# Quality-of-Experience driven Adaptive HTTP Media Delivery

A. El Essaili[1], D. Schroeder[1], D. Staehle[2], M. Shehada[2], W. Kellerer[2]* and E. Steinbach[1]

[1]Institute for Media Technology, Technische Universität München, Munich, Germany
[2]DOCOMO Communications Laboratories Europe GmbH, Munich, Germany

***Abstract*—This paper presents a Quality of Experience (QoE) driven approach for multi-user resource optimization in Dynamic Adaptive Streaming over HTTP (DASH) over next generation wireless networks. Our objective is to enhance the user experience in adaptive HTTP streaming by jointly considering the characteristics of the media content and the available wireless resources in the operator network. Specifically, we propose a proactive QoE-based approach for rewriting the client HTTP requests at a proxy in the mobile network. The advantage of the proposed approach is its applicability for over-the-top (OTT) streaming as it requires no adaptation of the media content. We compare our proposed scheme to both reactive QoE-optimized and to standard-DASH HTTP streaming. Our contributions are: 1) We first show that standard OTT DASH leads to unsatisfactory performance since the content agnostic resource allocation by the LTE scheduler is far from optimal, and we can achieve a clear QoE improvement when considering the content characteristics. 2) We additionally show that proactively rewriting the client requests gives control of the video content adaptation to the network operator which has better information than the client on the load and radio conditions in the cell. This results in additional gains in user perceived video quality. 3) A standard unmodified DASH client remains unaware of the proposed rewriting of the HTTP requests and can decode and play the redirected media segments.**

## I. INTRODUCTION

The enhanced capabilities of mobile devices and the improved capacities of wireless networks have led to a massive growth in mobile video consumption. Cisco's traffic forecast argues that video traffic is substantially growing and will also dominate in mobile networks in the future [1].

RTP/UDP-based streaming requires a specialized streamer and is often blocked by firewalls. On the other hand, traditional HTTP/TCP progressive download is widely deployed nowadays (e.g., YouTube). Nevertheless, it does not support intra-session rate adaptation which results in frequent stallings under bandwidth limitations. Recently, Dynamic Adaptive Streaming over HTTP also referred to as MPEG-DASH or 3GP-DASH [2], has been standardized for mobile multimedia streaming. It re-uses the HTTP/TCP over IP networking approach and provides an entire streaming framework (media representation, transport and dynamic rate adaptation) which is compatible with the standard HTTP protocol.

For mobile media delivery, the wireless link remains the main bottleneck [3]. Specifically, mobile operators face the challenge of allocating the scarce network resources among multiple clients while maximizing the user quality of experience (QoE). DASH is specially designed to adapt the video quality to mobile networks with limited and highly variable resource availability. In the wireless network this means that DASH adapts the video quality individually for every user to the resources allocated by the scheduler in the eNodeB. The eNodeB, however, is not content-aware and the scheduler assigns resources only based on channel conditions and without considering the characteristics of the transported content. QoE-driven resource allocation is an approach that optimizes the network resources by taking into account both the content characteristics and the channel conditions. QoE-based resource allocation over wireless networks has been proposed for traditional RTP/UDP streaming (e.g., [4]) but has not yet been studied for adaptive HTTP media delivery. While in-network content adaptation (e.g., transcoding) can be costly, adaptive HTTP streaming provides inherent adaptivity by encoding the same content at multiple bit-rates.

So far adaptive HTTP media delivery has been mainly studied from an end-to-end server-client perspective and the mobile network is treated as a black-box [3], [5]. Indeed, [6] concludes that the TCP throughput should be twice the video bit-rate to ensure a good streaming performance. As this over-provisioning may not be feasible for resource-constraint wireless transmission, resource management strategies that adapt to the individual user conditions and media characteristics should be considered.

In this paper, we propose a QoE-driven multi-user DASH scheme that optimizes the adaptive HTTP media delivery to multiple clients in a wireless cell. The standard DASH streaming functionality is by definition semi-decentralized, i.e., a) the client is responsible for estimating the streaming rate and makes segment decisions and b) the base station performs the overall resource allocation. To the best of our knowledge, this is the first work that considers the benefit of proactively adapting the streaming rate and the network resources to the user perceived quality in adaptive HTTP streaming. Also, different from rate adaptation schemes for single DASH streams which adjust to throughput variations, our QoE-based multi-user resource allocation approach directly considers the impact on the user quality of experience given that the streamed contents exhibit different rate-distortion characteristics.

The rest of the paper is organized as follows. In the next section, we review related work. In Section III, we present our proposed QoE-based adaptive HTTP system. Section IV outlines our system architecture. Section V describes the

---

*now with Institute for Communication Networks, Technische Universität München, Munich, Germany experimental results and Section VI concludes the paper.

## II. RELATED WORK

The DASH protocol defines the media presentation description (MPD) and segment formats for adaptive HTTP streaming but the rate control strategies are not part of the standard [2]. Different client-based rate adaptation approaches have been proposed to enhance the user perception in adaptive HTTP media delivery. [7] evaluates the end-to-end QoE in adaptive HTTP streaming over LTE. A client-driven adaptation algorithm that aims at minimizing the rebuffering events is considered. In [8], audio-visual metadata (rate-quality information) is added to the MPD as an extension to the *Subset* element of DASH. Each client computes the optimal rates individually that maximize its audio-visual quality. All these approaches, however, optimize the HTTP streaming of a single client without further considering the influence on other DASH users sharing the same network resources.

Meanwhile, the multi-user DASH problem has been less explored so far. Among the few works on multi-user adaptive HTTP streaming, [9] considers a fair scheduler for adapting the HTTP/TCP video transmission at the last-hop wireless link. In [10], network management for adaptive HTTP video delivery across multiple clients is considered. The target bit-rate is determined by the network based on available throughput estimates of all users. The authors in [11] conclude that a simple rate shaping policy in a residential gateway can improve the adaptive HTTP experience among two competing clients. All of these works, however, do not exploit the media content information. Most recently, [12] proposed a rate adaptation algorithm, WiDASH, for optimizing the adaptive HTTP streaming across multiple wireless clients. The optimization is carried out at a proxy located at the edge of the wireless network which splits a TCP connection between the DASH server and the wireless user. WiDASH first prioritizes the video streams and then transcodes the incoming DASH streams to a target bit-rate. The proposed approach, however, does not consider the individual content characteristics of the different clients and aims at stabilizing the user throughput. Also, different from our work, the authors of [12] propose to transcode the DASH stream, similar to typical RTP/UDP based optimizations (e.g., [4]), which is costly and may react too late.

## III. QOE-BASED ADAPTIVE HTTP SYSTEM

In contrast to UDP streaming which is push-based, DASH is a pull-based client driven streaming protocol. Multiple bit-rate encodings of the same content are generated and segmented at the DASH server. A DASH client uses the MPD to learn about the representations and dynamically requests the media segments that match its available transmission capacity.

In this work, we study how adaptive HTTP media delivery can benefit from QoE-based resource optimization in the mobile network. Specifically, we consider the QoE optimization of multiple DASH users in a wireless cell. A schematic depiction of our proposed QoE-based adaptive HTTP system is given in Figure 1. Multiple mobile clients are simultaneously downstreaming different DASH content over an LTE network. At the DASH server, the utility information of each content is first extracted and added to the MPD. In the mobile network, e.g., at eNodeB, a QoE optimizer parses the MPD to extract relevant QoE parameters from the streamed content. It collects utility and channel information about the different clients and determines their optimal transmission rates as described in [4].

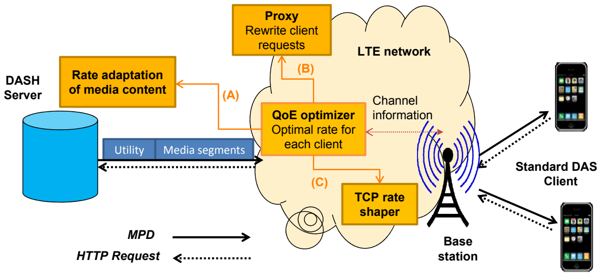


Fig. 1. System image of QoE-driven adaptive HTTP mobile media delivery.

We study different approaches for applying the result of the QoE optimization from [4] to DASH. Specifically, we require that our adaptation schemes should work with any standard DASH client. Figure 1 depicts three methods for optimizing the DASH delivery at the server, a proxy and inside the mobile network, respectively:

(A) When the DASH server and the mobile clients are contained in the operator network, the mobile operator can fully optimize the media delivery. This corresponds to the case of managed content where the server can adapt the streaming rate of each client to the target rate returned by the QoE optimizer. Without loss of generality, we assume that the server can encode the media content at the optimal rate. This provides an upper limit on the achievable QoE.

(B) Internet video streaming is dominated by over-the-top (OTT) content where the DASH server lies outside the operators' network. Considering the multiple bit-rate encodings within the MPD, we propose a proxy-based approach for redirecting the client HTTP requests to the closest lower representation from the MPD which matches the QoE optimization result. The client requests are rewritten on-the-fly at the proxy and forwarded to the DASH server. At the server, no further adaptation of the media content is required. Also, each client will decode and play an optimized representation for its requested segment. In fact, both the DASH server and the DASH clients are unaware of the proposed proxy operation.

(C) An alternative optimization strategy is to optimize the bit-rate of each client without interfering with the client decisions. In other words, the TCP throughput for each client is shaped according to the QoE optimizer feedback. The actual streaming rate is determined by the client which reacts to the throughput changes.

In all three cases, the client is an unmodified standard DASH client. The approaches only differ in how the QoE-based resource allocation result is exploited for dynamic rate adaptation for overall QoE optimization.

## IV. SYSTEM ARCHITECTURE

### A. Application model

We express the user satisfaction or QoE for real-time video streaming on a Mean Opinion Score (MOS) scale [13]. The utility function for video streaming is defined in [4] as a function of the application data rate $R$ by:

$$U = MOS(R) \tag{1}$$

We assume a simple linear mapping between the MOS and the peak signal-to-noise ratio (PSNR) [14]. Please note that more complex mappings could be used (e.g., [15]). MOS can take on any value between 1.0 (30 dB) and 4.5 (42 dB), which represent the worst and best QoE, respectively.

We consider two options for providing meta information about the streamed DASH content. In the first case, we use the simple parametric model from [16] which requires three pairs of rate and distortion to represent each video sequence. Please note that this represents a generic MOS-Rate function which can be delivered for instance at the beginning of the streaming session. The model from [16] can be used to generate an arbitrary set of MOS-Rate operating points.

In the second case, the utility information is provided in the form of MOS-Rate pairs for each representation in the MPD. Although the DASH protocol does not explicitly define how to transmit utility information, it provides various options for this [17]. The utility can, for example, be signaled in the initialization segments of the MPD. In DASH, one initialization segment is allowed per representation. Alternatively, the utility of different representations within a program period can be added to the *Subset* element of DASH [8].

Figure 2 shows the utility curves for three different video sequences using the parametric model representation and the MOS of the actual DASH representations.

### B. Radio model

We consider a long-term radio model with optimization periods in the order of seconds. Our objective is to determine the resource share (i.e., physical resource blocks (PRBs) in LTE) of each client in each optimization cycle. This allows us to integrate our QoE-based optimization on top of the state-of-the art schedulers for LTE without the need to modify the scheduling mechanisms already deployed.

We use the radio link layer model originally proposed in [18]. It defines the data rate $R_k$ for user $k$ as a function of its resource share $\alpha_k$ and its maximum achievable rate $R_{max,k}$ if all the PRBs are allocated exclusively to user $k$, cf. (2).

$$R_k = f_k(\alpha_k) = \alpha_k R_{max,k} \quad 0 \leq \alpha_k \leq 1, \forall k \tag{2}$$

In each optimization round, a new $R_{max,k}$ is determined for each client based on its average channel statistics in the last 2 seconds. We use the link layer model from the 3GPP LTE recommendations [19] to determine the achievable throughput per PRB for a given Signal-to-Noise ratio (SNR).

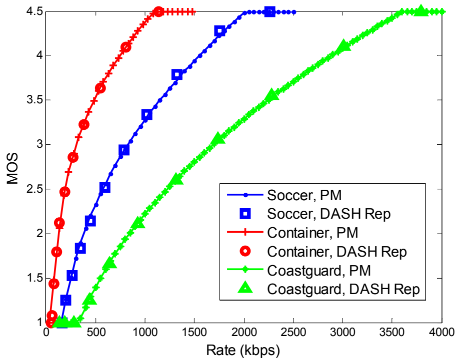


Fig. 2. Utility curves considered in this work.

### C. QoE-based resource optimization

The objective of the QoE-based resource optimization function is to maximize the overall user satisfaction. In this work, we use the objective function from [4] which maximizes the sum of utilities of all users. The optimization problem for $K$ clients is formulated as:

$$\underset{(\alpha_1,\ldots,\alpha_K)}{\arg\max} \sum_{k=1}^{K} U_k(\alpha_k) \tag{3}$$

$$subject\ to \quad \sum_{k=1}^{K} \alpha_k = 1, \quad R_k \geq R_{min,k} \tag{4}$$

where (3) determines the resource share of each user that maximizes the sum of utilities. (4) constrains on the available resources and defines a minimum rate that should be allocated to each user (e.g., lowest representation).

Each $\alpha_k$ value corresponds to the fraction of total PRBs assigned to user $k$ in each optimization round. A gradient-based greedy algorithm, similar to the work in [20], is used to determine the values of $\alpha_k$. Depending on the utility information type we differentiate between:

1) Continuous QoE optimization (**QoE**): In the case parametric meta data about the streamed content is available, the algorithm will search for the set of $\alpha_k$ values that maximizes (3). For arbitrary small $\alpha_k \to 0$, the algorithm can choose from a continuous set of rates for each user. The optimal bit-rate is returned by the QoE optimizer.

2) Discrete QoE optimization (**QoE-d**): When the actual MOS-Rate values of the DASH representations are available, the algorithm will choose from a discrete set of operating points. The set of $\alpha_k$ values corresponds to the encoding rates which are defined in the MPD. The actual target representation rate from the MPD is returned by the QoE optimizer.

## V. SIMULATION RESULTS

### A. Experimental setup

We consider a single LTE cell with 8 clients requesting different DASH videos. The videos are encoded at different quality levels with the H.264/AVC video codec. Specifically, a total of 11 different quantization parameters ranging from 20 to 40 are used at the encoder to generate 11 representations of the same video. For our experimental evaluations, we choose the Microsoft Smooth Streaming client [21] and the DASH-enabled VLC client [22] without modifying the clients. Furthermore, we use a standard HTTP server and emulate the wireless network. In other words, a resource shaper is placed between the DASH server and the clients that limits the data rates per client to the output of the QoE optimizer. We consider 50 simulation runs in order to study the impact of different mobility patterns. All users start streaming at the same time. Table I summarizes the simulation parameters.

In our simulations, we compare the following schemes:

- **QoE-Server**: The server encodes the video stream at the optimal rate returned by the QoE optimizer. This corresponds to approach (A) in Section III.
- **QoE-Proxy**: The optimal rate is first signaled by the QoE optimizer to the proxy. The proxy then chooses the closest lower available streaming rate from the MPD, rewrites the client request and forwards it to the DASH server. This represents approach (B) where we assume that the DASH server is outside the control of the network operator.
- **QoE-d-Proxy**: Similar to the QoE-Proxy scheme. However, the optimizer uses the discrete utility representation and returns the target rate (MPD compatible) to the proxy which again rewrites the client request. This also represents approach (B).
- **QoE-Reactive**: Each client gets a TCP throughput equal to the optimal rate determined by the QoE optimizer. The streaming rate, however, is only determined by the media streaming client. This corresponds to approach (C).
- **Non-Opt**: Standard OTT DASH streaming where the transmission rate is determined by the content-agnostic LTE scheduler, and the streaming rate is dynamically decided by the standard DASH client. This represents our reference scheme for comparison.

### B. Performance evaluation

We first consider the Microsoft Smooth Streaming client for all approaches in our simulations. We measure the MOS from 20 to 60 seconds to exclude the client dependent start-up behaviour. We evaluate the distribution of average mean MOS for the different schemes in order to highlight the differences in the overall performance (Figure 3). The QoE-Server approach represents the optimal performance when all users are streaming at their optimal rates. The MOS for the QoE-Proxy approach will drop compared to the QoE-Server scheme as only a discrete set of representations is available. Meanwhile, the MOS degradation is less noticeable in the QoE-d-Proxy approach and is close to the optimal value (QoE-Server) as the QoE optimizer considers the actual DASH representations in the optimization problem. The QoE-Reactive scheme improves the perceived video quality compared to the non-optimized DASH scheme (Non-Opt). The MOS for the QoE-Reactive scheme, however, drops compared to the proactive approaches as the client reacts late to throughput changes and does not always converge to the best representation level. Please note that for the QoE-Reactive and Non-Opt schemes the streaming rate is only determined by the client. The QoE-Reactive, QoE-Proxy, QoE-d-Proxy and QoE-Server schemes improve the average user satisfaction by 0.2, 0.36, 0.48 and 0.57 on the MOS scale compared to the Non-Opt scheme, respectively.

Furthermore, we evaluate the mean MOS for the individual videos in order to explain and further highlight the benefits of the QoE optimization that considers the content characteristics (Figure 4). The Non-Opt scheme provides a very good performance for the less demanding users but fails for more demanding ones like *bus*, *coastguard* and *harbour*. Meanwhile, the QoE-based schemes allocate the resources among the users such that the overall user satisfaction is maximized. This results in substantial gains in perceived video quality for the demanding users while maintaining the MOS for the less demanding videos.

TABLE I
SIMULATION PARAMETERS

| **LTE parameters** | |
|---|---|
| System Bandwidth | 5 MHz |
| Number of PRBs | 25 |
| PRB size | 12 subcarriers |
| Subcarrier spacing | 15 KHz |
| Bandwidth per PRB | 180 KHz |
| SNR averaging cycle | 2 sec |
| Link layer model | [19] |
| Channel model | Urban macrocell |
| Shadowing | disabled |
| User speed | 30 km/h |
| Cell radius | 1000 m |
| Default scheduler | Round Robin (RR) |
| **Simulation parameters** | |
| Video codec | H.264 AVC, CIF, 30 fps |
| Application type | Adaptive HTTP streaming |
| Segment size | 2 sec |
| Number of clients | 8 |
| Simulation runs | 50 |
| Simulation time | 60 sec |

### C. Temporal quality analysis

We assess the temporal quality of our QoE-Proxy approach and compare it to the QoE-Reactive and Non-Opt schemes. We perform two separate experiments where all clients are using the Microsoft Smooth Streaming (respectively the DASH-enabled VLC [22]) as standard clients.

Figure 5 shows the signal-to-noise (SNR) ratio of three users who are undergoing dynamic channel variations. Figure 6 (a)-(f) show the requested representations for the different schemes. The available transmission rate for each user is shown for the non-optimized (Throughput RR) and the optimized (Throughput QoE) cases. The Non-Opt and QoE-Reactive schemes indicate the corresponding representations as requested by the media client in both cases, respectively. For our QoE-Proxy approach, the client is unaware of the rewriting of the HTTP requests and can decode the redirected segments using both clients.

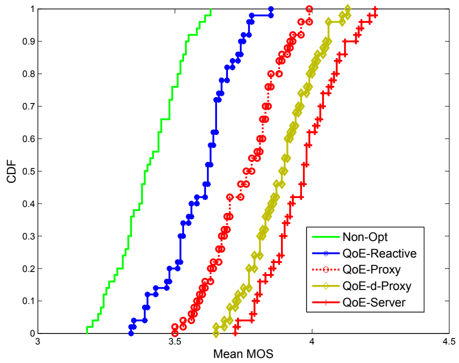


Fig. 3. CDF of the mean MOS for 8 users.

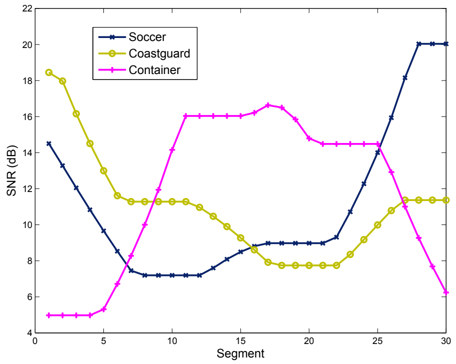


Fig. 5. Channel variations as a function of time (segment = 2 seconds).

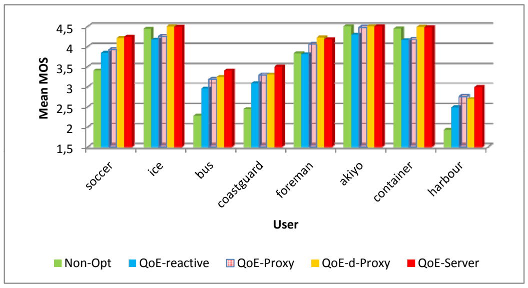


Fig. 4. Individual performance averaged over 50 simulation runs.

In Figure 6 (a) and (b), both clients start streaming at a low bit-rate whereas our QoE-Proxy approach can start streaming at a much higher rate. Indeed, the user is experiencing favorite channel conditions and rewriting the requests cancels the slow-start behaviour of the standard clients. The Microsoft Smooth Streaming client smoothly increases the representation rate while the DASH-enabled VLC client immediately switches to a higher rate after the start-up phase (5 segments). As the channel conditions of the *Soccer* user deteriorates, our proxy approach switches to a lower representation while the other approaches can continue playout at a higher rate for some time, as they have already buffered these representations during the start-up phase. As the channel conditions improve gain (after 15 segments), our QoE-Proxy approach can quickly switch to a higher rate where as the other approaches are late to react.

Figure 6 (c) and (d) consider the demanding *Coastguard* user who undergoes some bad channel conditions after a good start-up phase. In this case, the DASH-enabled VLC client will run into a sequence of ”rebuffering” events (e.g., after 14, 19, and 25 segments for the QoE-Reactive approach). The Microsoft Smooth Streaming client can continue to play the video at a higher rate before it switches to a lower representation. This is again explained by the buffered segments when the user was in an excellent channel condition.

Figure 6 (e) and (f) consider the *Container* user whose channel quality is bad at the beginning of the streaming session and starts improving afterwards. The Microsoft Smooth Streaming client and the DASH-enabled VLC client slowly adapt to the channel improvements and would only converge to the rate of the QoE-Proxy scheme after 27 and 24 segments, respectively.

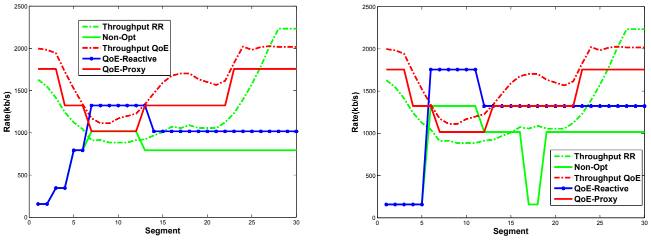
These results show that the DASH-enabled VLC client provides the worst user experience as it fails to maintain quality during deep channel fades and runs into ”rebuffering” mode. The Microsoft Smooth Streaming client keeps a large buffer which allows it to maintain a good quality when the channel degrades but often reacts very late and does not converge to the best representation. Meanwhile, our proposed QoE-Proxy approach, can fully utilize the network resources and provide the best possible representation for each client under different channel conditions.

### D. *Perceptual video quality assessment*

We additionally compare the perceived video quality of our proposed QoE-Proxy approach with the QoE-Reactive scheme using the Microsoft Smooth Streaming client. Figure 7 shows snapshots captured at different time instants during the start-up phase and after the quality has stabilized. The figures show remarkable gains in perceived video quality that can be achieved by proactively adapting the video transmission rate by our QoE-Proxy approach.

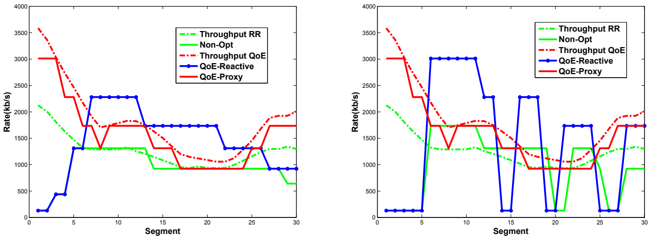
## VI. CONCLUSION

In this paper, we present a QoE-based approach for jointly optimizing the adaptive HTTP media delivery across multiple clients in a wireless cell. By considering the media characteristics of each content and the channel conditions of the mobile users the optimal streaming rate is determined. We first show that QoE-based multi-user resource allocation improves the user experience compared to non-optimized OTT adaptive HTTP streaming, when using the same client adaptation algorithm. In addition, we propose a QoE-based proxy approach for redirecting the HTTP client requests to the optimal streaming rate, and which can be still decoded by a standard DASH client. While client-based adaptation approaches react late to channel variations and fail to stream at the optimal rate, we demonstrate that by proactively adjusting the streaming rate significant gains in user perceived video quality can be achieved. For our performance evaluations, we worked with two standard adaptive HTTP clients, the Microsoft Smooth Streaming and the DASH-enabled VLC [22].



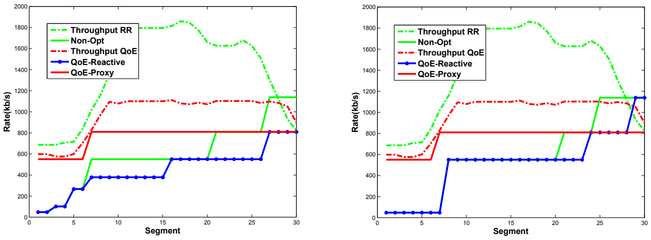
(a) Soccer: Microsoft Smooth Streaming

(b) Soccer: DASH-enabled VLC [22]



(c) Coastguard: Microsoft Smooth Streaming

(d) Coastguard: DASH-enabled VLC [22]



(e) Container: Microsoft Smooth Streaming

(f) Container: DASH-enabled VLC [22]

Fig. 6. Temporal assessment of the different optimization approaches. Throughput QoE and throughput RR are the available rates for each user as determined by the QoE optimizer and RR scheduler, respectively.

(a) QoE-Proxy 2 seconds

(b) QoE-Reactive 2 seconds

(c) QoE-Proxy 15 seconds

(d) QoE-Reactive 15 seconds

(e) QoE-Proxy 29 seconds

(f) QoE-Reactive 29 seconds

Fig. 7. Comparison of the visual quality for our QoE-Proxy approach and the QoE-Reactive scheme.

## REFERENCES

[1] Cisco white paper, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016,” Tech. Rep.

[2] T. Stockhammer, “Dynamic adaptive streaming over HTTP - standards and design principles,” *Proc. MMSys 2011, California, USA*, Feb. 2011.

[3] A. Begen, T. Akgul, and M. Baugher, “Watching video over the web: Part 2: Applications, standardization, and open issues,” *IEEE Internet Computing*, vol. 15, no. 3, pp. 59–63, May-June 2011.

[4] S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer, “QoE-driven cross-layer optimization for high speed downlink packet access,” *Journal of Communications*, vol. 4, no. 9, pp. 669–680, Oct. 2009.

[5] K. Ma, R. Bartos, S. Bhatia, and R. Nair, “Mobile video delivery with HTTP,” *IEEE Comm. Magazine*, vol. 49, no. 4, pp. 166–175, Apr. 2011.

[6] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “Multimedia streaming via TCP: An analytic performance study,” *ACM Trans. Multimedia Comput. Commun. Appl.*, May 2008.

[7] O. Oyman and S. Singh, “Quality of experience for HTTP adaptive streaming services,” *IEEE Comm. Magazine*, vol. 50, no. 4, pp. 20–27, Apr. 2012.

[8] T. Thang, Q. Ho, J. Kang, and A. Pham, “Adaptive streaming of audiovisual content using MPEG DASH,” *IEEE Trans. on Consumer Electronics*, vol. 58, no. 1, pp. 78–85, Feb. 2012.

[9] W. Pu, Z. Zou, and C. Chen, “New TCP video streaming proxy design for last-hop wireless networks,” *Proc. IEEE ICIP 2011, Brussels, Belgium*, Sept. 2011.

[10] K. Ma and R. Bartos, “HTTP live streaming bandwidth management using intelligent segment selection,” *Proc. IEEE Globecom 2011, Texas, USA*, Dec. 2011.

[11] R. Houdaille and S. Gouache, “Shaping HTTP adaptive streams for a better user experience,” *Proc. MMSys 2012, North Carolina, USA*, Feb. 2012.

[12] W. Pu, Z. Zou, and C. Chen, “Video adaptation proxy for wireless dynamic adaptive streaming over HTTP,” *Proc. Packet Video Workshop 2012, Munich, Germany*, May 2012.

[13] ITU, *Methods for subjective determination of transmission quality (ITU-T Rec. P.800)*, International Telecommunication Union, Aug. 1996.

[14] VQEG, “Final report from the video quality experts group on the validation of objective models of video quality assessment.”

[15] S. Wolf and M. Pinson, “Video quality measurement techniques,” NTIA Technical Report TR-02-392, Tech. Rep., June 2002.

[16] L. Choi, M. Ivrlac, E. Steinbach, and J. Nossek, “Sequence-level methods for distortion-rate behavior of compressed video,” *Proc. IEEE ICIP’05, Genova, Italy*, Sept. 2005.

[17] ISO/IEC IS 23009-1, “Information technology- Dynamic adaptive streaming over HTTP (DASH)- Part I: Media presentation description and segment formats,” Tech. Rep., Apr. 2012.

[18] A. Saul et al., “Cross-layer optimization using model-based parameter exchange,” *Proc. IEEE ICC’07, Glasgow, Scotland*, June 2007.

[19] 3GPP TR36.942, “Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios,” 3rd Generation Partnership Project (3GPP), Tech. Rep., Jan 2011.

[20] D. Jurca and P. Frossard, “Media flow rate allocation in multipath networks,” *IEEE Trans. on Multimedia*, vol. 9, no. 6, pp. 1227 –1240, Oct. 2007.

[21] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” *Proc. MMSys 2011, California, USA*, Feb. 2011.

[22] C. Mueller and C. Timmerer, “A VLC media player plugin enabling dynamic adaptive streaming over HTTP,” *Proc. ACM Multimedia 2011, Arizona, USA*, Nov. 2011.