A CONTROL-THEORETIC APPROACH TO RATE ADAPTATION FOR DYNAMIC HTTP STREAMING

Chao Zhou[+], *Xinggong Zhang*[+*], *Longshe Huo*[#], *Zongming Guo*[+]

[+]Institute of Computer Science & Technology, Peking University, Beijing, P.R. China
Email: {zhouchaoyf, zhangxg, guozongming}@pku.edu.cn
[#]China Unicom Research Institute, Beijing, P.R. China
Email: huols@chinaunicom.cn

ABSTRACT

Recently, dynamic adaptive HTTP streaming has been widely used for video content delivery over Internet. However, it is still a challenge how to switch video bitrate under time-varying bandwidth. In this paper, we propose a novel control-theoretic approach to adapt video segments in dynamic HTTP streaming. The rate control is based on a sink-buffer, which has an overflow-threshold and an underflow-threshold. The objective is to maximize the playback quality while keeping the receiver buffer from either overflow or underflow. Using control theory, we formulate this rate control scheme as a proportional (*P*) control system, which exists oscillations and steady-errors. Furthermore, we design a proportional derivative (*PD*) controller to improve its adaptation performance. The conditions for stability and settling time of the *PD* controller are also derived. Numerous experiment results demonstrate the effectiveness of our proposed *PD* control scheme for dynamic HTTP streaming.

Index Terms— Dynamic HTTP Streaming, Proportional Derivative Controller, Rate Adaptation

1. INTRODUCTION

In recent years, HTTP- based method has been widely used for the delivery of video content over the Internet[1, 2]. Contrary to the past RTP/UDP, the use of HTTP over TCP is easy to configure and in particular, it greatly simplifies the traversal of firewalls and network address translators (NAT). Besides, it is cheap to be deployed since it employs standard HTTP servers and it also can be easily deployed within Content Delivery Networks (CDN). HTTP streaming is preferred by more and more content providers, including Microsoft smooth streaming[3], Apple HTTP live steaming[4], Adobe HTTP dynamic streaming[5], Akamai[6], etc. The

*Corresponding author. E-mail:zhangxg@pku.edu.cn

This work was supported by National Natural Science Foundation of China under contract No. 61071082, National Basic Research Program(973 Program) of China under contract No.2009CB320907 and National Key Technology R&D Program of China under Grant 2012BAH18B03.

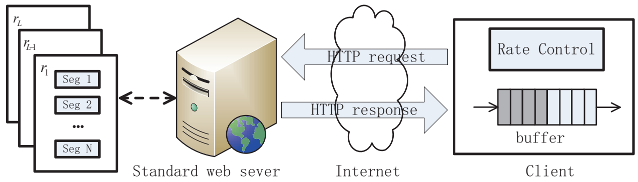


Fig. 1. Dynamic HTTP Streaming System Infrastructure

Motion Picture Experts Group (MPEG) and 3rd Generation Partnership Project (3GPP) have recently moved to support segmented file delivery with their Dynamic Adaptive Streaming over HTTP (DASH) initiatives[7]. All these systems are able to do dynamic bitrate adaptation using segmented HTTP-based streaming. Fig.1 shows the infrastructure of a typical segmented dynamic HTTP streaming system. At the server side, video content is transcoded into multiple discrete bitrates. The different encodings are stored as multiple segment files for each bitrate.

However, providing high quality segmented HTTP-based streaming service is still challenging. The challenge is mainly arising from time-varying bandwidth. Dynamic bitrate adaptation techniques have drawn much research attention since it is able to automatically throttle the video quality to match the available bandwidth, so that the user receives the video at the maximum possible quality. The proposed bitrate adaptation techniques can be classified into two main categories: *1*)bandwidth-based[3, 8], and *2*)buffer-based[6, 9, 10]. In the bandwidth-based rate adaptation technique, it switches up or down the bitrate with estimated network bandwidth. Most of the rate adaptation schemes adopted in some commercial vendors belong to this category. This is demonstrated by experimental evaluation[11]. The main drawbacks of bandwidth-based rate adaptation schemes are that *i*)it is hard to estimate the throughput accurately due to the complex network conditions, and *ii*)time-varying bandwidth would lead to short-term bitrate switching and deteriorate user experience of streaming services. For the buffer-based rate adaptation technique, the rate selection decision is made to provide a smoothly video playback through preventing buffer from either overflow or underflow. Buffer-based technique has been adopted in Akamai[6, 9, 10]. However, it adjusts the bitrate at the server side. This makes it has limitation in supporting the large-scale multimedia delivery since it will dramatically increase the burden on the web server or cache.

In this paper, we present a novel control-theoretic approach to switch video segments in dynamic adaptive HTTP streaming. The rate control is based on a sink-buffer, which has two thresholds: an overflow threshold and an underflow threshold. The adaptation objective is to maximize the playback quality, and the decision that switch-up or switch-down operations between different bitrates is made through preventing the receiver buffer from either overflow or underflow. Two actions are implemented. The first is the sleeping mechanism, which is employed to prevent selecting an unnecessary low bitrate when the occupation of receiver buffer is too high. The other is the bitrate-reset mechanism, which is adopted to avoid buffer underflow in advance when the occupation of receiver buffer is too low. Using control theory, we formulate this control scheme as a proportional (*P*) control system, which exists oscillations and steady-errors. Furthermore, we design a proportional derivative (*PD*) controller to improve its adaptation performance. The stability and settling time of *PD* controller are also analyzed.

Numerous experiment results demonstrate the following main advantages of our proposed scheme: *i*) a video level with higher bitrate is selected as possible as it can, to make better use of network resources. *ii*) bandwidth spikes can be compensated by buffered data, without causing short-term bitrate switching; *iii*) for long-term available bandwidth variations, it selects the best bitrate in time and ensures a continuous video playback.

The main contributions of our work are summarized as:

- We propose a sink-buffer based adaptive HTTP streaming scheme. The decision that switch-up or switch-down operations between different bitrates is made through preventing the receiver buffer from either overflow or underflow and a smoothly video playback is provided. Moreover, a sleeping mechanism is adopted to prevent selecting an unnecessary low bitrate, and a bitrate-reset mechanism is adopted to avoid buffer underflow in advance.
- We formulate this scheme into a proportional controller using control theory, and analyze its inherent flaws. Furthermore, we propose a *PD* control scheme to improve its rate adaptation performance. The conditions for stability and settling time of the *PD* controller are also derived.
- Extensive experiments are conducted over difference scenarios to investigate the dynamic behaviour of the proposed rate adaptation approach. Numerous results demonstrate the effectiveness of our proposed rate adaptation scheme for dynamic HTTP streaming.

This paper is organized as follows. We present the receiver buffer based rate adaptation model in Section 2. A *PD* controller is presented for rate adaptation and its conditions for system stability and settling time are derived in Section 3. We show numerous results in Section 4, and conclude the paper in Section 5.

Table 1. Major symbols used in the paper

$b_{\max}$	maximal size of buffered media data to prevent buffer overflow
$b_{\min}$	minimal size of buffered media data to prevent buffer underflow
$\Re$	set of available video levels for video content that $\Re = \{r_i \| 1 \le i \le N\}$, where r_i is the bitrate of i-th level and it satisfies that $r_i < r_j, \forall i < j$
L	duration in seconds for each segment
$b(t)$	buffered media data at time t
t_k	the time when segment k is downloaded completely
$r(k)$	selected bitrate for segment k, $r(k) \in \Re$
$p(t)$	playback bitrate at the time t, $p(t) \in \Re$
$d(t)$	download bitrate at the time t
$Q(x)$	quantization function that $Q(x) = \underset{r \in \Re, r \le x}{\arg\max}\, r$

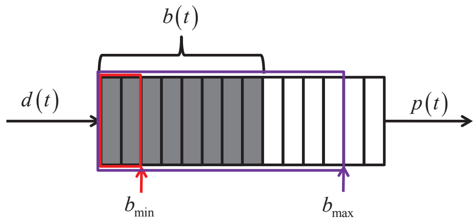


Fig. 2. Receiver buffer model.

2. RECEIVER BUFFER BASED RATE ADAPTATION

We show some of the important symbols used in the paper and their explanations in Table 1. In our work, the client makes a rate adaptation decision depending on the receiver buffer. Fig.2 shows the receiver buffer model, in which $b(t)$ is buffered media data at time t, $d(t)$ and $p(t)$ are the download and playback bitrate respectively. In order to prevent buffer from either overflow or underflow, the size of buffered media data must be in the range $[b_{\min}, b_{\max}]$, where $b_{\min}$ and $b_{\max}$ are the minimal and maximal size of buffered media data to ensure a continuous video playback..

We define t_k as the time when segment k is downloaded completely, then we have:

$$b(t_k) = b(t_{k-1}) + \int_{t_{k-1}}^{t_k} (d(t) - p(t))dt \tag{1}$$

We assume that $p(t)$ and $d(t)$ remain constant in the period $(t_{k-1}, t_k]$. Then the time consumed to download k-th segment is

$$\Delta t_k = t_k - t_{k-1} = \frac{r\,(k)\,L}{d\,(t)}, t \in (t_{k-1}, t_k] \tag{2}$$

where $r(k)$ is the bitrate of segment k and L is the segment duration in seconds. It is worth to note that the video source is pre-transcoded into multiple segments with the same length in seconds. We always have $t \in (t_{k-1}, t_k]$ in the flowing of this paper. Now, Eq.(1) can be rewritten as:

$$\begin{aligned} b\,(t_k) &= b\,(t_{k-1}) + (d\,(t) - p\,(t))\Delta\, t_k \\ &= b\,(t_{k-1}) + r\,(k)\,L\left(1 - \frac{p\,(t)}{d\,(t)}\right) \end{aligned} \tag{3}$$

Estimating $p(t)$ and $d(t)$ is beyond the scope of this paper and they can be estimated by any existing methods. In this paper, the history bitrates are adopted and we assume that $p(t) = p(t_{k-1})$ and $d(t) = d(t_{k-1})$ for $t \in (t_{k-1}, t_k]$.

2.1. Buffer overflow model

When the playback bitrate is smaller than the average download bitrate, the receiver buffer $b\,(t_k)$ is a monotonic increasing function of $r(k)$ and overflow may happen. To ensure buffer not overflow, i.e. $b(t_k) \leq b_{\max}$, we have

$$r(k) \leq \frac{b_{\max} - b(t_{k-1})}{\left(1 - \frac{p(t)}{d(t)}\right) L} \tag{4}$$

Eq.(4) indicates an upper bound of $r(k)$ to prevent buffer overflow. The upper bound, which is denoted as $r^{over}(k) = max(r(k))$, decreases as $b(t_{k-1})$ increases. However, there is no need to select a very low bitrate when $b(t_{k-1})$ is high since there is enough buffered media data to ensure a continuous video playback. We expect that the upper bound of $r(k)$ should be no smaller than the average download bitrate. Otherwise, a sleeping mechanism is adopted to idle the rate adaptation scheme for a certain period.

Therefore, if $r^{over}(k) \geq d(t)$ is satisfied, the selected bitrate for segment k is

$$r(k) = Q\left(r^{over}(k)\right) \tag{5}$$

where $Q(x)$ is quantization function and its definition is given in Table 1.

On the other hand, if $r^{over}(k) < d(t)$, the sleeping mechanism is adopted to idle the rate adaptation scheme. We denotes t_s as the idle time. After t_s seconds, the size of buffered media data is

$$b\,(t_{k-1} + t_s) = b\,(t_{k-1}) - p\,(t)\,t_s \tag{6}$$

substituting it into (4) and let $r^{over}(k) \geq d(t)$, we have

$$t_s \geq \frac{(d(t) - p(t))\,L + b(t_{k-1}) - b_{\max}}{p(t)} \tag{7}$$

Thus, when $r^{over}(k) < d(t)$, we idle the rate adaptation scheme $min(t_s)$ seconds. The key advantage of the sleeping mechanism is to limit the maximum amount of buffered media data, hence avoiding selecting an unnecessary low bitrate.

2.2. Buffer underflow model

When the playback bitrate is higher than the average download bitrate, the buffered media data $b\,(t_k)$ is a monotonic decreasing function of $r(k)$ and underflow may happen. To ensure buffer not underflow, i.e. $b(t_k) \geq b_{\min}$, we have

$$r(k) \leq \frac{b_{\min} - b(t_{k-1})}{\left(1 - \frac{p(t)}{d(t)}\right) L} \tag{8}$$

Eq.(8) indicates an upper bound of $r(k)$ to prevent buffer underflow. The upper bound, which is denoted as $r^{under}(k) = max(r(k))$, decreases as $b(t_{k-1})$ decreases. When $b(t_{k-1})$ is too small leading to $r^{under}(k) < r_1$ (r_1 is the smallest bitrate in set $\Re$), there is no available bitrate in set $\Re$ can ensure buffer not underflow.

Therefore, when $r^{under}(k) < \alpha\, r_1(\alpha \geq 1)$, the bitrate-reset mechanism is adopted to set $r(k)$ to the smallest bitrate in $\Re$, i.e. $r(k) = r_1$. The key advantage of the bitrate-reset mechanism is to reset the selected bitrate compulsively when there is too little buffered media data, hence avoiding buffer underflow in advance. The intuition behind this bitrate-reset mechanism is that selecting a low bitrate is good for preventing buffer underflow since $b\,(t_k)$ is a monotonic decreasing function of $r(k)$ when $p(t) > d(t)$.

On the other hand, if $r^{under}(k) \geq \alpha r_1(\alpha \geq 1)$, the selected bitrate for segment k is

$$r(k) = Q\left(r^{under}(k)\right) \tag{9}$$

and obviously $r(k) \geq r_1$.

2.3. System analysis by control theory

We have resolved the cases of overflow in section 2.1 and underflow in section 2.2 respectively. When the average download bitrate is equal to the playback bitrate, i.e. $p(t) = d(t)$, the size of buffered media data keeps unchanged. In this case, the selected bitrate for segment k is set equal to the bitrate of last segment, i.e. $r(k) = r(k-1)$.

Now, the receiver buffer based rate adaptation scheme is summarized as following two cases:

•$Sleeping$. When $p(t) < d(t)$ and $r^{over}(k) < d(t)$ are met concurrently, the scheme would idle t_s seconds.

•$Rate\ adjustment$. Otherwise, the selected bitrate for segment k-th is given in Eq.(10). Furthermore, Eq.(10) include two states: constant bitrate in sub-formulas (iii-iv) and dynamic bitrate in sub-formulas (i-ii). For the former, the bitrate is set equal to the bitrate of last segment or the smallest bitrate in set $\Re$. For the latter, we analyze it by control theory.

$$r(k)=\begin{cases} Q\left(r^{over}(k)\right), & \text{if } \ d(t)>p(t)\,\&r^{over}(k)\geq d(t) \quad \text{(i)}\\ Q\left(r^{under}(k)\right), & \text{if } \ d(t)<p(t)\,\&r^{under}(k)\geq \beta r_1 \quad \text{(ii)}\\ r(k-1), & \text{if } \ d(t)=p(t) \quad \text{(iii)}\\ r_1, & \text{else} \quad \text{(iv)}\end{cases} \tag{10}$$

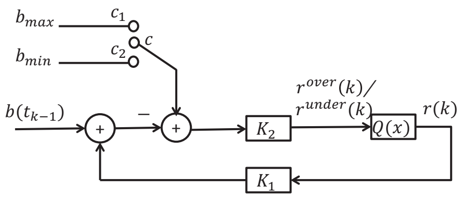


Fig. 3. Block diagram of the proportional control rate adaptation.

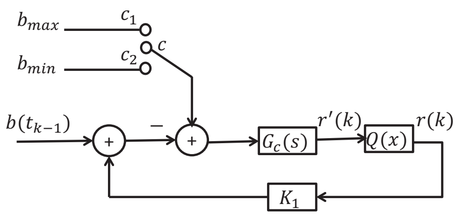


Fig. 4. Block diagram of the modified control system.

First we rewrite the two sub-formulasn (i-ii) in Eq.(11), where the constraints have been omitted for clear representation.

$$r(k)=\begin{cases} Q\left(\frac{b_{\max}-b(t_{k-1})}{\left(1-\frac{p(t)}{d(t)}\right)L}\right) \text{(i)}\\ Q\left(\frac{b_{\min}-b(t_{k-1})}{\left(1-\frac{p(t)}{d(t)}\right)L}\right) \text{(ii)}\end{cases} \tag{11}$$

The rate adaptation method in Eq.(11) turns out to be a proportional (*P*) control system as Fig.3 shows. Contactor c is a selector switch, it connects to contactor c_1 if $d(t)>p(t)$ and connects to c_2 when $d(t)<p(t)$. K_1, K_2 are system parameters which are given as

$$K_1=\left(1-\frac{p(t)}{d(t)}\right)L, K_2=\frac{1}{K_1} \tag{12}$$

In the above *P* controller, the output (selected bitrate) may be oscillated and never converged. This is mainly because that the forward and feedback paths are all proportional. Besides, *P* controller generally has steady-state error, and the error is increasing with time. In the next section, we modify this *P* control system and an effective *PD* controller is designed to improve the rate adaptation performance.

3. RATE ADAPTATION CONTROL PROBLEM

We modify the forward proportional path in Fig.3 with our proposed new controller $G_c(s)$ and redraw the rate control block diagram in Fig.4. Parameters K_1 is given in Eq.(12) and s denotes the Laplace variable. The $G_c(s)$ block represents the rate control component to be designed. The goal of $G_c(s)$ is to stabilize the control system while have small settling time.

The effect of the quantizer $Q(\cdot)$ is to add a quantization error $e_q(k)=r(k)-r'(k)$ to $r'(k)$. This is equivalent to consider $e_q(k)$ as a disturbance and we are able to take the quantizer out of the control loop. Then we can compute the transfer function of the control system as

$$G(s)=\frac{G_c(s)}{1+K_1G_c(s)} \tag{13}$$

and $G_c(s)$ is transfer function for the controller. We choose a proportional derivative (*PD*) controller:

$$G_c(s)=K_p+K_ds \tag{14}$$

since it is able to predict the tread of error variation. By substituting Eq.(14) in Eq.(13), it turns out:

$$G(s)=\frac{\frac{1}{1+K_1K_p}(K_p+K_ds)}{1+\frac{K_1K_d}{1+K_1K_p}s} \tag{15}$$

For system stability, we have the following proposition for the *PD* controller.

Proposition 1. *The rate adaptation control system in Fig.4 is stabilized by the PD controller, if the parameters satisfy the following conditions:*

$$\begin{cases}K_p>-\frac{1}{K_1}\\ K_d>0\end{cases} or \begin{cases}K_p<-\frac{1}{K_1}\\ K_d<0\end{cases} \tag{16}$$

Proof. $G(s)$ has only one pole $s_p=-\frac{1+K_1K_p}{K_1K_d}$. The system stability is ensured by $s_p<0$, i.e.

$$\begin{cases}1+K_1K_p>0\\ K_1K_d>0\end{cases} \text{or} \begin{cases}1+K_1K_p<0\\ K_1K_d<0\end{cases} \tag{17}$$

solving these inequalities in (17), we derive the conditions in proposition 1. □

Settling time, which denotes the response time to approach steady state, is also very crucial for a control system. In this work, we expect that the system settling time is shorter than the duration of one media segment L. This expectation is satisfied by the following proposition for the *PD* controller.

Proposition 2. *The* 5% *step response settling time for the control system in Fig.4 is shorter than* L *by the PD controller, if the parameters satisfy the following conditions:*

$$\begin{cases}K_p>-\frac{1}{K_1}\\ K_d>\max\left(\frac{1+K_1K_p}{K_1L}\ln\frac{-20}{K_1K_p},0\right)\end{cases} \tag{18}$$

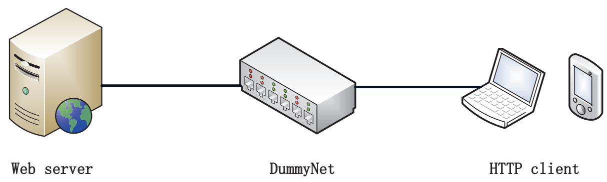


Fig. 5. Experimental topology for dynamic HTTP streaming.

or

$$\begin{cases} K_p < -\frac{1}{K_1} \\ K_d < \min\left(\frac{1+K_1K_p}{K_1 L}\ln\frac{-20}{K_1K_p}, 0\right) \end{cases} \tag{19}$$

Proof. The step response for the control system in Fig.4 is

$$g_u(t) = \frac{1}{1+K_1K_p}\left(K_p - \frac{1}{K_1}e^{-\frac{1+K_1K_p}{K_1K_d}t}\right)$$

and its 5% step response settling time is

$$T_s = \frac{1+K_1K_p}{K_1K_d}\ln\frac{-20}{K_1K_p}$$

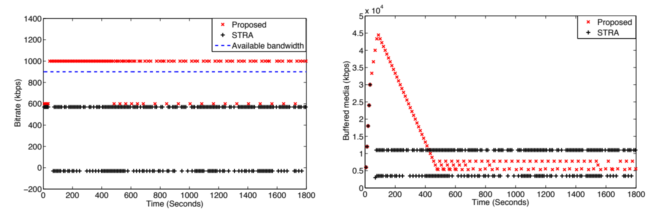
Applying the constraints in proposition 2, we have $T_s < L$. $\square$

4. EXPERIMENTAL RESULTS

We evaluate the performance of our proposed rate adaptation approach using the network topology in Fig.5. The smoothed throughput based rate adaptation (STRA) proposed in [8] is implemented for comparison. DummyNet allows us to control the available bandwidth which is also referred to the downstream path's end-to-end bottleneck. The test video sequence is transcoded into four bitrates $\Re = \{r_1 = 200, r_2 = 600, r_3 = 1000, r_4 = 1400\}$(kbps). Parameters $T = 10$s and $\alpha = 1$. The receiver buffer size is $S = 50$Mbits with $b_{\max} = 0.9S$ and $b_{\max} = 0.1S$. Three different scenarios have been considered in order to investigate the dynamic behaviour of the two considered rate adaptation methods: *1*) available bandwidth keeps constant; *2*) available bandwidth under short-term variations; *3*) available bandwidth under long-term variations. In each scenario, the selected bitrate and buffered media size are illustrated. Because of a lot of overlap in discrete bitrate selection, the plots for the two methods have been staggered vertically to make them easier to read. All plots in the same category have the same bitrate. In all the following figures, when selected bitrate is zero, it denotes sleeping mechanism is implemented and the scheme is idle for some time.

Scenario 1: constant available bandwidth

We consider the scenario that available bandwidth is 800kbps and keeping constant. Fig.6(a) plots the selected bitrate. In STRA, rate selection decision is made depending on pre-defined bandwidth thresholds. Since available bandwidth (800kbps) is smaller than r_3 and it is higher than threshold for r_2, only the segments with bitrate 600kbps is selected. The sleeping mechanism is also adopted in STRA and its sleeping time is determined to prevent the buffer from underflow. The available bandwidth is higher than the selected bitrate, therefore, the algorithm idles at some times and the bandwidth resource is wasted. In our proposed scheme, rate selection decision not only depends on the estimated available bandwidth, but also depends on the playback bitrate and buffered media size. When there is enough buffered media data, a higher bitrate is able to be selected. This is demonstrated in Fig.6(a) that the segments with bitrate r_3 which is higher than available bandwidth are selected at some times. On the other hand, a lower bitrate is selected to ensure a continuous video playback. Due to the initial buffered media data before playback, the segments with bitrate r_3 are requested for a long time at the beginning. Then the selected bitrate is switched between r_2 and r_3 and the bandwidth resource is effectively utilized.



(a) Requested bitrate for video segment.

(b) Buffered media size.

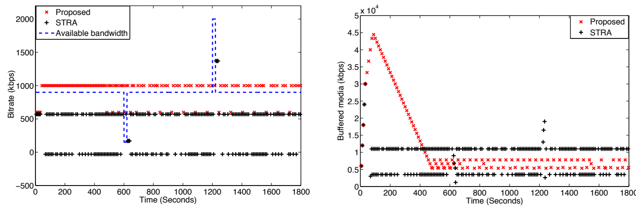
Fig. 6. Performance comparison under persistent available bandwidth conditions.

The buffered media size is illustrated in Fig.6(b). At the beginning, the buffered media size is increasing since playback has not started in both plots. Then using the sleeping mechanism, the buffered media size always stays at a low level to save memory resources of the client in STRA. In our proposed scheme, the buffered media size varies more often since it selects segments with both bitrates r_2 and r_3. Fig.6(b) shows that no buffer overflow or underflow happens. Thus, a continuous video playback is ensured.

Scenario 2: available bandwidth with short-term variations

In this scenario, we summarize a number of experiments where the available bandwidth goes through positive or negative "spikes" that last for only a few seconds as shown in Fig.7(a). The spikes last for 10s. Such short-term variations are common in practice and we think that a good rate adaptation scheme should be able to compensate for such spikes using its buffered media, without causing short-term rate switching.

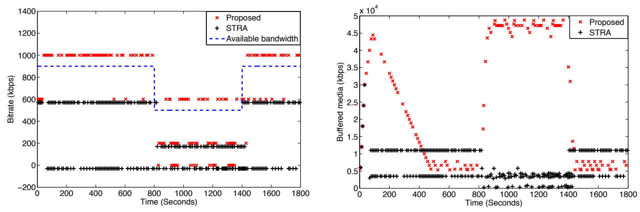
Fig.7(a) shows that a lower bitrate r_1 is selected when a negative spike happens and a higher bitrate r_4 is selected when a positive spike happens in STRA. Besides, the rate switching is too late that it does so sometimes after the end of the spikes. While in our proposed scheme, these spikes are



(a) Requested bitrate for video segment.

(b) Buffered media size.

Fig. 7. Performance comparison under short-term available bandwidth variations.



(a) Requested bitrate for video segment.

(b) Buffered media size.

Fig. 8. Performance comparison under long-term available bandwidth variations.

compensated by the buffered media data. Its performance is close to that the available bandwidth keeps constant as shown in Fig.6(a). The buffered media size is illustrated in Fig.7(b). Short-term rate switching causes buffered media size vary heavily as Fig.7(b) shows. These results demonstrate that our proposed scheme is robust in the scenario that available bandwidth under short-term variations.

Scenario 3: available bandwidth with long-term variations

In this experiment, we consider abrupt drops of the available bandwidth. It drops from 900kbps to 500kbps with a period of 800s. Fig.8(a) shows that the selected rate switches from r_2 to r_1 when available bandwidth drops in STRA. While our proposed scheme selects bitrate r_2 at some times, which is higher than available bandwidth. The reason has been explained in Scenario 1. The buffered media size keeps at a much higher level in our proposed scheme than that in STRA as shown in Fig.8(b). This is reasonable since available bandwidth is low, high level of the buffered media size is more favorable for ensuring a continuous video playback.

5. CONCLUSIONS

We have proposed a receiver buffer based rate adaptation scheme for dynamic HTTP streaming. In this approach, the rate adaptation decision is made through preventing the receiver buffer from either overflow or underflow. Moreover, a sleeping mechanism is adopted to prevent selecting an unnecessary low bitrate, and a bitrate-reset mechanism is adopted to avoid buffer overflow in advance. We analyze this scheme by control theory and it turns out to be a *P* control system. Since *P* controller has some inherent disadvantages, we improve it and a *PD* controller is proposed to improve the rate adaptation performance. The conditions for stability and settling time of the *PD* controller are derived. At last, numerous experiment results demonstrate the effectiveness of our proposed rate adaptation scheme for dynamic HTTP streaming.

References

[1] A. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web: Part 1: Streaming Protocols," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar. 2011.

[2] R. Kuschnig, I. Kofler, and Hellwagner H., "Evaluation of HTTP-based Request-Response Streams for Internet Video Streaming," *in Proc. ACM MMSys11*, pp. 245–256, Feb. 2011.

[3] A. Zambelli, "IIS smooth streaming technical overview," *Microsoft Corporation*, 2009.

[4] R. Pantos and W. May., "HTTP Live Streaming," *IETF Draft,*, jun. 2010.

[5] D. Hassoun, "Dynamic streaming in flash media server 3.5," *http://www.adobe.com/devnet/ashmediaserver/.*

[6] "Akamai HD Network Demo," *http://wwwns.akamai.com/hdnetwork/demo/flash.*

[7] ISO/IEC JTC 1/SC 29/WG 11 (MPEG), "Dynamic Adaptive Streaming over HTTP," w11578, CD 23001-6, Guangzhou, China, Oct. 2010.

[8] Chenghao Liu, Imed Bouazizi, and Moncef Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," *in Proc. ACM MMSys11*, pp. 169–174, Feb. 2011.

[9] L. De Cicco, S. Mascolo, and C.T. Abdallah, "An Experimental Evaluation of Akamai Adaptive Video Streaming over HSDPA Networks," *IEEE Int. Symp. Computer-Aided Control System Design*, pp. 13–18, Sep. 2011.

[10] L. De Cicco, S. Mascolo, and Palmisano V., "Feedback Control for Adaptive Live Video Streaming," *in Proc. ACM MMSys11*, pp. 145–156, Feb. 2011.

[11] S. Akhshabi, A. C. Begen, and Dovrolis C., "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP," *in Proc. ACM MMSys11*, pp. 169–174, Feb. 2011.