Software-related Energy Footprint of a Wireless Broadband Module

Mikael Asplund
mikael.asplund@liu.se

Anton Thomasson
antth455@student.liu.se

Ekhiotz Jon Vergara
ekhiotz.vergara@liu.se

Simin Nadjm-Tehrani
simin.nadjm-tehrani@liu.se

Department of Computer and Information Science
Linköping University
581 83 Linköping, Sweden

ABSTRACT

Energy economy in mobile devices is becoming an increasingly important factor as the devices become more advanced and rich in features. A large part of the energy footprint of a mobile device comes from the wireless communication module, and even more so as the amount of traffic increases. In this paper we study the energy footprint of a mobile broadband hardware module, and how it is affected by software, by performing systematic power consumption measurements. We show that there are several cases where the software does not properly take into account the effect that data communication has on the power consumption. This opens up for potential energy savings by creating better applications that are aware of the energy characteristics of the communication layer.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Experimentation, Measurement, Performance

Keywords

3G, Energy footprint, Power consumption, Wireless broadband

1. INTRODUCTION

The battery lifetime is one of the decisive factors when developing faster and better mobile devices. A modern mobile phone can easily be discharged in less than a day with active use. End users have been accepting shorter battery life in exchange for better screens and better connectivity for some time. However, there is a point where the nuisance of charging the phone will outweigh the perceived benefits of increased performance.

While the energy consumption is an essential part of developing new hardware components, it has been a neglected aspect in the software development. This is likely due to the fact that people don't consider that software significantly affects power consumption. However, in order to meet the increased demands on performance with limited battery power, we need to employ every possible power saving measure, including those that need to be governed at application level. In this paper we will focus on the energy footprint of a 3G mobile broadband module. Such a module can be found in modern laptops and tablets and it contributes to a significant portion of the total energy consumed by the device.

Unfortunately, there seems to be some misunderstanding of how energy consumption is affected by the behaviour of software applications. One such misunderstanding is the belief that the consumed energy is simply proportional to the amount of data transmitted. As we and others before us have shown [16, 12], this is not the case. Instead, the energy consumption of a mobile communication device is foremost dependent on the communication mode that the device is working in. This in turn is not only affected by the data transmissions, but also by network timers and control traffic. Thus, to properly understand how software affects energy consumption one needs to actually measure it, and not just look at indirect metrics such as the amount of data sent.

In this paper we argue that application software is indeed contributing to a lot of wasted energy by not acknowledging the characteristics of the connection that a device is using. By performing actual power consumption measurements on a mobile broadband module we have identified three specific cases where the software contributed to a significant waste of energy. These cases are (1) using 3G instead of 2G when not needed (2) streaming live data continuously instead of in bursts (3) applications and operating systems that spuriously creates non-essential TCP connections.

The contributions of this paper are twofold; (1) a thorough experimental study of the energy consumption of a mobile broadband module using both synthetic and real-world traffic and (2) identification of three cases of energy waste due to software not being optimised for reducing the energy footprint. A key strength of our measurements is that the we

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiWac'11, October 31–November 4, 2011, Miami, Florida, USA.
Copyright 2011 ACM 978-1-4503-0901-1/11/10 ...$10.00.

are able to isolate the power consumption of the broadband module from the rest of the system.

The rest of the paper is organised as follows. Section 2 describes the physical setup that we have used to perform the measurements and Section 3 provides some basic background data on the energy footprint of a 3G communication module. Section 4 to 6 each describe one aspect of the energy footprint that could be reduced with software measures. Finally, Section 7 briefly describes related work and Section 8 concludes the paper and provides some directions for future work.

2. PHYSICAL SETUP

We will now proceed to describe the physical setup that we have used to measure the power consumption related to cellular communication. The measurements have been done on the Ericsson F3307 which is a 3G mobile broadband module intended to be integrated in a mobile device such as a laptop or tablet. It is advertised as being optimised for tablets due to its low power consumption (4mA when not connected), and supports a maximum data rate of 7.2Mbit/s.

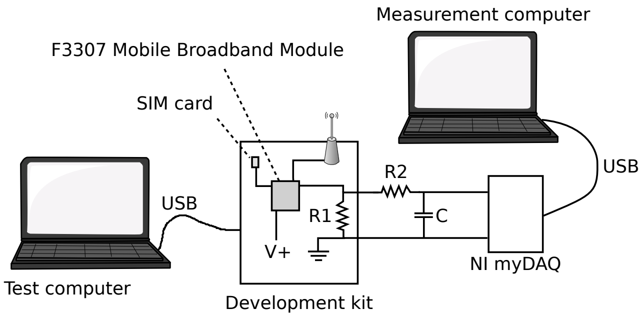

Figure 1: Measurement setup

Figure 1 shows the setup that we used to measure the power consumption of this module. We used a development kit for the module provided by Ericsson that we connected to the test computer using a USB cable. The majority of the tests were performed with a test computer running Ubuntu Linux version 10.10 (with the only exception of OS-related tests where also Windows was used). The development kit had an on-board shunt resistor (R1 in the figure) of 0.1Ω. By measuring the voltage over this resistor it is possible to derive the power consumption of the module. The voltage was measured using a National Instruments myDAQ data acquisition unit. In order to make sure that the measurements would not suffer from any anti-aliasing effects a simple low-pass filter (R2 = 10KΩ and C = 1μF in the figure) was introduced between the development kit and the data acquisition unit. The signal was sampled at 1kHz and the cutoff frequency of the low pass filter was approximately 16Hz.

All measurements were performed in the same location at the university which meant that the signal strength did not vary considerably. Most of the measurements were done using a SIM card from TeliaSonera which provided access to a 3G subscription that allowed full use of the available capacity. TeliaSonera is 5th largest telecom operator in Europe with a total of 157 million customers. We also performed some measurements with a SIM card from the telecom operator Hi3g Access AB under the brand name "3". Both networks are UMTS[1] type networks that implements the HSPA[2] standard. The 2G measurements were performed with in the TeliaSonera network using the GPRS[3]/EDGE[4] standard (sometimes called 2.5G).

3. BASIC ENERGY FOOTPRINT

In order to understand the factors influencing the energy consumption of a broadband module one must consider the 3G radio state machine. Thus, in this section we will present some background information regarding the 3G mode of operation and how this affects the power consumption of the mobile device.

Figure 2 shows the possible states a 3G terminal can be in. Note that some network operators do not implement the standby state, but these are the states used by TeliaSonera. The mobile device can choose to connect and disconnect, but the other state transitions are controlled by the operator. The placement of the states along the y axis corresponds to the power consumption and the placement along the x axis can be seen as an indication of the performance possible in that state. When the device is in idle state it is not connected to the base station, so no communication is possible. Moreover, some signalling is required before a connection can be established. The state marked as "Standby" has the same low energy consumption as the idle state, but allows much faster transition to the states where traffic can be sent.

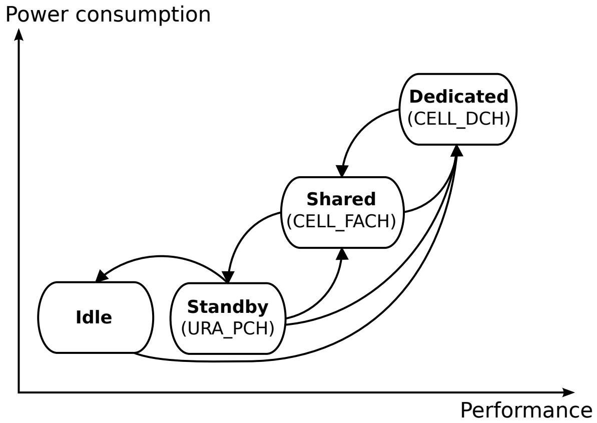

Figure 2: 3G Connection states (adapted from [18])

The shared and dedicated states both allow data to be sent and received with the difference that the latter provides a dedicated channel for communication allowing significantly higher data rates. Of course, this comes at a cost of a higher power consumption. The shared state is intended to allow small amounts of control traffic without the need of a energy-intensive high speed channel. After approximately 90 seconds in idle mode the F3307 module will go to an even more power-saving idle state (this it not part of the 3G radio state machine shown in Figure 2 but rather a device specific state).

[1] UMTS: Universal Mobile Telecommunications System

[2] HSPA: High Speed Packet Access

[3] GPRS: General Packet Radio Service

[4] EDGE: Enhanced Data rates for GSM Evolution

Figure 3 shows our measurements of the average power consumption for the different states. Low power idle consumes on average 0.02W, whereas normal idle and standby consumes 0.2W. Being idle in the shared stated consumes 0.5W. For the dedicated state there are three bars, two with receiving data at 32kbit/s and 1024kbit/s respectively and one where data is transmitted (at maximum speed).

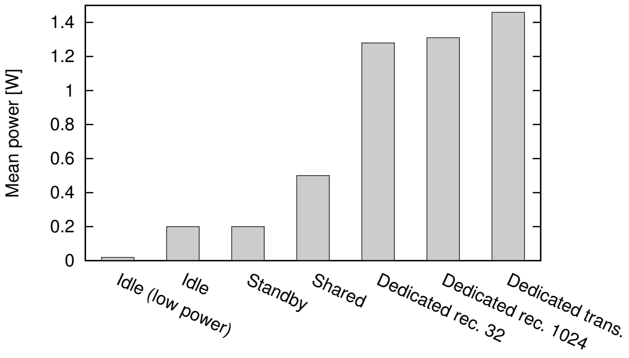

Figure 3: Energy consumption of different 3G states

Based on this data we can conclude that the energy cost of receiving data at high rate is not much higher than receiving at a low rate. We also see that the cost of transmitting is higher than receiving, although not significantly so. For the mobile device, this means that it should not use a dedicated channel other than in order to send or receive data. Unfortunately, as already mentioned, the mobile device does not control what state it is in.

The operator uses timers to switch states for a device. If the device is in dedicated mode and no data is sent during a some time period it will be moved to the shared state. Similarly, there are timers from shared to standby and from standby to idle. Long timers mean that the device stays in the faster states resulting in a lower latency but at higher energy consumption. Traditionally, many operators have used relatively long timers. However, with the sharp increase of smart phones and tablets the load on the network operators have increased which means that they tend to reduce the length of the timers to allow more terminals in the same cell.

At the same time, device manufacturers have also realised the heavy cost of the long tails which has lead to implementing a feature called “fast dormancy”. This means that the device will simply drop the connection and go to idle state instead of waiting for the network timers. The problem with this approach is the long latency involved in returning from the idle state. In an ongoing effort to fix this problem the latest 3G standard includes a mechanism with which the device can signal its desire to go to a low power state (thereby allowing the operator to move the device to standby state without waiting for the timer to expire).

Figure 4 demonstrates a typical example of a connection to a TeliaSonera base station. Before the time 3s, the connection is in standby and the only energy cost is when the radio circuit periodically becomes active. The traffic causes the connection to jump up to dedicated mode (first performing some signalling traffic) where it stays until the data has been received plus an additional 3 seconds. At time 10s, the timer at the operator expires and moves the device to the shared mode. The timer in this state is set to 5 seconds after which the device returns to standby.

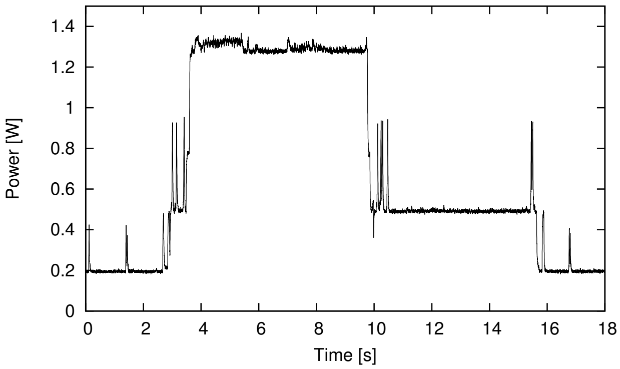

Figure 4: Power profile for operator TeliaSonera

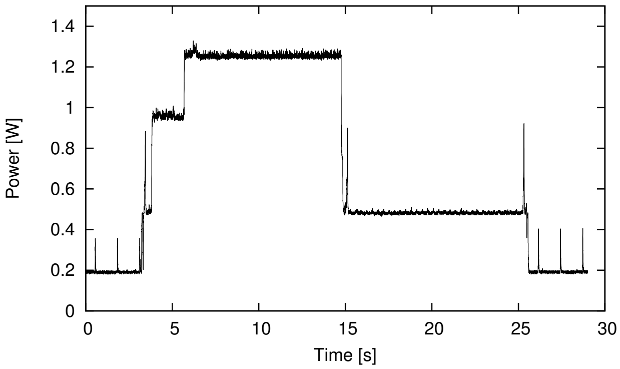

Figure 5: Power profile for operator 3

Figure 5 shows a similar power profile using the 3 network. There are a couple of interesting differences compared to the TeliaSonera case. First of all, since 3 has not implemented a PCH standby state, the device starts from the idle state meaning that it takes a longer time to go to the dedicated channel (a ping request takes approximately 2.5 seconds with the 3 network compared to 0.7 with TeliaSonera). Secondly, the network timers are considerably longer in 3’s network (approximately 8s from dedicated to shared, and 10s from shared to idle). This means that the user will not suffer so much from the longer latency associated with going from idle state (since the device will stay in the shared channel), but this will also result in an increased power consumption.

4. 2G VERSUS 3G

We will now go on to show the first of the three cases in this paper where software could contribute to reducing the energy consumption in a mobile device. The power consumption when receiving data using 2G is considerably smaller compared to that of 3G. However, at the same time, since the data rate of 2G is only a fraction of 3G, a device must stay connected considerably longer with 2G than with 3G. If one were to only consider the energy cost while receiving/transmitting data, then 3G provides a better energy per bit ratio and would be preferable. However, due to the fact that the device stays connected for some time after the transmission has ended small transmissions are more costly in 3G.

Figure 6 shows the total amount of energy (not counting the idle consumption) consumed by the broadband module for transferring a file of fixed size. On the x-axis the file size is shown on a logarithmic scale. The values are the average of 10 file transfers, and the error bars correspond the 95% confidence interval. The reason for the high variability of the 2G connection for large file sizes is that the data rate of the connection varied considerably, leading to large variations in transfer time.

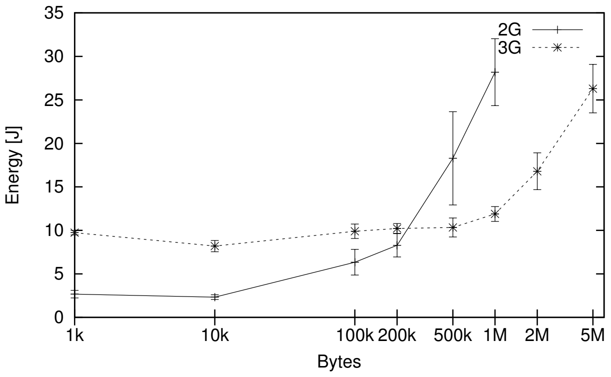

Figure 6: Energy consumption of 2G vs 3G

We can see from this result that for this particular setting, transferring files of less than 200KB should be done using a 2G connection, and larger files should be transferred over 3G. Note though, that for interactive applications such as web browsing, the high latency of 2G makes it a bad choice. However, for background operations such as syncing a mail box, where the timing is not an important factor, 2G should be the preferred connection mode.

Note that the crossover point will not be the same for all networks and settings. This will depend on factors such as the signal strength and the network timers used by the operator. Thus, for the best performance, the choice of whether a transmission should be done using 2G or 3G should be based on a more detailed model of the energy consumption that includes these factors. However, for practical purposes, it suffices to know that if the data is measured in a few kilobytes or less, then using 2G will consume less energy.

5. STREAMING

The second case of software-caused energy waste is that of live multimedia streaming. Note that the term "live" is of importance. There are many multimedia streaming services such as Youtube and Spotify where the content is being distributed after it has been produced. We will revisit Spotify in the next section, but for now we concentrate on live services such as Internet radio and streaming of sport events. This means that there is a constant stream of data that is being sent to the mobile device.

Due to the high bandwidth of a 3G connection, only a fraction of it will be used by the streaming service, the rest will be unused. Now this would not be a problem if not for the fact that the energy cost is more or less the same no matter the data rate (as we demonstrated in Section 3). Clearly, there is a potential for saving energy by fully utilising the available bandwidth when transmitting and being idle or in standby when not.

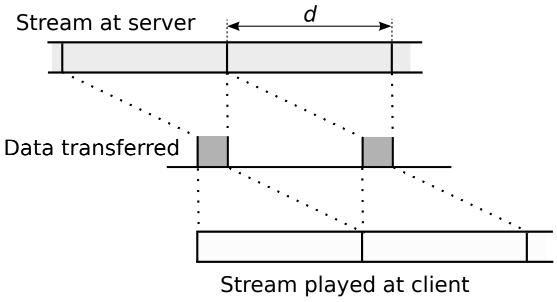

Figure 7: Energy-saving scheme for live streaming

We will now present the energy-saving scheme that can be used for any such live streaming service. Figure 7 shows the scheme where a live stream which is produced at the server side is divided into segments with duration *d*. These segments are transmitted in bursts over the data link and then played back at the client. Since the data is not arriving continuously, the client needs to have a buffer that can store an entire segment of the stream. Moreover, since a segment cannot be transmitted before it is produced, this means that the stream will be played at the client with a delay of at least *d*.

Figure 8 shows the resulting energy savings for different data rates on the x-axis and different segment durations *d* (corresponding to the delay with which the stream is played at the client) as separate curves. The upper curve shows the power consumption for a normal continuous data stream. We can see that for a typical Internet radio data rate of 96kbit/s it is possible reduce the power consumption by as much as 70% using this scheme.

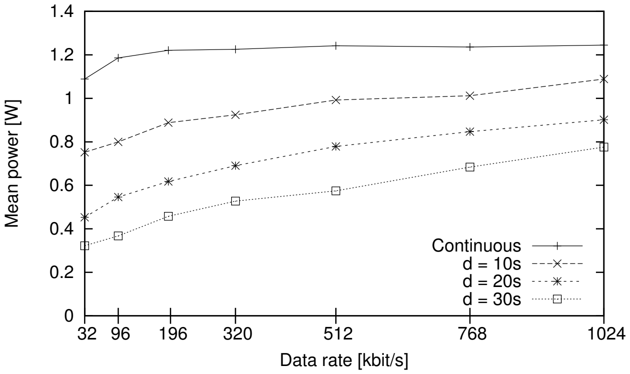

Figure 8: Energy savings due to buffering

Given the large savings that can be done, it is a bit surprising that this has not already been implemented in common streaming applications. Most such applications already use a buffer to deal with delayed and lost packets, and it would be very easy to extend this functionality for power-saving reasons. Note that fact that there is an additional delay in the playback does not mean that the user has to wait for that time before the stream starts playing. The server just needs to start by sending the last complete segment of the stream to the client.

6. OS AND APPLICATIONS

The two examples given so far were quite generic and based on synthetic data loads. We will now turn to the measurements we have performed on real applications and operating systems.

Starting with the latter, figures 9 and 10 show examples of the power consumption during two minutes of Ubuntu 10.10 and Windows 7 respectively. Both machines were clean with no applications running. As we can see, there is a substantial difference in the amount of network traffic that is being transmitted. When running Ubuntu, a short connection is done roughly every minute, making the broadband module switch temporarily to shared state, after which it returns to the standby mode.

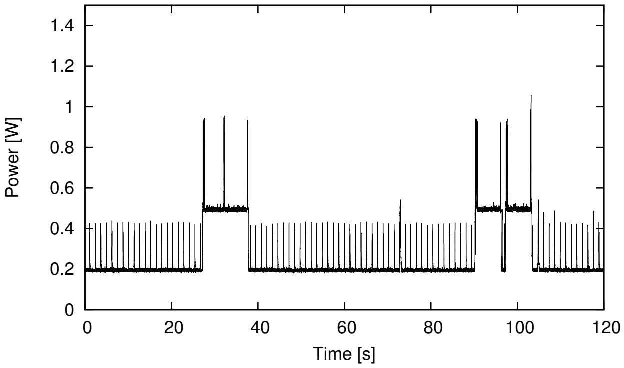

Figure 9: Power consumption example of idle Ubuntu

In Windows on the other hand, there are frequent connections (some even taking the device to a dedicated channel). This means that the device rarely goes to standby and then only for a brief period. The cause of these network accesses is network monitoring and domain management polls.

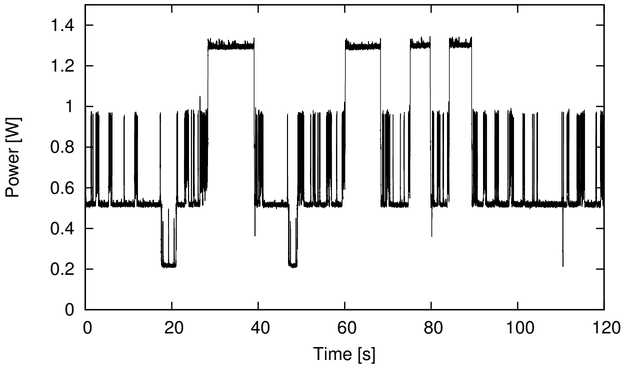

Figure 10: Power consumption example of idle Windows 7

As a result, the average power consumption during 10 minutes of idle time is 0.29W for Ubuntu and 0.79W for Windows. For a tablet with 25Wh of battery (as for example advertised for the iPad2) and 10 hour battery time, this 0.5W added power consumption would reduce the battery time of the device from 10 to 8 hours. We can see no justification for spending this excessive amount of energy for performing network accesses that has no real benefit for the user.

Finally, we have measured the impact of power consumption of some popular applications, Figure 11 shows the results. Beginning with the Internet telephony program Skype, the result is very much expected. Since voice communication requires a constant stream of data packets with low latency the device will use the dedicated channel resulting in an average consumption of 1.3W. Next, we tested with the music streaming service Spotify. Here the result was more surprising. Since the music in Spotify is not produced live, the user device should be able to download a segment of the song and then go to standby mode while the rest of the segment is played. Actually, this is what Spotify does, but still the connection stays in dedicated state. The explanation turned out to be the peer-to-peer component of Spotify. In order to make the content delivery more efficient, Spotify uses a peer-to-peer network to distribute content. The traffic to maintain this network and share the locally cached songs causes the device to stay in the dedicated channel. In order to test the energy impact of this traffic we performed an additional experiment where only the traffic to Spotify's servers were let through the firewall (thus excluding any peer-to-peer contacts). The result, displayed as "Spotify FW", shows that when only the actual music data is allowed to use the mobile connection, the energy cost reduces by half.

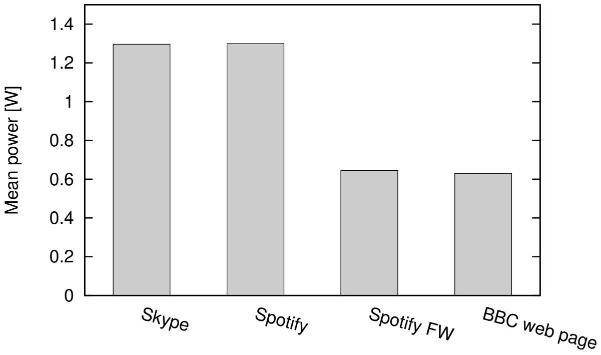

Figure 11: Power consumption of different applications

The rightmost bar shows the mean power usage when reloading the bbc.com web page every 30 seconds. Although this does not correspond to any real user behaviour, it gives a rough indication of the power consumption related to normal web surfing.

7. RELATED WORK

Recent years have seen a growing interest in measuring the energy consumption in mobile devices towards green networking. A typical setup for measurement exploits built-in software in mobile phones, e.g. Nokia Energy Profiler (NEP) [4, 11, 19, 10, 16]. Other works use external power metering hardware [17, 13, 6, 3], though some [10, 16] focus on WLAN infrastructure and adhoc modes.

Recently, mobile network operators have also become interested in the energy savings approaches [5]. However, new proposals often adopt an infrastructure-centric approach rather than considering a holistic approach including user terminals [8, 7, 9, 20].

The potential for power saving based on various notions of adaptation is well-recognised in the literature. Among some recent studies that propose adaptive schemes for sending, receiving or intermediate brokering of data streams we find the following works. On the transmission end, the notion of Delay-Tolerant Applications (DTA) captures the intuition that in order to reduce the energy consumption some applications can defer their transmissions depending on user demands. TailEnder, developed by Balasubramanian et al. [4], employs traffic shaping over cellular networks using an online scheduling algorithm for DTA. The system takes advantage of bursty transmissions and it also uses prefetching to predict future data accesses and save energy by fetching them in advance. Bartendr [14] exploits the fact that transmissions are less costly when the received signal is strong by transmitting data in periods of good signal strength in cellular networks. Bartendr decides to transmit DTA data based on a fixed signal strength threshold and modulates the traffic of the streams to download more data when strong signal is available. Armstrong et al. [2] present an energy-efficient Web browsing content update combining the reduction on transferred data by using caches and Web page updates via bursty data transmissions from a proxy. The pre-knowledge about the volume of data that has to be downloaded allows them to intelligently select which link to use (GPRS or WLAN).

While these works confirm the motivation for our work they are of less relevance to our actual measurements. The closest work is that presented by Balasubramanian et al. [4] in which NEP is used as a basis for measurement. Their methodology uses synthetic traffic with variations in data transfer sizes, inter-transfer time, geographical and temporal variations or mobility (horizontal handovers). We refine the results presented in that paper by 1) making physical measurements on a real cellular network module, and 2) extending the range of the measurements in the 3G case for data size.

Perrucci et al. [11] focus their study on the energy consumption of 2G and 3G according to common usage of the networks, such as SMS, voice calls and data traffic. Irrespective of the SMS size, and for 5 minute long voice calls, they show that UMTS consumes more energy (e.g., 50% more in voice calls). They suggest SMS compression for long messages, which could also be applicable to other types of data. However, the study shows that the energy per bit of 3G is significantly smaller, suggesting an intelligent vertical handover between 2G and 3G depending on the usage pattern and considering the energy cost of the handoff. Our work is based on isolated measurements of energy consumption for the data transfers as opposed to the combined CPU, memory access and other overheads that are included in a phone-based measurement.

A recent study by Wang et al. [17] presents the energy consumption of data transfers over WLAN and cellular networks (EDGE and HSPA). Their methodology is based on receiving/sending synthetic data traffic varying the packet size and the sending interval. They show that mobile devices would benefit from big packet sizes, high data rates and bursty transmissions to allow longer low power consumption periods in the interfaces.Their measurement setup is based in external hardware and a particular smartphone.

Another interesting study is performed by Rice et al. [13] where a fine-grained power measurement framework with automated test execution is used to quantify energy consumption of modern smartphones. They report that the power consumption differs between hardware and software generations and that in their particular location 2G was more consuming than 3G.

Our work confirms and extends the above works by 1) using cellular module and physical measurements isolating the energy consumption of data transfers, and 2) identifying the impact of system software and unsolicited network connections.

An application-centric energy study of mobile Youtube is performed by Xiao et al. [19] showing that videos downloads in WLAN is more energy efficient than in 3G based on the data transfer pattern of the application. They compare progressive download and download-and-play operations concluding that over WLAN their cost is the same.

Cool-Tether [15] is a WiFi access point focused on serving web pages using the cellular connection from smartphones. A cloud-based server fetches all the data of the Web page requests and sends it in a single burst to a certain number of smartphones that act as forwarders in order to speed up the transfer. They propose a reverse WLAN infrastructure mode for tethering where the smartphones act as clients delegating the energy consumption burden to other devices.

Stratus [1] implements concepts of Bartendr and Cool-Tether using aggregation to perform bursty transmissions based on a cloud-based proxy, scheduling transmissions in periods of strong signal reception and employing data compression between the proxy and the smartphone.

One of the most detailed studies of the tail effect in the 2G/3G cellular environment has been carried out by Qian et al. on actual data traces in late 2009 [12]. This work is an excellent source for understanding the operator-dependent timing configurations and their theoretical impact on user battery drainage. Again, the measurements that relate to our work are performed based on mobile phone battery discharges and not on the isolated upload/download by the wireless module. In addition, their streaming data was not live. Therefore, our insights on live streaming and unsolicited network connections are novel in this context.

8. CONCLUSIONS AND FUTURE WORK

Dealing with large volume of data transfers over wireless networks is a challenge to address by both the research community and the communications industry. To solve the problem by adjusting the cellular network settings at the operator end only does not solve the problem entirely. The end user battery life time has to be considered as part of a holistic solution. This paper contributes towards finding the holistic approaches by clarifying the end-user side of the equation using real measurements on a wireless module, thus isolating the energy footprint of cellular communication from the general battery drain in the device. The insights of the paper, are of two sorts. First, we have confirmed earlier knowledge on the importance of the channel states and the timers for the end user device energy consumption, by actually doing physical measurements. Thus, we can now motivate that more work on adaptation middleware on the user device end is worthwhile. We have done this study both using synthetic work loads and real applications (commonly used by smart phones and tablets).

Second, and to our knowledge for the first time, we have detected that as well as application level scheduling of traffic, the unsolicited (OS connections) and implicit background connections (like Spotify) make a difference for how long the user device battery lasts. We have quantified these measures using a modern GSM/3G module, and current applications in isolation from other factors (e.g. screen rendering and CPU/memory use on the device).

Work on energy-efficient wireless networking has only begun and there are several directions for future work in this area. The one most closely related to the measurements presented in this paper is an adaptation middleware that works on top of an OS. By using application and OS related information as well as cellular network state it will optimise use of the battery resource.

9. ACKNOWLEDGEMENTS

This work was financially supported by Linköping University, Institute of Technology and partially by the Swedish Research Council (VR). The authors wish to thank colleagues at Ericsson Radio Systems, Pär Emanuelsson, B-O Hertz, Ronny Holmberg and Claes Alströmer for providing the equipment and active support of the work.

10. REFERENCES

[1] B. Agarwal, P. Chitnis, A. Dey, K. Jain, V. Navda, V. N. Padmanabhan, R. Ramjee, A. Schulman, and N. Spring. Stratus: energy-efficient mobile communication using cloud support. In *Proceedings of the ACM SIGCOMM*, 2010. doi: 10.1145/1851182.1851272.

[2] T. Armstrong, O. Trescases, C. Amza, and E. de Lara. Efficient and transparent dynamic content updates for mobile clients. In *Proceedings of the 4th international conference on Mobile systems, applications and services (MobiSys)*. ACM, 2006. doi: 10.1145/1134680.1134687.

[3] Atheros. Power consumption and energy efficiency comparisons of wlan products, `http://www.atheros.com/media/resource/resource_15_file2.pdf`. White paper, 2004.

[4] 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 (IMC)*, 2009. doi: 10.1145/1644893.1644927.

[5] M. Etoh, T. Ohya, and Y. Nakayama. Energy consumption issues on mobile network systems. In *Proceedings of the 2008 International Symposium on Applications and the Internet (SAINT)*. IEEE Computer Society, 2008. doi: 10.1109/SAINT.2008.84.

[6] L. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001. doi: 10.1109/INFCOM.2001.916651.

[7] I. Humar, X. Ge, L. Xiang, M. Jo, M. Chen, and J. Zhang. Rethinking energy efficiency models of cellular networks with embodied energy. *IEEE Network*, 25(2), 2011. doi: 10.1109/MNET.2011.5730527.

[8] M. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo. Optimal energy savings in cellular access networks. In *IEEE International Conference on Communications Workshops*, 2009. doi: 10.1109/ICCW.2009.5208045.

[9] Z. Niu, Y. Wu, J. Gong, and Z. Yang. Cell zooming for cost-efficient green cellular networks. *IEEE Communications Magazine*, 48(11), 2010. doi: 10.1109/MCOM.2010.5621970.

[10] M. Pedersen, F. Fitzek, G. P. Perrucci, and T. Larsen. Energy and link measurements for mobile phones using ieee802.11b/g. In *Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2008. doi: 10.1109/WIOPT.2008.4586031.

[11] G. P. Perrucci, F. Fitzek, G. Sasso, W. Kellerer, and J. Widmer. On the impact of 2G and 3G network usage for mobile phones' battery life. In *Proceedings of the European Wireless Conference (EW)*, 2009. doi: 10.1109/EW.2009.5357972.

[12] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3g networks. In *Proceedings of the 10th annual conference on Internet measurement (IMC)*. ACM, 2010. doi: http://doi.acm.org/10.1145/1879141.1879159.

[13] A. Rice and S. Hay. Measuring mobile phone energy consumption for 802.11 wireless networking. *Pervasive Mobile Computing*, 6(6), 2010. doi: 10.1016/j.pmcj.2010.07.005.

[14] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking (Mobicom)*, 2010. doi: 10.1145/1859995.1860006.

[15] 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 *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT)*. ACM, 2009. doi: 10.1145/1658939.1658952.

[16] E. J. Vergara, S. Nadjm-Tehrani, M. Asplund, and U. Zurutuza. Resource footprint of a manycast protocol implementation on multiple mobile platforms. In *Proceedings of The Fifth International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*. IEEE, 2011. To appear.

[17] L. Wang and J. Manner. Energy consumption analysis of wlan, 2g and 3g interfaces. In *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom), International Conference on Cyber, Physical and Social Computing (CPSCom)*. IEEE Computer Society, 2010. doi: 10.1109/GreenCom-CPSCom.2010.81.

[18] P. Willars. Smartphone traffic impact on battery and networks, `https://labs.ericsson.com/developer-community/blog/smartphone-traffic-impact-battery-and-networks`, 2010.

[19] Y. Xiao, R. Kalyanaraman, and A. Yla-Jaaski. Energy consumption of mobile youtube: Quantitative measurement and analysis. In *Proceedings of The Second International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, 2008. doi: 10.1109/NGMAST.2008.26.

[20] J. Zhou, M. Li, L. Liu, X. She, and L. Chen. Energy source aware target cell selection and coverage optimization for power saving in cellular networks. In *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom), International Conference on Cyber, Physical and Social Computing (CPSCom)*, 2010. doi: 10.1109/GreenCom-CPSCom.2010.56.