An Analysis of Power Consumption in Mobile Cloud Computing

Abdelmounaam Rezgui[1](✉) and Zaki Malik[2]

[1] Department of Computer Science and Engineering, New Mexico Tech, Socorro, NM, USA
rezgui@cs.nmt.edu
http://www.cs.nmt.edu/∼rezgui
[2] Department of Computer Science, Wayne State University, Detroit, MI, USA
zaki@wayne.edu
http://www.cs.wayne.edu/∼zaki

Abstract. With the rapid proliferation of mobile devices, mobile cloud computing is emerging as an increasingly omnipresent paradigm enabling users to use battery-powered mobile devices to access a wide range of compute-intensive applications hosted on the clouds. Often, the assumption is that mobile devices consume less power when they access an application run on the cloud than when the application is run on the device itself. This, however, is increasingly questionable with the significant recent progress in improving power efficiency of mobile devices (e.g., using ultra low power GPUs). This paper aims at analyzing and comparing the benefits of these two alternatives using *mobile cloud gaming* as an example. Our evaluation shows that, despite the recent advances towards reducing power consumption in mobile devices, mobile cloud computing remains the best of the two alternatives in a wide range of scenarios.

Keywords: Mobile cloud gaming · GPUs · NICs · Power consumption · Visualization as a Service (VaaS) · Offloading

1 Introduction

Mobile devices (mobile phones, tablets, and ultra mobile PCs) are driving a phenomenal market shift. A Gartner report (Table 1) predicts that, by 2017, device shipments will reach more than 2.9 billion units, out of which 90 % will be mobile devices [18]. The growth is particularly strong in mobile phones. A June 2015 Ericsson report indicates that the total number of mobile subscriptions worldwide in Q1 2015 was 7.2 billion, 40 % of which are associated with smartphones. The report also predicts that, by 2016, the number of smartphone subscriptions will surpass those of basic phones, and the number of smartphones will reach 6.1 billion by 2020 [4]. This growth is accompanied by an equally phenomenal boom in mobile applications. According to the research firm MarketsandMarkets, the total global mobile applications market is expected to be worth $25 billion by 2015 (up from about $6.8 billion in 2010) [17]. A 2012 study by the Application

M. Helfert et al. (Eds.): CLOSER 2015, CCIS 581, pp. 263–278, 2016.
DOI: 10.1007/978-3-319-29582-4_14

Table 1. Worldwide device shipments by segment (thousands of units) [18].

Device type	2012	2013	2014	2017
PC (Desk-based and notebook)	341,263	315,229	302,315	271,612
Ultramobile	9,822	23,592	38,687	96,350
Tablet	116,113	197,202	265,731	467,951
Mobile phone	1,746,176	1,875,774	1,949,722	2,128,871
Total	2,213,373	2,411,796	2,556,455	2,964,783

Developers Alliance found that 62 % of the U. S. online population owned *app-capable* devices and that 74 % of those device owners use mobile applications. A March 2015 ReportLinker study estimated that there were more than 3.17 million applications available on various app stores [25].

As the rendering capabilities of mobile devices improves, mobile applications are becoming increasingly graphics-intensive. This requires intensive computations that quickly drain the device's battery. Several solutions are being developed to reduce power consumption in graphics-intensive mobile applications. Some solutions are to be used at development time while others are to be used when the application is running. The former focus on tools that help developers estimate power consumption at development time. For example, in [30], the authors present SPOT (System Power Optimization Tool), which is a model-driven tool that automates power consumption emulation code generation. In [7], the authors use program analysis during development time to estimate mobile application energy consumption. The latter type of solutions focus on reducing power consumption of hardware components such as the GPU or NIC at run-time. Examples include the *racing to sleep* technique (that sends data at the highest possible rate), wide channels, and multiple RF chains [6].

A third alternative is application offloading, the process of running compute-intensive tasks on servers (often in the cloud) and delivering the results of these computations to mobile devices through their wireless interfaces. However, these wireless interfaces also may consume substantial amounts of power when receiving large amounts of data as is typical in many modern, interactive, graphics-intensive mobile applications. It is therefore important to understand the power consumption implications of the two alternatives: running the graphics-intensive application on the cloud or on the mobile device itself.

In this paper, we use *mobile cloud gaming* as an example to analyze and compare these two alternatives in terms of power consumption. We show through actual hardware specifications that, despite the recent introduction of ultra low power GPUs for mobile devices, it remains far more power efficient to offload graphics-intensive tasks to cloud servers. To make our discussion concrete, we focus on two cases of mobile devices: (i) notebooks and (ii) smartphones. In both cases, we only consider gaming using the device's WiFi interface (not its cellular interface.) The reason for this is that the high latency and high cost make mobile cloud gaming using cellular networks (UMTS, LTE, etc.) an impractical alternative for most consumers. We will elaborate on this in Sect. 4.

1.1 Paper Organization

This paper is organized as follows. We first discuss some current approaches that aim at reducing power consumption in mobile cloud computing. In Sect. 3, we give an overview of mobile cloud gaming. In Sect. 4, we contrast cellular-based and WiFi-based mobile cloud gaming from the perspectives of power consumption, throughput, latency, and cost. In Sects. 5 and 6, we present power consumption trends in modern mobile GPUs and 802.11 network cards. In Sect. 7, we quantitatively evaluate and compare power consumption of a gaming session in the two previously mentioned scenarios in the context of notebooks. We repeat the same analysis for smartphones in Sect. 8. We summarize the conclusions from our study in Sect. 9.

2 Current Approaches for Power Saving in Mobile Cloud Computing

Research approaches for reducing power consumption in mobile cloud computing environments have focused on one (or both) of the two following directions: (i) adding hardware/software layers in the vicinity of the mobile devices, e.g., cloudlets and (ii) context-aware offloading.

2.1 Cloudlets

The term "cloudlet" was first introduced by M. Satyanarayanan and his team at Carnegie Mellon University [1]. A cloudlet "represents the middle tier of a 3-tier hierarchy: mobile device — cloudlet — cloud" [27]. It serves as a "data center in a box" with the goal of bringing the cloud closer to the mobile device. Several researchers have used the idea of cloudlets to develop offloading approaches. A promising direction is to offload compute-intensive processing to the cloudlet instead of offloading to the cloud. An example of work that used the idea of cloudlets is [16] where the authors propose an architecture for mobile cloud computing that includes a middle layer composed of cloudlets between mobile devices and the cloud infrastructure. Cloudlets are deployed next to IEEE 802.11 access points and are used as a "service point" that improves the performance of mobile cloud services accessed by nearby mobile devices. The authors also propose an offloading algorithm that decides whether or not to offload. The algorithm takes into consideration the energy consumption for task execution and the network status while satisfying constraints related to task response time. To further improve performance, the authors introduce a data caching mechanism deployed at cloudlets.

2.2 Context-Aware Offloading

In this approach, the current "context" of the mobile device is taken into account when making decision as to whether or not to offload to the clouds. The term "context" may mean different things. For example, in [32], the authors consider a mobile cloud computing architecture with multiple resources (e.g., mobile ad-hoc network, cloudlet, and public clouds.) They propose a context-aware offloading system that takes into account these resources to provide code offloading decisions that help in selecting the wireless medium and the potential cloud resources to be used as the offloading location based on the device's context. In [3], the authors present an algorithm called MAO (Mobile Application's Offloading) triggered by a "context" that consists of the current CPU load and state of charge (SoC) of the battery. The algorithm also differentiates between interactive and delay tolerant mobile applications. When the algorithm cannot satisfy a user's quality of experience (QoE) and/or energy efficiency requirements, it rejects the job.

In [14], the authors consider the case of multiple tasks that dynamically arrive at the nodes of a mobile ad hoc network-based cloud computing environment. They propose a set of online and batch scheduling heuristics that aim at improving performance and reducing energy consumption by offloading compute-intensive applications. Their experimental evaluation focused on both user-centric and system-centric metrics such as the average makespan, the average waiting time, the average slowdown and the average utilization.

3 Mobile Cloud Gaming

Mobile cloud computing (MCC) is the process of offloading compute-intensive tasks from mobile devices to cloud servers [28,29]. The purpose is often to save power on the mobile device and/or access servers with much higher computing power. A prime example of MCC is *mobile cloud gaming* which is the process of providing video games on-demand to consumers through the use of cloud technologies. One benefit is that the cloud, instead of the user's device, carries out most of the computations necessary to play the game, e.g., complex graphical calculations. This is obviously a tremendous advantage in case the player uses a battery-powered, mobile device. Even when power is not a critical issue for the user's device, cloud gaming still provides other cloud services, e.g., storage. Cloud gaming enables power savings also on the cloud itself as it makes it possible that several players simultaneously share cloud GPUs. For example, Nvidia's VGX Hypervisor manages GPU resources to allow multiple users to share GPU hardware while improving user density and the utilization of GPU cycles [22]. To illustrate, a single cloud gaming-capable Nvidia VGX K2 unit requires 38 W per cloud user [24], whereas a comparable single-user Nvidia GTX 690 consumer unit requires 300 W to operate [23]. In this case, cloud gaming can reduce the overall graphics-related power consumption by 87 %.

4 Cellular-Based vs. WiFi-Based Mobile Cloud Gaming

Mobile cloud gaming may be achieved using cellular connections or WiFi connections. While both options are technically possible and relatively comparable in terms of power consumption, the WiFi option seems much more attractive when we consider throughput, latency and cost. In this section, we present results from recent studies analyzing power consumption, throughput, latency, and cost in both scenarios:

4.1 Power Consumption and Throughput

In [2], the authors analyze power consumption of smartphones. In particular, they studied power consumption of the two main networking components of the device: WiFi and GPRS (provided by the GSM subsystem). The test consisted of downloading a simple file via HTTP using `wget`. The files contained random data, and were 15 MiB for WiFi, and 50 KiB for GPRS. While the test was not a gaming session, it still gave valuable insights. The experiments showed that WiFi achieved a throughput of 660.1±36.8 KiB/s, and GPRS 3.8±1.0 KiB/s. However, they both show *comparable* power consumption far exceeding the contribution of the RAM and CPU (Fig. 1). The experiments also showed that, with the increase in throughput possible using WiFi, CPU and RAM power consumption also increases reflecting the increase in the cost of processing data with a higher throughput.

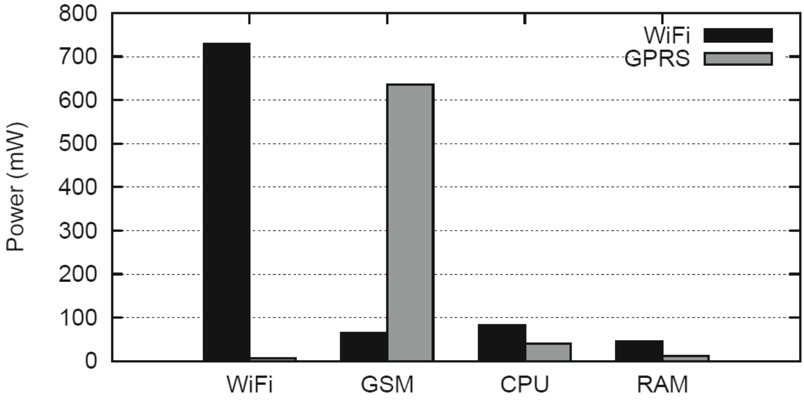


Fig. 1. Power consumption of WiFi and GSM modems, CPU, and RAM [2].

4.2 Latency

In the context of mobile cloud gaming, latency refers to the timespan between a user's action and the corresponding reaction [12], e.g., time between the action of pressing a button and seeing a character in the game move as a result of that action. High latency is a real challenge in mobile cloud gaming. Wireless connections (WiFi and cellular) and even wired residential end host round trip times (RTTs) can exceed 100 ms [13]. To many gamers, this is the point when a game's responsiveness becomes unacceptable. A recent effort to reduce latency in mobile cloud gaming is Outatime, a speculative execution system for mobile cloud gaming that is able to mask up to 250 ms of network latency [13]. It produces speculative rendered frames of future possible outcomes, delivering them to the client one entire RTT ahead of time.

While latency is an issue in both cellular-based and WiFi-based mobile gaming, WiFi connections typically have much less latency than cellular connections [12].

4.3 Cost

Cost is also a major factor in favor of WiFi-based mobile cloud gaming. For example, in [12], the authors give an analytical assessment that shows that the cost (from cellular data transfer) of a gaming session of one hour would be about 2.36 Euros without including the likely additional usage fee to be paid to the cloud gaming provider.

As we may conclude from the previous discussion, WiFi-based mobile cloud gaming is currently more practical than cellular-based mobile cloud gaming. We, therefore, limit our discussion to this option in the remainder of this paper.

5 Power Consumption Trends in Modern Mobile GPUs

It is currently generally true that GPUs offering a good rendering capability consume much power for operation and cooling. To illustrate the current power consumption trends of mobile GPUs, we list in Table 2 some modern notebook GPUs and their respective power consumptions. The table suggests that playing a game on a notebook equipped with one of the listed GPUs may not be a viable option. For example, the Dell Precision M6700 mobile workstation (which Dell touted as the "world's most powerful 17.3" mobile workstation") is equipped with the Nvidia Quadro K5000M GPU. The configuration can pull 98W of power when running on battery under a heavy CPU or GPU load. This means that it would be possible to drain the system battery in about an hour [20]. Even with this limited ability to support long running, compute-intensive applications, this configuration costs more than $2K. Better battery life may be possible but with much more expensive configurations. Efforts are underway to develop mobile devices with power efficient computing components (e.g., multicore CPUs and ultra low power GPUs) and batteries that can run compute-intensive applications (e.g., games and other graphics-intensive applications) for many hours. For example, Nvidia is introducing Tegra 4, a mobile GeForce GPU with up to 72 custom cores, a quad-core ARM Cortex-A15 processor with a fifth Companion Core that further improves performance and battery life. According to Nvidia, a battery of a capacity of 38 watt-hours would be sufficient to operate a Tegra

Table 2. Energy consumption of some modern notebook GPUs.

GPU card	Power consumption (Watts)
NVIDIA GeForce GTX 680M SLI	2×100
AMD Radeon HD 7970M Crossfire	2×100
NVIDIA GeForce GTX 680MX	122
NVIDIA GeForce GTX 675M SLI	2×100
GeForce GTX 680M	100
Quadro K5000M	100
AMD Radeon HD 7970M	100

4 mobile device running a gaming application between 5 and 10 h. This corresponds to a power consumption (for the entire device) of 4 to 8 W [9]. However, it is expected that mobile devices with these high-end configurations will remain beyond the reach of average users for the foreseeable future.

6 Power Consumption Trends in Modern Notebook NICs

The original 1997 release of the IEEE 802.11 standard operated in the 2.4 GHz frequency band and provided a data bit rate of 1 to 2 Mb/s. The standard release approved in February 2014 (known as 802.11ad) operates in the 2.4/5/60 GHz frequency bands and provides a data bit rate of up to 6.75 Gbit/s. While higher bit rates often translate into higher power consumption, this is less true in recent ultra-low power 802.11 standards. For example, today's fastest 3 antenna 802.11n device can achieve 450 Mbps. A single antenna 802.11ac device can achieve a similar bit rate with similar power consumption. This means that a typical tablet with single antenna 802.11n 150 Mbps WiFi can now support 450 Mbps with 802.11ac without any increase in power consumption or decrease in battery life [19].

7 Graphics-Intensive Applications: GPUs vs. NICs

To assess the benefits of using a mobile GPU versus offloading to the cloud, we consider gaming as it is a typical example of graphics-intensive mobile applications. Specifically, we consider four modern games that rely heavily on GPUs. We compare two scenarios in terms of power consumption. In the first scenario, the game is run entirely on the mobile device and uses only its GPU. In the second scenario, we consider an execution where the game is run on a cloud server and the mobile device only receives and renders sequences of frames produced by the server. We analytically evaluate power consumption in these two scenarios and show that, with modern wireless technology, offloading is a far better alternative to running graphics-intensive applications using the device's GPU. To make the comparison even more in favor of the GPU-based alternative,

Table 3. Average frame rate of some combinations of GPU cards, games, and resolutions.

GPU card	GRID autosport	Watch dogs	Titanfall	Thief
	L|M|H|U	L|M|H|U	L|M|H|U	L|M|H|U
GeForce GTX 770M (75 W)	199.6|130.3|92.6|46.5	80.7|66.1|27.7|19.8	60|60|59.3|48.3	57.1|51.3|46.8|26.6
GeForce GTX 860M (60 W)	192.15|109.65|88|47.2	71.2|60.7|27.7|18.9	60|60|59.5|42.4	60.5|52.7|44|23.95
GeForce GTX 850M (40–45 W)	166.65|99.33|68.3|34.7	61.8|52.3|20.75|14.7	60|59.7|53.25|34.3	46.45|39.6|36.65|18.2
GeForce GTX 765M (50–75 W)	191.9|130.7|74.1|34.8	81.3|56.9|21.1|	60|59.7|54.3|35.6	58.2|43.1|37|19.1

we ignore the power consumption of the device's disk. We assume that, when a graphics-intensive application is run on a mobile device, most of the power is consumed by the device's GPU. This is becoming increasingly true with the wide availability of mobile devices with solid-state disk drives.

To compare power consumption in the two scenarios, we first present a simple model that captures the interactions between the player and the gaming application. We will assume that, during a given gaming session of duration t, the player takes an action after every r seconds on average. We call r the *reactivity* of the player. To respond to the player's action, the application generates a video stream of length v seconds.[1] So, during the entire session, the application generates t/r video sequences whose length is v seconds each. In total, the application generates tv/r seconds of video during the given gaming session.

7.1 Scenario 1: Gaming Using the Mobile Device's GPU

To assess the power consumed by a notebook's GPU in a gaming session, we used the benchmark presented in [21]. The benchmark has a large number of notebook GPUs and a number of popular games. For each combination of game and GPU card, the benchmark gives the average number of frames per second (fps) that the GPU card achieves with four different resolution levels: Low (L), Medium (M), High (H), and Ultra (U). The benchmark considers that a frame rate of 25 fps is sufficient for fluent gaming. For the purpose of this study, we considered four GPU cards and four 2014 games, namely GRID Autosport, Watch Dogs, Titanfall, and Thief. Table 3 gives the frame rates obtained in the given combinations[2]. The resolutions in the table are as follows: Low (1024×768), Medium (1366×768), High (1920×1080 for the first two games and 1366×768 for the last two games), and Ultra (1920×1080). Table 3 also gives power consumption for the four GPU cards.

As an example, consider a mobile device equipped with a GPU of type Nvidia GeForce GTX 850M. As shown in Table 3, this GPU card will consume between 40 and 45 W in one hour. We will show that offloading to the cloud (Scenario 2) brings an order of magnitude reduction in terms of the power consumed by the mobile device.

[1] This is to simplify our discussion. In practice, the application likely generates two video sequences of different lengths in response to two different actions.

[2] The missing value in the last row corresponds to a test that could not be run because the GPU card could not support a sufficiently acceptable frame rate.

7.2 Scenario 2: Mobile Cloud Gaming

We now evaluate the required data bit rate that the NIC card of a notebook would have to support to achieve the same game fluency (i.e., 25 fps) for one of the four GPU cards of Table 3. As an example, consider again the Nvidia GeForce GTX 850M (which is the best of the four GPUs in terms of power consumption.) For the game GRID Autosport and for low resolution, the Nvidia GeForce GTX 850M is able to support 166.65 fps which is: $166.65 \times 1024 \times 768 \times 8 = 1048471142.4$ bits/s (assuming a color depth of 8 bits/pixel). Thus the NIC card would have to operate at a bit rate of about 1.05 Gb/s. A similar computation for the Ultra high resolution level gives us a bit rate of: $34.7 \times 1920 \times 1080 \times 8 = 575631360$ bits/s. Thus, to support the same gaming fluency at the Ultra-high resolution level, the NIC would have to operate at 575 Mb/s. Note that the required bit rate at the Ultra-high resolution level is almost half of that of the required bit rate at the low resolution level because the GPU supports a lower frame rate at the Ultra-high resolution level. To support these bit rates, the mobile device's NIC would have to be 802.11ad compliant. The 802.11ad standard is able to support bit rates up to 6.77 Gbit/s.

To evaluate the power consumed by the device's wireless networking card during the considered gaming session, we will assume a model of a wireless networking card that consumes ρ_{tx} watts when in transmit mode and ρ_{rx} watts when in receive mode. With single-antenna 802.11 devices, the devices cannot send and receive simultaneously. This normally implies that one has also to take into account the cost of frequently switching the device's radio between the transmit and the receive mode. However, this is changing as mobile devices are now increasingly being equipped with MIMO (multiple-input and multiple-output) technology enabling the use of multiple antennas at both the transmitter and receiver. In fact, Mobile Experts predicts that the use of MIMO technology will reach 500 million PCs, tablets, and smartphones by 2016 [15]. As a result, we will only take into account power consumption due to transmission, reception, and idling. We will note the power consumption of the radio during idling by ρ_{id}.

Let μ_t and μ_r be the transmission and reception rates respectively. Let l be the length of the packet sent to the application when the player takes an action. The time needed to transmit this packet is then: l/μ_t. Let t be the length of the entire gaming session (in seconds). During the time t, the device transmits t/r times where r is the player's reactivity (defined earlier). The total time during which the device transmits is therefore:

$$\frac{tl}{r\mu_t}\text{secs.} \tag{1}$$

The corresponding power consumption during the period of time t is:

$$P_{tx} = \frac{\rho_{tx}tl}{r\mu_t} \tag{2}$$

To evaluate the power consumed by the device's receiver, recall that our model assumes that, to respond to each player's action, the application generates

Table 4. Power consumption for the Intel Dual Band Wireless-AC 7260 802.11ac, 2×2 Wi-Fi Adapter [8].

Mode	Power (mWatts)
Transmit	2000
Receive	1600
Idle (WLAN associated)	250
Idle (WLAN unassociated)	100
Radio off	75

a video stream of length v seconds. The devices spends v/μ_r seconds to receive each of these video streams. Since we have t/r of these video streams during the considered time period of length t, the device's NIC receives video streams during:

$$\frac{tv}{r\mu_r}\texttt{secs}. \quad (3)$$

Let P_{rx} be the power that the device's NIC consumes to receive the t/r video sequences. P_{rx} can be given by:

$$P_{rx} = \frac{\rho_{rx} tv}{r\mu_r} \quad (4)$$

The device's NIC is in the idle mode when it is not transmitting and not receiving. This occurs during:

$$t - \frac{tl}{r\mu_t} - \frac{tv}{r\mu_r}\texttt{secs}. \quad (5)$$

The power consumed by the device's NIC while idling is therefore:

$$P_{id} = \rho_{id} t(1 - \frac{l}{r\mu_t} - \frac{v}{r\mu_r}) \quad (6)$$

Let $P_{NIC}(t)$ be the power consumed by the wireless NIC during the t-second gaming session. $P_{NIC}(t)$ is then:

$$\begin{aligned} P_{NIC}(t) &= P_{tx} + P_{rx} + P_{id} \\ &= \frac{\rho_{tx} tl}{r\mu_t} + \frac{\rho_{rx} tv}{r\mu_r} + \rho_{id} t(1 - \frac{l}{r\mu_t} - \frac{v}{r\mu_r}) \end{aligned}$$

In practice, one must consider values for ρ_{rx} that accommodate high reception rates (for high definition gaming) and values for ρ_{tx} that correspond to low transmission rates since the user's actions usually translate into short packets.

To illustrate, we consider the case of an HP EliteBook Folio 1040 G1 Notebook PC. This notebook is equipped with the Intel Dual Band Wireless-AC 7260 802.11ac Wi-Fi Adapter whose power consumption is given in Table 4 [8].

Assume that the NIC card is 80 % of the time in reception mode, 10 % of the time in transmit mode, and is idle (but associated) 10 % of the time. If we apply our power model to this WiFi adapter, power consumption in one hour would be (approximately):

$$\begin{aligned} P_{NIC}(t) &= P_{tx} + P_{rx} + P_{idle} \\ &= 0.1 \times 2000 + 0.8 \times 1600 + 0.1 \times 250 \\ &= 1505 \texttt{ mW} \end{aligned}$$

assuming the highest Rx and Tx power levels.

Considering the example of a notebook equipped with a GPU of type Nvidia GeForce GTX 850M (Sect. 7.1), we can estimate that, in one hour, the GPU card will consume about between 0.8×40 W and 0.8×45 W, i.e., between 32 W and 36 W, assuming a GPU utilization of 80 % similar to our assumption of the NIC card being in the Rx mode 80 % of the time.

From the results obtained in the two scenarios, it is clear that using the wireless networking interface in a gaming session consumes much less power than using a modern GPU card installed on the same device. Specifically, the power consumed using the wireless card would be around $(1505/34000) \times 100$, i.e., around 4.42 % of the power consumed by the on-device GPU.

8 Mobile Cloud Gaming Using Smartphones

We now compare power consumption between GPU-based gaming and cloud-based gaming on smartphones.

8.1 Power Consumption of GPU-Based Gaming on Smartphones

In [10], the authors measured power consumption of a Qualcomm Adreno 320 GPU in a Google Nexus 4 smartphone. They used two games in their tests: Angry Birds (2D game) and Droid Invaders (3D game). The authors report results for a gaming session that lasted 560 s for Angry Birds and 505 s for Droid Invaders. Throughout the two gaming sessions, power consumption remained approximately at around 1750 mW for Angry Birds and at around 2000 mW for Droid Invaders. We will use the average of these two numbers (1875 mW) as an estimate of the average power consumption of both 2D and 3D games.

8.2 Power Consumption of Cloud-Based Gaming on Smartphones

To compare power consumption of cloud-based gaming with GPU-based gaming, we first need to evaluate the NIC bit rate that would be necessary to provide a gaming experience comparable to the one achieved through GPU-based gaming. For this, we used results from the GFXBench 3.0 benchmark, a cross-platform OpenGL ES 3 benchmark designed for measuring graphics performance, render quality and power consumption on several types of devices including smartphones. In particular, the benchmark has battery and stability tests that measure

Table 5. Frame rates for the Adreno 320 GPU on a Google Nexus 4 and on a Samsung Galaxy S4 using the Manhattan benchmark [5].

Smartphone model	GPU	Resolution	Frame rate
Google Nexus 4 (LG E960)	Adreno 320	1196 × 768	9.2
Google Nexus 5	Adreno 330	1794 × 1080	10.1
Samsung GT-I9507 Galaxy S4	Adreno 320	1920 × 1080	5.4
Samsung GT-I9515 Galaxy S4 Value Edition	Adreno 320	1920 × 1080	5.1
Samsung Galaxy S4 Active (GT-I9295, SGH-I537)	Adreno 320	1920 × 1080	5.1
Samsung Galaxy S4 (GT-I9505, GT-I9508, SC-04E, SCH-I545, SCH-R970, SGH-I337, SGH-M919, SPH-L720)	Adreno 320	1920 × 1080	5.1

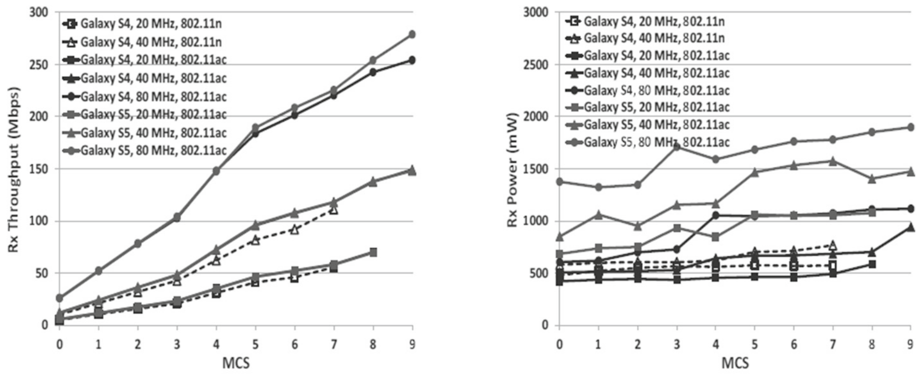
the devices battery life and performance stability by logging frames-per-second (fps) performance and expected battery running time while running sustained game-like animations [5]. We focused on results for the Adreno 320 GPU on a Google Nexus 4, which is the same configuration used in the GPU-based scenario of the previous section.

Table 5 shows the frame rate for several tests using the Manhattan benchmark [5]. Row 1 of the table shows that the Adreno 320 GPU on a Google Nexus 4 achieved a frame rate of 9.2 fps. Considering this frame rate and the given resolution (1196 × 768), the NIC bit rate that would be necessary to achieve a similar gaming experience can be derived as: $9.2 \times 1196 \times 768 \times 24$ (bits/pixel) $= 202810982.4$ bps ≈ 203 Mbps.

We now turn to evaluating the power needed on the NIC to sustain this bit rate. For this, we use the results from [26] where the authors experiment with a variety of smartphones supporting different subsets of 802.11n/ac features. In particular, the authors measured throughput and power consumption in a Galaxy S4 using different configurations. Based on their findings for the Galaxy S4 used in the experiment, only 802.11ac offers Rx throughput levels sufficient for the considered gaming bit rate (of 203 Mbps).

Figure 2 (reproduced from [26]) shows that the best Rx throughput with 802.11ac was about 250 Mbps. Power consumption in this case was about 1100 mW.

The authors did not provide measurements for the throughput and power consumption in transmit mode with 802.11ac. They, however, measured throughput and power consumption in transmit mode with 802.11n. Figure 3 shows their results. In particular, the results show that it is possible to achieve a Tx throughpout of more than 40 Mbps with as little power as 800 mW. Note that, in a cloud-based gaming session, a Tx throughout of 40 Mbps is typically sufficient. The authors also measured power consumption of the Galaxy S4 when it is in non-communication modes, i.e., power saving mode (PSM) or idle. Their results (Table 6) show that the highest 802.11ac power consumption in PSM was 31 mW



(a) Rx throughput. (b) Rx power consumption.

Fig. 2. 802.11ac throughput and power comparison for Galaxy S4 and Galaxy S5 with a channel width of 20/40/80 MHz and FA on [26].

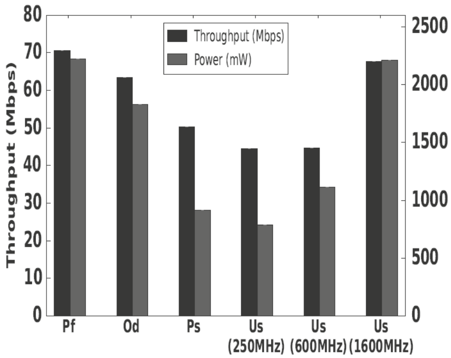


Fig. 3. Comparison of different CPU Governors/Frequencies for Galaxy S4 (802.11n) [26].

Table 6. Power consumption (in mW) in non-communicating modes [26].

Configuration	PSM	Idle
802.11n, 20 MHz, SS	24 ± 16	398 ± 7
802.11n, 40 MHz, SS	25 ± 5	413 ± 2
802.11ac, 20 MHz, SS	22 ± 9	374 ± 7
802.11ac, 40 MHz, SS	20 ± 9	425 ± 3
802.11ac, 80 MHz, SS	19 ± 10	529 ± 11

and that the highest 802.11ac power consumption when idle was 540 mW. The relatively high idle mode power consumption of larger channel widths (80 MHz) has also been observed by other studies (e.g., [31]).

Based on all the previous results from [26] and assuming that, in a cloud-based gaming session, the device's 802.11 adapter spends 80 % of the time receiving, 10 % of the time transmitting, and 10 % of the time idle, the total power consumed in one hour by the 802.11 adapter would be:

$$\begin{aligned} P_{NIC}(t) &= P_{tx} + P_{rx} + P_{idle} \\ &= 0.1 \times 800 + 0.8 \times 1100 + 0.1 \times 540 \\ &= 1014 \text{ mW} \end{aligned}$$

Comparing power consumption in the two scenarios: using GPU-based gaming (which is 1875 mW as derived in Sect. 8.1 and cloud-based gaming (which is 1014 mW as derived in this section), we conclude that, in the considered smartphone configuration, cloud-based gaming can potentially result into a power saving of about 46 %.

9 Conclusion

Reducing power consumption in mobile devices is crucial. Mobile cloud computing is one alternative that has been increasingly used to reduce power consumption on mobile devices. While offloading is generally accepted to be effective, little research has been conducted to quantify the exact difference in terms of power consumption between scenarios where mobile devices access applications run on the clouds and scenarios where those same applications are run on the mobile devices themselves. In this paper, we used mobile cloud gaming as a case study to analyze and compare power consumption in the two scenarios. Our study shows that substantial savings in power consumption may be achieved when graphics-intensive applications are run on the clouds instead of mobile devices. We call the computing model that enables mobile devices to access advanced cloud-based visualization capabilities Visualization-as-a-Service (VaaS). Based on our analysis, we posit that VaaS is a viable computing model despite the recent advances in terms of low power hardware for mobile devices.

In a survey of computation offloading for mobile systems [11], the authors predict that "mobile computing speeds will not grow as fast as the growth in data and the computational requirements of applications." As a result, offloading will remain a natural solution to the problem of improving performance while reducing energy consumption of mobile devices. We concur with this prediction and believe that more work is needed in the area of mobile cloud computing both in terms of new architectures and in terms of new offloading techniques.

References

1. Elijah: Cloudlet-based Mobile Computing. http://elijah.cs.cmu.edu
2. Carroll, A., Heiser, G.: An analysis of power consumption in a smartphone. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, USENIXATC 2010, pp. 21–21. USENIX Association, Berkeley (2010). http://dl.acm.org/citation.cfm?id=1855840.1855861
3. Ellouze, A., Gagnaire, M., Haddad, A.: A mobile application offloading algorithm for mobile cloud computing. In: 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 34–40, March 2015
4. Ericsson: Ericsson mobility report: on the pulse of the networked society. Technical report, Ericsson, June 2015
5. GFXBench: Gfxbench 3.0 directx (2015). http://www.gfxbench.com
6. Halperin, D., Greenstein, B., Sheth, A., Wetherall, D.: Demystifying 802.11n power consumption. In: Proceedings of the International Conference on Power-Aware Computing and Systems. HotPower, Vancouver (2010)
7. Hao, S., Li, D., Halfond, W.G.J., Govindan, R.: Estimating mobile application energy consumption using program analysis. In: Proceedings of the the International Conference on Software Engineering (ICSE), San Francisco, California, May 2013
8. Hewlett Packard: HP EliteBook Folio 1040 G1 Notebook PC. Technical report (2013)
9. Hruska, J.: Nvidia's Tegra 4 Demystified: 28nm, 72-core GPU, Integrated LTE, and Questionable Power Consumption (2013). http://www.extremetech.com
10. Kim, Y.G., Kim, M., et al.: A novel GPU power model for accurate smartphone power breakdown. ETRI J. **37**(1), 157–164 (2015)
11. Kumar, K., Liu, J., Lu, Y.H., Bhargava, B.: A survey of computation offloading for mobile systems. Mob. Netw. Appl. **18**(1), 129–140 (2013)
12. Lampe, U., Hans, R., Steinmetz, R.: Will mobile cloud gaming work? findings on latency, energy, and cost. In: Proceedings of the 2013 IEEE Second International Conference on Mobile Services, MS 2013, pp. 103–104. IEEE Computer Society, Washington (2013). http://dx.doi.org/10.1109/MS.2013.21
13. Lee, K., Chu, D., Cuervo, E., Kopf, J., Grizan, S., Wolman, A., Flinn, J.: DeLorean: using speculation to enable low-latency continuous interaction for mobile cloud gaming. Technical report, Microsoft Research, August 2014
14. Li, B., Pei, Y., Wu, H., Shen, B.: Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. J. Supercomput. **71**(8), 3009–3036 (2015)
15. Madden, J.: MIMO adoption in mobile communications forecast: devices by operating system and user type, worldwide, 2010–2017, 1Q13 Update. Technical report, Mobile Experts, June 2011
16. Magurawalage, C.M.S., Yang, K., Hu, L., Zhang, J.: Energy-efficient and network-aware offloading algorithm for mobile cloud computing. Comput. Netw. **74**, 22–33 (2014)
17. MarketsandMarkets: World Mobile Applications Market - Advanced Technologies, Global Forecast (2010–2015). Technical report, MarketsandMarkets (2010)
18. Milanesi, C., Tay, L., Cozza, R., Atwal, R., Nguyen, T.H., Tsai, T., Zimmermann, A., Lu, C.K.: Forecast: devices by operating system and user type, worldwide, 2010–2017, 1Q13 Update. Technical report, Gartner, 28 March 2013

19. Netgear: Next Generation Gigabit WiFi - 802.11ac. Technical report (2012)
20. Notebook Review: Dell Precision M6700 Owner's Review (2015). http://forum.notebookreview.com/dell-latitude-vostro-precision/679326-dell-precision-m6700-owners-review.html
21. NoteBookCheck: Computer Games on Laptop Graphic Cards (2014). http://www.notebookcheck.net/Computer-Games-on-Laptop-Graphic-Cards.13849.0.html
22. Nvidia: Building Cloud Gaming Servers (2015). http://www.nvidia.com/object/cloud-gaming-benefits.html
23. Nvidia: GeForce GTX 690 Specifications (2015). http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-690/specifications
24. Nvidia: Grid GPUs (2015). http://www.nvidia.com/object/grid-boards.html
25. ReportLinker: Global Mobile Application Market 2015–2019. Technical report, ReportLinker, March 2015
26. Saha, S.K., Deshpande, P., Inamdar, P.P., Sheshadri, R.K., Koutsonikolas, D.: Power-throughput tradeoffs of 802.11n/ac in smartphones. In: Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM), Hong Long, Spain, 26 April–1 May 2015
27. Satyanarayanan, M., Chen, Z., Ha, K., Hu, W., Richter, W., Pillai, P.: Cloudlets: at the leading edge of mobile-cloud convergence. In: 2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), pp. 1–9, November 2014
28. Shiraz, M., Gani, A., Khokhar, R., Buyya, R.: A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. IEEE Commun. Surv. Tutorials **15**(3), 1294–1313 (2013)
29. Soliman, O., Rezgui, A., Soliman, H., Manea, N.: Mobile cloud gaming: issues and challenges. In: Daniel, F., Papadopoulos, G.A., Thiran, P. (eds.) MobiWIS 2013. LNCS, vol. 8093, pp. 121–128. Springer, Heidelberg (2013). http://dx.doi.org/10.1007/978-3-642-40276-0_10
30. Thompson, C., Schmidt, D.C., Turner, H.A., White, J.: Analyzing mobile application software power consumption via model-driven engineering. In: Benavente-Peces, C., Filipe, J. (eds.) PECCS, pp. 101–113. SciTePress (2011)
31. Zeng, Y., Pathak, P.H., Mohapatra, P.: A first look at 802.11ac in action: energy efficiency and interference characterization. In: Proceedings of the 13th IFIP International Conferences on Networking, Trondheim, Norway, 2–4 June 2014
32. Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R.: A context sensitive offloading scheme for mobile cloud computing service. In: 2015 IEEE 8th International Conference on Cloud Computing (CLOUD), pp. 869–876, June 2015