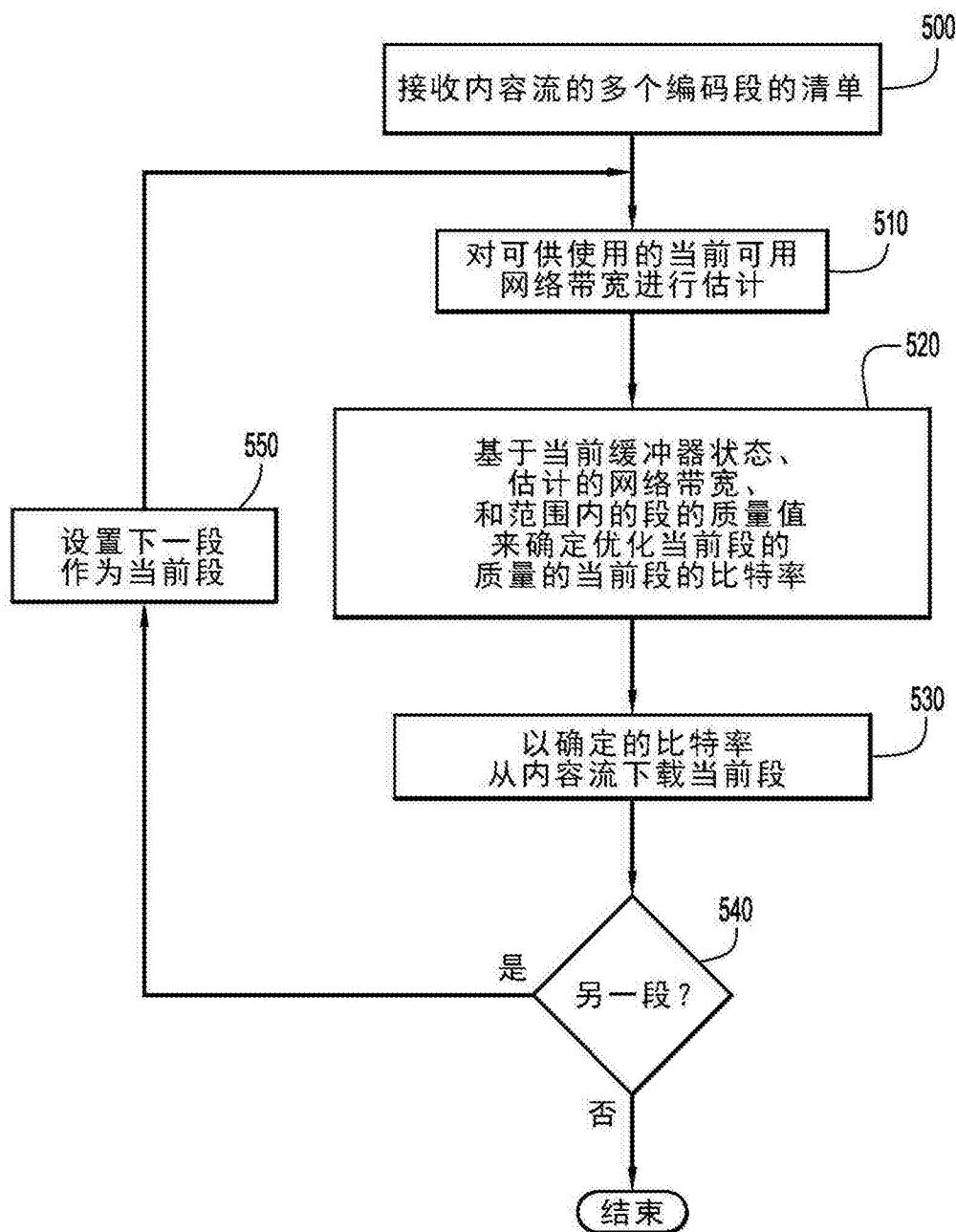


图 5

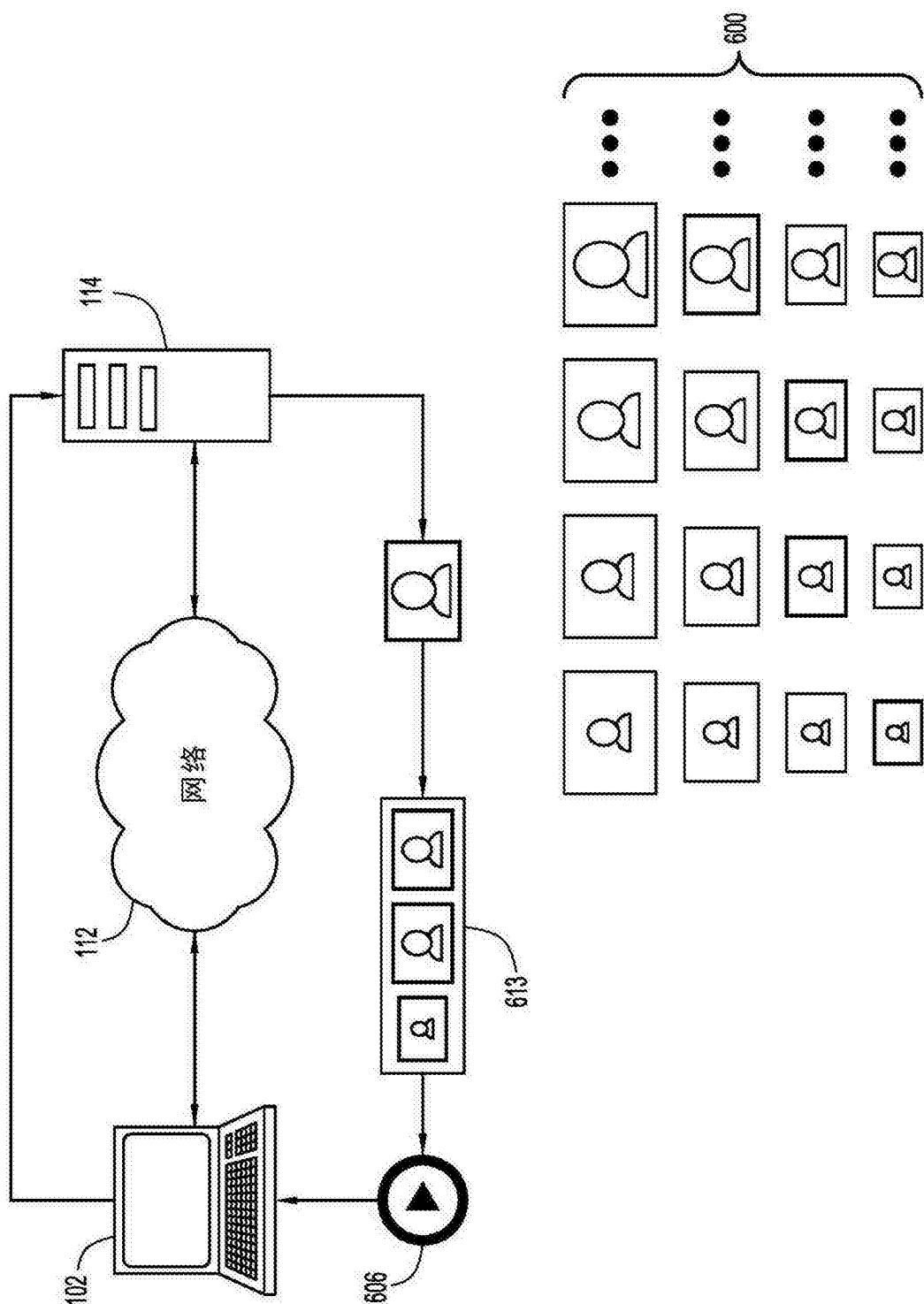


图 6