User-Aware Energy Efficient Streaming Strategy for Smartphone Based Video Playback Applications

Hao Shen and Qinru Qiu

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York, USA

{hshen01, qiqiu}@syr.edu

Abstract

We propose a methodology to design user-aware streaming strategies for energy efficient smartphone video playback applications (e.g. YouTube). Our goal is to manage the streaming process to minimize the sleep and wake penalty of cellular module and at the same time avoid the energy waste from excessive downloading. The problem is modeled as a stochastic inventory system, where the real length of video playback requested by the smartphone user is considered as demand that follows a stochastic process. Through user behavior analysis, a Gaussian Mixture Model (GMM) is constructed to predict the user demand in video playback, and then an energy efficient video downloading strategy will be determined progressively during the playback process. Experimental results show that compared to a static downloading strategy that is optimized by exhaustive trail, our method can reduce the wasted energy by 10 percent in average.

Key words: smartphone, video download, 3G, energy, Inventory Theory, Gaussian Mixture Model

1. Introduction

Battery life has continuously been one of the critical factors for smartphone user satisfaction. The increasing complexity of hardware and applications in the smartphones outpaces today's battery technology [1]. Although the percentage of power consumption of different components (e.g., processor, LCD, WiFi card, cellular interface) varies in different applications and different system settings, the cellular interface will soon become the most dominant energy consumer, which consumes more than 50 percent of total power consumption of the smartphone, when it is used as the network interface [2].

Video downloading and playback is one of the most common activities on the smartphone that has high energy consumption. When a user starts to watch a video, how long he (or she) is going to continue is unknown. Many traditional video players (application programs) try to download as many video data to the memory as possible during the time the user is watching in order to reduce glitches. Such buffering strategy may download more data than the user is going to watch if the user quits the video at an early time. It is shown that in Youtube, 60% videos are only being watched for less than 20% of their duration due to various reasons [4]. So a lot of downloaded data is useless as a result. On the other hand, to avoid excessive downloading, some video players buffer small amount of data periodically as a burst. Whether the next chunk of data will be downloaded is based on the user's watching progress. However, such downloading strategy will either keep the network interface active all the time or keep on switching it back to active from low power mode. In both cases, there will be energy wastes due to idle power consumption and the switching overhead of the network interface. Although such overhead is negligible for today's WiFi interface with the efficient implementation of power saving mode [3], it is still quite substantial for the cellular interface.

*This work is supported in part by NSF under grant CNS-0845947

Downloading just enough video that the user is going to watch in a burst is the most energy efficient strategy, however, it is not possible to accurately predict the real playback length as it is a random variable that has strong dependency on the user behavior and the video contents. The problem is very similar to a stochastic inventory system [9] in operation research where user demands follows stochastic distributions and the supply over-stocking and shortage are associated with costs.

We analyze video watching activities of different smartphone users and propose a stochastic model to capture the distribution of the real playback length. Based on the model the amount of data to be buffered is determined so that the expected energy waste on the cellular interface is minimized. As the video progresses, the estimated distribution of the remaining demands is dynamically updated and consequently affects the amount of data to be downloaded in the next burst. To the best of the authors' knowledge, this is the first work aiming at finding the best buffering strategy during video playback for energy saving in the network interface. The main technical contributions of this paper are summarized as the following: (1) A Gaussian Mixture Model (GMM) is proposed to capture the distribution of the video playback length for different users. (2) Based on the GMM generated, we apply stochastic inventory theory to find the best buffering strategy (i.e. how much video data should be buffered in each burst) for minimum energy wastes on the cellular network interface. (3) An Android application is developed to collect real video playback activities. The proposed model and buffering strategy are evaluated using the real data.

2. Background

2.1 Power consumption of cellular interface

The cellular interface is in general far less energy efficient than the WiFi interface. Its transmission energy per bit is usually much higher and varies with the signal strength [6]. It also has ramp energy and long tail energy state when no useful data is transferred [2][5], therefore putting cellular interface into lower power idle state and bringing it back when needed always associate with a non-negligible energy waste. Figure 1 shows the state machine of the 3G cellular interface [2][5]. As soon as the 3G interface receives a transmission request, it leaves the low power mode and enters the CELL_FACH (Forwarded Access Channel) state, which has very low throughput (less than 15kbps). If the amount of data to be sent or received is greater than the threshold, it will soon enter the CELL_DCH (Dedicated Channel) state, which has high throughput and energy consumption. When downloading is completed, the 3G interface will enter CELL_FACH state again after Inactivity Timer 2 has expired, and later switches to the low power IDLE state after the Inactivity Timer 1 has expired. Our measurement found that the Inactivity Timer 1 is about 4 seconds while the Inactivity Timer 2 is about 7 seconds. Please note that this information is vendor specific and it changes for different phones using different cellular networks ([5]).

2.2 Existing buffering strategy in YouTube

Figure 1. The radio resource state machine of 3G interface

Very little information about the downloading strategy used by YouTube can be found in published works. We tried our best to test it and found that it's quite different from phone to phone and potentially place to place. Typically the YouTube software always pre-buffers certain amount of video data ahead of user's current watching progress. If the signal strength is high and data can be transferred quickly, then the cellular interface will go to low power idle state when 'enough' data has been downloaded. Otherwise the cellular interface will keep buffering video data to keep up with the user watching progress. Our study shows that the amount of pre-buffering is a fixed value for a particular phone, although it varies over different phones. Based on the data provided by PCWorld, the average 3G data download speed is around 2 Mbps (Megabits per second) [10]. This is much faster than the data rate of a regular quality YouTube video, which is measured to be around 300 kbps. As we can see, if each pre-buffer process provides enough data to support the playback for a time that is longer than the count down period of Timer 1 and 2, it creates an opportunity for the cellular interface to go to low power state.

Figure 2 shows the measured power trace of a Nexus S smartphone with AT&T 3G network during the time YouTube is playing. The amount of packages received and transmitted is also shown in the figure. As we can see, the power consumption of the phone is periodically reduced to about 50% of its peak value when there is no transmission and receiving activities. This is because the 3G interface enters low power mode. We can also see that there is a visible delay from the time that transmission and receiving activity stops to the time that the power reduces. This timeout period is the overhead of turning off the 3G interface as discussed in Section 2.1.

The goal of this paper is to model different users' watching behavior (i.e., length of video playback) and then try to find the best strategy of streaming (i.e. the optimal size of pre-buffering) to minimize the energy wastes on cellular interface.

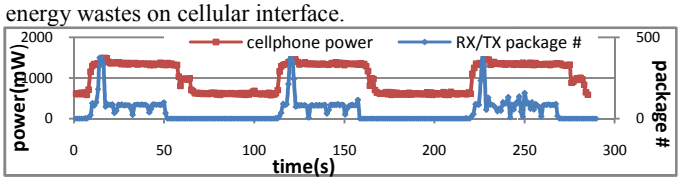

Figure 2. Power trace and network activities during YouTube playback

3. User Behavior Modeling and Downloading Strategy Optimization

3.1 GMM based playback duration modeling

The actual length of video playback is a random variable that is affected by user habits and the content of video clips. Although it cannot be accurately predicted in advance, its distribution can be learned through user behavior analysis of the past. Such model must be user specific, because different users watch videos clips with largely different average length [4]. Even for a single user, the length of video playback will have large variations. A user may have different video watch behavior (i.e. the watch time) toward different genres of clips. For example he or she may watch a music show for a very short time while spend longer time on a movie. There are different classes of scenarios of video watch (e.g., music show, movie and etc.); some of these scenarios probably even cannot be clearly identified. For each scenario, the video playback length is a random variable that has relatively small variance, while overall video playback length is a mixture of these random variables.

Based on the above discussion, we propose to model the distribution of playback length using a *Gaussian Mixture Model* (*GMM*) [8]. A GMM is a weighted sum of M component Gaussian densities given by the following equation: $p(x) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i)$, where x is a D-dimensional feature vector, w_i, i=1, …M are weights, and $g(x|\mu_i, \Sigma_i)$, i = 1, …M, are the component Gaussian densities with mean vector μ_i and variance vector Σ_i. In our problem, the feature vector has one element, that is, the length of playback. The values of M, μ_i and Σ_i are learned from training set, which is collected from past user behavior.

3.2 Energy efficient streaming strategy

3.2.1 Deciding pre-buffering size

In the next, we will present our energy efficient streaming strategy, which determines the amount of data to be pre-buffered at minimum energy waste. We model the streaming process as a stochastic inventory system [9]. Deciding how much data to be pre-buffered for a video player is analogous to finding how much inventory to replenish for a company. An inventory system has two cost components [9]: *holding cost* which is the cost associated with the storage of the inventory until it is consumed and *shortage cost* which is the cost when the amount of the commodity demanded exceeds the available stock. In our video buffering system, the *holding cost* is the energy wasted for downloading those excessive video data that are not used, while the *shortage cost* is the energy wasted to wake up a cellular interface in order to download more data when the user tries to watch beyond current buffering point. Please note that both holding cost and shortage cost are wasted energy. We do not consider the energy required to download the video data as cost, as long as these data are actually used.

We use D to denote the amount of video data that is actually watched by user (i.e. demand). D is a random variable. We use y to denote the decision on the size of pre-buffering (i.e. inventory). Both D and y are measured by the length of video playback time in seconds. The holding cost $h(D, y)$ is defined using Equation (1):

$$h(D,y) = \begin{cases} 0 & when\ y \le D \\ (y-D)e_h & when\ y > D \end{cases} \qquad (1)$$

,where constant parameter e_h gives the energy needed to download one second of video data. Based on the definition, if the amount of data pre-buffered is less than the demand, then there is no holding cost. Otherwise, holding cost will be the amount of energy needed to download the extra data.

If there is a shortage, the cellular interface will be turned on in the future to download more data. The more shortage we have, the more frequent the cellular interface will be turned on and hence more switching overhead. Therefore the shortage cost is defined as (2):

$$s(D,y) = \begin{cases} (D-y) * e_{sw}/\alpha & when\ y < D \\ 0 & when\ y \ge D \end{cases} \qquad (2)$$

, where constant parameter e_{sw} is the energy overhead to switch off and wake up the cellular interface. The parameter α represents the expected downloading size in the future. Therefore, $(D-y)/a$ gives the expected times that the cellular interface will be turned on.

Obviously, the real value of α is unknown at the time when the shortage cost is calculated. In our experiment, we learn this parameter from the analysis of the training set. We denote the ratio of e_{sw} and α as e_s, $e_s = e_{sw}/\alpha$. Based on the definition, if the pre-buffered data is more than the demand, then there is no shortage cost. Otherwise, the more shortage we have, the more frequently the cellular interface will be turned on to fill in the data, hence the shortage cost is proportional to the size of shortage.

The overall *cost* is the sum of holding cost and shortage cost given by Equation (3): $C(D,y) = h(D,y) + s(D,y)$ (3)

If we use $\varphi_D(\varepsilon)$ to represent the probability density function (PDF) of *D*, then the following equation gives the expected cost *C(y)*:

$C(y) = E[C(D,y)] = \int_0^y (y-D)e_h\varphi_D(\varepsilon)d\varepsilon + \int_y^\infty (D-y)e_s\varphi_D(\varepsilon)d\varepsilon$ (4)

The maximum and minimum value of *C(y)* can be found by solving the equation: $C'(y) = 0$ (5)

We can also prove that: $C''(y) = (e_h + e_s)\varphi_D(y) \geq 0$ (6)

Therefore, the minimum value exists.

Theorem 1 ([9]). The optimal size (y^0) of video data to be pre-buffered with minimum energy waste satisfies the following equation:

$\Phi(y^0) = e_s/(e_h + e_s)$ (7)

where Φ is the cumulative density function (CDF) of the playback length *D*. Because all CDF are monotonically increasing functions, the value of y^0 can be found easily using binary search.

3.2.2 Adaptive streaming strategy

As mentioned in Section 3.1, we use GMM model to capture the distribution of playback length. The model is learned from the usage history of the specific user, and its CDF is denoted as $G(D)$. *D* is an estimation of playback time without any prior knowledge. As the playback process goes on, we gather more information about *D*. For example, if the video has played for *A* seconds and it is still going on, we know that $D > A$. Therefore, the distribution model of *D* should be updated to reflect this information. As shown in Figure 3, let *g*(*D*) be the original PDF of the *D*. After knowing that *D* is greater than *A*, we can set the PDF to be 0 for any *D* less than *A*. In order for the remainder of the PDF to have an integration of 1, the area of the blue shaded part will be redistributed to the rest of the *g*(*D*), and we get an updated PDF $g'(D|D > A)$ as the following:

$$g'(D|D > A) = \begin{cases} 0 & if\ D < A \\ g(D)/(1 - G(A)) & if\ D \geq A \end{cases}$$

where *G*(*A*) is the CDF of *g*(*D*) at point A, in other words, the area of the blue shaded part in Figure 3. Note that *G*(*A*) is known after the model is given. The updated CDF $G'(D|D > A)$ can be calculated as:

$$G'(D|D > A) = \int_A^D g'(t|t > A)dt = [G(D) - G(A)]/[1 - G(A)]$$

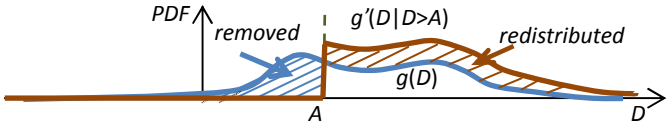

Figure 3 Adjusting the PDF based on obtained usage information

After that, the optimal size of data to be pre-buffered next is determined based on the updated CDF *G'*(*D*|*D*>A). This procedure will continue as the video watching activity goes on.

4. Experimental Results

4.1 User behavior model evaluation

We developed a lightweight Android application that monitors and collects the users' YouTube video playback activities. We take advantage of the Android 'logcat' tool, which shows the debug messages ejected from both operating system and applications. Our study shows that each time a new video in YouTube starts, a "WatchActivity" intent of YouTube will be written in the log buffer. Whenever the video stops playing because the user quits or pauses the video, an audio hardware sleep message is ejected. Similarly, whenever the video starts playing because the user starts a new video or resumes an old one, an audio hardware wakeup message is ejected. Combining the audio hardware message and "WatchActivity" information, we are able to detect those time when the video is paused and remove it from the playback length. Our application doesn't require root privilege and we installed it to five Android phones belonging to five different users. The application is running as a background service. The users are encouraged to watch YouTube videos as usual. The application records the history of their video playback length in a log file. After one and a half months of experiment, the log file is collected. In order to collect enough data, we recorded YouTube playback activities no matter the user uses WiFi or cellular access. We assume that the user behavior does not change significantly in both cases.

We use the first half of the collected traces as training samples and apply Matlab function "*gmdistribution.fit*" to learn the GMMs for each user. The function performs maximum likelihood estimation. We kept on increasing the number of Gaussian components until there is no significant improvement of the log-likelihood of the training sequence. For the purpose of comparison, we also model the training samples using normal distribution and exponential distribution. After constructing the model, we use the second half of the collected traces to test our model and calculate RLH (root likelihood) of each model.

Table 1. Comparison of user behavior models

Users	1	2	3	4	5
Means of Gaussian Components	{107,724, 2004}	{59,294, 2193}	{61, 1808}	{45, 361}	{43, 327}
RLH(GMM)	0.0009	0.0020	0.0039	0.0022	0.0061
RLH(Exp)	0.0009	0.0010	0.0016	0.0019	0.0049
RLH(Norm)	0.0003	0.0004	0.0004	0.0013	0.0022

Table 1 shows the comparison results. Each column represents different user. The second row gives the means of the Gaussian components in the GMM model for each user. As we can see, there are noticeable differences from user to user in terms of the distribution of their YouTube watching time. For users 1 and 2, their GMM models contain 3 Gaussian components; while for the other users there are only 2 Gaussian components. The mean of these Gaussian components also varies significantly. We believe that such difference is mainly due to the different watching interests of various users. The third, fourth and fifth rows in Table 1 give the RLH (root likelihood) of the testing sequence for GMM, exponential distribution and normal distribution models respectively. As we can see, the RLH of GMM model is 1.6 times and 4.4 times of the RLH for exponential and normal distribution models respectively. This indicates that the GMM is more suitable to describe the distribution of playback time.

4.2 Comparison of energy savings

In the second set of experiment, we compare inventory theory based streaming strategy with heuristic strategy, which buffers fixed amount of data each time. For the inventory theory based approach, different user behavior models, including GMM, exponential, and normal distribution models, are tested to show the impact of good user behavior modeling. In order to demonstrate the effectiveness of the inventory theory based streaming strategy without worrying about the accuracy of user behavioral model, we created a synthetic testing trace that follows ideal GMM distribution. It has three Gaussian components whose means are (50, 500, 2000).

As mentioned in Section 3.2, in this work, we focus on reducing the energy wastes due to either excessive downloading or frequent on-off switches. The power consumption on cellular interface varies from phone to phone and from time to time on the same phone because of different signal strength. To have a simple comparison, we use the amount of wasted video data to represent energy waste and it is measured by the length of playback time in seconds. Based on the definition of stocking cost, we know that e_h is equivalent to 1 second (of wasted video data). Using Monsoon power monitor[7] we measured the energy dissipation of a Nexus S smartphone with AT&T 3G network. Our data show that, with moderate wireless signal strength, the energy overhead to switch the cellular interface off (i.e. the energy dissipation in the timeout period) is approximately equal to the energy that is needed to download 16 seconds video data for YouTube. Therefore, e_{sw} in Equation (2) is set to 16.

Two static streaming strategies that have constant pre-buffering size are implemented. The first one buffers 100 seconds video data each time and is referred as Static(100s). We then sweep the pre-buffering size and test it using the training set of each user. The one that gives the minimum energy waste is selected and be applied to the testing sequence. We refer this approach as Static(best).

Figure 4 compares the wasted seconds (as the measurement of wasted energy) for different users and different buffering strategies. For the synthetic trace, the inventory theory based strategy reduces about 38% energy waste compared to the static strategies. We can also see that, the downloading strategy based on GMM model has 52% and 36% lower wasted energy that the strategies based on normal and exponential distribution models respectively. This indicates the importance of having accurate model.

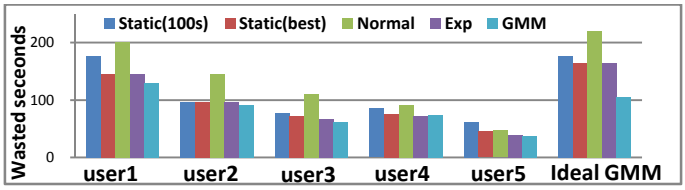

Figure 4 Comparison of wasted seconds for different users and different buffering strategies

However, for the testing traces from real users, the inventory theorem based strategy using GMM model has only 10% and 20% reduction of wasted energy in average compared to the Static(best) and Static(100s). And compared to the strategies based on normal and exponential distribution models, our model has 31% and 5% less energy waste in average. The relative efficiency of our strategy reduces for real trace partly because the GMM based user behavior model is not perfectly trained probably due to insufficient training data. On the other hand, exponential distribution is not a bad model to characterize the statistics of playback time, because there is much less probability that a user watches a long video than he/she watches a short video. Therefore, streaming strategy using exponential model also gives good energy reduction compared to others.

In order to show how the proposed streaming strategy works, we performed a detailed analysis for trace collected from user1. Figure 5 shows the buffering points determined based on the proposed streaming strategy. We can observe that the buffering points are generally around the means of the Gaussian components (107, 724, 2004). Furthermore, the points are sparse before 2000 and then become very dense after that. The streaming strategy algorithm tries to determine when the user will most likely to stop the video and it will set the buffer point to around that time. For example, after watching the video for 200 seconds, the user usually will not stop for another 300 seconds. So the next buffer point will be set to a little over 500 second and more than 300 seconds of video data will be downloaded to reduce the switching overhead. After the user watches for 2000 seconds, the video could end at anytime. Therefore, very few data will be downloaded each time to avoid waste.

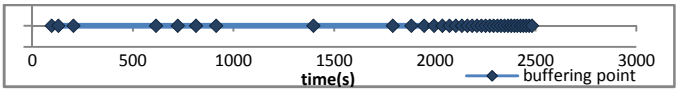

Figure 5 Buffering points of user 1 by the GMM approach

5. Conclusions

In this paper, we use GMM to model different user's YouTube playback time and then apply Inventory Theory to find out the optimal buffering points during the video playback that minimizes energy waste on the cellular interface. Our evaluation based on real field study and simulation demonstrates that our approach can save about 10% more energy than the best static buffering method.

6.References

[1] F.Ding, F.Xia, W.Zhang, X.Zhao and C.Ma, "Monitoring Energy Consumption of Smartphones," *iThings/CPSCom'11*, pp.610-613, Oct.2011.

[2] L.Zhang, B.Tiwana, Z.Qian, Z.Wang, R.P.Dick, Z.M.Mao and L.Yang, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," *CODES/ISSS '10*, pp.105-114, 2010.

[3] N.Ding, A.Pathak, D.Koutsonikolas, C.Shepard, Y.C.Hu and L.Zhong, "Realizing the Full Potential of PSM using Proxying," *INFOCOM 2012,* pp.2821-2825, Mar.2012.

[4] A.Finamore, M.Mellia, M.M.Munafo, R.Torres and S.G.Rao, "Youtube Everywhere: Impact of Device and Infrastructure Synergies on User Experience," *IMC'11,* pp.345-360, 2011.

[5] F.Qian, Z.Wang, A.Gerber, Z.M.Mao, S.Sen and O.Spatscheck, "Characterizing Radio Resource Allocation for 3G Networks," *IMC'10*, pp.137-150, 2010.

[6] A.Schulman, V.Navda, R.Ramjee, N.Spring, P.Deshpande, C.Grunewald, K.Jain and V.N.Padmanabhan, "Bartendr: A Practical Approach to Energy-Aware Cellular Date Scheduling," *MobiCom'10*, pp.85-96, 2010

[7] Monsoon Solutions Inc. http://www.msoon.com/LabEquipment/PowerMonitor/ .

[8] Mixture model. http://en.wikipedia.org/wiki/Mixture_model

[9] E. Porteus, "Foundations of stochastic inventory theory," *Stanford Business Books,* Aug,2002.

[10] M. Sullivan, "3G/4G Performance Map: Data Speeds for AT&T, Sprint, T-Mobile, and Verizon," *PCWorld*, 2012, *http://www.pcworld.com/article/254888/3g4g_performance_map_data_speeds_for_atandt_sprint_tmobile_and_verizon.html*