Nokia Windows Mobile's Power Consumption Measurements and Analysis

Luan RUÇI[1], Luan KARÇANAJ[2], Olimpjon SHURDI[2]

[1]PhD (under graduation), FTI, Polytechnic University of Tirana, Albania

[2]Prof.Dr at Kristal University of Tirana, [2]Prof.Asoc at FTI, Polytechnic University of Tirana, Albania

luan.rucci@gmail.com, luankarcanaj@gmail.com, oshurdi@fti.edu.al

ABSTRACT

Recent mobile Smartphone devices have a very rich set of components and can handle multiple general purpose programs that sometimes seem not to be so UE friendly, specifically for UE internal resources, power drain, and signalling storm. Understanding their usage and impact on UE, mobile network and end user perception, will help to better optimize "frustrating" situations for both ends. Nowadays, reducing power consumption seems to be the most crucial design for both mobile and other small computing devices that are not always connected to any power source. Good energy management requires a good understanding of where and how the energy is used. To conclude we will present a detailed analysis of power consumption for a recent mobile phone, Nokia Lumia 625 model. We measured and discussed the significance of the power drain through various components, apps and identified the most promising areas to focus on for further improvements of power management. We would like to present this power breakdown in benchmarks as well as in a number of realistic usage scenarios. These results would better validated by overall power measurements of other Smartphones of the same or different OS's. Being able to generate a device-specific scalable power consumption model is therefore crucial for understanding, designing, and implementing better mobile application software. A proper energy accounting infrastructure will help both application developers and Smartphone users to extend the battery life of their devices and make decisions about where to spend the remaining device power, potentially in real-time measurements.

Keywords: *Smartphone, Energy measurement, Applications, Power tools, Mobile communication, Test scenario.*

1 INTRODUCTION

Modern high-end mobile phones combine the functionalities of a pocket-sized communication device with PC-like capabilities, resulting in what are generally referred to as Smartphone's. They integrate such diverse functionality and have a direct impact on energy consumption. Modern Smartphone's use heterogeneous multicore SoC which includes CPU, GPU, DSP and various application specific accelerators. It provides opportunities to realize compute intensive applications on a battery-powered and resource-limited mobile device by assigning each task or user activity to the most suitable computing core. The rich functionality increases the pressure on battery lifetime, and deepens the need for effective energy management [1]. Mobile devices derive the energy required for their operation from batteries. Hence, optimal management of power consumption of these devices is critical. Unfortunately, today Smartphone's battery technology has not kept up with this evolution, thus making the new power hungry for capabilities and also a hinder for further development. A core requirement of effective and efficient management of energy is a good understanding of where and how the energy is used: how much of the system's energy is consumed by each part of the system and under what circumstances. In fact, today's Smartphone mobiles present significant differences in terms of power consumption signatures depending on the manufacturer, operating system and other contextual factors such as network coverage. Understanding how energy is being consumed by the hardware components is essential in order to design energy-aware systems [12]. By monitoring software and hardware components we will focus on power consumed by specific group of hardware components (contribution brought by them). Our approach is to measure the power consumption of a modern mobile device with Lumia Windows Mobile OS which to our knowledge is not done till now. Specifically speaking, we produced a breakdown of power distribution to CPU&RAM, touchscreen, audio, video and various networking interfaces like 3G, 2G, WiFi and

Bluetooth. We derived an overall energy model of the device as a function of the main usage scenarios. This should provide a good basis for focusing future energy-management research for different versions of Windows mobile devices. As per authors on [2] a proper energy accounting infrastructure will assist both application developers and Smartphone users to extend the battery life of their devices and to make informed decisions about where to spend the remaining device power, potentially in real-time computing. For the purpose of this study we focused on Nokia Lumia 625 Windows mobile device model.
The paper is structured as follows: In Section 2 we describe our measurement platform and benchmarking methodology. In Section 3 we describe each experiment and provide their respective results, and in Section 4 we perform an analysis and reach conclusions based on the results' validation. In Section 5 we describe briefly the limitations to our work.

2 METHODOLOGY

In this section, we will describe the hardware and software used in the experiments, and will explain our benchmarking methodology. As schematic for measurement we use the practical one used by many authors on this field, by measuring the voltage drop on a sense resistor in series with phone battery. There are two elements part of our experimental setup: the device-under-test (DuT) in our case Nokia Lumia 625 Smartphone and the hardware data acquisition system DaQ or GDM 8246 tool.
As per authors on [6] there exists a negligible difference in the usage patterns of two phones of the same manufacturing. However, there is sufficient literature to suggest that Smartphone usage among different human users can lead to different energy models. Furthermore, user mobility (static user or on move), OS software updates, indoor or outdoor kind usage, geographical region and climate, are some of the other factors which may cause significant changes in the power model with the passing of time. Our tests are done while being static (not in the move) and in some tests we show the RSSI signal level, which seems to be one of the main factors to be considered.
There are a lot of testing's and measurements done by many authors handled on real mobile phones [1,2,3,4,5,13,14,15] and open architecture devices like Openmoko Neo Freerunner.

2.1 Tools, Tests and Measurements

2.1.1 Device under testing

We used Nokia Lumia 625, whose main HW info's (related power aware) can be found on [7]. Also, further info's can be found on Nokia Lumia 625 Service manual, freely downloadable from Internet and this meant, removing and preparing the battery for making it ready for our measurements. What we need to highlight is the battery's info and its usage time only which can be used for comparisons against our testing: *Battery model:* BP-4GWA, *Battery capacity:* 2000 mAh, *Battery voltage:* 3.7 V, *Maximum standby time*: 23 days, *Maximum talk time (2G):* 23.9 h, *Maximum talk time (3G):* 15.2 h, *Maximum music playback time:* 90 h, Maximum *video playback time:* 6.8 h, *Maximum cellular network browsing time:* 7.2 h, *Maximum Wi-Fi network browsing time:* 8.7 h.
Nokia Lumia 625 is a Windows mobile OS 8.0 model (our SW version is 8.0.10517.0) and has two main views, making it easy for users to keep up with what's going on. To switch between the **Start screen** and the **Apps menu**, simply swipe left or right as shown in Figure 1. Start screen can have animated live tiles. The tiles can be rearranged and resized up to three sizes. Viewing the apps running in the background in order to control their usage is really important for our tests during power drain measurements. User data backups could be organized through SkyDrive which store users stuff in the clouds. All this apps in background and other proactive user activities have a big impact on phone battery power consumption and from operator's point of view on signalling part generated.

Figure. 1. Start menu and swiping to Apps menu

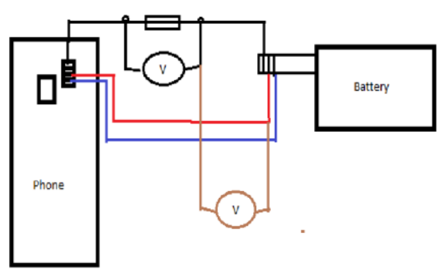

Figure. 2. Measurements schematic

2.1.2 Experimental Testing Setup

To calculate the power consumed by any component, both the supply voltage and current should be determined. To measure current, we inserted sense resistors on the power supply rails of the relevant component. This was not so simple on the DuT we selected, since most of them have been designed with no removable battery. The importance to our testing is that almost all today Smartphone's are factory-populated with 0 ohm [1]. The peak voltage drop did not exceed more than 2-3% of the supply voltage and therefore presented an acceptably small perturbation. With a known resistance and measured voltage drop, current can be easily determined by Ohm's law. For our easiness, we presented only the voltage drops on the resistance, current and consumed power to be calculated. There exists no "manufactured" available method to divide the measurements into per-subsystem or per-application readings on any platform. Therefore, there have been created several statistical models which divides the power consumption into subsystems but which were not considered in our case for the moment. An overall would be formula 1 as per authors on [14]:

Power used = Cpu&Ram + Display + Graphics + Network + Gps + Audio + Mic + :: + Wifi (1)

We ran two main types of benchmarks keeping the same logic as per authors at [1]. First, a series of micro-benchmarks designed to independently characterize components of the system, particularly their normal usage power consumption (voltage drops). Second, we ran a series of ***macro-benchmarks*** based on real usage scenarios. Few phone applications software's were used to help running extra tests. For most of the benchmarks, we ran 10 iterations.

2.1.3 Measurement Hardware and Synchronization

The phone's power consumption was measured by inserting a high-precision 0.22Ω measurement resistor in series between a battery terminal and its connector on the phone as per Figure 2 (the widely used method by many authors in the area of study). We used a Digital meter G^{W}Instek – GDM-8246 sampling board (0.02% DCV accuracy) to measure the voltage across the phone battery and also the voltage drop across our measurement (where the sense resistor was connected via twisted-pair wiring). The key characteristics and accuracy of this DaQ hardware can be found as per info's on [6]. Inserting the measurement resistor increased the circuit resistance, and therefore its power consumption. This was not a problem for our purposes as this is typically less than ~2% of the total power. Our measurement approach yields the power directly consumed (voltage drops) to a group of components and always including CPU and RAM. Many of the features in the energy trace last only a fraction of a second. For example, a scan for available wireless networks lasts around 500 ms, while the transmission of a single packet takes only a few milliseconds [1]. Switching the backlight of the phone from on to off decreases the power consumption of the device to very low values giving the possibility to measure in some cases the contribution of specific measured component. Interesting is the fact that during our first testing we found out that even on 2G or 3G after a certain period of time being on start menu or any other window the voltage drops stabilize between ~30 to 33mV. In our testing's we always waited until the voltage drops stabilized to ~32mV on the resistor and only then we start to run the tests (kind of synchronisation). When RSSI value was needed, we measured it with Nokia_E6 *CellTrack91* application sided nearby our Nokia Lumia 625 since the secret code (##3282#) for seen RSSI on Lumia phones is not available on this OS update our DuT has.

3 RESULTS

3.1 Baseline cases

Prior to running any benchmarks, we established the baseline power state of the device, when no applications requiring any data update or synchronization are running and interfering with our measurements. There are two different cases to consider: *suspended* and *idle* [1]. For the idle case, there is also the application-independent power consumption of the backlight to consider. Other cases are considered in our benchmarking.

3.1.1 Suspended & Idle device

A mobile phone would typically spend a large amount of time in a state when it is not actively used. This means that the application processor is idle, while the communications processor performs a low level of activity, as it must remain connected to the network: be able to receive calls, data updates, SMS messages, etc. As this state tends to dominate the time during which the phone is switched on, the power consumed in this state is critical to the battery lifetime. [1] The device is in the idle state if it is fully awake (not suspended) but no applications are active. This case constitutes the static contribution to power of an active system. We ran this case with the backlight turned off, and we observed a very low voltage drops to ~3 to 5mV. Based on our understanding it can be attributed to OS where sub-devices are put into low-power sleep modes (where appropriate). Main contributors of a phone in Idle are: Cpu, Networks interface and very few from other components. *Accuweather &, Microsalat* apps are as live tiles and allowed on background tasks.

Network	Data	12 Hours /night	2 hours/day
2G (GSM)	OFF	1%	0%
	ON	3%	1%
3G (WCDMA)	OFF	1%	0%
	ON	3%	1%

Table 1. (%) Total power drain during Idle state test (screen off)

For measurement, we used a *Battery Tile & log pro* application on start screen allowing its background task and logging (it logs power status every 30 minutes). As it can be seen when mobile is in idle no any obvious difference between 2G and 3G for the same apps used (1 hour periodicity update for Gmail and Outlook is allowed) and no other apps left opened and running. Adding more apps on Start screen and allowing more apps in background and in live tiling, we strongly believe that power consumption rapidly increases.

3.1.2 Restarting device

In this benchmark we tried to have the mobile device on 2 main network states (2G EDGE and 3G HSPA) and for this we used *RestartNow* application on the start screen (right down in the start screen on Figure 1). All apps running in and the rest are set as per Table 2. On background allowed apps are*: Acuweather, Microsalat, Viber, Battery tile&log pro*, and volume). As phone background photo is chosen nature (sea and winds). Phone colour is set to 1 and Brightness to low.

Conditions	Vibration	Sound	Brightness	Touch	Theme	WiFi	Color Profile
Status	OFF	OFF	Low	Normal	Steel	OFF	1

Table 2. Tests conditions

For some reasons when the *RestartNow* tool is ready to get restart command the voltage drops to 30--40mV from ~80 -- 90mV and some spikes during any Phone network updates stage are seen on this state. We could conclude to 7 main voltage drops (which rely on mobile phone restart phase). We have done 5 iterations for each network status. Too large variation in powering (voltage drop) is observed during restarting phase. Unfortunately, the measurement tool provides accurate data but on second bases, and there is seen too large fluctuation within a second which do not give high accuracy on capturing accurate log data.

Net Status	Results
2G EDGE (~ 50 sec)	1. ~40mV when tool is ready to restart. 2. ~100mV directly after restarts cmnd given. 3. ~10 -- 50mV when powering on. 4. ~100 -- 130mV stabiles for half of time (when booting up) 5. spikes to 210 --240mV for 2-3 sec) and goes back to 6. ~160mV during 2G Net search. 7. ~80 -- 120mV (sometimes spike of 150mV) when stabilize
3G HSPA (~ 50 sec)	1. ~30 -- 40mV when tool is ready to restart (Note 1). 2. ~120 --160mV directly after restarts cmnd given. 3. ~15 -- 50mV during restart. 4. ~100 -- 130mV stabiles for half of time (when booting up) 5. spikes of 210 --230mV for 2-3 sec) and goes back to 6. ~120 -- 160mV during 3G Net search. 7. ~75 -- 110mV (spike of 150mV) when stabilize

Table 3. Phone restarts (voltage drops) tests results

Main contributors are: Cpu&Ram, network and display, graphics, apps (only 4 apps have background task on). This is the only test where we have seen so huge alteration in power consumption and it can be contributed to many factors and sub-devices and processes involved in a restart phase.

3.1.3 Display

Running fewer applications also can save power when using IPS LCD. Many studies show that the component consuming the most intense energy is the display, which means that the type of display plays an important role in power consumption. We measured here the voltage drops during touch, swipe and shift through apps. IPS stands for in-plane switching. IPS-LCDs feature two transistors for each pixel. IPS-LCD may consume more power than a TFT-LCD which features only one transistor for each pixel. OLED screens have a different way of powering.

3.1.3.1 Touch & Swipe tests

During this test, we changed network interface to 2G with no data, locked the background tasks for the 4 previous mentioned apps, and pinned out from main window the *Acuweather* and *Microsalat* in order to eliminate as much as possible any data or background activity to them. We used in this test the *MultiTouchTest* application (black screen was chosen) running on main window and tested conditions and results as per table 4. High touch sensitivity allows use of the touch screen with most of the types of gloves.

a) Touch sensitivity: The drop contribution before test was ~ 30mV and the touch contribution was in spikes of 20 to 25mV (spike for millisecond to seconds depending on time duration) in both cases as per below table:

	Condition	Results
Touch Sensitivity (Brightness - low)	Normal	~30 mV → 48 ±2mV (within 1 sec) (~ 20mV contribution in drops)
	High	~30mV → 52 ±5mV (within 1 sec) (~ 25mV contribution in drops)

Table 4. Touch sensitivity voltage drops tests results (average of 10 iterations)

Main contributors are: Cpu&Ram, display, graphics and network.

b) Swipe_1: In this benchmark we kept the screen pressed for 1 and 3 seconds for both sensitivity conditions: Normal and High. No other obvious difference was noticed between the conditions and voltage drops as per table 5 results. So from ~30mV voltage drop before the test was run, it reached to values of 80--115mV during testing. So touching gives a considerable contribution in voltage drops. Main contributors are: Cpu&Ram, display, graphics, apps.

	Condition	Results
Swipe_1	Normal 1s (down to up)	~30mV → 95 ±10mV (during 1 sec) (~ 60--70mV contribution in drops)
	Normal 3s (down to up)	~30mV → 107 ±7mV (during 3 sec) (~ 70--80mV contribution in drops)
	High 1s (down to up)	~30mV → 97 ±7mV (during 1 sec) (~ 60--70mV contribution in drops)
	High 3s (down to up)	~30mV → 106 ±6mV (during 3 sec) (~70--80mV contribution in drops)

Table 5. Tests results of Swiping voltage drops (average of 10 iterations)

c) Swipe & Scroll through: In this test we just swiped between start screen and apps menu. Also the shift from top down in apps menu was tested. Each test started when there was a stable voltage drop (~32 mV) on sense resistor. Furthermore, in the main screen the *Acuweather* and *Microsalat* apps were removed in order not to interfere with our measurement since they are live tiles.

	Condition	Results
Swipe _2 & Scroll through *(normal / low)*	1 full portrait apps (~10 apps / rows)	~30 → 72 ±8mV (during shift) (~ 30--40mV contribution in drops)
	X full portrait apps (continuous)	~30 → 82 ±8mV (during shift) (~ 45--60mV contribution in drops)

	left→ right (start to apps screen)	~32 → 70 ±5mV (during shift) (~ 30--40mV contribution in drops)
	right→ left (apps to start screen)	~32 → 71 ±6mV (during shift) (~33--45mV contribution in drops)

Table 6: Swiping & Shift voltage drops tests results (average of 10 iterations)

Main contributors are: Cpu&Ram, display, graphics, network interfaces and used apps.

3.1.3.2 Screen Brightness

In this benchmark we changed the brightness of the phone on its 3 levels and monitored the voltage drops on start screen for 30sec on each iteration, the time of lock screen set by us (only one full start screen with apps except *Acuweather* and *Microsalat* which were removed). A high contribution in power drain (voltage drop) can be noted from *low* to *high* brightness change (almost tripled). Values for Low case also verify almost all cases tested and measured.

	Condition	Results
Screen Brightness (1 full start screen)	Low	33 ±3mV
	Medium	50 ±5mV
	High	85 ±5mV

Table 7. Brightness voltage drops tests results (average of 10 iterations)

Measurements were taken directly on setting menu windows. When the theme background was changed from Dark to Light we did not see any obvious impact on power consumption as suspected. Furthermore, changing the accent colour to Yellow, cobalt, red and orange did not brought any increase on power consumption on Start menu (when measuring ~32 ±2mV). We are taking into consideration the fact of performing this test again in the future and re-evaluate it.

	Condition	Results
Theme Background (on Settings menu)	Dark	32 ±3mV
	Light	35 ±3mV

Table 8. Theme Background voltage drops tests results (average of 10 iterations)

When on Start menu the apps were flipping (the state when an apps can be deleted, resized or moved) the main consumption goes from 30mV drop to 62 ±3mV, i.e. the consumption is almost doubled when no apps flipping compared to "static" case (measurement of start menu, and one full page of apps only). Main contributors: Cpu&Ram, display, graphics, network interface and apps. Running fewer applications can also save power when using LCD, *Carroll at al* in [1] shows that the most energy intense component are the display (with 400mW including LCD panel, touch-screen, backlight and graphics accelerator).

3.2 Micro-benchmarks

As mentioned before, we used micro-benchmarks to determine the contribution to overall power from various system components.

3.2.1 CPU and RAM

The CPU utilization can have large effects on the overall rate of current discharge depending on the frequency at which the CPU is maintaining. The higher the CPU frequency is, the greater the rate of current discharge. To all authors is interesting the establishing of the power consumption of Cpu and Ram, rather than making comparisons between different platforms' algorithms. On this stage of our tests we were not able to distinguish the power consumed directly from CPU and RAM. We can highlight main tests done for this purpose from many authors where they measured the average CPU and RAM power at fixed core frequencies of 100MHz and 400 MHz. For Perrucci et al in [4] loading the CPU from 0% to 100% with various tasks during the measurements the power drain increase rapidly.

3.2.2 Network

In this benchmark we stressed the main networking components of the device: 2G GSM and 3G UMTS Network Interfaces, WiFi and Bluetooth. Testing for different signal strength or in fast moving is not taken under consideration for this phase. However, 3G+ technologies become more energy efficient when transmit large volumes of data. The work by Balasubramanian *et al.* in [3] goes a bit deeper in the analysis of IEEE 802.11 standards and cellular networks (using exclusively NEP as measurement tool). They found that cellular networks present high tail energy overhead by staying in high energy-states after completing a data transfer. This effect is lower in GSM than in 3G networks.

3.2.2.1 2G and 3G Network Interfaces

On 3G UMTS wireless communication power footprint at the end user is not only affected by the amount of data transferred, but also affected by the received signal strength RSSI and the static network parameters configured at the network end (operator or RNC). An extra test is made to measure the voltage drop when manually switching from both networks 2G and 3G (assume forced or manual HO). A spike of high voltage drop is seen for few milliseconds and it can be dedicated to network change process (changing manually from setting menu), and the rest to scanning the network.

Tests	Data	RSSI 2G / 3G	Results
2G → 3G	OFF	(-67 / -60dBm)	~33 → 60 ±15mV spike to 130mV
3G → 2G	OFF	(-67 / -60dBm)	~32 → 70 ±10mV spike to 140mV

Table 9. Manual switching from 2G to and from 3G voltage drop tests results (average of 5 iterations)

Measuring the RLC data buffers and timers for discovering the channels state transitions will be considered in the future. Main contributors: Cpu&Ram, Network IF, display (including touch), graphics, apps in background.

3.2.2.2 WiFi

Authors at [13] had studied cases of standby average power consumption with the phone and an idle WiFi connection. In our test case we just measured the power drain during WiFi interface state change from off to on and alsowhen scanning for any available access point. Interestingly in this case is that WiFi actually has the lowest idle power cost, followed by 2G and then 3G. Only the consumption of the wireless networking hardware because the baseline power consumption of the phone (CPU, backlight etc) has been subtracted from the trace. Although the spikes are more frequent (every 100 ms, corresponding to receiving base access points), the base power is lower than maintaining a connection to the cellular network. Main contributors: Cpu&Ram, WiFi Interface, network, display & backlight, graphics.

	Conditions	Results
WiFi	OFF to ON	32→ 70 ±5mV
	Scanning	32→ 50 ±5mV

Table 10. WiFi use voltage drop test results (average of 10 iterations)

3.2.2.3 Bluetooth

This benchmark is designed to measure power breakdown for Bluetooth interface by being sending a song of 6.18 MB and detail info can be found at [8] and receiving a video of 2MB size with a Nokia E6 phone as per table below. During the tests the background is dark, Brightness low and Network is 2G with no data available (no update is allowed during the tests) and WiFi interface set to off.

Bluetooth Tests	Distance	Results
connecting	10cm	31 → 45mV
receiving 2MB video	10cm	30 → 60 ±5mV
	3m	32 → 55 ±5mV
sending 6.18MB songs	10m	33 → 70 ±5mV
	3m	33 → 95 ±5mV

Table 11. Bluetooth send-receive voltage drop test results (average of 10 iterations)

Considerably during tests, it is noticed that power consumption increase when distance increases from 10cm to 3m and as well time is increased for transferring the files in both directions. For example sending the song from DuT got almost 100 second for 3m distance and for 10cm distance ~ 45 seconds. Also testing on this DuT (Nokia Lumia) shows that receiving consumes less than sending a file as per table figures. Furthermore Bluetooth transferring can continue on background showing to us real impact (as all other components except Bluetooth and Cpu&Ram will be “disabled”). Main contributors: Cpu & Ram, Bluetooth if, Network, display & backlight, graphics.

3.3 Macro-benchmarks

Here are tests used to determine power consumption under a number of typical usage scenarios of a Nokia Lumia. Specifically we examined audio and video playback, Sms text messaging, voice calls, emailing, web browsing, You Tube and internet sharing.

3.3.1 Audio playback

This benchmark is designed to measure power in a system being used as a portable media player. During the test background is dark, Brightness is low and Network is 2G with no data available (not allowing any update during the test), WiFi IF is set to off. The sample music is a 6.18 MB, 214-second stereo 44.1 kHz MP3 [8], with the output to phone and also a pair of stereo headphones used. Nokia Lumia *Music & Videos* application is used while the MP3 file is loaded from the main internal memory of the phone. The measurements are taken with main screen on the media player and when the backlight off (which is representative of the typical case of someone listening to music or podcasts while carrying the phone in their pocket). Table 12 shows the power breakdown (voltage drops) for this benchmark at medium volume 15 and maximum volume 30 and averaged over 10 iterations. Before testing the average voltage drop is ~30mV when the application is running and ready for starting the file, and during testing the results show the audio subsystem (amplifier and codec) causing a voltage drop contribution between 10 to 20mV. So slight contribution is seen and it can be noted saving power in headphone when using them for a long time, but not for short period.

	Volume	Results
Audio 6.18MB MP3, 214 sec	medium (15)	30 → 45 ±10mV (screen on) 30 → 42 ±6mV (screen on & headphones) **3** → 15 ±7mV (backlight off) **3** → 11 ±5mV (backlight off & headphones)
	max (30)	32 → 55 ±10mV (screen on) 32 → 43 ±7mV (screen on & headphones) **3** → 24 ±7mV (backlight off) **3** → 12 ±6mV (backlight off & headphones)

Table 12. Audio playback voltage drop tests results (average of 10 iterations)

While the MP3 file loading from SD card, the cost of doing so is negligible at < 2% of total power. Main contributors: Audio Amplifier + Codec, Cpu&Ram, Network IF or GSM, and Graphics.

3.3.2 Video playback

During the test, background is set to dark, brightness is set to low & medium and network is 2G with no data (not allowing any update during the measurements), WiFi set off, colour temperature & saturation at 1. In this benchmark we measured the power requirements for playing a video file by changing the Volume to medium and high and Brightness to low and medium. We used a 129 sec, 19.2 MB stereo Google video clip [9] with and without headphones, and played it Nokia Lumia *Music & Videos* application. The voltage drops averaged over 10 iterations is shown in Table 13. Network interface or GSM power is again included. Here the CPU should be the biggest consumer of power (other than backlight). The energy cost of loading the video from the SD card is negligible as studied from many other authors as well, with an average power of average power of 0.2% over the length of the benchmark.

	Volume	Results
Video 19.2MB MP4, 129 sec	medium (15)	30 → 65 ±6mV (B_low) 30 → 62 ±3mV (B_low & headphones) **49** → 82 ±4mV (B_medium) **49** → 82 ±2mV (B_medium & headphones)
	max (30)	30 → 68 ±10mV (B_low) & spike to 85mV 30 → 65 ±6mV (B_low & headphones) & spike to 75mV **49** → 88 ±10mV (B_medium) & spike to 103mV **49** → 84 ±4mV (B_medium & headphones)

Table13. Video playback voltage drop tests results (average of 10 iterations)

When volume is changed from medium 15 to maximum 30 there is seen only a slight increase in power consumptions (voltage drops increases), and significantly when the brightness change to medium the increase in voltage drops and variation is more visible and also spike of high drops are seen when the volume change rapidly on the video itself. Furthermore using or not the headphones for short time do not have a visible impact since the changes seen in consumption are really low as per table 13. Voltage drops contribution measured by testing is ~ 35 to 40mV. In reality loading online videos directly from YouTube for example can rapidly drown the power and it can result in an empty battery in few hours. Main contributors: Amplifier & codec, Network If or GSM, Cpu&Ram, audio, display, graphics and apps for video.

3.3.3 Text messaging, SMS

During the test, background is dark, brightness is low and network is 2G and 3G with no data (not allowing any update during the test), WiFi off, colour temperature & saturation is 1 and RSSI is measured through another Nokia E6 (*CellTrack91* application) with -65dBm or 5 bars on Lumia 625. We benchmarked the cost of sending an SMS by using a trace of real phone usage. This consists of loading the contacts application and selecting a contact, typing and sending a 20-character message (**a** letter in the case), then returning to the home screen including swiping up & down and right and left and also multi touching for writing the SMS and finding the contact. Normally in these tests we measure the cost of power consumption for GSM transaction an additional time till SMS is send and delivery is coming in few seconds. Here SMS delivery confirmation is allowed.

Network	Touch Nr	RSSI	Results	Time
2G	32 (12 + 20)	-57dBm	~ 35 mV→ 80 ±20mV touching ~ 65 ±10 mV during send & spike 90mV	~ 35 sec
3G	32 (12 + 20)	-52 & -65dBm	~ 35 mV→ 80 ±20mV touching ~ 80 ±10mV during send & spike 110mV	~ 35 sec

Table 14. SMS sending over both networks voltage drop tests results (average of 10 iterations)

From experimental results larger energy consumption is on 3G networks for text messaging (SMS) compared to 2G networks. The energy consumption of sending text messages increases linearly with the length of the message while the signal strength clearly affects the time required to transmit the message in both types of networks. During our tests signal level was really strong. Power consumed is again dominated by the display components (touching or swiping). Also Network itself impact for the RSSI value, message length for the time of sending. Main contributors are: Backlight, Network and its RSSI, Cpu&Ram, display and graphics .

3.3.4 Phone call

The same test condition like for SMS remains also for this benchmark. Table below shows the power consumption when making a GSM and 3G phone call (outgoing and incoming). The benchmark is trace-based, and we do not include loading the dialler application and searching for a number (excluding touching and swiping), but only dialling a number directly, and making a 25 sec call (for outgoing). The dialled device (Nokia E6) was configured to automatically accept the call after ring. Also is tested the incoming call to this Nokia Lumia and just monitoring status for 10 sec and measure the voltage drops. For many authors in case of voice services, using GSM requires less energy than UMTS networks. Controversy to this statement we have found that on GSM the call was more power consuming compared to 3G outgoing call (probably RSSI slight difference). Here we plan to perform tests again concluding on better approaches. Networking and its RSSI are the most significant ones and the rest have very little impact. Main contributors: Network and its RSSI, Backlight, Cpu&Ram, display, graphics.

Network	Data	Duration	RSSI	Direction	Average
2G	OFF	20 + 5 s	-65dBm	outgoing	~32 → 90 ±12mV when talking ~ 62 ± 5mV when no talks (due to DRX)
				incoming	~32 --> 105 ±5 mV
3G	OFF	20 + 5 s	-57dBm = 5 bars -71dBm = 4 bars	outgoing	~32 → 75 ±10mV when talking (no any change when RSSI decreases)
				incoming	~32 → 100 ±20mV

Table 15. Voice call over both networks voltage drop tests results (average of 10 iterations)

3.3.5 WiFi and Internet sharing

For this benchmark, we used Internet Explorer on PC and opening the Linkedin, Gmail and CNET web paging by browsing on them. On Gmail page we have uploading 3 times the song of 6.19MB MP3 and also the video of 19.2MB MP4 and we measured the cost of PC sending and receiving emails through Nokia Lumia Internet sharing option (3G and Wifi on). Phone and PC are ~ 30cm away from each others. During tests is observed that uploading is more power consuming (huge power drown or voltage drops) compare to downloading. Also uploading took more time than downloading as depending on customer features and network conditions (like BH or any other user profile set by operator). Download and upload are considered when mobile screen is locked so display, LCD and graphics are excluded from consumption during voltage drop measurements.

	Conditions	Distance	Results
3G and WiFi ON (internet sharing ON)	Phone	30 cm	50 ±3mV Screen ON 20 ±3mV (locked Screen)
	PC connecting	30 cm	95 ±10mV & spike to 150 mV 63 ±3mV (locked Screen)
	Linkedin , CNET , Gmail	30 cm	65 → 75 ±5mV (locked screen idle)
	Downloading (6.18 MB song file) Downloading (19.2 MB s file)	30 cm	110 ±10mV
	Uploading (6.18 MB song file) Uploading(19.2 MB s video file)	30 cm	100 ±25 mV & spike to 150 mV (locked screen) 150 ±30 mV & spike to 210 mV (locked screen)

Table 16. 3G & Internet sharing voltage drop tests results (average of 10 iterations)

Main contributors: WiFi, Cpu&Ram, Networks & RSSI, display and backlight, graphics.

3.3.6 Emailing

For this benchmark, we used Nokia Lumia Gmail application tiled on phone start screen to measure the cost of synchronizing, sending and receiving emails, by attaching or downloading a video, music or photos. In the download test we used a video attached on the inbox email and for uploading just sent a new email with uploading (attaching) few many photos. Measurement results are as per table 17. Applications requiring updates (background allowed) on start screen are removed and blocked for not allowing them to be updated during our tests. 3G network dominates more in voltage drops here for download and upload but it can differentiate for its speed.

Network	RSSI	Results
2G EDGE	-67dBm	32 → 70 ±10mV navigating 32→ 85 ±20mV synchronizing (screen on) 32 → 90 ±20mV downloading (screen on) 32→ 100 ±15mV uploading (screen on)
3G HSPA+	-69dBm	32 → 75 ±10mV navigating 32 → 100 ±20 mV synchronizing (screen on) 32 → 100 ±25mV downloading (screen on) 32 → 110 ±25mV uploading (screen on)

Table 17. Email, download/upload on 2G and 3G voltage drop tests results (average of 10 iterations)

Main contributors: Backlight, Network If & RSSI, Cpu&Ram, display (touching and swipe) and graphics.

3.3.7 YouTube

On this benchmark we test the voltage drops when playing a video directly from YouTube application on the phone. Volume is at middle and colour setting at 1, brightness low. Email and other apps requiring data connection for synchronization are removed and their background tasked are blocked for not allowing such update interfering with our voltage drops measurements. On 2G since the data speed is not so promising always the phone remained on buffering. While on 3G the data are much faster and no buffering seen during time we tested (tests done not during BH period). Main contributors: Network & RSSI, Cpu&Ram, display (touching and swipe) and graphics.

Network	RSSI	Results
2G EDGE	-63 dBm	32 → 90 ±15mV during & when apps opened 32→ 85 ±20mV when trying to open the video (buffering)
3G HSPA+	-57 dBm	33 →140 ±20mV during opening YouTube apps 33→ 115 ±15mV when playing the video

Table 18. YouTube video playing on 2G and 3G voltage drop tests results (average of 10 iterations)

3.3.8 Web browsing

On our last benchmark we measured the power consumption for a web-browsing workload using both 2G EDGE and 3G HSPA+ connections. Here we used the Nokia Lumia Bing tool to navigate on web, so the benchmark is consisted of loading the browser application, selecting a bookmarked web site and browsing several pages. We used the Balkanweb website to navigate for a while. 3G clearly seems more power consuming during browsing and navigating inside the web page as per table 19 results.

Network	RSSI	Results
2G EDGE	-50dBm	32 → 90 ±15mV during & when apps opened 32→110 ±15mV when searching and opening a webpage
3G HSPA+	-57dBm	32 → 120 ±15mV during opening Bing apps 32→ 110 ±15mV when searching 32→ 170 ±20mV when opening a web and navigating

Table 19. Web browsing on 2G and 3G voltage drop test results (average of 10 iterations)

Main contributors: Network & RSSI, Cpu&Ram, display (touching and swipe), graphics, apps for browsing.

4 Analysis and Conclusions

Non-linearity of subsystems it's common knowledge that most subsystems are nonlinear, such that the power consumed at a given time interval strongly depends on the state in which the device was operating. For example, different pixel colour and brightness states have a strong effect on the amount of power consumed by the display subsystem. Our results aims to show the majority of power consumption (voltage drops) that can be attributed to the Network Interfaces modules and the display, including the IPS LCD panel, touchscreen, swiping, and the backlight. In all tests except the 3G and 2G intensive benchmarks, the brightness of the backlight is one of the most critical factors in determining power consumption. However, this largely depends on the user's brightness preference. Audio or Video playback are mostly influenced from file type (length, MP3 or MP4, codec's, or sample rates and so on), Brightness rather than audio volumes. Furthermore, the effects of display content and brightness on power consumption are more tightly coupled. Audio displayed static power consumption during the measurements. During testing network interfaces for a phone call the GSM module consumes less power when DTX & DRX features are enabled on customer base stations. Merely maintaining a connection with the network consumes a significant fraction of total power. Dimming the backlight during a call, are clearly good policy for saving power even with the large 2G and 3G consumption. Navigating on any webpage, videos playback on YouTube and downloading or uploading directly from the phone are ones of the more data-intensive uses of mobile devices and resulting in high power consuming. When phone is on Idle and network is 2G or 3G power consumption seems really low as it can be contributed to OS where components are put into low-power sleep modes and this confirms the Lumia maximum standby time. For many authors RAM, audio and SD have little effect on the power consumption of the device. Having more applications on Main menu with Background allowed and with live tiling can rapidly increase the battery consumption. The below figures present voltage drops for almost all cases we benchmarked.

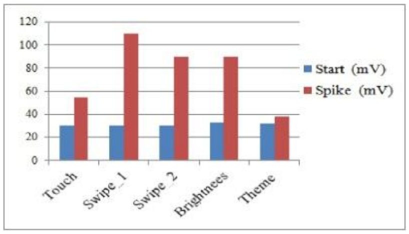

Fig. 3. Baselines Cases

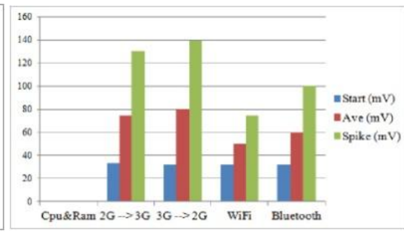

Fig. 4. Micro benchmarks

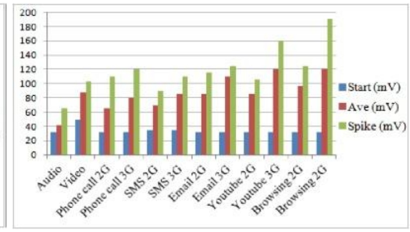

Fig. 5. Macro benchmarks

5 Limitations to our work and Future work

One of the main limitations to our work is that we were not able to provide accurate power consuming based on low level analysis of mobile phone, which requires handling use of logging of specific component file drivers or specific process analysis for mobile OS, which is not allowed for Windows mobile OS or at least this OS version we have. So all our tests are based on measuring the voltage drops directly after the voltage drop on resistor stabilizes at an average value which is around 30 to 32mV when screen is on and ~3mV when screen is locked. In the future our main aim is to build a battery saver application and perform again testing and compare them. We plan to do a better evaluation of each component contribution in an automated way by loading test scripts and running as background application.

References

1. A. Carroll and G. Heiser: An analysis of power consumption in a smartphone, in *Proceedings of the 2010USENIX conference on USENIX annual technical conference, ser. USENIXATC'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 21–21*
2. A. Rice and S. Hay: Measuring mobile phone energy consumption for 802.11 wireless networking, in *Pervasive Computing and Communications (PerCom), 2010*
3. N. Balasubramanian, A. Balasubramanian, and A.Venkataramani: "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, ser.IMC '09. New York, NY, USA: ACM, 2009, pp. 280–293*
4. G. P. Perrucci, F. H. P. Fitzek, G. Sasso, W. Kellerer, and J. Widmer: On the impact of 2G and 3G network usage for mobile phones' battery life, in *Wireless Conference, 2009. EW 2009. European, 2009, pp. 255–259*
5. A. Rice and S. Hay: Decomposing power measurements for mobile devices, in *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference, April 2 2010, pp. 70 –78*.
6. GDM-8246 tool, *http://www.gwinstek.com/en/product/productdetail.aspx?pid=39&mid=80&id=198.*
7. Nokia Lumia 625 HW and SW specification, *http://www.nokia.com/global/products/phone/lumia625/*
8. MP3 song file of 6.18 MB size, can be found under *http://www.KerkojMuzik.com*
9. Google Chrome Speed Tests video file 19.2MB, *http://www.youtube.com/watch?v=nCgQDjiotG0*.
10. N. Jacobsson, M. Lopes, Sh. Islam. Energy Consumption In Wireless Networks, *Progress Report.*
11. Nokia Lumia service manual,in *http://download.support.nokia.com/ncss/PUBLIC/en_IN/webpdf/100000336186/Nokia_Lumia_625_UG_en_GB*
12. A.Abdelmotalib, Zh.Wu: Power Consumption in Smartphones (Hardware Behaviourism), in *IJCSI International Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol. 9, Issue 3, No 3, May 2012*
13. A. Rice and S. Hay: Measuring mobile phone energy consumption for 802.11 wireless networking, in *Pervasive Computing and Communications,article in Press, www.elsevier.com/locate/pmc, July 2010*
14. R.Murmuria, J.Medsger, A.Stavrou and J.M. Voas: Mobile Application and Device Power Usage Measurements, *in SERE 2012*
15 H.Haverinen, J.Siren and P.Eronen: Energy Consumption of Always-On Applications in WCDMA Networks, *in 2007*