Evaluation of MPEG-DASH over CCNx over different TCPs

Takeshi Muto, Wang Wu, Suphakit Awiphan, Kenji Kanai and Jiro Katto
Graduate School of Fundamental Science and Engineering, Waseda University
55N-6F-09, 3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
E-Mail: t_muto@katto.comm.waseda.ac.jp

Abstract— **In recent years, HTTP Streaming has become the main streaming technology along with evolution of CDN (Content Delivery Networks) and high-speed networks. It can avoid packet losses and firewalls by using reliable TCP transport and port 80 for HTTP, respectively. Furthermore, international standardization of MPEG-Dynamic Adaptive Streaming over HTTP (DASH), a streaming technology which can change bit rate adaptively, has also been completed. On the other hand, a new networking paradigm called Content Centric Network (CCN) has emerged where the content becomes the core of the communication model. In this paper, we evaluate performances of MPEG-DASH streaming over CCN/TCP/IP and CCN/UDP/IP. Though CCN packets are usually conveyed by UDP, we carry out experiments over various TCPs to investigate cooperation between application layer rate control and TCP's congestion control.**

Keywords— MPEG-DASH, TCP, CCN

I. INTRODUCTION

First, today's Internet is based on host-to-host communication where the users specify the location of the content for its retrieval using network addresses (i.e. URLs). However, users often care about the content itself only rather than its location. Content Centric Network (CCN) has been proposed by Van Jacobson to overcome this situation. The main concept of this network paradigm is to provide name-based routing and content caching throughout the network. It has no notion of host at its lowest level - a packet "address" names the content, and not the location [1]. It provides persistence due to the naming of the content, and availability, offering global delivery and low-latency thanks to in-network caching mechanism. Clients are able to send requests of the content without knowing the network topology. CCNx [2] is an open source project to develop and evaluate a new approach in CCN, developed by Palo Alto Research Center (PARC). It runs as an overlay of the current IP based network, and supports both TCP and UDP.

Second, video streaming over hypertext transfer protocol (HTTP) over transmission control protocol (TCP) has become popular due to the benefits it has. Recently, the technology of adaptive HTTP streaming has been proposed. The whole mechanism is to prepare a multi-bit rate encoded media and transmits while handling varying bandwidth conditions so that it can dynamically adapt to bandwidth fluctuations. There are proprietary solutions for adaptive HTTP streaming such as Microsoft Smooth Streaming [3], and Adobe HTTP Dynamic Streaming [4], both utilizing nearly the same architecture. In addition, ISO/IEC MPEG has standardized an adaptive HTTP Streaming referred to as MPEG-Dynamic and Adaptive Streaming over HTTP (DASH) [5].

II. EVALUATION

We have used a network emulator (Packet Storm) between a single connection to set the values of the bandwidth, delay, and packet loss ratio into a fixed value. We installed VLC Media Player and CCNx on our local machines to achieve CCN environment. We also try different TCPs (TCP-Reno[6], TCP-Vegas[7], and CUBIC-TCP[8]) for each streaming session to see how they affect the video delivery. 1280x720 YUV format video is encoded using the DASHEncoder. Moreover the video content is encoded into multiple bit rates: 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500kb/s.

A. Fixed Packet Loss Ratio

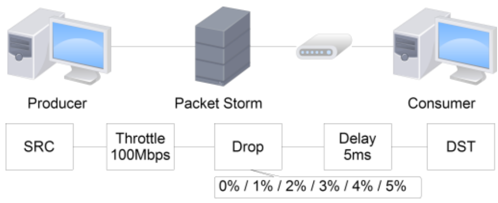

Fig. 1. Network Topology

As shown in Fig. 1, the bandwidth is set to 100Mbps and the delay is set to 5ms. Furthermore, we have run this experiment five times, changing the packet loss ratio to 1, 2, 3, 4, and 5% to see the effect on video delivery.

Figure 2 shows the average streaming bitrate for each packet loss ratio among the TCP variants, average congestion window (cwnd) during the playback, rate estimation measured by the VLC Media Player, and the percentage of selected representation during the playback when packet loss ratio is set to 5%. From the results, we could see that there is a correlation between cwnd and rate estimation. When packet loss ratio increases, rate estimation decreases since this value is estimated by the same algorithm as packet train (group of packets / time to receive). Excessive packet losses increase the value of the denominator, causing the rate estimation to decrease. Cwnd also has the same tendency due to this effect. As the packet loss ratio becomes higher, the average cwnd for the loss-based methods becomes about the same value as that of delay-based method, though loss-based methods have higher values when packet losses do not happen. We could see that the rate estimation of CUBIC-TCP is also the highest in this circumstance.

Because TCP-Vegas is a delay-based method, packet losses do not have a severe effect on its flow control, so it relatively keeps a stable value. CUBIC-TCP relatively has a higher value of streaming bitrate than TCP-Reno due to its aggressive algorithm. When packet loss ratio is set to 5%, CUBIC-TCP has the highest streaming bitrate since the selection of representation was the highest, requesting segments with 1500kbps throughout the whole playback. As for RTT, when packet loss ratio is set to 5%, TCP-Vegas had the highest value, though the average difference was only 3ms among the TCP variants. Since the link was 100Mbps, it could be said that buffer overflow did not occur.

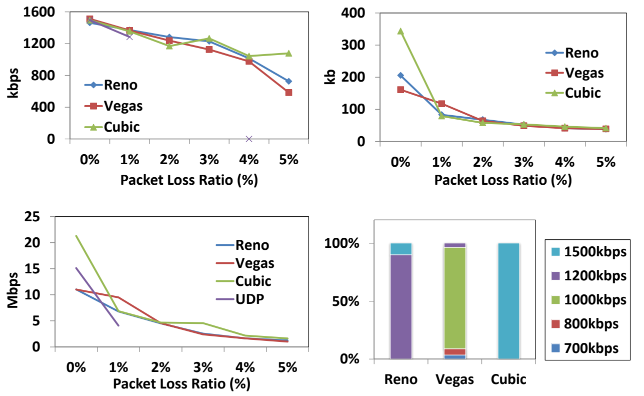

Fig. 2. Average Streaming Bitrate (Top Left), Average Congestion Window (Top Right), Average Rate Estimation (Bottom Left), Percentage of Representation (packet loss ratio = 5%) (Bottom Right)

As for UDP case, when the packet loss ratio goes over 2%, it is unable to play the video. In CCN, when Content Object for an Interest is lost, Interest expires, and retransmission of Interests takes place. From this experiment, it could be said that retransmission of the Interests are too slow in this implementation (Interest timeout is set to 5s), so the consumer cannot receive the data on the demanded time. It could be also said that flow control of Interests, decided by VLC, needs improvement.

B. Fixed Network Bandwidth

In our second experiment, we have narrowed the value of network bandwidth (Figure 3) to see how it affects the streaming bitrate of the playback of video among the TCP variants.

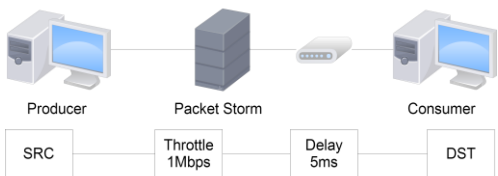

Fig. 3. Network Topology

Fig.4 shows the average streaming bitrate and percentage of selected representation during the playback. Table 1 shows the average results during the playback. As shown in Table 1, though CUBIC-TCP has the highest value of cwnd, this is caused of its aggressive algorithms which accumulate the buffer in the bottleneck excessively, causing an increase in RTT, and having the highest ratio of packet losses. This is the reason why the rate estimation does not differ among the TCPs. In addition, CUBIC-TCP requested more segments with 600kbps, but TCP-Vegas had the highest value of streaming bitrate among the TCP variants. This is because of the number of packet losses that occurred are small. Finally, streaming over UDP had the highest streaming bitrate. Though packet loss ratio was 1.4%, its "Best-Effort Delivery" resulted in high rate estimation, requesting 600 kbps throughout the whole playback.

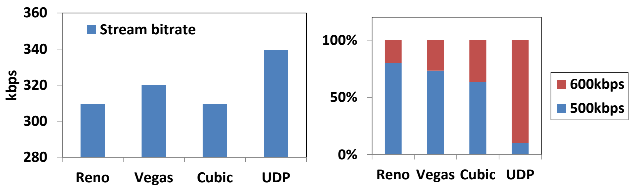

Fig. 4. Average Streaming Bitrate (Left), Average Representation (Right)

TABLE I. RESULTS DURING PLAYBACK (AVERAGE)

	Reno	**Vegas**	**Cubic**	**UDP**
RTT (ms)	115	61	195	
Cwnd (kb)	206	86	599	
Rate Estimation (kbps)	575	578	581	602
Packet Loss Ratio (%)	0.26	0.22	0.58	1.4

III. CONCLUSION

Overall, rate estimation is the core for MPEG-DASH delivery since the selection of representation reflects it. Algorithms of packet transmissions are efficient for the estimation and we could say that packet retransmissions disturb the rate estimation since UDP had the highest value and requested segments with higher representations. Packet losses certainly have a severe effect on them but aggressive flow controls can overcome this problem. Delay-based method did not actually request high representations, but its algorithms avoids buffer overflow, which decreases packet losses and results in high streaming bitrates. In order to satisfy low latency without reducing large throughput, delay/hybrid system is desirable.

IV. REFERENCES

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in Proc. of ACM CoNEXT' 09

[2] CCNx, http://www.ccnx.org/

[3] Microsoft Smooth Streaming, http://www.iis.net/download/smoothstreaming

[4] Adobe HTTP Dynamic Streaming, http://www.adobe.com/products/httpdynamicstreaming/

[5] C. Mueller, S. Lederer, and C. Timmerer, "ITEC - Dynamic Adaptive Streaming over HTTP", [Online], http://www-itec.aau.at/dash

[6] W.R.Stevens: "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and FastRecovery Algorithms", RFC 2001, Jan.1997.

[7] L.Brakmo, S.O'Malley and L.Peterson: "TCP Vegas: New Techniques for Congestion Detection and Avoidance," ACM SIGCOMM'94, Apr.1994.

[8] I.Rhee and L.Xu: "CUBIC: A New TCP-Friendly High-speed TCP Variant", PFLDnet 2005, Feb.2005.