GPS Tethering for Energy Conservation[1]

Yuepeng Qi†, Chansu Yu‡, Young-Joo Suh¶ and Si Young Jang¶

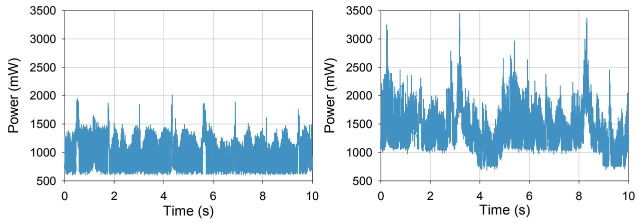
† Dept. of CS, North Carolina State University, Raleigh, NC 27695
‡ Dept. of ECE, Cleveland State University, Cleveland, OH 44115
¶ Division of ITCE & Dept. of CSE, POSTECH, Pohang, S. Korea
Email: yqi@ncsu.edu, c.yu91@csuohio.edu, {yjsuh, spencer}@postech.ac.kr

Abstract — Due to the low-speed GPS transmission and poor signal quality, localization is considered an expensive operation in battery-operated smartphones. This is even worse for apps that need a single location fix such as Foursquare and Flickr rather than continuous navigation because it cannot exploit the less-demanding "warm start" mode of GPS operation. This paper proposes GPS-Tether (GTR) which shares and reuses the known GPS coordinates among mobile devices in the proximity. Benefiting from unremitting Wifi discovery behaviors of smartphones, GTR does not need an additional protocol and does not disrupt operations of non-GTR clients. Experiments based on Samsung Galaxy smartphones and Monsoon Power Monitor show that GTR can extend smartphones battery usage as much as 19% compared to Assisted GPS (A-GPS). One important design goal is that it should require minimal changes for the implementation of GTR, which has been accomplished by incorporating the Wifi virtualization technique.

I. Introduction

Localization of a mobile device is an essential component in numerous *location-based services* (LBS) [1]. A recent study shows that one third of social media users tag their posts with their location and three quarters of smartphone owners get directions or other information based on their current location [2]. Interestingly, a large number of non-LBS apps use location information although they do not appear so. For example, Rovio's *Angry Birds* includes the third-party *Flurry ad network* that uses GPS to identify the user's location, which consumes 7% of the power consumed by the app [3]. Forecasted by latest market researches, LBS market is estimated to grow from $8.12 billion in 2014 to $39.87 billion in 2019 [4].

However, frequent GPS sensing could significantly shorten the battery lifetime of a smartphone [5, 6]. For example, our measurement shows that Samsung Galaxy S4 consumes 65% more energy when GPS interface is on as shown in Fig. 1. This is mainly due to (i) the low speed communication (50bps), which refers to inability of duty cycling, (ii) poor signal quality, which demands heavy signal processing [7], and (iii) the increasing trend of GPS usage in mobile devices. Referring to the last point, mobile apps such as *Foursquare* or *Flickr* perform a single location fix rather than continuous navigation as seen in a conventional GPS receiver. While the latter gets benefitted from location fixes during the immediate past, sporadic location fixes in the former demand searching and receiving signals from satellites anew and thus imply a higher power consumption as well as a longer delay. *Assisted GPS* (A-GPS) shortens *the time-to-first-fix* (TTFT) by having the world-wide network of location servers (operated by, for example, Qualcomm, Broadcom or Google) to provide GPS data through the 3G/LTE [7, 8]. But it is known to consume more power (although it provides a faster fix) and is not desirable on battery-operated mobile devices [6, 8, 9]. We will discuss A-GPS in more detail later in this paper.



(a) GPS off (b) GPS on

Fig. 1. Power consumption in mW for 10-sec period (Samsung Galaxy S4 with Google Map running)

This paper proposes an energy-efficient GPS tethering mechanism, called *GPS-Tether* (GTR), which conserves battery power of a smartphone (GTR client) by opportunistically retrieving GPS coordinates from its neighboring smartphone (GTR server) rather than activating its own GPS interface. This is similar to A-GPS but the help comes from the proximity and without incurring the complexity corresponding to the dedicated network protocols and the world-wide deployment of location servers mentioned above. It is noted that GTR should not be confused with Google's *Tether GPS* [10]. While the former is implemented mostly at a link layer, the latter uses a TCP connection and is to support mobile devices that do not

[1] This research was supported in part by the US National Science Foundation under Grants CNS-1338105 and DUE-1245900, and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2065379).

978-1-4799-8406-0/15/$31.00 ©2015 IEEE

have a GPS receiver such as tablets.

Contributions of this paper are two-fold:

- First, GTR uses the conventional *Wifi discovery procedure* to deliver GPS coordinates, which is the first of this kind in the literature and has the following two advantages. (i) It incurs no or minimum extra energy consumption on the client and the server side and (ii) it does not assume any protocol addition and is compatible with existing industry standards.
- Second, this paper considers the problem of rendezvous of two Wifi devices. In the context of GTR, one assumes the role of AP (GTR server), more correctly, *mobile AP* or *mobile hotspot*, and the other is a client (GTR client). In outdoor environment with no Wifi infrastructure, each client searches for an available Wifi AP at all channels on a regular basis, constantly triggering the Wifi discovery procedure. To make sure a mobile AP meets with the client without being awake all the time is a challenging problem. This becomes more complicated different smartphones use different searching patterns and intervals (see Table 1 in Section III) but may be useful in other smartphone apps.

Our evaluation based on prototyping using Samsung Galaxy Nexus and Galaxy S4 smartphone shows that GTR saves up to 19% energy compared to A-GPS. It is noted that smartphones may not be willing to help others and would not offer GPS information without an adequate reward. This is an important subject but is beyond the scope of this paper. Please refer to [11] for the corresponding discussions.

The rest of this paper is as follows. Section II briefly introduces GPS localization and explains the mechanism of A-GPS technology in smartphones. In section III, GTR scheme is proposed, followed by its implementation and evaluation in section IV. Finally section V concludes the paper and presents the future work.

II. Background and Related Work

A. Global Positioning System (GPS)

GPS is a satellite-based navigation system that can be used to identify the location of a GPS receiver [12]: (i) Sixteen *Monitor stations* distributed around the world continuously collect transmissions from the GPS satellites and send them to the *Master control station*. (ii) The Master control station, located at Colorado Springs, USA, computes satellite orbits (called *ephemerides*) and clock errors to generate the *Navigation message*. (iii) Four *Ground antennas*, located at North America, Atlantic Ocean, Indian Ocean, and Pacific Ocean, upload the Navigation message to each satellite three times a day (or once a day). (iv) More than thirty satellites orbit Earth twice a day. They simultaneously and continuously broadcast the Navigation message. (v) A smartphone (GPS receiver) determines its position using the Navigation messages it receives from the satellites.

Since each Navigation message consists of five 300-bit sub-frames transmitted over a slow 50bps link, one complete transmission takes 30 seconds. The first frame (subframe 1 in Fig. 2) contains timing information, the next two frames (subframe 2 & 3) contain ephemeris data which gives the satellite's own precise orbit and is generally valid for 30 minutes. The last two frames (subframe 4 & 5) contain *almanac* data which refers to coarse orbit and status information of all satellites and is valid for up to several months. Due to the number of satellites around the Earth, it takes 25 frames or at least 12.2 minutes to obtain the whole almanac data. Note that this does not happen at a GPS receiver because it only sees a subset of satellites. But it still needs 12.2 minutes because it has to tune its receiver to each of the freq/codes one after the other.

When a smartphone does not have any prior information about GPS coordinates, TTFF (*time-to-first-fix*) would be at least 12.2 minutes to identify its current position (*Cold start*). But typically, it has some GPS data from the recent past and thus is able to search a known subset of satellites to receive the Navigation messages. TTFF would be several tens of seconds to several minutes (*Warm start*).

B. Assisted GPS (A-GPS)

A-GPS has been developed to reduce TTFF, which is particularly important in numerous smartphone apps. A mobile device with A-GPS enabled can download GPS data through the cellular network. Typically, the A-GPS servers can obtain a position estimate of the mobile device through the cell tower where the mobile device is positioned. With that position estimate, the A-GPS server knows which satellites should be visible in the view of that mobile device, and sends GPS data to the mobile device. The mobile device utilizes this GPS data first and switches to real data as it becomes available. However, A-GPS has several shortcomings. First, it could be more energy consuming than GPS itself [6, 8, 9]. Second, it uses costly cellular data rather than free GPS data. Third, the A-GPS architecture depends on layering on the OS design and the cellular networking [7, 8]. Fourth, location privacy can easily be compromised by contacting location servers.

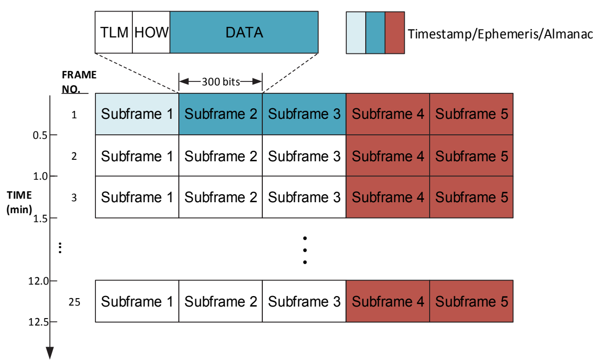


Fig. 2. GPS Navigation message

C. Related Work on Energy-Efficient GPS Localization

A high energy GPS operation has been one of primary concerns in battery-powered mobile devices particularly due to the popularity of location-based services (LBS) in recent years. As discussed in Introduction, this high energy is contributed by low speed communication (50bps) and intensive signal processing. This has been addressed in the literature, which essentially aims at minimizing the use of GPS receiver based on application-level hints, mobility analysis, and neighbors' help. Application-level hints are useful, for example, when multiple LBS apps are executing on a single smartphone and requesting location information individually. Whenever possible, an earlier request is delayed and is combined with a later one, which minimizes the total number of location requests [13].

Numerous sensors such as accelerometer, compass sensor, Wifi SSID (Service Set ID), and cell ID can be used to analyze user mobility, which help to use GPS infrequently [14-16]. A basic idea is to request GPS operation only when distance moved or time traveled exceeds a specified threshold. But it may incorporate uncertainty-aware estimation algorithm [14], interpolation mechanism [15] and/or contextual information such as places to visit and travel path [16]. In certain cases, no GPS receiver is used but just cheap sensors are used for localization to save more energy. They use accelerometer and magnetometer sensors and cell tower ID for the estimating the user location [17, 18]. Wifi triangulation, Bluetooth vicinity, and cell-tower association can be applied for positioning [5, 19].

Sharing location information among the neighbors is another direction for achieving energy efficient localization. In its simplest form, a mobile device receives GPS signal and estimates its location, which is shared by its neighbor [10, 20-21]. This sharing can be done via Bluetooth [20] or Wifi [10]. In its more complex form, a set of portable devices in close proximity is cooperative to determine their position or improve an estimate of their position by sharing the raw positioning data [13]. Another set of studies exploits the fact that for any two GPS receivers within the distance of 150 km, the propagation delay is the same in its integer value in millisecond but not in values in sub-millisecond. Energy-constraint device receives the sub-millisecond part of the propagation delay from the GPS but fetches the integer part from other receiver, which can be either cell-tower [22, 23] or cloud service servers [7].

III. Energy Efficient GPS Tethering

This section proposes an energy-efficient GPS localization mechanism in a reasonably populated area, called *GPS-Tether (*GTR*)*. As mentioned in Introduction, the basic idea is to share GPS data among smartphones in the proximity by utilizing the *Wifi discovery procedure*. More specifically, a mobile device that estimates its position based on signals from GPS satellites becomes a server *(GTR server)*, elects itself to become a mobile AP (mobile hotspot), and is ready to provide position information whenever another mobile device asks for it. One

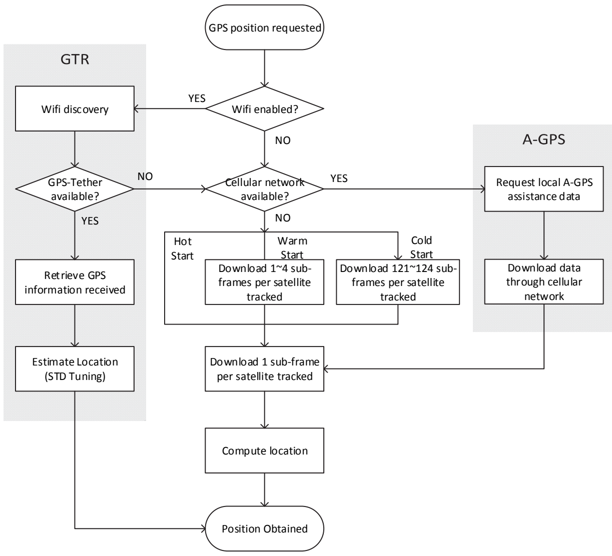


Fig. 3. Positioning mechanism including GPS, A-GPS and GTR

that needs position data *(GTR client)* looks for the opportunity to acquire the shared position information from other devices. If unsuccessful, it turns on its GPS interface to acquire GPS signals and to become a GTR server itself. Please see Fig. 3 for the overall process of obtaining position information.

A. Design of GTR

One of design goals of GTR is to implement it with little changes on the client devices. To achieve this goal, GTR employs *Wifi virtualization* as shown in Fig. 4. Note that conventional Wifi functionality is implemented in an original Wifi module. This allows a simpler design because *Virtual Wifi* executes most of extra operations for GTR. As shown in the figure, the GTR architecture consists of *Ghost AP* (GAP), *Ghost client* (GCL), STD Tuning (*signal, time and distance tuning*), and *Duty cycle control*. GAP retrieves position information from the Location Manager in Android and passes this information to GCL of another mobile device.

For compatibility reasons, position information is piggybacked on *probe request* (PRRQ) and *probe response* (PRRP) management frames which implement the Wifi discovery procedure. To conform to the Wifi standard, it uses *vendor specific information field* in the two management frames (PRRQ and PRRP). Each vendor-specific information element carries OUI (*organization identifier*) and contents. The number and size of vendor-specific information elements is limited only

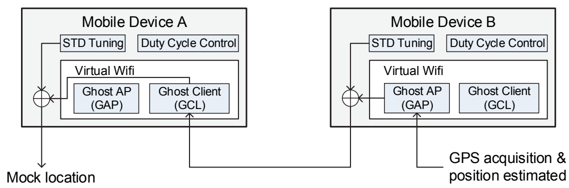


Fig. 4. GTR architecture

by the maximum frame size. It is observed that many Wifi devices including Apple's Aironet and Broadcom router utilize this field for their own specific needs. The proposed GTR algorithm uses this field and thus does not disrupt other clients which does not understand this particular OUI and the corresponding contents. Moreover, since no real association is built up with a GPS server, it doesn't need to disconnect with a previous AP if there is any.

B. STD Tuning

STD tuning addresses the following issues. (i) Signal strength of PRRQ and PRRP - Error distance of GPS is around 10~50m. When position information is obtained from a neighboring mobile device, error distance could reach 150m because Wifi signal travels around 100m in outdoor environment. Therefore, signal strength should be taken into account when considering to share at the GTR server side. At the GTR client side, signal strength can be used to compare and choose a better position data when there exist multiple GTR servers. (ii) Time - GTR server does not continuously acquire GPS signals. It does not pass position information when GPS acquisition and the corresponding position estimation occurs long before. On the other hand, it may simply pass this information and leave the decision up to the GCL. (iii) Distance - Mobile devices incorporate many different kinds of sensors, some of which can be used to estimate the distance traveled from a certain past. GAP may not want to pass the position information if it travels much after the GPS acquisition. However, it could also lead this decision up to the GCL of the other mobile device.

C. Duty Cycle Control

One of the challenges in the design of GTR is that it increases the energy cost of GAP because it has to make its Wifi interface active for a long duration because it does not know when other mobile devices request for position information. Duty cycling is the mechanism that inactivates GAP's Wifi interface to reduce the energy cost while ensuring the rendezvous with any potential clients (GCL). It is based on the scanning pattern employed in the Wifi discovery procedure.

Active scan. A main component in Wifi discovery procedure is *Active scanning*. This has been heavily studied in the literature for an efficient handoff [24]. A caveat is that there are multiple Wifi channels to scan and there can be zero or multiple APs on each of the channels. A client broadcasts PRRQ into a channel first. Receiving the PRRQ, APs send PRRP to the client. The client waits for the PRRPs until a limited channel access time, called MinCT expires. If it receives at least one PRRP, the channel access time is extended to MaxCT. This process repeats for each Wifi channel. IEEE 802.11 standards imposes no value on those parameters and no restrictions on scanning algorithm [25, 26] but MinCT and MaxCT are typically 10ms and 30ms, respectively.

Table 1. Frequency of active scanning

Device	Group interval	Number of repeats per group	Gap between repeats
iPhone 4	45s	7	40ms
iPhone 5	45s	7	40ms
Galaxy Nexus	30s	10~12	20ms
Galaxy S4	300s	12(indoor) 4(outdoor)	40ms

* Note that, with display on, they scan more frequently. Group interval, number of repeats per group and gap between repeats are 10s (varies), 4, 20ms for iPhone 4. They are 2^nms, 7 and 20ms for iPhone 5, 15s, 7 and 20ms for Galaxy Nexus, and 5/10s, 6, and 40ms for Galaxy S4.

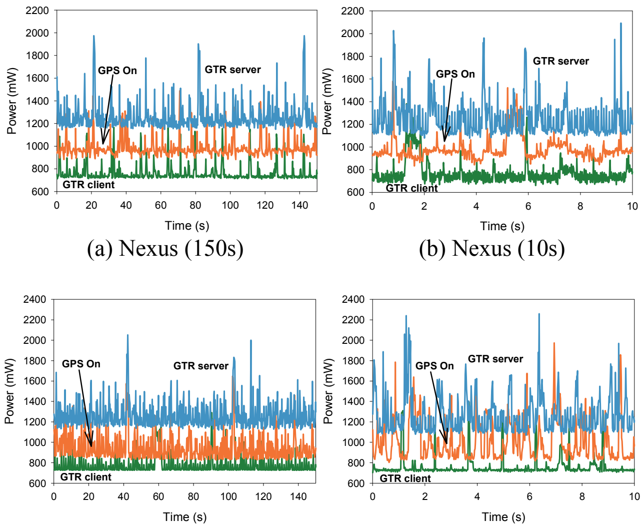
Duty cycling. It is observed that scanning does not always start from channel 1, is not always sequential, does not always spend the same time at every channel, and is different from devices to devices [21, 27]. As shown in Table 1, it is performed with an interval and with the number of repeats per scan. For example, iPhone 5 (with display off, outdoor environment) repeats active scanning (~0.24s) and sleep (~44.8s) every 45 seconds. During the active scan, it scans all 11 (or 13) Wifi channels, which is repeated 7 times considering that it may miss some PRRPs due to interference or collisions. There exists the interval of 40ms between two repeats.

Since the GTR design wishes to pass position information between devices in the close proximity (relatively strong signal and not fragile), it is assumed that only one scan suffices, making the GAP to sleep to conserve energy. Therefore, for ensuring that GAP rendezvous with a GCL request, the duty cycle that GAP should employ is: sleeps for 200ms and wakes up for 40ms, which would continue no more than a few minutes.

Duty cycle optimization. Duty cycling can further be optimized with specific SSID. It is known that some smartphones (like iPhone 4, Galaxy Nexus) remember Wifi networks they have associated with, and prioritize them in the next scan. When it comes to scanning a remembered SSID, those smartphones wait shortly for corresponding probe response after transmitting a probe request, rather than wait for a MinCT mentioned earlier. For instance, iPhone 4, in outdoor environment, waits for 1ms for PRRP while searching each kept SSID. By rendering GTR to use a specific SSID, it can benefit in terms of duty cycling and thus energy cost. In the previous example, GAP wakes up for the first few ms and sleeps the most of 240ms.

D. Implementation of GTR

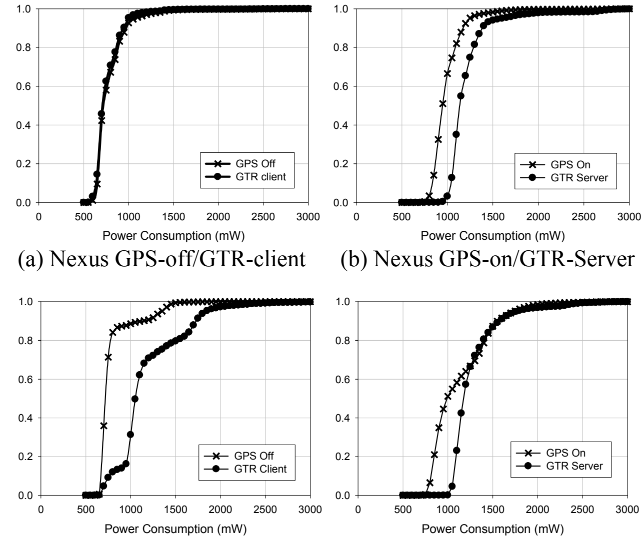
Android 4.1.1 and kernel 3.0.31 [27, 28] are adopted for the implementation of GTR. On receipt of PRRP in this programming environment, the function ieee802_11_parse_elems() is called to retrieve AP information embedded in the *information element parameters*. In practice, most of modifications have been done in file ieee802_11_common.c/h, which defines the processing after receiving a



(a) Nexus (150s) (b) Nexus (10s)

(c) S4 (150s) (d) S4 (10s)

Fig. 5. Comparison of energy consumption (In Nexus, average power is 1253, 761 and 988mW for GTR server, client and GPS-on client, respectively. In S4, it is 1270, 770 and 950mW.)



(a) Nexus GPS-off/GTR-client (b) Nexus GPS-on/GTR-Server

(c) S4 GPS-off/GTR-client (d) S4 GPS-on/GTR-server

Fig. 6. Comparison of energy consumption in cumulative distributed function

PRRP. Note that this function and files are a part of *wpa_supplicant layer* that implements the 802.11 standards [25]. We omit the details in this paper for brevity.

GPS module in a smartphone tries to pinpoint its position only when an application demands it. Android OS contains the android.location package (*Location Manager*) which provides API to determine the current location through GPS or *Network Location Provider* using cell tower and Wifi signals. The core function in this package is requestLocationUpdates(), where three parameters are (i) the type of location provider to use, (ii) the minimum time interval (minTime) between notifications, and (iii) the minimum change in distance between notifications.

With respect to the first parameter, Android provides mock locations for developers in order to test location-aware applications. When mocLocationProvider is enabled in the Location class, a developer-defined location data can be injected as GPS position data instead of GPS coordinates from GPS hardware [29]. Note that the second parameter, minTime, can control the switch on/off of the GPS component. It can potentially rests GPS module for the specified duration to conserve power. For example, when minTime is set as 35 seconds, GPS module goes to sleep state temporarily for the next 35 seconds.

IV. Performance Evaluation of GTR

This section evaluates the energy performance of GTR in comparison to conventional client with GPS on. In fact, this conventional client uses A-GPS (combination of GPS and help from cellular network) mentioned in Section II. Note that a direct comparison with GPS (with no help from cellular network) would look GTR more advantageous. We compare energy performance of GTR server, GTR client and GPS-on client. It is expected that GTR server consumes more power than GPS-on client but the combination of GTR server and GTR client consumes less than two GPS-on clients. Since a single GTR server can serve more than one GTR clients, it becomes more advantageous when there are more clients.

To choose experimental platform for evaluation, the following conditions should be considered: a smartphone provides a detachable battery for easy power measurement; a smartphone supports tethering functionality for the implementation of virtual Wifi. We used Samsung Galaxy Nexus and Galaxy S4. Monsoon Power Monitor [30], an external, real-time power-monitoring device is used to measure the power consumption. Air cap [31] and Wireshark [32] are utilized to monitor and analyze packets transmission.

Fig. 5 shows the power consumption of GTR server, GTR client, and GPS-on client on Galaxy Nexus and Galaxy S4 during the 150 and 10-second interval. While the power is measured at a very fine grain (every 200us), they are averaged at a larger scale to show the overall trend. The former (Fig. 5(a) & (c)) shows the data every 250ms and the latter (Fig. 5(b) & (d)) shows the data every 16ms, each of which is the average of 1250 data and 80 data, respectively. It is clear, in Fig. 5(a) and (b), that GTR server consumes 21% more power than GPS-on but the combination of GTR server and client would be comparable to two GPS-on clients for Galaxy Nexus. Note that peaks are observed at every 100ms in GTR server, which is beacon messages as a mobile AP. We don't apply duty cycling in this experiment for convenience.

Galaxy S4 GTR server also consumes approximately 21% more power than GPS-on as shown in Fig. 5(c) and (d). Even

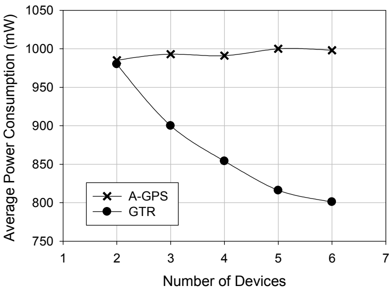


Fig. 7. Average energy consumption with varying numbers of devices (Note that for GTR, six devices refers to one GTR server and five GTR clients)

though Galaxy Nexus uses a dedicated GPS chipset (SiRFstarIV™ GSD4t) and Galaxy S4 uses an integrated chipset from Broadcom (BCM 4752), the trend is somewhat the same regardless of the components used in those devices.

Fig. 6 shows the cumulative density function (CDF) of the power consumption during the 200-second duration measured every 200us. Fig. 6(a) compares GTR Client and GPS Off client on a Galaxy Nexus device. The only additional energy consumption comes from the connection to and communication with the GTR server, which is shown to be negligible. Note that this is not the case in Galaxy S4 as shown in Fig. 6(c). Again, we suspect it is due to the choice of chipset for GPS and Wifi, as mentioned above. It is interesting to observe knee points for in Fig. 6(c), which is not shown in Fig. 6(a). It is considered that GTR Client and GPS Off client spend about 34% and 17% of the time for extra activities (including Wifi access), respectively. On the other hand, it is clear that GTR server spends more percentage of the operation time in higher energy state as shown in Fig. 6(b) and (d) due to the tethering function.

However, with the increasing number of GTR Clients, the additional energy consumption of the GTR Server is paid off. Fig. 7 compares the average energy consumption when the number of devices increase in case of Galaxy Nexus. When the number of devices is six, the energy consumption is about 19% less than six GPS On devices.

V. Conclusions and Future Work

For outdoor localization, GPS localization is accurate but costly in terms of energy. With the increasing popularity of LBS applications on mobile devices, particularly those with sporadic single location fix, it is imperative to employ an effective and efficient localization mechanism. Most research of energy efficient outdoor localization focused on RSS-based and cheap sensor based solutions. Assisted GPS (A-GPS) focuses on first-time-to-fix (FTTF) and leaves energy cost as a secondary issue. This paper presents the design, implementation, and evaluation of a new energy efficient outdoor localization for smartphones, called GPS Tether (GTR), which restrains the activation of GPS module and instead obtains location fix from neighbors. In the future, the authors will refine the STD tuning at the server and the client side, which is critical in boosting energy performance of GTR. Duty cycle control constitutes an important future work.

References

[1] A. Malm. "Mobile Location-based Services." Berg Insight. 2013
[2] K. Zickuhr, Location-Based Services, Pew Research Internet Project, September 2013.
[3] http://www.pcmag.com/article2/0,2817,2401797,00.asp
[4] Marketsandmarkets. "Location Based Services (LBS) Market - Worldwide Forecasts and Analysis". 2014
[5] I. Constandache, S. Gaonkar, M. Sayler, R.R. Choudhury and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in IEEE INFOCOM, 2009, pp. 2716-2720.
[6] L. Zhang, B. Tiwnan, Z. Qian, Z. Wang, R.P. Dick, Z.M Mao, and L Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in IEEE/ACM/IFIP CODES+ISSS, ACM, 2010, pp. 105-114.
[7] J. Liu, B. Priyantha, T. Hart, H.S. Ramos, A.F Loureiro and Q.Wang, "Energy efficient GPS sensing with cloud offloading," in ACM SenSys, 2012, pp. 85-98.
[8] N. Vallina-Rodriguez, J. Crowcroft, A. Finamore, Y. Gruneberge, and K. Papagiannaki, "When assistance becomes dependence: characterizing the costs and inefficiencies of A-GPS," ACM SIGMOBILE Mobile Computing and Communications Review, 2013, 17(4): 3-14.
[9] A. Pathak, Y.C. Hu, and M. Zhang, "Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof," in EuroSys '12.
[10] "TetherGPS," Google Play, 2013
[17] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Warland, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in IEEE INFOCOM, 2012.
[12] http://www.navipedia.net/index.php/Main_Page
[13] Z. Zhuang, K.H Kim, and J.P. Singh, "Improving energy efficiency of location sensing on smartphones," in MobiSys. ACM, 2010, pp. 315-330.
[14] T. Farrell, R. Cheng, and K. Rothermel, "Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances," in IDEAS, 2007, pp. 129-140
[15] S. Gaonkar, J. Li, R.R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: sharing and querying content through mobile phones and social participation," in ACM MobiSys, 2008.
[16] D.H. Kim, Y. Kim, D. Estrin, and M.B srivastava, "Sensloc: sensing everyday places and paths using less energy," in ACM SenSys, 2010, pp. 43-56.
[17] X. Zhu, Q. Li, and G. Chen, "APT: Accurate outdoor pedestrian tracking with smartphones," in IEEE INFOCOM, 2013, pp. 2508-2516.
[18] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao, "WheelLoc: Enabling continuous location service on mobile phone for outdoor scenarios," in IEEE INFOCOM, 2013.
[19] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in ACM MobiSys, 2010, pp. 285-298.
[20] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in ACM MobiSys, 2010.
[21] R. Wagner, V. Lee, and R. Phillips, "Gps tracking system and method employing public portal publishing location data," U.S. Patent Application 13/281,181[P]. 2011-10-25.
[22] H.S. Ramos, T. Zhang, J. Liu, N.B. Priyantha, and A. Kansal, "LEAP: a low energy assisted GPS for trajectory-based services," in ACM Ubiquitous, 2011, pp. 335-344.
[23] P. Misra, W. Hu, Y. Jin, J. Liu, N. Wirstrom, and T. Voight, "SparseGPS: energy efficient GPS acquisition via sparse approximation," in ACM SenSys, 2013, pp. 29.
[24] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S. Park, and W. Kim, "Scanless Fast Handoff Technique Based on Global Path Cache for WLANs," Journal of Supercomputing, 66(3), 2013.
[25] IEEE 802.11, Standard, IEEE, 2012.
[26] G. Castignani, A. Arcia, and N. Montavont, "A study of the discovery process in 802.11 networks," in ACM SIGMOBILE Mobile Computing and Communications Review, 2011, 15(1): 25-36.
[27] Android Source, http://source.android.com/source/downloading.html/.
[28] Android Kernel, http://source.android.com/source/building-kernels.html/.
[29] Android Mock Application http://developer.android.com/reference/ android/test/ mock/MockApplication.html/.
[30] Monsoon Solutions Inc. http://www.msoon.com/LabEquipment/Power Monitor/.
[31] AirPcap. http://www.airpcap.nl/.
[32] Wireshark. http://www.wireshark.org/.