

基于 HTTP 的流媒体自适应码率切换算法^{*}

潘鹏程 李国平 王国中

(上海大学通信与信息工程学院 上海 200444)

摘要: 基于 HTTP 的动态流媒体自适应(DASH)系统已经广泛地应用于互联网上的视频传输。针对客户端缓冲区受限和网络环境不稳定的情况,提出了一种新的基于缓冲区的码率控制算法。该算法将缓冲区分成不同等级,并采用滑动窗技术对视频的下载时间进行平滑处理,使缓存尽最大可能地维持在均衡区内。该算法在向上切换码率时采用保守策略,而在向下切换时则利用逻辑斯谛方程来决定码率切换的幅度。此外,该算法采用快速启动策略,能使系统开始时下载码率迅速匹配网络带宽。实验结果表明,算法能在复杂的网络环境中,充分利用带宽和缓冲区,降低码率的切换次数。

关键词: DASH; 滑动窗; 缓冲区; 网络带宽; 逻辑斯谛方程

中图分类号: TN919.85 **文献标识码:** A **国家标准学科分类代码:** 510.5025

Adaptive rate switch algorithm based on HTTP streaming

Pan Pengcheng Li Guoping Wang Guozhong

(School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China)

Abstract: Dynamic adaptive streaming over HTTP (DASH) has been widely used in the video transmission. With the limited client buffer and varying network, this paper proposes a new adaptive bitrate switching algorithm based on the client buffer. The proposed algorithm divides the client buffer into several buffer levels and uses sliding window to analyze the download time of recent video segments in order to maintain the media buffer size in the balance interval. The algorithm uses a conservative strategy when switching to bit rates upwards, and when switching down, the logistic equation is used to determine the magnitude of the bit rate switching. In addition, this paper also proposed a fast start strategy, in which download rate could match the network bandwidth quickly after the system starts. Experiment results show that the proposed algorithm can make full use of bandwidth and buffer, and reduce the number of rate switching in complex network environment.

Keywords: DASH; smooth window; buffer; network bandwidth; logistic equation

0 引言

近年来,随着信息技术与多媒体技术的发展,基于互联网的视频应用得到了快速的发展。目前,在欧洲方面,人们从多媒体内容网站 Youtube 和 Netflix 上的下载流量已占所有互联网下载流量的 30%^[1],而在北美则更为惊人地达到了 50%^[2]。人们对视频数量和质量上的强烈需求对多媒体网站在技术上提出了较高的要求。传统的流媒体技术主要是使用 RTSP(real time streaming protocol)和 RTP(real-time transport protocol)协议,它们都是基于 UDP 协议,在传输视频内容时容易出现丢包现象,并且当穿越网络地址转换(network address translation,NAT)和防火墙时需要额外的处理。此外,RTP 和 RTSP 需要专门的服

务器和专用的 CDN(content delivery network),这会带来高额的部署成本。基于 HTTP 的动态自适应流媒体技术采用 TCP 协议,利用 TCP 的“3 次握手”机制来保证连接的可靠性。另外,使用 HTTP 协议能轻松穿越 NAT 以及防火墙,同时还能保证使用普通的 Web 服务器就能完成部署^[3]。基于这些优势,目前,苹果、微软、Adobe 等互联网巨头都推出了自己的基于 HTTP 自适应流化方案,如微软的 IIS Smooth Streaming^[4],苹果的 LiveHTTPStreaming^[5]以及 Adobe 的 HTTP Dynamic Streaming^[6]等。2011 年,MPEG 组织发布了动态自适应流媒体技术国际标准 MPEG-DASH,此后在 2014 年 5 月推出 DASH 标准的第 2 版^[7]。

随着 MPEG-DASH 标准的推出,学术界也开始对

收稿日期:2017-02

^{*} 基金项目:国家自然科学基金(61271212)、国家“863”计划(2015AA015903)资助项目

DASH 技术进行研究。Sodagar^[8]介绍了 MPEG-DASH 标准的内部结构、自适应运作机制。Liu 等人^[9]利用分片时长和分片下载时间的比值作为码流自适应切换的度量,提出了一种码流切换的策略。张洋洋等人^[10]利用缓冲区下溢概率,并基于最大偏差原理估计当前网络带宽,进而选取最合适的码流进行下载。Chao 等人^[11]提出了固定区间缓存的码率算法即只有缓存不在固定区间时才引发码率调整,并且向上或向下调整码率都采用保守策略。Kalva 等人^[12]研究了可分级 MPEG 编码的情况下,自适应流的端到端传输开销情况。Akhshable 等人^[13]对目前几种常见的流媒体方案进行了测试,并指出了这些方案中存在的一些不足之处。

本文提出一种基于缓冲区的码率切换算法。该算法将缓冲区划分为不同的等级,并利用滑动窗分析窗内分片的下载时间。当进行码率切换时会根据当前的缓存状态和网络带宽情况采用不同的切换策略:当码率上调时采用保守策略逐级切换;当码率下调时根据逻辑斯谛方程决定切换的幅度;如果视频缓存低于重置阈值时,采用激进策略即立刻将码率切换至最小保证播放的流畅。此外,为了避免带宽的浪费,本文还设置了快速启动策略,在较短时间内保证下载片段的码率与网络带宽相匹配。两种算法的使用,能保证在视频流畅播放的基础上码率的变化幅度和频率都较小,使用户获得良好的体验。

1 自适应码率切换算法

1.1 系统模型

DASH 的系统架构如图 1 所示,它由一个普通的 Web 服务器和客户端组成。Web 服务器上的媒体资源注释模块包含了按种类分类的视频资源和按码率等级分类的视频资源。媒体资源预处理模块为视频数据提供了转码、打包、分组、封装等功能,因此视频资源可以高效的通过网络传输到客户端。元数据的内容来源于媒体资源注释模块,一旦客户端向服务器发送视频内容请求,服务器会立即做出回应,放置了视频各个码率的片段以及一份用来描述这些片段相关信息的文件 MPD (media presentation description)^[14]。客户端由下载模块、播放模块、码率控制模块组成。客户端工作时,首先向服务器请求下载 MPD 文件,得到可供选择的视频等级集,通过码率控制模块选择向服务器请求下载相应的视频片段。

1.2 算法目标

码率切换的目的,是为了用户能在网络带宽有限并且处于波动的情况下能有更好的视频观看体验。为此,可以对本文算法设置以下几个目标^[15]:

- 1) 不能允许播放过程出现停滞现象;
- 2) 视频播放的平均码率尽量要高,充分利用带宽;
- 3) 尽量减少码率切换次数;
- 4) 尽量将视频播放的启动时间缩短。

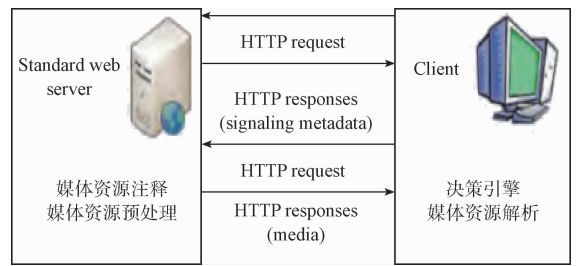


图 1 基于 HTTP 动态自适应流媒体架构

在上面的目标中,首先应该保证第 1 条,即要确保缓冲区中视频片段数量不能太少,当缓冲区中片段数量较少时,应采用激进的码率调整策略,使得缓存数据增加。第 2 条与第 3 条由码率切换算法决定,但这两条是一对矛盾。本文通过缓冲区分级以及采用滑动窗来降低码率突然变化而导致的码率切换,避免了频繁的码率调整。快速启动算法的引入确保了带宽能被充分利用。最后一条取决与算法如何最快获取到第一个视频片段,与码率切换无关。

1.3 本文提出算法

自适应算法中对于码率的切换涉及到网络带宽以及缓冲区状态。在缓冲区状态方面,当缓冲区状态良好时,视频播放才会流畅;反之,如果缓冲区状态不好,缓存片段较少,便会出现播放卡顿现象。而缓冲区的大小有两种表示方式,一种是以存储空间即 Byte 为单位,另一种是以时间长度即 s 为单位。在 DASH 标准中定义了一个片段固定的时长,而且现在的视频采用都是压缩编码,每一片的字节大小并不相同。

1) 将时间作为缓冲区的长度单位并将其划分不同的区域:重置区域、下溢区域,均衡区域、上溢区域,如图 2 所示。

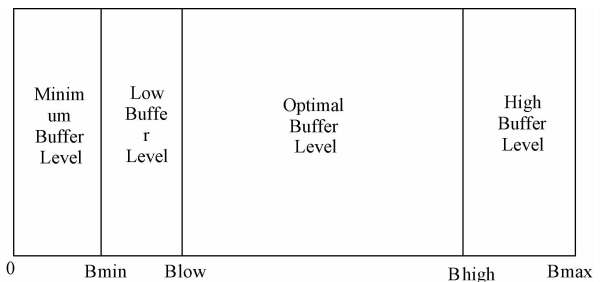


图 2 缓冲区状态模型

2) 在网络带宽方面,由于网络链路具有未知性,且自适应系统对网络又十分敏感,所以将单次传输的速率作为自适应传输算法的基本参数是不可靠的,但网络链路在一定的范围类会呈现一定的趋势,因此引入滑动窗口的概念,将以滑动窗为单位的统计结果作为当前网络状况的参考,如图 3 所示。

通过统计滑动窗 W 内视频片段的下载时间并利用截尾取均值法得到 T_w :

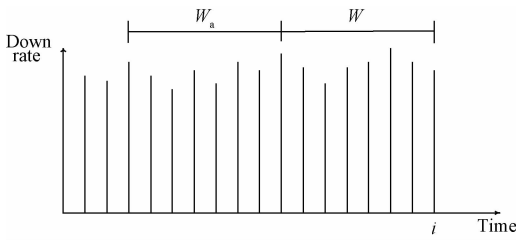


图 3 网络滑动窗示意图

$$T_w = \frac{\sum W - \max(W) - \min(W)}{n - 2} \quad (1)$$

将其作为当前视频片段的参考下载时间,其中 n 为滑动窗 W 内的视频片段数目, $\max(W)$ 为 W 内下载时间最大值, $\min(W)$ 表示 W 内下载时间最小值。引入流畅指数 Q 来反映当前的网络状况是否支持现有码率的下载。

$$Q = \frac{D}{T_w} \quad (2)$$

式中: D 为单个分片的固定时长。当 $Q > 1$ 时,说明当前网络状况良好;当 $Q < 1$ 时,说明网络状况变差,当前带宽不能支撑现有视频码率的下载,有视频播放卡顿的风险。

3) 当系统启动后,客户端为了减少视频播放的延迟,会选择最低码率的视频片段进行下载。由于最低码率视频的数据量较小,所以缓存将会迅速增加至均衡区域。本文算法的核心思想是将视频缓存维持在均衡区间内,尽可能地减少视频的码率切换。当网络状况较好时,下载的视频数据将会超过上溢阈值 B_{high} 进入上溢区间,为了充分利用网络带宽,当流畅指数 Q 满足式(3)时下一个视频片段的请求下载码率将会上调。

$$Q > 1 + \max\left(\frac{r_j + 1 - r_j}{r_j}\right) \quad j = 1, 2, \dots, N \quad (3)$$

式中: j 是当前视频片段请求下载的码率等级, r 为其对应的码率。

4) 在上调码率时采用单级别码率上调的保守策略即请求下一视频片段的码率 $r_j + 1$,该机制能减少因网络尖峰而引起的码率切换上的阶跃跳变。如果式(3)不能满足,则证明当前网络并不能满足下一等级码率视频下载所需的带宽,为避免频繁切换带来的用户体验的下降,算法维持当前码率不变。但是为了防止缓存上溢现象严重,客户端视频下载模块应睡眠一段时间,睡眠时间为:

$$T_s = (B_c - B_{high}) \times D \quad (4)$$

式中: B_c 表示当前视频缓存数目。

当网络链路发生拥堵,网络带宽下降时,缓冲区中的视频数量会逐渐减少,当缓存低于下溢阈值 B_{low} 时,为了避免出现视频播放停滞的现象,需要降低请求的视频码率。下调后的视频码率为:

$$R_{next} = \text{floor}\left(\frac{1}{1 + \mu} R_c\right) \quad (5)$$

式中: R_c 表示当前下载的视频码率, μ 为码率切换系数,决

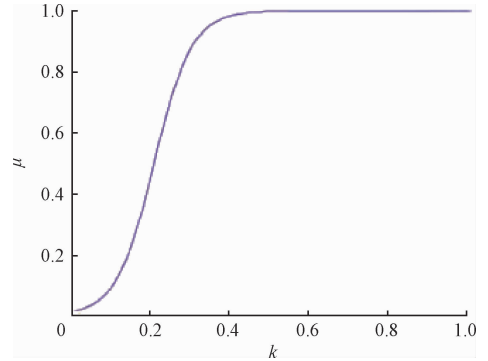
定了码率切换的幅度。本文使用逻辑斯谛方程模型来建立 μ 和 Q 之间的关系。

$$\mu = \frac{1}{1 + e^{-m(k - k_0)}} \quad (6)$$

式中: k_0 为逻辑斯谛方程的中心值, m 为方程的陡峭参数。 k 由网络阻塞参数由流畅指数 Q 和前一窗的流畅指数 Q_a 决定即:

$$k = \frac{Q - Q_a}{Q_a} \quad (7)$$

由于逻辑斯谛方程具有自跟随性即根据相邻窗之间的差距自动执行码率切换,能在码率切换中实现单次多级别的码率跳变,还拥有良好的归一性即在饱和阶段缓慢增长并趋于停止而这又很好的限制了码率跳变的幅度。如图 4 所示,当网络阻塞参数 k 增加至 0.25 后, μ 将会迅速上升,直到 μ 为 1 即码率调整幅度最大为当前码率的一半。

图 4 μ 与 k 的关系

而如果当网络状况继续变差,缓存降至重置区域时,立刻将视频码率切换至最低码率以防止视频播放出现停滞现象。如果当缓存处于重置区域或下溢区域时,网络状况变好,则维持当前码率,直至视频缓存升至均衡区。

5) 在上面的算法中有一个明显的缺点即在系统开始时,只有将均衡区填满时,算法才可能向高等级的码率进行切换。由于我们始终选择最低等级的码率作为初始码率,因此需要很长时间才能将下载码率切换到与网络带宽相匹配的等级。比如将均衡区的范围设置在 16~32 s,那么这就意味着缓存到达 32 s 时才能向高等级的码率进行切换,而一般网络带宽又是远远高于最低码率,这样便会造成网络带宽的极大浪费。因此,我们引入快速启动算法,能在系统开始时下载码率最快速地与网络带宽相匹配。

由于在系统刚开始时,视频下载数量并不能填满滑动窗,因此引入单个片段的流畅指数 P ,即:

$$P = \frac{D}{T_i} \quad (8)$$

当缓存位于重置区域时,将满足条件设置的严格一点即 $P > \alpha_1$,如果满足则将码率上调一级,不满足则维持当前码率。当缓存上升至下溢区域时,将条件适当放宽即设为 $P > \alpha_2$ ($\alpha_2 < \alpha_1$),如果此时网路状况满足条件则上调一级码

率,不满足这维持当前码率。上述条件能够使缓存还未至均衡区时将视频的下码率在较短时间内切换到与当前网络带宽相匹配的等级,有效的避免了网络带宽的浪费,提高了用户的体验。

2 实验结果与分析

本文系统的服务器端是在 Ubuntu14 上部署的 Apache 服务器,客户端使用基于 MPEG—DASH 标准的参考平台 libdash,使用 TC 进行流量。本文根据多次的实验结果,选取如下的参数配置:在服务器端,视频资源会被预先切分为 300 个切片,每个切片分为 13 个等级,即 $R = \{100, 200, 350, 500, 700, 900, 1100, 1600, 2300, 2800, 3400, 4500, 6400\}$,分片周期 T 为 2 s;在客户端,设置一个长度为 40 s 的视频缓冲区,均衡区的上限 B_{high} 设置为 32 s,下限 B_{low} 设置为 16 s,重置阈值设置为 8 s,滑动窗口长度设置为 16 s,逻辑斯谛方程中的陡峭参数设为 21,中心值 k_0 取值是 0.25,快速启动策略中 α_1 设为 2, α_2 设为 1.5。将模拟不同的网络状态并与文献[9]中的 SFT 算法进行比较。

第一种是在稳定持久的网络带宽变化下即将初始网络带宽设置为 600 Kbit/s,并将此带宽维持 100 s,然后将网络带宽变为 1 600 Kbit/s,同样将其持续 100 s,然后再将带宽变回 600 Kbit/s 持续 100 s,实验结果如图 5 所示。

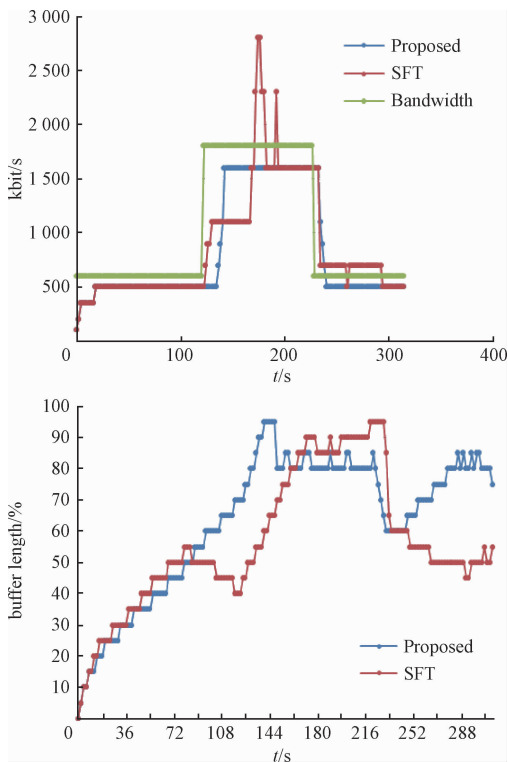


图 5 稳定带宽时码率和缓冲状态对比

从图 5 中可以看到,在开始的前 100 s,两种算法的码率相差不多,基本与当前带宽相符合,缓存也在逐步增加;

但当带宽增加后,SFT 算法便开始出现码率突变产生的阶跃跳变现象甚至会选择比当前带宽大很多的码率并且缓存也出现了先下滑后在上升的现象,而本文算法的码率变化则是平滑上升,波形平整并且始终保证所选择码率低于带宽同时缓存一直保持着较高的水平。

在 200 s 后,两种算法都随着带宽的下降而下降,但 SFT 算法并没有经过中间的码率的过度,并且还出现了码率的波动。在缓冲区方面视频数量也在不断减少。而本文算法则是逐级下降并在降至低于带宽的码率后能保持码率不变,在缓存方面则是在保证码率平稳切换的前提下出现了暂时的下降,但当播放码率稳定后的便开始上升并将其保持在均衡区内。

第二种情况是短时间内发生大幅度的带宽波动即将带宽设置为 1 200 Kbit/s,在 120 s 时将其变为 700 Kbit/s 持续 6 s 后恢复为 1 200 Kbit/s,在 210 s 时将带宽设置为 1 700 Kbit/s 持续 6 s 后变为 1 200 Kbit/s,实验结果如图 6 所示。从图 6 中可以看出,本文算法在面对短时间的码率剧变情况时,保持码率的稳定,而 SFT 算法在带宽发生剧变时会码率选择出现较大幅度的波动,并且本身还存在码率抖动的毛刺现象。在缓存方面,前期 SFT 算法缓存上升较快,但到 120 s 之后,两者之间相差不多。

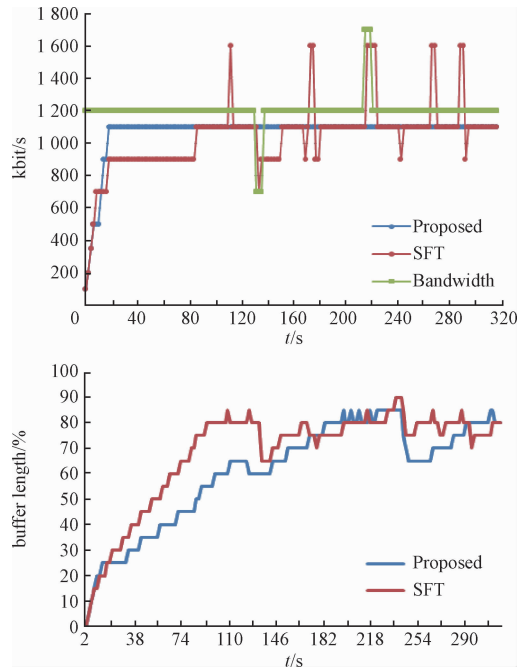


图 6 短时大幅度带宽波动时码率和缓冲状态对比

此外,还在实际网络环境中比较两种算法的表现。在实验室的电脑上运行客户端并向 DASH 标准组织的服务器上请求视频资源,实验结果如图 7 所示。从图 7 中可以看出,本文算法的码率在开始时是逐级上升,SFT 算法存在码率跳变现象,并且在视频播放的中期 SFT 算法出现了码率的波动。而本文算法则成功地避免了码率的切换,带

来了良好的用户体验。另外,在缓冲区方面,本文算法的缓冲区利用率要高于 SFT 算法,这也成为算法抵御网络带宽的波动的重要条件。

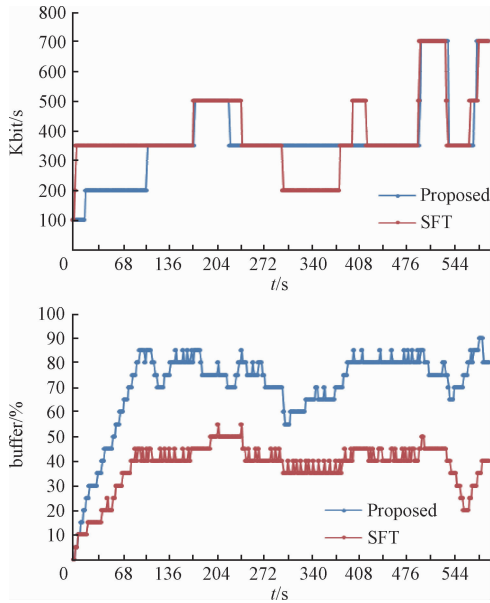


图 7 实际网络环境中码率和缓冲状态对比

4 结 论

本文首先对 DASH 技术以及 MPEG-DASH 标准进行了简单的介绍,然后提出了基于缓冲区的自适应码率控制算法,其目标是将视频缓存数量保持在均衡区内,同时尽可能地减少视频码率的切换。当缓冲中视频数量超出或低于均衡区的上限或下限时便会结合网络带宽状况进行相应的调整。另外,为了能在系统开始时视频片段下载码率快速匹配网络带宽,本文有提出了快速启动策略。实验结果表明,本文提出的算法可以在保证视频清晰度最优即充分利用带宽的情况下,减少码率的切换次数,提高缓存的利用率,从而带来较好的用户体验。

参考文献

- [1] Sandvine Inc. Global internet phenomena, latin america and north america[R]. 2015.
- [2] Sandvine Inc. Global internet phenomena, asia-pacific and europe[R]. 2015.
- [3] 朱秀昌, 陈杰. 基于 HTTP 的视频流网络传输[J]. 南京邮电大学学报:自然科学版, 2016, 36(3):1-9.
- [4] ZAMBELLI A. Smooth streaming technical overview [EB/OL]. (2009-03-31) [2011-12-01]. <http://www.iis.net/download/smoothstreaming>.
- [5] MULLER C, LEDERER S, TIMMERER C. An evaluation of dynamic adaptive streaming over http in

vehicular environments [C]. Proceedings of the 4th Workshop on Mobile Video, 2012: 37-42.

- [6] OYMAN O, SINGH S. Quality of experience for HTTP adaptive streaming services [J]. Communications Magazine, 2012, 50(4): 20-27.
- [7] ISO/IEC. Information technology-dynamic adaptive streaming over HTTP (DASH)-Part 1: media presentation description and segment formats[S]. 2014.
- [8] SODAGAR I. The MPEG-DASH standard for multimedia streaming over the internet [J]. IEEE Computer Society, 2011, 18(3):62-67.
- [9] LIU C, BOUAZIZI I, GABBOUJ M. Rate adaptation for adaptive HTTP streaming[C]. ACM Conference on Multimedia Systems, ACM, 2011:169-174.
- [10] 张洋洋, 陈双武, 完颜许哲, 等. 基于 HTTP 的自适应视频传输算法[J]. 小型微型计算机系统, 2016, 37(7):1572-1576.
- [11] CAO Y, YOU X, WANG J, et al. A QoE friendly rate adaptation method for DASH [C]. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2014:1-6.
- [12] KALVA H, ADZIC V, FURHT B. Comparing MPEG AVC and SVC for adaptive HTTP streaming[C]. IEEE International Conference on Consumer Electronics, 2012:158-159.
- [13] AKHSHABE S, BEGEN A C, DOVROLIS C. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP[C]. ACM Sigm Conference on Multimedia Systems, 2011:157-168.
- [14] STOCKHAMMER T. Dynamic adaptive streaming over HTTP: Standards and design principles [C]. ACM Sigm Conference on Multimedia Systems, 2011:133-144.
- [15] MILLER K, QUACCHIO E, GENNARI G, et al. Adaptation algorithm for adaptive streaming over HTTP[C]. Packet Video Workshop, IEEE, 2012: 173-178.

作者简介

潘鹏程,1992 年出生,硕士,主要研究方向为动态自适应流媒体传输。

E-mail:phonor@shu.edu.cn

李国平,1974 年出生,博士,高级工程师,主要研究方向为数字音视频编解码技术、网络传输技术。

王国中,1962 年出生,博士,教授,主要研究方向为数字音视频编解码技术、多媒体通信。