



Copyright © IEEE.
Citation for the published paper:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of BTH's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Catching the Download Train: Energy-efficient File Downloading On Smartphones

Selim Ickin

Blekinge Institute of Technology
Sweden

selim.ickin@bth.se

Thomas Zinner

University of Wurzburg
Germany

zinner@informatik.uni-wuerzburg.de

Katarzyna Wac

University of Geneva
Switzerland

katarzyna.wac@unige.ch

Markus Fiedler

Blekinge Institute of Technology
Sweden

markus.fiedler@bth.se

Abstract—The most energy-consuming applications in battery life-constrained smartphones are the ones that comprise data transmission, especially via the 3G interface. Scheduling download activities on smartphones is especially necessary, if there are multiple asynchronous downloads scattered over a long duration. The latter scenario highly increases the energy consumption of smartphones. In this paper, we investigate energy consumption with the focus on file downloading while scheduling multiple file downloads in two scenarios: serialized and parallel. We repeat the experiments on a single smartphone via its 3G and also via WiFi tethering via another smartphone. We assess the performance of the two scenarios via measurement of power consumption and corresponding download duration in a realistic environment.

I. INTRODUCTION

Smartphones are used in everyday life anywhere and anytime, as they have become incubators for ubiquitous computing. The limitations of the battery, however, restrict the usage of smartphones and are seen as a possible drawback stopping the growth of mobile computing and thus prohibiting its widespread adaptation [1]. In parallel, as the battery life constrains the service time of the applications, energy becomes one of the most important influential factors on the user perceived Quality of Experience (QoE) [2]. Most of the energy is consumed on a smartphone while sending and receiving data over the network, with the clear domination of 3G data module's power consumption [3].

The amount of data exchanged in-between smartphones and corresponding servers, the used access technology, as well as the timing of data transmission have a significant impact on the overall energy consumption of the smartphone. While using the cellular interface, the smartphone's cellular data module oscillates in-between Radio Resource Control (RRC) states. Low power consuming state, *e.g.*, IDLE, provides basic connectivity. If the data transmission rate increases, RRC state shifts to rather higher power consuming states such as Dedicated Channel (DCH) or Forward Access Channel (FACH) [4]. The oscillation of RRC states are basically caused by the asynchronous data activities of different applications over the cellular data interface. During the RRC state transitions, also due to the necessary signaling overhead, additional transition delays are induced. These additional delays increases the total energy consumption. In addition, during the active (DCH) state of RRC, the available network bandwidth is not fully utilized due to rather low volume data traffic per application. In order to avoid the high number of oscillations amongst the RRC states, it is important to schedule the network data activities of

different applications. This is done by scheduling the network data activities such that the RRC state is kept IDLE for a longer period of time, but it is utilized as much as possible while it is fully active, *e.g.*, in DCH state. The scheduling can be done by delaying particular network activities, and performing/clumping them with the late ones. This way, both the overall energy consumption and data transmission duration can be reduced. Hence, it is appealing to transmit data from multiple applications simultaneously and thus reducing the number of state transitions. Multiple Transport Control Protocol (TCP)-based data flows can be transmitted in a serialized or in a parallel manner.

In this paper, we focus on large size file downloading (*e.g.*, downloading videos/documents for later offline usage) on smartphones. We first measure the energy consumption and the download duration when multiple files are downloaded on a smartphone via 3G asynchronously, and then when the downloads are scheduled (*e.g.*, parallel or serialized). Next, we repeat the measurements when the downloads are performed through WiFi tethering via another smartphone. The former measurements help to decide on scheduling if the user wants to download multiple files immediately via the available 3G, at the cost of high energy consumption. The latter measurements help to identify the energy gain when the same files are downloaded via rather low energy consuming WiFi, but at the cost of extra waiting time if the WiFi is not available. We measure the total power consumption via Monsoon power monitoring tool [5], and the download duration via our download tool in a realistic environment. In the measurements, we involve two different smartphones, and focus on large-sized file downloading from a dedicated file server. The overall download duration and power consumption highly depend on the Domain Name Server (DNS) resolution time (if any), initial TCP three-way handshake time, packet latency, and the available end-to-end network bandwidth. Thus, we break down and study the energy consumption in three different phases.

This paper is structured as follows. Brief background knowledge on the power consumption for different wireless access technologies and scheduling of file download activities are explained in Section II. We review the related work in Section III. In Section IV, the testbed and the data collection are discussed in details. The measurement results for different downloading scenarios are given in Section V. The paper is concluded and the future directions are presented in Section VI.

II. CATCHING THE DOWNLOAD TRAIN

Universal Mobile Telecommunications Systems (UMTS) networks are efficient for large data transfers [6] and the RRC consists of basically three power states. At DCH, if there are no packet transmission, the RRC state demotes from a higher power level to a lower one after some inactivity period, which gives some extra time for transmission of the delayed packets. Each packet that is received or transmitted within this inactivity period resets the inactivity timer. If there is no packet received within the inactivity duration, User Equipment (UE) sends a Signalling Connection Release Indication (SCRI) signal to the Radio Network Controller (RNC), *i.e.*, indicating that the currently assigned radio resources are not needed. The RRC state demotes from DCH to FACH, and then if there is no data activity on the link for another inactivity duration, the RRC demotes to the IDLE state. Still, for large data transfers, WiFi tethering might highly reduce the terminal energy consumption as power consumption of WiFi module is less than the 3G component. WiFi power consumption consist of four different states, *e.g.*, high power, idle, PSM light sleep, and PSM deep-sleep, depending on the transmitted and received packets per second [1]. WiFi inactivity timeout is platform specific, and it is much less as compared to 3G.

The sketches for different file downloading scenarios via 3G are given in Fig.1a, Fig.1b, and Fig.1c. Fig.1a depicts the scenario when large sized file downloads are initiated as scattered over a long duration with no scheduling (*e.g.*, asynchronously) on a single smartphone. We break down and study the power consumption within three different phases. The area of the black rectangles represent the total energy consumption on the smartphone during channel allocation via RRC, and establishing a connection with the server. This is referred as *Phase 1* throughout this paper. During Phase 1, the three-way handshake takes place after the DNS resolution (if any). If there is no traffic due to the lack of bandwidth, the RRC state keeps at Phase 1 for some duration, which is a waste of energy. Thus, it is preferred to keep the Phase 1 duration as short as possible. *Phase 2* starts with arrival of the first TCP segment that contains the file content. Phase 2 energy consumption of different files are represented with gray rectangles. The CPU utilization and the throughput increase as compared to Phase 1, which cause the RRC to stay at DCH state. After the last data packet is received at the smartphone, Phase 2 waits for an extra inactivity period, then the *tail phase* starts. There is an extra tail duration, comprising an inactivity timer to release

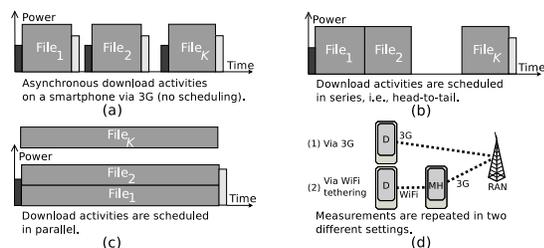


Fig. 1: The energy consumption at Phase 1 (black), Phase 2 (gray), and tail phase (white) are depicted for asynchronous downloading (a), serialized downloading (b), and parallel downloading (c), respectively.

the resources in-between the smartphone and the Radio Access Network (RAN). The tail phase, depicted with white rectangles appended to each download in Fig.1a, causes inefficient usage of resources if there are multiple scattered requests [7]. For all individual downloads in a smartphone, Phase 1 (black) and the tail phase (white) exist, and thus induce high energy consumption. In Fig.1b, the energy consumption is depicted when multiple file downloads are scheduled in series. As a file download starts immediately after the previous one, the number of Phase 1 and Phase 3 states reduces. Fig.1c depicts the download scenario when multiple files are downloaded in parallel. We refer to the formed energy consumption area, as a result of multiplexing different file download activities, as two-dimensional *energy train*. The length of the train is the sum of phase durations; and the height depicts the corresponding mean power consumption of the phases.

III. RELATED WORK

According to Huang *et al.* [7], 10 MB data downloading via 3G and LTE require 34.77 and 1.62 times the energy of WiFi, respectively. In order to reduce the energy consumption on a smartphone, Vergara *et al.* [8] studied a Cross-Layer Burst Buffering (CLBB) algorithm that schedules the background 3G traffic of a smartphone in order to fully utilize the network link when active and in parallel increase the IDLE duration. SPeeDY (SPDY) [9] is an application-layer protocol developed by Google in order to transport web content with low latency. It is multiplexing HTTP requests in a single TCP connection in order to increase the link utilization. In addition, it is also stated that the power consumption at an experimental mobile hotspot (MH) increases by 23 mW for every 1 Mbps increase in the throughput. Sharma *et al.* presents an energy-efficient architecture, Cool-Tether [10] in order to utilize the available smartphones with MH's existing Internet access. Blenk *et al.* studied the tradeoff; while single TCP flows typically do not utilize the available bandwidth on a cellular link due to variable capacities, parallel transmissions yield an increased average transmission duration [11]. In this paper, we consider both the ground truth power measurements and the download durations in the evaluation of energy consumption. Then, we provide energy-efficient recommendations for multiple file downloading depending on the context.

IV. MEASUREMENT METHOD

Energy consumption is a function of the power consumption and duration of a download session. The power consumption and the duration are measured on the smartphone during serialized and parallel downloading in two use cases: (i) via 3G, (ii) via WiFi tethering over another smartphone, MH (see Fig.1d). In the first use case, D is directly connected to the Internet via Telenor's 3G (HSDPA and HSPA+) SIM card with 6 Mbit/s downlink peak rate. In the second use case, D is connected to the Internet via the MH that uses the same SIM card. MH is a non-rooted Samsung Galaxy S4 with 1.9 GHz CPU running Android v.4.3 (kernel v.3.4); and D is a rooted Samsung Galaxy S with 1 GHz CPU running Android v.4.2.2 (kernel v.3.0). In order to automatize measurements, we developed a download tool for the smartphone to perform the HTTP downloads. For each file download, separate TCP connection (*e.g.*, sockets and threads) are created. We record

the ground truth power measurements with the Monsoon power monitoring tool [5]. Tcpdump tool helps to record the data packets during file downloading, however running it on the smartphone induces additional overhead and increases the power consumption. Thus, we have used tcpdump tool to measure only the average Phase 1 duration. Throughout all experiments, the download durations are recorded directly via our download tool. During the experiments, screen brightness is set to minimum, and all irrelevant applications and services are switched off. We created multiple files with exactly the same size and used them during the experiments. The choice of the file size was arbitrary. We wanted to have minimum 30 s download time to distinguish clearly the phases in power consumption. The SIM card used in the experiments allowed up to 6 Mbit/s downlink. Thus, we used 20 MB files in the experiments.

A. Download Duration

The time timestamps when the downloads are initiated and when they are completed, are recorded by our download tool on the local storage, *i.e.*, sdcard, of the smartphone. Accordingly, we calculated the total download duration for the serial and parallel scenarios as the difference between the last and the first timestamps. We divided the total download duration, T_{download} , into two phases. Phase 1 is the initial connection duration, T_{P1} , *i.e.*, in-between when the first SYN is sent and the first TCP segment with the file content is being received. We recorded the relevant metrics via tcpdump such as the timestamp of the first SYN sent from the smartphone and the timestamp of the first TCP segment containing the content data. Phase 2 duration is calculated as the difference between the total download duration measured by the download tool (T_{download}) and the Phase 1 duration (T_{P1}) measured by the tcpdump. The DNS resolution durations are neglected. The download durations depend on the available link capacity at the “time of the day” of the experiment. In order to minimize this effect, we iterated each experiment for more than 30 times in a random order. For parallel downloading, we have validated that all K downloads are performed simultaneously 98% of the time. In serialized downloading, we have also validated (via 31 iterations) that the difference between the serialized downloads is in the range 90 ms to 400 ms for 90% of the time. However, when the file downloads are serialized, *i.e.*, queued, the download completion time of a file depends on the download time of the previous files, except for the first download. Thus, we define the overall processing time as the mean of download completion time of each serialized file, k . The average processing time, for K files in series, $\bar{T}_{\text{processing}}$ is computed as in Eq. 1, where T_{download} is the total download time it takes to download K files.

$$\bar{T}_{\text{processing}} = \frac{1}{K} \cdot \sum_{k=1}^K \left(\frac{k}{K} \cdot T_{\text{download}} \right) \quad (1)$$

If the file download is not queued, such as for parallel downloading, requests do not have to wait, thus processing time is equal to the total download time.

B. Power Measurements

Power measurements are conducted using the Monsoon power monitoring tool. Between each experiment, we wait

until the RRC state switches to IDLE. Thus, we reduce possible side-effects between the measurements. Firstly, the power consumption of D is measured while downloading the files via the 3G. The power consumption pattern is studied for the three different phases: Phase 1, Phase 2, and tail phase. The tail phase duration is not measured, as this is highly depending on the operating system and network configuration. Thus, we assume the state of the art tail phase duration, 12 s [4].

The power measurements differ for D and MH while downloading via WiFi tethering. For D, power is consumed by transmitting data via the WiFi interface. Here, we only consider Phase 2 since the other phases are very short and thus their impact is negligible. If there is no data activity, the power consumption drops within significantly shorter time as compared to 3G. For the mobile hotspot, the energy measurements depend on the data sent and received via the WiFi and the 3G interface. Accordingly, all three phases are considered. In order to reduce the measurement error, enough samples are collected, for each download phase, during steady state.

C. Energy Consumption: Downloading via 3G

At a single smartphone, the file downloads are executed for different scenarios on D via the 3G interface: (1) one download, 1S; (2) two files being simultaneously downloaded, 2P; (3) two files being downloaded in a serialized manner, 2S; (4) four files being simultaneously downloaded, 4P; (5) four files being downloaded in a serialized manner, 4S. In serialized downloading, the download tool executes the downloads of K different files in a sequential order. The download of a file immediately starts when the download of the previous file is completed. In parallel downloading, the download tool executes the downloading of the K different files simultaneously. We calculate the total energy consumption for a multiple file downloading scheme via 3G on a single smartphone, D. E_{total} is the total energy consumption during file downloading. “P1”, “P2”, and “tl” are the subscripts for the power consumption of Phase 1, Phase 2, and the tail phase, respectively.

$$E_{\text{total}} = \bar{P}_{P1} \cdot \bar{T}_{P1} + \bar{P}_{P2} \cdot \bar{T}_{P2} + \bar{P}_{tl} \cdot \bar{T}_{tl} \quad (2)$$

\bar{P}_{tl} and \bar{T}_{tl} are the mean power consumption and the duration of the tail phase, respectively. \bar{P}_{P1} is the mean power consumption of the smartphone during Phase 1, and \bar{T}_{P1} is the mean duration of Phase 1. \bar{T}_{P1} highly depends on the RRC state switching delay from IDLE to DCH. \bar{P}_{P2} and \bar{T}_{P2} are the mean power consumption and the mean duration at Phase 2 during K downloads. When the downloads are scheduled, there is only one Phase 1, and one tail phase, as they are clumped forming an energy train. All the parameters are measured for serialized and parallel downloading, separately.

D. Energy Consumption: Downloading via WiFi-tethering

The measurements for WiFi tethering were conducted using one smartphone acting as MH, and one acting as D, which actually performs the downloads. MH uses the 3G for Internet access, and in parallel shares its Internet access with D smartphone via WiFi tethering. In this use case, D downloads the files from a dedicated server on the Internet via WiFi tethering over MH that has 3G. We measure the power consumption and the download duration at D and as well as the power consumption on MH on the same scenarios

as in Section IV-C, *i.e.*, serialized or parallel downloading. The power consumption of MH is measured in different scenarios: when WiFi tether is OFF (baseline measurements); when WiFi tether is ON with no connections; when WiFi tether is ON and D is connected; when WiFi tether is ON and D is downloading a single file via MH, 1S; when WiFi tether is ON and D is downloading two files in parallel via MH, 2P; and when WiFi tether is ON and D is downloading four files in parallel via MH, 4P. The total energy consumption during WiFi tethering is the sum of energy consumed at MH and the energy consumed at the connected device, D.

V. MEASUREMENT RESULTS

We measured the download duration and the power consumption while downloading with respect to a set of scenarios in realistic environment. The collected steady-state power consumption values are normally distributed ($R^2 > 0.9$). The power consumption is broken down into Phase 1, Phase 2, and the tail phase. Phase 2 and Phase 1, in 3G, have the highest and the least power consumption, respectively.

A. Power Consumption During File Download

First, the power measurements when D downloads files via its individual cellular data interface (3G) are given. Next, we discuss the power measurements during WiFi tethering. The measurements are presented through Table I-II as the 95 % confidence levels are negligibly low (mean: 0.47mW) due to high number of samples (see last columns of the tables).

1) *Downloading via 3G*: The measured power consumption values on D for different file downloading scenarios via the 3G interface are given in Table I. In the table, from left to right, the mean, standard deviation, minimum, maximum of the power consumption values, and the corresponding number of samples at each scenario are given. The power consumption values at Phase 1 and at the tail phase are given at rows 1 – 2, respectively, which are lower than the Phase 2 power consumption values as presented at rows 3 – 7. Row 3 shows the power consumption when there is only one download (1S). In rows 4 – 5, which belong to the serialized downloading, no significant difference in the power measurements are observed when different number of files are scheduled. This is expected as there is only one established TCP connection at a time during downloading. On the other hand, when the downloads are executed in parallel, the power consumption slightly increases with the number of simultaneous downloads as given in rows 6 – 7. In these scenarios, the number of established connections at a time is equal to the number of simultaneous downloads, *i.e.*, threads. Thus, we believe that the power consumption increases with the CPU utilization and the increased throughput.

2) *Downloading via WiFi Tethering*: The measured power consumption values of D for the same scenarios, while downloading exactly the same file via WiFi tethering over MH’s 3G interface, are presented in Table I in rows 8 – 10. D uses the WiFi interface, and the power consumption values are reduced as compared to using the 3G interface. When the downloads are executed in parallel, slight increase in the power consumption values is observed. However, this gives an extra overhead to MH. Next, we present the power consumption values on MH.

TABLE I: Power Consumption (mW) on Samsung S (D).

Row	Scenario	Mean	Std.	Min.	Max.	Samples
1	3G, P1	1153	167	703	2370	275014
2	3G, Tail	1276	86	1063	2217	312511
3	3G, 1S, P2	1496	283	1003	3597	1300008
4	3G, 2S, P2	1508	298	1006	3799	1550004
5	3G, 4S, P2	1514	313	955	3928	3500005
6	3G, 2P, P2	1662	354	931	3717	1950007
7	3G, 4P, P2	1956	409	1114	4077	3150007
8	WiFi tether, 1S	957	240	386	2950	1575017
9	WiFi tether, 2P	1266	333	547	3278	1975008
10	WiFi tether, 4P	1350	356	514	3330	2175009

TABLE II: Power Consumption (mW) on Samsung S4 (MH).

Row	Scenario	Mean	Std.	Min.	Max.	Samples
1	3G, P1	997	130	661	2889	475005
2	3G, Tail	1354	120	1047	3050	350006
3	3G, TetherOFF	549	106	371	2021	925004
4	3G, TetherON	845	179	628	2715	1475003
5	3G, D Connected	844	182	632	3112	700001
6	3G, TetherON, 1S, P2	1952	310	1304	4137	615005
7	3G, TetherON, 2P, P2	2146	288	1411	4391	625003
8	3G, TetherON, 4P, P2	2179	336	1374	4474	1325004

3) *Impact of WiFi Tethering on MH*: The power consumption values measured at MH during different WiFi tethering scenarios are given in Table II. Phase 1 and the tail phase power consumption values are given in rows 1 – 2, respectively. Rows 3 – 5 present the power consumption values when the WiFi tethering on MH is switched OFF, switched ON, and D is connected (without download), respectively. Rows 6 – 8 present the Phase 2 power consumption values when D is downloading a single file, two files in parallel, and four files in parallel, respectively. There is a significant difference in power consumption in-between the scenarios when the WiFi tethering is enabled or not. However, there is no significant difference if the D smartphone is connected or not given that D does not download data. The power consumption of MH increases with the number of concurrent downloads.

B. Download Durations

We first cleaned the measurement data from a few outliers that were confirmed as measurement errors. The cleaned Phase 1 duration dataset comprises 31 iterations. Phase 1 duration is calculated as $4.6 s \pm 0.3 s$ with 95 % confidence level. The duration of a single file download via 3G is given in the first row of Table III. The mean, standard deviation, minimum, maximum of the parallel download durations via 3G on D for two parallel (2P), and four parallel (4P) are given in the second and third rows of Table III, respectively. The download duration increases with the number of concurrent downloads on a single smartphone. The total download duration, during both the two serial and four serial download scenarios, are given in the fourth and fifth rows of Table III. We applied one-way Anova test on the individual download durations during two and four serial downloading as the download durations are normally distributed ($R^2 = 0.98$). We conclude that there is no difference in-between the download durations ($p > 0.005$) of each file during serialized downloading. The total download duration for serialized file downloading via the 3G interface is

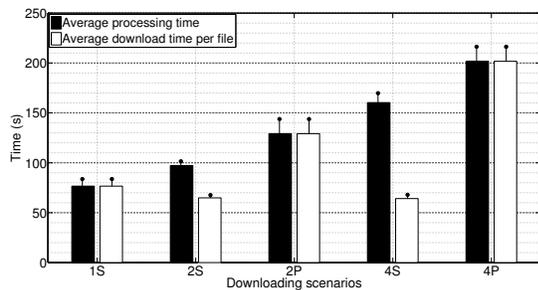
less than the total download duration for parallel downloading. For parallel downloading, this indicates that the available link is too much utilized, with high number of TCP connections at a time, causing packet losses and retransmissions. Phase 2

TABLE III: Download duration (s) for multiple downloads.

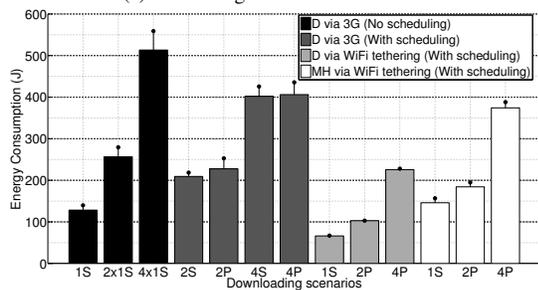
Row	Scenario	Mean	Std.	Min.	Max.	Iterations
1	3G, 1S	76.6	20.1	48.0	122.7	31
2	3G, 2P	129.2	42.2	65.3	235.7	32
3	3G, 4P	201.7	43.4	138.6	320.7	34
4	3G, 2S	129.7	15.8	96.5	158.6	31
5	3G, 4S	256.5	42.9	188.5	329.4	31
6	WiFi tether, 1S	69.2	13.9	45.1	103.8	29
7	WiFi tether, 2P	81.2	12.0	63.6	111.9	28
8	WiFi tether, 4P	167.2	16.6	136.6	216.4	30

durations are calculated as the difference between the total download duration, T_{download} and Phase 1 duration, T_{P1} . \bar{T}_{P1} is assumed 4.6 s for all smartphones that receive the data via the 3G interface. The last three rows, 6–8, depict the durations of the same downloads via WiFi tethering over MH. The average processing times and the download times per file, for serialized and parallel downloading on a single smartphone, are presented in Fig. 2a together with the upper 95 % confidence levels. There is no queuing in parallel downloading, but the download time is higher. Thus, the average processing time is 33 % and 26 % higher than in serialized downloading, when two and four files are scheduled, respectively.

Based on the power and download duration measurements, the calculated energy consumption values are given in Fig. 2b together with the 95 % confidence levels. The energy consumption when two and four files are downloaded without scheduling are depicted with 2x1S and 4x1S on the x-axis of Fig. 2b, respectively. Energy consumption is reduced by 15 % and 22 % when two and four files are scheduled, respectively.



(a) Processing and download time.



(b) Energy consumption.

Fig. 2: Processing time and energy consumption.

Serialized downloading yields less energy consumption due to lower total download time. The sum of the energy consumption of D and MH in WiFi tethering is higher than in 3G scenario, however the energy consumption of D decreases 48 % in WiFi tethering scenario, as expected.

VI. CONCLUSION

This paper presents power consumption and download duration measurements for downloading multiple files (20 MB each) with different scheduling on a smartphone via realistic settings. Based on the measurements, we show that scheduling of two or four downloads in series consumes less energy, less processing time and less per-file downloading time, as compared to the scenario when the same files are downloaded in parallel. Performing the downloads via WiFi tethering reduces the energy consumption, with the cost of waiting time for an available hot spot. The energy consumption of MH increases with the amount of download requests from the connected device, however MH can be connected to a power source. This way, it can provide Internet access to more smartphones, which are not connected to the power grid. We believe that WiFi tethering is a great facility, which needs to be managed and populated in a smart way in order to decrease the total energy consumption and to increase the utilization of cellular data link per smartphone. The results obtained are specific to 20 MB file size. We plan to conduct further measurements via WiFi tethering with a wide variety of smartphones, with varying file sizes, with various peak data rates to model the tradeoff between energy consumption and the waiting time.

REFERENCES

- [1] J. Manweiler, R. R. Choudhury, "Avoiding the rush hours: WiFi energy management via traffic Isolation," IEEE Transactions on Mobile Computing, vol.11, no.5, pp. 739, 752, May 2012.
- [2] S. Ickin *et al.*, "Factors influencing quality of experience of commonly used mobile applications," Communications Magazine, IEEE, vol.50, no.4, pp.48,56, April 2012.
- [3] A. Carroll and G. Heiser, "An analysis of Power Consumption in a Smartphone," USENIX Conference on USENIX Annual Technical Conference (USENIXATC'10), pp. 271–284, Berkeley, CA, USA, 2010.
- [4] J. Huang *et al.*, "RadioProphet: Intelligent radio resource deallocation for cellular networks," Passive and Active Measurement, Springer International Publishing, vol. 8362, pp. 1-11, 2014.
- [5] Power Monitor, Monsoon Solutions Inc., [Online], Available: <http://www.monsoon.com/LabEquipment/PowerMonitor/>.
- [6] G. P. Perrucci *et al.*, "On the impact of 2G and 3G network usage for mobile phones' battery life," Wireless Conference, EW. European, vol., no., pp.255, 259, 2009.
- [7] J. Huang *et al.*, "A close examination of performance and power characteristics of 4G LTE networks," 10th international conference on mobile systems, applications, and services (MobiSys '12). ACM, NY, USA, 2012, pp. 225-238.
- [8] E. J. Vergara *et al.*, "Kernel level energy-efficient 3G background traffic shaper for android smartphones," 9th International Wireless Communications and Mobile Computing Conference (IWCMC), vol., no., pp.443,449, 2013.
- [9] SPeeDY, [Online], Available: <https://developers.google.com/speed/spdy>.
- [10] A. Sharma *et al.*, "Cool-Tether: energy efficient on-the-fly wifi hot-spots using mobile phones," 5th international conference on Emerging networking experiments and technologies (CoNEXT '09), ACM, NY, USA, 109-120. 2009.
- [11] A. Blenk *et al.*, "Dynamic HTTP download scheduling with respect to energy consumption," 24th Tyrrhenian International Workshop on Digital Communications (TIWDC), Italy, 2013.