Shaping HTTP Adaptive Streams using Receive Window Tuning Method in Home Gateway

Chiheb Ben Ameur
Orange Labs
Rennes, France
chiheb.benameur@orange.com

Emmanuel Mory
Orange Labs
Rennes, France
emmanuel.mory@orange.com

Bernard Cousin
IRISA, University of Rennes 1
Rennes, France
Bernard.Cousin@irisa.fr

***Abstract*—In this paper, we describe a new method, called RWTM (Receive Window Tuning Method) that shapes HTTP adaptive streams. It employs the flow control in the gateway to improve the quality of experience (QoE) of users. Our use case is when two HTTP Adaptive streaming clients are competing for bandwidth in the same home network. Results show that our proposed method considerably improves the QoE; it improves the video stability, the fidelity to optimal video quality level selection and the convergence speed to the optimal video quality level.**

Keywords— Traffic Shaping; Quality of Experience; HTTP Adaptive Streaming; TCP flow control; Bandwidth Management

I. MOTIVATION

HTTP Adaptive Streaming (HAS) is a streaming video technique based on downloading video segments of short duration (two seconds in general), called chunks, from a HAS server to a HAS client. Each chunk is encoded at multiple quality levels, also called encoding bit rates, or profiles. After filling its playback buffer with chunks, the player in the HAS client will request for chunk periodically in the steady-state phase. The Steady-State phase includes periods of activity (ON periods [1] or Active periods [2]) followed by periods of inactivity (OFF periods [1] or Idle periods [2]). The player estimates the available bandwidth during the ON period to select the video quality level for the next chunk.

The Quality of Experience (QoE) of HTTP adaptive stream depends mainly on three criteria: 1- Video quality level stability; because the frequent change of video quality level bothers the user; 2- Fidelity to optimal quality level selection; the user prefers watching the highest feasible quality level that the available bandwidth allows. This quality level is called optimal quality level; 3- Convergence speed: The user prefers to achieve watching the optimal quality level as soon as possible. The delay that the player takes to reach the optimal quality level is called convergence speed [3].

Our use case is when two HAS clients are competing for bandwidth in the same home network and they request chunks from the same HAS server. In this case, the two main causes of QoE degradation are congestion events and competition between HAS streams. In fact, congestion events -which occurs frequently in the home gateway [5] - reduces the bitrate of packets sent to the HAS clients which degrades the QoE. In addition, when two players compete, they may do false estimation of available bandwidth; for example, if the ON period of the first player coincides with the OFF period of the second player, the first player will overestimate its available bandwidth. As a consequence, it will not select the optimal quality level for the next chunk, hence the degradation of QoE.

The objective of our study is to improve the three criteria of QoE cited above when HAS clients are competing for bandwidth.

The remainder of this paper is organized as follows. In Section II we present our method, called the "Receive Window Tuning Method" (RWTM). Section III presents the results of our experimentations. In Section IV we conclude the paper and give future directions for our work.

II. RECEIVE WINDOW TUNING METHOD

Our proposed method "Receive Window Tuning Method", RWTM, is based on TCP flow control. In fact, TCP uses this flow control mechanism in order to prevent a sender from sending more packets than the receiver capacity. In each TCP segment the receiver specifies the maximum amount of data to be buffered. This specification is sent to the sender in the receiver's advertised window field, *rwnd*. The sender can not exceed the amount *W=min(rwnd, cwnd)* of bytes, called the sender's send window, when sending packets. Where *cwnd* is the congestion window size. As a consequence, if *rwnd* becomes constant and *cwnd* exceeds it, the send window will still be constant and equal to *rwnd*. Therefore, the theoretical sending rate will be always limited by the maximum bitrate *rwnd/RTT*, which can be considered as the shaping rate. Where *RTT* is the round-trip time between the client and the server.

The idea is to modify the value of *rwnd* to have a sending rate limited by *rwnd/RTT*. For instance, in Linux, it is possible to set the parameter *net.ipv4.tcp_rmem* in the client side. However, this sets the maximum socket buffer size for all TCP connections on the client [4], without regard to the specific needs or constraints of each connection. Instead, it is possible to modify the header of each Acknowledgment packet (ACK) sent from the HAS client to the HAS server at the gateway. Another advantage of using the gateway is that the gateway is the device that has information about incoming traffic of all clients of the same home network. So, it has the capability to manage the bandwidth of the home network.

Moreover, the shaping rate, *rwnd/RTT*, requires *RTT* estimation. This estimation is possible in the gateway by using a *passive estimation of RTT* described in [7]. It uses only

acknowledgments (ACKs) sent from clients to the server. We use this estimation only one time for each requested chunk.

Accordingly, RWTM operates as the following: The gateway detects the number of active HAS clients in the home gateway by sniffing the SYN and FIN packets of each TCP connection. Then it computes the shaping rate for each HAS client. The gateway computes the *RTT* value for each HAS client every time it asks for a new chunk. Then, it computes the new value of *rwnd* for each client. The gateway replaces the *rwnd* field of all ACK packets headers sent from HAS client to HAS server by using the computed value of *rwnd*.

III. RESULTS

We proposed our test bed architecture that emulates the real use case. We used two Linux machines that emulate the behavior of the HAS player. The gateway is a Linux machine that reduces the download bit rate to 8 Mbps. We modeled the best effort network by using an RTT with a mean value of 100 ms and with a standard deviation of 7 ms. Our HAS server is a simple Apache server that stores all chunks with different quality levels. All tests are using five video quality levels denoted by 0, 1, 2, 3 and 4. Their encoding rates are 248 kbps, 456 kbps, 928 kbps, 1632 kbps, and 4256 kbps respectively. The shaping rate for each quality level is 10% slightly higher than its encoding rate.

We measure and compare performances of our proposed method (RWTM) with the case without traffic shaping (w/o). Accordingly, we use three metrics [6]: instability (I), fidelity (F), which is the portion of time in which the HAS client asks for optimal quality, and convergence speed (CS) [3]; the time that the HAS player takes to reach a stable optimal quality level for at least 60 seconds.

Moreover, we used three different scenarios: scenario (1): both clients plays simultaneously; scenario (2): Client 2 starts 30 seconds after client 1, and scenario (3): both clients starts simultaneously and we stop the client 2 30 seconds after the beginning of the test. The total duration for each test is equal to 3 minutes.

We run 60 tests for each scenario. The average values of metrics of client 1 for each scenario are listed in table I.

TABLE I. PERFORMANCE METRIC MEASUREMENTS AVERAGE VALUES

Scenario		**1**	**2**	**3**	**Average**
Instability I (%)	***W/o****	7.47	3.87	2.18	4.5
	RWTM	1.63	1.43	1.63	1.56
Fidelity F (%)	***W/o***	49.54	32.64	86.06	56.08
	RWTM	94.98	96.58	95.19	95.58
Convergence speed CS (s)	***W/o***	DC**	DC	20.1	___
	RWTM	19.55	10.98	14.36	14.96

*Without Shaping **Do not converge

For the first scenario, when comparing with the case without shaping (w/o): RWTM is 4.5 times more stable (I= 1.63 % vs. I= 7.47%), has a fidelity score higher by 91.7% (F= 94.98% vs. F=49.54%) and is able to converge rapidly.

For the second scenario, RWTM still have high performances than w/o; it is 2.7 times more stable (I= 1.43 % vs. I= 3.87%), has a fidelity score higher by 91.7% (F= 94.98% vs. F= 49.54%) and is able to converge rapidly.

For the third scenario, we see that performances of w/o are better than in scenarios 1 and 2. This is expected because the client becomes alone for 150 seconds in the home gateway. Nevertheless, RWTM still have better performances than w/o. The instability metric values of RWTM and w/o are close; RWTM is slightly more stable. RWTM has a fidelity score higher by 10.6% (F= 95.19% vs. F= 86.06%) and is able to converge 28.5% faster (CS= 14.36 s vs. CS= 20.1 s) than w/o.

The last column of table I indicates the average performance values of the three scenarios. We can conclude that RWTM is 2.8 time more stable (I= 1.56% against I= 4.5%), 70.4% more faithful to optimal quality level (F=95.58% against F=56.08%) than w/o. RWTM is also able to converge easily and rapidly in 14.94 seconds in average.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel method to shape HTTP adaptive streams in order to improve user experience. Our use case is when two HAS clients are located in the same home network and are competing for bandwidth. Our method, called RWTM (Receive Window Tuning Method) shapes HAS traffic by modifying the client advertised window. It is implemented in the home gateway over TCP layer. It also employs a passive *RTT* estimation for each required chunk to have a more accurate shaping. By comparing players' requested chunks of HAS clients for different scenarios, we found that our proposed method results in a major reduction of instability, an important increase in its fidelity for optimal quality level and has the ability to converge rapidly to the optimal quality level.

Moreover, other shaping methods were proposed to improve the HAS quality of experience. Our future work consists in comparing performances of RWTM with other recent methods.

REFERENCES

[1] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen. "What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth?". NOSSDAV, 2012.

[2] B. J. Villa and P. E. Heegaard. "Group based traffic shaping for adaptive http video streaming," IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), 2013

[3] R. Houdaille and S. Gouache. "Shaping http adaptive streams for a better userexperience". in ACM Multimedia Systems Conference (MMSys), 2012.

[4] A. Mansy, B. Ver Steeg, and M. Ammar. "Sabre: A client based technique for mitigating the buffer bloat effect of adaptivevideo flows". Technical report, Georgia Tech, 2012.

[5] L.Stewart, D. Hayes, G. Armitage, M. Welzl, and A. Petlund. "Multimedia-unfriendly TCP Congestion Control and Home Gateway Queue Management,". ACM Multimedia Systems Conference (MMSys),, 23-25 February 2011.

[6] J. Jiang, V. Sekar, and H. Zhang. "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive". ACM CoNEXT, 2012.

[7] H. Jiang and C. Dovrolis. "Passive estimation of TCP round-trip times". ACM Computer Comm. Review, 32(3):5–21, July 2002.