



面向 ICN 的动态自适应流媒体算法

辛永辉,李 杨,林 涛

(中国科学院声学研究所 北京 100190)

摘 要:信息中心网络(information-centric networking, ICN)与传统 IP 网络的重要区别在于,前者的路由器节点具备缓存能力,能将热门内容推送到更靠近客户的网络边缘,从而降低客户获取内容的时延。然而 ICN 中的节点缓存特性会导致传统的自适应流媒体协议出现视频抖动甚至播放中断,降低客户的观看体验。提出了 ICN 中的动态自适应流媒体算法,在保证客户端播放流畅的同时,尽可能地提升视频片段的码率,提升客户的观看体验。通过在现有的 ICN 平台中进行仿真实验,证明了该自适应算法在性能上优于传统的算法。

关键词:自适应流媒体算法;信息中心网络;缓存

doi: 10.3969/j.issn.1000-0801.2014.09.006

Dynamic Adaptive Streaming Algorithm in ICN

Xin Yonghui, Li Yang, Lin Tao

(Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Information-centric networking (ICN) has different type of router node from the traditional network, which has caching facilities, and popular content will be pushed to the edge of the network closer to the user, reducing the delay of the user's access to the content. However, ICN's caching will result in the shake or even the interruption of video play when client uses the traditional adaptive streaming protocol, reducing the user's viewing experience. An algorithm for dynamic adaptive streaming in ICN was proposed, ensuring smoothly playback as well as improving the average bitrate of play thus to enhance the user's viewing experience. Simulation experiment on the existing ICN platform proves that the adaptive algorithm performs better than the traditional algorithms.

Key words: adaptive streaming algorithm, information-centric networking, caching

1 引言

当今互联网经历着从面向主机的点到点通信模式到以内容为中心的通信模式的转变,与内容位置相比,客户更关心的是内容本身。为了适应互联网的这一改变,业界提出了以 ICN (information-centric networking, 信息中心网络)^[1]为代表的下一代网络,该网络架构中的路由器具有缓存功能,以便后续为具有相同内容的请求提供服务。

目前多媒体业务,尤其是视频业务在整个互联网的数据流量中正占据越来越大的比重。研究显示,到 2016 年,视

频内容将占据全球移动数据流量的 55%以上^[2]。基于 HTTP 的动态自适应流媒体 (dynamic adaptive streaming over HTTP, DASH) 因具有部署简单、不受防火墙影响等优点而受到广泛关注^[3]。码率自适应是 DASH 的重要特点之一,但选择合适的码率自适应算法仍然是其研究的热点和重点。参考文献[4]提出了一种基于段抵达时间(segment fetch time)的码率自适应标准(SFTM),参考文献[5]也给出了两阶段式的自适应算法,它们都能有效地改善客户的观看体验。但简单地将传统的基于端到端的自适应算法移植到 ICN 中会出现一些问题,如参考文献[6]中所示 DASC(DASH



over CCN)。具体来讲,ICN的缓存会造成传统方法对网速的误判,导致客户端缓冲区下溢甚至中断,需要尽早判断下载速度提升的原因并及时进行处理。

本文提出了一种基于预估和中断机制的动态自适应流媒体算法。该算法在下一片段码率的选取上,可以使用现有的任何自适应算法,区别在于若在预估的时间内该码率的文件没有下载完成,就意味着码率过高,应该立即中断该码率的下载,并对码率进行下调,重新发出下载请求,这样可以尽早知道是否对网速进行了误判并进行下一步处理。

2 DASH在ICN应用的问题

本文在现有的ICN传输机制^[7]的基础上,通过修改DASH的VLC播放器的插件^[8],添加自适应流媒体算法。这里先简述ICN的传输机制和MPEG-DASH中媒体文件的自适应算法,然后在此基础上分析传统自适应算法在ICN应用中出现的问题。

2.1 ICN传输机制

ICN架构主要由具有缓存功能的ICN节点组成,这些节点也称为ICN路由器,具有路由和缓存两大功能,以目前备受关注的中心网络(content centric network,CCN)为例^[9]。内容的获取采用客户端驱动模式,网络中主要有两种数据:兴趣(Interest)分组以及内容(Data)分组,格式如图1所示。客户向邻近的路由器节点发出兴趣分组,若该路由器缓存中有对应的内容,则返回内容分组;若没有缓存,就按照一定的路由机制向外转发该兴趣分组,若收到数据分组则原路返回该数据。

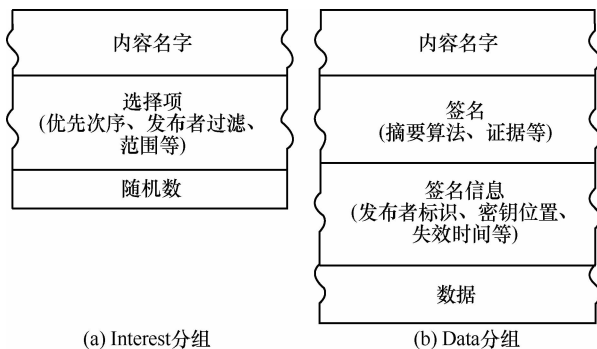


图1 ICN中的数据分组类型

2.2 MPEG-DASH基本模型

MPEG-DASH是一个基于HTTP的动态自适应流媒体标准,具有简单易用、适应多种网络环境、无需特殊代理等

优点。在实现中,媒体文件被分隔成若干个小片,并根据需要进行 N 种格式的编码,记录在XML格式的索引文件MPD(media presentation description)中。客户端要根据MPD文件向服务器发出请求,本文假设客户端均已获取MPD文件。

(1) MPEG-DASH中客户端与服务器的交互

分片文件存储在服务器上,客户端通过探查网络可用带宽并使用HTTP请求最优的码率文件,分片下载完成后,客户端就可以顺序播放。由于分片之间相互独立,没有间隔与重叠,故可以平滑播放。当网络带宽动态变化时,客户端采用自适应算法调整即将请求的码率文件来实现“自适应”。DASH客户端和服务器的简单交互流程如图2所示,具体介绍如下。

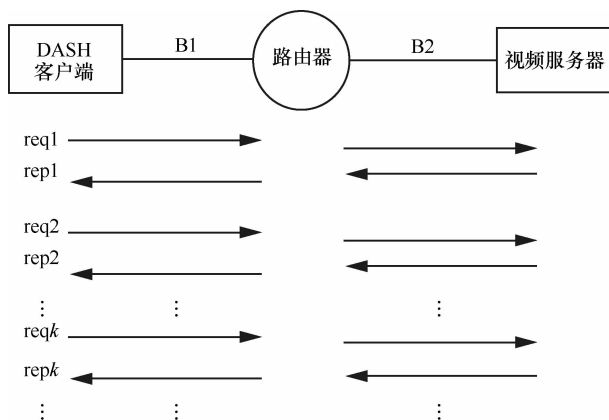


图2 DASH客户端和服务器的简单交互流程

步骤1 客户端根据已获得的MPD文件,发出对该视频分片为 S_1 、码率为 R_1 的请求req1,服务器通过rep1响应给客户端该文件。

步骤2 客户端收到rep1后,根据MPD文件,发出对分片为 S_2 、码率为 R_2 的请求req2,服务器做出相应回应。

步骤3 客户端收到rep2后,继续发出对片段为 S_k 、码率为 R_k 的请求reqk,然后收到服务器回复的repk,如此下去,直至观看结束或终止。

(2) 传统的自适应算法举例

Liu Y N等人在参考文献[6]中实现了MPEG-DASH在CCN中的原型,并采用rate-based自适应策略,即根据前一片段的下载速度估算整条链路的带宽,再从MPD文件中选取低于该估算带宽的最大码率进行下载。

假定分片的时长都为 τ ,码率为 R_k 的片段 S_k 请求开始的时间为 t_k^b ,其中 $t_1^b=0$,下载结束的时间为 t_k^e ,则片段 S_k

的下载速度为 $V_{R_k} = \tau R_k / (t_k^c - t_k^b)$, 片段 S_{k+1} 的请求码率 R_{k+1} 为 MPD 文件的码率集中低于 V_{R_k} 的最大码率。为保证较快的启动时延, 第一片的请求码率为 MPD 文件中的最低码率 $R_{\min}$ 。

Miller K 等人在参考文献[5]中提出了基于 HTTP 自适应流媒体的自适应 (adaptation adaptive streaming over HTTP, HAS) 算法, 该算法分为快速启动和正常状态两个阶段, 不同的阶段中下一片码率的选取、开始下载的时延需要根据最近一段时间 t 内下载的 n 个完整片段的平均速度以及当时整个缓冲区的长度而定。该算法为了保证码率调节的稳定性, 设置了不同的调节系数 $\alpha_1, \alpha_2, \dots, \alpha_5$, 以确保稳步调节, 减少码率的变动次数来提高客户的视觉体验。为保证快速启动, 第一片段的请求码率也是 MPD 文件中的 $R_{\min}$ 。

2.3 问题分析

rate-based 自适应策略只是简单地根据前一片段文件的下载速度, 确定下一片段的请求码率, 该方法能充分利用带宽, 紧跟网络变化并做出调整, 但当中间节点具有缓存功能时, 该算法自然会将缓存节点的下载网速误认为整条链路的网速, 导致发出对过高码率片段的请求并长时间等待, 这种现象根据缓存节点缓存内容的不同发生的次数也不同。如图 3 所示, 假设该视频的码率集为 $R = \{100 \text{ kbit/s}, 200 \text{ kbit/s}, 300 \text{ kbit/s}, \dots, 900 \text{ kbit/s}, 1000 \text{ kbit/s}\}$, 客户端到 ICN 路由器的局域网带宽为 $B_1 = 1000 \text{ kbit/s}$, ICN 路由器到视频服务器的外网带宽 $B_2 = 250 \text{ kbit/s}$ 。

客户第一次观看该视频, 请求最小码率为 100 kbit/s 的 S_1 , 下载时间为 1 s , 测得的网速为 200 kbit/s , 于是请求码率为 200 kbit/s 的 S_2 。经过一段足够长的时间 t , ICN 路由器会缓存 $S_1: 100 \text{ kbit/s}, S_2 \sim S_{10}: 200 \text{ kbit/s}$ 的文件; 客户第

二次观看时, 依然请求 100 kbit/s 的 S_1 , 由于 ICN 路由器的缓存, 客户端很快收到回复, 测得下载速度为 1000 kbit/s , 于是请求 1000 kbit/s 的 S_2 , 但 ICN 路由器没有缓存该文件, 于是向服务器转发, 客户端会在发出请求后 10 s 左右完成下载, 测得的网速又变为 200 kbit/s , 继而请求 200 kbit/s 的 S_3 , 如此进行下去。其中, 客户端获得的码率一直在 200 kbit/s 和 1000 kbit/s 中跳变, 客户观看效果很差; 同时下载 1000 kbit/s 将会耗时 10 s 左右, 这段时间内播放器很可能由于缓冲区下溢导致播放中断。

HAS 算法使用两阶段法, 能较好地对网络变化做出稳步调整, 但步骤比较繁琐, 涉及的调节系数较多, 而且是根据经验设定。在快速启动阶段, 选择低码率快速填充缓冲区, 网速误判影响较小, 但在正常阶段, 面对有缓存的网络也暴露出同样的缺点, 对于带宽误判只能长时间地等待。如图 4 所示, 同一个视频被请求多次后在 ICN 路由器中可能缓存有多种码率, 当客户端执行 HAS 自适应策略后, 经过一段时间 t 发出对 1000 kbit/s 片段 S_{10} 的请求, 而 ICN 路由器缓存的只有 900 kbit/s 的 S_{10} , 没缓存 1000 kbit/s , ICN 路由器不得不向外网转发, 客户端也因此被迫等待 10 s 左右, 在此期间播放器的缓冲区有可能因耗尽导致播放中断。

3 ICN 中的动态自适应算法

3.1 DASC 架构和网络拓扑

本文在 DASC 系统基础上提出适用于 ICN 的自适应流媒体算法^[7]。因为 ICN 还处于研究阶段, 如果能够融合 DASH 应用对于其发展有重要意义。图 5 展示了 DASC 的架构, 其中有阴影的代表原有的 DASH 模型, 无阴影的表

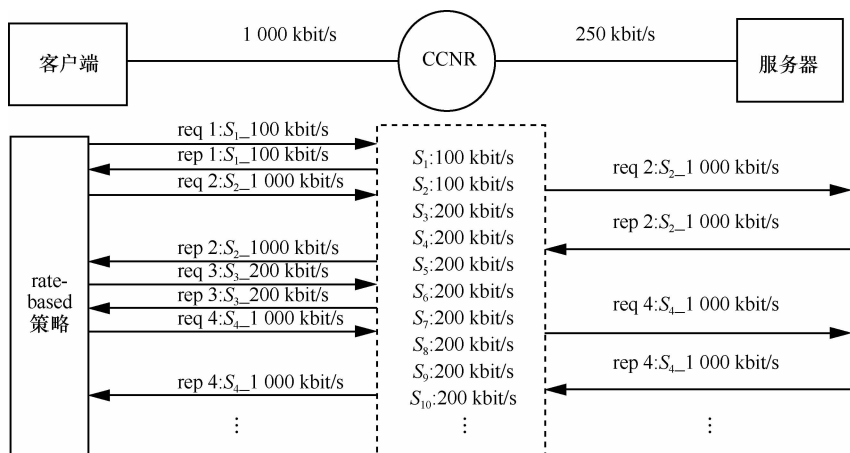


图3 rate-based 自适应策略示例

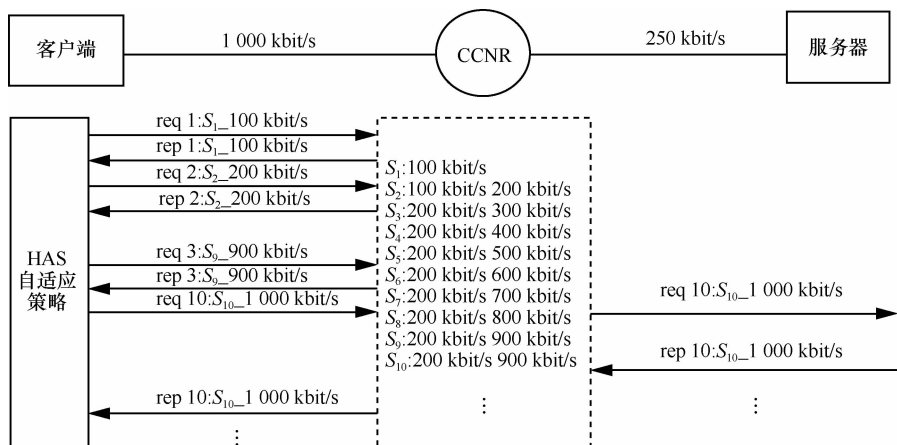


图4 HAS自适应算法示例

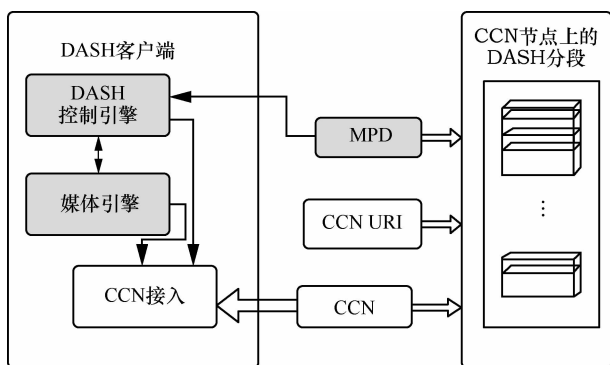


图5 DASH的架构

示CCN或者ICN部分。

3.2 ICN中的自适应算法

传统的自适应算法是基于下载的速度来估算当前的网速,在逐跳模式并且具有缓存的ICN中直接照搬极易误判网络带宽,从而引发一系列问题,为了尽量避免此问题,同时也尽可能地利用网络资源,提出了ICN中的动态自适应算法。

为了充分利用信道带宽并保证客户观看视频的流畅性和高质量,本文提出的自适应算法遵循以下4个原则:

- 尽量避免播放中断;
- 在带宽允许的情况下尽量提供高质量的视频片段,即平均视频质量和最小视频质量尽量大;
- 尽量使视频切换次数少,因为从客户体验出发,一直保持在低码率的视频比在高低码率间不断变化的视频带来的观看体验要好;
- 启动时延要小,即从客户选择播放某个视频到该视频第一个切片开始播放的时间要尽量短。

在这4个标准中,第1个标准应该具有最高的优先

级,因为造成客户体验最差的是视频播放的停滞或中断,也就意味着为了保证客户端视频质量的流畅性,在一定程度上可以牺牲视频的质量。第2个和第3个标准需要进行权衡,充分利用带宽的最好方法是选择目前网络支持的最高码率的视频,但在变化的网络中这会使得视频的切换次数大幅增加。同时第2个和第4个标准也需要进行折中,因为要降低启动时延就要求客户端下载第一个视频切片的时间最短,这样第一次只能请求最低的码率,在一定程度上造成了带宽的浪费,最终的判决过程是在这4个标准中权衡选择。

具体的自适应算法流程如图6所示,表1对涉及的主要符号进行了说明。

步骤1 开始时 $t=0, k=1$,为保证最小启动时延,请求第一片段 S_1 的最小码率 $R_1=R_{\min}$,并等待第 S_1R_1 下载完。

步骤2 下载完 S_k 后, $k=k+1$,可以计算出下载的网速

$$V(k,n) = (\sum_{i=1}^n TD(k-i+1))/n, \text{ 根据某种自适应算法(可以是}$$

RB或者HAS等)得出 R_k ,预估下载 S_k 的时间 $PTk = \alpha \cdot R_k \cdot \tau / V(k-1,n)$ 。

步骤3 发出对 S_k 的请求,同时设定倒计时计时器 $Tck=PTk$, S_k 下载完成或 $Tck=0$ 时中断计时。

步骤4 计时中断时若 $Tck>0$,说明可以进行下一片段的请求;否则说明未及时得到 S_kR_k ,为了防止长时间不必要的等待,需要提前下调待请求 S_k 的码率。

步骤5 下调 S_k 的码率可以分步进行,先下调到较小的 R_k' ,然后请求 S_kR_k' ,若 S_k 第二次发生中断,则需要立即下载到最低码率 $R_k=R_{\min}$ 。

该算法对获取下一片码率的算法没有限制,可以是传

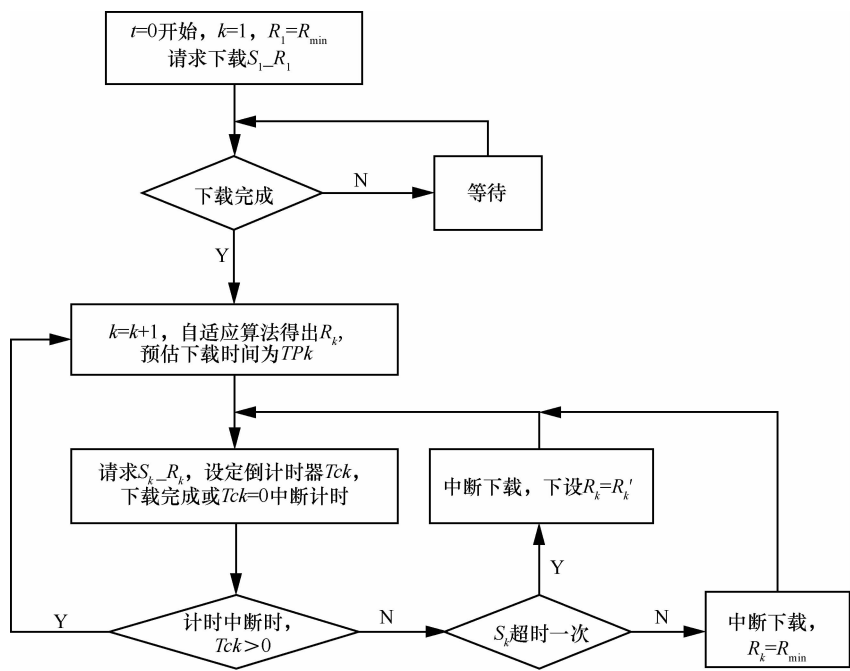


图 6 ICN 中动态自适应算法流程

表 1 主要符号

符号	含义
α	大于 1 的调节系数
S_k	请求视频的分片编号序号: $S_1, S_2, S_3, \dots$
R_k	片段 k 的请求码率
S_k-R_k	某一视频分段, 其分片编号为 S_k , 码率为 R_k
τ	分片时长, 默认分片时长为 2 s
TDk	片段 k 的真实下载时间
$R_{\min}$	MPD 文件码率集中的最小值
$R_{\max}$	MPD 文件码率集中的最大值
R'	码率集中小于 $R (R > R_{\min})$ 的一个码率
$V(k, n)$	片段 k (包括) 向前数 $n (n < k)$ 片的平均下载速度
TPk	片段 k 的预估下载时间

统的任意自适应算法, 如 rate-based 或者 HAS, 但在后续过程中新引入了预判中断机制和分步下调方法。具体如下。

(1) 预判中断机制

当已知前 n 个片段的下载速度时, 可以通过某种自适应算法给出即将下载片段 S_k 的码率 R_k 。这种基于下载的方式估算链路带宽的方法避免不了对网速的误判, 尤其是当网络中有缓存节点时。假设算出的下载速度为 $V(k, n)$, 根据自适应算法得到的分片码率为 R_k , 已知分片时长 τ 、片段 S_k 的预估下载时间。发出对该码率的请求后, 考虑到网络的未知性, 认为若能在预估时间内下载完成, 说明网络链路跟预期接近, 否则意味着链路状况比预期的糟糕, 或由于网络中间节点没缓存该高码率片段。为了防止长时

间的等待, 提前终止该请求并开始对该片段的低码率文件发出请求或许更有意义。

(2) 分步下调方法

中断后, 应该请求该片段低码率的文件, 但关于较低码率的设定和下调的次数没做具体的限定。考虑到客户观看视频的连续性, 即客户看完某个视频的片段 S_k , 很有可能也看了片段 S_{k+1} , 这样网络中缓存片段 S_{k+1} 的可能性非常大; 同样考虑到网络缓存的影响, 对于同一个视频, 后请求的客户观看的平均质量要比前面的好, 在下调片段 S_k 的码率 R_k 时, 可以充分利用网络中可能缓存的多种码率资源 (没有 R_k , 但很可能有比 R_k 低一个或者两个等级的码率 R'_k), 分步下调码率, 第一次下调为 $R_k=R'_k$, 请求下载。若该片段 S_k 不止一次发生中断, 为避免播放停滞, 应立即下调为 $R_k=R_{\min}$ 。

4 仿真结果与分析

本仿真中采用如图 7 所示的物理拓扑, 分为两个场景: 场景一中内网带宽为 100 Mbit/s, 与外网相连的出口带宽稳定在 250 kbit/s, 用以验证仿真的可行性; 场景二中内网带宽也为 100 Mbit/s, 而出口带宽为一个动态变化的网络, 用以测试本文提出的算法的准确性。实验中调节系数 $\alpha=1.5$, 所采用的视频片段总数为 300, 每个片段默认等时长 $\tau=2$ s, 且每个片段均有 200 kbit/s、250 kbit/s、300 kbit/s、400 kbit/s、500 kbit/s 共 5 种码率供选择。

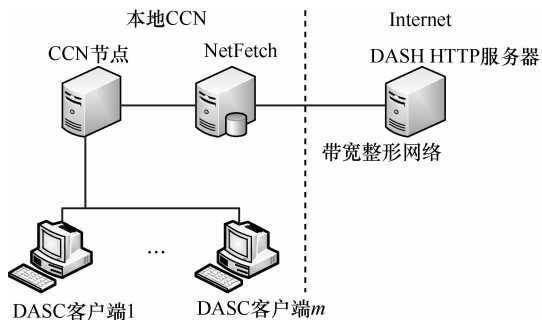


图 7 实验拓扑

场景一:图 8 展示了本地节点没缓存、有缓存时前两个客户的播放码率。图 9 给出了没有缓存以及有缓存时前 8 个客户端的码率比例。

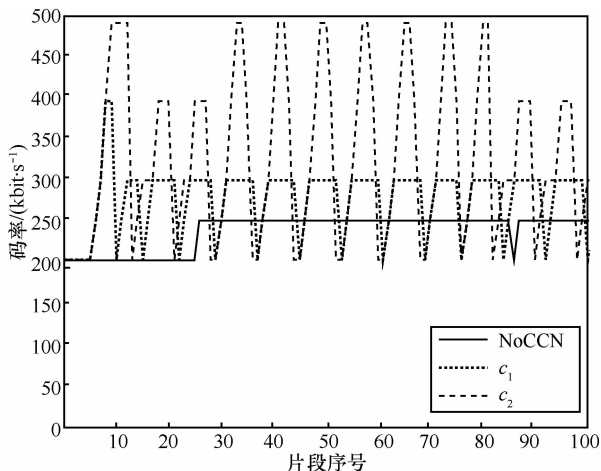


图 8 DASH/DASC 播放码率对比

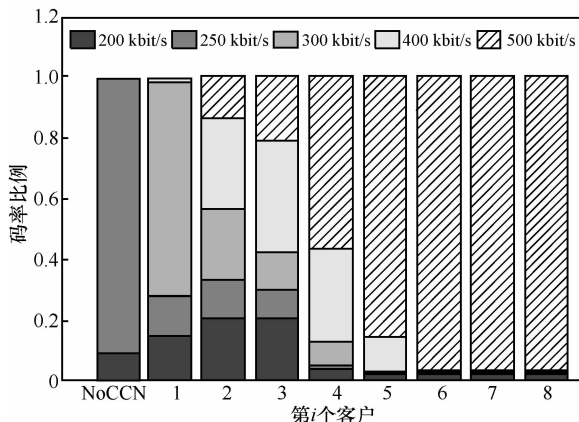


图 9 DASH/DASC 请求码率比例对比

对比图 8 和图 9, DASH 客户端请求了少量的 200 kbit/s 和较多的 250 kbit/s 码率的片段, 而 DASC 客户端 1 请求了部分 300 kbit/s 码率的片段, DASC 客户端 2 请求了部分 400 kbit/s 和 500 kbit/s 码率的片段; 而第 7 个和第 8 个客

户端请求的几乎是最高码率的片段。由于网络的缓存, 后续的 DASC 客户端可以利用邻近节点缓存的内容进而突破出口带宽小的限制。然而后续的一些客户端(第 2 个和第 3 个)却请求了较多的低码率(200 kbit/s)片段, 这主要是因为自适应策略对网速进行了误判, 误以为从缓存节点下载的速度为客户端到服务器整条链路的网速, 导致缓存区面临下溢的危险而被迫请求码率较低的片段。

场景二: 本次实验中, 客户端缓冲区默认设置 3 个缓冲区阈值, 当缓冲区长度小于缓冲区最小值 $Buffer_{min}=10$ s 时, 直接请求最低码率来填充; 尽可能地将缓冲区长度控制在低值 $Buffer_{low}=20$ s 和高值 $Buffer_{high}=50$ s 之间。动态网络的带宽变化如图 10 中粗实线所示, 基于已有的 HAS 算法, 本文提出的基于预判机制的自适应算法 PHAS 中也采用 HAS 方法, PHAS 的客户 1 和客户 2 的请求码率在图 10 中分别用两种虚线表示, 可见能够动态地适应网络的变化。图 11 给出了前 8 个客户端请求码率的比例, 由于缓冲

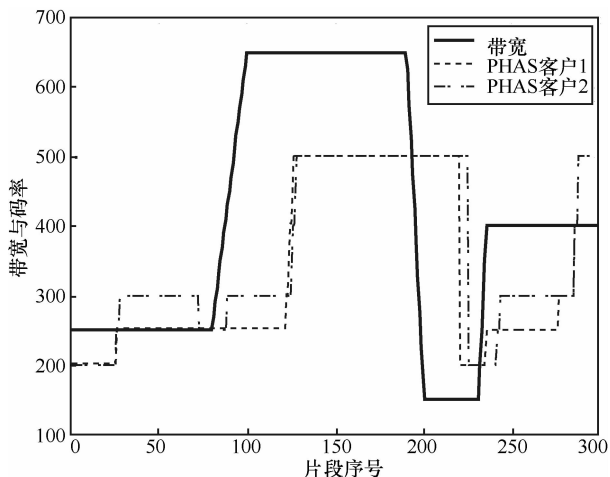


图 10 动态网络中请求码率的变化

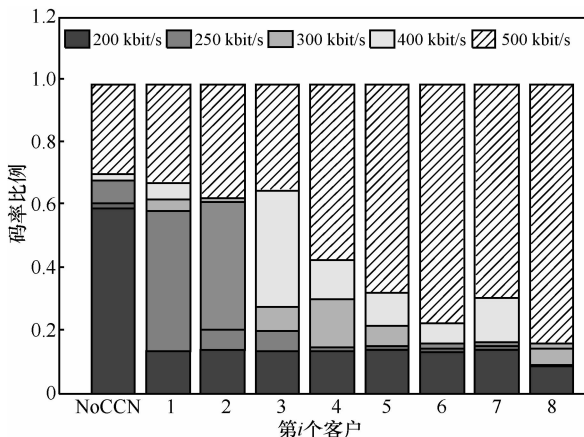


图 11 动态网络中不同客户端请求码率比例对比

区最小阈值的限制,每次开始请求的一些片段均为低码率,后续阶段对码率进行调整。

图12和图13给出了HAS和PHAS两种自适应算法的对比。在图12的缓冲区对比中,当出口带宽降低时,由于时延特性,HAS算法继续请求高码率的片段,由于这些高码率片段的邻近节点没有缓存,只能将请求转发至源服务器获取,客户端也只能等待请求的内容回来,尽管在此期间缓冲区已降到 $Buffer_{min}$ 附近,随时有播放中断的可能;而使用PHAS方法,可以清楚地看到缓冲区只是降到了 $Buffer_{low}$ 附近,大大降低了播放中断的可能性。图13中是请

求码率的对比,在出口带宽降低后,PHAS和HAS都做出了正确的判断,降低了码率,但PHAS算法比HAS算法更能充分地利用网络中缓存的资源,将码率尽快地调整回去,从而提高了客户的观看体验。

5 结束语

随着互联网视频、IPTV等业务的快速发展,动态自适应流媒体技术也得到了广泛研究。本文在现有ICN的基础上模拟了简单的MPEG-DASH应用,并通过仿真表明,改进的自适应算法要优于传统的自适应算法,能给客户更好

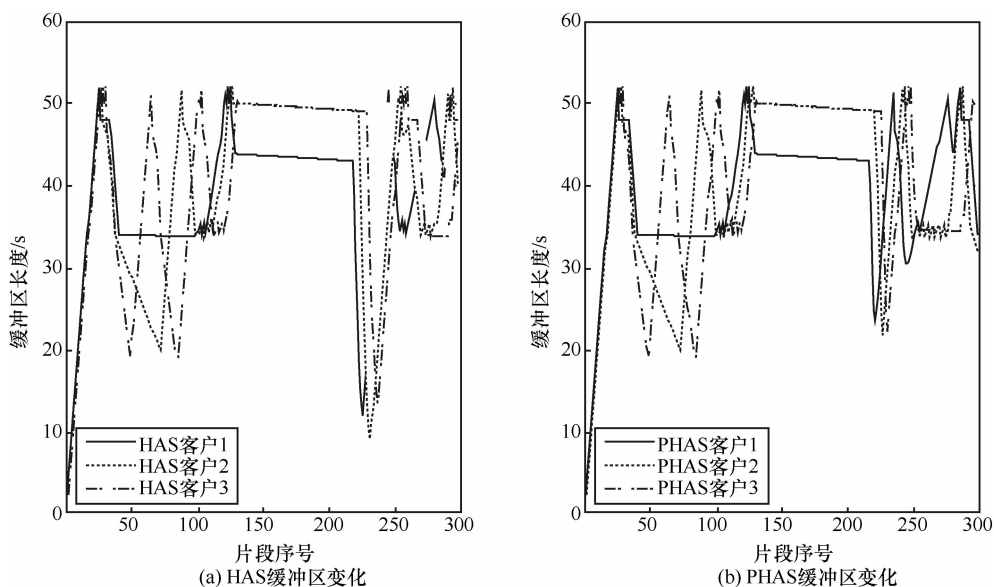


图12 动态网络中客户端缓冲区的变化

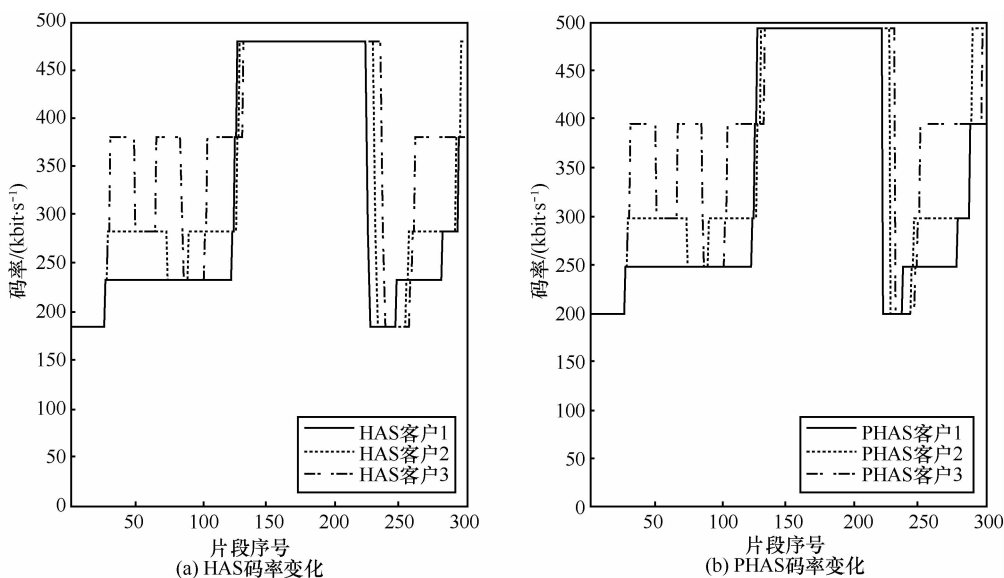


图13 动态网络中客户端请求码率的变化



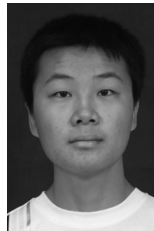
的观看体验。然而在 ICN 中部署 MPEG-DASH 存在的问题还有很多,还有很大的研究价值和应用发展空间,值得进行更深入的研究和应用开发工作。

参考文献

- 1 Ahlgren B, D'Ambrosio M, Marchisio M, *et al.* Design considerations for a network of information. Proceedings of the 2008 ACM CoNEXT Conference, Madrid, Spain, 2008
- 2 Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2011-2016. Tech Rep, May 2012
- 3 Sodagar I. The MPEG-DASH standard for multimedia streaming over the internet. IEEE Computer Society (S1070-986X), 2011, 4(18): 62~67
- 4 Akhshabi S, Narayanaswamy S, Begen A C, *et al.* An experimental evaluation of rate-adaptive video players over HTTP. Signal Processing: Image Communication (S0923-5965), 2012, 4(27): 271~287
- 5 Miller K, Quacchio E, Gennari G, *et al.* Adaptation algorithm for adaptive streaming over HTTP. Proceedings of 19th International Conference on Packet Video Workshop (PV), Munich-Garching, Germany, 2012: 173~178
- 6 Liu Y, Geurts J, Point J C, *et al.* Dynamic adaptive streaming over CCN: a caching and overhead analysis. Proceedings of 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 2013: 3629~3633
- 7 Ccnx project v0.6.0. <http://www.ccnx.org>
- 8 Müller C, Timmerer C. A VLC media player plugin enabling dynamic adaptive streaming over HTTP. Proceedings of the 19th ACM International Conference on Multimedia, New York, NY, USA, 2011: 723~726

- 9 Jacobson V, Smetters D, Thornton J, *et al.* Networking named content. Proceedings of the 2009 ACM CoNEXT Conference, Toronto, Canada, 2009

[作者简介]



辛永辉,男,中国科学院声学研究所博士生,主要研究方向为下一代互联网与未来网络、内容分发与缓存优化。



李杨,女,博士,中国科学院声学研究所助理研究员,主要研究方向为下一代互联网与未来网络、内容分发与缓存优化。



林涛,男,博士,中国科学院声学研究所副研究员,主要研究方向为未来网络、分布式内容缓存、可扩展路由。

(收稿日期:2014-07-20)

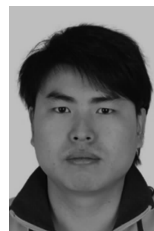
(上接第 17 页)

- 13 Qiao Y, Li T, Chen S G. One memory access bloom filters and their generalization. Proceedings of the 30th Annual IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, 2011
- 14 DJB hash function. <http://www.partow.net/programming/hashfunctions/#DJBHashFunction>, 2012
- 15 Wang Y, Dai H C, Jiang J C, *et al.* Parallel name lookup for named data networking. Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), Houston, Texas, United States, 2011
- 16 Wang Y, Tai D Z, Zhang T, *et al.* Greedy name lookup for named data networking. Proceedings of ACM SIGMETRICS, Pittsburgh, PA, 2013
- 17 Wang Y, Dai H C, Zhang T, *et al.* GPU-accelerated name lookup with component encoding. Elsevier Computer Networks, 2013, 57(16): 3165~3177

[作者简介]



刘斌,男,博士,清华大学计算机科学与技术系教授,2006 年国家杰出青年获得者,主要研究方向为高性能路由体系结构、未来互联网体系结构、软件定义网络和绿色互联网。



汪漪,男,博士,清华大学计算机科学与技术系助理研究员,主要研究方向为未来互联网体系结构、软件定义网络。

(收稿日期:2014-08-12)