Characterize Energy Impact of Concurrent Network-Intensive Applications on Mobile Platforms

Zhonghong Ou
Dept. of Computer Science and Engineering
Aalto University, Finland
zhonghong.ou@aalto.fi

Shichao Dong
Dept. of Computer Science and Engineering
Aalto University, Finland
shichao.dong@aalto.fi

Jiang Dong
Dept. of Computer Science and Engineering
Aalto University, Finland
jiang.dong@aalto.fi

Jukka K. Nurminen
Dept. of Computer Science and Engineering
Aalto University, Finland
jukka.k.nurminen@aalto.fi

Antti Ylä-Jääski
Dept. of Computer Science and Engineering
Aalto University, Finland
antti.yla-jaaski@aalto.fi

Ren Wang
Circuits and Systems Lab
Intel Labs
Portland, USA
ren.wang@intel.com

ABSTRACT

The cellular network bandwidth increases significantly in the past few years, stimulated by many popular network-intensive applications, such as video streaming and cloud storage usages. Meanwhile, more and more users enjoy the multitasking feature of mobile devices and concurrently run a number of applications. Given these two trends and the fact that extended battery life remains to be a critical factor for small form factor devices, e.g. smartphones and tablets, it is imperative to understand the energy impact of multiple applications running concurrently on such platforms.

In this paper, we characterize and understand the energy and performance impact of concurrent applications via a comprehensive set of carefully designed experiments. Specifically, we focus on network-intensive applications since most usage models today are driven by always-on communication activities. We make several significant contributions to shed light on understanding the energy behavior of concurrent applications. Firstly, we find out that running multiple network-intensive applications concurrently can significantly improve energy efficiency, up to 2.2X compared to running them separately. Secondly, we observe that power consumption from CPU and System on Chip (SoC) are the primary culprits of power dynamic for network-intensive applications; while communication components, including Network Interface Card (NIC), poses very little power consumption variation with different throughput. Thirdly, we investigate, in detail, the significant impact of signal strength on the energy consumption and throughput performance. Our findings and analysis can be applied to provide helpful guidance for a wide range of research aiming to optimize mobile device energy efficiency, e.g. transmission scheduling and protocol design in cellular networks.

Categories and Subject Descriptors

C.2.1 [**Computer Communications Networks**]: Network Architecture and Design—*Wireless communication*; B.8.0 [**Hardware**]: Performance and Reliability—*General*

Keywords

Energy consumption; concurrency; signal strength; network-intensive applications; mobile platform

1. INTRODUCTION

Energy consumption of mobile devices remains to be an important factor for various parties, ranging from device manufacturers, mobile operators, to end users. Battery capacity has been increasing in the past few years; battery life of mobile devices, however, is not catching up proportionally. The reason lies in the recent tremendous growth in network deployment and energy-hungry applications enjoyed by mobile users, and demand from these applications surpasses battery capacity improvement. In the power breakdown of mobile handsets, wireless communication components, including Wi-Fi and cellular network interfaces, account for a prominent part of the overall power consumption [3].

On the other hand, network throughput of cellular networks has been increasing significantly in the recent years. Compared with Wideband Code Division Multiple Access (WCDMA) network, High-Speed Packet Access (HSPA) network can provide significantly higher network throughput and shorter latency, e.g. from 384 Kbps (kilo bits per second) to several Mbps (mega bits per second). [4] Long-Term Evolution (LTE) network can drive the bandwidth and latency improvement even further. The high bandwidth can enable certain usage scenarios that are not possible before, e.g., watching a Youtube video while syncing a Dorpbox file at the background concurrently.

Given these two trends, it is fundamentally important to understand the energy characteristics of running multiple network-driven applications concurrently. For example,

MobiArch'13, October 4, 2013, Miami, Florida, USA
Copyright 2013 ACM 978-1-4503-2366-6/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505906.2505909.

what is the overall power consumption of having two TCP downloading concurrently compared with only one? Will they impact each other in terms of network throughput? What is the energy and performance impact when making cellular phone calls while performing data communications? In order to answer these questions, in this paper, we conduct a comprehensive set of carefully designed experiments in HSPA network, which is currently the most widely used cellular infrastructure for data communications [6], and analyze the results to gain insights.

We make several significant contributions in this paper, summarized as follows:

(1) By measuring and analyzing platform power and performance of concurrent workloads, we find out that running multiple network-intensive applications concurrently can significantly improve energy efficiency, up to 2.2X compared to running them separately. Furthermore, when download and upload operations are conducted concurrently, power consumption follows closely with upload alone scenario, while network throughput is only slightly impacted.

(2) We observe that power consumption of the communication components, including Network Interface Card (NIC), varies very little with different throughput and network condition; while power consumption from CPU and SoC plays the major role in terms of power dynamic for network-intensive applications;

(3) We investigate in detail the impact of signal strength on energy consumption and throughput performance. We find out that signal strength impacts the achievable throughput significantly, and in turn, plays a fundamental role in energy consumption of network driven applications.

The rest of the paper is structured as follows. Section 2 presents related literature and Section 3 describes the environment setup and evaluation methodology for measurements. Section 4 presents the detailed results and analysis on performance and energy from the experiments. Section 5 analyzes the impact of signal strength variations on power consumption and network throughput. In Section 6, we discuss the potentially optimizations guided by our findings and analysis, and in Section 7 we conclude the paper.

2. RELATED WORK

The existing research efforts of energy optimization in cellular networks can be divided into three categories: utilizing multiple radio interfaces, optimizing tail energy, and exploiting radio resource control.

Utilizing multiple radio interfaces: Provided that different radio interfaces present diverse radio characteristics and different ranges, Pering et al. [12] explored polices to switch among these interfaces to save energy. Similarly, Rahmati et al. [15], Nicholson et al. [11], and Hou et al. [5] proposed to leverage supplementary strength from Wi-Fi and cellular networks to optimize data transmission based on their respective channel conditions. Sharma et al. [16], on the other hand, proposed to utilize cellular radio links in the vicinity to set up Wi-Fi hotspots on the fly to provide affordable connectivity.

Optimizing tail energy: Balasubramanian et al.[1] observed that GSM and 3G networks pose a high tail energy overhead, and proposed a protocol named TailEnder to reduce energy consumption of common mobile applications. Based on a feature named fast dormancy, Qian et al. [13] proposed a tail optimization protocol to enable cooperation between handsets and radio access networks to eliminate tail energy. Similarly, Liu et al. [8] tried to schedule multiple transmissions to the tail time of other transmissions to alleviate tail energy.

Exploiting radio resource control: Yeh et al. [17] analyzed the impact of an inactivity timer on energy consumption and reconnection cost in CDMA2000 and WCDMA networks, and suggested proper parameters for the timer to optimize power consumption. Qian et al. [14] analyzed the state machine transitions of 3G network, and pinpointed the inefficiencies caused by interplay between smart phone applications and state machine behavior. Huang et al. [7] examined performance and power characteristics of 4G LTE network. They also evaluated the impact of LTE parameters on the energy usage of mobile handsets.

Different from these work, we focus on characterizing energy impact of running multiple network-intensive applications concurrently on mobile platforms, and analyze their impact on the overall power consumption.

3. ENVIRONMENT SETUP

Google Nexus S phone is used for the measurements. Various applications have been measured, including multiple TCP connections for uploading, downloading, VoIP (Skype), Youtube streaming, and different combinations from them. 95% confidence intervals are used throughout the paper where appropriate.

3.1 Power Consumption

We use Monsoon power monitor [10] to measure power consumption of mobile phones. The power monitor is a power supply for the mobile phones under test. It records current consumption and voltage supply (constant at 3.7V) per 0.2 ms (5000 times per second). By multiplying current with supply voltage, we can acquire the corresponding power consumption. In order to measure power consumption of mobile phones on the move (under varying signal strength condition), we use an Uninterruptible Power Supply (UPS) to supply the Monsoon power monitor, and then drive a car to different locations to acquire the desired signal environment.

3.2 TCP Throughput

As TCP traffic accounts for more than 80% of the overall mobile phone data traffic [9], we mainly focus on TCP traffic in this paper, and have developed two benchmarks, i.e. TCP_Down and TCP_Up, to measure TCP download and TCP upload throughput. Multiple TCP connections can be set up by the benchmark tools. For the TCP_Down benchmark, an Apache server is deployed as the server. The mobile phone under test downloads files from the server, throughput is calculated every 500 ms on the phone. Note that to avoid power consumption from file operations, instant throughput records are put into a vector in the memory. After downloading is completed, the vector is written into a file. For TCP_Up benchmark, the mobile phone under test uploads a file to a server, then throughput is calculated on the server side per 500 ms. Furthermore, we use Linux traffic control (**tc**) utility to achieve the desired throughput in Section 4.2.

3.3 Signal Strength

For tracking signal strength variations, we acquire the Arbitrary Strength Unit (ASU) value from Android OS once per 500 ms. ASU is an integer value ranging from 0 to 31.[1] Then by using the formula: $dBm = 2 \cdot ASU - 113$, we can acquire the signal strength power in Watts. Thus, dBm value is always odd in our measurements, which we will see in Section 5.

4. ENERGY IMPACT OF CONCURRENT APPLICATIONS

In this section, we analyze the impact of multiple network-intensive applications running concurrently on power consumption of mobile platforms. The measurements were conducted in an indoor office environment. Signal strength value centers around -65 dBm, and fluctuations are within 4 dBm.

4.1 Power Consumption vs. CPU Utilization

We first model the relationship between CPU utilization ratio and the associated power consumption of the platform. To achieve the desired level of CPU utilization ratio, we vary the time the CPU is awake and asleep. When the CPU is awake, it is performing mathematical summation in a loop until it falls asleep. Note that in this measurement, no networking activity is involved. The relationship of power consumption and CPU utilization ratio is illustrated in Fig. 1. From the figure, unsurprisingly, as the CPU utilization ratio increases, power consumption increases proportionally. The relationship can be best fitted by a linear function, as shown in Equation 1:

$$P(x) = 9.2 \cdot x + 160 \ (mW) \quad (1)$$

Interestingly though, is that the minimal CPU utilization ratio that we can acquire is around 5%, which is interpreted to 215 mW of power consumption. This phenomenon can be explained by the fact that when the CPU starts to be active, the associated components on the System on Chip (SoC) have to be activated at the same time, including memory controller, memory, and chipset. Power consumption from the CPU and these components constitute the base power for the whole platform. The smaller the base power accounts for, the better the energy-proportionality [2] of the processor is. Nevertheless, it is impossible to completely eliminate this base power. Given the fact, the problem becomes to how to amortize the base power. Furthermore, because our focus is network-intensive applications, it is straightforward to ask the similar question: Does the base power phenomenon also exist in networking component? In other words, does power consumption from communications components also reveal a large base? This inspires us to analyze the relationship among power consumption, network throughput, and CPU utilization ration, in the subsequent section.

4.2 Power Consumption vs. Throughput

As mentioned before, we use Linux utility **tc** to control network throughput. The relationship of power consumption vs. CPU utilization ratio, network throughput vs. CPU utilization ratio for download is depicted in Fig. 2, while for upload illustrated in Fig. 3. Note that only one TCP

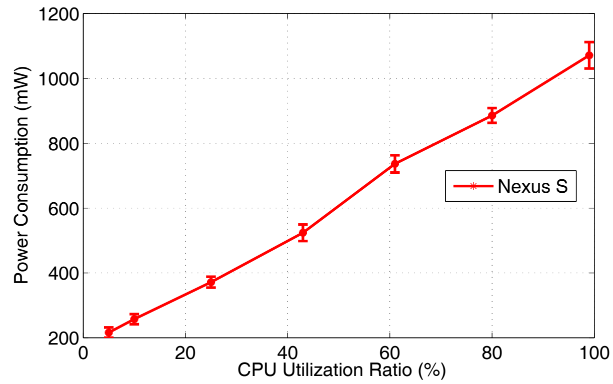

Figure 1: Power consumption vs. CPU utilization ratio

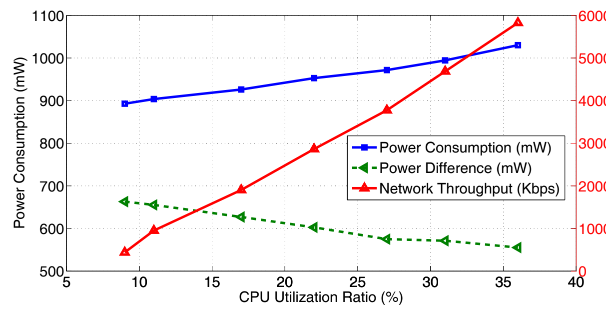

Figure 2: Download power consumption vs. CPU utilization ratio

connection is used in this experiment. Furthermore, it is noteworthy that the independent variable in these two figures is network throughput. To be consistent with Fig. 1 in terms of presentation, we make CPU utilization ratio as the x-axis. Nevertheless, the overall tendency is the same. In these two figures, the red curve stands for network throughput (the righthand y-axis), while the other two curves are related to power (the lefthand y-axis). As illustrated in the figures, expectedly, when the network throughput increases, the associated power consumption and CPU utilization ratios increase as well. The power dynamic (from 1000 mW to 1500 mW) for upload is much larger than that for download (from 900 mW to 1050 mW). This is because in uploading operation, power amplifier (PA) contributes a significant fraction of the overall power consumption of a mobile handset. When more data have to be transmitted, more power has to be spent on the PA.

To provision a coarse-grained break down of the overall power consumption, we use Equation 1 to approximately represent CPU and SoC power consumption. By deducting this power from the overall power consumption (depicted as blue curves in Fig. 2 and Fig. 3), we can acquire the power consumed by communication components (shown as green curves). Similar approach is also used in Section 4.4 to acquire power difference(cf. $power\Delta$ column in Table 1). From the green curves, it is interesting to see that power consumption from networking-related components is very stable. In the downloading scenario (cf. Fig. 2), the power share from

[1] ASU value of 99 means unknown or undetectable signal.

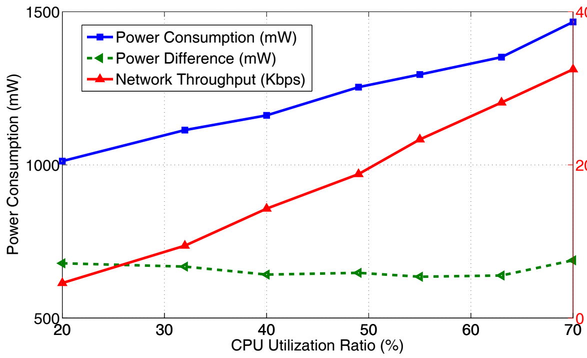

Figure 3: Upload power consumption vs. CPU utilization ratio

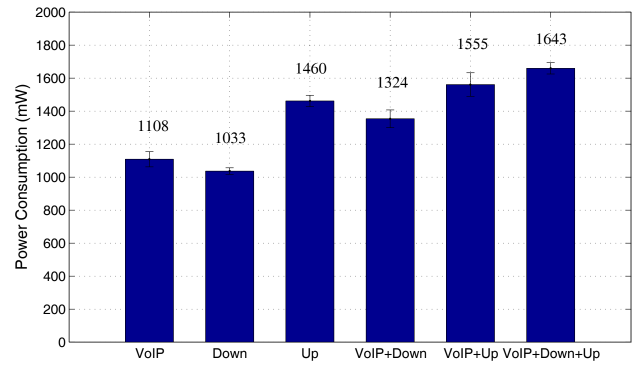

Figure 4: Power consumption of VoIP + download + upload

networking components is even decreasing slightly. One conclusion can be made from the figures: for network-intensive applications, CPU and SoC power consumption is the primary culprit; while power consumption from communication components are relatively stable.

4.3 Multiple Concurrent TCP Connections for the Same Type of Application

Before we analyze multiple diversified applications, it is beneficial to understand the energy behaviors of multiple TCP connections for the same type of application. The results are shown in Table 1, i.e. *Down_local*, *Down_EC2*, and *Up_local*. The numbers following the application name indicate how many instances of the application are running concurrently. Note that to analyze the potential impact from wide area network, we also conducted TCP downloading experiments from Amazon EC2 platform. In these measurements, multiple instances were run to host the Apache servers (each instance hosts one Apache server). Then the mobile client (i.e., Google Nexus S) sets up one TCP connection with each of these Apache servers. From the table, it is interesting to notice that because EC2 limits the wide area traffic from a single instance, adding a small amount of power $(1043 - 968 = 85mW)$, the network throughput can be doubled. It is noteworthy that EC2 represents a typical type of application, where network throughput is throttled by the server. Server-controlled throttling is widely used in Internet applications, one well-known example is Youtube. For these applications, energy consumption can easily be reduced by half, as the bottleneck is from the server side rather than client side. From these experiments, we can conclude that the number of TCP connections does not pose a significant impact on power consumption. The overall power consumption is primarily impacted by the network throughput.

4.4 Multiple Concurrent Applications

Given the fact that power consumption is mainly impacted by the overall network throughput, without loss of generality, we use a single local TCP connection in the subsequent sections. Power consumption for the combination of VoIP, download, and upload is depicted in Fig. 4. From the figure, it is clearly seen that energy saving from running multiple applications concurrently is multiple fold. For example, compared with running VoIP and downloading separately, energy saving from running VoIP and downloading concurrently is $1 - 1324/(1108 + 1033) = 38.16\%$. Namely, energy efficiency improvement is $(1108 + 1033)/1324 = 1.6X$. Similar energy saving can be achieved by running VoIP and uploading concurrently. Drive it to an extreme, if we run VoIP, downloading, and uploading all together, the energy efficiency improvement is $(1108{+}1460{+}1033)/1643 = \mathbf{2.2X}$. On the other hand, as demonstrated in Table 1, network throughput for downloading and uploading is only slightly decreased. This observation is consistent with other combinations, e.g. Youtube + download + upload (cf. Table 1), and cellular phone call + download + upload (not shown because of space limit).[2]

Another interesting phenomenon is from the column $Power\Delta$, which is acquired by deducting CPU power (from Equation 1) from the overall platform power consumption. For the downloading involved activities (labeled as *), the power difference is centering around 530 mW; while for uploading involved activities (labeled as $\uparrow$), the power difference is fluctuating around 700 mW. This confirms our conclusion in Section 4.2 that CPU/SoC is the primary culprit of power consumption in network-intensive applications, whilst power consumption from communications components stays relatively stable.

5. IMPACT OF SIGNAL STRENGTH

In the previous section, we analyze power consumption in a fixed environment, i.e. at a fixed signal strength level. In this section, we extend the analysis to cover the whole range of signal strength. Without loss of generality, we only measure download, upload, and *combo* scenarios. *Comb* means conducting uploading and downloading concurrently.

The impact of signal strength on network throughput is depicted in Fig. 5, and impact on power consumption is illustrated in Fig. 6. To provide a comparative study, TCP throughput and power consumption from download alone

[2]Note that for Youtube streaming applications, the screen was kept on (otherwise Youtube does noes stream); while for the other experiments, the screen of the phone was kept dark. Power consumption from the display is 434 mW, which is deducted from the original reading. Therefore, because of the deduction, the power shown in Table 1 for Youtube is slightly underrated.

Table 1: **Power consumption of running multiple applications concurrently**

Combination	Power(mW)	Throughput (Kbps)	CPU (%)	$Power\Delta$ (mW)
Youtube	1008(±24)	604(±15)	56%(±3%)	332
VoIP	1108(±18)	30(±1)	56%(±2%)	432
Down_local(1)	1033(±25)	6050(±58)	36%(±1%)	541(∗)
Down_local(2)	1066(±26)	6180(±65)	39%(±2%)	547(∗)
Down_EC2(1)	958(±20)	2890(±43)	24%(±2%)	577(∗)
Down_EC2(2)	1043(±22)	5880(±48)	37%(±2%)	542(∗)
Down_EC2(3)	1063(±21)	6025(±45)	42%(±2%)	517(∗)
Down_EC2(4)	1070(±23)	6080(±50)	44%(±2%)	505(∗)
Up_local(1)	1460(±21)	3150(±43)	61%(±5%)	738(↑)
Up_local(2)	1480(±24)	3210(±48)	64%(±5%)	731(↑)
Youtube+Down_local	1160(±25)	5718(±65)	53%(±4%)	512(∗)
Youtube+Up_local	1506(±30)	2829(±56)	71%(±5%)	696(↑)
Youtube+Up_local + Down_local	1560(±35)	2372(Up)+4750(Down)	77%(±5%)	691(↑)
VoIP+Down_local	1324(±25)	5730(±76)	63%(±3%)	584(∗)
VoIP+Up_local	1555(±45)	2731(±65)	72%(±5%)	732(↑)
VoIP+Up_local + Down_local	1643(±51)	2420(Up)+4578(Down)	78%(±3%)	765(↑)

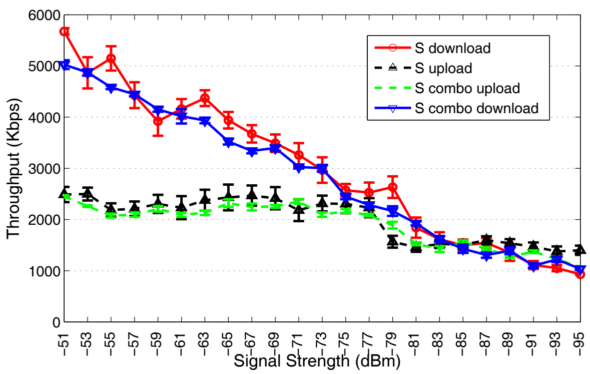

Figure 5: Signal strength vs. throughput

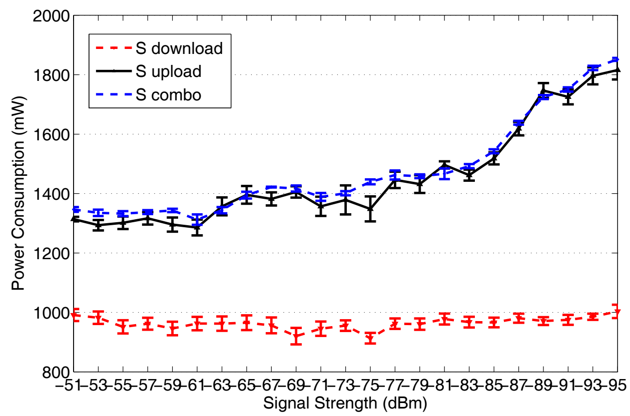

Figure 6: Signal strength vs. power

and upload alone scenarios are also depicted in the respective figures. Figure 5 demonstrates that downlink and uplink transmission in *combo* scenario are slightly impacted compared with separate transmissions. The impact of *combo transmission* on downlink throughput is bigger compared with on uplink transmission. The reduced throughput for download transmission is within 1 Mbps. While for upload transmission, the impact from *combo* transmission is very small.

Power consumption for *combo* transmission, on the other hand, demonstrates interesting trends. The overall tendency of *combo* power consumption follows very closely with upload alone scenario. This can be explained by the fact that when upload transmission is started, common components for upload and download transmissions, e.g. baseband processing units, are triggered. Baseline power consumption constitutes a large fraction of the overall power consumption for the common components. Furthermore, power amplifier component, which is unique for uplink transmission, poses a large power dynamic. Thus, upload power consumption dominates the overall *combo* power consumption. A takeaway message from this observation is that mobile applications can take advantage of uploading periods as much as possible to schedule downloading tasks concurrently. In this way, download comes almost free in terms of power consumption, whilst network throughput is barely impacted.

6. IMPLICATIONS AND DISCUSSION

In this section, we present the implications of our observations to energy optimization in cellular networks, and then discuss the limitations of our work.

6.1 Implications

The observations we make in this paper have several important implications for energy optimization research. Firstly, because power consumption from communication components pose a relatively small dynamic, while CPU/SoC power retains a large dynamic in network-intensive applications, more focus should be put on CPU/SoC components for energy optimization. For example, utilizing interrupt bonding to lower the frequency CPU getting interrupted by incoming packets is one potential approach to reduce CPU power consumption. Secondly, for delay-tolerant applications, the triggering events can be based on cellular voice call, VoIP etc, real-time applications. Because power consumption from these real-time applications are inevitable.

Thirdly, for downloading applications that can tolerate certain level of delay, e.g. email sync, they should be scheduled to utilize uploading periods from other applications as long as possible.

6.2 Discussion

The phone model (Google Nexus S) we used to conduct experiments in this paper is a few years old. Newer phone models, e.g. Samsung Galaxy SIII, might have used advanced features to lower the base power consumption. As a matter of fact, we repeated some of the measurements in this paper using Galaxy SIII. We found out that the base power for SIII has been significantly decreased to tens of milliwatts, compared with around 200 mW for Nexus S. Nevertheless, the observations we made in this paper are applicable to SIII. For example, running several network-driven applications concurrently can result in multiple fold of energy saving in SIII. This makes the observations of our paper valuable for potentially a large range of mobile platforms. Provided that commercial 4G LTE networks are rolling out fast in the recent years, and that the available network bandwidth from LTE networks are much higher than previous generations of cellular networks, the findings we observe in this paper demonstrate good indication to optimize energy-efficiency of mobile platforms in the future.

7. CONCLUSION AND FUTURE WORK

In this paper, we investigated the energy impact of running multiple applications concurrently on mobile platforms. Through comprehensive measurements and analysis, we made several important observations. Firstly, we found out that by scheduling multiple applications concurrently, the energy efficiency improvement is significant, reaching up to 2.2X. Secondly, we observed that with network-intensive applications, CPU/SoC is the primary culprit for power consumption dynamics, while communication components pose very small variation in terms of power consumption. Thirdly, we observed and analyzed the significant impact signal strength poses on the energy consumption of network driven applications. These observations pose good indication of optimizing power consumption of mobile platforms in cellular networks. In the future, we plan to apply a hybrid scheduling approach to achieve optimal energy saving for mobile platforms, i.e. coarse-grained scheme to schedule multiple applications concurrently, and fine-grained approach to schedule network traffic at packet-level.

8. ACKNOWLEDGMENTS

The research conducted in this paper has been funded by the Finnish funding agency for technology and innovation (Tekes) in Massive Scale Machine-to-Machine Service (MAMMotH) project (Dnro 820/31/2011), and by the Academy of Finland under grant number 253860.

9. REFERENCES

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the IMC '09*, pages 280–293, 2009.

[2] L. Barroso and U. Holzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.

[3] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *USENIX ATC'10*, pages 21–21, 2010.

[4] H. Holma and A. Toskala, editors. *WCDMA for UMTS: HSPA Evolution and LTE*. John Wiley & Sons Ltd., West Sussex, United Kingdom, fifth edition, 2010.

[5] X. Hou, P. Deshpande, and S. Das. Moving bits from 3g to metro-scale wifi for vehicular network access: An integrated transport layer solution. In *ICNP '11*, pages 353 –362, 2011.

[6] Global mobile market share. `http://www.informatandm.com/about/wcis/`.

[7] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *MobiSys '12*, pages 225–238, 2012.

[8] H. Liu, Y. Zhang, and Y. Zhou. Tailtheft: leveraging the wasted time for saving energy in cellular communications. In *MobiArch '11*, pages 31–36, 2011.

[9] G. Maier, F. Schneider, and A. Feldmann. A first look at mobile hand-held device traffic. In *PAM '10*, pages 161–170, 2010.

[10] Monsoon Power Monitor. `http://www.msoon.com/LabEquipment/PowerMonitor/`.

[11] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom '08*, pages 46–57, 2008.

[12] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *MobiSys '06*, pages 220–232, 2006.

[13] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck. Top: Tail optimization protocol for cellular radio resource allocation. In *ICNP '10*, pages 285 –294, 2010.

[14] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3g networks. In *IMC '10*, pages 137–150, 2010.

[15] A. Rahmati and L. Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In *MobiSys '07*, pages 165–178, 2007.

[16] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding. Cool-tether: energy efficient on-the-fly wifi hot-spots using mobile phones. In *CoNEXT '09*, pages 109–120, 2009.

[17] J.-H. Yeh, J.-C. Chen, and C.-C. Lee. Comparative analysis of energy-saving techniques in 3gpp and 3gpp2 systems. *Vehicular Technology, IEEE Transactions on*, 58(1):432 –448, jan. 2009.