Adaptive Online/Offline Smoothing for Streaming Videos over Best-Effort Networks

Chia-Hu Chang [2], Chu-Chuan Lee [1,3], and Pao-Chi Chang [2]

[1] Department of Electrical Engineering, National Central University, Taiwan, R.O.C.
[2] Department of Communication Engineering, National Central University, Taiwan, R.O.C.
[3] Chunghwa Telecom Co., Ltd., Taiwan, R.O.C.
E-mail: { chchang, cclee, pcchang }@vaplab.ee.ncu.edu.tw

Abstract

The real-time video traffic often exhibits significant burstiness due to natural variations within and between scenes. Unfortunately, the current best-effort Internet cannot offer sufficient quality of service (QoS) guarantees to streaming videos, which degrades the received picture quality significantly. Therefore, an end-to-end strategy is required for delivering video data over Internet. The traffic smoothing technique is one of the approaches for handling fluctuations in bandwidth demands of video traffic. This paper proposes an end-to-end adaptive video streaming mechanism by integrating the TCP-Friendly Rate Control (TFRC) with the online/offline video traffic smoothing algorithms. Using TFRC, the proposed framework can dynamically estimate the current available network bandwidth and then adaptively determine the transmission schedule for a video stream. The adaptation of transmission schedule is based on the current network condition, the available server and client buffer spaces, and the characteristics of video frames. Simulation results show that the proposed framework can effectively reduce the packet loss rate that is resulted from variations in delay and available bandwidth. No packet is timeout in all tested cases when the proposed mechanism is used. More importantly, the proposed system can be applied to both offline and online smoothing cases.

I. Introduction

With technical advances in video compression and network delivery, multimedia applications over Internet are dramatically boosted. The video streaming services without waiting for complete download have received tremendous attentions from both academia and industry side. However, the current best-effort Internet cannot offer sufficient quality of service (QoS) guarantees to streaming videos, which degrades the received picture quality significantly. Therefore, an end-to-end strategy is required for delivering video data over Internet. There exist two approaches to handling the burstiness of video traffic. The first approach is the source rate control that alters the quantization parameter (QP) and/or the video frame rate to achieve rate adaptation [1]. However, the rate control mechanism with a Constant Bit Rate (CBR) video output may degrade the picture quality, particularly at scene changes or intervals with significant detail or motion. With the same average bandwidth, a VBR encoding generally offers higher quality than that of CBR encoding [2]. The second approach is the traffic smoothing that is usually implemented at the server or proxy/gateway to smooth the burstiness of video streams without compromising the encoded video quality. Given the encoded video frame bitrate and the available buffer capacity of end system (sender and receiver), a work-ahead smoothing algorithm was utilized [3]. This scheme can simplify the resource allocation problem between the video server and network facilities by determining a series of CBR transmission rates for a video stream. Traffic smoothing can offer substantial reductions in the bandwidth variability for streaming videos with constant quality while avoiding both buffer underflow and overflow at the server and client sides.

A number of approaches for traffic smoothing have been discussed in the past years. Salehi et al. [3] proposed a smoothing strategy to utilize effectively the client buffer space and reduce the rate variability for pre-stored videos. Rexford et al. [4] developed a window-based online smoothing scheme for live video services. However, [3] and [4] require the network to support both rate and delay guarantees, based on resource reservation requests from the video server. The variable bandwidth and network delay, which result from a best-effort network such as today's Internet, may severely degrade the received picture quality even the traffic smoothing scheme is used. In addition, current multimedia applications generally utilize the UDP protocol, which does not provide the congestion control function, to delivery the video data. This may lead to congestion collapse and starvation of TCP traffic in the Internet.

Regarding the best-effort Internet environment, the TCP-friendly rate control (TFRC) is widely suggested to handle the network congestion. It was developed at ACIRI by Floyd et al [5][6]. TFRC is a rate-based and end-to-end congestion control mechanism with lower throughput variation over time. It is suitable for applications such as Internet telephony or streaming media that require a smooth network delivery rate. Based on TFRC, [7] proposed a TCP-compatible source rate control algorithm taking into account the characteristics of multimedia flows such as variable packet size and delay. However, it did not consider the solutions of online/offline video traffic smoothing techniques. In addition, the problem of variable video quality due to the fluctuation of available bandwidth is not discussed in [7], either. Thus, streaming videos over best-effort networks sill pose many challenges.

To address these challenges, this paper proposes an end-to-end adaptive video streaming framework for video streaming applications over best-effort networks by integrating TFRC with the traffic smoothing mechanism. Since the current Internet only offers the best-effort service, it is essential for end systems (sender and receiver) to actively perform the feedback control so that the sender can determine its transmission rate adaptively. The proposed framework is composed of a traffic smoother, a bandwidth predictor, a packet scheduler, a video parser, a QoS monitor and an end-to-end feedback control mechanism. Initially, the traffic smoother employs the smoothing algorithm to compute a transmission schedule for a video stream based on the video characteristics and the current available buffer size form video parser with an appropriate playback delay. The QoS monitor is kept at the receiver to infer the network condition and the status of received traffic. These QoS information is conveyed back to the sender through the feedback control protocol. Based on feedback information, the bandwidth predictor estimates the available bandwidth according to a TCP throughput model and feeds related information to the traffic smoother and packet scheduler. Then the packet scheduler adaptively adjusts the calculated transmission schedule to ensure the smooth delivery quality for real-time video streams with low quality degradation.

The remainder of this paper is organized as follows. Section II presents a brief review on the TCP-friendly rate control used in this paper. Section III describes the proposed adaptive video streaming mechanism. Section IV discusses the experimental results. Finally, Section V concludes this paper by pro-and-con discussion.

II. Overview of TCP-Friendly Rate Control

A TFRC mechanism usually uses a TCP throughput model to determine transmission rates. In order to estimate the TCP's throughput, this study adopts the J. Padhye's model [8] written as:

$$R_{TCP} = \frac{S}{t_{RTT}\sqrt{\frac{2bp}{3}} + t_{RTO}(\min(1, 3\sqrt{\frac{3bp}{8}}))\,p(1+32p^2)} \qquad (1)$$

where R_{TCP} is the estimated transmission rate in bytes/second and S is the average packet size in bytes. The term t_{RTT} is the round trip time (RTT) in seconds and t_{RTO} represents the TCP retransmit timeout value in seconds. The parameter b denotes the number of packets acknowledged by a single TCP acknowledgement and p is the loss event rate, between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted. This model gives an upper bound on the transmission rate. Because many TCP implementations do not use delayed acknowledgements, we set b = 1 in this paper.

A. Estimation of Round-Trip Time and Timeout Value

When TFRC is activated, the sender uses the feedback information and the algorithm $t_{RTT} = q \times t_{RTT} + (1-q) \times t_{RTT'}$ to measure the RTT, where q is set to 0.9 and $t_{RTT'}$ is the sample of RTT gathered from the time interval between the sending of a data packet and the reception of the corresponding ACK. The retransmit parameter t_{RTO} is estimated as max{4R, one second}, which is found to work reasonably well in providing fairness with TCP.

B. Estimation of the Loss Event Rate

Obtaining an accurate and stable loss event rate is of primary importance for TFRC. The loss rate measurement is performed at the receiver, based on the detection of lost or marked packets from the sequence numbers of arriving packets. To calculate the loss event rate p, we first calculate the average loss interval. This is done using a filter that weights the n most recent loss event intervals l_i in such a way that the measured loss event rate changes smoothly. For weights w_i:

$$w_i = \begin{cases} 1, & 1 \le i \le \frac{n}{2} \\ 1 - \dfrac{i - \frac{n}{2}}{\frac{n}{2}+1}, & \frac{n}{2} < i \le n \end{cases} \qquad (2)$$

The average loss interval $\bar{l}$ is calculated as follows:

$$\bar{l} = \frac{\sum_{i=1}^{n} w_i l_i}{\sum_{i=1}^{n} w_i} \qquad (3)$$

Thus, the loss event rate, p is simply:

$$p = 1/\bar{l} \qquad (4)$$

III. Adaptive Video Streaming Framework

Fig. 1 shows the proposed end-to-end smoothing

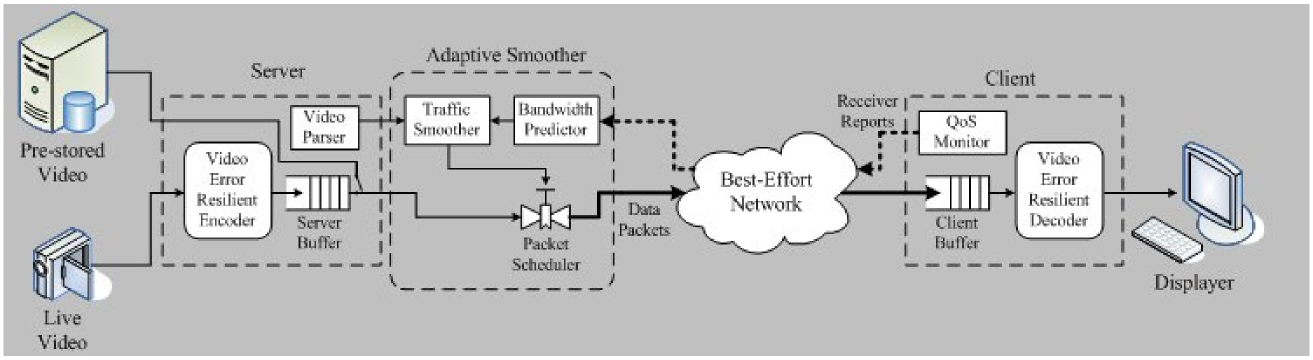

Fig. 1. Architecture of the proposed framework.

framework for streaming videos over best-effort networks, which is applicable to both pre-stored videos and live videos. The detailed operations are described as follows.

A. Transmission schedule determination

Initially, the traffic smoother uses the smoothing algorithm to determine a transmission schedule for a video stream based on the video characteristics and the available buffer size. A compressed video stream consists of N frames, where frame i requires f_i bytes of storage. The sender must always transmit enough data to receiver for preventing undesirable client buffer underflow, where

$$L(t) = \sum_{i=1}^{t} f_i \ , t = 1, \cdots, N \tag{5}$$

$L(t)$ is the lower bound of the cumulated received date at the client side. If the amount of cumulated received data is less than $L(t)$, a buffer underflow happen. In order to accommodate the variable network delay of Internet, we set $N' = N + D_s - D_{offset}$, $L'(t) = L(t - D_s + D_{offset})$ for $t \in [D_s - D_{offset} + 1, N']$, and $L'(t) = 0$ for $t \in [0, D_s - D_{offset}]$, where D_s is the satrtup latency in frames and D_{offset} is the offset in frames. By means of the above bound shifting, a guard area for accommodating the network delay is created. On the other hand, to prevent the client buffer overflow, the client upper bound $U'(t)$ is defined as (6), where D_{offset} = 0, B_c is the client buffer size, and $\beta \in [0,1)$.

$$U'(t) = \min(L'(t-1) + (1-\beta) \times B_c, L'(N')) \tag{6}$$

Given the upper and lower bounds, the proposed mechanism then determines a valid transmission schedule $R(t)$ that meets the condition: $L'(t) \le R(t) \le U'(t)$.

B. QoS Monitor

During the delivery period, a warning feedback of client buffer overflow is activated if the QoS monitor finds the current client buffer status satisfy (7).

$$BI(T_{TP}) \times T_{fb} \ge (1-\beta) \times B_c \tag{7}$$

where BI is the buffer occupancy increase rate in bytes/second, T_{TP} is the trace period time in seconds, and T_{fb} is the feedback period in seconds. On the other hand, if the deposit $D_{deposit}$ satisfies (8), then a warning feedback of client buffer underflow is activated.

$$D_{deposit} \le \min(\left\lceil \frac{D_{offset}}{2} \right\rceil, \left\lceil \frac{t_{RTT}}{\Delta} \right\rceil) \tag{8}$$

where Δ is the unit of time discretization (e.g., 1/30 of a second) and $D_{deposit}$ is defined as $D_{deposit} = \left\lfloor \frac{(t_i^d - t_i^r)}{\Delta} \right\rfloor$, where t_i^r is the received time of the i-th packet in seconds and t_i^d is the display time of the i-th packet in seconds.

Based on the feedback information, the bandwidth predictor estimates the available bandwidth according to the TCP throughput model and feeds related information to the traffic smoother and packet schedular. The packet scheduler then adaptively adjusts the transmission schedule based on the following rate adaptation rules.

C. Rate Adaptation Rule

The key components of Fig. 2 include the TFRC-like Rate Increase Policy, the Shared Rate Decrease Policy, and the Urgent Policy, which illustrate the overall operation on determining adaptively the transmission schedule. In the initial stage of the rules, the sent time of the first packet t_{sent} is set to the current time t_{now}, and the last transmission rate R_{last} is set to the rate R_{smooth} determined by the smoothing algorithm. After sending the first packet, the mechanism determines the waiting interval T_{wait} according to the current rate R_{act}. Whenever a feedback is received by the sender, the budget of permitted delay time is estimated based on $T_{budget} = (D_{deposit} - D_{offset}) \times \Delta$ and the rate adaptation

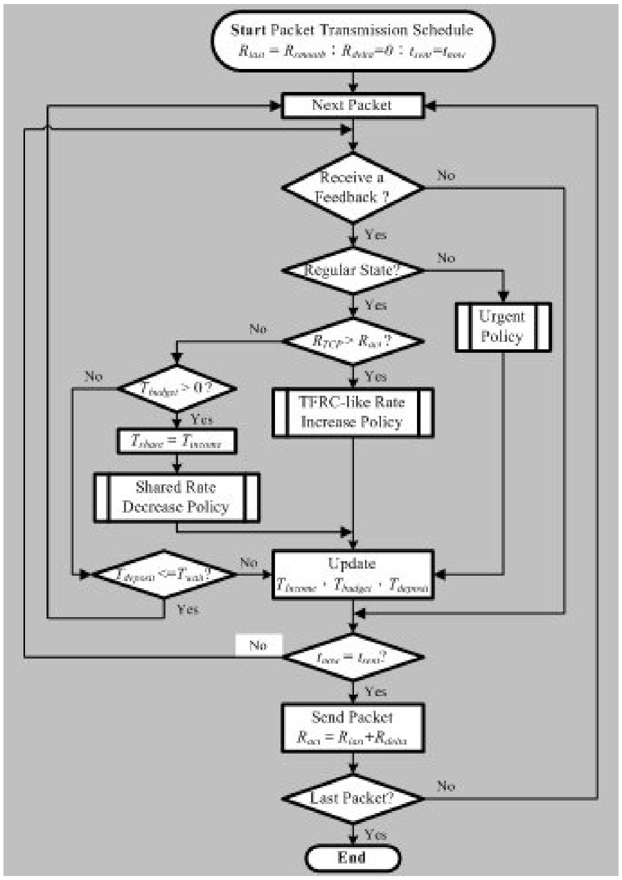

Fig. 2. Flow chart of rate adaptation algorithm.

process is executed.

(1). TFRC-like Rate Increase Policy

If the current network condition obtained from the feedback information is allowed to increase the transmission rate, the proposed mechanism determines the increase amount of transmission rate R_{delta} by

$$R_{delta} = \min(S/t_{RTT}, R_{TCP} - R_{last}) \tag{9}$$

The above operation also increases automatically the budget of permitted delay time. Given R_{delta}, the smoothing algorithm then determines the most suitable transmission schedule that meets the buffer capacity limitation, the video frame encoding rate, and the available bandwidth at the same time. This adjustment is in a TFRC-like manner to achieve relatively lower variation of throughput. The detailed operations are presented in Fig.3.

(2). Shared Rate Decrease Policy

Whenever the network congestion occurs, the proposed mechanism decreases the transmission rate by utilizing the shared rate decrease policy, as expressed by

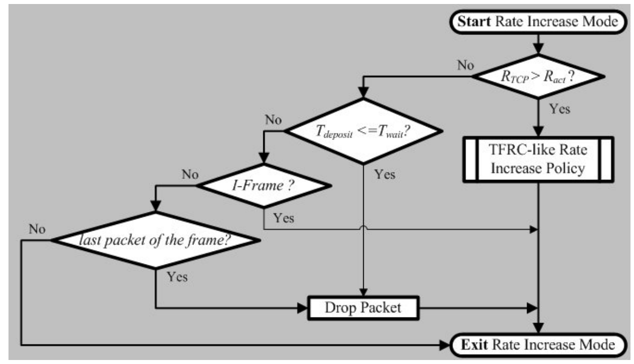

Fig. 3. Flow chart of Rate Increase Mode.

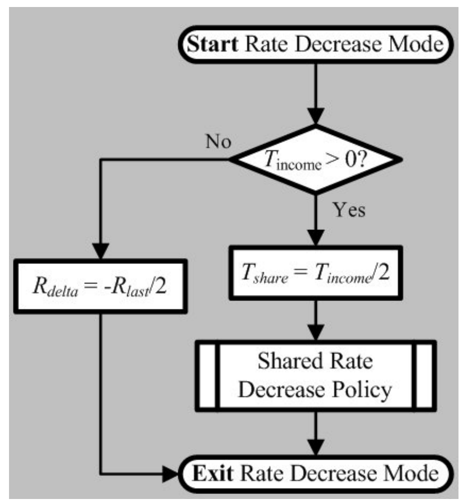

Fig. 4. Flow chart of Rate Decrease Mode.

$$T_{wait}(i, T_{share}) = T_{wait}^{i} + \frac{S_i}{\sum_{j=1}^{n} S_j} \times |T_{share}| \tag{10}$$

where $T_{wait}(i, T_{share})$ is the new waiting interval of the i-th packet in seconds, T_{wait}^{i} is the original waiting interval of the i-th packet in seconds, S_i is the packet size of the i-th packet in bytes, and T_{share} is the share time in seconds. Fig.4 describes the detailed operations.

(3). Urgent Policy

If the feedback information shows that the current buffer usage is close to underflow or overflow, the proposed system enters the urgent policy. When the client buffer underflow or the server buffer overflow warning occurs, the proposed mechanism operates in the TFRC-Like Rate Increase Mode. Alternatively, when the client buffer overflow or the server buffer underflow warning occurs, the proposed mechanism operates in Shared Rate Decrease Mode.

IV. Simulation Results and Discussions

In this paper, both the simulation environment and the proposed mechanism are constructed by the network simulator NS2 [9], as shown in Fig.5. There are 4 connections that share a bottleneck link with several bandwidth settings that presented in the figure. Among these connections, two connections are traditional TCP flows, one connection is the traditional CBR UDP flow, and one connection is the video streaming flow using the proposed mechanism. This work uses the MPEG-4 [10] error resilient video streaming scenario at average 1Mbps with 30f/s frame rate. The adopted group of picture (GOP) structure is 30 frames consisting of IBBP frame pattern. Besides, the decoding and composition delay are ignored for the sake of simplicity.

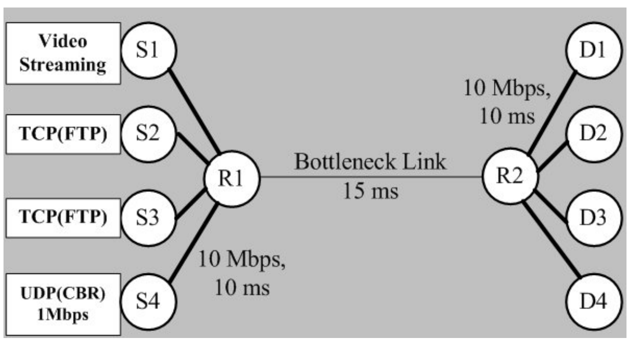

Fig. 5. Network topology.

To evaluate performances of the proposed mechanism, two traditional traffic smoothing schemes without considering the TFRC protocol, i.e., the offline smoothing and online smoothing schemes, are compared. Herein we define five parameters which are used in the following simulation cases.

$$\text{COR} = \frac{\text{Overflow Counts}}{\text{Number of Received Packets}} \times 100\% \quad (11)$$

$$\text{CUR} = \frac{\text{Underflow Counts}}{\text{Number of Received Packets}} \times 100\% \quad (12)$$

$$\text{CLR} = \frac{\text{Number of Congestion Lost Packets}}{\text{Number of Sent Packets}} \times 100\% \quad (13)$$

$$\text{TR} = \frac{\text{Number of Timeout Packets}}{\text{Number of Received Packets}} \times 100\% \quad (14)$$

$$\text{TLR} = \frac{\text{Number of All Lost Packets}}{\text{Number of All Video Packets}} \times 100\% \quad (15)$$

This study first considers the pre-stored video over best-effort network. Herein the offline smoothing scheme is used and the connection that transmitting the video stream is examined. In the simulation case, we set the playback delay and the D_{offset} to 333ms and 6 frames, respectively. Comparing the simulation results presented in Tables 1 and 2, the proposed system has the lower total packet loss rate than the traditional traffic smoothing method. No packet is timeout when the proposed mechanism is used. In contrast, when the traditional traffic smoothing without considering the variable network environment is used, the possibility that the received packets is timeout and the client buffer is underflow is up to 3.34% and 2.65%, respectively.

The proposed mechanism also can be applied for live videos over best-effort network. Meanwhile, the online smoothing scheme is used. In the simulation case, this study sets the playback delay = 1 second, window size = 30 frames, sliding distance = 15, and D_{offset} = 6 frames. Although the transmission rate may be bounded by the real-time video encoding rate, the proposed system still has the lower total packet loss rate than the traditional traffic smoothing method, as shown in Tables 3 and 4. It is noted that, no packet is timeout when the proposed mechanism is applied. Inversely, when the conventional online smoothing without reflecting the variable network condition is utilized, the possibility that the received packets is timeout and the client buffer is underflow is up to 10.03% and 6.27%, respectively.

V. Conclusion

This study proposes an end-to-end adaptive video streaming framework for video streaming applications over best-effort networks, by integrating the TFRC with the video traffic smoothing mechanism. The sender can adaptively adjust the transmission rate by reflecting both the current network condition and the current receiver status. Simulation results show that the proposed mechanism can effectively reduce the packet loss rate that is resulted from the variations in delay and available bandwidth. No packet is timeout in all tested cases when the proposed mechanism is used. More importantly, the proposed framework can be applied to both offline and online smoothing cases.

References

[1] Z. Chen and K. N. Ngan, "Object-based rate control for MPEG-4 video object coding," in Proc. ISCAS'O4, pp. III-973-6, May 2004.

[2] I. Dalgic and F. A. Tobagi, "Performance evaluation of ATM networks carrying constant and variable bit-rate video traffic," IEEE J. Select. Areas Commun., vol. 15, Aug. 1997.

[3] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley, "Supporting Stored Video: Reducing Rate

Variability and end-to-end Resource Requirements through Optimal Smoothing," IEEE/ACM Trans.
[4] Networking, vol. 6, pp. 397–410, Aug. 1998.
[5] S. Sen, J. L. Rexford, J. K. Dey, J. F. Kurose and D. F. Towsley, "Online Smoothing of Variable-Bit-Rate Streaming Video," IEEE Transactions on Multimedia, vol. 2, no. 1, pp. 37–48, Mar. 2000.
[6] S. Floyd et al., "Equation-based Congestion Control for Unicast Applications," Proc. ACM SIGCOMM, Stockholm, Sweden, Aug. 2000, pp. 43–56.
[7] S. Floyd and K. Fall, "Promoting the Use of End-to-end Congestion Control in the Internet," IEEE/ACM Trans. Net., vol. 7, no. 4, pp. 458–472, Aug. 1999.
[8] J. Vieron, C. Guillemot, "Real-time Constrained TCP-compatible Rate Control for Video over the Internet," IEEE Trans. Multimedia, Vol. 6, no. 4, pp. 634–646, Aug. 2004.
[9] J. Padhye et al., "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," IEEE/ACM Trans. Net., vol. 8, no. 2, pp. 133–145, Apr. 2000.
[10] UCB/LBNL/VIANT. (1998) Network Simulator – NS (version 2).
[11] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model version 18.0," N3908, Jan. 2001.

Table 1 Performances for offline video streaming with proposed method over best-effort networks.

Performance	Offline Video Streaming with Adaptive Smoothing					
Buffer Size 512KB	Bottleneck Bandwidth (Mbps)					
	6.5	5.5	4.5	3.5	2.5	1.5
COR	0.00	0.00	0.00	0.00	0.00	0.00
CUR	0.00	0.00	0.00	0.00	0.00	0.00
TR	0.00	0.00	0.00	0.00	0.00	0.00
CLR	0.51	0.64	0.81	1.21	1.63	28.21
TLR	0.51	0.64	0.81	1.21	1.63	28.25

Table 2 Offline video streaming with traditional smoothing over best-effort networks.

Performance	Offline Video Streaming with Traditional Smoothing					
Buffer Size 512KB	Bottleneck Bandwidth (Mbps)					
	6.5	5.5	4.5	3.5	2.5	1.5
COR	0.00	0.00	0.00	0.00	0.00	0.00
CUR	0.50	0.50	0.42	1.09	1.28	2.65
TR	0.81	0.88	0.80	1.31	1.64	3.34
CLR	0.60	0.73	0.93	0.97	1.56	27.38
TLR	1.40	1.60	1.72	2.26	3.18	29.80

Table 3 Online video streaming with proposed mechanism over best-effort networks.

Performance	Online Video Streaming with Adaptive Smoothing					
Buffer Size 128KB	Bottleneck Bandwidth (Mbps)					
	6.5	5.5	4.5	3.5	2.5	1.5
COR	0.00	0.00	0.00	0.00	0.00	0.00
CUR	0.00	0.00	0.00	0.00	0.00	0.00
TR	0.00	0.00	0.00	0.00	0.00	0.00
CLR	0.53	0.75	0.91	1.01	1.89	27.46
TLR	0.53	0.75	0.91	1.01	1.89	27.48

Table 4 Online video streaming with traditional smoothing over best-effort networks.

Performance	Online Video Streaming with Traditional Smoothing					
Buffer Size 128KB	Bottleneck Bandwidth (Mbps)					
	6.5	5.5	4.5	3.5	2.5	1.5
COR	0.00	0.00	0.00	0.00	0.00	0.00
CUR	0.17	0.45	0.55	0.50	2.17	6.27
TR	4.79	5.15	5.19	7.43	7.43	10.03
CLR	0.59	0.63	0.81	1.16	1.60	27.71
TLR	5.36	5.73	5.95	5.98	8.91	34.95