Energy consumption reduction via context-aware mobile video pre-fetching

Alisa Devlic*†, Pietro Lungaro*, Pavan Kamaraju‡, Zary Segall*, and Konrad Tollmar*
*Mobile Service Lab, Royal Institute of Technology (KTH), Kista, Sweden
†Ericsson Research, Kista, Sweden
‡Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA
Email: devlic@kth.se, pietro@kth.se, pavan4@umbc.edu, segall@kth.se, konrad@kth.se

***Abstract*—The arrival of smartphones and tablets, along with a flat rate mobile Internet pricing model have caused increasing adoption of mobile data services. According to recent studies, video has been the main driver of mobile data consumption, having a higher growth rate than any other mobile application. However, streaming a medium/high quality video files can be an issue in a mobile environment where available capacity needs to be shared among a large number of users. Additionally, the energy consumption in mobile devices increases proportionally with the duration of data transfers, which depend on the download data rates achievable by the device. In this respect, adoption of opportunistic content pre-fetching schemes that exploit times and locations with high data rates to deliver content before a user requests it, has the potential to reduce the energy consumption associated with content delivery and improve the user's quality of experience, by allowing playback of pre-stored content with virtually no perceived interruptions or delays. This paper presents a family of opportunistic content pre-fetching schemes and compares their performance to standard on-demand access to content. By adopting a simulation approach on experimental data, collected with monitoring software installed in mobile terminals, we show that content pre-fetching can reduce energy consumption of the mobile devices by up to 30% when compared to the on demand download of the same file, with a time window of 1 hour needed to complete the content prepositioning.**

***Keywords*-context-aware content pre-fetching; reduction of energy consumption; mobile video**

I. Introduction

We are witnessing an increasing adoption and popularity of mobile data services. Several factors contributed to this phenomenon: (1) arrival of smartphones and tablets that can deliver superior user experience; (2) novel Over-the-Top (OTT) services and content platforms exploiting the app-based content provision model; and (3) a flat rate mobile Internet pricing model. According to a Cisco's study [1], video traffic accounted for 52% of the mobile data traffic at the end of 2011 and will account for two thirds (over 70%) of the world's mobile data traffic by 2016.

The majority of video content being streamed over the Internet today is encoded in the 1000-8000 kbps range [2]. In order to start streaming a video file on a mobile device instantaneously and without playback interruptions, the mobile device's bandwidth needs to be equal to or larger than the bit rate of the encoded video.

New generation of mobile communication systems (e.g., HSPA and LTE) offers downlink data rates up to 100 Mbit/s to mobile users. However, streaming a medium/high quality video in mobile environments can be a challenge, where available capacity needs to be shared among large number of users, intermittent connectivity occurs due to high mobility and low signal coverage, and handover to base stations of different connectivity types and vendors can cause variations in data rates, interruptions, and long waiting queues, resulting in a low quality of experience (QoE). QoE is defined as the overall acceptability of an application or a service, as perceived subjectively by the end user [3].

This paper presents a family of content pre-fetching methods that exploit the times and locations with high data rates to deliver the content before the user requests it (the concept known as content pre-fetching [4]), having a potential to reduce the energy consumption by shortening the time to transfer the data over the radio interface [5] and improve a user's QoE by playing the pre-fetched content from the terminal memory without interruptions on the user's request.

This paper shows that pre-fetching content at high data rates only (69% larger than average) can reduce energy consumption in a mobile device up to 72% when compared to the download of the same content on demand, initiated at the same time. For the cost of a single file download, 3 video files of the same size (160MB) can be pre-fetched. However, this scheme (operator-like prefetching) requires an accurate and a priori knowledge of users' data rates, which might not often be feasible.

The second proposed method (OTT pre-fetching) uses a mobile device to estimate the achievable data rates by periodic probing of the channel quality and combines this probing phase with the transfer of the remaining content bits at target data rates. Whenever probing reveals low achievable data rates, the data retrieval operation is paused, in order to limit a potential increase in the energy consumption associated with a file download. This energy cost can potentially be reduced by reducing the frequency of the probing phase, however if network resource probing is too seldom, some good channel conditions might be missed, potentially leading to a reduced goodput. We illustrate this trade-off in the paper and estimate the energy costs & the time that is needed to complete the content prepositioning using

978-0-7695-4875-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ISM.2012.56

different polling strategies. With this combined information, content providers can decide when to schedule pre-fetching that can result in more energy-efficient or (delivery) delay-sensitive mobile video delivery.

The experiments show that OTT content pre-fetching can reduce by up to 30% energy consumption when compared to the on demand download of the same file (performed at the same wide area network interface), with a time window of 1 hour needed for content prepositioning. The success rate of the OTT pre-fetching is shown to be by up to 32% higher than the success rate of the random access strategy.

Content pre-fetching and its impact on energy savings have been investigated in many related works. In [6], [7] pre-fetching is scheduled based on predictions of WiFi availability and cellular signal strength, respectively, achieving up to 60% energy savings. In another work [8] pre-fetching is based on predicting what data is needed and when it will be used, by observing the user behavior and availability of WiFi connectivity, power & signal strength at different locations, thus achieving up to 70% savings. However, while WiFi availability can be used as indicator of high data rates, signal strength cannot indicate variations in the user bandwidth that occur due to sharing of aggregated cell capacity with others. We consider only downlink data rates to schedule pre-fetching, investigating the energy savings based on frequency of context probing and target pre-fetching data rate, which to our knowledge has not previously been studied.

II. Context-aware content pre-fetching

Pre-fetching is a content-delivery method which decouples the time when a user requested content is downloaded to the user's terminal from the time when this content is accessed and consumed by the user. The benefit of content pre-fetching is that it reduces the time that is needed to access the content (once the content is pre-fetched and cached in the terminal), thus improving the user's QoE.

In order to increase the likelihood of serving the user requested content from the cache and avoid a potentially high communication and energy cost associated to pre-fetching of large amounts of data on the mobile device, it is important to carefully consider the type of data that should be pre-fetched (by anticipating the user's future requests).

Content providers, like Amazon or YouTube, already have some knowledge about their users' preferences, which they can use to carefully select the data for pre-fetching [9] [10]. However, reducing the amount of data for pre-fetching cannot itself reduce the time that is needed to download the data. Instead, content providers need to determine the best times or conditions to perform the pre-fetching that will result in more energy efficient content delivery for the user. Context-aware content pre-fetching considers context information (i.e., any information about the state of resources in the device, network, or user mobility) to decide when to initiate and when to stop content pre-fetching.

A. Pre-fetching methods

This subsection introduces two context-aware pre-fetching methods that initiate pre-fetching when a user download data rate reaches or exceeds the target prefetching data rate ($\hat{R}$). They differ in the way how they obtain a user data rates.

1) *Operator-like prefetching (OP PRE)* - the method representing the view that is closest to a mobile operator, with a detailed a priori knowledge of the users' connectivity and data rates.
2) *Over-the-Top prefetching (OTT PRE)* - the method that is envisaged to run on mobile devices, *without* any prior knowledge of connectivity or data rates. It is based on periodically probing the channel quality to estimate the achievable data rates.

In contrast to pre-fetching, on demand download downloads the content independently of the data rates performance. The difference between the prefetching and download is illustrated in Figure 1, using the following metrics: *pre-fetching SLA*, *pre-fetching cost*, and *downloading time*.

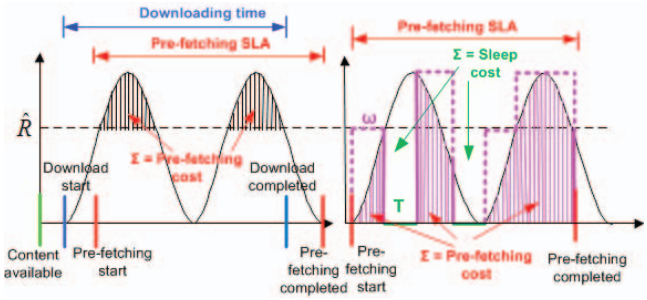

Figure 1. Evaluation metrics for content downloading and pre-fetching

The *pre-fetching SLA* represents the duration from the start until the end of content pre-fetching, which is initiated by a specific condition (e.g., data rate threshold or periodic time interval) and which needs to be completed before the content is offered to the user for download/viewing.

The *pre-fetching cost* refers to the time spent actively pre-fetching the content, with the OP PRE representing the *ideal case*, where the pre-fetching is performed *only* at data rates that are equal to or above $\hat{R}$ (shown in Figure 1 to the left).

The *downloading time* denotes the time that is needed to download the content on demand.

Besides $\hat{R}$, the OTT PRE uses two additional parameters to implement periodic probing: the wake up time (ω) and the sleep time (τ). During ω, the method pre-fetches bits, computes the data rate during this period, and checks if the obtained data rate is equal to or above $\hat{R}$. If this is the case, it continues pre-fetching bits until the end of file transfer round; otherwise it goes to sleep for τ seconds, stopping the pre-fetching of the content until this time expires, after which the pre-fetching period is restarted (see Figure 1 to

the right). The total sleep time during which the pre-fetching was stopped is referred to as the *sleep cost*.

The benefit of performing periodic channel probing is being able to estimate the achievable data rates without involvement of a mobile operator and providing the pre-fetching conditions directly to the content provider. However, periodic checking of data rates has the associated energy cost of pre-fetching some of the content bits at data rates that are *lower* than $\hat{R}$. This cost can potentially be reduced by reducing the probing frequency, hence with a potential risk of missing the pre-fetching opportunity if the device is not frequently exposed to the target data rates. The number of pre-fetching opportunities can potentially increase if $\hat{R}$ is carefully chosen to reflect the frequency of the device experiencing the same data rates throughout the day.

In the remaining of the paper we will experimentally determine the potential energy savings that can be achieved by OTT PRE and OP PRE with respect to the on demand download. Given that the OP PRE results will reveal the upper bound of the content pre-fetching performance that can be obtained in the ideal scenario, this comparison will tell us if the data rates can be determined by the device itself, without incurring a large additional energy cost. Additionally, the time needed to complete pre-fetching will be estimated for different $\hat{R}$ and τ values.

III. Method

The method used in this paper consists of collecting the user data on a mobile device, sending this data to the external server, and processing this data - i.e., simulating different content delivery methods using the traces we collected, rather than executing these methods "live" on the mobile device. This enables us to compare the performance of these methods (in terms of energy savings) on identical user data in order to ensure an accurate comparison.

We used the COSEM living laboratory testbed [11], consisting of a server and an Android application, to periodically send a file of predefined size to a mobile device and collect data that monitors the device connectivity performance. The server deploys the tasks to mobile devices, processes the incoming information, and displays the collected data on the web interface. The application collects every second the following information from the device: the number of sent and received bytes from the server, the terminal location, and the connectivity type with the corresponding absolute times, sending the log every 15 minutes to the server.

The collected data was processed in Matlab. The number of received bits in each second of the file transfer with the corresponding timestamps were extracted from the log and used to compute the download data rates. These data rates were used as an input into a set of simulations developed for on demand download, OP PRE, and OTT PRE.

The start times of simulation were chosen randomly from the user log for each realization, being used as a reference point for all content delivery methods. When the end of the log file was reached, the execution continued from the beginning of the file again.

We ran 10000 realizations of the described algorithms on a user trace, recorded the results, and averaged them across all runs. Each realization consisted of delivery of a 160MB file, which corresponds to an average high quality YouTube video of approximately 4 minutes encoded at 5512kbps (5000 kbps for video and 512 kbps for audio stream).

IV. Experiment

We illustrate our method on the example of a single user that was periodically downloading a video file of 13MB, with 10s of pause time after the downloading round has been completed. The experiment was run for 3 days on a Samsung Galaxy SII phone running Android version 2.3. During these 3 days the user's device was connected to Internet through the mobile access networks only. Connecting the device to WiFi would increase the energy reduction to even greater extent, therefore this represents the worst case scenario.

The data rate log used in the experiment is illustrated on Figure 2, from which the pause times have been removed. It can be observed that data rates are in range from 10 kByte/s up to 950 kByte/s, with the average data rate of 280 kByte/s.

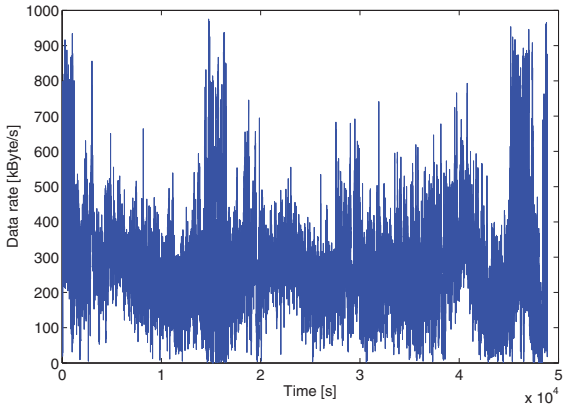

Figure 2. Data rate vs. time log used in the experiment

Figure 3 illustrates the average pre-fetching costs for OTT PRE and OP PRE, obtained using different $\hat{R}$ and τ values. Since the OP PRE is always performed on data rates that are equal to or greater than $\hat{R}$, the prefetching cost decreases when increasing $\hat{R}$.

Figure 3 also shows that at $\hat{R}$ of 900kByte/s the OP PRE can reduce the energy consumption in the mobile device by up to 72% when compared to the downloading time of the same content on demand. For the cost of downloading a single file on demand 3 video files of the same size can be pre-fetched at the average prefetching data rate 780 kByte/s (see table I). However, this scheme requires full knowledge of available data rates in time, which is often not feasible.

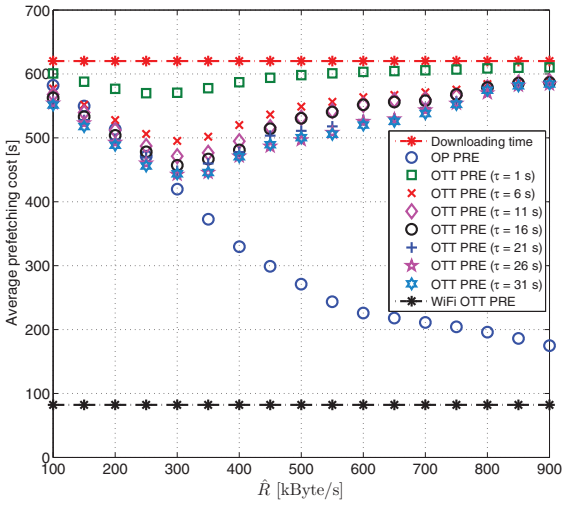

Figure 3. Average prefetching costs

Table I
NUMBER OF FILES THAT CAN BE PREFETCHED USING OP PRE FOR THE COST OF SINGLE DOWNLOAD

Target data rate [kByte/s]	100	300	500	700	900
Number of files	1.07	1.48	2.29	2.94	3.54
Average data rate [kByte/s]	297	400	609	777	936

The upper line in Figure 3 depicts the average (on demand) downloading time, while the lower line shows the OTT PRE cost in the case if the user would have been connected to WiFi. It is interesting to notice that all the OTT PRE costs tend to converge to the downloading time value in case of very high $\hat{R}$. This can be explained by a few or even none of the available data rates being equal to or higher than 900 kByte/s. In case a device would have a WiFi interface turned on, the actual pre-fetching cost would depend on the duration of the user stay under the coverage of WiFi APs during the pre-fetching period.

The OTT PRE costs (shown in Figure 3) decrease until the point where most of the available data rates are equal to or greater than $\hat{R}$. This point is depicted as the minimum of prefetching costs curves, at which the lowest pre-fetching cost can be achieved. After this point the availability of data rates that meet the required threshold decreases, causing the increasing amount of bits to be prefetched on lower data rates during the wake up time period, which in turn increases the prefetching cost. The lowest pre-fetching cost can be observed at the $\hat{R}$ of 300 kBytes with the τ of 31 second. When compared to the downloading time, the energy consumption reduction of 30% can be achieved.

Figure 3 also illustrates that the OTT PRE cost at a particular $\hat{R}$ decreases with the higher τ values. This can be explained by the fact that when the OTT scheme detects the low data rates, it pauses the pre-fetching for τ seconds before it wakes up to pre-fetch bits again. The longer the sleep time is, the more likely is that data rates will be uncorrelated with those from the previous wake up period. Thus, assuming the low data rates were detected in the current wake up period, a longer sleep time will increase the probability of moving away from these low data rates.

However, this scheme does not provide any guarantees that we will hit high data rates while periodically prefetching bits. In fact, by sleeping longer we also miss some opportunities for pre-fetching at high data rates, causing the content to be pre-fetched mostly during the probing windows of 1 second, thus resulting in the reduced goodput.

Despite the outlined limitations, our proposed OTT scheme results in potential energy cost reductions. To evaluate the success rate of the proposed OTT scheme, we computed it and compared it to the random access strategy (RA). The RA represents the uniform probability of finding data rates that are equal to or greater than $\hat{R}$ in a time window that is equal to the pre-fetching SLA. The success rate of the RA method is calculated as the number of seconds the data rates were equal to or above the threshold during the pre-fetching period, divided by its duration. The OTT success rate was calculated as the ratio of the number of seconds the prefetching was performed on the data rates that are equal to or greater than $\hat{R}$ and the pre-fetching cost at this $\hat{R}$. Figure 4 illustrates this comparison result.

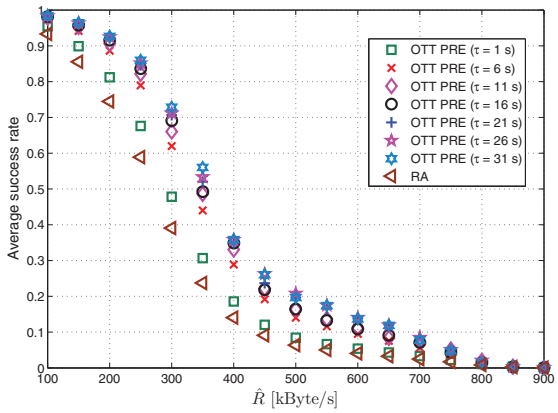

Figure 4. OTT prefetching compared to Random access strategy

Figure 4 shows that the OTT pre-fetching scheme has a higher success rate than the RA scheme for all $\hat{R}$ up to 850 kByte/s, where they converge. At the $\hat{R}$ of 300 kByte/s the OTT success rate is up to 32% greater than the RA's rate, at which point the success rate of the RA's scheme is already at 40%. The success rate of OTT scheme falls below 50% between 350 and 400 kByte/s, which is reflected in Figure 3 by an increase in pre-fetching cost.

Figure 5 illustrates the average pre-fetching SLAs for the

same $\hat{R}$ and τ values as in Figure 3. Using these values, pre-fetching can be scheduled *in advance* to complete before the content is offered to the user for download/viewing.

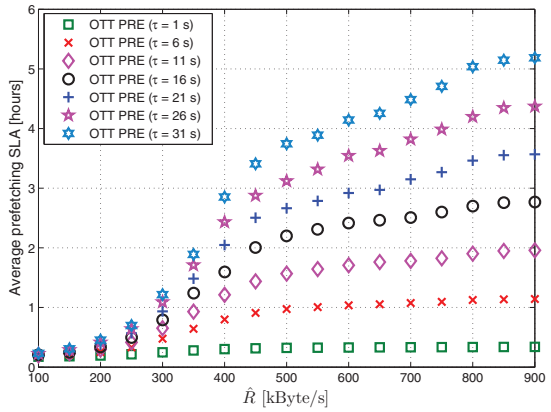

Figure 5. Average prefetching SLAs

In this scenario, when selecting the highest $\hat{R}$ of 900 kByte/s and τ of 31 second, the pre-fetching should be scheduled 5.2 hours in advance of offering the content to user for download/viewing. In order to achieve the lowest prefetching cost, the prefetching should be scheduled 1.2 hours earlier, using $\hat{R}$ of 300 kByte/s and τ of 31 second.

This paper does not provide recommendations to content providers about which target data rates or sleep times to use. Instead, it provides the estimated pre-fetching costs and the pre-fetching completion time. With this information content providers can make more informative decisions about scheduling of content pre-fetching that can improve the user's QoE (by making the prefetching more energy efficient or more (delivery) delay sensitive for their users). We envisage the content provider or the end user to provide their requirements about the maximum delivery delay and/or energy cost, using the potentially new types of SLAs.

V. Conclusion

This paper proposes a simple content prefetching scheme that is based on the target prefetching data rate and the sleep time. The potential energy cost reduction of 30% can be achieved when compared to download of the same content on demand, using the same wide area network interface and a time window of 1 hour needed to complete pre-fetching.

The proposed scheme yields the highest energy cost reductions for target pre-fetching data rates that are closer to the user's average data rate and for shorter channel probing frequencies that allow one to faster avoid the poor network conditions, thus increasing the likelihood of pre-fetching at higher data rates. The success rate of the proposed method is by up to 32% higher than the success rate of the random access strategy. The obtained results are specific to the user and his/her pattern of achievable data rates, however we have observed that the larger variation in data rate distribution results in the larger energy reduction gains.

The proposed scheme can be learned by a terminal, by probing the data rates and applying the described method to their scenario. Our next steps include enhancing our method with additional information (i.e., signal strength, location, and multiple radio access networks), enabling pre-fetching based on predicted context information, and comparing the obtained energy savings with the results from this paper.

References

[1] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016. White paper.

[2] YouTube. (2012) Video encoding, Suggested resolutions and bitrates. [Online]. Available: http://support.google.com/youtube/bin/static.py?hl=en&page=guide.cs&guide=1728585&topic=1728588

[3] ITU - International Telecommunication Union, "Definition of Quality of Experience (QoE)," Reference: TD 109rev2 (PLEN/12), Jan. 2007.

[4] P. Lungaro, Z. Segall, and J. Zander, "Context-aware RRM for opportunistic content delivery in cellular networks," in *Proc. of the 3rd International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2010)*, Athens, Greece, Jun. 2010, pp. 175–180.

[5] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. ACM SIGCOMM Internet Measurement Conference (IMC'09)*, Chicago, Illinois, USA, Nov. 2009, pp. 280–293.

[6] A. Rahmati and L.Zhong, "Context-based network estimation for energy-efficient ubiquitous wireless connectivity," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 54–66, 2011.

[7] A. S. et al., "Bartendr: A practical approach to energy-aware cellular data scheduling," in *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom 2010)*, Chicago, Illinois, USA, Sep. 2010, pp. 85–96.

[8] N.H.Walfield and R.Burns, "Smart phones need smarter applications," in *Workshop on Hot Topics in Operating Systems (HotOS 2011)*, Napa Valley, CA, May 2001, pp. 1–5.

[9] J. Davidson, B. Liebald, J. Liu, P. Nandy, and T. V. Vleet, "The youtube video recommendation system," in *Proc. ACM Conference on Recommender Systems(RecSys'10)*, Barcelona, Spain, Sep. 2010, pp. 293–296.

[10] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[11] P. Lungaro, C. Viedma, Z. Segall, and P. Kumar, "An experimental framework to investigate context-aware schemes for content delivery," in *Proc. IEEE Vehicular Technology Conference (VTC Fall)*, San Francisco, CA, USA, Sep. 2011, pp. 1–5.