2012 IEEE Fifth International Conference on Cloud Computing

Energy-as-a-Service (EaaS): On the Efficacy of Multimedia Cloud Computing to Save Smartphone Energy

Majid Altamimi, Rajesh Palit, Kshirasagar Naik
Department of Electrical and Computer Engineering
University of Waterloo, Canada
Email: {maltamim,rpalit,snaik}@uwaterloo.ca

Amiya Nayak
School of Information Technology and Engineering
University of Ottawa, Canada
Email: anayak@site.uottawa.ca

***Abstract*—In spite of the dramatic growth in the number of smartphones in recent years, the challenge of limited energy capacity of these devices has not been solved satisfactorily. However, in the era of cloud computing, the limitation on energy capacity can be eased off in an efficient way by offloading heavy tasks to the cloud. It is important for smartphone and cloud computing developers to have insights into the energy cost of smartphone applications before implementing the offloading techniques.**

In this paper, we evaluate the energy cost of multimedia applications on smartphones that are connected to Multimedia Cloud Computing (MCC). We have conducted an extensive set of experiments to measure the energy costs to investigate whether or not smartphones save energy by using MCC services. In other words, we investigate the feasibility of MCC to provide the Energy-as-a-Service (EaaS). Specifically, we compared the energy costs for uploading and downloading a video file to and from MCC with the energy costs of encoding the same video file on a smartphone. The aforementioned comparison was performed by using HTTP and FTP Internet protocols with 3G and WiFi network interfaces. All the experiments were conducted on an Android based HTC Nexus One smartphone. Our results show that MCC provides the smartphones with many multimedia functionalities and saves smartphone energy from 30% to 70%.

***Index Terms*—Energy Costs; Power Consumption; Smartphone; Handheld Device; Multimedia Application; Cloud Computing; Multimedia Cloud Computing; Mobile Cloud Computing.**

I. Introduction

Smartphones are becoming increasingly popular because of their capabilities and functionalities. Their small size and light weight make them very easy to carry, and they provide useful services as they run PC-like applications. In contrast to that, smartphones have some unique constraints, such as limited battery energy, processing, and memory capacity. In recent years, some of these constraints, such as memory and storage capacity, have been addressed to some extent. However, the advances in the semiconductor and telecommunication technologies are faster than that of the battery capacity. Therefore, energy constraint, which is result of limited capacity of the smartphone battery, has not been solved satisfactorily.

Smartphones are rich in communication interfaces, applications, and other resources such as sensors and Global Positioning System (GPS). Multimedia applications such as video playing and gaming are very much resource intensive in terms of processing and data transfer rates [1]. Consequently, they consume much energy and drain smartphone battery very quickly. In fact, those classes of applications are attracting much attention of smartphones users [2]. As they require more resources, smartphones quite often do not meet the expectations of users in performance and battery lifetime. For example, smartphones can play a narrow multimedia file format because of their limited processing and energy capacity. As a result, users require and demand more advances in smartphones to enhance their capability for multimedia applications.

Cloud Computing (CC) is a new computing paradigm that is promising in various aspects, such as virtually unlimited computing resources and availability. It provides data center resources such as processing, networking, and storage capabilities to the end user with required functionality. If CC provides a multimedia functionality, which includes storage, encoding, and play on-demand, then it is called Multimedia Cloud Computing (MCC) [3], [4], [5]. MCC can access any multimedia content on the Internet and supply it to a user in a desired file format when a user provides the targeted multimedia Universal Resource Locator (URL). A user can take advantage of the encoding capability of MCC by uploading a multimedia content in any file format, and then request the uploaded file in another file format. In addition, MCC is responsible to adapt the suitable multimedia encoding for an end-user by known the playing client.

In the era of CC, especially MCC, most of the smartphone constraints can be eased off by offloading heavy task from the smartphone to the CC. An example of heavy task is the video encoding where there is no existing of efficient encoding application on smartphones. In particular, a heavy energy consuming application is offloaded to the CC for smartphone energy saving. Thus, MCC appears to be promising to fill the gap between smartphone performance limitations and expectation of the users by the Energy-as-a-Service (EaaS) service. Actually, the MCC provides the multimedia services but at the smartphone user perspective the MCC provides what

978-0-7695-4755-8/12 $26.00 © 2012 IEEE
DOI 10.1109/CLOUD.2012.72

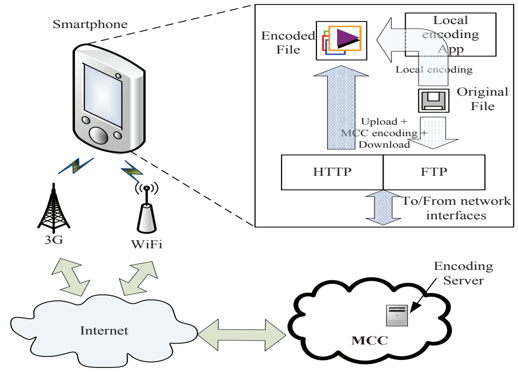
we call “EaaS”.

Since the offloading to MCC is in its nascent state, it is important to understand whether or not MCC extends the smartphone battery life by the EaaS. The aim of this paper is to address the problem of running multimedia application on smartphones, and investigate the benefit of using MCC framework in this regard. This study will confirm that MCC provides an effective solution (*i.e.,* EaaS) to extend smartphones battery life and enhance their multimedia capabilities. We present an extensive evaluation of the energy costs of smartphones and setup a large number (more than a hundred) of experiments on smartphones to measure their energy for running multimedia applications. Furthermore, we experimentally evaluate the energy cost on smartphones when the offloading technique is used. This evaluation has been conducted on a real MCC.

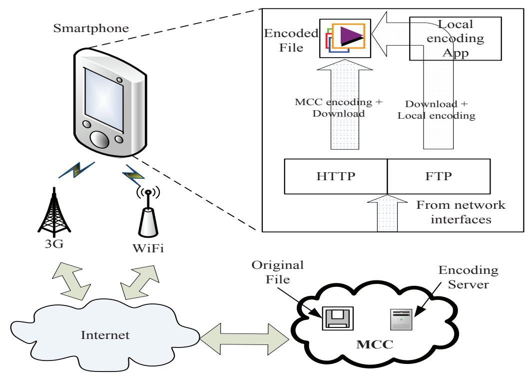
Figure 1 depicts two broad experimental scenarios related to the location of the original file to be encoded. In Fig. 1(a), the original file is available on the smartphone itself. On the other hand, the original file is available in the cloud as in Fig. 1(b). For uploading and downloading files to and from the cloud, we consider the energy implications of: *(i)* using the HTTP (Hypertext Transfer Protocol) and FTP (File Transfer Protocol) protocols at the application level; and *(ii)* using the 3G (Third generation mobile telecommunication) and WLAN (Wireless Local Area Networks) communications at the wireless access interface level. Using Fig. 1(a), we compare the energy cost of locally performing file encoding on a smartphone with the total energy cost of performing the same operation in the cloud, including the uploading and downloading communication costs. Similarly, using Fig. 1(b), we compare the energy cost of downloading an encoded file with the total energy cost of downloading the original file and performing encoding on a smartphone. Therefore, we perform experiments with eight scenarios for each of Fig. 1(a) and Fig. 1(b).

Our results give researchers much insight into the energy cost of such applications, which is important to implement offloading techniques. Indeed, the measurement of energy costs in this paper helps the developers of MCC to design efficient offloading that save energy of smartphones. The results show that MCC provides the smartphones with multimedia functionalities and saves smartphone energy from 30% to 70%. To our knowledge, this is the first study to evaluate energy costs of applications on smartphones connected to real MCC. Specifically, by the means of experiments, we show the following:

- We measured the energy costs for playing an online video and show the different phases of energy consumption, such as ‘download only,’ ‘download-and-play,’ and ‘play only’ (Section IV-B).
- As energy consumptions vary with time, we have measured the costs of sending and receiving file over HTTP and FTP protocols via 3G and WiFi interfaces, and present the statistics of the results (Fig. 5-8).
- We investigated whether or not smartphones save energy by using MCC EaaS service. This investigation is done by evaluating the energy costs for uploading and downloading a video file to and from MCC using HTTP and FTP protocols though 3G and WiFi connections. Then, we compare the results with the energy costs of doing video encoding on the smartphone for the same video (Fig. 11-14).
- In the aforementioned investigation, we consider two broad experimental scenarios (Fig. 1) related to the location of the original file to be encoded (Fig. 11-14).



(a) Encoding scenarios where the original file exists on the smartphone



(b) Encoding scenarios where the original file exists on the MCC

Fig. 1. Multimedia encoding scenarios

The rest of the paper is organized as follows: Section II summarizes the related work. Our system model and experimental setup are described in Section III. The details of our system evaluation and experiments are shown in Section IV. Section V summarizes our findings and the limitations of our experiments. This paper is concluded in Section VI.

II. Related Works

We focus on the approaches that involve task offloading to servers on the Internet. We categorize the approaches into three major classes: *(i)* using cloud computing [6], [7], [8]; *(ii)* using power-aware web proxy [6], [9]; and *(iii)* using local powerful servers [10], [11], [12], [13]. In this paper we adapt the CC approach for offloading multimedia applications.

Kelenyi et al. [6] proposed a strategy to save energy of handheld devices using the CC. In their strategy the cloud servers are used as a BitTorrent client to download torrent pieces on behalf of a mobile handheld device. While the cloud server downloading the torrent pieces, the mobile handheld device switch to sleep mode until the cloud finishes the torrent processes and upload the torrent file in one shot to the handheld device. This strategy saves energy of smartphones because downloading torrent pieces from torrent peers consumes more energy than downloading a one burst of pieces from the cloud. Kumar et al. [7] study the energy cost of offloading mobile computation to the cloud for saving energy on mobile devices. The cost of communication and computation is studied to show the impact of each of these factors on offloading cost. In [8], the energy cost has been studied for three cloud computing services: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). This study considers the total energy consumption on the user end, networking, and data center for all cloud services. A comparison is provided in this work between the system parts to show the impact of each part.

Xiao et al. [14] present a study of energy costs of using mobile YouTube (m.youtube.com) on mobile devices (Nokia S60) for 3G and WLAN network accesses. In this study, the authors consider two methods of downloading and playing video: progressive download and download-and-play. Energy cost data is collected by the Nokia Energy Profile application that itself runs on the mobile device to measure the current and voltage of the device battery. The analysis reveals that 3G consumes 1.45 times more energy than WLAN. Moreover, for each network access, the network traffic is the same for both progressive download and download-and-play. In general, the download-and-play consumes more energy than progressive download because the network modules continue to remain active for a while after the download is finished.

Evaluating of smartphone energy cost or power consumption is not new one; however, this topic is an area that has become increasingly important. This is reflected in current researches [15], [1]. In particular, these recent work, as well as our work, are examples of the work that have conducted experiments on an Android platform for many reasons as we describe in Section IV-A. However, to the best of our knowledge, there is no work on the evaluation of energy cost of offloading multimedia applications on smartphones and investigate the offloading to CC.

III. System Model and Experimental Setup

Our system consists of two major parts: smartphones and MCC where both are linked to the Internet, as depicted in Fig. 2. The smartphones are connected to the Internet through a WLAN (*i.e.* WiFi) or a cellular data access point (*i.e.* 3G-HSDPA). These smartphones provide all of multimedia functionalities to the end users. For instance, the user can play/recored a video or audio, and show/capture photos. The multimedia functionalities partially or fully interact with the corresponding MCC. On the other hand, the MCC is a special type of cloud computing where its data center provides the users with all needs of multimedia functionalities such as storage and processing. Moreover, the MCC has the capability to deal with a wide range of multimedia types and formats.

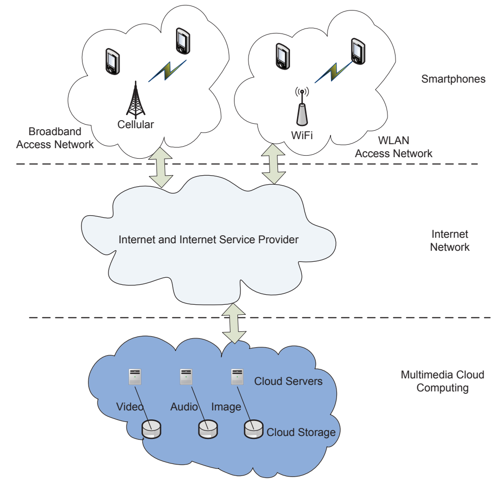


Fig. 2. System model.

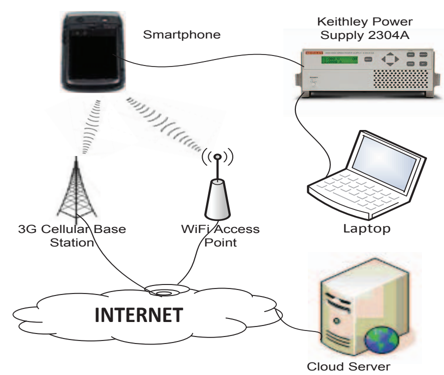


Fig. 3. Experiments setup.

Based on our system, we setup our experiments as shown in Fig. 3. The setup mainly includes: *(i)* smartphone (*i.e.* HTC Nexus One) that runs desired multimedia applications, store data, and upload and download via the Internet; *(ii)* access point for our lab WLAN and local Internet Service Provider (ISP) 3G; and *(iii)* high speed power supply (*i.e.* Keithley 2304A) working as a battery for the smartphone and recording the reading of voltage and current on a laptop.

IV. Evaluating The Energy Costs

In this section, we present the methodology of our experiments and the statistics of our experimental results. We conduct our experiments in two major parts: network related application and cloud experiments. The following subsections give the details of each of these parts.

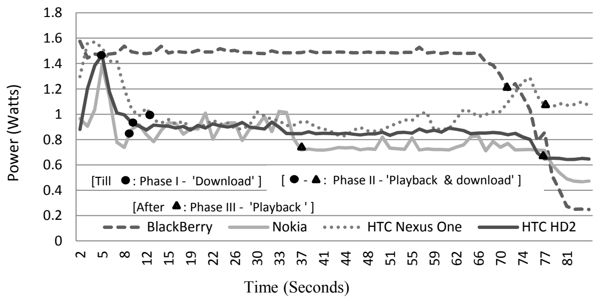


Fig. 4. Power consumption of multimedia application progressive download via WiFi for a set of smartphones.

A. Methodology

We did an extensive comparison of energy cost of applications on several smartphones [16], [17]. This comparison demonstrates that all smartphones are not comparable with respect to configurations, but they exhibit the same kind of energy cost behavior for each application. To illustrate this, Fig. 4, which is from our work in [16], shows the pattern of power consumption of some smartphones in downloading and playing multimedia file. We choose Android OS as smartphone platform since it has the biggest market share in the smartphone industry. We use Android based HTC Nexus One as it is popular, easy to access its battery contact pins, and full of multimedia and communication functionality. Hence, we use this smartphone in all our experiments for the consistency of our experiments.

For general smartphone battery usage, we study the power consumption instead of energy because the power gives a good insight the device consumption regardless of the file size or the time required to finish a task. However, the total energy is used to show a comparison between specific tasks as we see next. This is because the total energy is more meaningful metric to compare one particular task executed on two different processing rates. In addition, we measure the speed of the network interface to demonstrate the obtained data rate at the user level.

The smartphones access MCC via the Internet and the smartphone applications that are connected to the cloud are considered to be a Network Related Application (NRA). At the beginning of studying NRA, network interfaces (*i.e.* 3G and WiFi) are considered because each of these interfaces has its own characteristics, such as coding overhead. As a result, each network interface consumes unequal level of power and provides different data rates. Nevertheless, the Internet protocols (*i.e.* HTTP and FTP) should be taken into account for NRA. In other words, the network interfaces and the used protocols are the major factors that impact the energy costs of these applications. Thus, we examine the energy costs of common network interfaces and Internet protocols for the smartphones.

For NRA multimedia applications there are two scenarios in downloading the content from the MCC through the Internet: *(i)* progressive download, and *(ii)* download-and-play.

TABLE I
ENERGY CONSUMPTION ($\mu J/B$) OF SMARTPHONE

		3G	WiFi
Download	HTTP	11.3	4.92
	FTP	6.10	1.92
Upload	HTTP	14.56	2.20
	FTP	12.17	1.08

In progressive download, the process can be divided into three phases: *(a)* download from the server; *(b)* when enough data is in cache, application starts playback; and *(c)* playback continues after the download is complete. On the other hand, download and play consists of two parts: *(a)* download the entire file from the server to the smartphone storage; and *(b)* play the content form its local storage.

We know that a smartphone is a set of hardware and software. The hardware subsystems of a smartphone include CPU, RAM, HDD, sensors, and network interfaces; the software includes the operating system and applications. Energy is consumed in a smartphone by these hardware parts, which are managed by the operating system based on the needs of the applications. As a result, each application requires a specific amount of energy depending on the services that are required by each hardware part. In reality, these services are highly affected by the applications' settings and configurations. Moreover, the users interact with smartphones' applications in different ways. Therefore, the same application leads to different amounts of energy consumption. This is because of the large space of application settings, and configurations, and number of users interactions with the applications [18], [16]. For this reason, we provide our results at the user level and avoid breaking down the energy consumption for each hardware subsystem. Our results present the overall energy or power consumptions by the applications since our focus in this study is to compare the total energy consumed by an offloading process to MCC and the total energy consumed in smartphone to process a multimedia file.

B. Network Related Application Experiments and Results

We conducted experiments to measure the power consumption and the obtained data rate for uploading and downloading over HTTP and FTP protocols using the 3G and WiFi interfaces. The results are shown in Fig. 5-8. We use box plot to represent the statistical details of the result. The box plot represents from the bottom to the top the following: smallest observation, lower quartile ($Q1$), median ($Q2$), upper quartile ($Q3$), and the largest observation. From these figures, we observe that the FTP protocol supports higher data rate with less power consumption than the HTTP protocol. By this aspect, the FTP protocol is more efficient for task offloading.

From the energy saving point of view, the performance metric for network interfaces and protocols is the energy consumption per byte. Table I displays the summary from above figures the average energy consumption per byte obtained from our experiments.

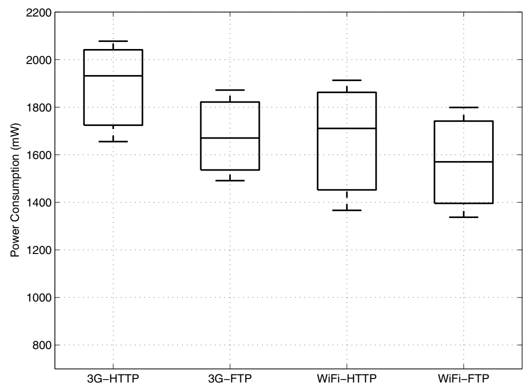


Fig. 5. Power consumption (upload) for the network interfaces and protocols

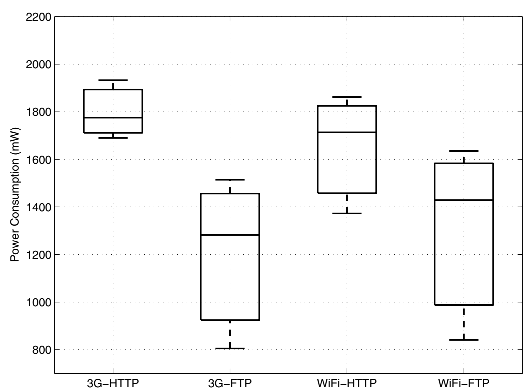


Fig. 6. Power consumption (download) for the network interfaces and protocols

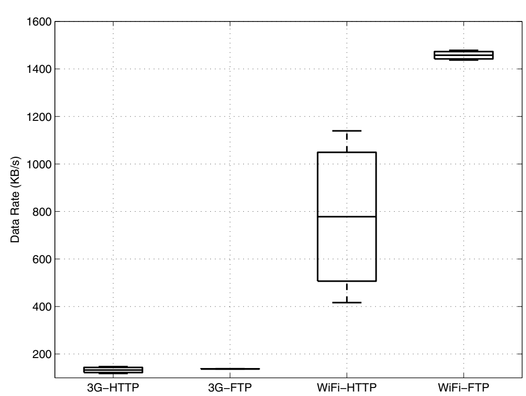


Fig. 7. Data rate (upload) for the network interfaces and protocols

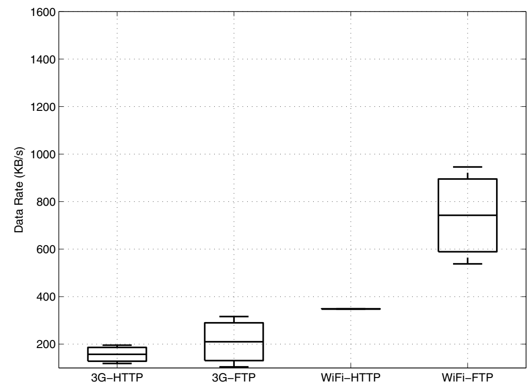


Fig. 8. Data rate (download) for the network interfaces and protocols

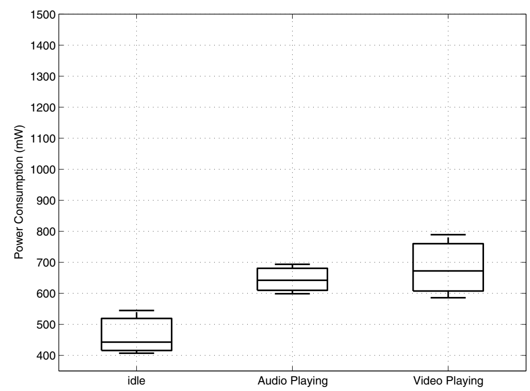


Fig. 9. Power consumption of multimedia application for playing content from storage.

We perform experiments for each of downloading scenarios, which are download-and-play and progressive download, that are directly have effect on the offloading technique. In download-and-play, we configure the client to download a file from the Internet and we measure the energy costs at the download time. After the download is completed, we allow the media application to play the downloaded file while we read the energy costs. In this scenario, the power consumption of playing the media is independent of the network interface. At playing stage, the power consumption of playing the media is shown in Fig. 9. We notice that the power consumption of this stage is similar for the playing when the file is on the device local storage. We show the idle case in Fig. 9 as a base for comparison purpose when the device is configured when the LCD is ON and the client application is OFF.

In the progressive download scenario, we configure the media client to play as there are enough data to start. After we measure the energy costs, the phases of this scenario can be easily distinguished as three phases. The first phase occurs when the client starts to download as much data as possible, which causes fluctuations in the power consumption as we see in Fig. 4. The second phase occurs when the client downloads and plays the media at the same time. Figure 10

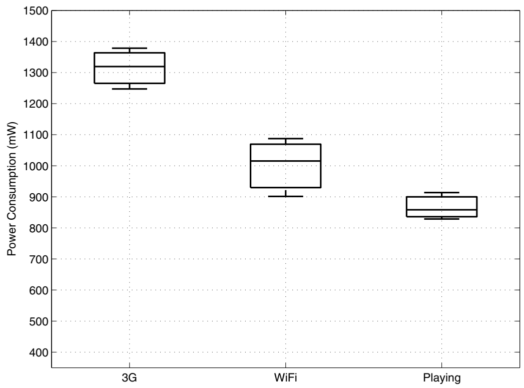


Fig. 10. Power consumption of progressive download of video file.

shows the power consumption of the second phase for the network interfaces. Finally, the last phase of this scenario is media playing only after the download is completed. The power consumption of this phase is shown in Fig. 10 as playing regardless of the interface that has been used. The third phase consumes more power than the playing from the storage shown in Fig. 9. The rais in the power consumption of the third phase may be due to the progressive download keeping the downloaded file in the cache memory of the device where this memory is a new power consumer element.

C. Cloud Experiments and Results

The MCC needs much evaluation with respect to the services that it provides to smartphones, in our case the EaaS service. In fact, saving smartphone energy is necessary for all of the MCC services [19], [20], [21]. To investigate whether or not smartphones save energy by using EaaS service of the MCC, we conduct further experiments on a real MCC. As we discussed in the introduction, video encoding requires heavy processing that drain smartphone battery if it is done on the smartphone processor. Besides, the data exchange with the MCC over the Internet consumes energy for a smartphone to do the encoding on the MCC. In these experiments, we investigate and measure the energy costs of using the MCC on HTC Nexus One smartphone. These experiments are conducted on the encoding service of the encoding cloud computing (*i.e., www.encoding.com*).

This cloud provides the smartphones with encoding functionality to a wide range of multimedia formats to convert from one format to another. The smartphones can provide the multimedia file to the cloud in many ways: *(i)* FTP or HTTP file link, where the file exists on a web server; *(ii)* Cloud file access link, where the file exists on a cloud; and *(iii)* direct upload, where the file exists on the user device. Once the file is uploaded, desired output format is specified and the conversion request will start. Finally, the MCC renders the converted file to the smartphone in many ways: *(i)* upload to FTP to HTTP server; *(ii)* cloud storage; and *(iii)* email link.

Most of the smartphones can play a narrow range of multimedia. For example, the HTC Nexus One can recognize and play mp4, H.263, and H.264 video formats. If the user of this device wants to watch Flash Video (flv) video, he needs to convert it to another format supported by the same device. In particular, the flv format is very common to website hosting but no client, except a Youtube client, can play it on smartphones. The same difficulty applies to other video formats as well.

TABLE II
PROPERTIES OF THE VIDEO FILES

Parameters	Original flv	Converted mp4 at settings: Default	Converted mp4 at settings: Customized
Length (minutes:seconds)	3:31	3:31	3:31
File size (MB)	23.97	8.21	131.5
Frame width (pixel)	640	320	640
Frame height (pixel)	360	240	360
Data rate (Kbps)	909	258	5057
Frame rate (frame/s)	24	24	24
Audio bit rate (Kbps)	114	63	172
Channel	2	2	2
Audio sample rate (KHz)	44.1	44	44

In the experimental setup, we consider all encoding scenarios for a multimedia file as depicted in Fig. 1. There are four scenarios related to the location of the original file to be encoded as follows.

1) The first scenario is the local encoding using the smartphone's encoding application (*i.e.* Local encoding in Fig. 1(a)).
2) The second scenario is the offloading technique by uploading the original multimedia file and doing the encoding by the MCC then downloading the encoded multimedia file (*i.e.* Upload + MCC encoding + Download in Fig. 1(a)).
3) The third scenario is the local encoding using the smartphone's encoding application but after download the original file from the MCC (*i.e.* Download + Local encoding arrow Fig. 1(b)).
4) The fourth scenario is the encoding using MCC after load the original file from cloud storage then downloading the encoded multimedia file (*i.e.* MCC encoding + Download arrow Fig. 1(b)).

In this subsection, we investigate the energy costs for each of these four above scenarios. We choose a song video in flv format and convert it into mp4 video format. Table II lists the original and encoded video parameters. It is obvious that each individual file format has unique performance parameters such as the data bit rate. In both of MCC and smartphone encoding application, the encoding parameters are kept to default settings. Unfortunately, the settings are different as shown in Table II. The default and customized parameters are the parameters of encoded by MCC in default and customized setting, respectively. For fair comparison, we customized the encoding parameter for the MCC that match the parameters of the smartphone encoding application. Thanks to MCC that gives us the opportunity to configure the conversion parameters since the smartphone application does not.

We compare the energy costs for all of four scenarios as

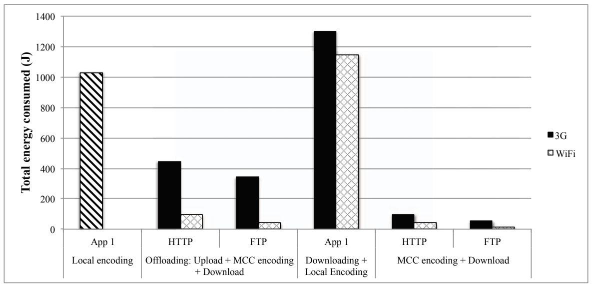


Fig. 11. Total energy consumed for encoding a video file where the settings is **default** on both of the MCC and the smartphone.

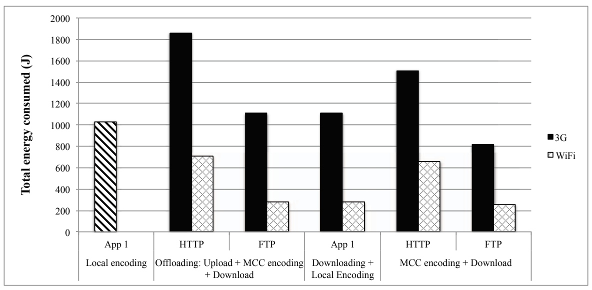


Fig. 12. Total energy consumed for encoding a video file where the settings is **customized** in the MCC and **default** in the smartphone.

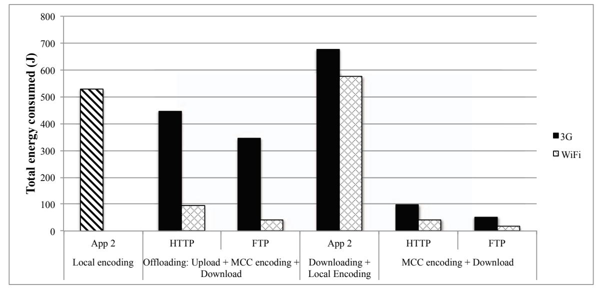


Fig. 13. Total energy consumed for encoding a video file where the settings is **default** in the MCC and **customized** in the smartphone.

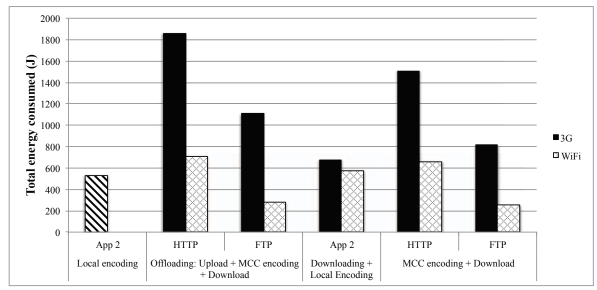


Fig. 14. Total energy consumed for encoding a video file where the settings is **customized** in both of the MCC and the smartphone.

shown in Fig. 11. We present here the total energy costs for different network interfaces (*i.e.* 3G and WiFi) and Internet protocols (*i.e.* HTTP and FTP). In these results, if a network interface and an Internet protocol are used for uploading a file to MCC, the same interface and protocol is used for download the converted file. Therefore, there is no permutation between network interfaces and Internet protocols in uploading and downloading are presented here. This is because we belief that the users tend to use the same configurations at a time. Figure 11 reveals the capability of MCC to provide EaaS service for smartphones. MCC provides at least 60% reduction in energy if the 3G interface with HTTP protocol is used. In addition, this figure shows the effect of different network interfaces and Internet protocols as we expect. The WiFi interface consumes less energy that the 3G interface for transmitting the same amount of data. Similarly, the FTP protocol consumes less energy than the HTTP protocol for transmitting the same amount of data.

Figure 12 shows the comparison when the MCC settings is identical as smartphone application settings. This figure shows the impact of the file size in the offloading process since the customized settings required larger file size as show in Table II. In this case, the offloading is not an option if the 3G interface is the only available interface. From this figure, we can argue why we examine each of the network interfaces and the Internet protocols. It is worth mentioning that the customized configuration enlarge the encoded file without any noticeable improvement in the Quality of Experience (QoE).

On the other hand, for the local encoding application, we test two converting application versions which are PC-like applications, namely APP1, on smartphone (*i.e.* MP3 Media Converter, version 1.1.0.3, by Ansha Team) and optimized version, namely APP2, for the smartphone (*i.e.* MP3 Media Converter (Neon), version 1.1.0.2, by Ansha Team). The optimized version exploits the multimedia acceleration capability of the smartphone that is called Single Instruction Multiple Data (SIMD). All of these applications are available on the 'Market' of Android. Similarly, Fig. 13 and 14 compare the total energy consumed in customized application and offloading to MCC with default and customized settings, respectively.

V. Discussions and Limitations of our approach

Our experiments shows that the NRA performance highly depends on the network interfaces and protocols. In general, it is shown that 3G interface consumes more power than the WiFi interface. This means offloading via 3G has to be done carefully. Moreover, the FTP protocol consumes less power than the HTTP protocol. Recommending a single interface is difficult because each one provides the end user a unique experience. For instance, 3G supports a large range communication while WiFi supports short range. However, choosing interface decision depends on the NRA requirements such as having a specific data rate.

In several readings, there is a large fluctuation of the energy consumption that could be caused by the cloud server at the cloud end or the network condition. This also depends on the contents of the media as it has been proven in [22]. For example, the server provides a high data rate at the beginning of video progressive download to guarantee a certain level of quality of experience. In contrast, the server uses other procedure for the audio files. We exclude the dependency of the energy cost on the cloud and the network conditions.

In our comparison experiments, we have been limited to the local applications that have fixed conversion parameters and produce a large file. For fair comparison, we customize the conversion parameters on the MCC to match the conversion parameters of the local applications. Therefore, the energy cost of downloading the converted file is quite high. We should mention that the MCC can convert a typical video file on its size, format, and quality for Android smartphone. Our experiments reveal that the MCC auto conversion parameters would reduce the downloading energy costs for the smartphone from 90% to 95% where the custom configuration could save from 30% to 70%.

In all of our experiments we use the Internet protocol IPv4 that does not distinguish the information contents. If the smartphones support IPv6, the experience is likely to be different because this protocol offers Quality of Service (QoS) and multimedia priority procedure and could offer efficient energy solution and offloading techniques [23].

VI. Conclusions

Our study clearly indicates that offloading heavy applications, namely multimedia applications, from smartphones to MCC is beneficial. MCC significantly reduces the energy consumption on smartphones by the EaaS service. Moreover, MCC enriches smartphones capabilities for multimedia applications.

At this time when the CC is in its infant state, the importance of evaluating the benefit of MCC to overcome smartphone constraints motivates us to conduct this study. A large number of experiments has been performed for common network interfaces (3G and WiFi) and protocols (HTTP and FTP). The location of the original file has been considered. This paper provides a wide range of comparison between possible encoding location, original file place, encoding configuration parameters, network interfaces, and Internet protocols. The results reveal the potential of MCC by reducing smartphones energy consumptions on multimedia applications at least 30%.

We would conduct more experiments on other multimedia types such as audio and images to generalize our finding. This study opens new opportunities to be investigated. Optimum algorithms, architectures, and implementations for this offloading technique is needed to reach best offloading case. Finally, modeling the MCC to handle the offloading is important to implement efficient offloading.

References

[1] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, 2010, pp. 21–34.

[2] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A First Look at Traffic on Smartphones," in *Proceedings of the 10th annual conference on Internet measurement*, 2010, pp. 281–287.

[3] M. Altamimi and K. Naik, "The Concept of a Mobile Cloud Computing to Reduce Energy Cost of Smartphones and ICT Systems," in *Proceedings of the First international conference on Information and communication on technology for the fight against global warming*, ser. ICT-GLOW'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 79–86.

[4] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia Cloud Computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.

[5] W. Song and X. Su, "Review of Mobile cloud Computing," in *Proc. IEEE 3rd Int Communication Software and Networks (ICCSN) Conf*, 2011, pp. 1–4.

[6] I. Kelenyi and J. K. Nurminen, "CloudTorrent - Energy-Efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services," in *Proc. 7th IEEE Consumer Communications and Networking Conf. (CCNC)*, 2010, pp. 1–2.

[7] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[8] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, January 2011.

[9] K. Naik, "A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices," Dept. of ECE, University of Waterloo, Waterloo, ON, Canada, Tech. Rep. 2010-13, 2010.

[10] K. Yang, S. Ou, and H.-H. Chen, "On Effective Offloading Services for Resource-Constrained Mobile Devices Running Heavier Mobile Internet Applications," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 56–63, 2008.

[11] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using Bandwidth Data to Make Computation Offloading Decisions," in *Proc. IEEE Int. Symp. Parallel and Distributed Processing*, 2008, pp. 1–8.

[12] X. Zhao, P. Tao, S. Yang, and F. Kong, "Computation Offloading for H.264 Video Encoder on Mobile Devices," in *Proc. IMACS Multiconference Computational Engineering in Systems Applications*, 2006, pp. 1426–1430.

[13] G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli, "Studying Energy Trade Offs in Offloading Computation/Compilation in Java-Enabled Mobile Devices," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 9, pp. 795–809, September 2004.

[14] Y. Xiao, R. S. Kalyanaraman, and A. Yla-Jaaski, "Energy Consumption of Mobile YouTube: Quantitative Measurement and Analysis," in *Proc. Second Int. Conf. Next Generation Mobile Applications, Services and Technologies NGMAST '08*, 2008, pp. 61–69.

[15] N. Vallina-Rodriguez, P. Hui, J. Crowcroft, and A. Rice, "Exhausting Battery Statistics: Understanding the energy demands on mobile handsets," in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, ser. MobiHeld '10. ACM, 2010, pp. 9–14.

[16] R. Palit, R. Arya, K. Naik, and A. Singh, "Selection and Execution of User Level Test Cases for Energy Cost Evaluation of Smartphones," in *Proceeding of the 6th international workshop on Automation of software test*, 2011, pp. 84–90.

[17] R. Arya, R. Palit, and K. Naik, "A Methodology for Selecting Experiments to Measure Energy Costs in Smartphones," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, july 2011, pp. 2087 –2092.

[18] A. Manjunatha, A. Ranabahu, A. Sheth, and K. Thirunarayan, "Power of Clouds in Your Pocket: An Efficient Approach for Cloud Mobile Hybrid Application Development," in *Proc. IEEE Second Int Cloud Computing Technology and Science (CloudCom) Conf*, 2010, pp. 496–503.

[19] W. Itani, A. Chehab, and A. Kayssi, "Energy-Efficient Platform-as-a-Service Security Provisioning in the Cloud," in *Proc. Int Energy Aware Computing (ICEAC) Conf*, 2011, pp. 1–6.

[20] S. Ferretti, V. Ghini, F. Panzieri, and E. Turrini, "Seamless Support of Multimedia Distributed Applications Through a Cloud," in *Proc. IEEE 3rd Int Cloud Computing (CLOUD) Conf*, 2010, pp. 548–549.

[21] M. Rodriguez-Martinez, J. Seguel, M. Sotomayor, J. P. Aleman, J. Rivera, and M. Greer, "Open911: Experiences with the Mobile Plus Cloud Paradigm," in *Proc. IEEE Int Cloud Computing (CLOUD) Conf*, 2011, pp. 606–613.

[22] J. F. M. Bernal, L. Ardito, M. Morisio, and P. Falcarin, "Towards an Efficient Context-Aware System: Problems and Suggestions to Reduce Energy Consumption in Mobile Devices," in *Proc. Ninth Int Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR) Conf*, 2010, pp. 510–514.

[23] H. Yuan, C.-C. J. Kuo, and I. Ahmad, "Energy Efficiency in Data Centers and Cloud-Based Multimedia Services: An Overview and Future Directions," in *Proc. Int. Green Computing Conf*, 2010, pp. 375–382.