New TCP Video Streaming Proxy Design for Last-Hop Wireless Networks

Wei Pu[1], Zixuan Zou[2], Chang Wen Chen[1]
[1]Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260-2000, USA
{weipu, chencw}@buffalo.edu
[2]Media Technology Research Department, Huawei Technologies Co. LTD., Shenzhen 518129, P. R. China
zixuan.zou@huawei.com

***Abstract*—TCP based HTTP video streaming is becoming more and more popular in recent years. However, for last-hop wireless access, due to TCP's additive increase/multiplicative decrease (AIMD) congest control mechanism, when interference and fading are strong, TCP streaming behaves poorly. In this research, we propose a new design of TCP video streaming proxy at the wireless base station. The new proxy's mission is two folds. First, it transparently isolates the wireless network from the wired network. Specifically, for the last-hop wireless transmission, it adopts a new variant of TCP, we call it raw-TCP, which in combination with a fair scheduler component in the proxy, results in an increased average throughput with low dynamics. Second, through video adaptation, i.e. priority guided packet reordering and deadline driven adaptive truncation, the proxy eliminates playback jitter while maintaining small visual quality variation between successive frames. These two functions fundamentally differentiate our research from the vanilla split-TCP schemes. In our research, the TCP proxy is a multimedia aware element while in traditional split-TCP, the relay node is multimedia oblivious. Simulation results using *ns-2* verify the effectiveness of the proposed solution.**

I. INTRODUCTION

Two classes of video streaming technologies are mature and widely used in multimedia communications nowadays. The first one is peer to peer (P2P) streaming. Generally speaking, P2P streaming requires significant system resources, e.g. CPU time, memory, network bandwidth. For wireless video streaming, users' playback devices are usually battery powered, with less powerful processor and less memory. Therefore, P2P video streaming is much less popular in wireless video streaming than it is in wired networks.

The second category of video streaming technology is TCP based HTTP video streaming. Although TCP is theoretically deemed to be inappropriate for video streaming, engineering practices have found strong evidences that TCP streaming's advantages outweigh its drawbacks. Compared with UDP based P2P video streaming, TCP based approach has major advantages of: 1) being able to penetrate firewalls without adding explicit rules; 2) requiring no additional user end communication software beyond a regular web browser; 3) guaranteeing fairness among concurrent TCP flows. However, in last-hop WiFi access networks (which is one of the most widely used network topology nowadays, e.g. in home, campus, coffee shop, airport, etc.), fading and interfering make the underlying TCP's AIMD congestion control algorithm too conservative in reducing the congestion window. This results in increased traffic dynamics and decreased average throughput. In contrast, high throughput with low dynamics is very much desired for video streaming. Balancing these conflicting factors in TCP video streaming constitutes the major challenge in contemporary mobile media applications.

The mismatch between TCP's congestion control algorithm and wireless channel's error prone characteristics is a well known technical challenge, which has been extensively studied. Among the previous works, the most natural wisdom is divide-and-conquer, i.e. separating the long TCP connection into cascade wired section and wireless section (split-TCP). Existing research results from both theoretical analyses [13][16] and experimental measurements [12][14] confirm that split-TCP is able to improve long TCP connection's throughput. However, for wireless TCP video streaming, the solution of simply adopting split-TCP has not been working well. The reasons are:

1) Compared with direct TCP connection, split-TCP does increase end-to-end throughput. However, due to the impairment of wireless channel, the wireless section of split-TCP becomes a new bottleneck and caps the end-to-end throughput.

2) Split-TCP does not pay any attention to resolve throughput dynamics. However, high throughput dynamics is the primary cause for playback jitter and large initial delay.

This research addresses the above drawbacks by proposing a new TCP proxy design, which not only transparently splits a long TCP connection from video source to a user into two cascade TCPs, but also performs video adaptation and last-hop TCP enhancement at the proxy node for wireless transmission. We summarize the main contributions of this paper as follows.

First, we design a new reliable relay protocol for splitting a long TCP at the TCP proxy. Rather than using vanilla TCP in both component TCPs like split-TCP, we propose raw-TCP, for wireless connection. Raw-TCP conforms to TCP's syntax, while offloads its congestion control function to a flow based fair wireless packet scheduler in the new proxy. Second, we wedge a video adapter in the TCP streaming proxy. It reorders and truncates video segments according to rate-distortion criteria. The new proxy does not require any modification of the protocol stack of either the video source or users.

The rest of the paper is organized as follows. Section II summarizes related work. Section III presents the new TCP

This research is supported by a Gift Funding from Huawei Technologies.

video streaming proxy design in detail. Section IV reports *ns-2* simulation results to evaluate the proxy's performance. Section V concludes this paper with a summary.

II. Related Work

A. TCP Streaming

TCP was originally designed for wired networks. As wired network's transmission error rate is very small, TCP deems transmission error as a sign of network congestion. However, in wireless networks, transmission error due to channel fading and interference occur frequently. Sardar et al [4] summarized solutions for last-hop wireless TCP. Among the solutions listed in [4], we believe that split-TCP is the most suitable basis for wireless TCP video streaming.

Split-TCP was proposed in [12][14] and analyzed in [13][16]. It splits the long TCP into cascade multiple short TCPs. However, to the best of our knowledge, all of the previous works on split-TCP guarantee reliable transmission. For general data transmission, keeping data integrity is necessary. This feature is also one of the primary advantages of TCP over unreliable protocols like UDP. However, for video communication, thanks to video decoder' error concealment capability, we can intentionally relax this reliability constraint to help timely delivery.

Wang et al [5] analyzed the impact of TCP's congestion control algorithm to CBR video streaming. They got a conclusion that network bandwidth of twice of the video stream's average bit rate is needed to guarantee fluent playback. Wang et al [8] studied the feasibility of explore path diversity for TCP video streaming. Tullimas et al [9] proposed to use multiple parallel TCP to combat throughput dynamic. Goel et al [6] minimized latency by dynamically tuning TCP's send buffer. These works solve the TCP streaming problem solely from networking aspect, without considering the signal processing aspect of video error concealment.

B. Streaming Proxy

To help video transmission, various proxy based strategies have been proposed [3][1][10][11][7]. SProxy [3] caches video segment to improve VoD response time. Medusa [1] improves multicast performance. In [11], the authors proposed a multi-worker model and selective packet drop (SPD) algorithm. Their solution cannot address TCP's high throughput dynamic and they only optimize the proxy's buffer without considering the impact of playback time. Huang et al [10] proposed a proxy server at the conjunction of wired and wireless networks. In their model, they assume that the wireless network can provide QoS guarantee while our work is based on contention based WiFi MAC. Furthermore, they assume a priori knowledge of the variable rate constraint imposed by a TCP-friendly rate control. Our scheme does not impose such restrictions. In fact, in [10], they treat TCP and TFRC as alternative solutions for the wired component TCP in split-TCP. This paper integrates TCP, fair scheduling, and video adaptation to improve streaming performance. Ma et al [7] optimized HTTP streaming server using paced output. We would like to point out that our work is very different from existing video proxy approaches as we solve different problems in different application scenario.

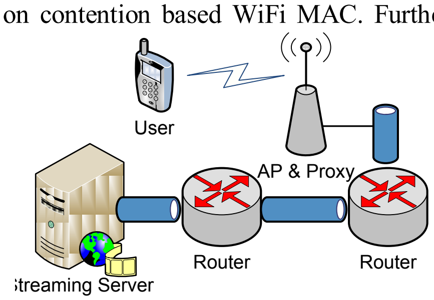

Fig. 1. Illustration of the scenario we study in this research.

III. New TCP Video Streaming Proxy Design

In this research, we study wired-cum-wireless networks, in which H.264/AVC encoded video programs are sent from the source using TCP connection to a wireless user via a WiFi access point, as illustrated in Fig. 1.

A. Video Adaptation

In our scheme, video data are fist partitioned into fixed time length segments. Segments are encoded independently using H.264/AVC video coding standard [2]. Segmentation helps improve the response time of random access operations. Similar mechanism can be found from Apple's *HTTP live streaming* and Adobe's *HTTP dynamic streaming* solution. We assume that each video segment lasts for I seconds.

Due to wireless link dynamics, the network cannot always guarantee to deliver the whole segment before its decoding deadline. We reorder the video stream within a segment according to rate-distortion criteria to minimize the affect of data truncation. Important data are placed at the front of the buffer, seeking for early delivery. The adapter adaptively truncates the video stream according to decoding deadline, historical buffer usage, and wireless link condition. The truncation algorithm trades off video quality with playback jitter.

1) Packet reordering

H.264/AVC contains an extra network abstract layer (NAL), which adapts coded bit stream for network transmission. Specifically, NAL attaches one-byte NAL unit headers to each NAL unit (NALU), which carries the information of: 1) 5-bit T field, which indicates the type of the corresponding NALU; 2) 2-bit R field, which indicates the importance; 3) 1-bit F field, i.e. the forbidden bit, which is used as a hint for decoder's error concealment operation. NALU header makes media aware network element, e.g. TCP streaming proxy in this work, able to reorder/prioritize video segments without decoding.

In this research, we assume frame granularity reordering, which can be fast processed by the wireless access point by parsing NALU headers. Note that by enabling data partition function in H.264/AVC encoder, it is possible to support finer granularity reordering, which, of course, requires more computation and memory resources.

Without loss of generality, assume that video is encoded with hierarchical B structure, as illustrated in Fig. 2. N_B B frames are inserted into the IPPP… structure. Let the frame rate of the basic IPPP… structure be f_{IP}, then B frame's frame rate is approximately:

$$f_B = f_{IP} \cdot N_B \tag{1}$$

Let the layer i of B frames' rate is f_{Bi}. Then as illustrated in Fig. 2, we have:

$$\begin{cases} f_{B_{i+1}} = 2 f_{B_i} \\ f_B = \sum_{i=1}^{L} f_{B_i} \end{cases} \tag{2}$$

where L is the number of layers in the B frame hierarchy. E.g., in Fig. 2, L=3, N_B=8. Assume that IPPP… forms layer 0. Reordering is performed on segment basis. NALUs within one segment are ordered according to the following rules:

a) Non-VCL (Video Coding Layer) NALUs always precede VCL NALUs. This is because non-VCL NALU usually contains important decoding parameter information for the decoder and these NALUs are usually small.

b) Lower layer NALUs are in front of higher layer NALUs. Within the same layer,

b.1) Layer 0 NALUs are ordered chromatically.

b.2) Assume that layer i>0 frames are denoted by f_{0i}, f_{1i}, f_{2i}, …. We group it into i ordered sets.

$$S_j^i = \left(f_{ji}, f_{(j+2^{i-1})i}, f_{(j+2\cdot 2^{i-1})i}, f_{(j+3\cdot 2^{i-1})i}, \cdots\right), 0 \le j < 2^{i-1} \tag{3}$$

NALUs in layer i are ordered as:

$$\left(S_0^i, S_1^i, \cdots, S_{i-1}^i\right) \tag{4}$$

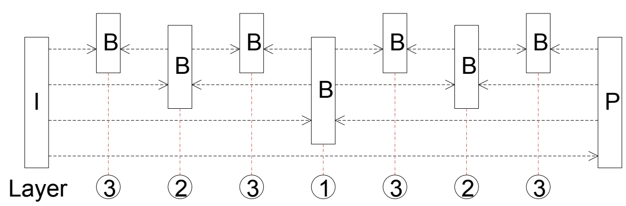

Fig. 2. Example of B frames hierarchy.

The above order guarantees that decoded video quality deteriorates gradually after truncation from tail. It is easy to tell that our frame reordering policy is also directly applicable to H.264/SVC extensions with temporal scalability. Note that if a higher priority NALU has not arrived at the proxy, the proxy selects the most important NALU currently available and sends it.

2) Adaptive truncation

Assume that the beginning of transmission time of the i^{th} segment is denoted by t_i. Its transmission duration is T_i. Segment i is scheduled to begin to play at time Δ_0+iI, where Δ_0 is initial playback delay. There are two requirements in designing the truncation algorithm: a) no data from segment i should be transmitted from the proxy after Δ_0+iI; b) neighboring decoded frames' visual quality should change slowly. These requirements can be formulated as:

$$\begin{cases} t_i + T_i \le \Delta_0 + iI \\ T_i \le \dfrac{\Delta_0 + (i-1)I - (t_{i-1} + T_{i-1})}{2} + I \end{cases} \tag{5}$$

The second equation in (5) restricts neighboring frames' quality fluctuation. In fact,

$$\Delta_0 + (i-1)I - (t_{i-1} + T_{i-1})$$

can be seen as available time pool for segment i. It is the accumulated saved time from previous segments. We force segment i to use at most half of this time pool to protect the pool against exhaustion by single segment whose transmission time period is with very bad link condition. This helps keep video quality deteriorates gradually in bad time.

B. Protocol Design

To ease media processing, contemporary HTTP/TCP streaming protocols attach each transmission unit with a header containing time stamp and other necessary information, e.g. Adobe's real time messaging protocol (RTMP). In this paper, we adopt a simple solution of wrapping each H.264 NALU with a RTP header. This makes it possible for TCP proxy to process video data simply and easily.

In our scheme, the innovation lies on optimizing the last-hop wireless transmission. The proxy located at the wireless AP splits a long TCP into wired and wireless parts. The wired part uses vanilla TCP while the wireless part uses a simplified variant of TCP. We call it raw-TCP. Raw-TCP inherits TCP's syntax while disables TCP's AIMD congestion control algorithm. As discussed above, AIMD is too conservative in wireless networks. Disabling it can better explore wireless channel's capacity. However, unilaterally disabling congestion control can exhaust limited wireless bandwidth. To solve this problem, we propose a fair scheduler at the proxy to enforce fair wireless bandwidth allocation.

The new proxy approach is convenient, without modification of the protocol stack of both the video source and users. Furthermore, the proxy is easy to implement. Both the relay protocol and video adaptation algorithms consume very little system resources. These features make our solution promising for practical deployment. The block diagram of the proxy is illustrated in Fig. 3.

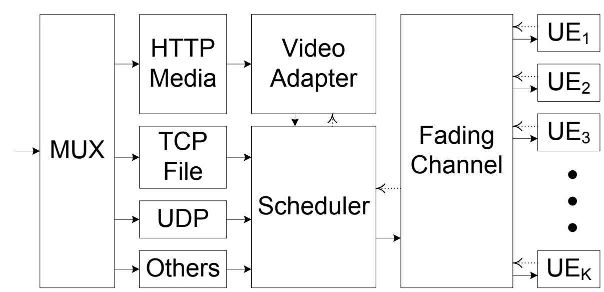

Fig. 3. Block diagram of the proxy.

IV. SIMULATION

In this section, we use *ns-2* to verify the correctness and effectiveness of the proposed new TCP video streaming proxy design. We use H.264's JM codec. We use 4CIF 300 frame *soccer* test sequence with frame rate 30fps. Segment length I=10s, initial playback delay is also set to be 10s. Each segment repeats the same video content. B frames hierarchy's layer number L=3. To isolate the contribution of the new proxy design to error concealment algorithm, basic frame copy error concealment is adopted in the experiment. Deficit round robin (DRR) [15] is used as fair scheduling algorithm at the proxy. Wireless networks use IEEE 802.11g with data transmission rate set to 54Mbps. 802.11g network's frame size is chosen to 2312 Bytes (maximum allowed MAC layer frame size in IEEE 802.11 protocol). Both TCP‟s sending buffer and receiver‟s advertised buffer are set to be large enough so that the simulation results are not affected by these issues.

A. Simulation Topology

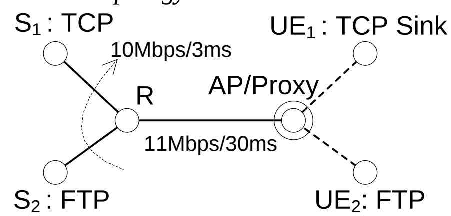

Fig. 4. Simulation topology.

Simulation topology is illustrated as Fig. 4. Wired links are specified by *capacity/delay* pair. There are two infinite background FTP traffics, one from S_2 to *AP* (in wired network) and one from *AP* to UE_2 (in wireless network). Video stream sources at S_1 and sinks at UE_1.

B. Simulation Results

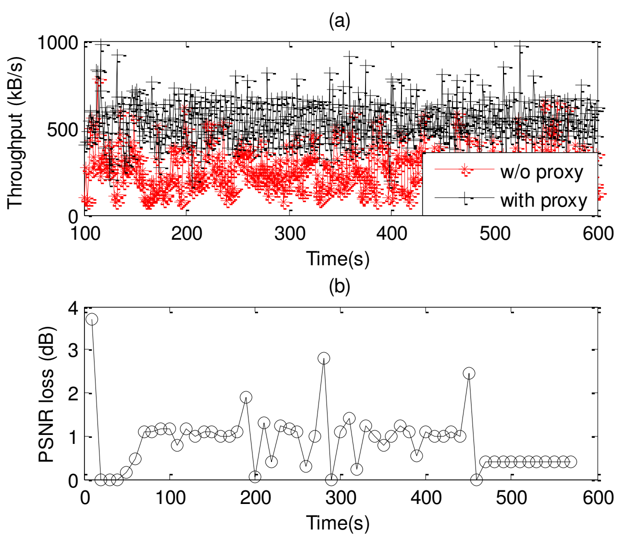

Fig. 5. Throughput and PSNR results.

Fig. 5(a) presents the throughput simulation result. When the new proxy is used, the video stream‟s average throughput is improved from to 270kB/s to 530kB/s. Furthermore, beside average throughput benefit, with TCP proxy, the peak-average throughput ratio is 979/530=1.85 while without TCP proxy, this ratio is 784/270=2.9. These two ratios mean an improvement of streaming video quality. Fig. 5(b) presents each segment‟s average decoding PSNR loss. As the encoded video‟s average bit rate is 553kB/s, which is much higher than the supported throughput without TCP proxy (i.e. average 270kB/s), a big portion of video pictures cannot be decoded in this case. For the case of using TCP proxy, except for the very first segment, PSNR losses are well smoothed between successive frames and finally converge to steady state. Note that this absolute PSNR loss can be reduced by using advanced error concealment algorithm at the decoder side. From these results, we can see that our new proxy design can effectively improve wireless video streaming quality.

V. Conclusion

In this research, we proposed a new TCP proxy solution for video streaming over wireless networks. The proxy is transparent to both video source in wired networks and wireless downstream user. It performs video adaptation operations based on video‟s rate-distortion criteria to resolve short term throughput dynamic. It splits long TCP and performs last-hop TCP enhancement. It does not require modification to either the source server or end users‟ protocol stack. Simulation results show that this new design significantly improves the average throughput, visual quality of decoded video and reduces playback jitters. Furthermore, this new TCP proxy is flexible enough to integrate new video processing functions, such as video format transcoding, content aware adaptation, etc.

References

[1] S. Sen, N. K. Madabhushi, and S. Banerjee, “Scalable WiFi media delivery through adaptive broadcasts,” in *Proc. USENIX NSDI'10*, San Jose, CA, USA, April 28-30, 2010.

[2] “Part 10: Advanced Video Coding,” *ISO/IEC* 14496-10:2010 — Information technology — Coding of audio-visual objects.

[3] S. Chen, B. Shen, S. Wee, and X. Zhang, “SProxy: A caching infrastructure to support Internet streaming,” *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 1062-1072, Aug. 2007.

[4] B. Sardar and D. Saha, “A survey of TCP enhancements for last-hop wireless networks,” *IEEE Communications Survey & Tutuorals*, 3rd Quarter 2006, vol. 8, no. 3, pp. 20-34, 2006.

[5] B. Wang, “Multimedia streaming via TCP: An anlytic performance study,” *ACM Trans. Multimedia Comput., Commun. and Applications*, vol. 4, no. 2., article 16, May 2008.

[6] A. Goel, C. Krasic, and J. Walpole, “Low-latency adaptive streaming over TCP,” *ACM Trans. Multimedia Comput., Commun. and Applications*, vol. 4, no. 3., article 20, Aug. 2008.

[7] K. J. Ma, R. Bartoš, and S. Bhatia, “Scalability of HTTP pacing with intelligent bursting,” in *Proc. IEEE ICME'09*, New York City, June 28-July 3, 2009.

[8] B. Wang, W. Wei, and Z. Guo, “Multipath live streaming via TCP: scheme, performance and benefits,” *ACM Trans. Multimedia Comput., Commun. and Applications*, vol. 5, no. 3., article 25, Aug. 2009.

[9] S. Tullimas, T. Nguyen, and R. Edgecomb, “Multimedia streaming using multiple TCP connections,” *ACM Trans. Multimedia Comput., Commun. and Applications*, vol. 4, no. 2., article 12, May 2008.

[10] L. Huang, U. Horn, F. Hartung, and M. Kampmann, “Proxy-based TCP friendly streaming over mobile networks,” in *Proc. ACM WoWMoM'02*, Sep. 28, 2002, Atlanta, GA, USA.

[11] C.-F. Wong, W.-L. Fung, C.-F. J. Tang, and S.-H. G. Chan, “TCP streaming for low-dealy wireless video,” in *Proc. QShine'05*, Orlando, FL, USA, Aug. 22-24, 2005.

[12] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley, “Inference and evaluation of split-connection approaches in cellular networks,” in *Proc. ACM PAM*, 2006.

[13] F. Baccelli, G. Carofiglio, and S. Foss, “Proxy caching in split TCP: Dynamics, stability and tail asymptotics,” in *Proc. IEEE INFOCOM'08*, Phoenix, AZ, USA, April 15-17, 2008.

[14] M. Luglio, M. Y. Sanadidi, M. Gerla, and J. Stepanek, “On-board satellite „split TCP‟ proxy,” *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, Feb. 2004.

[15] M. Shreedhar and G. Varghese, “Efficient fari queueing using deficit round robin,” in *Proc. ACM SIGCOMM'95*, Cambridge, MA, USA, Aug. 28-Sep. 1, 1995.

[16] F. Xie, J. L. Hammond, and D. L. Noneaker, “Steady-state analysis of a split-connection scheme for Internet access through a wireless terminal,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 515-525, June 2004.