Demonstrating the Stalling Events with Instantaneous Total Power Consumption in Smartphone-based Live Video Streaming

Selim Ickin, Markus Fiedler
School of Computing
Blekinge Institute of Technology
Karlskrona, Sweden
selim.ickin, markus.fiedler@bth.se

Katarzyna Wac
Institute of Services Science
University of Geneva
Geneva, Switzerland
katarzyna.wac@unige.ch

***Abstract*—The smartphone usage nearly tripled in 2011 according to Cisco Virtual Networking Index. There is a high demand of energy for using popular mobile applications, which run on smartphones with limited battery life. Video streaming applications are widely used on mobile devices, and their high power consumption exhibits high variance during a live streaming session, due to varying conditions on network and application levels. Recent studies focus on the averaged power consumption statistics, while there is lack of observation on the fluctuations of the *instantaneous total power consumption* of the smartphones. Network based applications consume power at all layers of the communication stack, and any fluctuation in the total power consumption during a video streaming can reveal a possible misbehaviour such as a *stalling event*. Until now, these events are investigated in Quality of Experience (QoE) studies through installation of high-energy demanding and hard-to-deploy network measurement tools on users' mobile devices. In this paper, we demonstrate an experiment, where a user experiences a stalling event on the smartphone and observes the live instantaneous power consumption values through Mobile Power Monitoring Tool (MPMT) and Software Visualisation Tool (SVT), simultaneously. We confer that the instantaneous total power consumption likely reveals the misbehaviours such as stalls during a video play-out in live video streaming on smartphones that can facilitate energy efficient QoE studies.**

***Keywords*-Power consumption; energy efficient; stalling; video streaming; visualisation tool.**

I. Introduction

Multimedia applications are immensely desired by smartphone users, but they may be costly with respect to network capacity and energy [1]. Based on our previous study [2], the power consumption is one of the most important key influential factors on the overall perceived quality of smartphones, which we referred to QoE. Recent studies mostly focus on monitoring the averaged values of overall power consumption per application basis to increase the battery performance of handheld devices. Yet, there is still lack of focus on the variations in the power consumption measurements. In network based applications, the communication stack consists of standardized functions distributed into different protocol layers that consume energy on the communication systems. Thus, during the play-out of a video streaming application, any abnormal interrupt on one of those layers influences the Instantaneous Total Power Consumption (P_n).

Most popular video applications work based on transmission-controlled streaming, and a stalling event, so-called *freeze*, is a common impairment and it is considered as a key influence factor in user's recent perceived video quality [3]. Probing the underlying network-layer metrics during user studies inorder to identify the influencial factors for poor user experiences, needs high-energy demanding and hard-to-deploy monitoring tools. In addition, an instantaneous increase in the delay and packet loss rate does not ensure that the video streaming is interrupted by a stalling event, due to its dependency on the size of the jitter buffer. The freezes and the corresponding fluctuations in the measured power values might be caused for different reasons. For example, during video streaming on a mobile terminal, when there is a 'hiccup' in the network traffic, some HTTP Live Streaming (HLS) clients with long playout buffer are known to deactivate their network module. Derivation of robust power models that can represent the worst-case network scenarios can empower implementation of energy efficient QoE measurement tools.

In this paper, we demonstrate that the power consumption metric has some potential to identify the misbehaviours in all layers of the communication stack during video streaming that have consequences such as stalling events. Based on our preliminary tests, we hypothesize that the fluctuations in the power consumption metric are likely correlated with the occurrence of stalling events throughout video streaming. We conducted our preliminary experiments, while a live video is streamed and displayed on the Android smartphone screen. P_n values are collected and visualised through Mobile Power Monitoring Tool (MPMT) [4], and Software Visualisation Tool (SVT), simultaneously.

978-3-901882-46-3

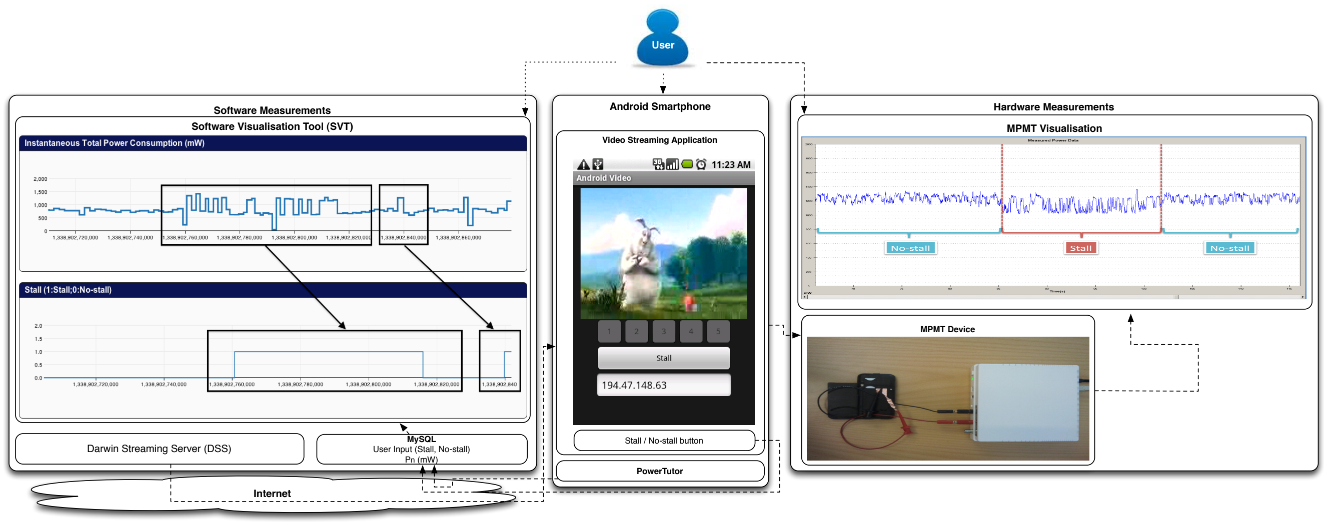

Figure 1. Testbed used for demonstrating the simultaneously collected P_n values through SVT and MPMT during live video streaming.

II. Related Work

In [5], it is claimed that the mobile devices have different power efficiency for Internet streaming. An on-line power estimation and model generation framework, PowerTutor, has been described in [6], to inform developers and users regarding the power consumption implications in application design and use. Wireless interface power consumption with respect to varying states (on/off), transfer (uplink/downlink) speeds, security settings (off/WEB/WPA), radio signal strength, and data types was studied in [7]. In [8], together with the mean power consumption, the standard deviation of the power consumption data is transformed into information to detect abnormal energy consumption in buildings. In [9], anomaly detection methods are applied to the data from Symbian client to distinguish between normal and abnormal behaviour, *e.g.*, malicious software activity.
The most important difference between the existing studies and ours is that we focus on the instantaneous total power consumption on a smartphone to detect stalling events directly during the steady state of live video streaming.

III. Experiment Setup

The experimental setup is summarized in Fig.1. There are three main components; software measurements, hardware measurements, and the Android smartphone. The latter runs the video streaming application that we have previously implemented for QoE evaluation to stream the video from the streaming server [10]. The Real Time Streaming Protocol (RTSP) protocol is used for streaming via radio interface over the Internet through the dedicated Darwin Streaming Server (DSS) framework on a MacOSX (10.6.8). The video is MPEG-4 compressed with dimensions 176x144, and 25 fps. The video was streamed at a rate of 481 kbit/s. The smartphone is installed with our live video player application and and modified version of the open source project PowerTutor. P_n is measured as the total instantaneous power consumption of all components of the smartphone, and we designed the following testbed to demonstrate the effect of a stalling event with the live P_n values.

A. Hardware Measurements

Hardware measurements are conducted through the MPMT device. It comes with its own software that visualises the measurements. The experimental setup is established while smartphone's battery is bypassed by MPMT. It is connected to PC for visualisation, and works with high precision [4], *i.e.*, it generates 5000 measurement samples (in milliwatts) per second. The consumed power at each sample is measured as the product of the instantaneous voltage and current.
We consider the measurements through MPMT as ground truth, however it has disadvantages in terms of portability. Especially in QoE studies, conducting non-obtrusive experiments is vital. In addition, with this setup alone, it is a challenge to synchronize the timestamps of the power consumption values, Quality of Service (QoS), and the user behaviour metrics, simultaneously. Therefore, we supported the experiments with software measurements.

B. Software Measurements

Software measurements consists of two components; SVT and modified version of the open source project

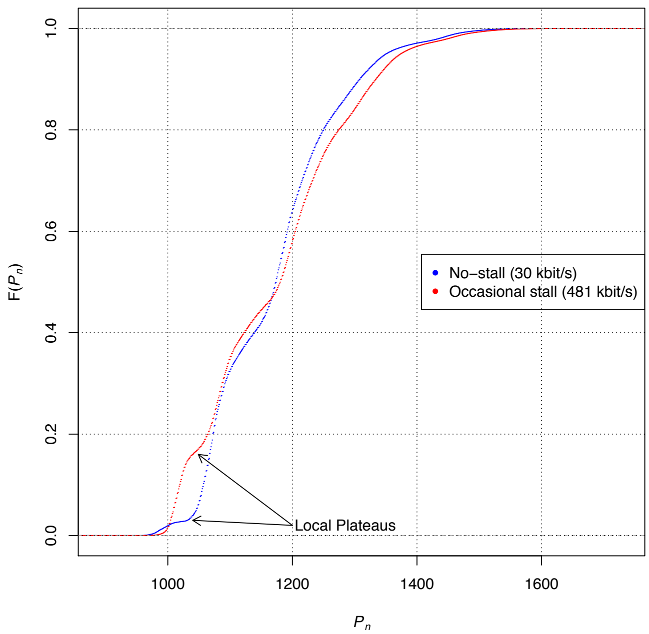

Figure 2. Empirical cumulative density functions are given based on *No-stall* and *Occassional stall* datasets collected through live streaming of identical movies with different data rates.

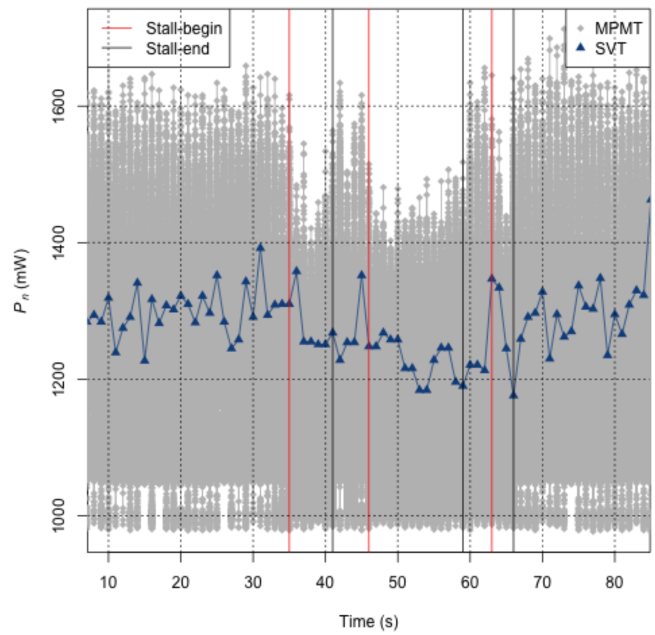

Figure 3. Simultaneous measurements from MPMT, SVT, and the corresponding three stalling regions marked by the user are illustrated.

PowerTutor [11]. The details regarding the power measurements in PowerTutor are found in [6]. PowerTutor runs a background service that samples the P_n values per second and sends to a dedicated MySQL server through the Internet in the form of JSON objects. SVT is our visualisation tool that is implemented via the RRDtool [12]. It fetches the values stored in the database and visualises it on the HTML web page.

IV. Preliminary Observations

On the right hand side of Fig. 1, P_n values are presented in a video streaming scenario that consists of a stalling event. *Stall* and *No-stall* regions are depicted with red and blue, respectively. Increased fluctuations of P_n values around a lower average is visible during a *Stall* as compared to the *No-stall* region. On the left hand side of Fig. 1, the live measurements from SVT are illustrated with the corresponding region marked as *Stall* by a user. The timestamps are labeled in different time scales, thus at a first glance, the user reaction looks delayed with respect to the fluctuation region in the power consumption, although it is not. We have collected two datasets through MPMT with 5000 measurements per second during live video streaming; first involves 1269 seconds streaming data without any *Stall* period and the latter involves 1570 seconds streaming that involves *Occasional stalls*. The *No-stall* and *Occasional stall* datasets are collected with identical video sources that are streamed at a rate of 30 kbit/s and 481 kbit/s, respectively. Empirical Cumulative Distribution Function (ECDF) for two different datasets are depicted in Fig. 2. The first *local plateau* for 'No-stall' is visible at 1025 mW with ECDF of 0.025, while the first *local plateau* for 'Occasional-stall' is at 1050 mW with ECDF of 0.15.

We delved further into the stalling regions, and collected further data through MPMT and SVT, then synchronized and visualised the data in Fig, 3. P_n values from MPMT and SVT are illustrated with gray and blue, respectively. The regions marked by the user as stalling periods are shown by the beginning and the end of the stalling events with red and black vertical lines, respectively. During those stalling periods, there exists increased fluctuations, and drop on P_n values.

V. Demonstration

We will use the testbed in Fig.1 during the demo. The audience will be asked to watch a 2 minutes long video that is streamed via the radio interface of the smartphone. During the experiments, the user will press the *Stall* button as a stall is visible in the video, and similarly press the *No-stall* button when the corresponding stall ends. For the demonstration, we present the live P_n values measured through both the MPMT and the SVT, and the *stalling* regions defined by the user during the live streaming of a video. HTC Dream G1 that runs Android 1.6 will be used, and the applications and services that are irrelevant to the demo will be turned off during the experiments. Since it is more probable to experience freezes while streaming through higher bitrates, we will stream at a high-enough rate of 481 kbit/s during the demonstration.

VI. Limitations

There are many factors that might cause fluctuations in P_n, *e.g.*, user click streams, active network interface, radio modules, data transfer speed, and GPS polling period, however analyzing all available factors is beyond the scope of our work. Indeed, live video streaming applications in a smartphone is not as user interactive as other popular genre of mobile applications such as gaming, *i.e.*, users do not touch the phone screen often while watching a smooth video play out in good network conditions. Therefore, we have neglected the user behaviour during the video streaming experiments. During the demonstration, we are also aware that the time difference between the stalling region marked by the user and the power drop depends also on the user reaction time, *i.e.*, the time it takes for the user to press the *Stall* button upon a stall event.

VII. Conclusion

In this paper, we demonstrate the stalling events and their influence on Instantaneous Total Power Consumption (P_n) through Mobile Power Monitoring Tool (MPMT) and Software Visualisation Tool (SVT). We have observed energy savings during the stalling events while streaming live video to the Android smartphone. We have considered the hardware measurement as accurate, and taking this as reference, we tried to verify the results obtained through the software measurements. On both measurements, we observed that increased fluctuations on the power consumption values during a not-user-interactive live video streaming session at its steady-state on a smartphone likely tells that either there existed a stalling event, which was caused by the communication channel, or due to another unexpected reason related to the application. Energy measurements are done via built-in applications in almost all mobile devices, therefore once robust power models for those misbehaviours are identified and deployed, the need of collecting other network indicative QoS metrics with high-energy demanding tools may diminish in the future. Then, new research methods based on power consumption and energy savings can be suggested for QoE studies. By this way, those user studies on network depending applications can be done in a non-obtrusive and more energy efficient way.

As future work, we want to show that the power consumption values for stalling and non-stalling regions during a live video streaming session are statistically significantly different. We will be repeating the experiments, while assuring to cover a representative dataset for real world scenarios during not-user-interactive video streaming.

References

[1] R. Duan; B. Mingsong; C. Gniady, "Exploring memory energy optimizations in smartphones," Green Computing Conference and Workshops (IGCC), 2011 International , vol., no., pp.1-8, 25-28 July 2011.

[2] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J. Hong, A. K. Dey, Factors Influencing Quality of Experience of Commonly-Used Mobile Applications. IEEE Communications Magazine, Special Issue on QoE Management in Emerging Multimedia Services, April 2012.

[3] T. Hossfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, R. Schatz, "Quantification of YouTube QoE via Crowdsourcing," Multimedia, International Symposium on, pp. 494-499, 2011.

[4] Power Monitor, (Online: Verified December, 2010). Avialable at http://www.msoon.com/LabEquipment/PowerMonitor/.

[5] Y. Liu, F. Li, L. Guo, and S. Chen. A measurement study of resource utilization in internet mobile streaming. In Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video (NOSSDAV '11). ACM, New York, NY, USA, 33-38.

[6] L. Zhang and B. Tiwana, et.al. "Accurate online power estimation and automatic battery behavior based power model generation for smartphones", In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2010.

[7] M. Marcu and D. Tudor. "Energy consumption model for mobile wireless communication", In *Proceedings of the 9th ACM international symposium on Mobility management and wireless access, MobiWac' 11.* pp. 191-194, New York. 2011.

[8] J. E. Seem. Using intelligent data analysis to detect abnormal energy consumption in buildings, Energy and Buildings, Volume 39, Issue 1, January 2007, Pages 52-58.

[9] A. Schmidt, F. Peters, F. Lamour, and S. Albayrak. Monitoring smartphones for anomaly detection. In Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications. 2008.

[10] S. Ickin, K. Vogeleer, M. Fiedler, and D. Erman. "On the choice of performance metrics for user-centric seamless communication", In Third Euro-NF IA.7.5 Workshop on Socio-Economic Issues of Networks of the Future, 2010.

[11] PowerTutor. (Online: Verified May, 2012). Avialable at http://powertutor.org.

[12] RRDtool, (Online: Verified May, 2012). Avaliable at http://oss.oetiker.ch/rrdtool/