

LETTER

Cross-Layer Management for Multiple Adaptive Streaming Clients in Wireless Home Networks

Duc V. NGUYEN^{†a)}, Huyen T. T. TRAN[†], Nam PHAM NGOC^{††}, *Nonmembers,*
and Truong Cong THANG[†], *Member*

SUMMARY In this letter, we propose a solution for managing multiple adaptive streaming clients running on different devices in a wireless home network. Our solution consists of two main aspects: a manager that determines bandwidth allocated for each client and a client-based throughput control mechanism that regulates the video traffic throughput of each client. The experimental results using a real test-bed show that our solution is able to effectively improve the quality for concurrent streaming clients.

key words: adaptive streaming, cross-layer management, throughput control, wireless home network

1. Introduction

HTTP Adaptive Streaming (HAS) has emerged as a dominant technology for video delivery over the Internet [1], [2]. In addition, streaming-enabled devices such as tablets and smartphones have become extremely popular these days. So, it is common to have multiple HAS clients sharing a home access link at the same time.

In our previous study [1], we have noticed the drastic fluctuations when multiple TCP connections share the same link. Some other studies also show that multiple concurrent clients result in oscillating bitrates and unfair bandwidth shares [3], [4]. A key reason of this fluctuation is the competition of multiple streams with on-off traffic patterns [4].

Multi-client streaming has been studied previously in the context of highway vehicular networks [5]. In [5], Scalable Video Coding (SVC) format is employed, and methods for resource allocation and layer selection specific to scalable video are proposed. In such vehicular networks, the conflicts among multiple connections are avoided by allocating different resource segments (e.g. time slots) to different clients. In our study, the difference is that multiple TCP connections continuously compete with one another. In [6], [7], free viewpoint video is delivered by multi-path streaming. Especially, a video is represented by multiple description coding (MDC) to cope with packet losses. However, it should be noted that such multiple paths are independent and used to transport a single video.

There have been some studies that try to manage mul-

multiple HTTP streaming clients in the literature. In [8], the authors propose to implement a manager in the residential gateway that applies traffic shaping for each client in a home network, so that each client perceives its bandwidth as intentionally allocated by the manager. However, this traffic shaping does not consider the fluctuation due to on-off traffic pattern on the access link. In [4], traffic shaping is deployed at the server side, which is effective in flattening the on-off traffic pattern. In [9], a packet scheduler is implemented at the server to enforce fairness among the clients. A mechanism of limited TCP congestion window (at the server side) is investigated in [11] for the purpose of a smooth traffic flow. The drawback of such server-based solutions is the high complexity at the streaming server. In [10], the authors propose to control a server's sending rate by limiting the number of bytes on-the-fly using HTTP range request. However, this may result in too many requests, which affect the server performance, while the on-off traffic pattern is still present. A recent study in [12] and [13] tackles this issue by the complex network-assisted approach, where the bitrates are decided for multiple clients in different policies in the network. However, TCP throughput control is not considered in this study.

In this letter, we propose a cross-layer solution for managing multiple clients in a home network. The solution consists of the following main features:

- Our solution does not require a dedicated residential gateway, which is expensive or complicated for users.
- Traffic shaping (or regulating) at the transport layer is done by a throughput control module at each client. Specifically, the TCP receiver window at the clients is adjusted for this purpose.
- A bandwidth manager, which can be deployed at any locations (e.g. server, gateway, clients), decides the bandwidth for each client, so that the traffic of all clients will not exceed the capacity of the access link. As the client will ultimately change the requested bitrate of its video streaming application, this operation actually belongs to the application layer.

To the best of our knowledge, our study is the first one that provides a flexible framework for managing multiple clients of HTTP adaptive streaming, without modifying the transport mechanism at servers or gateways.

Manuscript received October 12, 2017.

Manuscript revised February 6, 2018.

Manuscript publicized February 27, 2018.

[†]The authors are with the University of Aizu, Aizuwakamatsu-shi, 965–8580 Japan.

^{††}The author is with Hanoi University of Science and Technology, 1 Dai Co Viet, Hanoi, Vietnam.

a) E-mail: nvduc712@gmail.com

DOI: 10.1587/transinf.2017EDL8227

2. Proposed System

2.1 Overview

The architecture of the proposed system is shown in Fig. 1. Clients in a home network connect to the servers through an access link (bottleneck). The manager, which locates inside the home network in this figure, communicates with the clients to exchange information. Note that, if the manager stays outside the home, it can recognize the clients of the same home by IP address and the content of client messages.

Communication flows between the components are shown in Fig. 2. At the beginning of a streaming session, each client retrieves the metadata that contains content-related information (e.g. available bitrates) from the server. The client makes decisions on which video segments to be downloaded based on the metadata and the allocated bandwidth from the manager. For first video segment, the client simply selects the lowest bitrate.

During the streaming session, the client performs two additional tasks. First, it periodically measures and sends throughput information to the manager. Second, it continuously listens for updates from the manager and passes the received updates to the throughput control module that is

responsible for controlling the download throughput of the client.

The manager decides the portion of bandwidth to be allocated to each client based on throughput information from the clients. Messages containing the allocated bandwidth are sent to the clients if there is a change in the number of clients or the available bandwidth of the access link. In the following, the manager and the throughput control module are described in detail.

2.2 Bandwidth Allocation by the Manager

In our solution, the tasks of the bandwidth manager are 1) estimating the total bandwidth and 2) determining the portion of bandwidth for each client.

Based on the data reported by the clients, the manager computes the instant total bandwidth by summing up throughputs of individual clients. The estimated bandwidth is computed using the moving average equation as follows.

$$B^e(k) = \begin{cases} \alpha B^e(k-1) + (1-\alpha)B(k), & k > 0 \\ B(k), & k = 0 \end{cases} \quad (1)$$

where $B(k)$ and $B(k)^e$ denote the k^{th} instant bandwidth and estimated bandwidth. The parameter α is set to 0.8. This step is similar to the case of having two connections (for audio and video) presented in our previous work [1].

In the context of multiple clients, the portion of bandwidth allocated to each client may depend on a number of factors including user preference and device capabilities. Currently, we assume that each client is associated with a weight that represents its importance. Denote N as the number of clients and w_i the weight of client i ($1 \leq i \leq N$). Bandwidth R_i allocated to the client i is determined as follows.

$$R_i = \frac{w_i \times B^e(k)}{\sum_{j=1}^N w_j}. \quad (2)$$

When there are changes in the number of the clients or the total bandwidth, the manager re-computes values of $\{R_i, 1 \leq i \leq N\}$ and sends updating messages to all clients. At a given segment interval, the selected bitrate is the highest video bitrate that is lower than the allocated portion of bandwidth.

2.3 Throughput Control Module at Each Client

The task of the throughput control module is to regulate the download throughput of each client as allocated by the manager. Our method is to modify the receiver window parameter of the TCP receiver. As our method is implemented at the client side, the workload and complexity of the servers would be significantly reduced in comparison to server-side methods. According to [14], sending rate R_i of the TCP sender of client i could be approximated as follows:

$$R_i = \frac{\min(rwnd_i, cwnd_i)}{RTT_i} \quad (3)$$

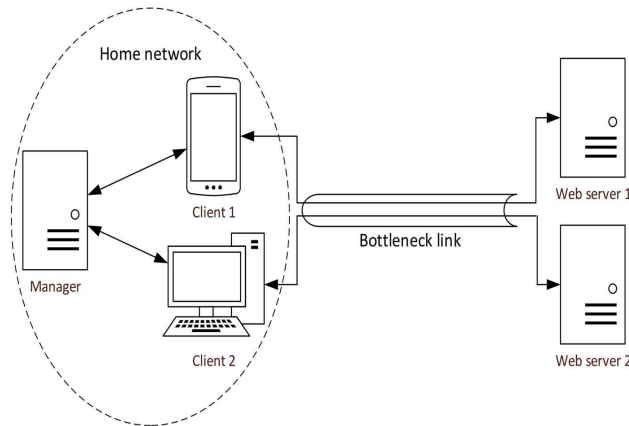


Fig. 1 Architecture of the proposed system.

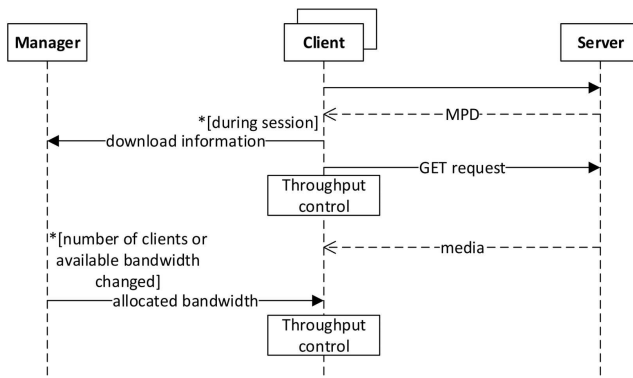


Fig. 2 Communication flows between the components of the proposed system.

where $rwnd_i$ denotes the client's TCP receiver window, $cwnd_i$ the server's TCP congestion window, RTT_i the round-trip time delay. The value of $\min(rwnd_i, cwnd_i)$ is known as the sender's sending window.

In practice, today's modern networks have very high quality and $cwnd_i$ is mostly at high values. Even if $cwnd_i$ is the smaller than $rwnd_i$, the download throughput of the related connection will simply be lower than expected and cause no harms to the other connections. Thus the throughput of the client i can be regulated by modifying its receiver window $rwnd_i$.

So, given a desired sending rate R_i and the expected round-trip time RTT_i^e of client i , the target receiver window size W_i is computed as follows.

$$W_i = R_i \times RTT_i^e. \quad (4)$$

In HTTP streaming, RTT can be obtained at application layer [16]. Specifically, for each video segment, the client computes a segment-based RTT , which is the time from when the request for the segment is sent until the first byte of that segment is received. Similar to the total bandwidth, RTT_i^e is also estimated using the moving average equation as follows.

$$RTT_i^e(k) = \begin{cases} \delta RTT_i^e(k-1) + (1-\delta)RTT_i(k), & k > 0 \\ RTT_i(0), & k = 0 \end{cases} \quad (5)$$

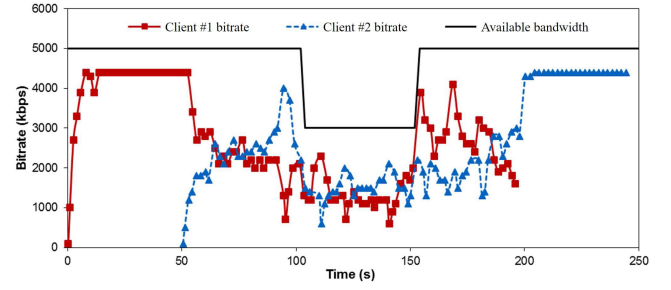
where $RTT_i(k)$ and $RTT_i^e(k)$ are the measured RTT and the estimated RTT at the k^{th} video segment. Note that $RTT_i(0)$ can be obtained when client i requests the metadata at the beginning of the session. Currently, the value of the smoothing factor δ is set to 0.8. Good values of δ and α , which could be obtained in a manner similar to that of [1], will be reserved for our future work.

3. Experimental Results and Discussions

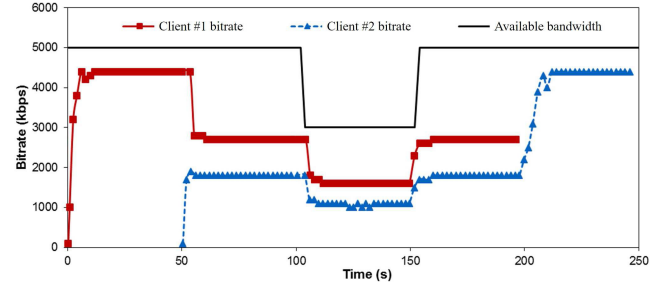
In our experiments, the clients and the manager run on separate computers in a home network, connecting to a media server through a bottleneck link. DummyNet tool [10] is installed on the server as network emulation. RTT of the link is set to 50ms in all experiments. Each client can switch its bitrate from 100kbps to 5Mbps, with a step size of 100kbps. All segments have the same duration of 2 seconds.

The proposed system is compared to un-managed case where each client does not know about the presence of the others and selects video bitrate based on its own measured throughput (i.e., the instant throughput based method described in [2]).

In the first experiment, we consider the scenario in which two clients, denoted by client #1 and client #2, share a bottleneck link with variable bandwidth. The bandwidth is initially set to 5Mbps, then decreased to 3Mbps at $t = 100$ s, and finally increased to 5Mbps at $t = 150$ s. Client #2 starts 50s after client #1, and each client lasts for 200 seconds.



(a) The un-managed case.



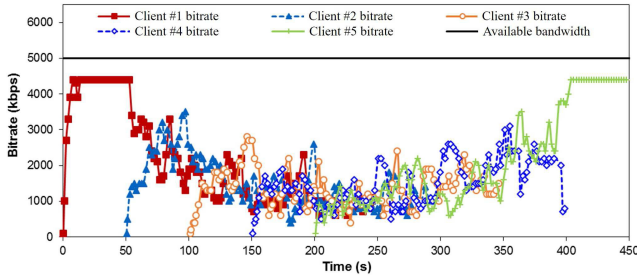
(b) The proposed system.

Fig. 3 Throughput and bitrate of the two clients over time.

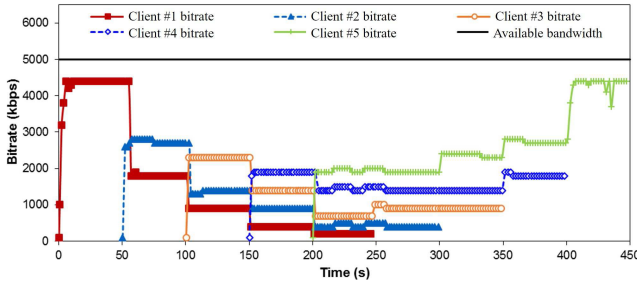
The weights of client #1 and client #2 are 0.6 and 0.4, respectively. Note that, if only one client is on the link, the weight of that client is set to 1 to allow it to use the whole bandwidth.

Figure 3 (a) shows the bitrate curves of the un-managed case where the clients run independently without bandwidth management. During the first 50 seconds, client #1 quickly switches to the bitrate of 4400kbps and remains stable. As client #2 starts at $t = 50$ s, both clients become unstable. We can see significant fluctuations in bitrate of both clients until client #1 terminates, thereby resulting in frequent changes in video quality. After client #1 terminates at $t = 200$ s, client #2 increases its bitrate to 4400kbps and remains unchanged until the end of the streaming session at $t = 250$ s. It can also be seen that the clients occasionally experience unfair bandwidth share. For instance, from $t = 155$ s to $t = 180$ s, client #2 received more bandwidth than client #1 as shown by the clients' selected bitrates.

The results of our proposed system are shown in Fig. 3 (b). First, it can be clearly seen that the bitrate of each client is very stable and changes in accordance with the available bandwidth. Moreover, when there are concurrent clients, the bitrate of each client is also proportional to its weight. Specifically, from $t = 50$ s to $t = 100$ s, the bitrates of client #1 and client #2 are stable at 2700kbps and 1800kbps, respectively. As the bandwidth drops to 3Mbps at $t = 100$ s, the manager quickly detects the change and sends updating messages to the clients. Accordingly, client #1 and client #2 reduce their bitrate to 1600kbps and 1000kbps, respectively. Similarly, when the bandwidth increases to 5Mbps at $t = 150$ s, both clients switch their bitrate to the same values as before. It means that the throughput control module is effective in avoiding the unnecessary competition between



(a) The un-managed case.



(b) The proposed system.

Fig. 4 Bitrate of the five clients over time.

TCP connections.

The second experiment is to investigate the proposed system with different numbers of clients. We consider the scenario where there are five clients sharing a bottleneck link with a stable bandwidth of 5 Mbps. The clients are started sequentially every 50 seconds, and each client lasts for 250s. The weights of the clients are as follows: $\{0.4, 0.6\}$ for 2 clients, $\{0.2, 0.3, 0.5\}$ for 3 clients, $\{0.1, 0.2, 0.3, 0.4\}$ for 4 clients, and $\{0.05, 0.1, 0.15, 0.3, 0.4\}$ for 5 clients. This is to avoid the overlaps between the bitrate curves, thus giving a clear result. Given the number of clients, the corresponding set is selected and the weights are assigned to each client.

The results with the un-managed case are shown in Fig. 4(a). Similar to the previous experiment, we can see that bitrate fluctuations occur whenever there are more than one client on the link at the same time. However, it seems that the fluctuations decrease as the number of concurrent clients increases. Meanwhile, the results of our proposed system are consistent with the increasing number of clients as shown in Fig. 4(b). Every time a new client joins the system, the bitrate of each client changes according to the updated information from the manager. It can be seen that the bitrate of all clients are mostly stable except some minor fluctuations occurring with client #2, client #4, and client #5 in the interval from $t = 200$ s to $t = 250$ s.

4. Conclusions

In this letter, we have proposed a cross-layer management framework for multiple HTTP clients running on different devices in a home network. Our solution consisted of a bandwidth manager and a client-side throughput control

module. The experiment results showed the ability of our proposed approach to effectively control bandwidth of each client, thus helping to significantly improve the quality of experience. In the future work, we will focus on the optimization problems for deciding the clients' bitrates to improve the overall benefit.

References

- [1] T.C. Thang, Q.-D. Ho, J.W. Kang, and A.T. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Trans. Consum. Electron.*, vol.58, no.1, pp.78–85, Feb. 2012.
- [2] T.C. Thang, H.T. Le, A.T. Pham, and Y.M. Ro, "An evaluation of bitrate adaption methods for HTTP live streaming," *IEEE J. Sel. Areas Commun.*, vol.32, no.4, pp.693–705, April 2014.
- [3] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: Picking a video streaming rate is hard," *Proc. ACM Internet Measure Conf.*, pp.225–238, 2012.
- [4] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A.C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," *Proc. 23th ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV '13) Workshop*, pp.19–24, 2013.
- [5] R. An, Z. Liu, H. Zhou, and Y. Ji, "Resource allocation and layer selection for scalable video streaming over highway vehicular networks," *IEICE Trans. Fundamentals*, vol.E99-A, no.11, pp.1909–1917, Nov. 2016.
- [6] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding of free viewpoint video for multi-path network streaming," *Proc. 2012 IEEE Global Communications Conference (GLOBECOM)*, CA, USA, pp.2150–2155, Dec. 2012.
- [7] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding and recovery of free viewpoint video for wireless multi-path streaming," *IEEE J. Sel. Topics Signal Process.*, vol.9, no.1, pp.151–164, Feb. 2015.
- [8] M. Adams and C. Phillips, "Optimizing fairness of HTTP adaptive streaming in cable networks," *Proc. Spring Technical Forum*, pp.1123–1132, 2012.
- [9] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," *Proc. 3rd Multimedia System (MMSys '12)*, pp.1–9, 2012.
- [10] D. Havey, R. Chertov, and K. Almeroth, "Receiver driven rate adaptation for wireless multimedia applications," *Proc. 3rd Conf. Multimedia Systems Conference, USA*, pp.155–166, 2012.
- [11] S. Nazir, Z. Hossain, R. Secchi, M. Broadbent, A. Petlund, and G. Fairhurst, "Performance evaluation of congestion window validation for DASH transport," *Proc. ACM NOSSDAV '14*, NY, pp.67–72, 2014.
- [12] T. Ojanpera and H. Kokkonen-Tarkkanen, "Wireless bandwidth management for multiple video clients through network-assisted DASH," *2016 IEEE 17th Int'l Symposium on a World of Wireless, Mobile and Multimedia Net (WoWMoM)*, Coimbra, pp.1–3, 2016.
- [13] T. Ojanpera and H. Kokkonen-Tarkkanen, "Experimental evaluation of a wireless bandwidth management system for multiple DASH clients," *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, pp.1–7, 2016.
- [14] J.F. Kurose and K.W. Ross, *Computer Networking: A top-down approach*, Pearson, Boston, 2010.
- [15] L. Rizzo, "DummyNet: A simple approach to the evaluation of network protocols," *SIGCOMM Comput. Commun. Rev.*, vol.27, no.1, pp.31–41, Jan. 1997.
- [16] T.C. Thang, H.T. Le, H.X. Nguyen, A.T. Pham, J.W. Kang, and Y.M. Ro, "Adaptive Video Streaming over HTTP with dynamic resource estimation," *IEEE/KICS Journal of Communications and Networks*, vol.15, no.6, pp.635–644, Dec. 2013.