DISTRIBUTED-COLLABORATIVE MANAGED DASH VIDEO SERVICES

Kemal E. Sahin, K. Tolga Bagci, A. Murat Tekalp

Department of Electrical and Electronics Engineering
Koç University, 34450 Istanbul, Turkey

ABSTRACT

We propose a new distributed-collaborative managed DASH video service architecture over software defined networks (SDN) that enables fair and stable video quality to heterogeneous resolution clients. The proposed service is managed by the video service provider (VSP) in collaboration with the network service provider (NSP), where groups of clients sharing a network slice with a reserved throughput collaborate with each other to compute their own fair-share bitrates. Our novel distributed service architecture allows each client to share its buffer status with other clients in the same collaboration group so that each client can estimate a group-buffer-status aware fair-share bitrate, enforce this rate by TCP receive-window size control over a network slice reserved for the group, and perform application-level DASH video rate adaptation that is consistent with this enforced fair bitrate. Experimental results show that the proposed collaborative video service outperforms the traditional competitive DASH clients in terms of (i) minimizing quality fluctuations per client, (ii) fairness among heterogeneous DASH clients, and (iii) maximizing the total goodput of reserved network slice.

Index Terms— SDN, managed DASH service, TCP receive window, stable video quality, quality fairness

I. INTRODUCTION

Dynamic adaptive streaming over HTTP (DASH) is a popular method for streaming video over the Internet[1]. A DASH server makes video available at multiple bitrates (quality levels) that are encoded as fixed-duration segments. DASH clients first download the media presentation description (MPD) file to access content related information and then request video segments sequentially at quality levels that best fit their current player buffer status and network conditions.

HTTP runs over transport control protocol (TCP), and TCP rate control is unaware of the DASH bitrate adaptation logic at the application layer. Furthermore, multiple DASH clients sharing a link are unaware of each other. As a result, multiple competing DASH streams poorly interact especially in the presence of congestion [2], and we observe i) instability of goodput (video quality) per DASH client, ii) unfairness between competing heterogeneous DASH clients, and iii) that the total goodput of all DASH clients is less than the available network throughput by some margin.

We propose a novel distributed-managed DASH video service architecture over software defined networks (SDN) to address these problems, where groups of DASH clients sharing a network slice with reserved throughput collaborate with each other to compute their own fair-share bitrates. Related works are discussed in Section II. The proposed service architecture and method is described in Section III. Experimental results that are presented in Section IV demonstrate the superiority of the proposed method over traditional competitive DASH services. Conclusions are presented in Section V.

II. RELATED WORKS

There are many works on QoE-aware DASH streaming and/or QoE-fairness among heterogeneous DASH clients, which can be classified as i) centralized, controller-based solutions [4], [3], [5], [6], [7], ii) video service provider (VSP) based solutions [8], [9], and iii) player-only solutions [10], [11]. However, none of these works provide an analysis of the interaction between TCP rate control mechanism and QoS resource reservation.

QoE refers to video quality experienced by the clients, which is related to several perceptual and technical parameters [12]. Factors that influence QoE are discussed in [13]. The evaluation of QoE with respect to different bitrate adaptation logics is presented in [14]. Wang et al. [15] propose to measure video quality using the structural similarity metric (SSIM), and observe that different bitrates are required to achieve the same video quality for different resolutions.

When considering clients with heterogeneous video resolution, users with higher resolution need higher bitrate to experience the same video quality; hence, for QoE-fairness, client bitrates should be proportional to their video resolution. Georgopoulos [3] proposes providing QoE-fairness to heterogeneous clients by dynamically allocating network slice to each device based on SSIM [15] with the assistance of SDN.

In our recent work [16], we discussed the interaction between TCP rate control and quality of service (QoS) reservation for a group of clients, and proposed alternative cooperative streaming architectures to ensure stable and fair video quality to collaborating clients. This paper extends the distributed collaborative streaming model in [16] to the case where the capacity of network slices reserved for the VSP is dynamically managed by the NSP depending on other traffic.

978-3-901882-98-2

III. DISTRIBUTED-MANAGED DASH SERVICE

The proposed distributed-managed DASH service framework aims to provide stable goodput at a fair rate to a group of heterogeneous DASH clients that share a network slice/link with the assistance of the VSP and NSP, and enable each client to set their TCP receive-window size and DASH video adaptation rate accordingly. The proposed video service architecture is presented in Section III-A. Dynamic grouping of DASH clients at the VSP is described in Section III-B. The computation of fair-share bitrate by clients is described in Section III-C.

III-A. Service Architecture

The proposed system architecture is depicted in Fig. 1, where the VSP has a management server (VSP-M) that manages collaboration between the VSP and NSP and between DASH clients. The NSP hosts a traffic engineering manager (TEM) that allocates resources and a VSP interface (VSP-I) that enables sharing network and client related information with the VSP. The service management workflow is as follows: 1) A client requests video service from the VSP-M. 2) VSP-M informs the NSP about the service request with source and destination IP addresses. 3) NSP computes the shortest path route for the client. 4) NSP sends the most recent list of slices reserved to VSP on all links along with the VSP client set on each slice. Note that reserved VSP slice on each link isolates VSP clients from other clients. 5) VSP-M computes the new requested reserved bitrate for each slice given the most recent client set on each slice. 6) NSP updates the reserved VSP slice capacities on each link. 7) The VSP-M determines and informs groups of clients that will collaborate with each other to compute their fair-share bitrate. 8) Clients perform TCP receive window control and DASH video rate adaption.

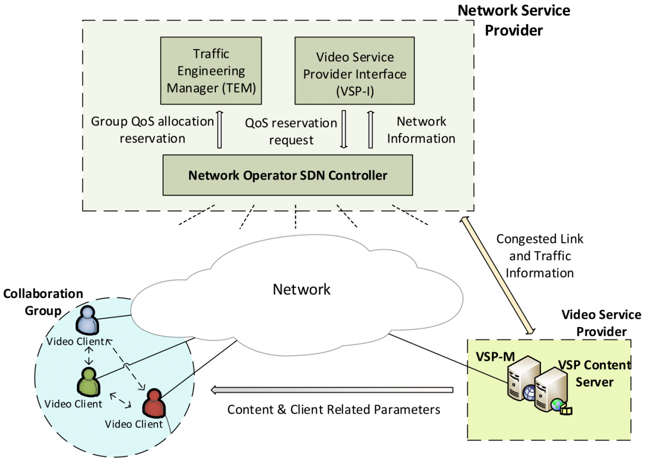

Fig. 1.The proposed distributed managed-service architecture.

III-B. Dynamic Formation of Collaboration Groups

Collaboration groups refer to those clients sharing a network link/slice over a particular link. In addition to receiving slice-state information from the VSP, clients in the same group exchange their buffer status information so that they can compute their group-buffer-aware fair-share bitrates. Collaboration groups are formed by the VSP using Algorithm 1 dynamically considering entry/exit of clients at random times.

The input of Algorithm 1 is the slice/link set $L = \{l_j\}$ shared by the VSP at Step 1. Each slice l_j has a reserved capacity C_j and the set of clients D_j on that slice as features. Steps 2-3 set the initial qV and determine fair-share bitrates r_i for different video resolutions $i = 1, \ldots, M$ for this qV, respectively, and the equation at Step 3 is explained in [3], [16]. In the while loop, Steps 5-7 calculate the congestion ratio CR_j of each slice l_j, where CR_j is defined as the sum of fair-share bitrates of all clients over the slice j, divided by the reserved capacity C_j, and N_i^j refers to the number of clients with resolution type i over slice j. Step 8 sorts slices/links in decreasing order of congestion ratio values. Steps 9-14 exclude clients that are already members of a collaboration group over a slice with a higher congestion ratio from subsequent collaboration groups and deduct their bitrates from the slice capacities C_j for all other j accordingly. Step 15 notifies all clients over slice j about their group membership and their reserved capacity C_j. Since the network state and number of DASH clients vary over time, Step 16 repeats the grouping procedure periodically every Δt seconds.

Algorithm 1: Dynamic Grouping @VSP

```
1  Input: $L = \{l_j\}$, $j = 1, \ldots, N$
2  $qV = 0.96$;
3  $r_i = (\frac{qV - c_i}{a_i})^{\frac{1}{b_i}}, i = 1, \ldots, M$;
4  do
5      foreach slice $l_j$ in $L$ do
6          Count the number $N_i^j$ of clients on $l_j$ with video
             resolution $i = 1, \ldots, M$;
7          Compute congestion ratio $CR_j = \frac{\sum_{i=1}^{M} r_i \cdot N_i^j}{C_j}$;
8      Sort links in $L$ based on $CR$;
9      Create $D = \bigcup_{j=1}^{N} D_j$;
10     Initialize an empty set $D^*$;
11     foreach slice $l_j$ in $L$ do
12         Remove client ids in $D^*$ from $D_j$;
13         Adjust $C_j$ considering the removed clients;
14         Add client ids in $D_j$ to $D^*$;
15         Notify clients in $D_j$ about their group and $C_j$;
16     Sleep for $\Delta t$ seconds;
17 while true;
```

III-C. Distributed Fair-Share Bitrate Calculation

Each collaborative DASH client computes its own fair-share bitrate using Algorithm 2. The VSP informs each client about the number of clients N_i with video resolution i, $i = 1, \ldots, M$, where M is the number of different resolutions, video quality coefficients, a_i, b_i, c_i (given in Table 2 of [3]) and reserved capacity for the group C. In addition, clients periodically share their buffer status, in terms of duration d_{ij} for all clients j with resolution i, with each other. The outputs of Algorithm 2 are the fair-share bitrate r_i for clients with resolution type i, and the weights α_{ij} related to the buffer status of client j with resolution type i.

Algorithm 2: Distributed Fair-Share Bitrate @Clients

```
1  **Inputs:** $a_i, b_i, c_i$ $C$, $d_{ij}$, $i = 1, \ldots, M, j = 1, \ldots, N$
2  **Outputs:** $r_i$, $\alpha_{ij}$
3  $\epsilon = 100$;
4  $\Delta qV = 0.01$;
5  $qV = 0.8$;
6  $total = 0$;
7  $\bar{d}_i = \frac{\sum_{j=1}^{N_i} d_{ij}}{N_i}, i = 1, \ldots, M$;
8  **while** $|C - total| > \epsilon$ **do**
9      $r_i = (\frac{qV - c_i}{a_i})^{\frac{1}{b_i}}, i = 1, \ldots, M$;
10     $\alpha_{ij} = 2 - \frac{d_{ij}}{\bar{d}_i}, i = 1, \ldots, M$ and $j = 1, \ldots, N_i$;
11     $W_i = \sum_{j=1}^{N_i} \alpha_{ij}, i = 1, \ldots, M$;
12     $total = \sum_{i=1}^{M} W_i \cdot r_i$;
13     **if** $total < C$ **then**
14         $qV = qV + \Delta qV$;
15     **else**
16         $\Delta qV = \frac{\Delta qV}{2}$;
17         $qV = qV - \Delta qV$;
```

In Algorithm 2, in Steps 3-6, the values of convergence parameter ϵ and step size ΔqV are set, and quality value qV and $total$ are initialized. Step 7 calculates the average buffer fullness $\bar{d}_i$ for clients with video resolution i. We iterate over Steps 8-17 until $|C - total| < \epsilon$. Step 9 calculates the fair-share bitrate for video resolution i for the current value of qV using Eqn. 1 in [16]. Step 10 computes the group-buffer-aware bitrate weight α_{ij} of client j with video resolution i. Step 11 computes the sum W_i of weights for all N_i clients with video resolution i. The current value of $total$ bitrate allocated to all clients is calculated in Step 12 as the sum of products of the fair-share bitrates r_i and weights W_i. In Steps 13-17, if the current value of $total$ bitrate is less than capacity C, qV is incremented by the step size ΔqV. Else, the step size is halved, and qV is decremented by the new step size. At convergence, each client j with video resolution i computes its group-buffer-aware fair-share bitrate as $r_i \cdot \alpha_{ij}$, updates its DASH adaptation rate according to this fair-share bitrate, and sets its TCP receive-window size as $r_i \cdot \alpha_{ij} \cdot RTT$.

IV. EVALUATION

IV-A. System Setup

We evaluate the performance of the proposed system over an SDN test-bed, whose data plane consists of a partial mesh network generated by Mininet [17] using OVS switches, which are connected to a Floodlight [18] controller. In our implementation, TEM and VSP expose resource allocation and NSP collaboration functionalities, respectively, as web services employing JSON based APIs.

Table I. Resolutions and encoding parameters for test video.

Content	Resolution	Bitrates (Mbps)
Tears Of Steel	1080p	1.5 , 2.4, 3, 4, 6, 10
	720p	0.5, 0.8, 1.5, 2.4
	360p	0.5, 0.8
Big Buck Bunny	1080p	1, 1.2, 1.5, 2.1, 2.4, 2.9, 3.3, 3.6, 3.9
	720p	0.5, 0.57, 0.78, 1, 1.2, 1.5
	360p	0.37, 0.50, 0.57

The VSP is represented by one of the hosts that runs a management and a content server using HTTP 1.1 with persistent connections. Content server stores the MPD manifest files and test videos given in Table I with the segment durations of 4 seconds. Our DASH client uses Java sockets for communication with the VSP-M and group members, and also to set calculated receive buffer sizes. We estimate round-trip time RTT as the time difference between the instant when a new segment request is sent to the content server and the instant when the first byte of the response is received.

IV-B. Results

We presented several results on the performance of the proposed VSP-managed distributed-collaborative DASH service using the video *Tears Of Steel* in [16], where we reported the mean and variance of video quality qV, as well as the mean number of stalls (MNS) and the mean duration of stalls (MDS) over 10 repetitions of each experiment. In this paper, we report a new experiment using two different videos shown in Table 1, where we have three heterogeneous clients, one 1080p, one 720p, and one 360p client, and the capacity of the network slice reserved for these clients is managed by the NSP dynamically depending on the volume of other traffic.

In the first experiment, all clients start streaming *Big Buck Bunny* with an initial slice capacity of 5 Mbps. After 120 seconds, the NSP reduces the slice capacity to 4 Mbps. Then, after 300 seconds, the NSP informs the VSP that the reserved slice capacity has increased to 6 Mbps.

In the second experiment, all clients start to stream *Tears Of Steel* with an initial slice capacity of 7.5 Mbps. After 120 seconds, the NSP reduces the slice capacity to 6.5 Mbps. After 300 seconds, the slice capacity increases to 8.5 Mbps.

In both tests, clients adjust their fair-share bitrate, DASH adaptation rate, and TCP receive-window size as slice capacity varies. It can be seen from both Table II and Table III that collaborative DASH clients provide better QoE-fair mean qv values that are closer to each other, and in the distributed-collaborative system 1080p client does not suffer from a low mean qv value as the 1080p competitive DASH client does. Furthermore, collaborative DASH clients achieve smoother quality value as can be seen in Figure 2b and 2c compared to those of competitive clients shown in Figure 2a and 2c. Moreover, collaborative DASH clients outperform competitive

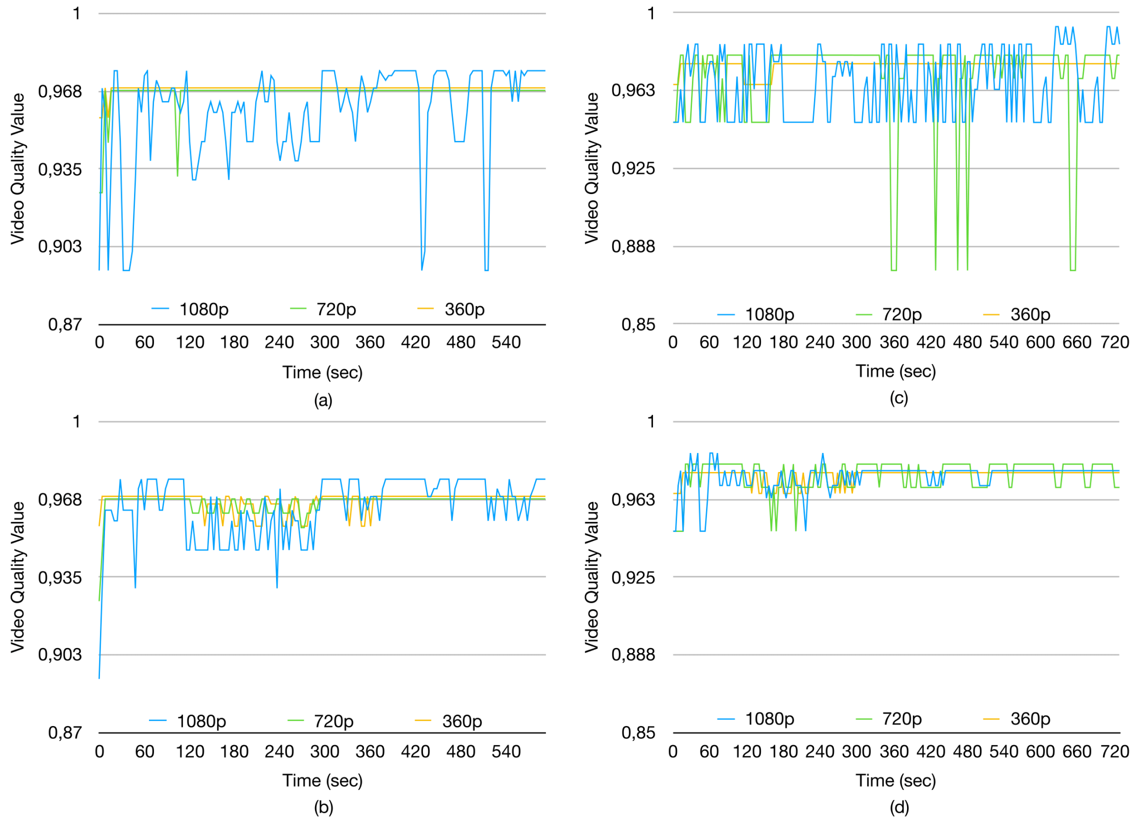

Fig. 2. Comparison of competitive and collaborative DASH over a variable capacity slice: a) competitive and b) collaborative clients streaming Big Buck Bunny; c) competitive and d) collaborative clients streaming Tears Of Steel

Table II. Three heterogeneous clients streaming Big Buck Bunny with 4 sec. segments over a variable capacity slice

Method	Resolution	Mean QV	Var QV	MNS	MDS
Competitive	1080p	0,9558	0,0005	7	17,5
	720p	0,9664	0,00005	1	0,75
	360p	0,9684	0,000004	0	0
VSP-Managed (Distributed)	1080p	0,9646	0,0001	2	0,5
	720p	0,9662	0,00002	0	0
	360p	0,9669	0,00001	0	0

Table III. Three heterogeneous clients streaming Tears Of Steel with 4 sec. segments over a variable capacity slice

Method	Resolution	Mean QV	Var QV	MNS	MDS
Competitive	1080p	0,9652	0,0002	12	46,92
	720p	0,9699	0,0004	8	18,48
	360p	0,9744	0,000009	0	0
VSP-Managed (Distributed)	1080p	0,9735	0,00004	4	3
	720p	0,9739	0,00006	2	0,8
	360p	0,9741	0,00001	0	0

clients in 720p and 1080p videos, when the mean number of stall and mean duration of stall values are considered.

V. CONCLUSIONS

When video services for a group of clients are provided over QoS-reserved network-slices, we observe that QoS reservation over a shared network slice alone is not sufficient to provide steady and fair QoE to heterogeneus DASH clients due to the nature of TCP congestion control mechanisms. Thus, we propose an SDN-assisted VSP-managed collaborative video service architecture to calculate the group-buffer-aware fair-share bitrate for each client, and enforce this rate by TCP receive-window control at each client as well as setting the application-level DASH video adaptation rate accordingly. VSP-managed distributed collaboration with dynamic grouping of clients is a novel service model which is an alternative to today's OTT service model to achieve QoE stability and fairness in the presence of heterogeneous DASH clients under varying network and client entry/exit conditions. Experiments show that the proposed VSP-managed distributed collaboration provides smooth QoE with low quality fluctuations and outperforms competitive DASH clients in stall and quality variability results.

ACKNOWLEDGEMENTS

This work has been supported by the TUBITAK project 115E299. A. M. Tekalp also acknowledges support from Turkish Academy of Sciences.

VI. REFERENCES

[1] ISO/IEC 23009-1:2014 “Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats,” MPEG International Standard, 2014.

[2] S. Akhshabi, L. Anantakrishnan, A.C. Begen, and C. Dovrolis, “What happens when HTTP adaptive streaming players compete for bandwidth?” in ACM Proc. of Int. Workshop on Network and Operating System Support for Digital Audio and Video, 2012, pp. 9-14.

[3] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, “Towards network-wide QoE fairness using openflow-assisted adaptive video streaming,” in ACM SIGCOMM, New York, 2013, pp. 15-20.

[4] H. Nam, K. Kim, J. Kim and H. Schulzrinne, “Towards QoE-aware video streaming using SDN,” in Global Communications Conf., 2014.

[5] E. Liotou, G. Tseliou, K. Samdanis, D. Tsolkas, F. Adelantado and C. Verikoukis, “An SDN QoE-service for dynamically enhancing the performance of OTT applications”, in Quality of Multimedia Experience, 2015.

[6] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, “SDN-based QoE optimization for HTTP-based adaptive video streaming,” in Int. Symp. on Multimedia, 2015.

[7] A. Bentaleb, A. C. Begen, and R. Zimmermann, “SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking,” Proc. ACM Conf. on Multimedia, pp 1296-1305, 2016.

[8] M. Barbera, A. Lombardo, C. Penarello, and G, Schembra, “Active window management, an efficient gateway mechanism for TCP traffic control”, IEEE Conf. on Communications (ICC), 2007.

[9] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, “Server-based traffic shaping for stabilizing oscillating adaptive streaming players,” in ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 19-24, 2013.

[10] J. Jiang, V. Sekar, and H. Zhang, “Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE,” in Int. Conf. on Emerging Networking Experiments and Technologies, France, 2012, pp. 97-108.

[11] C. Zhou, L. Chia-Wen, and Z. Guo, mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming, IEEE Trans. on Multimedia, vol. 18, no. 4, pp. 738-751, 2016.

[12] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, “A survey on quality of experience of HTTP adaptive streaming,” IEEE Comm. Surveys and Tutorials, vol. 17, no. 1, pp. 469 - 492, Sep. 2014.

[13] O. Oyman and S. Singh, “Quality of Experience for HTTP Adaptive Streaming Services”, IEEE Communications Magazine, no. 4, 2012.

[14] C. Timmerer, M. Maiero, B. Rainer, S. Petscharing, D. Weinberger, C. Mueller, and S. Lederer, “Quality of experience of adaptive HTTP streaming in real-world environments,” IEEE Comsoc MTC E-Letter, vol. 10, no. 3, May 2015.

[15] Z. Wang, L. Lu, and A. C. Bovik, “Video quality assessment based on structural distortion measurement,” Signal Processing: Image Communication, special issue on objective video quality metrics, vol. 19, Jan. 2004.

[16] K.T. Bagci, K. Sahin, and A.M. Tekalp, “Compete or collaborate: Architectures for collaborative DASH video over future networks,” IEEE Trans. on Multimedia, vol. 19, no. 10, pp. 2152-2165, October 2017.

[17] “Mininet Overview.” [Online]. Available: `http://mininet.org/overview`

[18] “Floodlight Controller.” [Online]. Available: `http://www.projectfloodlight.org/floodlight`