Exploiting Mobility Prediction for Mobility & Popularity Caching and DASH Adaptation

Vasilios A. Siris, Xenofon Vasilakos, and Dimitrios Dimopoulos
Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences & Technology
Athens University of Economics and Business, Greece
{vsiris, xvas, dimdimopoulos}@aueb.gr

***Abstract*—We present our recent work investigating how mobility prediction can be exploited for improving the performance of mobile users in two directions: proactive caching requested content close to the network attachment points where a mobile has a high probability to connect to and DASH (Dynamic Adaptive Streaming over HTTP) video quality adaptation. For proactive caching we discuss a new model to proactively cache content based on both mobility prediction and content popularity. An important feature of the model is that it dynamically adapts caching decisions to the relative importance of the two factors. For DASH adaptation we discuss a procedure that exploits mobility and throughput prediction to select the quality levels of video segments requested by a DASH player in order to achieve improved QoE, in terms of both high video quality and few video quality switches.**

I. Introduction

Mobile traffic in 2015 grew 74%, continuing a more than 4,000-fold growth over the past 10 years, and is expected to increase nearly 8-fold from 2015 until 2020[1]. Mobile video was 55% of the total traffic by the end of 2015 and is expected to increase 11-fold from 2015 to 2020, becoming 75% of the total mobile traffic in 2020. Efficient support for video streaming in future mobile environments, in terms of both network resource utilization and energy consumption, will require the integration of heterogeneous wireless technologies with complementary characteristics; this includes cellular networks with macro, femto, and pico cells, Wi-Fi hotspots that support high throughput and energy efficient data transfer, in addition to technologies such as device-to-device communication. Additionally, there is an increasing trend towards HTTP-based adaptive streaming, where a video is partitioned into a series of segments which are encoded in multiple quality levels. This approach is followed by proprietary solutions, such as Microsoft's Smooth Streaming and Apple's HTTP Live Streaming (HLS), and by the Dynamic Adaptive Streaming over HTTP (DASH) standard. A DASH client requests each segment individually, allowing it to change the quality of each segment depending on the network conditions.

Prior work has provided evidence that mobility and throughput prediction is possible for both cellular [1] and Wi-Fi [2]. Such prediction can improve mobile network functions and lead to higher network efficiency. The contribution of this paper is to introduce two models for exploiting mobility prediction. The first model considers proactively caching content by jointly exploiting mobility prediction and content popularity. Mobility prediction in this case involves knowing the probability of a mobile connecting to future network attachment points. The second model exploits mobility and throughput prediction to improve the QoE (Quality of Experience) for DASH video streaming. Mobility and throughput prediction in this case involves knowledge of the average throughput that a mobile is expected to have available in different future time periods. The above two models are discussed separately in the current paper. However, the two models can be combined, e.g. in the case of scalable video coding where high quality bitstreams contain a baseline bitstream, which can be proactively cached since all video qualities require it.

The continuous reduction of storage costs has made it possible to increase the capacity of caches in small cell networks, such as femto/pico cells and Wi-Fi networks. There are two advantages that can be achieved by pre-fetching content in caches located in small cells close to the end users. First, content, such as web content and social networking notifications, will be immediately available to a mobile once it connects to the new attachment point, thus reducing the latency for receiving the content. Second, if the mobile's attachment point is in a small cell whose bottleneck is its backhaul, then pre-fetching content in a cache inside the small cell can help exploit the higher wireless throughput, thus overcoming the backhaul's capacity constraints, e.g. in the case of video transfer. The advantage of proactively caching content using mobility prediction, rather than content popularity, is that the heavy-tailed distribution of content popularity makes it inefficient to increase cache storage to accommodate less popular content [3]. On the other hand, mobility-based proactive caching can be effective in caching such less popular content, since the mobile transition probabilities determine the caching decisions. Regarding DASH adaptation, mobility and throughput prediction can allow scheduling the quality of video segments over a larger time window, which can potentially improve the overall video quality and reduce the number of quality switches. For example, knowledge of the

[1]Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020, Feb. 3, 2016

978-1-5090-2185-7/16/$31.00

future throughput can allow a player to download video segments in advance when the available throughput is high, thus compensating for periods of low throughput; indeed, the video segments downloaded in advance can be of higher quality than the quality that would be possible if they were downloaded during the low throughput periods.

An important focus of our work is to investigate how incomplete or inaccurate knowledge influences the gains of mobility prediction. For proactive caching, mobility prediction can involve different probabilities of a mobile connecting to future network attachment points. For DASH adaptation, the predicted throughput can vary widely or have inaccuracies.

In our previous work we investigated proactive caching based solely on mobility prediction [4], [5]. The proactive caching model discussed in this paper considers mobility prediction together with content popularity. The work in [6], [7] investigated mobility prediction for improving mobile video streaming, utilizing proactive caching, multi-source, and device-to-device video transfer. DASH adaptation was not considered in [6], [7]. Finally, the work in [8], [9] considered mobile data offloading for delay tolerant traffic, which requires transferring a file within a time threshold, and delay sensitive traffic, which requires minimizing the file transfer time.

The rest of the paper is structured as follows: In Section II we present related work. In Section III we discuss our model for proactive caching that jointly considers mobility prediction and content popularity. In Section III we discuss our procedure for exploiting mobility and throughput prediction to improve the QoE for DASH video streaming. Finally, in Section V we conclude the paper.

II. Related work

The feasibility of using prediction for prefetching is investigated in [10], which however does not propose or evaluate specific prefetching algorithms. Prefetching for improving video file delivery in femtocell networks is investigated in [11], and to reduce the peak load of mobile networks by offloading traffic to Wi-Fi hotspots in [12]. Both these works consider content popularity to proactively cache content close to mobile users, before the content is requested. The proactive model presented in this paper differs from the above by jointly considering both mobility prediction and content popularity to proactively cache content. The work in [13] considered content popularity together with mobility to prefetch content at buffers located at the edge of a network. However, the proposed solution separates buffers for caching content based on popularity and for prefetching content based on mobility. On the other hand, the proactive caching model discussed in this paper jointly considers content popularity with mobility prediction, hence can dynamically adjust the buffer usage for the two types of caching, rather than pre-allocate buffer space to the two types as done in [13].

Bandwidth prediction for improving video streaming is investigated in [14], [15], [16]. The application of Markov Decision Processes (MDPs) for optimal quality selection is investigated in [17], [18], while quality adaptation based on the client buffer is investigated in [19]. The work in [20] investigated fairness in the case of multiple clients, with support from in-network mechanisms. The work in [21] also provides evidence that short-term throughput prediction can improve the performance of video streaming. From the above works, only [18], [21] consider exploiting throughput prediction. These schemes focus on short timescale adaptation strategies, whereas the work in this paper focuses on long timescale adaptation, which is appropriate when a mobile encounters cellular network segments and Wi-Fi hotspots with varying average throughput; the approach presented in this paper can be combined with short timescale adaptation strategies to adapt to fast throughput changes due to fast radio channel quality fluctuations. Our approach seeks to select a quality switching strategy across multiple connectivity segments, each providing a different average throughput. Such long timescale adaptation can yield significant gains when client-side buffering is readily available, which is the case with current smartphones and tablets, since such buffering can be used to download more video segments when the throughput is high. On the other hand, the approach in [18] used an MDP to determine an optimal video quality for each mobility segment independently. The work in [22] also considered using throughput prediction to select the video quality that maximizes the average bit rate, while trying to reduce the number of quality switches. We compare the adaptation procedure proposed in this paper with the procedure in [22]. The work in [23] investigated a DASH adaptation procedure that seeks to improve the QoE by using buffering to introduce intermediate quality levels, hence avoid switching between quality levels that are far apart. Finally, the work in [24], using crowdsourcing experiments, proposes a QoE model for quality adaptation that considers the video quality and quality switches.

III. Jointly utilizing mobility prediction and content popularity for proactive caching

We start by discussing our initial model for exploiting mobility prediction to proactively cache content requested by a mobile at caches close to the attachment point where the mobile will connect to with some probability, which we will refer to as mobile transition probability. The mobile transition probability can be estimated based on historical data [10]. The model presented below extends the one presented in [4], by considering the case where more than one mobiles request the same object. The estimated gain from proactively caching a requested object s is given by

$$Q_s^l(C_{\text{miss}} - C_{\text{hit}}), \tag{1}$$

where Q_s^l is the aggregate transition probability to the attachment point close to cache l of all mobiles requesting object s, C_{miss} is the cost for obtaining the object from its original remote server (cache miss), and C_{hit} is the cost for obtaining the object from the local cache (cache hit). Note that the model is not restricted to a specific definition of cost. If the cost refers to delay, then it can be a function of the distance to the remote server or cache, e.g. in number of hops. Alternatively, the cost

can be related to the network or monetary cost for transferring the requested object, e.g. when transferring data over a cellular network or when data transit or CDN costs are involved.

Based on the estimated gain (1), the rule for deciding whether to cache object s in cache l is

$$\begin{aligned} &\text{if} \quad Q_s^l(C_{\text{miss}} - C_{\text{hit}}) \geq p_l \qquad \text{proactively cache } s \\ &\qquad\quad \text{else} \qquad\qquad\quad \text{don't proactively cache } s \end{aligned} \tag{2}$$

where p_l is a congestion price for cache l, which increases (decreases) when the aggregate demand is above (below) the cache size. Alternatively, the price can be adjusted based on the cache utilization: the price increases if the utilization is above some target and decreases when the utilization is below the target. The target utilization can be set to a value close to 100%. The introduction of a congestion price in the decision rule (2) enables the efficient utilization of caches. Specifically, when the cache is underutilized, i.e. cache space is available, the congestion price decreases, thus allowing more objects to be proactively fetched in the cache based on (2). On the other hand, when the proactive cache demand is larger than the cache size, then the price increases which in turn, due to (2), reduces the number of objects that are proactively cached. It is important to note that using (2) to decide when an object should be cached has complexity $O(1)$ and is simpler than maintaining the estimated gains of all objects and, when the cache is full, evicting the objects with the smallest gain to make room for new objects with higher gain.

An object satisfying (2) and proactively cached is removed from the cache when the last mobile that has requested the specific object completes its handoff. Moreover, proactive caching requests are active throughout the duration of a mobile's handoff period; hence, if a request is at some point denied, the same request can later be accepted if (2) is satisfied, e.g. if more mobiles request the same object or the price has decreased due to cache space becoming available.

The above model has assumed that all objects have the same size. If the costs $C_{\text{miss}}, C_{\text{hit}}$ are independent of the object size, then the model can be extended to the case of variable size objects, by dividing the gain in (2) with the object size [4]. If the cost is a function of the object size and objects can be partially cached, then the model needs to be modified to consider the gains from partially cached objects [5].

The decision rule (2) does not explicitly take into account content popularity. However, popularity is indirectly considered, since more popular objects s are likely to be requested by more mobiles, which yields a higher aggregate transition probability Q_s^l to cache l of mobiles requesting object s.

Next we extend the model presented above to jointly consider mobility and popularity-based caching. The estimated gain considering both the mobile transition probability and the content popularity can be expressed as

$$\left(Q_s^l + w^l \cdot f_s^l\right)\left(C_{\text{miss}} - C_{\text{hit}}\right), \tag{3}$$

where f_s^l is the popularity of object s at cache l and w^l is a weight factor that depicts the gain achieved with popularity-based caching relative to mobility-based proactive caching. Specifically, w^l is the average number of requests for any object made to cache l from mobiles connected to the attachment point served by cache l, in the interval from the time an object is proactively cached until the time that the last mobile requesting the particular object has completed its handoff. Both w^l and f_s^l can be estimated online using measurements. Hence, w^l can capture the (possibly time-varying) demand for mobility-based proactive caching relative to popularity-based caching at l. The popularity f_s^l of object s at cache l can be estimated based on the inter-arrival time of consecutive requests for object s and the inter-arrival time of consecutive requests for any object, thus capturing spatial and time locality.

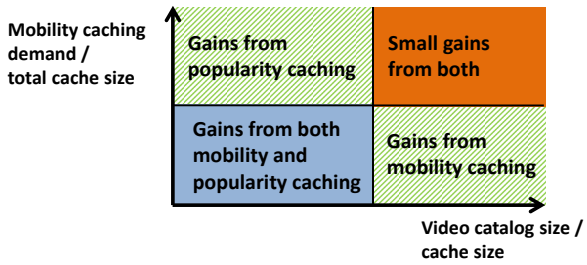


Fig. 1. Gains from mobility-based proactive caching and popularity-based caching.

Unlike the initial model, which considers only mobility-based caching, caches now contain objects that are proactively cached based on mobility and objects that are cached based on content popularity. Note that the two types of cached objects are not differentiated. Moreover, since caches are always full, there is a need for a cache replacement (eviction) policy to determine, when a new request to proactively cache content is received, which objects should be evicted from the cache in order to free space for the new content. We consider the following eviction policy: all cached objects are sorted in increasing gain based on (3), which jointly considers mobility and popularity; this can be implemented with a priority queue, which has a complexity of insertions $O(\log n)$. Cached objects with the smallest gain are removed first, to make space for objects with a higher gain. Indeed, a new proactive caching request is accepted if its expected gain is higher than the expected gain of the cached objects it would need to replace. The last condition forms the decision rule of when to accept proactive caching requests for joint mobility-based and popularity-based caching, in the place of (2).

An important feature of the proposed model based on (3) and the cache replacement policy discussed above is that the cache storage is not a priori partitioned, dedicating a part of the storage for caching objects requested based on mobility prediction and a part for caching objects based on content popularity. Rather, through the online estimation of the weight w^l in (3), the percentage of the available storage used for the two types of caching is dynamically adjusted, based on the relative gains from mobility prediction and content popularity. Also, as in the initial model, proactive caching requests are active throughout the duration of a mobile's handoff period.

The anticipated gains from mobility-based proactive caching

and popularity-based caching are shown in Figure 1. The vertical axis of Figure 1 depends on the number of mobiles, the accuracy of mobility prediction, and the cache size: a small number of mobiles or a high accuracy of mobility prediction, alternatively a large total cache size, yields a small value on the vertical axis, hence exploiting mobility prediction can lead to high gains. On the other hand, a small video catalog size or a large cache size leads to small values on the horizontal axis, hence exploiting popularity-based caching can have high gains. Indeed, user context information, such as user interests or social relations, can be utilized to reduce the video catalog size, hence increase the gains from popularity-based caching.

A. Evaluation

Next we present simulation results showing the gains of the model described above for jointly considering mobility and popularity-based caching, which we will refer to as EMPC (Efficient Mobility and Popularity-based Caching), compared to three other schemes: the initial model which only considers mobility-based proactive caching, which we will refer to as EMC (Efficient Mobility-based proactive Caching), pure popularity-based caching (MaxPop) which populates caches with the most popular items, and a naive scheme where objects are cached in all neighboring caches that have available space. The naive scheme implements a blind form of proactive caching that doesn't take into account the probability of a mobile connecting to different network attachment points, but does consider the current cell where a mobile is located and knowledge of its neighboring cells.

The simulated network topology contained 25 small cells randomly distributed over a 700×700 m^2 area, with 1.000 mobiles each requesting one (25) video file in the case of low (high) demand, when entering the area and when completing a handoff. The range of each small cell had average 71 m and standard deviation 5 m. Mobiles follow a skewed mobility model according to which 80% of the mobiles move towards the same direction, while the other 20% of the mobiles move to a different random direction (uniformly distributed), with speed 5 Km/hour and standard deviation 1.25. Later we also consider the case where the mobility has a lower skewness (60%) and when mobiles move in uniform directions. The video file size, popularity, and temporal locality was based on synthetic video requests produced with the GlobeTraff workload generator [25]. The average video size was 80 MB, while the local cache storage size was 8 GB. The cost C_{miss} depended on the hop distance and the delay for transferring data over the access network connecting the small cell to the core network. Specifically, $C_{\text{miss}} = n + 5.8$, where the number of hops n was normally distributed with mean 4.2 and standard deviation 1.05, based on [26] which shows that the inter-AS path length has remained practically constant and equal to 4.2 for over a period of 12 years, while approximately 95% of the AS-hop distances fall in the range [2.1, 6.3]. The factor 5.8 reflects the higher delay for transferring data over the provider's access network. The above give an the average cost for a cache miss $\overline{C_{\text{miss}}} = 10$, while the cost of a cache hit was $C_{\text{hit}} = 1$. Based on this, an upper bound for the reduction of the total cost, compared to the total cost when caching is not used, is 90%.

TABLE I
GAINS OF DIFFERENT CACHING SCHEMES.

Mobility caching demand	Video catalog size	EMPC	EMC	MaxPop	Naive
Low	Small	89.1%	81.4%	49.0%	62.6%
Low	Large	79.0%	49.2%	8.5%	30.1%
High	Small	77.2%	46.2%	51.0%	36.7%
High	Large	32.5%	11.8%	8.5%	11.6%

Table I shows the average gain for 5 runs of each simulation scenario, for two mobility-based proactive caching demand over total cache storage ratios, low: 0.4 and high: 10, and two video catalog over cache size ratios, low: 21.5 and high: 301. Gain is the reduction of the total cost achieved by a scheme, relative to the total cost when caching is not used. A summary of the main conclusions is the following:

- Low demand for mobility-based proactive caching & small video catalog: This corresponds to the bottom-left area in Figure 1, hence both mobility and popularity-based caching yield benefits. For this reason, the gain of EMPC is highest. The gain of EMC is also high, verifying that it indeed indirectly captures popularity through the aggregate mobile transmission probability in the proactive caching decision (2). The gains of MaxPop and Naive are high, relative to their gains in the other scenarios, indicating that they exploit knowledge of popularity and neighboring caches, respectively.
- Low demand for mobility-based proactive caching & large video catalog: This case corresponds to the bottom-right area in Figure 1, hence only mobility-based proactive caching yields benefits. For this reason, the gains for EMPC and EMC are the highest, significantly higher than MaxPop and Naive. MaxPop has the lowest gains.
- High demand for mobility-based proactive caching & small video catalog: This case corresponds to the top-left area in Figure 1, hence content popularity caching yields the most benefits. For this reason, the gains for EMPC and MaxPop are highest, followed by EMC. The Naive scheme has the lowest gains.
- High demand for mobility-based proactive caching & large video catalog: This case corresponds to the top-right area in Figure 1, hence both mobility and popularity-based caching yield small benefits. Nevertheless, the gains for the EMPC scheme are the highest, but lower than its performance in the other cases. The other schemes have small gains, while the gains of MaxPop are the smallest.

EMPC, which jointly considers mobility-based proactive caching and popularity-based caching, achieves the highest gains in all cases. EMC, which considers mobility-based proactive caching and only indirectly content popularity, exhibits worst performance compared to MaxPop when the demand for mobility-based proactive caching is high, due to the small benefits of mobility-based caching in this case, Figure 1. MaxPop has its highest gains when the video catalog

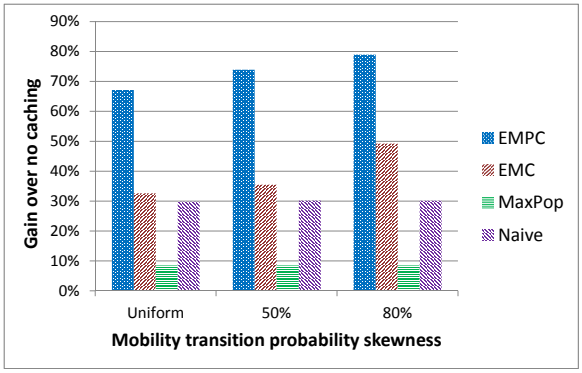


Fig. 2. Impact of mobility skewness on gains.

size is small, since in this case caching the most popular videos is more beneficial compared to the case where the video catalog is large. Finally, the Naive scheme has its highest gains when the demand for mobility-based proactive caching is low, since it considers knowledge of neighboring caches in its caching decisions; however, these gains are smaller than those achieved with EMPC and EMC, verifying that using mobility information can be beneficial.

Figure 2 shows the impact of mobility on the average gain for different mobility skewness, in the case of a low demand for mobility-based proactive caching and a large video catalog. As expected, the results for MaxPop and Naive are independent of the mobility skewness, whereas a higher skewness leads to higher gains for EMPC and EMC.

IV. Exploiting mobility prediction for DASH adaptation

In this section we present our approach to exploit mobility and throughput prediction to select the video quality levels for different video segments that are requested by a DASH player. We assume that mobility and throughput prediction provides the set $X = \{(x_i, t_i^b, t_i^e), i = 1 \dots I\}$, where x_i is the average throughput available to a mobile in mobility period i that begins at time t_i^b and ends at time t_i^e, and I is the total number of periods. Each mobility period can involve different access technologies, such as cellular or Wi-Fi. With DASH, a video file is partitioned into segments that are encoded in different quality levels, each corresponding to a particular bit rate. The video QoE is given by a function $Q(E)$, with $E = \{(e_k, t_k), k = 0 \dots K\}$, where the pair (e_k, t_k) denotes that the video quality level is switched to e_k at time t_k, and K is the total number of quality switches. In Section IV-A we discuss in more detail the QoE model we use, which jointly considers the impact of video quality and video quality switches.

The first step of the proposed procedure is to determine the maximum quality level for which there are no stalls during video playback. This can be determined by considering a buffer which is filled at a rate given by set X and emptied at the average video playout rate for different quality levels. A stall will occur if there is a buffer underflow, i.e. the buffer level becomes zero at some point in time. Let e^* be the highest video quality level for which there are no video stalls. The corresponding QoE will be $Q(\{(e^*, 0)\})$, which indicates that the whole video is played at a single quality e^* from the start, i.e. time $t = 0$, hence there are no video quality switches.

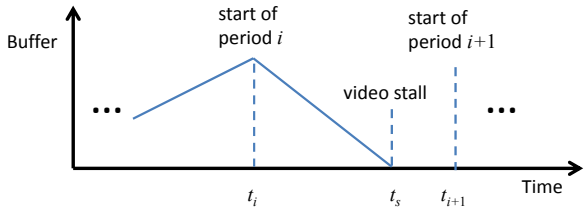


Fig. 3. Scenario with a video stall in mobility period i.

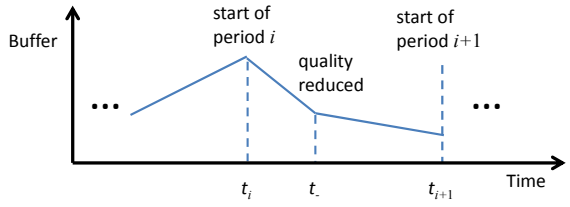
The second step of the proposed procedure involves checking whether switching to a higher quality level $e^+ > e^*$ can improve the QoE. A higher quality level results in better video quality, but there is a cost due to switching. Moreover, a video stall significantly reduces the user's QoE and should be avoided. Hence, to estimate the QoE we need to find when the video quality is switched to a higher level and when it is switched down back to the level for which there are no stalls. Assume that the video quality level is increased from e^* to e^+ at time t^+. Because the playout rate for level e^+ is higher than the rate for level e^*, there will be a time t_s where a video buffer underflow will occur, at which the video will stall, Figure 3. In order to avoid the video stall, at some time $t^- < t_s$ the quality level must be reduced to e^*. One case can be that this stall is avoided if at some point t^- satisfying $t_i^b < t^- < t_s$, the quality is reduced to e^*, where t_i^b is the time that the last mobility period, which includes the stall, begins; this is the case shown in Figure 4(a), where the stall can be avoided by reducing the video quality in the same mobility period, at some time prior to the video stall. If this is not possible, then the quality needs to be reduced when the mobile is in a previous mobility period; Figure 4(b) shows the case where the quality can be reduced in the previous mobility period to avoid the video stall. Based on the above, the video will be streamed at quality e^* from time 0 to t^+, at quality e^+ from time t^+ to time t^-, and then again at quality e^* from time t^-. Hence, the offered QoE is $Q(E)$, with $E = \{(e^*, 0), (e^+, t^+), (e^*, t^-)\}$, which involves two video quality switches. The actual procedure to find the time t^- would start from t_s and move back in time to find the latest point where the quality needs to be reduced to e^* in order to avoid the video stalling.

The approach outlined above can identify the time t^- when it is necessary to switch from the higher quality e^+ to e^*. What remains open is the time t^+ to switch from the lower quality e^* to the higher quality e^+. One approach is to consider all possible values of t^+ in the interval[2] $[0, T]$, where T is the whole duration of the video.

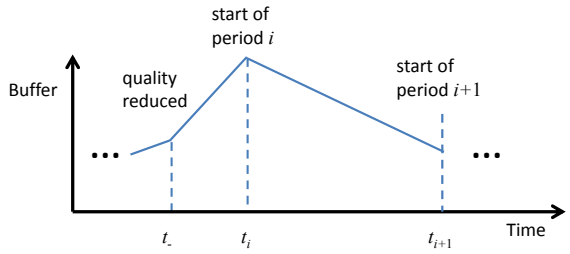
In summary the steps of the procedure discussed above are the following:

1) Identify the maximum encoding level e^* for which no stalls are expected during video playback. The QoE is

[2]Due to the video segmentation, the time obtains values that are multiples of one segment duration.



(a) Quality reduction in same period



(b) Quality reduction in previous period

Fig. 4. Quality reduction to avoid stall in Fig. 3.

$Q(\{(e^*, 0)\})$.

2) For different times $t^+ \in [0, T]$ increase the quality from e^* to e^+ and estimate the time t^- when the quality should be reduced to e^* to avoid video stalls. The QoE is $Q(\{(e^*, 0), (e^+, t^+), (e^*, t^-)\})$
3) Select from Steps 1 and 2 the quality level sequence that gives the highest QoE.
4) Considering the part of the video starting from the time t^- found in Step 3, go to Step 1.

The above procedure involves increasing the video encoding from e^* for which there are no stalls, to the immediately higher encoding level. The procedure can be extended to consider more than one encoding level higher than the baseline level e^*, increasing linearly the procedure's runtime.

Figure 5 shows a typical example of a dynamic (measurement-based) DASH adaptation procedure, which greedily tries to select the highest video quality for each segment. Such a greedy approach can end up performing frequent changes between possibly non-adjacent quality levels, which impacts a user's QoE. On the other hand, a smooth switching approach such as the one performed by the procedure outlined above involves less frequent changes, which are between adjacent video quality levels.

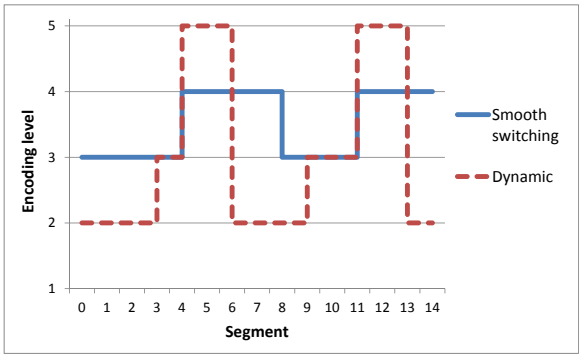


Fig. 5. Video quality levels for smooth switching and for the dynamic adaptation procedure in Sony's Multimedia for Android library.

TABLE II
AVERAGE THROUGHPUT IN DIFFERENT MOBILITY PERIODS.

Time interval (seconds)	Throughput (Kbps)
$0 - 12$	1500
$12 - 22$	600
$22 - 31$	1400
$31 - 45$	500
$45 - 60$	1000

A. Experimental evaluation

Next we present some indicative results comparing the proposed mobility-based smooth switching procedure, the scheme proposed by Miller et al. in [22], which solves an optimization problem where the target objective is a function of the difference in bit rates of the video quality levels involved in each quality switch, and the dynamic adaptation scheme performed by the default adaptation algorithm of Sony's open source Multimedia for Android Library[3]. We developed a DASH player based on this library, which can request video quality levels according to the proposed mobility-based smooth switching procedure and the procedure in the paper by Miller et al. [22]; these two procedures preselect the quality level for each video segment taking as input the throughput prediction, whereas the adaptation algorithm in Sony's library selects the quality level dynamically based on throughput measurements and the video buffer status. Additionally, our DASH player can measure the time and duration of video stalls. The mobile running the DASH player was connected to a Wi-Fi access point which in turn was connected to a server through a link on which we applied the wondershaper traffic shaping tool. The wondershaper tool enforces the throughput that the mobile can achieve in the periods shown in Table IV-A. Note that the time and throughput values in Table IV-A are averages; an important focus of our work is to investigate the impact of time and throughput estimation inaccuracies or errors on the user's QoE.

The video used in our experiments was a 60 seconds Big Buck Bunny clip, with 1 second video segments at 24 fps. Five video encoding levels where available, with bit rates 47, 425, 808, 1312, and 1663 Kbps, and frame resolution from 320x240 up to 1280x720.

The QoE model we considere captures the impact on the user's QoE of both the video quality and the video quality switches. The impact of the video encoding quality is captured through the Video Quality Metric (VQM), which was created by The National Telecommunications and Information Administration (NTIA) and measures the distance of a particular encoding quality from the best quality [27]. An approach to map the VQM to a Mean Opinion Score (MOS), based on user crowdsourcing experiments, is presented in [28]. To capture the impact of video quality switches on a user's QoE, we used the Switching Degradation Factor (SDF) from [29]. The SDF

[3]http://developer.sonymobile.com/2015/02/02/easy-mpeg-dash-streaming-with-multimedia-for-android-library-open-source/

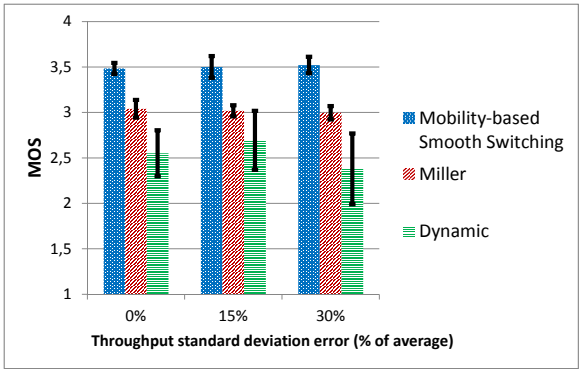


Fig. 6. QoE for different DASH adaptation approaches.

captures the influence of three factors on a user's QoE: the frequency of the video quality switches, the time at which they occur, and the quality levels involved in the switch. Hence, the QoE we consider is given by

$$QoE = M_q(VQM) - M_s(SDF)\,, \tag{4}$$

where $M_q()$ that maps the VQM to a MOS value is from [28], the Switching Degradation Factor (SDF) is from [29], and

$$M_s(SDF) = e^{b \cdot SDF} - c\,,$$

maps the SDF to a MOS value. The function $M_s()$ assumes an exponential dependence of the MOS on the SDF, and includes a parameter b representing a user's sensitivity to video quality switches. In the experiments reported in this paper we considered the values $b = 0.01$ and $c = 0.5$; ongoing investigations are considering additional values that correspond to different user sensitivities to video quality switches.

Figure 6 shows the average QoE of the three approaches investigated and the 95% confidence interval, for 10 executions of each experiment scenario. For each execution the start and end times of each period were randomly selected from a normal distribution with the mean values shown in Table IV-A and standard deviation 4 seconds, whereas the throughput followed a normal distribution with the mean values in Table IV-A and standard deviation 0%, 15%, and 30% of the mean. Figure 6 shows that both our mobility-based smooth switching procedure and the procedure of Miller et al. achieve a higher QoE compared to the dynamic scheme, which indicates that exploiting mobility and throughput prediction can improve the QoE. Moreover, improvements are achieved even when the throughput prediction error is large. Additionally, our results show that the quality switches with the procedure of Miller et al. and the dynamic procedure in Sony's library have a higher impact on the MOS, specifically 1.1 and 1.4 respectively, for throughput standard deviation error 15%, whereas the impact of the proposed smooth switching scheme is 0.6.

V. Conclusions and Future Work

We have presented two directions for exploiting mobility prediction to improve the performance of mobile users. The first direction involves pre-fetching content requested by a mobile in caches located in small cells that the mobile has a high probability to connect to, jointly considering mobility prediction and content popularity. The second direction involves exploiting mobility and throughput prediction for DASH adaptation to improve the QoE in terms of both high video quality and few video quality switches.

Further work on joint mobility and popularity caching, in addition to the evaluation for different mobility models and local (legacy) content request loads, is investigating optimizations for measuring the parameters to estimate the gain. Further work on the proposed DASH adaptation procedure is investigating its extension to incorporate short-term adaptation based on the measured throughput and video buffer, and the application of the QoE model that considers both video quality and video quality switches to dynamic (measurement-based) adaptation approaches. Finally, investigating DASH adaptation jointly with proactive caching for scalable video coding is another interesting research direction that combines the two models presented in this paper.

References

[1] J. Yao, S. S. Kahnere, and M. Hassan, "An Empirical Study of Bandwidth Predictability in Mobile Computing," in *Proc. of ACM WinTech*, 2008.

[2] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting Mobile Connectivity," in *Proc. of ACM MobiCom*, 2008.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker., "Web caching and zipf-like distributions: Evidence and implications," in *Proc. of IEEE Infocom*, 1999.

[4] V. A. Siris, X. Vasilakos, and G. C. Polyzos, "Efficient Proactive Caching for Supporting Seamless Mobility," in *Proc. of IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2014.

[5] ——, "Efficient Proactive Caching for Supporting Seamless Mobility," in *arXiv:1404.4754 [cs.NI]*, 2014.

[6] D. Dimopoulos, C. Boursinos, and V. A. Siris, "Multi-Source Mobile Video Streaming: Load Balancing, Fault Tolerance, and Offloading with Prefetching," in *9th Int'l Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom)*, 2014.

[7] V. A. Siris and D. Dimopoulos, "Multi-Source Mobile Video Streaming with Proactive Caching and D2D Communication," in *Proc. of Workshop on Video Everywhere, co-located with IEEE WoWMoM*, 2015.

[8] V. A. Siris and D. Kalyvas, "Enhancing Mobile Data Offloading with Mobility Prediction and Prefetching," in *Proc. of ACM MobiCom MobiArch Workshop*, 2012.

[9] V. A. Siris and M. Anagnostopoulou, "Performance and Energy Efficiency of Mobile Data Offloading with Mobility Prediction and Prefetching," in *Proc. of Workshop on Convergence among Heterogeneous Wireless Systems in Future Internet (CONWIRE), co-located with IEEE WoWMoM*, 2013.

[10] P. Deshpande, A. Kashyap, C. Sung, and S. Das, "Predictive Methods for Improved Vehicular WiFi Access," in *Proc. of ACM MobiSys*, 2009.

[11] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers," in *Proc. of IEEE Infocom*, 2012.

[12] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, and U. C. Kozat, "Proactive Seeding for Information Cascades in Cellular Networks," in *Proc. of IEEE Infocom*, 2012.

[13] F. Zhang, C. Xu, Y. Zhang, K. K. Ramakrishnan, S. Mukherjee, R. Yates, and T. Nguyen, "EdgeBuffer: Caching and prefetching content at the edge in the MobilityFirst future Internet architecture," in *Proc. of IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[14] J. Yao, S. S. Kahnere, and M. Hassan, "Quality Improvement of Mobile Video Using Geo-intelligent Rate Adaptation," in *Proc. of IEEE WCNC*, 2010.

[15] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen, "Mobile Video Streaming Using Location-Based Network Prediction and Transparent Handover," in *Proc. of ACM NOSDAV*, 2011.

[16] V. Singh, J. Ott, and I. Curcio, "Predictive Buffering for Streaming Video in 3G Networks," in *Proc. of IEEE WoWMoM*, 2012.

[17] D. Jarnikov and T. Ozcelebi, "Client Intelligence for Adaptive Streaming Solutions," *Signal Processing: Image Communication*, vol. 26, no. 7, pp. 378–389, 2011.

[18] A. Bokani, M. Hassan, and S. Kanhere, "HTTP-based Adaptive Streaming for Mobile Clients using Markov Decision Process," in *Proc. of Intl Packet Video Workshop*, 2013.

[19] G. Tian and Y. Liu, "Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming," in *Proc. of ACM CoNEXT*, 2012.

[20] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, pp. 28:1–28:24, Oct. 2015.

[21] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, and R. K. Sinha, "Can accurate predictions improve video streaming in cellular networks?" in *Proc. of Int'l Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2015.

[22] K. Miller, S. Argyropoulos, N. Corda, A. Raake, and A. Wolisz, "Optimal adaptation trajectories for block-request adaptive video streaming," in *Proc. of Packet Video Workshop*, 2013.

[23] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH System," in *Proc. of Multimedia Systems Conference (MMSys)*, 2012.

[24] T. Hossfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, pp. 320 – 332, 2015.

[25] K. V. Katsaros, G. Xylomenos, and G. C. Polyzos, "Globetraff: a traffic workload generator for the performance evaluation of future internet architectures," in *Proc. of IEEE Int'l Conference on New Technologies, Mobility and Security (NTMS)*, 2012.

[26] A. Dhamdhere and C. Dovrolis, "Twelve years in the evolution of the internet ecosystem," *IEEE/ACM Transactions on Networking (ToN)*, vol. 19, no. 5, pp. 1420–1433, 2011.

[27] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, Sept 2004.

[28] A. Devlic, P. Kamaraju, P. Lungaro, Z. Segall, and K. Tollmar, "QoE-aware optimization for video delivery and storage," in *Proc. of IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[29] D. Z. Rodrihuez, Z. Wang, R. L. Rosa, and G. Bressan, "The impact of video-quality-level switching on user quality of experience in dynamic adaptive streaming over HTTP," *EURASIP Journal on Wireless Communications and Networking*, no. 216, 2014.