Evaluation of Energy Profiles for Mobile Video Prefetching in Generalized Stochastic Access Channels

Alisa Devlic(✉), Pietro Lungaro, Zary Segall, and Konrad Tollmar

Mobile Service Lab, Royal Institute of Technology (KTH), Kista, Sweden
{devlic,pietro,segall,konrad}@kth.se

Abstract. This paper evaluates the energy cost reduction of Over-The-Top mobile video content prefetching in various network conditions. Energy cost reduction is achieved by reducing the time needed to download content over the radio interface by prefetching data on higher data rates, compared to the standard on demand download. To simulate various network conditions and user behavior, a stochastic access channel model was built and validated using the actual user traces. By changing the model parameters, the energy cost reduction of prefetching in different channel settings was determined, identifying regions in which prefetching is likely to deliver the largest energy gains. The results demonstrate that the largest gains (up to 70 %) can be obtained for data rates with strong correlation and low noise variation. Additionally, based on statistical properties of data rates, such as peak-to-mean and average data rate, prefetching strategy can be devised enabling the highest energy cost reduction that can be obtained using the proposed prefetching scheme.

Keywords: Energy profiles · Stochastic access channel · Mobile video prefetching

1 Introduction

1.1 Motivation

We are witnessing a large increase in mobile Internet data traffic in the last years, with predictions to increase 18-fold by 2016 (i.e., reaching 10.8 exabytes per month) [2]. The following trends contributed to this phenomenon: an increasing number of powerful mobile Internet devices (such as tablets and smartphones) that can deliver superior user experience, faster Internet connections, and a large amount of video streaming content available. According to a Cisco's study [2], video 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.

Video streaming in mobile environments can be a challenge, due to sharing of available capacity among large number of users and intermittent connectivity.

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 209–223, 2014.
DOI: 10.1007/978-3-319-11569-6_17

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. Content prefetching addresses these problems by decoupling the time when the content is prepositioned on a user's terminal from the time when this content is accessed and consumed by the user. By exploiting the times and locations with high data rates to prefetch content, the time needed to transfer data over a radio interface is reduced, resulting in energy consumption reduction when compared to the standard on demand access to content [1].

Content prefetching and its impact on energy savings have been investigated in many related works. In [7,8] prefetching is scheduled based on predictions of WiFi availability and cellular signal strength, respectively, achieving up to 60 % energy savings. In another work [6] prefetching is based on predicting what data is needed and when it will be used, by observing a user behavior and availability of WiFi connectivity, power & signal strength at different locations, thus achieving up to 70 % savings. N. Gautam and his colleagues [5] showed that energy savings of 84 % can be achieved by video prefetching over WiFi when compared to streaming over 3G, due to the high download data rates. However, while WiFi availability can be used as indicator of high data rates, its use is limited to the user's stay duration under the coverage of WiFi AP. Signal strength, on the other hand, *cannot indicate variations in a user bandwidth* that occur due to sharing of aggregated cell capacity with others. Even if signal strength is strong, the available bandwidth might be low, resulting in potentially increased energy consumption, if prefetching is performed under this condition.

1.2 Contribution

This paper evaluates the energy cost reduction of content prefetching in various network conditions. It generalizes results from our previous paper [3], where an opportunistic OTT context-aware mobile video pre-fetching scheme has been proposed and evaluated on a single user data rates log. Prefetching was scheduled based on *downlink data rates*, while the potential energy savings were investigated based on frequency of probing a channel quality and setting a target pre-fetching data rate, which to our knowledge has not previously been studied.

Note that obtaining a mobile user data rate traces on a large scale in per-second granularity is economically very costly. Therefore, in order to derive some conclusions about the prefetching performances, we synthetically generated data rates. A simple model, an autoregressive process of order 1, was used to simulate different network conditions and user behavior. Ten of actual user traces that were collected in Stockholm city area were fitted to this model with an error of up to 10 %. Due to having only three parameters, it was easy to generate data rates simulating different access channel states. Note that we do not claim that this model can accurately describe all channel conditions nor do we compare it with other models. Hence, it enabled us to evaluate prefetching performances of the data rates that can be fitted to this model, identifying regions with potentially largest energy reduction gains.

Finally, we found a dependency of a target prefetching data rate at which the maximum energy cost reduction is obtained to statistical data rate properties, such as mean and peak-to-mean data rate ratio. By combining these parameters, we created optimization guidelines for reducing energy cost that a mobile device can employ when prefetching video.

2 OTT Prefetching Scheme

This section briefly describes the over-the-top context-aware content prefetching (OTT PRE) scheme adopted from our previous paper [3]. The prefetching 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, combining this probing phase with the transfer of the remaining content bits at data rates that are equal to or higher than the target prefetching data rate ($\hat{R}$). Whenever probing reveals low achievable data rates (i.e., lower than $\hat{R}$), the data retrieval operation is paused in order to limit a potential increase in energy consumption associated with a file download.

On demand download downloads content *independently* of the data rates. The difference between prefetching and download is shown in Fig. 1, using the following metrics: *prefetching SLA*, *prefetching cost*, and *downloading time*.

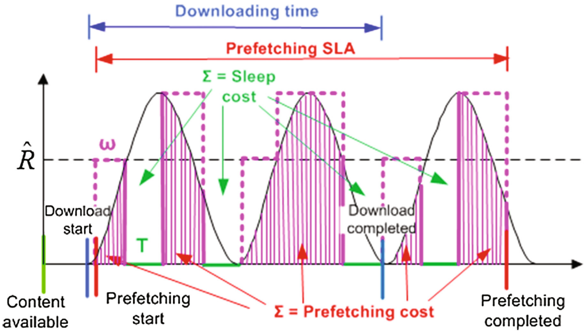

Fig. 1. Evaluation metrics for content downloading and prefetching

The *prefetching SLA* represents the duration from the start until the end of content prefetching, 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 *prefetching cost* refers to the time spent actively prefetching the content, while 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 prefetches 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 prefetching bits until the end of file transfer round; otherwise it goes to sleep for τ seconds, stopping the prefetching of the content until this time expires, after which prefetching is resumed. The total sleep time during which the prefetching was stopped is referred to as the *sleep cost*.

The benefit of periodic channel probing is estimating the achievable data rates without involvement of a mobile operator and using the estimated data rates as context information to drive the prefetching. However, periodic probing of data rates has the associated energy cost of prefetching 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 prefetching opportunity if the device is not frequently exposed to the target data rates. The number of prefetching opportunities can potentially increase if $\hat{R}$ is carefully chosen to reflect the frequency of the device experiencing the same data rates throughout a day. Therefore, it is important to: (1) evaluate if a device can estimate the data rates and achieve energy cost reduction while prefetching content, (2) determine under which channel conditions is the potential energy cost reduction the highest, and (3) estimate the corresponding prefetching SLA.

3 Actual User Data Rates

This section briefly examines the data rate logs that were collected by different users in Stockholm city area during different times of the year (see Table 1).

Table 1. Mobile user data rate logs

User ID	File size	Pause	Duration	Data rate range	Average data rate	Data plan
1	13 MB	10 s	3 days	[0.02–974.8] kByte/s	280.8 kByte/s	5 GB
2	5 MB	2 s	3 days	[1.4–350.7] kByte/s	163.6 kByte/s	5 GB, EDGE
3	50 MB	2 min	3 days	[0.4–976.5] kByte/s	540.7 kByte/s	50 GB
4	100 MB	2 min	2 days	[0.7–976.3] kByte/s	388.2 kByte/s	50 GB
5	200 MB	2 min	5 days	[0.1–976.5] kByte/s	646.3 kByte/s	50 GB
6	50 MB	2 min	3 days	[0.7–976.4] kByte/s	522.6 kByte/s	50 GB
7	50 MB	2 min	4 days	[1.4–974.9] kByte/s	707.2 kByte/s	no limit
8	50 MB	5 s	2 days	[1.0–970.8] kByte/s	628.7 kByte/s	no limit
9	50 MB	2 min	3 days	[0.6-976.5] kByte/s	550.9 kByte/s	50 GB
10	13 MB	5 s	8 days	[0.02–976.5] kByte/s	464.5 kByte/s	5 GB

The logs were collected by mobile devices periodically downloading a video file of a predefined size from a server for a couple of days and pausing for a predetermined time after completing each downloading round. During the experiment phones were connected to Internet through the mobile access networks only. The users that performed the logging were researchers from our University, a research centre in Kista, and one working professional, whose routes and stay durations at particular locations are shown in Fig. 2.

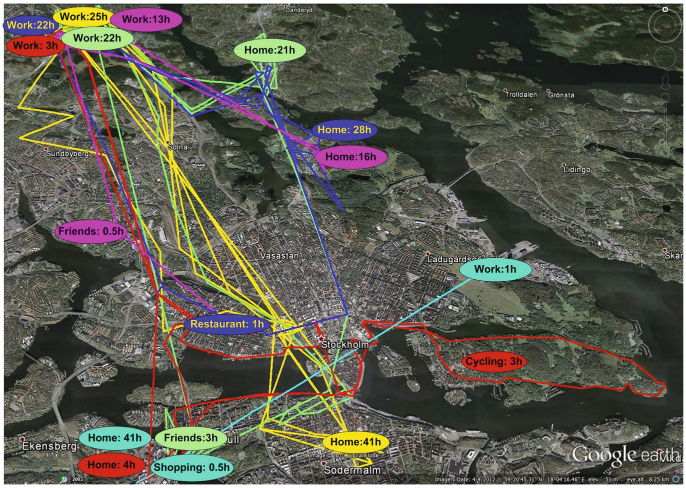

Fig. 2. Users mobility routes

Figure 3 shows a data rate log of user 1, from which pause times have been removed. Data rates were recorded every second, representing a sequence of data points at equally spaced time intervals. This motivated us to examine the time series models as a candidate for generating synthetic data rates.

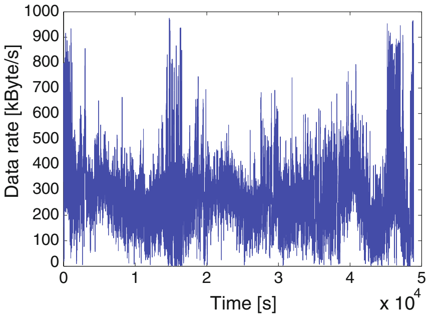

Fig. 3. Data rate vs. time log used in the experiment

Time series analysis consists of methods for extracting meaningful statistical properties and other characteristics of this data. It assumes that there is some internal structure, such as autocorrelation, trend, or seasonal variation that should be considered when analyzing or modeling such data. Our work has been driven by similar assumptions – that data rates exhibit certain statistical properties that can be used to identify a generative process from which our experimental data are drawn, representing one out of many realizations of this process.

Our **first goal** was to identify the underlying process and its parameters, in order to: (1) generate synthetic data rates whose prefetching results will be compareable to the results obtained with actual user data rates and (2) estimate the maximum energy cost reduction for different parameter values. The **second goal** was to relate the target prefetching data rate at which the maximum energy cost reduction can be achieved to statistical properties of data rates.

4 Method

As a first step in determining if there is some underlying stationary process that generated our experimental data, we checked if the data rate values arranged in time exhibit some serial correlation. To answer this question, we plotted the user data rates at time *t* against the data rates in previous period *t-1*, as shown in Fig. 4 on the example of user 1. A strong serial correlation between the current and previous data rate is indicated by the slope of linear regression line.

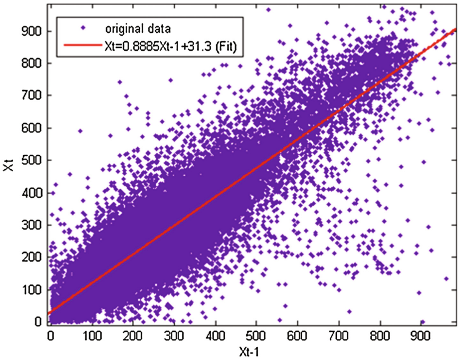

Fig. 4. Serial correlation between current and previous data rate

In order to identify the appropriate time series model for the data, we plotted the autocorrelation function (ACF) (see Fig. 5 to the left). The ACF plot illustrates exponential decay, indicating that our data can potentially be described by autoregressive (AR) process. In AR model future values depend on past time series values, while its order indicates how many lags in past they depend on.

The partial autocorrelation function (PACF), illustrated in Fig. 5 to the right, is used to determine the order of AR model, *p*. PACF removes the effects of the shorter lag autocorrelation from the correlation estimate at longer lags, cutting off abruptly to zero after lag *p*. By looking at the lag where PACF falls (close to) zero, we can conclude that the order of AR process should be 1.

AR(1) process is defined by a first order linear difference equation:

$$X_t = c + \phi_1 X_{t-1} + \epsilon_t \tag{1}$$

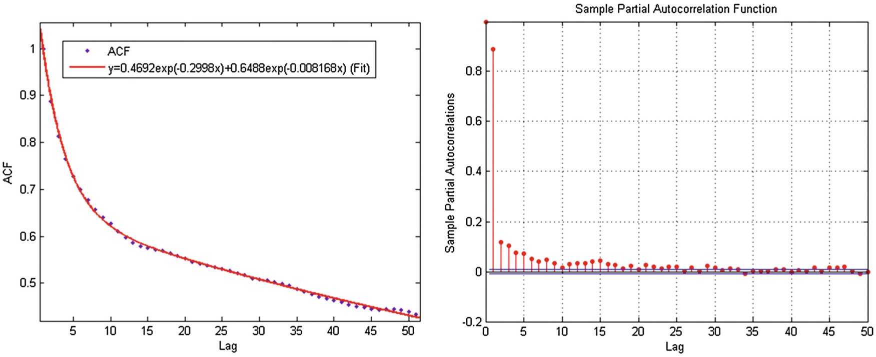

Fig. 5. Autocorrelation and partial autocorrelation function of actual data rates

where t is a point in time, c is constant, ϕ_1 is the autoregressive coefficient, and ϵ_t are Gaussian distributed error terms or innovations with zero mean and variance σ_ϵ^2 that introduce variability into the process.

Since X_t is a stationary process, its expected value does not change over time. Inserting $E[X_t] = E[X_{t-1}]$ into (1), we obtain:

$$E[X_t] = \mu = \frac{c}{1-\phi_1} \tag{2}$$

The autocovariance of X_t at lag s for $s\neq 0$ indicates how much a random variable changes with the time-shifted version of itself:

$$\gamma(s) = \phi_1 \gamma(s-1) \tag{3}$$

Raising (3) on the power of two gives the autocovariance of X_t at lag 0, which is the variance of AR(1) process:

$$\gamma(0) = Var(X_t) = \phi_1^2 Var(X_{t-1}) + \sigma_\epsilon^2 \tag{4}$$

Since $Var(X_t) = Var(X_{t-1})$, variance becomes:

$$Var(X_t) = \sigma_{AR}^2 = \frac{\sigma_\epsilon^2}{1-\phi_1^2} \tag{5}$$

We can now derive the equation for ϕ_1:

$$\phi_1 = \sqrt{1 - \frac{\sigma_\epsilon^2}{\sigma_{AR}^2}} \tag{6}$$

From (2) and (5) follows that AR(1) process can be described with the process mean μ, process variance σ_{AR}^2, and noise variance σ_ϵ^2. This approach has been adopted for modeling our data rates signal, as illustrated in Fig. 6.

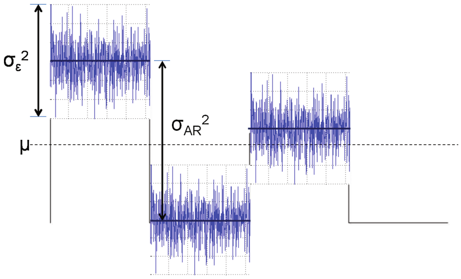

Fig. 6. Data rates described using AR(1) parameters

Data rates are fitted to AR(1) using Burg method, minimizing sum of squares of the error between original and estimated values [9]. $\hat{\epsilon_t}$ were estimated using:

$$\hat{\epsilon_t} = X_t - \hat{\phi_1} X_{t-1} - \hat{c} \tag{7}$$

and fitted to Gaussian probability distribution in order to obtain the noise mean ($\hat{\nu}$) and variance ($\hat{\sigma_\epsilon^2}$), checking if these innovations are uncorrelated.

The fitting results for all users data rates are shown in Table 2, showing that innovations mean values are close to zero, thus can be approximated by the white noise. Normalized Root Mean Square Error represents the fitting error (err).

Table 2. Fitting AR(1) parameters and residuals for mobile user data rate logs

User	1	2	3	4	5	6	7	8	9	10
$\hat{\phi_1}$	0.8885	0.5407	0.8899	0.8344	0.8228	0.8410	0.6276	0.6545	0.8966	0.7878
$\hat{c}$	31.3	75.1	59.5	64.3	114.5	83.1	263.4	217.2	56.9	98.6
$\hat{\sigma_\epsilon^2}$	4032.4	1318.7	11764.1	11040	10644.6	14558.7	18755.8	22313.6	9602.3	23041.3
$\hat{\nu}$	−4.6e-4	4.7e-3	−2.9e-3	−0.019	−3.2e-12	−1.9e-3	−1.2e-12	−4.8e-3	2.5e-3	2.9e-3
err	4.56 %	9.27 %	5.92 %	7.06 %	5.82 %	7.10 %	10.13 %	10.84 %	5.78 %	8.75 %

5 Prefetching Results

100 realizations of synthetic data rates were generated using AR(1) model with $\hat{\phi_1}$, $\hat{c}$, and $\hat{\sigma_\epsilon}$ parameters obtained from fitting the actual user data rates. 10000 iterations of content prefetching and on demand download were performed over each synthetic data rate realization with different starting indices. The target prefetching data rate was set to range from 100 kByte/s up to 800 kByte/s with a step of 50 kByte/s, while the sleeping time was 1 s up to 31 s with a step of 5 s at the end of each prefetching round (the same as in [3]). From each iteration we extracted the prefetching costs, downloading time, and prefetching SLAs obtained for different target prefetching data rates and sleep times (as defined in Sect. 2). Using these values the maximum energy cost reduction (E_{max}) was

computed in each iteration by finding the lowest prefetching cost that can be achieved by reduction in the duration of the content download time:

$$E_{max} = \frac{download\_time - min(prefetching\_cost)}{download\_time} \qquad (8)$$

Next, the minimum and maximum of 10000 E_{max} values were computed in order to obtain the E_{max} range for the fitted set of AR(1) parameters. The obtained E_{max} range is compared with the E_{max} obtained using the actual user data rates, resulting in 25.3–27.7 % and 30 %, respectively.

Table 3 shows the E_{max} obtained from prefetching over actual and fitted data rates of ten mobile users, demonstrating that the results are comparable.

Table 3. E_{max} obtained from prefetching over actual and fitted users data rates

User	1	2	3	4	5	6	7	8	9	10
E_{max} actual	30 %	7.2 %	23.6 %	29.6 %	15 %	23.7 %	6.6 %	11.9 %	29.8 %	32.1 %
E_{max} synthetic min	25.3 %	6.5 %	20.6 %	21.5 %	12.2 %	18.3 %	6.2 %	8.8 %	19.8 %	18.2 %
E_{max} synthetic max	27.7 %	8.8 %	22.4 %	26.8 %	13.3 %	20 %	7 %	10.6 %	22 %	19.6 %

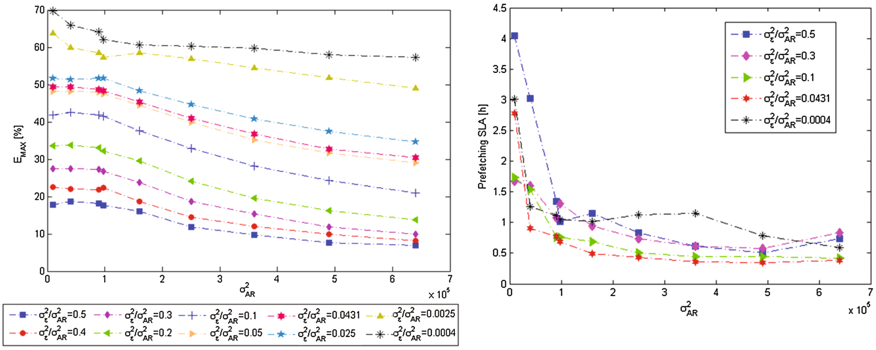

Fig. 7. E_{max} and prefetching SLA as a function of σ^2_{AR}

In order to simulate different access channel states and user behavior, we scanned the entire parameter space of the identified AR model (μ, σ^2_{AR}, and σ^2_{ϵ}), generating synthetic data rates[1]. The prefetching and on demand download simulations were performed over these data rates to determine their E_{max}.

Figure 7 to the left illustrates E_{max} as a function of σ^2_{AR}, for different $\sigma^2_{\epsilon}/\sigma^2_{AR}$. The $\sigma^2_{\epsilon}/\sigma^2_{AR}$ ratio determines a shape of data rates signal, representing the amount of serial correlation in time series data. Serial correlation determines how much information about the current data rate is contained in the previous

[1] c has been excluded from the parameter space, since it can be derived from (2). Hence, its impact on E_{max} is discussed with other model parameters in the text.

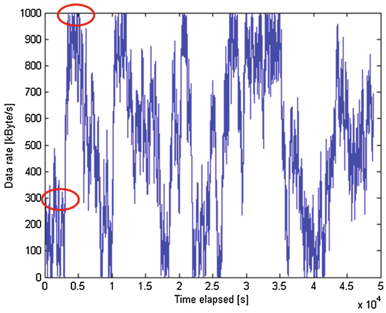

Fig. 8. Data rates generated with high correlation coefficient and little noise variance

value, which is repeated over various time periods. To preserve the same process, $\sigma_\epsilon^2/\sigma_{AR}^2$ was fixed in all experiments, while changing other parameter values.

It can be observed that E_{max} **decreases with an increase of** σ_{AR}^2 and σ_ϵ^2, given the fixed $\sigma_\epsilon^2/\sigma_{AR}^2$. This can be explained by the higher σ_{AR}^2 and σ_ϵ^2 values causing more frequent access to higher data rates, which leads to less difference in duration of on demand download and content prefetching.

Strong correlation and low noise variance of data rates caused by low $\sigma_\epsilon^2/\sigma_{AR}^2$ values result in **high** E_{max}. Such data rates exhibit a certain pattern for some time before they jump to significantly higher or significantly lower value, thus creating areas of longer staying periods at high and low data rates (depicted with red circles in Fig. 8), which in turn increases E_{max}.

Prefetching SLAs corresponding to E_{max} are plotted in Fig. 7 to the right. It can be seen that **prefetching SLA decreases with higher** σ_{AR}^2 to below one hour, except for $\sigma_\epsilon^2/\sigma_{AR}^2$=0.0004, where it *increases* with σ_{AR}^2 (of 500^2 and 600^2). This can be explained by the longer alternating periods of low and high data rates (created by high correlation and low noise variance) that increase the prefetching period. With further increase of σ_{AR}^2, the noise variance increases, leading to more frequent access to higher average data rates, thus decreasing the prefetching SLA.

Table 4 illustrates the impact of different constant (c) values on the average data rate ($\bar{R}$) and data rate range: the larger the c, the higher the $\bar{R}$. The data rate range also increases with higher c until the maximum data rate is reached, after which point the range starts decreasing if further increasing c.

Table 4. Range and average data rate for increasing c and ϕ_1=0.8885

μ	0	280.8	561.6	842.5
c	0	31.3	62.6	92.9
$\bar{R}$	123.3	290.7	561.5	802.7
range	0.5–587	0.5–840	25.3–998.7	268.2–999.9

As defined in (4), c can be derived from μ and ϕ_1. In Table 4, μ values are selected to yield different $\bar{R}$ in the 0.5–1000 kByte/s range, while ϕ_1 of 0.8885 was obtained by fitting a user data rates to AR(1) model with constant.

Figure 9 to the left illustrates that the **higher** c **results in lower** E_{max}. Prefetching SLAs corresponding to E_{max} are illustrated on the right side.

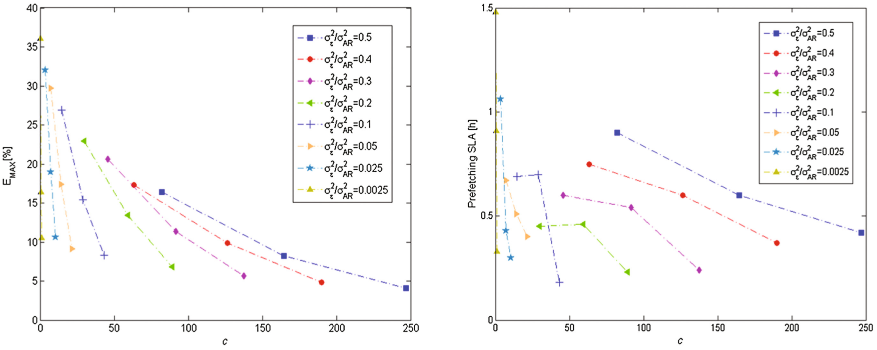

Fig. 9. E_{max} and prefetching SLA as a function of c for different $\sigma^2_\epsilon/\sigma^2_{AR}$, $\sigma^2_{AR} = 100^2$

6 Prefetching Recommendations

Figure 10 depicts $\hat{R}$ as a function of $\bar{R}$ and peak-to-mean ratio. This $\hat{R}$ is said to be *optimal*, since it was extracted from prefetching results with E_{max}. Observe that the **higher peak-to-mean** and **lower** $\bar{R}$ require **lower optimal** $\hat{R}$.

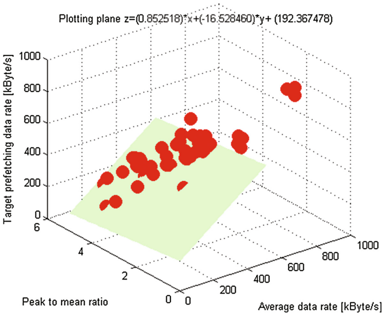

Fig. 10. Optimal $\hat{R}$ for $\bar{R}$ and peak-to-mean ratio

Fitting the optimal $\hat{R}$ to a plane with $\bar{R}$ and *peakToMean* variables yields:

$$\hat{R} = 0.853 * \bar{R} - 16.528 * peakToMean + 192.367 \tag{9}$$

with goodness-of-fit (R^2) being 0.8372 and root-mean-square error = 75.9483.

Figure 11 plots the original and estimated $\hat{R}$ as a function of $\bar{R}$ and *peakToMean*, computed from the entire model parameter space. Setting $\hat{R}$ to an optimal value can potentially maximize the energy cost reduction of a mobile device while prefetching video content using the proposed method.

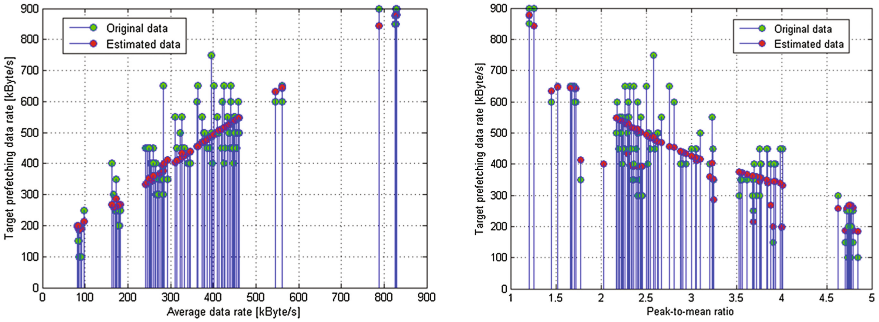

Fig. 11. Optimal $\hat{R}$ increases with higher $\bar{R}$ and peak-to-mean ratio

Table 5 predicts the optimal $\hat{R}$ for six different channel states ($\hat{R}_{est}$) that are extracted from users traces. Observe that $\hat{R}_{est}$ values of channels with strong correlation are similar to $\hat{R}$, while differing more for moderate correlation, due to a larger noise variance. Note that moderate correlation was observed in shorter data rate logs and where a mobile user behavior deviated from a daily routine (by visiting new locations with different data rate characteristics). However, the more precise (and longer) the log is, the closer $\hat{R}_{est}$ to optimal $\hat{R}$ are expected.

Table 5. Estimating optimal target prefetching data rates

ϕ_1	c	σ_ϵ^2	PeakToMean	$\bar{R}$	$\hat{R}$	$\hat{R}_{est}$
0.8899	59.5	11764.1	1.71	569.4	649.7	650
0.8885	31.3	4032.4	3.47	280.8	374.5	300
0.8704	15.6	2348.6	1.77	551	633.1	650
0.7878	98.6	23041.3	2.1	464.5	553.8	550
0.6545	217.2	22313.6	1.55	628.1	702.6	600
0.5407	75.1	1318.7	2.14	163.5	296.4	150

7 Discussion

The obtained results in this paper can be used in a real system, by monitoring a mobile user data rates and deriving the AR(1) model parameter values. The derived parameter values could be used by content providers to estimate the user's potential energy savings along with the time when the content will be available to the user for viewing. Additionally, using the fitted model parameter values, content providers can set the optimal prefetching parameters for the particular user in order to maximize their energy cost reduction or reduce the time to complete content prefetching. This flexibility of estimating the prefetching parameters to satify user preferences enables a content provider to define a new type of Service Level Agreement (SLA) that can guarantee a strict upper bound on content delivery delay to users, while allowing them to optimize their energy budget needed to complete mobile video content download.

A real system for mobile video prefetching has been implemented in our lab [4], which we plan to enhance with user preferences, model parameters learning, and estimation of the prefetching parameters that can produce the desired energy savings and delivery delay constraints. Such an enhanced system will be tested with real users and mobile devices.

8 Conclusion

This paper investigates the energy consumption reduction of content prefetching in different network conditions. A mobile device estimates the available downlink data rates by periodically probing the channel quality, prefetching the rest of content bits if the estimated data rate is equal to or higher than the set threshold.

The downlink data rates from actual user traces recorded in the mobile network were fitted to **autoregressive model of order one**. However, since AR coefficient was difficult to physically interpret, the following parameters were analytically derived: *process mean*, *process variance*, and *noise variance*. In order to generalize results concerning the potential maximum energy cost reduction (E_{max}) and the time needed to complete the prefetching (i.e., prefetching

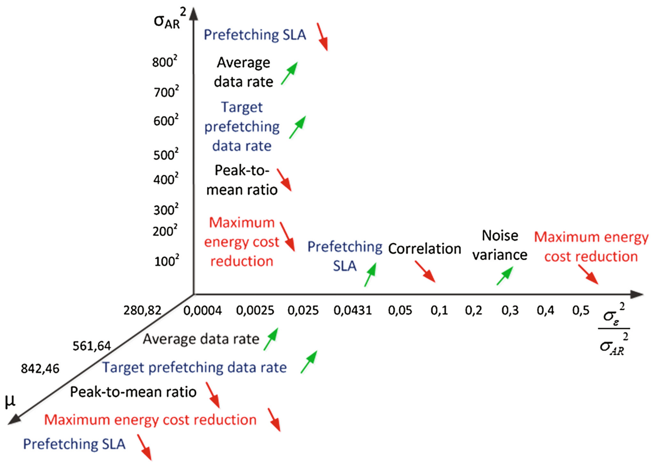

Fig. 12. Dependencies of prefetching results on model parameters

SLA), the entire model parameter space was used to generate synthetic data rates. Figure 12 illustrates conclusions of this evaluation, depicting how different prefetching metrics perform with the increasing parameter values.

The OTT prefetching provides high energy savings in the areas of alternating high and low data rates (indicated by **strong correlation between subsequent data rates and low noise variance**), since it stops downloading content as soon as it encounters the low available bandwidth, which would otherwise be performed on demand. Moreover, the **lower the average data rate and the higher the peak-to-mean ratio** (which can be achieved **by decreasing the process mean or process variance for the same** $\sigma_{\epsilon}^{2}/\sigma_{AR}^{2}$), the higher the energy savings. The $\sigma_{\epsilon}^{2}/\sigma_{AR}^{2}$ ratio determines the *shape* of data rates signal, representing a particular AR process type.

A **short prefetching SLA** is a result of frequent and long access to **high data rates**, which can be generated by **high process mean, high process variance, and/or low noise variance**.

Finally, based on identified dependencies of a target prefetching data rate on statistical properties of data rates (such as mean and peak-to-mean ratio), we proposed **recommendations on how to set** $\hat{R}$ in order to achieve E_{max}.

A deficiency of the OTT prefetching method is in periodic channel probing that estimates the available data rates, since the method prefetches some of the content bits at lower data rates, which decreases the potential energy savings. By employing longer sleep times, the method can faster avoid the areas with poor network conditions, thus increasing the likelihood of prefetching at high data rates. An additional knowledge about channel quality is, therefore, desired (such as signal strength, connectivity type, and cell IDs with the time when high data rates are usually available), either from historical user data or a mobile operator, in order to signal the method when there are good opportunities for prefetching.

Future work includes enhancing our method with this information and comparing the obtained energy savings with the results from this paper. Additionally, we plan to experiment with setting the prefetching parameters in the real system, optimizing the potential energy savings and prefetching delays according to user preferences and the fitted model parameters.

References

1. Balasubramanian, N., Balasubramanian, A., Venkataramani, A.: Energy consumption in mobile phones: A measurement study and implications for network applications. In: Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'09), Chicago, Illinois, USA, pp. 280–293, Nov 2009
2. Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, pp. 2011–2016. White paper
3. Devlic, A., Lungaro, P., Kamaraju, P., Segall, Z., Tollmar, K.: Energy consumption reduction via context-aware mobile video pre-fetching. In: IEEE International Symposium on Multimedia (ISM 2012), Irvine, California, pp. 261–265, Dec 2012

4. Kamaraju, P., Lungaro, P., Segall, Z.: A novel paradigm for context-aware content pre-fetching in mobile networks. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2013), Shangai, China, pp. 4534–4539, Apr 2013
5. Gautam, N., Petander, H., Noel, J.: A comparison of the cost and energy efficiency of prefetching and streaming of mobile video. In: Proceedings of the 5th ACM Workshop on Mobile Video (MoVid 2013), Oslo, Norway, Feb 2013
6. Walfield, N.H., Burns, R.: Smart phones need smarter applications. In: Workshop on Hot Topics in Operating Systems (HotOS 2011), Napa Valley, CA, pp. 1–5, May 2001
7. Rahmati, A., Zhong, L.: Context-based network estimation for energy-efficient ubiquitous wireless connectivity. IEEE Trans. Mobile Comput. **10**(1), 54–66 (2011)
8. Schulman, A., Navda, V., Ramjee, R., Spring, N., Deshpande, P., Grunewald, C., Jain, K., Padmanabhan, V.N.: Bartendr: A practical approach to energy-aware cellular data scheduling. In: ACM International Conference on Mobile Computing and Networking (MobiCom 2010), Chicago, Illinois, USA, pp. 85–96, Sept 2010
9. Stoica, P., Moses, R.: Spectral Analysis of Signals. Prentice Hall, Upper Saddle River, NJ (2005)