Improving Quality of HTTP Adaptive Streaming with Server and Network-Assisted DASH

Reza Shokri Kalan
Digiturk beIN Media Group, Istanbul, Turkey
reza.shokrikalan@digiturk.com.tr

***Abstract*—Having a high-level overview of network resources, traffic pattern and return feedback to clients help them to adapt to the appropriate video quality, and ultimately achieve better Quality of Experience (QoE). Network-assisted approach's offer flexibility in traffic engineering and resource management while it has an overview of the network condition and client distribution. Leveraging flexibility of Software Defined Networking (SDN) combined with Server and Network-Assisted DASH (SAND) features allows enhance the quality of video streaming. This study provides a network-assisted approach for improving QoE in Over-the-Top (OTT) applications with exchanging network and client information. We show that in cooperation between the SDN controller and DASH Aware Network Elements (DANEs), network resources are efficiently used. Consequently, message exchange between network elements and manage the video quality requested by the client through a central controller along with prefetching specific segments at the edge, and directing clients to the appropriate edge can improve the video quality.**

***Index Terms*—Adaptive streaming, SDN, DANE, CDN,**

I. Introduction

Over-the-Top (OTT) traffic already represents the main part of the daily IP traffic. While user demand high-quality video with fewer startup delays and less buffering, the delivery cost is a big concern on the network side. Moving toward HTTP chunk-based streaming technology allows content provides benefit from existing Content Delivery Networks (CDN) infrastructures for delivering and reduces dissemination costs. This architecture benefits from pull technology, where distributed caches store contents at the edge points and clients connect to edge and download content. Despite the popularity of HTTP Adaptive Streaming (HAS) technology, HAS still faces challenges in Quality of Experience (QoE) parameters; (i) *Fairness*: considering content and device characteristics and fairly sharing network resources between clients competing for bandwidth. For example, mobile devices need less bandwidth compared with big screen devices such as smart TV. (ii) *Stability*: providing consistent quality is one of the main challenges in HAS technology. Switching between bitrates negatively affects users' QoE. (iii) *Resources utilization*: network resources must be used efficiently to achieve overall performance, fair and stable service.

Client-side adaptation algorithms have limited information about network conditions and other clients' behavior. Thus, can perform unfairness, instability, and consequently under-utilization of network available bandwidth. Accordingly, in a lack of full control on client behaviour, service providers may not be able to satisfy user expectation and avoid user abandonment. The performance of video streaming applications mostly depends on network conditions and client distribution patterns. The tendency to network-assisted streaming technology eliminate the limitation of client-driven architectures by providing an overview of the network and clients.

Software Defined Networking (SDN) as a current and future network trend can provide network supports to HAS systems and enhance the performance of video streaming applications. The motivation behind this study is to address those challenges with assisted of centralized SDN controller and leveraging MPEG-DASH introduced Server and Network Assisted DASH (SAND). The SAND exchanged messages can be used by the SDN controller for enforcing or enhancing network-assisted streaming strategies (e.g., bandwidth utilization, bitrate adaptation). This orchestration makes possible to utilize CDNs resources in cooperation manner and improves QoE parameters by forwarding client request to appropriate CDNs. This needs real-time information about network conditions and clients statistics. This study introduces SDN-assisted CDNs cooperation where the central controller has an overview of network condition and available recourse, as well as the authority to control communication. Our previous study [1] focused only on switching between origin servers (which includes all available content), while this study focused more on forwarding client requests to a convenient cache on the edge that stores part of the content.

The rest of the paper is organized as follows: background and related works are given in Section II. The details of the proposed architecture, and the client implementation are provided in Section III. The test environment and comparative performance results are discussed in Section IV. Finally, Section V concludes the paper.

II. Background and related Works

A. Background

In the HAS technology, a single-bitrate video file encoded with multi-bitrate and segmented into fixed small size chunks in which each segment carries T_s time unit of video. Transforming single-bitrate file into different bitrates *representations* enable clients easily adjust to suitable video quality during the streaming time. A manifest file keeps the information about encoded video files and representations properties. Streaming video by the client starts with downloading the

978-3-903176-36-2 ©2021 IFIP

manifest. The quality of streaming depends on the quality of requested representation over time. MPEG-DASH introduced the SAND technology for the purpose of assisting clients and improving the efficiency of streaming sessions [1]. SAND introduces a message exchange mechanism and enables asynchronous network-to-client and network-to-network communication. With reference to QoE, SAND enables the client's performance and statistics to the controller and provides network real-time conditions to the client. Besides DASH client and regular network elements, DASH-Aware Network Elements (DANEs) and metric servers are also defined in the SAND technology. DANEs have intelligence about the DASH format and the characteristics. As shown in Fig. 1, DANEs are in charge of gathering status (e.g., buffer level, requested bitrate) from DASH clients. With this insight, DANEs estimate future incoming requests and as a result, make an efficient caching decision. On contrary, DANEs can send Parameters for Enhancing Reception (PER) messages to a client which helps client achieves better adaptation. Also, DANEs might predict and request further segments from the origin server and prefetch them. In this direction, DANEs send Parameters for Enhancing Delivery (PED) messages to the server.

Leveraging edge caches, where a copy of the content stored in geographically distributed PoPs (Points of Presences) eliminates the massive number of requests sent to the origin server and thus reduces latency. Regrading, in the case of suffering from overload in a certain geography, the OTT can provide extra bandwidth to make sure that everything is running smoothly. Having a global view of network architecture and resources can help clients to better adjust to high quality video by downloading video segments from suitable CDN. This also has a positive impact on total network performance. In addition, from the commercial point of view, the OTT providers have the flexibility for routing traffic via more economic CDN to the path (1).

$$Maximizing\ (Perceived\ quality/Delivery\ cost) \quad (1)$$

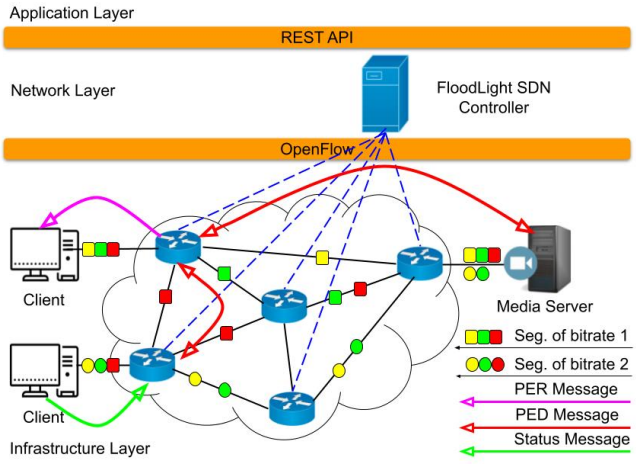

Fig. 1. System architecture and SAND message passing

Our solution relies on a centralized decision system that has control authority over the local decisions. Note that our proposed system does not control the origin servers or distribution logic. Rather it acts as an additional management layer to enable efficient resource management, improves the delivery systems, and assist clients to achieve better quality. The key insight behind this system is that the controller needs updated information about the network, client, and CDN status to make a network overview and effectively assist the client to choose the appropriate CDN and bitrate.

B. Related Works

In the relevant literature, the network-assisted approach orchestrates clients and triggers them with up-to-date network information in order to make effective use of network resources and better quality adaptation. The authors of [2] proposed a coordinated internet video control plane that can leverage global view of client and network conditions to dynamically optimize the video delivery. The authors in [3] proposed centralized and distributed architectures for collaboration between DASH clients and show that collaboration between clients helps them achieve a smooth bitrate.

Dealing with the subject of SDN-assisted adaptation, the authors of [4] considers SDN controllers in the network with a global view on the network activity. The proposed system improves QoE by providing two mechanisms for adaptation assistance: (i) explicitly signaling target bitrates to DASH players and (ii) network traffic management. However, it only enables stable streaming and fair sharing of network resources between DASH players, it does not prevent DASH streams to outperform in a network that has background traffic. To actuate the optimal solution in an SDN network, the authors of [5] compared bandwidth allocation and bitrate guidance. Experimental results show that bandwidth allocation improves the average video quality. But, bitrate guidance ensures fair video quality. As in the previous study, this study also does not consider background traffic.

To address client heterogeneity and scalability, authors in [6] introduced intelligent streaming architecture (SDNHAS) that leverages SDN capabilities of assisting HAS players in making better adaptation decisions. In order to load balancing and improving quality, OTT allows clients to switch between CDNs. Jiang et al. [7] introduced a a scalable predictive analytics system namely Critical Feature Analytics (CFA), to improves the QoE for Internet video applications. Compared to many prior efforts, CFA can provide near real-time quality estimates. Different from these studies listed here, we focus on utilizing centralisation approaches for video streaming and cache cooperation between multi-CDN.

To cope with resource limitations, various studies have proposed virtual CDN approaches. Authors in OPAC [8] consider the vCDN migration problem in a network or instantiates a new vCDN on demand to satisfy user quality requirements. Migration cost is the key concern of this study. However, same as the previous work there are no streaming quality metrics that represent end-users' performances. In our previous

works [9] [10] we are benefiting from the emergence of virtual cache (vDANE) in order to improve delivered video quality. However, those proposed works are more focused on optimal virtual cache location, rather than prefetching or cache cooperation which is the main topic of this study.

In [11], the authors proposed a new Video Delivery Network (VDN). Compared with a legacy system that clients adapt bitrates independently, VDN restrict bitrates automatically as they have the best view of the current resources and delivery costs. Author's in MacoCache [12] introduced intelligent video caching at the network edge (base station) instead of CDN to improve video streaming QoE. In this study, edge base stations can cooperate with each other and tries to increase the hit ratio at the edge. Same as previous work, authors in [13] investigate user behavior and request patterns in mobile video systems. However, both studies only applied to mobile devices and do not cover user and network heterogeneity. The proposed device-to-device cache cooperation [14] between mobile devices has the same limitation and only increases the QoE of the mobile client.

III. System Architecture Overview and Behavior

A. System Architecture Overview

The SDN enables applications such as traffic engineering and load balancing to define forwarding policies that are eventually translated to southbound-specific instructions. The network controller continuously monitors the network environment and available resources. In the event of a change in network resources and traffic pattern, the controller consequently responds appropriately. In our system architecture, the SDN controller gathers network information from network nodes (e.g., switches, servers, clients) using its southbound interface. Clients and DANEs, periodically send quality-related information to the controller, including the average quality of the received video representations requested by the clients. The controller retrieves this information and determines the upper bound for quality bitrate by leveraging two main functions:

- *Quality Decision*: pure client-side adaptation does not provide fair quality because of clients' greedy behavior. To eliminate such behavior, we introduced a qualıty decision function with two main responsibilities: (i) accurately collect quality metrics from clients, (ii) decision making and update clients. Based on quality parameters observed by clients' and with assisted of DANEs, the quality decision makes real-time decisions in response to client behaviors. This function triggers DANEs for enforcing/ enhancing to change quality or CDN.
- *SAND and DANE*: DANE has intelligence about the DASH property, therefore, it can be used for enhancing QoS/QoE parameters. Forwarding client requests to suitable CDN (or cache) significantly decreases redundant network traffic. Also, auditing the client's request bitrate with the assistance of the DANE's PER messages can result in fair and better quality.

When a client-side adaptation algorithm computed the appropriate bitrate, enforced by the quality decision to adaptable quality. This approach provides fair quality. It is worth emphasizing that, the quality decision only defines the upper bound limit. Therefore, the client can request a lower bitrate in case of buffer draining. As shown in (2), when player buffer fullness gets below a threshold, a client can quickly fill it by ignoring the suggested bitrate.

$$Bitrate = min\ (Client\ adaptation, Quality\ decision) \tag{2}$$

B. System Behavior

Typically, clients connect to the closest CDN (edge) and start streaming by requesting segment s of video at adapted bitrate b. If there is an entry for (s , b) at the edge, the video segment returned to the client. Otherwise, the central controller decides to forward it to another peer CDN or origin server. At the beginning of a new video streaming, there is no entry for the request (s , b) at the edge points. Thus, the controller forwards incoming requests toward the origin server. The origin server just-in-time packaging the requested content and return it back to the client in the reverse path. At the same time content cached in the related CDN. Also, the controller updates the entry of related content for addressing future requests. If more than one request reaches the edge for the same content at the same time, in the missing scenario, only one of them is sent back (origin) and the other requests are answered when the edge is updated with the new request. This reduces both network redundant traffic and server load.

SAND technology introduces an opportunity for the controller to use a real-time measurement of client performance. These statistics help the controller to create a model for better performance. As shown in *Algorithm 1*, different types of SAND messages are used to assist clients to achieve expected quality. A client has a few seconds buffering capacity. Therefore, if the client's current buffer level is below the threshold, the buffer should be filled quickly so that the buffer does not drain out. In such a case client request video from the lower representation which has minimum bitrate and quality. Otherwise, client requests affected by quality estimated by quality decision function (lines 2-12). Furthermore, SAND can lead clients to connect to a specific CDN to achieve better video quality. Additionally, this allows for bandwidth reservations (lines 15-19). Exchange information between DANEs helps efficiently forward client requests to a CDN that has cached requested content (lines 21-23).

IV. Experimental Results

A. Test Environment

To measure the performance of the proposed system, we implemented simulation using the $Mininet$ emulator. Further, we used *FloodLight* and *OpenFlow* as a controller and southbound interface respectively. For network topology, we applied *BellCanada* topology from the Internet Topology Zoo [15]. A $Poisson$ distribution with (λ=30 Mbps) is used for generating the network links bandwidths. The number of DASH clients is set to 100. We compared the proposed system with the

Algorithm 1: Assisted DASH adaption algorithm.

```
Input: qDane: rep.id received from DANE
       qClient : rep.id determined by the client
       qRequest : HTTP GET to selected server
let rep.id point to representation ID;
foreach HTTP GET request do
    switch message type received from the SAND do
        case assisted message do
            if (buffer.level >= threshold) then
                if (qClient >= qDane) then
                    qRequest ← qDane;
                else
                    qRequest← qClient;
                end
            else
                qRequest← qClient;
            end
            return qRequest;
        end
        case enforced message do
            if (message carries an appropriate CDN )
             then
                connect_to_CDN();
                continue streaming;
            end
        end
        case PED message do
            DANE exchange information DANE;
        end
    end
end
Output: Optimal video representation
```

conventional adaptation with autonomous clients, and BOLA [16]. In the BOLA and proposed assisted DASH, bitrate adaptation authorized by a central controller.

The video *Big Buck Bunny* is used for streaming during the simulation. The video contains 299 video segments with an equal length of 2 seconds video, which encoded with six different quality or representations. While the first representation (R1) has the lowest quality, the last representation (R6) has the highest quality. To achieve better video quality, clients need to get more segments from higher bitrate, while reducing the number of video stalls. The buffer capacity of each client is 24 seconds. During simulation, the client buffers at least 4 segments (8 seconds of video) before playing. This helps the client not to experience video buffering immediately after loading and playing the first segment. Therefore, clients wait for a short time to fill buffers. Of course, increasing this threshold will lead to less buffering time, but it will make users wait longer to start the video. During the simulation, an attempt has been made to make the network dynamic. Clients alternately log in and out of the network. The network also carries cross traffic. In all algorithms, client attachment points are randomly distributed over the network. Each simulation is repeated 10 times and the average values are presented in graphs and tables.

TABLE I
OBSERVED QUALITY OF EXPERIENCE METRICS

Approach	Startup delay (s)	Buffering (s)	Quality (Kbps)
Assisted DASH	5.2	30	3385
Conventional	5.9	102	2958
BOLA	5.3	39	3274

B. Quality of Experience

Table I list averaged quality parameters include startup delay, buffering duration, and video quality observed on the clients' side obtained from the simulations. It is seen assisted DASH has the lowest startup delay and buffering, while conventional approach experienced more startup delay and buffering. The reason for this is the exchange of PER messages between DANE and DASH client in order to send client requests to the best edge. DANE also predicts subsequent segment and requests it from the origin server and stores in caches. DANEs can also use PED messages to inform each other about available stored content in their own edge cache. Therefore, if 'Miss' happens, DANE will forwards the client's request to the appropriate edge that has cached the requested content. This avoids forwarding a client request toward the origin server located in log distance and reduces delay.

In the conventional adaptation, a client has autonomous adaptation decisions, so if there are more video segments in the buffer, the client has the freedom to request the next segments at the highest bitrate. This greedy behavior has two drawbacks. Firstly, network instability and link congestion for a short period of time. Second, unfair bandwidth allocation. The adaptation mechanism forces the client not to exceed a certain bitrate. In the context of video quality, assisted DASH clients also experience higher quality. Outlining that higher bitrate by itself does not mean better quality. To achieve satisfied QoE in video streaming, it is necessary to take startup delay and buffering duration into account as well. The observed value indicates that the proposed assisted DASH achieves better results in all metrics.

A further valuable factor to illustrate the effectiveness of assisted DASH in received video quality is a distribution of the received representation. Fig. 2 shows the distribution of the received representations. Compared with the conventional approach, BOLA and assisted DASH clients received fewer segments from the highest representation. It is because in order to provide fair bandwidth allocation and resource management, the controller has authority over client adaptation and eliminates clients' requests from the highest bitrate who has fulled buffer. At first glance, it seems that conventional clients displayed video in higher quality, but it also received more segments of the video in lower quality, which indicates more quality fluctuation. On the contrary, assisted DASH has more efficient quality adaptation with better representations distribution such that clients display video at a higher bitrate. As mentioned, despite more buffering and interruptions

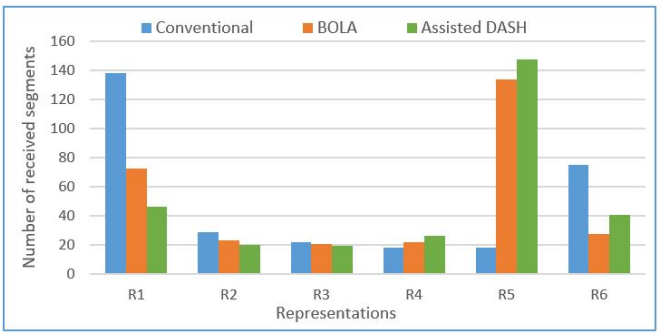

Fig. 2. Number of segments received per representation

in display time, even receiving high-quality video does not guarantee QoE performance. By considering this fact and the observed buffering results given in Table I, we can conduct that the assisted DASH achieves better performance and clients display video with better quality on average. This shows that the concept of a high-level network overview has a positive effect on the received video quality.

C. Prefetching and Server Efficiency

The concept of prefetching allows CDNs to predict and fetch a video at the edge before being requested by the player. As a result, clients experience better video quality. While edge caches have limited storage capacity (30 s), assisted DASH prefetches only segments whose bitrate is the same as the bitrate recommended by the quality decision function. While quality decision function defines the highest available bitrate for adaptation, clients request almost the same representation that its high probability is available in the cache. Thus, prefetching only this bitrate enhances cache efficiency when the cache has limited capacity.

Remind that packing content on the origin server is a costly task. When more request comes to origin server it can become a bottleneck and results to more response time. The video player has only a few seconds of buffer capacity, therefore bandwidth fluctuation or late response time can cause buffer drain out. In the context of server offload, while conventional and BOLA approaches have reduced 63% and 71% of origin server load respectively, the assisted DASH achieved better results by reducing 76% of origin server load.

V. Conclusion

Determining desirable target quality to be reached by each stream and then burdening the clients to stay within their limits provides benefits in terms of stability and low oscillation between bitrates. With a SAND assistance each edge cache can retrieve requested segment form its neighbors which is faster and bandwidth-effective than requesting from origin servers. Considering this architecture and following local hit and CDN hit steps, fetching from origin server has the lowest priority.

Our motivation question was whether it is possible to achieve high video quality where CDN infrastructure is being stressed by increasing traffic. To end this we implemented CDNs cooperation with assisted SAND architecture introduced by MPEG-DASH. Exchange information between clients and DANEs, and enforcing network-assisted streaming strategies can be leveraged by central controller. Based on network condition or clients context, the central controller with leveraging SAND enables enforcing and or enhancing client to quality adaptation. Furthermore, this central mechanism has authority in routing algorithm to save bandwidth. This is done by considering both the network available bandwidth and clients' distribution.

Our proposed study deep into network assist adaptation, where SAND technology supports real-time decision platform for optimizing streaming video. In this study, we attended to redirect clients to appropriate CDNs without considering 'token' authentication. To develop this work in the future, we intend to consider DRM and token authentication.

Acknowledgment

This study has been supported by Digiturk beIN Media Group, close cooperation with $R\&D$ team.

References

[1] R. S. Kalan, M. Sayit, and A. C. Begen, "Implementation of sand architecture using sdn," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2018.

[2] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," in *Proceedings of the ACM SIGCOMM*, 2012.

[3] K. T. Bagci, K. E. Sahin, and A. Tekalp, "Compete or collaborate: Architectures for collaborative dash video over future networks," *IEEE Transactions on Multimedia*, vol. 19, pp. 2152–2165, 2017.

[4] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An sdn architecture with dash assisting network elements," in *Multimedia Systems*, 2016.

[5] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and experimental evaluation of network-assisted strategies for http adaptive streaming," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016.

[6] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "Sdnhas: An sdn-enabled architecture to optimize qoe in http adaptive streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.

[7] J. Jiang, V. Sekar, H. Milner, D. Shepherd, I. Stoica, and H. Zhang, "{CFA}: A practical prediction system for video qoe optimization," in *NSDI*, 2016, pp. 137–150.

[8] H. Ibn-Khedher, M. Hadji, E. Abd-Elrahman, H. Afifi, and A. E. Kamal, "Scalable and cost efficient algorithms for virtual cdn migration," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*.

[9] S. Clayman, R. S. Kalan, and M. Sayit, "Virtualized cache placement in an sdn/nfv assisted sand architecture," in *2018 IEEE International Black Sea Conference on Communications and Networking*. IEEE, 2018.

[10] R. Kalan, M. Sayit, and S. Clayman, "Optimal cache placement and migration forimproving the performance of virtualized sand," 2019.

[11] M. K. Mukerjee, D. Naylor, J. Jiang, D. Han, S. Seshan, and H. Zhang, "Practical, real-time centralized control for cdn-based live video delivery," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 311–324.

[12] F. Wang, F. Wang, J. Liu, R. Shea, and L. Sun, "Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach," in *IEEE INFOCOM 2020*.

[13] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, 2017.

[14] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, "Cache less for more: Exploiting cooperative video caching and delivery in d2d communications," *IEEE Transactions on Multimedia*, vol. 21, no. 7, 2018.

[15] Topology-Zoo. [Online]. Available: http://www.topology-zoo.org, accessed 27-03-2021

[16] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE INFOCOM 2016*.