A Cross-Layer Bandwidth Allocation Scheme for HTTP-Based Video Streaming in LTE Cellular Networks

Stefania Colonnese, *Member, IEEE*, Francesca Cuomo, *Senior Member, IEEE*, Tommaso Melodia, *Senior Member, IEEE*, and Izhak Rubin, *Fellow, IEEE*

***Abstract*— This letter investigates the benefits of flexible resource allocation when performing HTTP-based adaptive streaming (HAS) across cellular systems such as long-term evolution (LTE). To guarantee video fluidity in the presence of fluctuations of the instantaneous video source rate and channel capacity, we consider an HAS-based proxy video manager and resource controller located at the cellular base station. Based on the channel quality observed by mobile clients, the manager allocates the wireless bandwidth to mobile clients for transmitting the video streams. We propose a cross-layer bandwidth allocation scheme that takes into account the channel quality as well as the video quality requirements and encoding rate fluctuations of the HAS video stream and minimizes the transmission delays experienced by users. This cross-layer bandwidth allocation achieves the optimum in terms of HAS streams delays and it outperforms different bandwidth allocations procedures and state-of-the-art LTE schedulers.**

***Index Terms*— 4G mobile communication, multimedia systems, optimal scheduling, quality of service.**

I. Introduction

MOBILE streaming services will be pervasive in 4G and future cellular platforms [1]. In the HTTP Adaptive Streaming (HAS) client-server architecture, the server stores a few versions of the same video content encoded at different average rates and parsed into short duration video "chunks"; the client requests chunks at the average rate that matches the end-to-end throughput [2], [3] and/or playout buffer status [4]. Switching among different rates induces visual quality fluctuations that affect the user Quality of Experience (QoE) [5], [6]. To prevent this effect, in this paper, we consider an architectural model under which a Base Station (BS) (e-NodeB in the LTE system) i) acts as a proxy between the client (User Equipment-UE) and the Video Server, and ii) optimally allocates the users' bandwidth, so as to maintain the users' selected average video rate and the playout buffer status.

The basic system assumptions are:

A1 the BS either partially pre-caches the video data in the local server's proxy or it loads side information on the average rates and the chunks sizes of the video streams;

A2 the BS estimates the average spectral efficiency of the user on a time period of a chunk duration (e.g., 2 seconds). This corresponds to several LTE Transmission Time Intervals (TTI) where periodical channel reports are collected from the UEs, as defined in LTE [7];

A3 the BS optimally allocates the available bandwidth BW_s [Hz], to the different UEs; the allocation acts as a constraint by the actual scheduling algorithms operating at the MAC layer.

The proposed architectural model fits the standard LTE architecture [8]; the LTE Radio Resource Manager operates by collecting the CQIs reported by the UEs (A2), it interacts with the upper layer to receive the bandwidth requests available at the proxy (A1), and then allocates the requested resources (number of Resource Blocks-RB) by also selecting the proper Modulation and Coding Scheme (MCS) (A3). All these operations are only at the BS side, and apart from the interaction with the proxy that provides the actual chunk size per user, all the other operations are fully standard compliant and both the server and the client are unaltered.

In this framework, we propose a new bandwidth allocation mechanism denoted as Dynamic Minimum Average Delay (D-MAD) that periodically operates at a temporal scale of the chunk duration (seconds), and jointly accounts for the users' fluctuating channel qualities to guarantee the targeted video QoE while minimizing the streams' playout delay. We evaluate the D-MAD in case of a LTE cellular system and we show that it outperforms different state-of-the art schedulers [8], [9].

II. Related Work

Several recent works address the problem of video streaming bandwidth allocation to avoid quality variations during the session. Recent approaches use proxy based adaptation mechanisms [10], [11]. The work in [10] uses a DASH proxy to provide scalable video services, whereas [11] optimizes network resource allocation by accounting for contents and channel characteristics and for client playout buffer levels.

As for scheduling of video packets in new generation cellular systems, in [12] the authors clarify key interactions between the application layer and the lower layers in 4G and 5G networks and propose a resource allocator that optimizes the user video quality while limiting application layer quality switches. The paper in [9] addresses progressive streaming via OFDMA systems, and takes into account the QoE of the end users, as expressed by the data stored in the playout buffer, in the user priority at the MAC scheduler. The procedure relies on the accurate knowledge of the current player buffer status, which needs for periodical updating from

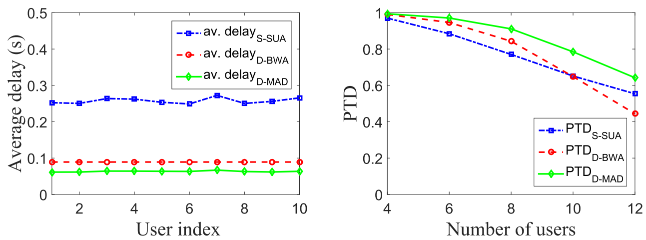
Manuscript received July 31, 2016; revised September 16, 2016 and November 5, 2016; accepted November 9, 2016. Date of publication November 14, 2016; date of current version February 9, 2017. The associate editor coordinating the review of this letter and approving it for publication was S. De.

S. Colonnese and F. Cuomo are with the Department of Information Engineering, Electronics and Telecommunications, University of Rome Sapienza, 00184 Rome, Italy.

T. Melodia is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.

I. Rubin is with the Electrical Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095 USA.

Digital Object Identifier 10.1109/LCOMM.2016.2628378



(a) Average delay (b) PTD

Fig. 1. (a) Average delay vs user index in case of $c \in \{7, 8, 9\}$; (b) PTD vs number of users for $c = 7$.

the client to properly account for encoding rate fluctuations and initial buffering and rebuffering events. Still, the paper confirms that scheduling can be optimized to reduce pauses at the UE. By sharing these principles (proxy based and QoE awareness), we herein adopt the chunk delay as QoE parameter to be included in the resource allocation. The proposed approach operates on a chunk by chunk basis and leverages the bandwidth efficiency that can be achieved with an optimal dynamic allocation [13].

III. Resource Allocation Scheme

We consider a system of N UEs served by the BS controller. Each user selects video contents at quality $q^{(i)}$ associated with a video stream whose net average rate is equal to $R(q)$ [bps]. User i experiences channel quality levels $c^{(i)}$, with $i = 0, \cdots N-1$. At a given time k, a user requests a chunk of video that contains $\lambda_k^{(i)}, i = 0, \cdots N-1$ bytes. The chunk size is inherently random and it depends on the video content and on the actual encoder settings. The bandwidth requested by each user is then computed as $B_k^{(i)} = \lambda_k^{(i)}/(\eta(c_k^{(i)})\alpha\tau)$ $[Hz]$, where $\eta(c_k^{(i)})$ represents the average spectral efficiency attained by user i under its reported channel quality level $c_k^{(i)}$ and α, with $\alpha < 1$, accounts for protocol overhead.

We propose a joint optimal allocation scheme, referred to as the Dynamic Minimum Average Delay (D-MAD). According to D-MAD, the allocated bandwidth, denoted by $\tilde{B}_k^{(i)}$, is selected dynamically chunk by chunk and jointly for all users.

The D-MAD scheduler targets HTTP streaming services with application-layer paced chunk download every τ seconds. With respect to a progressive streaming as in [9], the cumulative receiver buffer delay does not increase unless the actual chunk experiences a delay larger than τ seconds. This nonlinear behavior of delay accumulation is exploited by D-MAD, that does not target maximal video download rates but chunk delays as close to τ as possible. In case of bandwidth requests overcoming the available bandwidth, D-MAD assigns resources in an optimal way, so that the average of users' chunk delays is minimum. With this model the chunk delay δ_i can be expressed as follows:

$$\delta_i = \max\left(\left(B_k^{(i)}/\tilde{B}_k^{(i)} - 1\right)\tau, 0\right). \quad (1)$$

Let us denote by BW_s [MHz] the level of available cell bandwidth. Under D-MAD, the joint allocation process is performed to minimize the following objective function:

$$\sum_{i=0}^{N-1} \delta_i\left(B_k^{(i)}, \tilde{B}_k^{(i)}\right) = \sum_{i=0}^{N-1} \max\left(\left(B_k^{(i)}/\tilde{B}_k^{(i)} - 1\right)\tau, 0\right) \quad (2)$$

while satisfying $\sum_{i=0} \tilde{B}_k^{(i)} \leq BW_s$.

To solve the D-MAD allocation problem we introduce Lagrange multipliers and construct the following functional:

$$\Psi = \sum_{i=0}^{N-1} \max\left(\left(B_k^{(i)}/\tilde{B}_k^{(i)} - 1\right)\tau, 0\right) + \psi \sum_{i=0}^{N-1} \tilde{B}_k^{(i)}. \quad (3)$$

Differentiating w.r.t. $\tilde{B}_k^{(i)}, i = 0, \ldots N-1$, we obtain $\partial\Psi/\partial\tilde{B}_k^{(i)} = -B_k^{(i)}\tau/(\tilde{B}_k^{(i)})^2 + \psi = 0$ standing when $\tilde{B}_k^{(i)} < B_k^{(i)}$ for all i. Substituting $\tilde{B}_k^{(i)} = \sqrt{B_k^{(i)}\tau/\psi}$ into $\sum_{i=0}^{N-1} \tilde{B}_k^{(i)} \leq BW_s$ we compute ψ, so obtaining: $\tilde{B}_k^{(i)} = BW_s \cdot \sqrt{B_k^{(i)}}/\sum_{i=0}^{N-1}\sqrt{B_k^{(i)}}$. As the total bandwidth increases, using the Kuhn-Tucker conditions leads to: $\partial\Psi/\partial\tilde{B}_k^{(i)} = 0$ if $\tilde{B}_k^{(i)} < B_k^{(i)}$ and $\frac{\partial\Psi}{\partial\tilde{B}_k^{(i)}} < 0$ if $\tilde{B}_k^{(i)} \geq B_k^{(i)}$. The solution to the Kuhn-Tucker equations is then found as follows.

The minimal overall delay result is obtained by solving the following bandwidth allocation problem:

$$\begin{cases} \sum_{i=0}^{N-1} \delta_i = \sum_{i=0}^{N-1} \max\left\{\left(B_k^{(i)}/\tilde{B}_k^{(i)} - 1\right)\tau, 0\right\} \\ \tilde{B}_k^{(i)} = \min\left(B_k^{(i)}, \dfrac{\sqrt{B_k^{(i)}}}{\sum_{i=0}^{N-1}\sqrt{B_k^{(i)}}} \cdot BW_s\sqrt{\psi}\right) \end{cases} \quad (4)$$

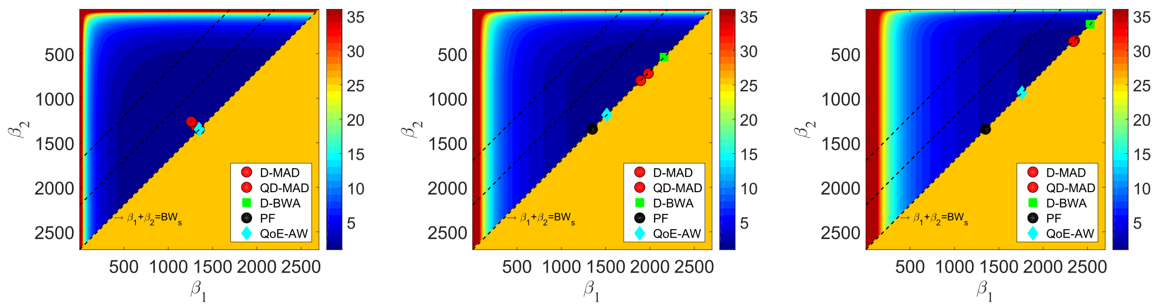
where ψ is such that $\sum_{i=0}^{N-1} \tilde{B}_k^{(i)} = BW_s$.[1]

The main novelties with respect to other cross-layer allocation procedures in the literature are as follows. Differently from progressive streaming approaches (e.g., [9]), D-MAD performs an allocation jointly for all users demanding for the bandwidth at a given time interval τ (chunk interval) by considering their dynamic bandwidth demands, $\tilde{B}_k^{(i)}$ constrained by their QoE target, δ_i, and by the system bandwidth BW_s. Our solution then optimizes the use of the overall bandwidth in a given chunk interval τ and at the same time assures a fair allocation by also considering QoE constraints. Furthermore, with respect to [12], our solution presents three important differences: i) it does not interfere with the rate selection which is left to the client server architecture, according to the DASH standard; ii) it explicitly accounts for the introduced chunk delays, which are a key factor affecting the QoE in HAS streaming; iii) it guarantees that the assigned bandwidth reaches the maximum available value, whereas in [12] the bandwidth allocation is actually realized by the so called enforcer stage, which does not exploit resources unused by a user to improve the rate of other users.

IV. Numerical Results

We first compare D-MAD with an allocation procedure named Dynamic, Bandwidth Weighted Allocation (D-BWA); in this case, the allocated bandwidth $\tilde{B}_k^{(i)}$ is selected dynamically chunk by chunk by assigning $\tilde{B}_k^{(i)} = B_k^{(i)}\left(\sum_{i=0} B_k^{(i)}\right)^{-1} \cdot BW_s$. D-BWA leads to equalization of

[1] The above formulas are implemented recursively until the condition $\sum_{i=0}^{N-1} \tilde{B}_k^{(i)} = BW_s$ is met; D-MAD complexity can be shown to be $O(N^2)$, but depending on the user bandwidth requests it boils down to $O(N)$.



(a) $B_1 = B_2 = 1308$, (b) $B_1 = 803, B_2 = 3209$ (c) $B_1 = 355, B_2 = 5137$

Fig. 2. Average delay per chunk $\overline{\delta}(\beta_1, \beta_2; B_1, B_2)$ (color bar expressed in seconds) versus allocated bandwidth (β_1, β_2) (kHz) for different bandwidth requests of two users.

the delays perceived by the different users, i.e. the delays computed as in Eq.(1) are either 0 or equal for all users, for a given chunk index k. For reference sake, we also consider the results achieved by statically and fairly dividing the bandwidth among the users by setting $\tilde{B}_k^{(i)} = BW_s/N$; we refer to this simple case as Static Single User Allocation (S-SUA) scheme.

We used H.265/HEVC 10.1 encoded videos with a spatial resolution of 1920x1080 [14], [15]. Traces correspond to five movies (*Harry Potter*,HP, *Finding Neverland*,FN, *Lake House*, *Speed*, *Blue Planet*). They are $K = 1200$ chunk long. We considered a video quality q at quantization level 25, resulting in video bit rates of 1.35, 1.34, 1.17, 1.20, 1.08 $Mbps$ each, respectively. The video chunk duration is constant and equal to $\tau = 2s$; conversely chunk sizes in bits are variable and approximately Gamma distributed [16], with mean value equal to 337, 334, 292, 299, 271 $KByte$, respectively. We select $\alpha = 0.64$ to model the cascade of a net throughput of about 80% both at the LTE bearer and at the TCP layer [16]. We consider a LTE BS operating over a $BW_s = B_{LTE} = 20\,MHz$, streaming video towards N users. Each user streams a randomly selected video, randomly cyclically shifted. We use CQIs c in $7-9$ and the associated net spectral efficiencies in [13].

With these positions, we compare D-MAD, D-BWA and S-SUA by measuring the average delay per chunk and the Probability of Timely Delivery (PTD) of chunk packets for each user, computed as the fraction of chunks encountering delay events during the video streaming process. In Figure 1, we plot the aforementioned two metrics. The left figure represents the average delay versus the user index when the users have a spectral efficiency $\eta(c)$ with c randomly selected in $c \in \{7, 8, 9\}$. Each point is obtained by averaging the numerical results over 50 Monte Carlo runs. We recognize that D-MAD outperforms the other schemes in terms of the attained average delay per chunk. Since each chunk carries few seconds of video even reducing the delay of just 0.2s per chunk significantly reduces the depletion rate of the client buffer, so increasing time of uninterrupted playout and improving the QoE. Secondly, we observe that D-MAD outperforms D-BWA and S-SUA also in terms of PTD. As expected, under D-BWA, each user observes the same delay value. The S-SUA exhibits much lower performance.

We then compare D-MAD with two state-of-the art LTE schedulers, namely Proportional Fair (PF) [8] and the QoE-aware (QoE-AW) scheduling in [9]. The former is a classical layered scheme while the latter addresses progressive streaming of video flows without considering the chunk timing and requires knowledge of the client playout buffer level.[2] D-MAD operates at a chunk temporal scale (s) while the two schedulers operate at the TTI scale (ms). For fair comparison, we consider the scheduling over one chunk, i.e. over $N_{TTI} = 2000$ TTIs. Firstly, we consider a toy case of $N = 2$ users, requiring each the HP video at encoding rate of 1351 $kbps$; we analyze three reference spectral efficiency pairs, summarized by (a) $\eta(c^{(1)}) = \eta(c^{(2)}) = 1.033$, (b) $\eta(c^{(1)}) = 1.683$, $\eta(c^{(2)}) = 0.421$, (c) $\eta(c^{(1)})) = 3.883$, $\eta(c^{(2)}) = 0.263$. We implemented PF and QoE-AW under the hypothesis that the video users share a fraction of the system bandwidth $BW_s = 0.6 \cdot 5\,MHz$, that corresponds to $NRB = 15$ RBs. Both the schedulers operate at the TTI/RB time/frequency scale, and we implemented a Quantized version of D-MAD (QD-MAD) allocating to each user the integer number of RBs closest to the D-MAD solution.

We plot in Fig.2(a)-2(c) the average delay $\overline{\delta}(\beta_1, \beta_2; B_1, B_2) = \tau \sum_{i=1}^{2} \max\{(B_i/\beta_i - 1), 0\}$ experienced by 2 users that attain from the scheduler the allocated bandwidths β_1, β_2 when they request bandwidths B_1, B_2, for the three cases (a) $B_1 = B_2 = 1308\,kHz$, (b) $B_1 = 803$, $B_2 = 3209\,kHz$, (c) $B_1 = 355$, $B_2 = 5137\,kHz$, respectively. The allocated bandwidth constraint is $\beta_1 + \beta_2 = BW_s$, represented by the dashed slanted straight-lines for different values of BW_s. The bandwidth pairs allocated by the schedulers lie on the line $\beta_1 + \beta_2 = 15 * 180 = 2700\,kHz$. Points below this line are not reachable; available bandwidth reduction due to concurrent services just corresponds to different parallel lines in the β_1, β_2 plane. We observe that QD-MAD is always almost perfectly overlapped to D-MAD, achieving the minimum average delay (that is represented in the figure by a deep blue color on the plane); the PF, by allocating the bandwidth resources fairly to the users, like the S-SUA approach, presents degrading performance as the users'

[2] The QoE-AW scheduler is implemented by computing at each TTI the RB priorities as in [9, eq. (2)], adopting the therein defined parameters apart for the target buffer that is $TB = 40s$. This condition has been chosen to observe buffer depletion without encountering rebuffering events, so as to evaluate the scheduler performances independently from different client-originated rebuffering policies. As for the PF, [9, eq. (2)] is modified by setting the buffer level weight to 1 for all users.

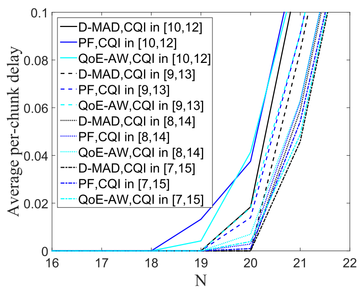


Fig. 3. Average per-chunk delay $\sum_0^{N-1} \delta_i$ vs number of users N.

requests become unbalanced; the QoE-AW scheduler, being dynamically adapted to the client buffer status, improves the performance with respect to PF, still it not reaches the optimal performance of D-MAD and QD-MAD, although using a client side additional information; the D-BWA approach definitely over-rates the bandwidth requests from user with low channel quality. Furthermore, we consider an increasing number of users and dynamic channel conditions. Specifically, we consider CQI uniformly randomly varying both in time and frequency, as $\overline{CQI} \pm 1, \pm 2, \pm 3, \pm 4$ around $\overline{CQI} = 11$. This results in an average η in $2.3 - 2.4$. PF and QoE-AW have full knowledge of CQI both in time (every TTI) and frequency, while D-MAD relies only on the average spectral efficiency. We compare the average per chunk delay $\sum_0^{N-1} \delta_i / N$ for D-MAD, PF and QoE-AW (Fig.3). For any number of users, the average delay is minimized by the MAD approach; besides, the QoE aware scheduler outperforms the PF scheduler. In fact, the PF scheduler is fair but it does not account any QoE parameter; the QoE-AW scheduler, suited to HAS streaming and progressive download, accounts for QoE by introducing heuristic weight of the client buffer occupancy; the MAD optimally allocates the bandwidth exploiting the chunk-based HAS protocol structure.

Finally, we have conducted some experiments considering 10 users, divided in 2 groups at CQI randomly varying within $4 - 6$ (LQ users) and $11 - 13$ (HQ users), streaming the video sequences HP and FN encoded at LQ average rates of $320, 247$ kbps and HQ rates $609, 480$ kbps, respectively. We implemented a D-MAD compliant Frequency Domain Packet Scheduling (FDPS) algorithm, computed by resorting to the same computational scheme as the QoE-AW scheduler, but applying the nonlinear weight targeting the occupied bandwidth level.[3] Numerical simulations have been carried out over 1190 chunks; the results, averaged over 10 runs, show that the percentage of timely delivered chunks using D-MAD, namely $PTD_{D-MAD} = 97.9\%$, improves with respect to average $PTD_{PF} = PTD_{QoE-AW} = 89.15\%$ achieved by state of the art FDPS scheduling; the same behavior is observed on the measured average delay per chunk $\sum_0^{N-1} \delta_i / N$, which sum up to $0.0063, 0.0938$, and 0.0937 for D-MAD, PF and QoE-AW, respectively.

[3]The D-MAD compliant FDPS scheduler computes the RB priorities as in Eq. (2) of [9], applying the nonlinearity $f(\cdot)$ to the actually allocated bandwidth with $offset = 1$, $priorityAtTarget = 20$, $a = 0.5$, and TB equal to the target D-MAD bandwidth.

V. Conclusions

We discussed a Dynamic Minimum Average Delay bandwidth allocation scheme for HAS streaming over LTE. The D-MAD periodically operates at a temporal scale of the video packets duration (seconds), and accounts for the users' fluctuating channel qualities to guarantee the targeted video QoE level while minimizing the streams' playout delay. The proposed architectural model requires changes only at the BS side, while leaving both server and clients unaltered. In reducing the video chunk transfer delay, the D-MAD reduces the user buffer depletion rate and the rate at which the application layer is required to dynamically adapt the quality of the video stream, thus improving the QoE in terms of video fluidity and video quality smoothness.

References

[1] H. Nam, K. H. Kim, B. H. Kim, D. Calin, and H. Schulzrinne, "Towards dynamic QoS-aware over-the-top video streaming," in *Proc. IEEE 15th Int. Symp. WoWMoM*, Jun. 2014, pp. 1–9.

[2] S. Colonnese, P. Frossard, S. Rinauro, L. Rossi, and G. Scarano, "Joint source and sending rate modeling in adaptive video streaming," *Signal Process., Image Commun.*, vol. 28, no. 5, pp. 403–416, 2013.

[3] S. Colonnese, F. Cuomo, T. Melodia, and R. Guida, "Cloud-assisted buffer management for HTTP-based mobile video streaming," in *Proc. ACM PE-WASUN*, 2013, pp. 1–8.

[4] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," in *Proc. IEEE ICCW*, Jun. 2015, pp. 1765–1770.

[5] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tut.*, vol. 17, no. 1, pp. 469–492, 1st Quart. 2015.

[6] C. Mueller, S. Lederer, and C. Timmerer, "A proxy effect analyis and fair adatpation algorithm for multiple competing dynamic adaptive streaming over HTTP clients," in *Proc. IEEE VCIP*, Nov. 2012, pp. 1–6.

[7] *LTE Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures Version 8.8.0 Release 8*, document 3GPP TS 36.213, 2009.

[8] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 678–700, 2nd Quart. 2013.

[9] J. Navarro-Ortiz, P. Ameigeiras, J. M. Lopez-Soler, J. Lorca-Hernando, Q. Perez-Tarrero, and R. Garcia-Perez, "A QoE-aware scheduler for HTTP progressive video in OFDMA systems," *IEEE Commun. Lett.*, vol. 17, no. 4, pp. 677–680, Apr. 2013.

[10] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 451–465, Mar. 2015.

[11] A. El Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 988–1001, Jun. 2015.

[12] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proc. ACM MobiCom*, 2013, pp. 389–400.

[13] I. Rubin, S. Colonnese, F. Cuomo, F. Calanca, and T. Melodia, "Mobile HTTP-based streaming using flexible LTE base station control," in *Proc. IEEE WoWMoM*, Jun. 2015, pp. 1–9.

[14] P. Seeling and M. Reisslein, "Video transport evaluation with H.264 video traces," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1142–1165, 4th Quart. 2012.

[15] *Video Trace Files and Statistics*. (2014). [Online]. Available: http://trace.eas.asu.edu/videotraces2/h265/

[16] S. Colonnese, S. Russo, F. Cuomo, T. Melodia, and I. Rubin, "Timely delivery versus bandwidth allocation for DASH-based video streaming over LTE," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 586–589, Mar. 2016.