International Journal of Science and Engineering Research (IJ0SER),

Vol 2 Issue 9 September -2014

Humanizing Power Saving Mode by the AITM Window

S.Ajith kumar

Department of Computer Science and Engineering

VLB Janakiammal College of Engineering and Technology, Tamilnadu

India

***Abstract*— This paper describes a collective power and throughput performance revision of Wi-Fi and Bluetooth usage in smartphones. The Existing paper divulges several fascinating occurrences and tradeoffs. But, the need for a distributed enactment of a PSM for ad-hoc networks, so we propose, Distributed PSM appears in the standard of IEEE 802.11 with additional sweats to enhance the standard enactment. Improving the IEEE 802.11 power saving mode (PSM) using the AITM window, yet, as our measurements indication, PSM is not effected by the phone's Wi-Fi card driver when connected to an ad-hoc network**.

***Keywords*— PSM, AITM window, IEEE820.11.**

I.INTRODUCTION

Smartphones are rapidly becoming the main computing and (data) communication platform. These days, smartphones are all prepared with Bluetooth and Wi-Fi, which balance their cellular communication capabilities. Bluetooth was formerly placed in mobile phones for personal-area communication, such as wireless earphones, synchronization with a nearby PC, and tethering. Wi-Fi was added more recently in order to improve the users' Web surfing experience whenever a Wi-Fi access point is available. In fact, new market researchers predict that between 2012 and 2014, depending on the source, WiFi equipped smartphones will outnumber all other WiFi enabled devices combined (laptops, tablets, WiFi enabled TVs, etc.). When examining the foreseen usage patterns of WiFi on smartphones, it appears that in addition to fast and possibly free Internet access (including VoIP and video), direct communication between nearby devices is of emergent interest.

Noticeable examples include media streaming either between smartphones or between a phone and another nearby wireless device (TV or computer) in a home or office setting. This is exemplified in the superfluity of new WiFi based streaming solutions, as well as being one of the main enthusiasms behind the WiGig initiative. Another scenario includes ad-hoc social networking and communication, such as iPhone's iGroups, Nokia's Instant Community, Mobiles, and WiPeer, to name a few. Such local communication can be theoretically performed either over WiFi or over Bluetooth. Some could be performed while relying on a mutual nearby access point (AP), while others might be more natural for WiFi ad-hoc mode. When considering these substitutions, important considerations include the obtainable throughput and power consumption of each.

As Bluetooth was deliberate for personal area communication, its transmission range is much shorter than WiFi. Hence, comparing the two is eloquent only when the devices are close enough so that both can be used. Given that Bluetooth also offers lower bandwidth, one might ask why inconvenience with it at all. The reason to still consider Bluetooth is that previous power studies have suggested that Bluetooth is much more power efficient than WiFi [1]–[4], which motivated multiple works on clever combinations of both Bluetooth and WiFi [1], [3], [5]–[7]. This issue is acute given that wireless communication is one of the main battery drainers in a mobile phone [1], [8]. It is important to notice that both a Bluetooth interface and a WiFi interface can be in multiple modes, or states, such as transmitting, receiving, connected but idle, disconnected and hidden, disconnected and seeking, etc. (the exact names in each technology are specified later in this work); each of these states may consume different power levels. One of the main criticisms of WiFi in past works has been that its power consumption in idle mode is on the same order of magnitude as sending and receiving [1], [8]–[11].

Also, as we just mentioned, Bluetooth was reported to be at least an order of enormousness more power efficient than WiFi [1]–[4]. Yet, when examining these works we have revealed some shortcomings, which have motivated this study: First, many of these studies were performed several years ago, and given the rapid change in technology, one cannot trust that they endure valid. Second, in most of these studies, each interface was measured individualistically and in isolation in a lab setting. Hence, it is not clear that the results hold for a mobile phone, in which the two RF boundaries are packed in close proximity, sometimes on the same chip. Moreover, from the user's point of view, it is interesting to know the total power consumed for a given operation due to such communication, including, e.g., code that might be running in the Kernel of the OS, freestanding the wireless interface. Finally, some of the more recent works that did quantity modern smartphones either did not examine the different states of WiFi and Bluetooth, concentrated only on WiFi, and/or did not look at the back-and-forth between bandwidth and power.

In this our work is Existing paper exposes several stimulating occurrences and tradeoffs, The conclusions from this study recommend preferred usage patterns, as well as operative recommendations for researchers and smartphone inventers. A power and throughput study of WiFi and Bluetooth on modern smartphones, in which we examine the relationship between obtained throughput and power, and the power consumption in the various states of the wireless interfaces. In the process, we determine several interesting singularities (some counter previous conventions) and draw some practical recommendations for imminent academic research and practical development.

But, the need for a distributed enactment of a PSM for ad-hoc networks, so we propose, Distributed PSM give the impression in the standard of IEEE 802.11 with additional sweats to enhance the standard enactment, the power saving mechanism (hereafter referred as PSM) for DCF in an IBSS. Time is alienated into beacon intervals. At the start of each beacon interval, each node stays wakeful for an ATIM window interval.

II.RELATED WORK

Energy efficiency is familiar as a dominant property of any mobile device, in particular for smartphones. Consequently, numerous papers try to address the question of where and how the energy is expended. Carroll and Haiser [12], for example, design a set of micro-benchmarks to unconventionally associate the power costs with a particular part of the smartphone system, such as CPU, RAM, WiFi, Audio, etc. in aOpenmoko Neo Freerunner phone. While their work considers multiple wireless interfaces available on the devices under test, the interfaces are not profiled beyond enabled and disabled mode, and only their power cost in various general benchmarks is considered. Few recent works examine energy consumption of the WiFi interface in mobile phones.

Compare between cellular interfaces, such as 3G, GSM, and WiFi. They utilize OS-specific methods to collect information on power usage, such as Nokia Energy Profiler. In comparison, our framework is generic and can be used virtually with any mobile phone.

Additionally, the (802.11b) WiFi interface considered in [13] is outdated and is profiled only in access point mode of operation. In another work, Xiao et al. [11] model and measure the power consumption of 802.11g WiFi interface on three mobile phones. Their experiments also consider only access point operation mode and are conducted with TCP traffic only. Power measurements of a WiFi card in various modes are reported in [9], [14] (more than a decade ago). Unlike these works, we have chosen to measure the performance of wireless interfaces that are integrated on-board. We believe that for an end-user, the performance of the integrated radio in different network settings (including various hidden costs, e.g., running kernel code and copying buffers) is even more relevant than the performance of a particular wireless card.

III.POWER SAVING MECHANISM IN IEEE 802.11

The power saving mechanism (hereafter referred as PSM) for DCF in an IBSS. Time is divided into beacon intervals [7]. At the start of each beacon interval, each node stays awake for an ATIM window interval. We describe the power saving mechanism using Fig. 1. In PSM, when any node has a packet destined for another node, the packet is announced during a subsequent ATIM window. For instance, in Fig. 1, node A announces a packet destined for node B by transmitting an "ATIM frame" during the ATIM window. The transmission of an ATIM frame is performed using the CSMA/CA (collision avoidance) mechanism specified in IEEE 802.11. When a node has sent an ATIM frame to another node, such as node A in our example, the node remains awake for the entire beacon interval. A node that receives an ATIM frame replies by sending an ATIM-ACK. Such a node remains awake for the entire beacon interval, after transmitting the ATIM-ACK.

In our example, node B sends an ATIM-ACK to node A and remains awake for the rest of the beacon interval. Transmission of one or more data packets from node A to node B can now take place during the beacon interval, after the end of the ATIM window.

A node that has no outstanding packets to be transmitted can go into the doze state at the end of the ATIM window if it does not receive an ATIM frame during the ATIM window. In Fig. 1, node C dozesafter the ATIM window, thus saving energy. All dozing nodes again wake up in PSM at the start of the next beacon interval.

AITM WINDOWS

In PSM specified in IEEE 802.11, all nodes use the same (fixed) ATIM window size, as well as identical beacon intervals [7]. Since the ATIM window size critically affects throughput and energy consumption, a fixed ATIM window does not perform well in all situations, as shown in [9]. If the ATIM window is chosen to be too small, there may not be enough time available to announce buffered packets (by transmitting ATIM frames), potentially degrading throughput. If theATIM windowis too large, therewould be less time for the actual data transmission, since data is transmitted after the end of the ATIM window, again degrading throughput at high loads. Large ATIM windows can also result in higher energy consumption since all nodes remain awake during the ATIM window. In particular, at a low load, large ATIM windows are unnecessary. Thus, a static ATIM window size cannot always perform well. We propose a dynamic mechanism for choosing an optimal ATIM window size.

IV.IMPLEMENTATION

A. *Power in Non-Communicating Modes*

The power consumption of the Bluetooth and WiFi radios in modes of operation that do not involve communication performed by a user. We refer to these modes as *non-communicating*, although in some cases, radios may transmit some packets (e.g., when searching for networks). In the base configuration, when all radios are off, all phones except for Galaxy have a fairly stable power consumption. Power fluctuations in the base configuration, reflected by the standard deviation, can be probably associated with the periodic scanning of the touch-screen performed by the phones. These fluctuations in the form of small peaks can be seen on Fig. 2aproduced for the Omnia phone. Our hypothesis is supported by the fact that the peaks are gone when the screen is turned off.

B. *Bluetooth*

We consider three configurations of the non-communicating Bluetooth radio: (1) discoverable mode, where other Bluetooth-equipped devices may discover the phone when inquiring for nearly located devices, (2) non-discoverable mode, and (3) scanning mode, when the phone searches for other near devices. In the discoverable mode, the additional power is required to listen to the inquiry requests sent by nearby devices. Still, the consumption of the Android-based phones is considerably higher than that of Windows Mobile-based. On the other hand, while the consumption of the Windows Mobile-based phones remains the same in both discoverable and non-discoverable modes, the Android-based phones almost do not use any additional power in the non-discoverable mode.

The instant power consumption of Omnia when its Bluetooth is turned on. It is worth noting that the Android based phones limit the duration of the discoverable mode to 120 seconds, after which the phones automatically switch to the non-discoverable mode, while the Windows Mobile-based phones may stay in the discoverable mode with no limit. These findings suggest that the two operating systems employ different approaches to Bluetooth radio management.

The Windows Mobile-based phones operate the radio in the same state, requiring same additional power, without relating to whether the device was set as discoverable or not. Differently, the Android-based phones are optimized to change the physical state of the radio based on the configuration of the ability to be discovered. In the common case, Bluetooth discovery is required rarely and for certain periods of time (e.g., when a user wishes to connect to the phone from her laptop or earphone, etc.).

Thus, in the long run, the approach employed by the Android-based phones is more power-efficient than the one employed by the Windows Mobile-based phones. We should note the high fluctuations of power costs in both discoverable and non-discoverable modes as measured with the Diamond phone. Although the Galaxy phone also has an unstable power cost, the standard deviation reported for the Diamond phone is exceptionally high. We did not find any acceptable explanation for this phenomenon, except for noticing that in several other configurations as discussed below, Diamond also has much worse power management when compared to other tested smartphones.

In the scanning mode, all phones, except for Diamond, consume a similar amount of additional power, while this amount (for all phones) is considerably higher than in other previously discussed modes. The power consumption of Diamondis outstandingly high. This could be related to the particular chip design on which the Bluetooth radio is located. Based on these results, we conclude that the scanning mode of Bluetooth in smartphones is very power-demanding and should be used very selectively. Several other works [4], [7] confirm our findings about these differences.

C. WiFi

For WiFi radio, we consider four non-communicating modes: (1) not-connected mode, where the radio is on, but not connected to any network, (2) searching mode, in which the radio scans for available networks, (3) connected to access point mode, and (4) connected to ad-hoc network mode.

The clear distinction between the first two modes exists only at the Omnia phone. In this phone, the WiFi radio in the first mode consumes almost no additional power, while in the second mode the consumption increases drastically. We were able to determine these modes by turning the access point on and off and noticing that the phone updates the availability of the access point only when its power consumption was raised. The transition between these two modes is caught in the right side of Fig. 2.

We note that we did not find neither regularity in the time intervals spent by the radio in each of the states nor any way of configuring these intervals.

D. Measurements with iperf

In this section, we report on results achieved with iperf, a standard tool for network performance measurements [15], which we have ported to Windows Mobile and extended to support Bluetooth communication. The technical details of these enhancements can be found in [16]. As iperf is not yet ported to Android, the experiments with this tool were conducted only on the Omnia and Diamond phones.

1) *Max Throughput Experiments:* For WiFi, we experimented with both UDP and TCP. For Bluetooth, we experimented with RFCOMM (the only protocol supported by iperf on Windows Mobile for Bluetooth communication [16].

This is not a real limitation, though, since RFCOMM, being a robust general-purpose and reliable protocol is the primary choice for any application requiring Bluetooth communication). In all experiments, we have used a Lenovo T61 laptop running Windows XP Service Pack 3 as a corresponding peer for communication with the phones. In addition, for access point network configurations, we used a Linksys WRT54GL wireless router. In all experiments with UDP, iperf was configured to send packets at a rate of 54Mb/s, which is the theoretical 802.11g link bandwidth. This is in order to achieve the maximal available link throughput. In the case of TCP and RFCOMM, iperf always strives to achieve the highest possible rate, which is ultimately controlled by these protocols' internal congestion and flow control mechanisms.

V. CONCLUSION

As mentioned in the Introduction, WiFi enabled smartphones are becoming tremendously popular. This poses a great impending for smartphone-based ubiquitous mobile ad-hoc networking, including new collective and cooperative applications that may utilize physical proximity between communicating devices (see, for example, Apple's iGroups and Nokia's Instant Community). The need for a distributed enactment of a PSM for ad-hoc networks. Distributed PSM give the impression in the standard of IEEE 802.11 [7] with additional determinations to enhance the standard enactment [2]–[3]. Yet, as our quantities show, PSM is not implemented by the phone's WiFi card driver when connected to an ad-hoc network. In future work is hindrance is partially addressed in recent works on WiFi availability extrapolation [7], [4]. The idea is to avoid penetrating for available networks in places that are known to be lacking access point coverage, based on location information or other available radios (e.g., Bluetooth [7]). This approach, however, is inappropriate for ad-hoc networks, which can be created fundamentally anywhere and anytime.

REFERENCES

[1] T. Pering, Y. Agarwal, R. Gupta, and C. Power, "Coolspots: Reducing the power Consumption of wireless mobile devices with multiple radio interfaces," in *Proc. ACM MobiSys*, 2006, pp. 220–232.

[2] P. Bonnet, A. Beaufour, M. B. Dydensborg, and M. Leopold, "Bluetooth based sensor etworks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 35–40, 2003.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Trafficsense: Rich monitoring of road and traffic conditions using mobile smartphones," Microsoft, Technical Report MSR-TR-2008-59, 2008.

[4] J.-C. Cano, J.-M. Cano, E. Gonzalez, C. Calafate, and P. Manzoni, "Power characterization of a bluetooth-based wireless node for ubiquitous computing," *Proc. IEEE ICWMC*, p. 13, 2006.

[5] R. Friedman and A. Kogan, "Efficient power utilization in multi-radio wireless ad hoc networks," in *Proc. OPODIS*, 2009, pp. 159–173.

[6] G. Ananthanarayanan and I. Stoica, "Blue-Fi: enhancing Wi-Fi performanceusingbluetooth signals," in *Proc. MobiSys*, 2009, pp. 249–262.

[7] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: an event driven energy saving strategy for battery operated devices," in *Proc. ACM MOBICOM*, 2002, pp. 160–171.

[8] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc

networking environment," in *Proc. IEEE INFOCOM*, 2001, pp. 1548–1557.

[9] P. Bahl, A. Adya, J. Padhye, and A. Walman, "Reconsidering wireless systems with multiple radios," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 5, pp. 39–46, 2004.

[10] Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, and A. Yla-Jaaski, "Practical power modeling of data transmission over 802.11g for wireless applications," in *Proc. 1st Int. Conf. on Energy-Efficient Computing and Networking (e-Energy)*, 2010, pp. 75–84.

A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of the 2010 USENIX Technical Conference*, 2010.

[11] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proc. ACM IMC*, 2009, pp. 280–293.

[12] NLANR/DAST, "Iperf," Available at http://sourceforge.net/projects/iperf.

[13] MSDN, "TCP/IPv4 Configurable Registry Settings," Available at http://msdn.microsoft.com/en-us/library/ms884977.aspx.

[14] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "802.11 powersaving mode for mobile computing in Wi-Fi hotspots: limitations, enhancements and open issues," *Wireless Netw.*, vol. 14, pp. 745–768, 2008.

[15] C. Sengul, A. F. H. III, and R. Kravets, "Reconsidering power management," in *Proc. IEEE BROADNETS*, 2007, pp. 799–808.

[16] E.-S. Jung and N. H. Vaidya, "Improving IEEE 802.11 power saving mechanism," *Wireless Netw.*, vol. 14, no. 3, pp. 375–391, 2008.