Energy Consumption of Mobile YouTube: Quantitative Measurement and Analysis

CONFERENCE PAPER · OCTOBER 2008

DOI: 10.1109/NGMAST.2008.26 · Source: IEEE Xplore

Yu Xiao
Aalto University

Antti Ylä-Jääski
Aalto University

Energy Consumption of Mobile YouTube:Quantitative Measurement and Analysis

Yu Xiao, Ramya Sri Kalyanaraman, Antti Yla-Jaaski
Helsinki University of Technology
Department of Computer Science and Engineering
FI-02015, Espoo, Finland
{*yuxiao,ramya*}*@cc.hut.fi, antti.yla-jaaski@tkk.fi*

Abstract

YouTube launched a customized website for mobile devices, which is gaining popularity among users. However, accessing multimedia based Internet services from mobile devices consumes energy, the most critical resource of handheld devices. The energy consumption of YouTube for mobile devices has not been widely studied. In this paper, we present a quantitative study of YouTube's energy consumption in mobile devices based on the measurements carried out on Nokia S60 mobile phones. We further investigate the energy characteristics based on network access technology (WCDMA and WLAN), download technology and storage media used. Our experimental results show that network transmission using WCDMA consumes more energy than when using WLAN. In addition, the power consumption is relevant with the status of network interfaces, and the impact on energy consumption based on the storage media used can be ignored.

1. Introduction

Online video is becoming one of the most popular Internet services, and more and more videos are streamed to mobile devices. According to the comScore's measurement report [4] as of April 2008, YouTube (www.youtube.com) is the largest online video sharing website. It has a customized site m.youtube.com for access from mobile devices. In this paper, we refer to the access to YouTube services from mobile devices as mobile YouTube.

It is essential to conserve energy when applications and services are used in resource constrained mobile devices. Mobile YouTube is an attractive Internet service that its usage is expected to increase in the near future. However, the large amount of network traffic and complex computation required for mobile YouTube can drain the battery in a short period. Therefore, there is a conflict in end-user demand and the services challenges to support the services with current technologies. Hence, energy consumption of mobile YouTube is an important study item.

Mobile YouTube supports online view, upload, rating, etc. It uses Adobe's Flash Video (FLV) files for video delivery from YouTube servers to clients. The standalone FLV file can be played with Adobe Flash Player, a web browser plug-in, or any other player that installs the FFDshow [6] filter. During the video delivery, mobile YouTube utilizes Adobe's progressive download technology. In contrast from traditional download-and-play, progressive download enables the beginning of playback once enough data has been downloaded instead of waiting until the download is completed. Progressive download is different from multimedia streaming because there is no resource reservation, and the whole file will be cached on clients instead of only parts of the file. For upload, users can upload their videos in a variety of formats other than FLV. YouTube server is responsible for the format conversion into FLV. During both download and upload, the files are transferred through HTTP over TCP.

In this paper, we present an energy consumption study of mobile YouTube based on energy measurements for different use cases. The use cases come from real world YouTube services, covering the features of network access technologies, download technologies, and storage media used. Subsequently, we compare the energy consumption using WCDMA and WLAN access technologies, while using progressive download, download-and-play and local playback respectively. Finally, based on the results, we summarize the energy characteristics of mobile YouTube.

The remainder of the paper is structured as follows. Section 2 reviews the related work in mobile YouTube and energy measurement. The use cases and experiment setup are described in Section 3. The experimental results are presented in Section 4. In Section 5, we discuss the

978-0-7695-3333-9 /08 $25.00 © 2008 IEEE
DOI 10.1109/NGMAST.2008.26

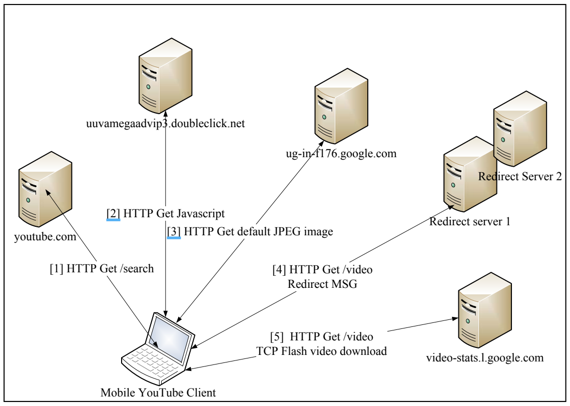

Figure 1. Establishment of TCP flash video download from YouTube server.

issues extended from the experiment results as well as the limitations of our measurements. Finally, Section 6 concludes the paper by discussing our future plans of research.

2. Related work

As a popular website with millions of users distributed around the world, YouTube's social and network traffic characteristics in large-scale networks have been widely discussed [2][7][12][18]. Likewise the network transport behaviors during progressive download have been discussed in [3] and [16]. However, the energy characteristics of YouTube or progressive download technology have not yet been presented.

According to [12] as well as our network traffic traces using *WireShark* (www.wireshark.org) in our campus network, the procedure of establishing TCP flash video download from YouTube server follows five steps (see Figure 1). Initially, the client sends a video search request to youtube.com, and gets javascripts from a second server. Later, the client gets default JPEG images of videos from ug-in-176.google.com. At this stage, the request for video delivery is redirected until it finds the media server (video-stats.l.google.com). Finally, the client sends HTTP Get request for flash video, and a TCP connection is established for video delivery. Since the YouTube servers contacted before establishing TCP connection may vary with network configurations, in our study, we focus on the energy consumed by video delivery which starts from the fifth step.

Previous studies of energy consumption focus on simulation-based analysis of low level system architecture and real-system measurements. In the simulation-based methods [1][15], energy consumption is the sum of energy consumed by internal modules which are estimated by gate-level or instruction-level simulators. Although these methods provide cycle-accuracy, the power measurements are time consuming and the results are hardware-specific. Real-system measurements are based on black-box methods, and can be applied to all kinds of mobile devices. They are mostly based on the precondition that energy is the product of power and duration, while power is the product of voltage and current [14]. Since the time interval is easy to calculate, the measurement of energy is often simplified to the measurement of voltage and current.

In order to measure the sample voltage and/or current during runtime in real-system measurements, a digital multimeter and a data acquisition board can be used. Later, the resultant data sets are processed with the help of data analysis software like *NI LabView* (www.ni.com) and *Agilent VEE* (www.agilent.com) [10][13]. In addition, power measurements can be performed by profiling software without external equipments, although such software is only available for certain mobile devices. In contrast, in [11], Nurminen et al. use profiling software - Nokia Energy Profiler [8] to measure voltage and current for Nokia S60 mobile phones during runtime. The Energy Profiler gives the total power consumption made by the mobile device. Apart from the power consumed by the peer-to-peer software, the Energy Profiler readings include the consumption made by the other inevitable components of the device. However, we believe their analysis is somehow limited since the influence of the profiler on the power consumption is not considered.

In our experiments, we use the same profiling software as [11], but we also take into account the power consumption of the profiling software and other daemon threads. Additionally, we consider the effect of battery temperature on power consumption.

3. Experiment setup

Our study is based on the energy measurement and analysis of different use cases. In this section, we describe our methodology for use case design, and test setup in terms of software, hardware, application and external influence.

3.1. Use case

Mobile YouTube includes two activities: network transmission and local processing. Network transmission can be video delivery from/to YouTube servers, while local processing includes local playback in FLV players and network

Table 1. Feature coverage of use cases.

	Use Case	Network	Operation	Storage
1	Online view via WCDMA	WCDMA	Progressive download	Cache
2	Online view via WLAN	WLAN	Progressive download	Cache
3	Download-and-play via WCDMA	WCDMA	Download-and-play	Phone memory
4	Download-and-play via WLAN	WLAN	Download-and-play	Phone memory
5	Playback from phone memory	Disabled	Local playback	Phone memory
6	Playback from flash drive	Disabled	Local playback	Flash drive
7	Replay	Disabled	Local playback	Cache
8	Upload via WCDMA	WCDMA	Upload	Phone memory
9	Upload via WLAN	WLAN	Upload	Phone memory

protocol processing during network download. Hence, an energy consumption study should cover individual energy consumption in each part as well as the interaction between them.

For network transmission, we measured energy consumption during both download and upload. In addition, we compared the energy consumption using different network access technologies. For local playback, we took different storage media into account. For local protocol processing and the interaction with network transmission, we considered the client application's different instantaneous activities, such as download, playback, or playback while download.

Considering the configurations of emerging mobile devices, our use cases cover two network access technologies (WCDMA and WLAN), four operations (progressive download, download-and-play, local playback and upload), and three kinds of storage media (cache, phone memory, external flash). We refer to orthogonal array testing strategy (OATS) [9] for the use case design since OATS helps to reduce the number of combinations and provide maximum coverage with a minimum number of test cases. As shown in Table 1, each column represents a testing variable, and each row represents a use case. The details of the use cases are described in the Appendix.

3.2. Test setup

Mobile YouTube client application. EmTube [5] is an open source YouTube browser and player for Symbian 60 mobile phones. It supports online browse, search and view of videos, download and save videos as local files, as well as playback from the local files. Especially, emTube provides three options for starting the playback: playback as soon as possible, playback after download finishes, and manual playback. In addition, it indicates the duration of download. YouTube for Mobile (YMobile) [17] is a YouTube client developed by Google. It supports browse, search, watch, rate, comment and upload as the computer supported website.

In our experiments, we used the Nokia N95 as testing device, which runs Symbian 60 3rd OS, and emTube for viewing videos online, downloading videos, and playing local FLV files, while YMobile for uploading videos to YouTube servers.

Data collection. To perform the power measurement, we use Nokia Energy Profiler for collecting battery current, voltage and battery temperature during runtime. We set 1 second as current sampling frequency and 10 seconds as voltage sampling frequency, since we observed a small standard deviation for current with 1 second sampling frequency, and an increase of 0.1W in power consumption when reducing the sampling interval to 0.25 second. The power consumption of energy profiler, Operating System and other daemon threads are defined as power consumption of default system, and is explained in Section 4.5. We take it into account while analyzing the energy consumed by mobile YouTube in Section 5.

Battery. Due to the chemical properties of battery, the battery temperature rises nonlinearly with the increase in continuous usage time, which further leads to the nonlinear rise of power. Hence, we use one battery for measurements to reduce the errors caused by the differences in the chemical properties among batteries. Further, we controlled the battery temperature in a range of 30.0 to 30.9$°C$ by repeating the experiments intermittently. The standard deviation of power in that temperature range is measured in Section 4.2.

Network access technology. The setting of network access point and the location of mobile terminals can possibly influence the signal strength, data rate, and further impact on the energy consumption. Hence, we maintain the same network settings and monitor the network conditions during the experiments. For example, our network settings are as below:

1. *WCDMA(3G)*: Internet subscription with both downlink and uplink bandwidth of 384kb/s.The average speed is 41kb/s with latency of 1.383 seconds.

2. *WLAN*: LINSYS WAP 200 wireless-g access point which is connected to campus Internet.

Table 2. Settings of WLAN access point.

Network mode	G-only	U-APSD	Disabled
Max data rate	54Mb/s	Beacon interval	100ms
Average RTT	92ms	Channel	11-2.462G
Downlink	5455kb/s	Uplink	1429kb/s

Video samples. According to [18], the average duration of YouTube video is 4.15 minutes with a median of 3.33 minutes. As test material, we selected Kaki King's music video - 'Pull Me Out Alive' (http://www.youtube.com/watch?v=pVYp2sgA9M0) for online view and download. It could be found in Featured selection during our testing period [Mar 14- 16, 2008]. The duration of the video is 238 seconds, and the file size is 9284 KB. For upload, we used '07032008002.mp4' with size of 3494KB. It was captured by Nokia N95 phone camera and uploaded to YouTube server on Mar 15, 2008.

External influence. User's behaviors during using YouTube services differ from each other. We formulate the user's behaviors with the help of use cases to reduce the error caused by different behaviors.

4. Experimental result

Based on the use cases described in Section 3.1 we performed the experiments to estimate the energy consumption of different mobile YouTube services. The data sets obtained from the experimentation are the raw data measured using the Energy Profiler.

4.1. Progressive download

In progressive download a three-phase process is followed. During the first phase, video is downloaded from YouTube server to mobile device. When the mobile device gets enough data into its cache, it starts playing and enters the second phase. In the second phase, both download and playback are in progress until the download completes. In the third phase, playback continues to the end. The three phases can be separated by the starting point of playback and the ending point of download.

Figure 2 shows the curves of average power consumption during progressive download via WCDMA and WLAN. The curves represent the execution of use cases 1 and 2 described in Section 3.1. From Figure 2 we can infer that for both the WCDMA and WLAN the curves can be divided into 3 segments which include a burst during the first several seconds and the horizontal traces. The burst is followed by the first horizontal trace and then the power consumption falls down to the lower level where it again follows a horizontal trace.

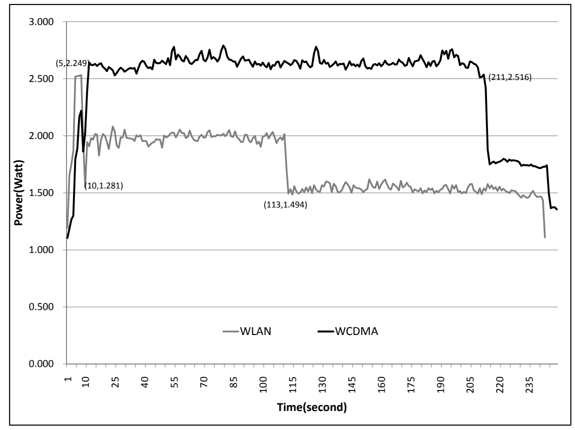

Figure 2. Power consumption during progressive download.

Table 3. Energy consumption during progressive download.

Connection	Phase	Duration(s)	Power(W)	Energy(J)
WCDMA	1	10	1.681	16.811
	2	201	2.639	530.523
	3	38	1.750	66.492
	Total	249	2.465	613.826
WLAN	1	4	1.616	6.463
	2	108	1.999	213.866
	3	131	1.530	201.894
	Total	243	1.738	422.244

In both the cases, the peak point of the bursts indicates the observed starting point of playback, and the ending point of the first horizontal trace represents the ending point of download. For example, we observed the starting point of playback and the ending point of download via WLAN as the 5th and 113th second, while in WLAN curve, the peak point is reached at the 5th second and the two horizontal traces intersect at the 113th second. Accordingly, the energy consumption during each phase can be accumulated separately. The statistics of energy consumption during each phase are presented in Table 3.

From Table 3 we infer that the total energy consumption during progressive download using WCDMA is around 1.45 times larger than using WLAN. In Phase 2 and 3, the power consumption of using WCDMA is greater than using WLAN, whereas we get the opposite result in Phase 1. The differences of power consumption during each phase can possibly be caused by network access technology used, network throughput, and processing capability of device. In addition, the energy consumed by download and playback could not be calculated respectively in this section, since

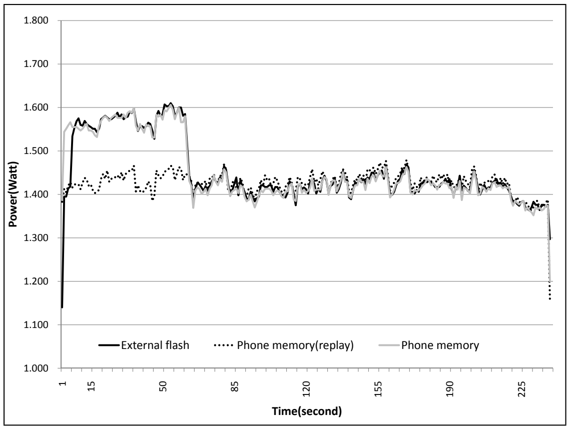

Figure 3. Power consumption of local playback.

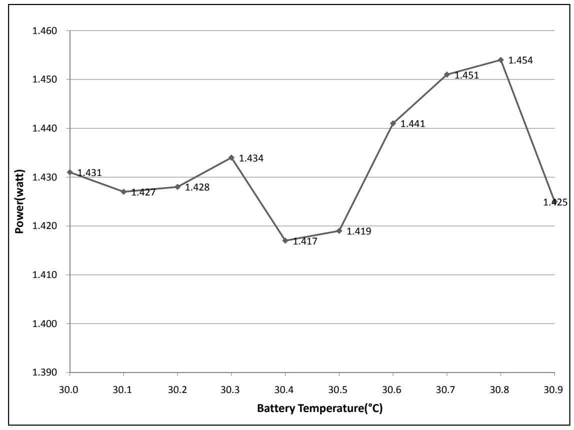

Figure 4. Power consumption during local playback at different battery temperatures.

Table 4. Power consumption during local playback.

Duration(s)	Phone memory(W)	External flash(W)	Cache(W)
61	1.565	1.555	1.431
177	1.412	1.416	1.424

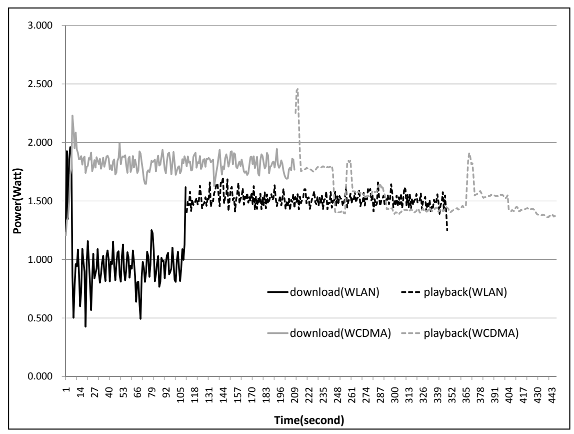

Figure 5. Power consumption of download-and-play.

they progress in parallel with each other during Phase 2. These questions will be explained in detail in Section 5.1.

4.2. Local playback

Figure 3 shows the power consumption during local playback from different storage media with the average battery temperature of $30.5°C$. The curves are the results of use case 5, 6 and 7 described in Section 3.1. For playback from phone memory and external flash, the power consumption during the first 61 seconds is higher than the rest period. It is caused by loading video from phone memory or external flash into cache. For replay, since the video has been loaded into cache during last playback, its power consumption is stable from the beginning. From the 62th second, the three curves overlap indicating the similar amount of energy consumption by all the three storage media used, namely phone memory, external flash and cache. The average power consumption of these storage media is listed in Table 4, and their average energy consumption is calculated as 281.078J, 281.136J and 275.128J respectively. The difference between cache and the others is mainly caused by loading video into cache.

The impact of battery temperature has non-linear effect on the power consumption of playback. Though, the variation in limited temperature range is small. As shown in Figure 4, the power consumption of playback from phone memory varies between 1.417W and 1.454W with standard deviation of 0.012W in the temperature range of 30.0 to 30.9 $°C$. Hence, the error of power consumption caused by battery temperature during playback is less than 0.037W.

4.3. Download-and-play

Download-and-play consists of two phases. The first phase involves downloading and storing the video in phone memory. In the second phase, the video is played from the phone memory immediately after the completion of first phase. We execute use case 3 and 4 to get the power consumption in both WCDMA and WLAN, and show the results in Figure 5. The statistical details are listed in Table 5.

Table 5. Energy consumption of download-and-play

Connection	Phase	Duration(s)	Power(W)	Energy(J)
WCDMA	1	209	1.815	379.426
	2	238	1.537	365.900
WLAN	1	110	0.971	106.833
	2	238	1.513	359.997

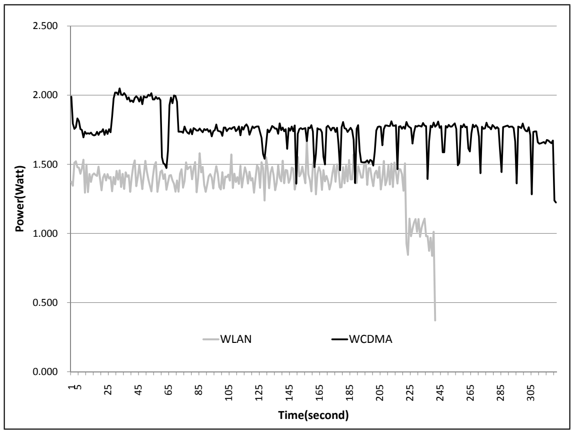

Figure 6. Power consumption during upload.

4.4. Upload

Figure 6 shows power consumption curves during video upload via WCDMA and WLAN. The curves represent the results of use case 8 and 9 in Section 3.1. Table 6 shows the comparison of energy consumption between WCDMA and WLAN during upload. Energy consumption in WCDMA is about 1.5 times more than that of WLAN. In addition, when compared with the download values in Table 3, upload costs more energy than download.

4.5. Default system

During our experimentation, the default system utilizes the basic components of the device. The settings of default system may vary with different use cases as described in Section 3.1. In general for all the use cases, we keep the device backlight on with maximum light sensor level, mute the system speakers, run Energy Profiler, open emTube, and close all the other applications.

Due to the different modes of network connections, we define five different states of the default system as shown in Table 7. WCDMA is disabled when the SIM card is removed from mobile phone, while WLAN is disabled by switching it off. WLAN is idle when it is switched on without any data transfer. WCDMA changes among three cell states: CELL_DCH, CELL_FACH and CELL_PCH. As shown in Figure 7, the average duration is 30 seconds for CELL_DCH, and 10 second for CELL_FACH in our test.

Table 6. Energy consumption during upload

Connection	Duration	Power(W)	Energy(J)
WCDMA	323	1.746	562.334
WLAN	242	1.382	334.482

Table 7. Power consumption of default states.

State	WCDMA	WLAN	Power(W)
1	Disabled	Disabled	0.270
2	Disabled	Idle	0.376
3	CELL_DCH	Disabled	0.742
4	CELL_FACH	Disabled	0.549
5	CELL_PCH	Disabled	0.282

5. Discussion

In this section, we will calculate the power consumption of network transmission and media playback during progressive download. In addition, we will analyze the impact from downlink traffic on the power consumption of download. Finally, we will figure out the limitations of our measurement.

5.1. Power consumption

Since playback is independent with network transmission, we consider the energy consumption of playback from cache to be equal to that of playback during progressive download. Let P1 be the average power consumption of state 2 of default system, P2 be the average power consumption during Phase 3 of progressive download, and P3, P4 be the power consumption of playback from cache during the first 61 seconds and the last 177 seconds respectively.

According to Section 4.5, during playback from cache, the power is consumed by playback and default system with state 1. Hence, P3 is 1.161W and P4 is 1.154W. Furthermore, the average energy consumption of playback from cache is 275.079J. According to Section 4.2, the power consumption of playback from cache during 62nd and 238th second can be considered as stable. As the Phase 3 of progressive download fit into this period, we assume that the power consumption of playback is stable during this period.

For progressive download via WLAN, since P2 equals to (P3+P1),the default system belongs to state 2 during Phase 3. Since the network interface is on during Phase 1 and 2, the default system is also in state 2 during the two phases.

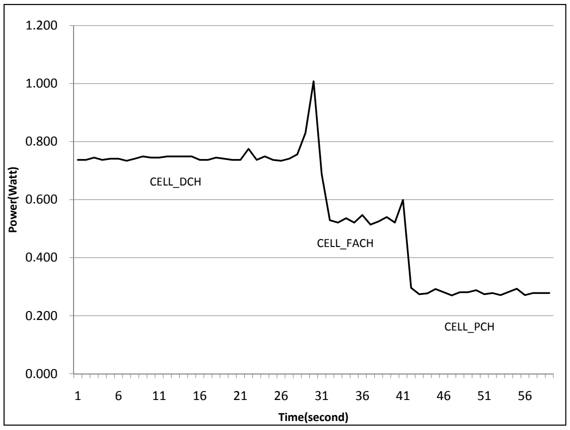

Figure 7. Power consumption of different states of WCDMA.

The power consumption of download in Phase 1 and playback in Phase 3 are calculated as given below:

$$Power(Phase1_{Download}) = Power(Phase1) - P1 \quad (1)$$
$$Power(Phase3_{Playback}) = Power(Phase3) - P1 \quad (2)$$

The energy consumption of playback during progressive download is the same as that of playback from cache, and the power consumption of playback happens during Phase 2 and 3. With the power consumption of playback during Phase 3, the energy and power consumption of playback in Phase 2 can be calculated as below.

$$Energy(Playback) = Energy(Playback_{Cache}) (3)$$
$$Energy(Playback) = Energy(Playback_{Phase2}) + Energy(Playback_{Phase3}) (4)$$
$$Power(Playback_{Phase2}) = Energy(Playback_{Phase2}) / Time(Phase2) (5)$$

During Phase 2, in addition to the default system, power is consumed by playback and download. As the energy consumption of playback and default system is known, the energy consumption of download can be estimated as follows.

$$Energy(Phase2) = Energy(Download_{Phase2}) + Energy(Playback_{Phase2}) + Energy(Default) \ (6)$$
$$Energy(Download_{Phase2}) = Energy(Phase2) - P1 * Time(Phase2) - Energy(Playback_{Phase2}) \ (7)$$

The average power consumption of playback, download during each phase is listed in Table 8.

For progressive download via WCDMA, we follow the same method for computing power consumption. The difference is that the default system belongs to different states. During Phase 1 and 2 of progressive download, the cell state is CELL_DCH due to the network transmission. During the first 30 seconds of the Phase 3, the cell state is CELL_DCH, and then move to CELL_FACH during the last 8 seconds. The statistics are listed in Table 9.

Table 8. Power consumption during progressive download via WLAN

Phase	Playback		Download	
	Power(W)	Energy(J)	Power(W)	Energy(J)
1	0	0	1.240	4.960
2	1.137	122.751	0.486	52.488
3	1.154	152.328	0	0
Total	1.156	275.079	0.513	57.448

Table 9. Power consumption during progressive download via WCDMA

Phase	Playback		Download	
	Power(W)	Energy(J)	Power(W)	Energy(J)
1	0	0	0.940	9.400
2	1.150	231.227	0.748	150.348
3	1.154	43.852	0	0
Total	1.151	275.079	0.757	159.748

For download-and-play via WCDMA, the playback starts when download finishes and lasts for 238 seconds. During playback, the state of cell changes from CELL_DCH to CELL_FACH and further to CELL_PCH which saves more power. This could be the reason for the lesser power consumption of playback in download-and-play than the playback in Phase 3 of progressive download.

5.2. Network traffic

Figure 8 describes the downlink traffic of video delivery from YouTube server during progressive download and download-and-play via WLAN. From the figure, we infer that, the duration of download in download-and-play equals to that of progressive download, and both the curves behave similarly. Each of them starts with a burst in the first several seconds, during which around 100KB is received from YouTube server. The downlink data rate is much lower after the burst and is relatively stable. For progressive download, the peak of the burst matches the observed starting point of playback in Figure 8. Hence, the data rate is not limited by local processing capability. It might have been caused due to the regulation of YouTube servers.

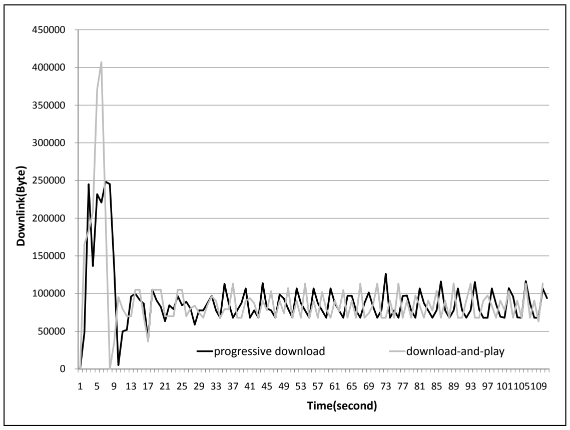

Figure 8. Downlink traffic during progressive download and download-and-play via WLAN.

5.3. Limitation of measurement

Limitations of our measurement are threefold. First, similar with other experiments with a live network, we could not control the network's behaviors such as the fluctuation of network bandwidth. Secondly, the temperature of mobile phone does not increase linearly. We use the average values from repeated measurements to get the results in certain range of temperature. Finally, it is difficult to find one client application with complete functions. For instance, YMobile does not offer the function of downloading videos to local storage, and emTube does not support upload. Despite the limitations, we believe that the results are relevant and useful for considering energy consumption of mobile YouTube.

6. Conclusion and future work

In this paper, we have presented a series of power measurements of mobile YouTube. The results show that WLAN consumes less energy than WCDMA during video download and upload. The cell states of WCDMA impact on the power consumption of the default system. For playback, the energy consumption of playback during progressive download equals to that of playback from cache. In addition, the energy consumption of playback from phone memory is on the same level as from external flash or cache, and the variation of power consumption with the temperature fluctuation in the scope of $1°C$ can be ignored.

The above results shed light on the energy-saving strategies. On the network side, network handover from WCDMA to WLAN during progressive download could be a possible option to save energy. Though, the overhead caused by handover should be taken into account. On the device side, cache optimization during playback is an alternative to reduce memory access and further save energy.

Appendix

	1
Name	Online view via WCDMA
Description	Energy consumption is measured as the user views the mobile YouTube video using WCDMA.
Precondition	Start and stop the Energy profiler as the user opens and exits the emTube. For emTube, set the 'start video' option as 'as soon as possible'.
Basic Flow	Open emTube, choose the network connection as 'WCDMA Internet', and browse the featured videos. Select the video by clicking, and exit emTube as the playback finishes.
	2
Name	Online View via WLAN
Description	Repeat Use Case 1 by choosing the WLAN connection.
	3
Name	Download-and-play via WCDMA.
Description	Energy consumption is measured as the user downloads a video from YouTube server using 'WCDMA Internet' and playback after download finishes.
Precondition	Repeat Use Case 1 by setting the 'start video' option as 'after download finishes'.
	4
Name	Download-and-play via WLAN.
Description	Repeat use case 3 by choosing WLAN connection.
	5
Name	Playback from phone memory
Description	Energy consumption is measured as the user watches the video from phone memory.
Precondition	Start and stop the Energy profiler as the user starts and completes watching the video.
Basic Flow	Open emTube, select the downloaded video file from the local directory of phone memory, play the video, and exit emTube as the video ends.
	6
Name	Playback from flash drive
Description	Repeat Use Case 5 by opening the video from the flash drive.
	7
Name	Replay
Description	Repeat Use Case 5, and as the playback completes, click replay to watch the video again.

Precondition	Use case 5
	8
Name	Upload via WCDMA
Description	Energy consumption is measured as the user uploads video from phone memory using WCDMA.
Precondition	Start and stop the Energy profiler as the user logs in and completes uploading the video.
Basic Flow	Open YMobile, choose the network connection as 'WCDMA Internet', login to the YouTube, and select the file to be uploaded from the phone memory. As the upload completes, exit from YMobile.
	9
Name	Upload via WLAN
Description	Repeat Use Case 8 by choosing the WLAN connection.

References

[1] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA*, pp. 83–94, 2000.

[2] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc.7th ACM SIGCOMM Conf. on Internet measurement*, pp. 1–14, New York, NY, USA, 2007. ACM.

[3] S. Chattopadhyay, L. Ramaswamy, and S. M. Bhandarkar. A framework for encoding and caching of video for quality adaptive progressive download. In *Proc. 15th Int. Conf. on Multimedia*, pp. 775–778, New York, NY, USA, 2007. ACM.

[4] comScore. More than 10 billion videos viewed online in the U.S. in February.http://www.comscore.com/press/release.asp?press=2190, Apr. 2008.

[5] emTube website:http://www.symbianfreak.com/news/007/12/emtube_freeware_flv_player_and_yt_client.htm, 2008.

[6] FFDshow codec website: http://www.freecodecs.com/download/ffdshow.htm, 2008.

[7] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube traffic characterization: a view from the edge. In *Proc. 7th ACM SIGCOMM Conf. on Internet measurement*, pp. 15–28, New York, NY, USA, 2007. ACM.

[8] Nokia Energy Profiler website: http://www.forum.nokia.com/main/resources/development_process/power_management/nokia_energy_profiler/, 2008.

[9] R. Krishnan, S. M. Krishna, and P. S. Nandhan. Combinatorial testing: learnings from our experience. *SIGSOFT Softw. Eng. Notes*, 32(3):1–8, 2007.

[10] S. Mohapatra, N. Dutt, A. Nicolau, and N. Venkatasubramanian. Dynamo: A cross-layer framework for end-to-end QoS and energy optimization in mobile handheld devices. *IEEE Journal on Selected Areas in Communications*, 25(4):722–737, 2007.

[11] J. Nurminen and J. Noyranen. Energy-consumption in mobile peer-to-peer - quantitative results from file sharing. *Consumer Communications and Networking Conference, 2008.*, pp. 729–733, 10-12 Jan. 2008.

[12] J. C. Paolillo. Structure and network in the YouTube core. In *Proc. 41st Ann. Hawaii Int. Conf. on System Sciences*, pp. 156, 2008.

[13] O. Riva. Contory: A middleware for the provisioning of context information on smart phones. In *Middleware*, pp. 219–239, 2006.

[14] V. Venkatachalam and M. Franz. Power reduction techniques for microprocessor systems. *ACM Comput. Surv.*, 37(3):195–237, 2005.

[15] N. Vijaykrishnan, M. T. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using SimplePower. In *ISCA*, pp. 95–106, 2000.

[16] Z. Yetgin and G. Seckin. Progressive download for 3G wireless multicasting. In *Proc. Future Generation Communication and Networking (FGCN 2007)*, pp. 289–295, 2007.

[17] YouTube for mobile website:http://www.youtube.com/yt_mobile_app, 2008.

[18] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch global, cache local: YouTube network traffic at a campus network: measurements and implications. In *Multimedia Computing and Networking 2008. Proc. SPIE, Vol. 6818, pp. 681805-681805-13 (2008).*, 2008.