



(12) 发明专利申请

(10) 申请公布号 CN 117241076 A

(43) 申请公布日 2023. 12. 15

(21) 申请号 202311156924.6

(22) 申请日 2023.09.08

(71) 申请人 山东大学

地址 266200 山东省青岛市即墨区滨海路
72号

(72) 发明人 张广辉 王子铭 于东晓 成秀珍

(74) 专利代理机构 青岛华慧泽专利代理事务所
(普通合伙) 37247

专利代理师 刘娜

(51) Int. Cl.

H04N 21/2662 (2011.01)

H04N 21/2187 (2011.01)

H04N 21/2343 (2011.01)

H04N 21/24 (2011.01)

H04N 21/4402 (2011.01)

H04N 21/442 (2011.01)

G06V 20/40 (2022.01)

G06V 10/764 (2022.01)

G06V 10/774 (2022.01)

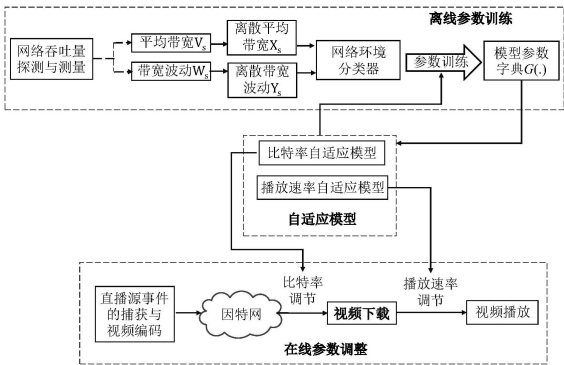
权利要求书3页 说明书8页 附图2页

(54) 发明名称

一种在高度波动网络下实现超低延迟视频直播的方法

(57) 摘要

本发明公开了一种在高度波动网络下实现超低延迟视频直播的方法,包括如下步骤:建立自适应模型;在虚拟视频流媒体环境下,将网络状态进行分类,针对不同类别的网络状态,利用建立的模型测试流传输性能,以流传输指标和实际播放延迟为目标函数,进行学习因子的优化训练,输出模型参数字典;视频直播过程中,获取当前网络带宽状态,根据模型参数字典得到当前最优学习因子,再将当前最优学习因子和设定的目标播放延迟代入模型,得到当前视频片段的播放速率和比特率,实现视频直播的在线动态调节。本发明所公开的方法通过对网络特征的学习,动态调整模型,从而实现稳定的播放延迟与较高的用户观看体验。



1. 一种在高度波动网络下实现超低延迟视频直播的方法,其特征在于,包括如下步骤:

步骤一,模型建立:建立以目标播放延迟和学习因子为自变量,以视频播放速率和比特率为因变量的自适应模型;

步骤二,离线参数训练:在虚拟视频流媒体环境下,将网络状态进行分类,针对不同类别的网络状态,利用建立的模型测试流传输性能,以流传输指标和实际播放延迟为目标函数,进行学习因子的优化训练,得到不同目标播放延迟和不同网络状态下对应的最优学习因子,输出模型参数字典;

步骤三,在线参数调整:视频直播过程中,获取当前网络带宽状态,结合设定的目标播放延迟根据模型参数字典得到当前最优学习因子,再将当前最优学习因子和设定的目标播放延迟代入模型,计算得到当前视频片段的播放速率和比特率,实现视频直播的在线动态调节。

2. 根据权利要求1所述的一种在高度波动网络下实现超低延迟视频直播的方法,其特征在于,步骤一建立的模型包括视频播放速率自适应模型和比特率自适应模型,具体如下:

视频播放速率自适应模型如下:

$$\theta_k = \begin{cases} 1 + \min\{g_k/\tau, \kappa_{\max}\}, & g_k > 0 \\ 1, & g_k = 0 \\ 1 + \max\{g_k/\tau, -\kappa_{\max}\}, & g_k < 0 \end{cases}$$

其中, θ_k 为视频播放速率, τ 是视频片段的物理时长,以秒为单位; κ_{\max} 为允许的最大播放速率变化, g_k 为最小化延迟偏差,计算公式如下:

$$g_k = l_k - \lambda_j$$

其中, λ_j 为设定的目标播放延迟, l_k 为实际播放延迟;

比特率自适应模型如下:

$$b_k = \beta \times c_k \times \frac{d_k}{\tau \times \theta_k}$$

其中, b_k 为视频片段k的比特率, c_k 是下载过去视频片段期间测得的平均网络带宽, d_k 是当前缓冲区占用率,以秒为单位; β 为学习因子,是一个可调参数,用于调节比特率选择的激进程度;

然后,将计算得到的比特率 b_k 映射到最接近的编码比特率级别:

$$h_k = \max\{h \mid \eta_h \leq b_k, h=0, 1, \dots, H-1\}$$

其中, η_h 表示第h级的编码比特率, h_k 则是最接近 b_k 的比特率级别,H表示编码比特率的级别总数;

最后,将视频片段k所选定的比特率级别 h_k 与片段k-1的比特率级别 h_{k-1} 进行差值比较,以将切换幅度限制在 ε 范围内:

$$h'_k = \begin{cases} h_{k-1} - \varepsilon, & \text{if } h_k < h_{k-1} - \varepsilon \\ h_{k-1} + \varepsilon, & \text{if } h_k > h_{k-1} + \varepsilon \\ h_k, & \text{otherwise} \end{cases}$$

其中, h'_k 为最终的比特率输出, ε 表示比特率的切换幅度最大阈值。

3. 根据权利要求1所述的一种在高度波动网络下实现超低延迟视频直播的方法, 其特征在于, 步骤二中, 离线参数训练的具体步骤如下:

步骤S1: 网络状态探测: 测量视频会话s的平均带宽 V_s 和带宽波动 W_s ;

步骤S2: 将测量的平均带宽 V_s 和带宽波动 W_s 分别线性映射为离散的网络状态 X_s 与 Y_s ;

$$X_s = \min(\lfloor V_s/\Delta_0 \rfloor, I-1); \quad Y_s = \min(\lfloor W_s/\Delta_1 \rfloor, J-1)$$

其中, X_s 为平均带宽网络状态, Y_s 为带宽波动网络状态, Δ_0, Δ_1 均为离散映射步长, I 为平均带宽的总状态数, J 是带宽波动的总状态数;

步骤S3: 将离散网络状态 X_s 与 Y_s 输入网络环境分类器, 实现对所有视频流会话的带宽数据的分类, 总共 $I \times J$ 个类别:

$$T_{i,j} = \{C_s | X_s = \langle i \rangle, Y_s = \langle j \rangle, \forall s\}, i = 0, 1, \dots, I-1, j = 0, 1, \dots, J-1$$

其中, $T_{i,j}$ 为每个类别下的带宽数据集, C_s 为传输视频会话s带宽数据, 其中 $s = 0, 1, \dots, S-1$, S 为视频会话总数;

步骤S4: 在不同网络类别 $T_{i,j}$ 下, 利用建立的模型分别进行参数训练, 测试不同网络类别下, 不同学习因子 β 以及不同目标播放延迟 λ_m 下的流传输性能, 以函数 $F(\cdot)$ 表示输出的目标函数:

$$\langle U_{m,i,j}, L_{m,i,j} \rangle = F(\beta, \lambda_m, T_{i,j})$$

其中, $U_{m,i,j}$ 为流传输指标, $L_{m,i,j}$ 为实际播放延迟;

步骤S5: 动态优化 β , 使 $U_{m,i,j}$ 最大化, 同时将 $L_{m,i,j}$ 限制在目标播放延迟 λ_m 范围内:

$$\max_{\beta} U_{m,i,j}, \quad s.t. \quad L_{m,i,j} \leq \lambda_m$$

步骤S6: 最终, 离线训练的输出为模型参数字典 $G(\cdot)$:

$$\beta_{m,i,j}^* = G(\lambda_m, X_i, Y_j)$$

模型参数字典 $G(\cdot)$ 包含三个键, 即目标播放延迟 λ_m 、平均带宽网络状态 X_i 和带宽波动网络状态 Y_j , 以及一个值, 即最优学习因子 β^* 。

4. 根据权利要求1所述的一种在高度波动网络下实现超低延迟视频直播的方法, 其特征在于, 步骤三中, 在线参数调整包括如下步骤:

步骤S1: 直播源事件通过摄像机捕获, 并在服务器端进行视频转码;

步骤S2: 直播视频通过互联网从服务器端到客户端进行传输, 并在传输过程中持续探测网络带宽状态;

步骤S3: 播放器通过在下载过程测量的网络带宽状态, 以及用户所指定的目标延迟输入模型参数字典 $G(\cdot)$ 中, 以获取最优学习因子 β^* , 并将其应用于自适应模型; 此步骤在视频下载过程中周期性执行;

步骤S4: 自适应模型进行播放速率和比特率的自适应调整。

5. 根据权利要求4所述的一种在高度波动网络下实现超低延迟视频直播的方法, 其特征在于, 步骤S3具体包括如下过程:

视频播放器于视频会话开始阶段, 即初始子会话中, 预加载多个视频片段, 其中预加载的总视频长度与目标播放延迟相同; 加载视频片段的过程中, 视频播放器将持续测量和记录用户的网络带宽, 以探测网络状况的变化;

在后续的子会话中,视频播放器首先通过在初始子会话中测量的网络带宽,并通过网络分类器确定网络类别;而后将网络类别索引以及用户所指定的目标播放延迟输入模型参数字典 $G(.)$ 中,以获取最优学习因子 β^* ,并将其下发于当前子会话,即将当前最优学习因子和设定的目标播放延迟代入模型,计算得到当前视频片段的播放速率和比特率,实现视频直播的在线动态调节;这一参数更新过程将在每个子会话的开始阶段重复进行,以持续追踪和检测网络条件的变化,直至视频下载结束。

一种在高度波动网络下实现超低延迟视频直播的方法

技术领域

[0001] 本发明涉及视频流媒体传输领域,特别涉及一种在高度波动网络下实现超低延迟视频直播的方法。

背景技术

[0002] 近年来,随着互联网带宽的大幅提升,视频直播服务得到了飞速的发展。针对视频直播技术的研究是当前业界所关注的热点科技问题,因为其对经济社会发展起到至关重要的影响。第一,直播服务在推动诸多民生工程建设方面发挥了重要作用,例如在线医疗、在线教育等方面;第二,直播服务所带动的新兴经济模式,例如电商直播、游戏直播等,已成为经济的新增长点;第三,传统直播模式,例如体育直播、新闻直播等,也融入了诸多新兴技术(例如虚拟现实,增强现实等),以改善用户观看体验。

[0003] 直播流媒体服务通常需要超低的视频播放延迟(秒级)。然而,在实际流媒体平台中,低延迟需求与提供高视频画质间存在内在冲突。例如,低延迟可以通过请求低比特率的视频内容实现,因为具有低比特率的视频块可以被迅速的下载和播放。然而,由于每个视频块只能在相应的直播源事件被捕获/渲染完成后才可传输,所以下载低比特率视频必然会导致大量带宽闲置,从而阻碍视频质量的改善(见图1);相反,下载高比特率的视频块可以更为充分地利用带宽资源,但与之相伴的高传输延迟会导致视频播放延迟的剧烈提升(见图2)。因此,从理论上讲,如果视频比特率能与网络的实时传输带宽完全匹配,则可同时获得低延迟和高视频质量的最优流传输性能。然而,在实际网络中,尤其是移动网络,带宽通常具有剧烈的波动性,因此在实践中实现该最优性能绝非易事。

[0004] 为此,近年来,基于DASH协议(Dynamic Adaptive Streaming over HTTP),业界提出了多种专门针对于低延迟视频直播的自适应流媒体算法,其通过动态调整视频比特率以尽最大限度地匹配网络带宽。然而,根据对当前最先进的流传输算法的测量研究表明,近乎所有算法的比特率自适应模型极易受网络条件变化的影响,即在高度波动的网络环境中,视频直播的播放延迟表现出较大的波动性,无法持续满足直播服务所需的质量需求。

[0005] 存在该问题的主要原因在于,现存算法通常采用一种普适于所有网络环境的单一通用自适应模型,使其难以在带宽敏感的直播流媒体环境中充分发挥性能潜力。例如,第一、当前基于启发式的算法通常采用单一的预编程逻辑,即按照固定的比特率控制规则进行在线决策。第二、虽然当前基于机器学习的算法会根据真实网络带宽数据训练自适应逻辑,但该类算法通常只生成一个单一的通用逻辑,并不会根据网络条件的变化进行动态逻辑更新。因此,当前所有类别的算法均无法很好的应对高度带宽波动的网络条件。

发明内容

[0006] 为解决上述技术问题,本发明提供了一种在高度波动网络下实现超低延迟视频直播的方法,以达到不仅可以实现对视频比特率的动态调整,还可以通过对网络特征的学习,动态调整模型,从而实现稳定的播放延迟与较高的用户观看体验的目的。

[0007] 为达到上述目的,本发明的技术方案如下:

[0008] 一种在高度波动网络下实现超低延迟视频直播的方法,包括如下步骤:

[0009] 步骤一,模型建立:建立以目标播放延迟和学习因子为自变量,以视频播放速率和比特率为因变量的自适应模型;

[0010] 步骤二,离线参数训练:在虚拟视频流媒体环境下,将网络状态进行分类,针对不同类型的网络状态,利用建立的模型测试流传输性能,以流传输指标和实际播放延迟为目标函数,进行学习因子的优化训练,得到不同目标播放延迟和不同网络状态下对应的最优学习因子,输出模型参数字典;

[0011] 步骤三,在线参数调整:视频直播过程中,获取当前网络带宽状态,结合设定的目标播放延迟根据模型参数字典得到当前最优学习因子,再将当前最优学习因子和设定的目标播放延迟代入模型,计算得到当前视频片段的播放速率和比特率,实现视频直播的在线动态调节。

[0012] 上述方案中,步骤一建立的模型包括视频播放速率自适应模型和比特率自适应模型,具体如下:

[0013] 视频播放速率自适应模型如下:

$$[0014] \quad \theta_k = \begin{cases} 1 + \min\{g_k/\tau, \kappa_{\max}\}, & g_k > 0 \\ 1, & g_k = 0 \\ 1 + \max\{g_k/\tau, -\kappa_{\max}\}, & g_k < 0 \end{cases}$$

[0015] 其中, θ_k 为视频播放速率, τ 是视频片段的物理时长,以秒为单位; κ_{\max} 为允许的最大播放速率变化, g_k 为最小化延迟偏差,计算公式如下:

$$[0016] \quad g_k = l_k - \lambda_j$$

[0017] 其中, λ_j 为设定的目标播放延迟, l_k 为实际播放延迟;

[0018] 比特率自适应模型如下:

$$[0019] \quad b_k = \beta \times c_k \times \frac{d_k}{\tau \times \theta_k}$$

[0020] 其中, b_k 为视频片段k的比特率, c_k 是下载过去视频片段期间测得的平均网络带宽, d_k 是当前缓冲区占用率,以秒为单位; β 为学习因子,是一个可调参数,用于调节比特率选择的激进程度;

[0021] 然后,将计算得到的比特率 b_k 映射到最接近的编码比特率级别:

$$[0022] \quad h_k = \max\{h \mid \eta_h \leq b_k, h=0, 1, \dots, H-1\}$$

[0023] 其中, η_h 表示第h级的编码比特率, h_k 则是最接近 b_k 的比特率级别,H表示编码比特率的级别总数;

[0024] 最后,将视频片段k所选定的比特率级别 h_k 与片段k-1的比特率级别 h_{k-1} 进行差值比较,以将切换幅度限制在 ε 范围内:

$$[0025] \quad h'_k = \begin{cases} h_{k-1} - \varepsilon, & \text{if } h_k < h_{k-1} - \varepsilon \\ h_{k-1} + \varepsilon, & \text{if } h_k > h_{k-1} + \varepsilon; \\ h_k, & \text{otherwise} \end{cases}$$

[0026] 其中, h_k' 为最终的比特率输出, ε 表示比特率的切换幅度最大阈值。

[0027] 上述方案中, 步骤二中, 离线参数训练的具体步骤如下:

[0028] 步骤S1: 网络状态探测: 测量视频会话s的平均带宽 V_s 和带宽波动 W_s ;

[0029] 步骤S2: 将测量的平均带宽 V_s 和带宽波动 W_s 分别线性映射为离散的网络状态 X_s 与 Y_s ;

$$[0030] \quad X_s = \min(\lfloor V_s / \Delta_0 \rfloor, I-1); \quad Y_s = \min(\lfloor W_s / \Delta_1 \rfloor, J-1)$$

[0031] 其中, X_s 为平均带宽网络状态, Y_s 为带宽波动网络状态, Δ_0, Δ_1 均为离散映射步长, I 为平均带宽的总状态数, J 是带宽波动的总状态数;

[0032] 步骤S3: 将离散网络状态 X_s 与 Y_s 输入网络环境分类器, 实现对所有视频流会话的带宽数据的分类, 总共 $I \times J$ 个类别:

$$[0033] \quad T_{i,j} = \{C_s | X_s = \langle i \rangle, Y_s = \langle j \rangle, \forall s\}, i = 0, 1, \dots, I-1, j = 0, 1, \dots, J-1$$

[0034] 其中, $T_{i,j}$ 为每个类别下的带宽数据集, C_s 为传输视频会话s带宽数据, 其中 $s = 0, 1, \dots, S-1$, S 为视频会话总数;

[0035] 步骤S4: 在不同网络类别 $T_{i,j}$ 下, 利用建立的模型分别进行参数训练, 测试不同网络类别下, 不同学习因子 β 以及不同目标播放延迟 λ_m 下的流传输性能, 以函数 $F(\cdot)$ 表示输出的目标函数:

$$[0036] \quad \langle U_{m,i,j}, L_{m,i,j} \rangle = F(\beta, \lambda_m, T_{i,j})$$

[0037] 其中, $U_{m,i,j}$ 为流传输指标, $L_{m,i,j}$ 为实际播放延迟 $L_{m,i,j}$;

[0038] 步骤S5: 动态优化 β , 使 $U_{m,i,j}$ 最大化, 同时将 $L_{m,i,j}$ 限制在目标播放延迟 λ_m 范围内:

$$[0039] \quad \max_{\beta} U_{m,i,j}, \quad s.t. \quad L_{m,i,j} \leq \lambda_m$$

[0040] 步骤S6: 最终, 离线训练的输出为模型参数字典 $G(\cdot)$:

$$[0041] \quad \beta_{m,i,j}^* = G(\lambda_m, X_i, Y_j)$$

[0042] 模型参数字典 $G(\cdot)$ 包含三个键, 即目标播放延迟 λ_m 、平均带宽网络状态 X_i 和带宽波动网络状态 Y_j , 以及一个值, 即最优学习因子 β^* 。

[0043] 上述方案中, 步骤三中, 在线参数调整包括如下步骤:

[0044] 步骤S1: 直播源事件通过摄像机捕获, 并在服务器端进行视频转码;

[0045] 步骤S2: 直播视频通过互联网从服务器端到客户端进行传输, 并在传输过程中持续探测网络带宽状态;

[0046] 步骤S3: 播放器通过在下载过程测量的网络带宽状态, 以及用户所指定的目标延迟输入模型参数字典 $G(\cdot)$ 中, 以获取最优学习因子 β^* , 并将其应用于自适应模型; 此步骤在视频下载过程中周期性执行;

[0047] 步骤S4: 自适应模型进行播放速率和比特率的自适应调整。

[0048] 进一步的技术方案中, 步骤S3具体包括如下过程:

[0049] 视频播放器于视频会话开始阶段, 即初始子会话中, 预加载多个视频片段, 其中预加载的总视频长度与目标播放延迟相同; 加载视频片段的过程中, 视频播放器将持续测量和记录用户的网络带宽, 以探测网络状况的变化;

[0050] 在后续的子会话中, 视频播放器首先通过在初始子会话中测量的网络带宽, 并通

过网络分类器确定网络类别;而后将网络类别索引以及用户所指定的目标播放延迟输入模型参数字典 $G(\cdot)$ 中,以获取最优学习因子 β^* ,并将其下发于当前子会话,即将当前最优学习因子和设定的目标播放延迟代入模型,计算得到当前视频片段的播放速率和比特率,实现视频直播的在线动态调节;这一参数更新过程将在每个子会话的开始阶段重复进行,以持续追踪和检测网络条件的变化,直至视频下载结束。

[0051] 通过上述技术方案,本发明提供的一种在高度波动网络下实现超低延迟视频直播的方法具有如下有益效果:

[0052] 本发明通过学习各种网络带宽状态,并针对不同的网络环境分别训练和优化自适应模型,并于在线流传输过程根据网络条件的实时变化动态调整模型参数。通过本系统,直播流媒体平台不仅可以实现对视频比特率的动态调整,还可以通过对网络特征的学习,动态调整模型本身,从而实现稳定的播放延迟与较高的用户观看体验。

[0053] 本发明在线流传输模块的实现只需对视频播放器做小幅度修改,大部分流传输工作流程保持不变。这为本发明在实际流媒体平台中的部署提供了极大的便利。

附图说明

[0054] 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。

[0055] 图1为在低比特率版本下直播视频片段下载示意图;

[0056] 图2为在高比特率版本下直播视频片段下载示意图;

[0057] 图3为在本发明实施例所公开的一种在高度波动网络下实现超低延迟视频直播的方法示意图;

[0058] 图4为基于子视频会话的在线模型参数调节时间序列示意图;

[0059] 图5为连续24小时内播放延迟变化的评估结果。

具体实施方式

[0060] 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。

[0061] 本发明提供了一种在高度波动网络下实现超低延迟视频直播的方法,如图3所示,包括如下步骤:

[0062] 步骤一,模型建立:建立以目标播放延迟和学习因子为自变量,以视频播放速率和比特率为因变量的自适应模型。

[0063] 建立的模型包括视频播放速率自适应模型和比特率自适应模型,具体如下:

[0064] 1、视频播放速率自适应模型

[0065] 当前诸多商用流媒体直播平台,例如Twitch、dash.js、Apple HLS.js等,均采用通过调节视频播放速率的方法控制播放延迟。播放速率被限制在一个较小的变化范围内(例如小于20%),使观众在观看视频时无法察觉速率的变化。因此,本发明的自适应模型也通过调节播放速率实现延迟控制。具体来说,在请求每个视频片段(例如 k)时,视频播放器将首先测量实际播放延迟 l_k ,然后计算其与目标延迟 λ_j 之间的偏差: $g_k = l_k - \lambda_j$,其中, λ_j 由用户输入,可以是任何带一位小数的正实数,例如1.5秒。

[0066] 本发明中播放速率自适应模型的最终目标是通过调整视频播放速率 θ_k 实现最小化延迟偏差 g_k ,因此总共需要考虑三种情况:

[0067] a) $g_k=0$,表示当前实际延迟已达到目标延迟,此情况下将保持正常的播放速率,即 $\theta_k=1$;

[0068] b) $g_k>0$,表示当前实际延迟大于目标延迟,则自适应模型将加快视频/音频播放,即 $\theta_k>1$,使播放追赶目标延迟;

[0069] c) $g_k<0$,表示当前实际延迟小于目标延迟,则自适应模型将调低播放速率,即 $\theta_k<1$,以缓冲更多视频数据,从而弥补网络带宽的波动,避免未来可能发生的播放卡顿。

[0070] 总体而言,播放速率自适应模型由下式表示:

$$[0071] \quad \theta_k = \begin{cases} 1 + \min\{g_k/\tau, \kappa_{\max}\}, & g_k > 0 \\ 1, & g_k = 0 \\ 1 + \max\{g_k/\tau, -\kappa_{\max}\}, & g_k < 0 \end{cases}$$

[0072] 其中, θ_k 为视频播放速率, τ 是视频片段的物理时长,以秒为单位; κ_{\max} 为允许的最大播放速率变化, g_k 为最小化延迟偏差,计算公式如下:

$$[0073] \quad g_k = l_k - \lambda_i$$

[0074] 其中, λ_i 为设定的目标播放延迟, l_k 为实际播放延迟。

[0075] 总体而言,本自适应模型的设计思路为,使每个视频段k在下载完毕时,实际延迟可以达到用户所设定的目标值。

[0076] 2、比特率自适应模型

[0077] 本发明的自适应模型模块在调节播放速率的同时,亦进行比特率调节。比特率自适应模型基于网络带宽测量、当前缓冲区水平以及视频播放速率的三输入参数共同确定最终的视频比特率。

[0078] 具体而言,在请求视频片段k时,假设 c_k 是下载过去w个视频片段期间测得的平均网络带宽, d_k 是当前缓冲区占用率(以秒为单位), θ_k 为由播放速率自适应模型所计算的视频播放速率。基于上述三个参数,自适应模型通过以下公式决策片段k的比特率(用 b_k 表示):

$$[0079] \quad b_k = \beta \times c_k \times \frac{d_k}{\tau \times \theta_k}$$

[0080] 其中, b_k 为视频片段k的比特率, c_k 是下载过去视频片段期间测得的平均网络带宽, d_k 是当前缓冲区占用率,以秒为单位; β 为学习因子,是一个可调参数,用于调节比特率选择的激进程度;

[0081] 在理想情况下,假定未来的网络带宽等于基于历史测量的评估值 c_k ,并使 $\beta=1$,则上述比特率自适应模型可转化为:

$$[0082] \quad (b_k \times \tau) / c_k \equiv d_k / \theta_k$$

[0083] 其中, τ 是视频片段的物理时长(以秒为单位)。该等式直接反应自适应模型的设计原理,即使当前缓冲区的所有视频数据在视频片段k下载完成时被完全消耗掉。基于此原理,视频画质则可实现合理的最大化,且不会触发任何播放卡顿。

[0084] 上述比特率自适应模型输出为比特率 b_k ,是一个连续值,但实际上,编码配置文件中的比特率级别是有限的离散值,因此需要,将计算得到的比特率 b_k 映射到最接近的编码

比特率级别：

[0085] $h_k = \max \{h | \eta_h \leq b_k, h=0, 1, \dots, H-1\}$

[0086] 其中, η_h 表示第 h 级的编码比特率, h_k 则是最接近 b_k 的比特率级别, H 表示编码比特率的级别总数;

[0087] 此外, 上述比特率自适应模型采用的是一种贪心策略, 即只关注当前视频片段比特率的最大化, 而不对未来进行任何长远规划。因此, 其所决策的比特率级别很可能在流传输过程中频繁切换, 从而导致视频画质的剧烈抖动与QoE的损失。为此, 本发明设计了一个限制比特率级别切换幅度的机制, 即将视频片段 k 所选定的比特率级别 h_k 与片段 $k-1$ 的比特率级别 h_{k-1} 进行差值比较, 以将切换幅度限制在 ε 范围内。具体实施方案为:

$$[0088] \quad h'_k = \begin{cases} h_{k-1} - \varepsilon, & \text{if } h_k < h_{k-1} - \varepsilon \\ h_{k-1} + \varepsilon, & \text{if } h_k > h_{k-1} + \varepsilon; \\ h_k, & \text{otherwise} \end{cases}$$

[0089] 其中, h'_k 为最终的比特率输出, ε 表示比特率的切换幅度最大阈值。

[0090] 步骤二, 离线参数训练: 在虚拟视频流媒体环境下, 将网络状态进行分类, 针对不同类别的网络状态, 利用建立的模型测试流传输性能, 以流传输指标和实际播放延迟为目标函数, 进行学习因子的优化训练, 得到不同目标播放延迟和不同网络状态下对应的最优学习因子, 输出模型参数字典。

[0091] 为了加快训练速度, 本发明采用虚拟流媒体仿真, 即使用在真实网络环境中收集的TCP带宽数据以复刻网络状态, 并执行流传输算法。上述自适应模型将被应用于虚拟视频流媒体环境, 实现在不同网络条件下参数的最优化训练。具体来说, 离线参数训练的具体步骤如下:

[0092] 步骤S1: 网络状态探测: 测量视频会话 s 的平均带宽 V_s 和带宽波动 W_s ;

[0093] 首先, 系统在流传输过程中实时测量视频流会话 s 的网络带宽, 并计算平均带宽, 记为 V_s , 带宽波动记为 W_s ;

[0094] 步骤S2: 将测量的平均带宽 V_s 和带宽波动 W_s 分别线性映射为离散的网络状态 X_s 与 Y_s ;

$$[0095] \quad X_s = \min(\lfloor V_s / \Delta_0 \rfloor, I-1); \quad Y_s = \min(\lfloor W_s / \Delta_1 \rfloor, J-1)$$

[0096] 其中, X_s 为平均带宽网络状态, Y_s 为带宽波动网络状态, Δ_0, Δ_1 均为离散映射步长, I 为平均带宽的总状态数, J 是带宽波动的总状态数;

[0097] 步骤S3: 将离散网络状态 X_s 与 Y_s 输入网络环境分类器, 实现对所有视频流会话的带宽数据的分类, 总共 $I \times J$ 个类别:

$$[0098] \quad T_{i,j} = \{C_s | X_s = \langle i \rangle, Y_s = \langle j \rangle, \forall s\}, i=0, 1, \dots, I-1, j=0, 1, \dots, J-1$$

[0099] 其中, $T_{i,j}$ 为每个类别下的带宽数据集, C_s 为传输视频会话 s 带宽数据, 其中 $s=0, 1, \dots, S-1$, S 为视频会话总数;

[0100] 步骤S4: 在不同网络类别 $T_{i,j}$ 下, 利用建立的模型分别进行参数训练, 测试不同网络类别下, 不同学习因子 β 以及不同目标播放延迟 λ_m 下的流传输性能, 以函数 $F(\cdot)$ 表示输出的目标函数:

[0101] $\langle U_{m,i,j}, L_{m,i,j} \rangle = F(\beta, \lambda_m, T_{i,j}), m=0,1,\dots,M-1, i=0,1,\dots,I-1, j=0,1,\dots,J-1$

[0102] 其中, $U_{m,i,j}$ 为流传输指标, 它不是固定不变的, 而可以由用户根据具体的服务需求进行手动配置, 旨在涵盖和满足任何用户所需的流传输性能, 如视频画质、播放卡顿、画质波动等。 $L_{m,i,j}$ 为实际播放延迟 $L_{m,i,j}$;

[0103] 步骤S5: 动态优化 β , 使 $U_{m,i,j}$ 最大化, 同时将 $L_{m,i,j}$ 限制在目标播放延迟 λ_m 范围内:

$$[0104] \quad \max_{\beta} U_{m,i,j}, \quad s.t. \quad L_{m,i,j} \leq \lambda_m$$

[0105] 上述优化问题可通过贝叶斯优化法求解, 整个过程离线执行。本发明的测试结果表明, 使用单核CPU可在5小时内求解出该问题。此外, 系统也可支持多核并行训练以降低时间花费。

[0106] 步骤S6: 最终, 离线训练的输出为模型参数字典 $G(\cdot)$:

$$[0107] \quad \beta_{m,i,j}^* = G(\lambda_m, X_i, Y_j)$$

[0108] 模型参数字典 $G(\cdot)$ 包含三个键, 即目标播放延迟 λ_m 、平均带宽网络状态 X_i 和带宽波动网络状态 Y_j , 以及一个值, 即最优学习因子 β^* 。

[0109] 步骤三, 在线参数调整: 视频直播过程中, 获取当前网络带宽状态, 结合设定的目标播放延迟根据模型参数字典得到当前最优学习因子, 再将当前最优学习因子和设定的目标播放延迟代入模型, 计算得到当前视频片段的播放速率和比特率, 实现视频直播的在线动态调节。

[0110] 离线训练的得到的模型参数字典 $G(\cdot)$ 将被应用于自适应模型, 并进而部署于在线流传输过程。 $G(\cdot)$ 将通过流媒体元数据 (例如DASH协议中的MPD播放列表) 被加载到视频客户端播放器中, 使播放器能够根据不断变化的网络条件定期更新自适应模型参数。为方便周期性参数更新, 本发明提出, 将一个视频会话划分为多个子会话, 其中每个子会话都具有固定的视频持续时间 (例如50秒)。视频播放器将持续检测和追踪流传输过程中网络条件的变化, 并在每个子会话开始阶段实施自适应模型参数的更新。

[0111] 在线参数调整具体包括如下步骤:

[0112] 步骤S1: 直播源事件通过摄像机捕获, 并在服务器端进行视频转码;

[0113] 步骤S2: 直播视频通过互联网从服务器端到客户端进行传输, 并在传输过程中持续探测网络带宽状态;

[0114] 步骤S3: 播放器通过在下载过程测量的网络带宽状态, 以及用户所指定的目标延迟输入模型参数字典 $G(\cdot)$ 中, 以获取最优学习因子 β^* , 并将其应用于自适应模型; 此步骤在视频下载过程中周期性执行;

[0115] 如图4所示, 视频播放器于视频会话开始阶段, 即初始子会话中, 预加载多个视频片段, 其中预加载的总视频长度与目标播放延迟相同; 加载视频片段的过程中, 视频播放器将持续测量和记录用户的网络带宽, 以探测网络状况的变化;

[0116] 在后续的子会话中, 视频播放器首先通过在初始子会话中测量的网络带宽, 并通过网络分类器确定网络类别; 而后将网络类别索引以及用户所指定的目标播放延迟输入模型参数字典 $G(\cdot)$ 中, 以获取最优学习因子 β^* , 并将其下发于当前子会话, 即将当前最优学习因子和设定的目标播放延迟代入模型, 计算得到当前视频片段的播放速率和比特率, 实现视频直播的在线动态调节; 这一参数更新过程将在每个子会话的开始阶段重复进行, 以持

续追踪和检测网络条件的变化,直至视频下载结束。

[0117] 步骤S4:自适应模型进行播放速率和比特率的自适应调整。

[0118] 参数被更新后的自适应模型将持续适应不断变化的网络环境,实现更加优化的比特率与播放速率自适应。

[0119] 总体而言,本发明在线流传输模块的实现只需对视频播放器做小幅度修改,大部分流传输工作流程保持不变。这为本系统在实际流媒体平台中的部署提供了极大的便利。

[0120] 性能评估

[0121] 本发明通过使用在一天内收集的TCP网络带宽数据进行连续24小时的实验评估并测量视频播放延迟变化(测量每小时的平均延迟),其中目标延迟被设定为1.5秒。为了解网络状况的变化,结果图中(即图5)绘制了每小时的网络带宽,其表现出剧烈的波动性,例如最低点为0.77Mbps,而最高点为5.99Mbps。结果如图5所示,在整个测试期间,本发明所提出的自学习系统实现了相对稳定的播放延迟,即始终环绕目标延迟1.5秒。这是因为本系统基于网络条件感知的自适应模型优化可以有效补偿网络带宽的剧烈变化,从而实现在长时间视频观看过程中提供稳定的服务质量。

[0122] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本发明。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实施例中实现。因此,本发明将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

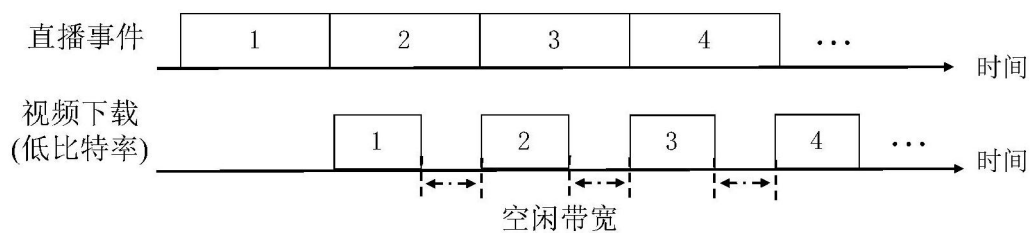


图1

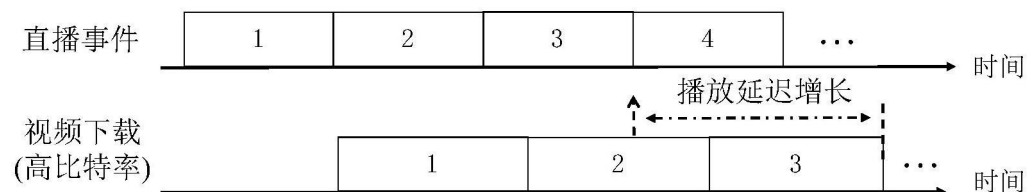


图2

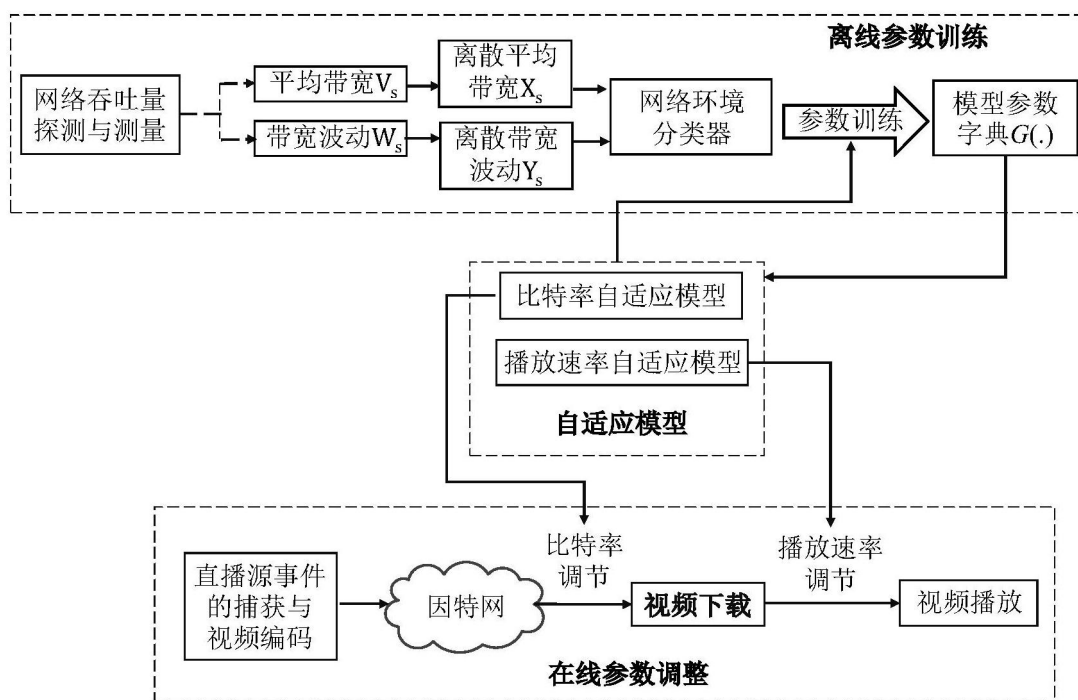


图3

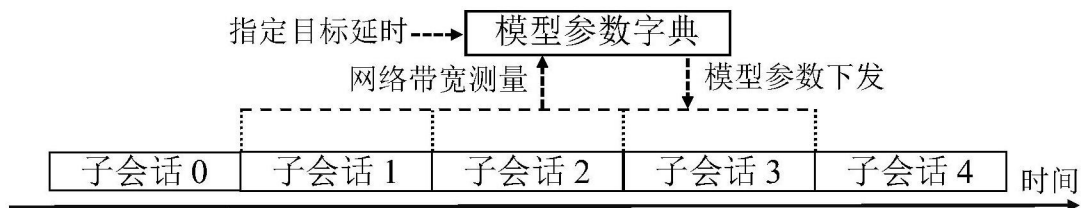


图4

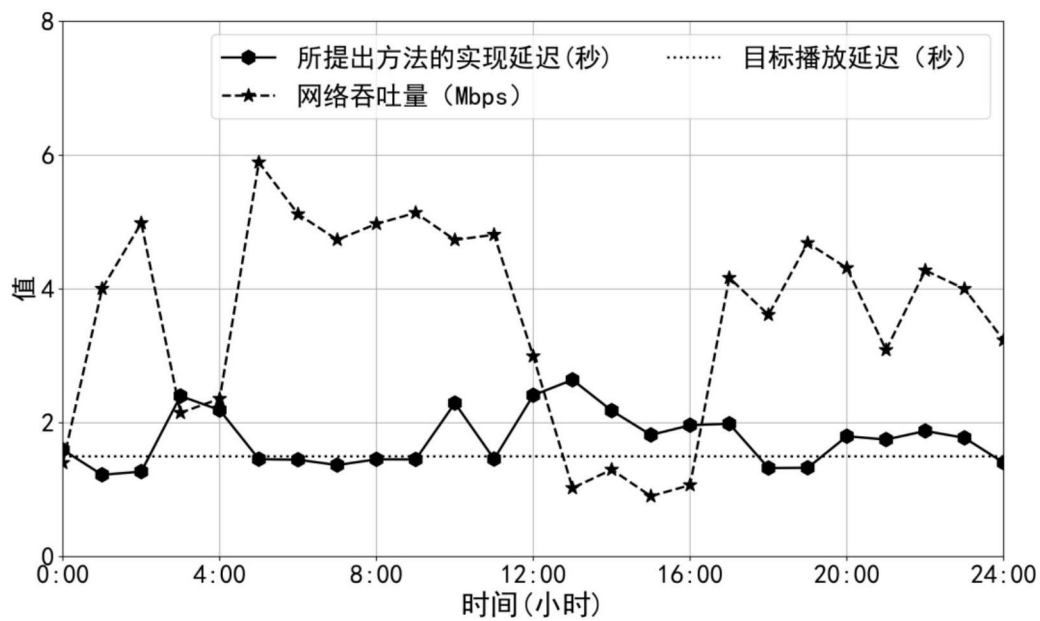


图5