A MS-Assisted GPS System For-Low Power Location-Based Mobile Applications Technical Report

Intern: Heitor Ramos <heitor.ramos@live.com>
Menthor: Jie Liu <jie.liu@microsoft.com>
Collaborator: Bodhi Priyantha <bodhip@microsoft.com>

August 20, 2010

1 Introduction

Location based applications have gained much attention in recent years mainly with the advent of mobile devices such as smart-phones, PC-tablets, and netbooks. These devices facilitate the position data collection and allows some applications to make use of this data. For instance, mobile search is taking advantage of the local information to improve the user experience.

Among the most common location sensors, GPS is the most accurate providing very small location error and allowing, for instance, very high accurate applications such as turn-by-turn navigation. Therefore, the sensors commonly used to collect location data are very energy hungry and tend to deplete the device's battery very quickly. Although GPS is the most accurate current location sensor, it is also the most energy hungry, thus, nowadays the use of GPS sensor is very restricted in mobile application scenarios.

Bearing in mind this constrained scenario, this work aim at studying the GPS theory and proposing some alternatives of using the GPS sensor in low energy consumption mode. The idea behind this proposal is to use a very low duty-cycle GPS, when it is possible, while the application requirements are met.

This work is organized as follows: Section 1.1 deeply discusses the GPS theory necessary to understand the rest of the text; Section 2 discusses the application scenarios that the proposed techniques are affordable and discusses some aspects of the GPS energy consumption; Section 3 presents some related work and highlights the main differences of our approach; Section 4 details our proposal and shows the evaluation results, while Section 5 concludes and provides some additional comments.

1.1 GPS basics

The Global Positioning System (GPS) is a satellite based radio navigation service created by the U.S Department of Defense (DoD) and offer the U.S military accurate estimates of position, velocity and time. Even though there are many comprehensive books on GPS subject such as Misra & Enge (2006), Borre et al. (2006), van Diggelen (2009) and Kaplan & Hegarty (2005), the rest of this section sums up the basic GPS concepts spread on these books, which are necessary to the comprehension of the rest of this work.

The GPS system was developed to provide precise estimates, roughly, with position error of $10\,\text{m}$, velocity error of $0.1\,\text{m/s}$, and time error of $100\,\text{ns}$, all in terms of root-mean-square (Misra & Enge 2006). The system was planned to be available to an unlimited number of users and to provide two types of service: (i) Standard Positioning Service (SPS), and (ii) Precise Positioning Service (PPS). The former is offered to peaceful civil use and the latter to DoD-authorized users, protected by cryptography.

The GPS system consists of three basic parts: space, control and user segments. The space segment is composed of a constellation of 24 satellites distributed in 6 orbital plans, being 4 satellite in each plane, in medium Earth orbit. Currently, there are 29 operational satellites to assure uninterrupted service. The constellation was planned in a way that all users with a clear view of sky have a minimum of four satellites in view, but is more likely that a user would see six to eight satellites (Misra & Enge 2006). The control segment is maintained and developed by the U.S. Air Force that monitors satellite orbits, maintains satellite healthy and GPS time, predicts satellite orbit and clock parameters, and updates satellite navigation messages. The user segment is composed of hundred of thousands of U.S. and allied military users of the PPS service and tens of millions of civil, commercial and scientific users of SPS service.

Each satellite broadcasts an unique signal in direction of the Earth surface and the GPS receiver estimates its position based on the time of arrival (TOA) of satellite signals. Each satellite transmits continuously using two radio frequencies in the L-band, namely Link 1 ($L1$) and Link 2 ($L2$), being $f_{L1} = 1575.42$ MHz and $f_{L2} = 1227.60$ MHz. Two signals are transmitted on $L1$, one for civil and other for DoD-authorized users, while the $L2$ channel is used only by DoD-authorized users. The signal transmitted on $L2$ band is encrypted and is not considered in the rest of this work.

1.1.1 GPS signal

The GPS signal consists of three components: (i) carrier, (ii) ranging code, and (iii) navigational data. Basically, each satellite broadcasts the carrier frequency $L1$ modulated with a unique pseudo-random (PRN) code, namely C/A code. It's a CDMA code used to identify the satellites on the receiver side, and helps on mitigate the undesirable effects of multi-path propagation and interfering signals perceived on GPS receiver antenna. It's modulated with a binary-coded message consisting of the satellite health status, ephemeris (satellite position and velocity information), clock bias parameter, and almanac (reduced-precision ephemeris data).

The C/A code is a sequence of 1023 bits, called chips. The duration of C/A chip is $1\,\mu$s, thus, the duration of each C/A code is 1 ms and the chipping rate is 1.023 MHz. The navigational data is transmitted in packets of 1500 bits sent at 50 bits/sec, with a bit duration of 20 ms. It takes 30 sec for one entire packet to be received.

The C/A signal at satellite can be modeled as

$$s_{L1}^{(k)}(t) = \sqrt{2P_{tmt}}x^{(k)}(t)D^{(k)}(t)\cos(2\pi f_{L1}t + \theta_{tmt}),$$

where P_{tmt} is the transmitted signal power for signal carrying on $L1$, $x^{(k)}$ is the C/A-code sequence of satellite k, $D^{(k)}$ is the navigational data of satellite k, f_{L1} is the carrier frequency of $L1$ band, and θ_{tmt} is the phase off-set. The C/A signal on $L1$ perceived on the receiver can be modeled as

$$r_{L1}^{(k)}(t) = \sqrt{2P_{rcv}}x^{(k)}(t-\tau^{(k)})D^{(k)}(t-\tau^{(k)})\cos(2\pi(f_{L1}+f_D)t + \theta_{rcv}) + n^{(k)}(t),$$

where $P_{tmt} \gg P_{rcv}$, the subscripts tmt and rcv stand for 'transmitted' and 'received', respectively; $\tau^{(k)}$ is the signal propagation time for the satellite k, f_D is the frequency shift deviation by Doppler effect, and $n^{(k)}(t)$ is the noise. Figure 1 illustrates the structure of the GPS signal, modulated by binary phase-shift key technique (BPSK).

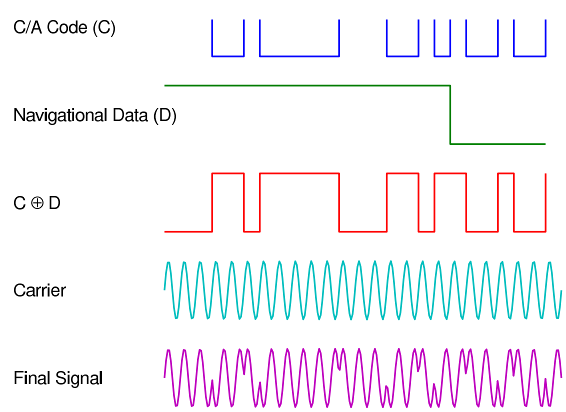

Figure 1: The structure of GPS signal. Source: Borre et al. (2006)

1.1.2 GPS receiver

A GPS receiver estimates its range to each visible satellite to solve its position equations. The range estimated by the receiver is known as pseudo-range due to the uncertainty of those measures during this process. Although the receiver's position can be calculated using three range measurements from known points, in case of GPS, four know points (satellite positions) will be necessary in order to accomplish the common clock bias range equivalent calculation. This follows from the fact that the GPS receiver is not tightly time synchronized with the satellites, so, all measures will be biased by the same value. Thus, the pseudo-range equations can be written as following:

$$\rho^{(k)} = ||X_s^{(k)}(t - \tau^{(k)}) - X|| + b + \epsilon^{(k)}, \tag{1}$$

where $\rho^{(k)}$ is the pseudo-range from the receiver to satellite k, $X_s^{(k)}(t - \tau^{(k)})$ is the position of satellite k at the time of transmission, $\tau^{(k)}$ is the propagation delay, X is the receiver position, b is the common clock bias range equivalent, and $\epsilon^{(k)}$ represents the unknown error in measurement. Thus, we need at least four linearly independent equations to solve for the four unknown variables, three for position and one for the common clock bias. When more than four satellites can be visible, we have an overdetermined system of equations, generally solved by least square method applied to minimize the difference between the computed pseudo-range and its measured value, this difference is called the pseudo-range residual.

Before calculate its own position, the receiver needs to conditioning the received signal in order to make it more manageable. Actually, the signal reaches the receiver with a very low power after travel about $20,000\,$km , about $-10^{-14}\,$W/m^2 ($-128\,$dbm) and it's very sensible to many natural and man-made radio-frequency interference. Those undesirable interferences should be removed in the receiver's frontend. Additionally, the receiver carrier frequency must be down converted from about $1.6\,$GHz to something more manageable, by a factor between 100 to 1000. After conditioned, the signal must be amplified to ignite the A/D (analog-to-digital) convert. After A/D conversion, the receiver performs at least three tasks: (i) acquisition, (ii) tracking, (iii) position calculation.

The main purpose of the acquisition phase is to find visible satellites and rough values of the carrier frequency and code phase. Basically during the acquisition phase the receiver perceive the signal's frequency different from its nominal value due the Doppler effect. Typically the frequency vary about $10\,$kHz from its nominal value. As reported in Borre et al. (2006), a regular receiver typically performs the search sweeping bins of $500\,$Hz. To find out the code phase, the receiver needs to search a space of $1023\,$chips for every frequency bin. Thus, a search space of $1023(2\frac{10{,}000}{500} + 1) = 41,943$ combinations is swept. For each bin of this search space the receiver correlates the local generated copy of satellite's C/A code with the observed signal.

Using the results of the acquisition phase, the receiver is able to reproduce a local copy of the signal transmitted by the satellites that will be used to track the signal. During the tracking step, the code phase and frequency estimates are refined and the receiver keep track of these measures during the time. Basically the tracking is comprised by two parts: (i) code tracking, generally implemented as a delay lock loop (DLL) where local copies are correlated with the observed signal, and (ii) carrier frequency/phase tracking. The receiver keeps tracking the signal continuously to follow the changes, and whenever the receiver loses a satellite, another acquisition is necessary.

During the signal tracking, if enough number of satellites can be found and the navigational data can be demodulated, the receiver has the necessary information to estimate its pseudo-ranges, to compute the satellite's orbits and its own position. Figure 2 shows a schematic view of the aforementioned modules of the GPS receiver.

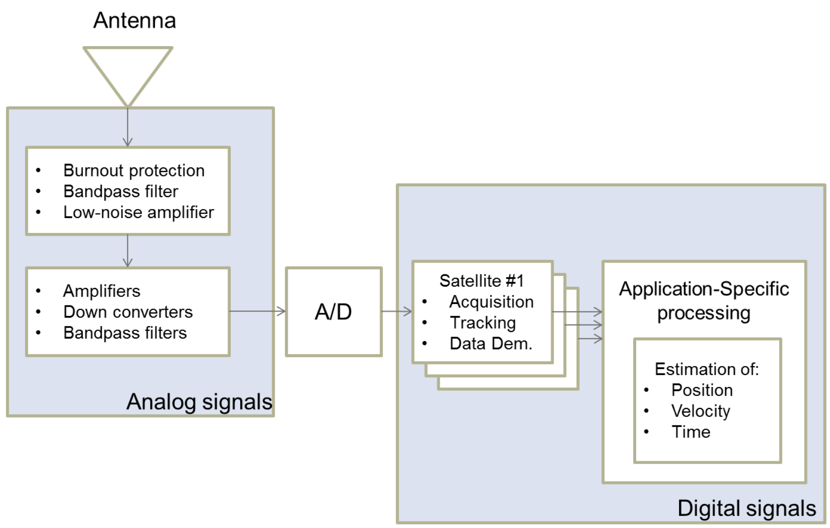

Figure 2: A schematic view of a GPS receiver: analog and digital signal processing

One important aspect of the GPS system is the time-of-first-fix (TTFF), i.e., the time passed until the GPS receiver unit can compute its position with a reasonable precision. Basically standards standalone GPS receivers can calculate its first position by a two different ways:

Cold start: the receiver don't use any a priori information and needs to perform all steps of the acquisition, tracking and navigation data decoding phases before compute the first position.

Warm start: the receiver uses a valid already stored almanac data and the receiver's last position in order to narrow the search space of the acquisition phase.

Some authors such as van Diggelen (2009) also describe the hot start. Basically it occurs when an user turns off the receiver for a short amount of time and the receiver have not only the almanac data but also an already decoded ephemeris stored in its memory. Additionally, the receiver's clock is suppose to be tight synchronized to satellite's clock and the code delay might be known. Typically, GPS receivers take about 1 min, 30 s, and 6 s, for cold, warm, and hot start, respectively (van Diggelen 2009).

1.1.3 Navigation equations

In order to estimate the pseudo-ranges by using the Eq. (1), the receiver needs a time reference to know in what time some portion of the signal was transmitted by the satellites. To do so, the time-of-week included in the field (HOW) is decoded from the navigation data. The satellite time-of-week represents the number of seconds passed since the last GPS week rollover (every Saturday midnight). Figure 3 shows the GPS navigational packet structure and its five sub-frames. Basically the navigation data is 1500 bits long, composed of five 300 bits long sub-frames. It means that each sub-frame takes 6 s to be transmitted. The first field (TLM) includes the packet preamble and some parity information, the second field (HOW) includes more parity information and the time-of-week (TOW), while the payload includes satellite's information and the navigational data itself.

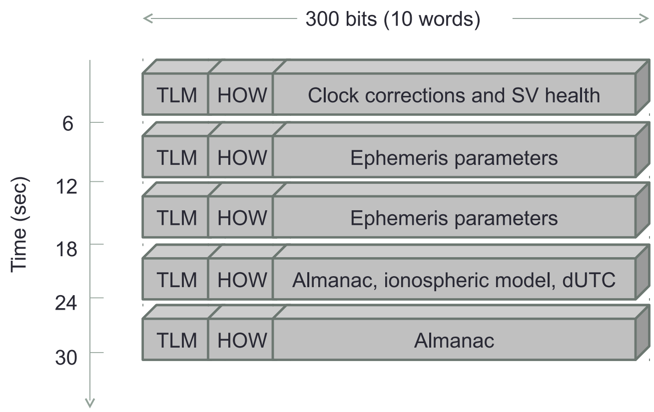

Figure 3: Navigational Data: one frame composed by six sub-frames. Source: Borre et al. (2006)

In other words, the receiver needs to decode the packet not only to get the navigational information but also to identify the packet borders that represent the time reference for the pseudo-range calculation. Let's considered that the binary data collected by the receiver is composed by chunks of 1 ms (one C/A code length), thus, when the receiver detects a sub-frame border, it realizes the millisecond component of the transmission delay, while the sub-millisecond is determined by the code range alignment during the tracking phase. Thus, the transmission delay can be seen as (Sirola 2001):

$$\tau^{(k)} = \frac{1}{1000}(N^{(k)} + \varphi^{(k)}), \tag{2}$$

where $N^{(k)}$ is the millisecond component estimated by the receiver as the time of the detection of the sub-frame sent by the satellite k, and $\varphi^{(k)}$ is the sub-millisecond component that is estimated by the observed code delay into the 1 ms chunk. Thus, applying Eq. (2) in Eq. (1) we get:

$$\frac{c}{1000}(N^{(k)} + \varphi^{(k)}) = ||X_s^{(k)}(t - \tau^{(k)}) - X|| + b + \epsilon^{(k)}, \tag{3}$$

where c is the speed of the light. The left side of the Eq. (3) represents the observed pseudo-range while the right side represents the calculated one.

As the Eq.(3) presents four unknowns (three for the receiver's position and one for the common bias), we need at least 4 satellites to have a well conditioned system. More often than not, receivers can see more than 4 satellites and we have an over-determined system of non-linear equations. The most usual way to solve this problem is by linearizing the equations and using the least-square method. To do so, we need to solve

$$\delta z = H\delta x + \epsilon, \tag{4}$$

where δx is the vector of updates to the a priori state (x, y, z, and b), $\delta z = z - \hat{z}$ is the vector of a priori measurements residual, being z the vector of measured pseudo-ranges and $\hat{z}$ the vector of predicted pseudo-ranges, ϵ is the unknown error (tropospheric model uncertain, and other stochastic errors), and

$$H_{k\times 4} = \begin{bmatrix} -e^1 & 1 \\ \vdots & \vdots \\ -e^k & 1 \end{bmatrix},$$

being e^k the three-dimensional unit vector that points from the predicted receiver's position to the satellite k. The matrix H is obtained after the linearization process of the Eq. (1). This process is out of the scope of this document and further discussion can be found in van Diggelen (2009). The initial position is usually assumed to be the center of the earth, i.e., $X = 0, Y = 0$, and $Z = 0$, when a better approximation is missing.

Some authors such as Syrjarinne (2000), Sirola (2001), Akopian & Syrjarinne (2009) and van Diggelen (2009) describe different methods to estimate the millisecond part of the Eq. (3) without decoding the data packet. Adopting those solutions, GPS devices are able to reduce its TTFF because the data sub-frame takes about $6\,\mathrm{s}$ to be transmitted. Here we will describe the last solution that we adopt in the rest of this work.

If the GPS receiver has a coarse time synchronization, an already decoded ephemeris/almanac, and, a good initial position guess, it can estimate the initial pseudo-ranges by using the initial position and the ephemeris/almanac to calculate the satellite position. This process adds another source of error, namely the coarse time error, because the receiver does not have a precise time reference. An error on the receiver's time reference will lead to wrong satellite position estimation. This error, differently as the common bias effect, leads to a different amount of errors on the satellites' position. The coarse time error is described in van Diggelen (2009) and can be model as:

$$\hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)}(t_{tx}) = \hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)} - \hat{z}^{(k)}(\hat{t}_{tx} + \delta_{tc}) \tag{5}$$

$$= -\nu^{(k)}.\delta_{tc}, \tag{6}$$

where δ_{tc} is the update to the a priori coarse-time state, t_{tx} is the actual time of transmission, $\hat{t}_{tx}$ is the coarse-time estimate of t_{tx}, $\nu^{(k)} = (e^k.v^{(k)} - \dot{\delta}_t^{(k)})$ is the pseudo-range rate, $e^{(k)}$ is the unit vector as described before, $v^{(k)}$ is satellite velocity vector, and $\dot{\delta}_t^{(k)}$ is the satellite-clock error rate. The two last can be obtained from the ephemeris. The Eq. (6) can be included in the navigation equations and the lest-square equations aforementioned will present one more state, the coarse-time error. Thus, the state update vector is $\delta x = \begin{bmatrix}\delta_x & \delta_y & \delta_z & \delta_b & \delta_{tc}\end{bmatrix}'$, and the matrix H is

$$H_{k\times 5} = \begin{bmatrix} -e^1 & 1 & \nu^{(1)} \\ \vdots & \vdots & \vdots \\ -e^k & 1 & \nu^{(k)} \end{bmatrix},$$

all terms were described before.

Even though the millisecond part of the pseudo-range can be estimated by the initial position and the coarse-time, any small error on this estimate can lead to a very inaccurate position calculation. For instance, an error of 1 ms produces results 300 km away from the actual location. Thus, to avoid problems with integer rollover, that typically incurs this amount of error, van Diggelen (2009) propose the following method to reconstruct the full pseudo-ranges. Let the transmission delay be modeled by:

$$N^{(k)} + \varphi^{(k)} = r^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)} \qquad (7)$$

$$= \hat{r}^{(k)} - d^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)} \qquad (8)$$

where $r^{(k)}$ is the actual geometric range from the satellite k, $\delta_t^{(k)}$ is the satellite clock errors obtained from the ephemeris at the a priori coarse time for the satellite k. As $r^{(k)}$ is unknown, we can substitute $r^{(k)} = \hat{r}^{(k)} - d^{(k)}$ where $\hat{r}^{(k)}$ is the estimated pseudo-range from the a priori position at the coarse time of transmission and $d^{(k)}$ is the error in $\hat{r}^{(k)}$. The method involves the choosing of a reference satellite, $k = 0$, where $N^{(0)} = round(\hat{r}^{(0)} - \varphi^{(0)})$ is the millisecond part of the pseudo-range of the reference satellite [1]. This value is used in order to reconstruct the millisecond pseudo-ranges for all other satellites relatively to the reference satellite. Thus, if we subtract the Eq. (8) from the reference satellite full pseudo-range we get:

$$N^{(k)} = N^{(0)} + \varphi^{(0)} - \varphi^{(k)} + \left(\hat{r}^{(k)} - d^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)}\right) - \left(\hat{r}^{(0)} - d^{(0)} - \delta_t^{(0)} + b + \epsilon^{(0)}\right).$$

We still don't know the values of $d^{(0)}$ and $d^{(k)}$, but considering that we have a initial position and coarse time close to the correct values (100 km and 1 min), the order of magnitude of $(-d^{(k)} + \epsilon^{(k)} + d^{(0)} - \epsilon^{(0)})$ is less than about 150 km, van Diggelen (2009) shows that we can correctly estimate $N^{(k)}$ by:

$$N^{(k)} = round\left(N^{(0)} + \varphi^{(0)} - \varphi^{(k)} + (\hat{r}^{(k)} - \delta_t^{(k)}) - (\hat{r}^{(0)} - \delta_t^{(0)})\right). \qquad (9)$$

This method avoids the undesirable integer roll over problem that leads up 300 km of error for each wrong estimate.

[1] van Diggelen (2009) recommends the use of the highest satellite in view as reference, and provides a good reason for that

1.2 AGPS

Assisted GPS (AGPS) is an augmentation technique used to improve some functionalities of the GPS system (Sirola 2001). It relies on further existing network infrastructure to provide additional information to a GPS receiver. For instance, mobile phones can use data communication such as 3G, EDGE, WIFI or Bluetooth in order to get all necessary extra information. Notice that the AGPS receiver makes measurements of the GPS satellites while grabs additional data from a network, thus, an AGPS receiver still needs to processes the satellite signals.

Basically, the AGPS system provides the information that allows the AGPS receiver to be aware of the set of expected visible satellite, the expected frequencies that should be swept, additional ephemeries and/or almanac data used to calculate the satellite's orbit, time synchronization, and initial position estimate. Provided by this assistance, AGPS receivers can be able to calculate its TTFF quickly, typically in the order of some few seconds in contrast of 1 min from the standalone GPS receiver.

An AGPS system can assist the receiver device by many different ways (van Diggelen 2009). For a cold start, it can provide all possible frequencies and code delays to narrow the acquisition search. In this case, it's still necessary to decode the TOW and ephemeris from the navigational data. For a warm start, the already stored almanac can be used in conjunction to a priori position and time in order to decrease the search space. It's also necessary to decode the TOW and ephemeris from the navigational data. For a hot start, a priori position and time can be used in the same way as in warm start. In this case, if a precise time is known, it's not necessary to decode the TOW and ephemeris.

AGPS systems can be classified in two general approaches: (i) MS-Assisted, where the position is calculated at a server, and (ii) MS-based, where the position is calculated at the receiver, helped by network assistance.

In MS-Assisted approach, as the position is calculated at a server, the GPS receiver's job is simplified. It needs to acquire satellites' signal and send the measurements to the server. Hence, the receiver doesn't need any satellite ephemeris and/or almanac. The receiver can send its pseudo-ranges or raw data. If the receiver sends the pseudo-ranges, it needs to execute the acquisition algorithms that can also be assisted by network data in order to narrow the search space. If fine-time assistance is available, the server can directly compute the expected code delays and send to receiver.

The MS-based approach is mostly used by hardware manufacture and the position is calculated at the receiver itself. In this approach, the receiver needs to have all the functions of a standalone GPS device and the assistance is performed to provide the expected Doppler frequencies and code delay, navigational data (almanac and/or ephemeris), coarse or fine time (for code delay assistance, fine-time is necessary), and an initial estimate of the receiver's position. As fine-time assistance, it's considered an accuracy better than 1 ms, otherwise, it's called coarse-time assistance. Table 1 is adapted from van Diggelen (2009) and shows the benefits of assistance by means of TTFF.

Table 1: Summary of AGPS assistance for different type of starts. Source: van Diggelen (2009)

Type of Start	Acquisition	Ephemeris	Time of Week	TTFF
Standalone Cold	≈ 30 s	From satellite data ≈ 30 s	From ephemeris	≈ 1 min
Standalone Warm	≈ 1 s	From satellite data ≈ 30 s	From ephemeris	≈ 30 s
Standalone Hot	≈ 1 s	In memory	Accurate real-time clock, or decoded from HOW ≈ 6 s	seconds
Assisted Cold, Coarse-Time	≈ 1 s	From server	decoded from HOW ≈ 6 s	seconds
Assisted Cold, Fine-Time	≈ 1 s	From server	Assistance time	≈ 1 s

2 Problem description

GPS devices are energy hungry and, when used in battery-enable devices, usually drains the battery energy quickly. For instance, Lin et al. (2010) present a relative energy usage of some sensors used to provide location. Observe in Figure 4 that GPS sensor is, together with Bluetooth, one of the most energy hungry sensor.

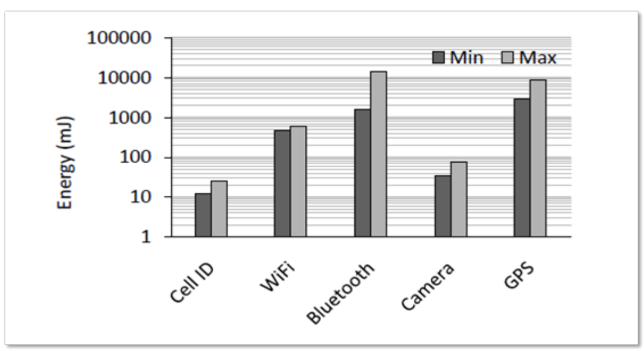

Figure 4: Relative energy usage of some common location sensors. Source: Lin et al. (2010)

With the advent of mobile systems, different location-based applications with different requirements in terms of location accuracy and positioning rate have emerged. Figure 5 illustrate some application scenarios and their different requirements in terms of accuracy and location query rate. In one extreme we have the turn-by-turn navigation that requires a strong accuracy and very high position query rate. For this application the accuracy must suffice to identify on which lane a driver is conducing his vehicle. Besides, turn-by-turn navigation requires a very high frequency position updates to keep tracking a moving vehicle precisely. On the other hand, some applications such jogging, hiking and weather report, require just a slight accuracy, sometimes just to identify in which block or neighborhood the user is locate and don't need to get location samples very often due the low speed that a person moves.

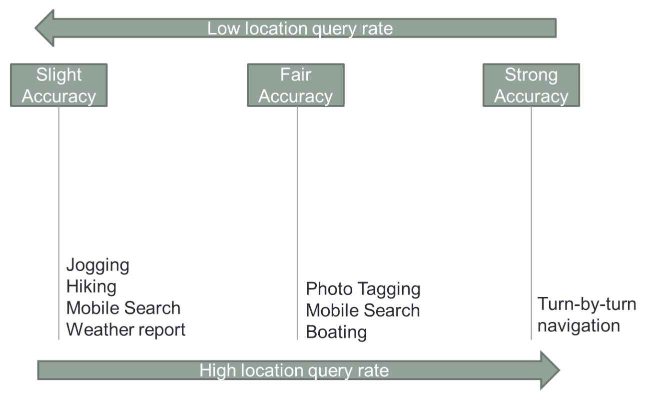

Figure 5: Different application scenarios

Thus, the problem that we are addressing in this work is how to devise a GPS-based system considering the mobile phone application scenario, in a such way that the energy-consumption of the location task is tight related to the application profile. We are thus interested in not only decrease the TTFF of the GPS receivers but mainly to save energy.

2.1 GPS energy

In this Section, we describe how GPS receivers drains the battery when they are active. We use a HTC desire smart-phone running Android 2.1 to trace a power/energy profile.

2.1.1 Energy model

Observing the receiver's building blocks shown in Figure 2, we can empirically derive the following energy model:

$$E_{GPS}(t) = \varepsilon_{on} + \varepsilon_{analog}(t) + \varepsilon_{digital}(t) + \varepsilon_{off},$$

where ε_{on} and ε_{off} represent the energy required to turn the GPS circuitry on and off, respectively; ε_{analog} represents the energy required to perform all the analog signal processing on the receiver, such as bandpass filter, amplifiers, frequency down conversion, and A/D conversion; $\varepsilon_{digital}$ represents the energy drained by the digital signal processing such as correlators, filters, demodulators, least-square calculation, i.e., all tasks required to perform the acquisition, tracking, data demodulation and position calculation. Observe that ε_{analog} and $\varepsilon_{digital}$ depend on the time that the system dwells on each phase, for example the acquisition phase can be performed quickly with network assistance, and thus, drain less energy. The terms ε_{on} and ε_{off} depend only on a fixed amount of energy drained during those phases.

The digital signal processing can be detailed as:

$$E_{Digital}(t) = \varepsilon_{acquisition}(t) + \varepsilon_{tracking}(t) + \varepsilon_{position}(t).$$

2.1.2 Power profile

We've performed two experiments in order to trace a GPS power profile[2]. In the first experiment we measure the power consumption when the GPS was trying to acquire satellites in an indoor environment. Thus, the tracking and position calculation were avoided and only the component $\varepsilon_{acquisition}(t)$ of the digital signal processing was measured. Obviously other components such as $\varepsilon_{on}, \varepsilon_{analog}$ and ε_{off} are present but, in this work, we are only interested on the digital side. So, for our solution we are interesting to know the differences on the power consumption when the GPS device performs the acquisition, tracking and position calculation. The second experiment aims at empirically estimating the power consumption when the GPS is used in outdoor environments. In this last case, all phases will be performed.

Figure 6 shows a power profile of the GPS device from the smart-phone HTC desire when trying to get a position in indoor environments. Note that we can identify some peaks when the GPS is trying to acquire satellites. Those peaks represent about 400 mw of power consumption. Figure 7 shows a power profile

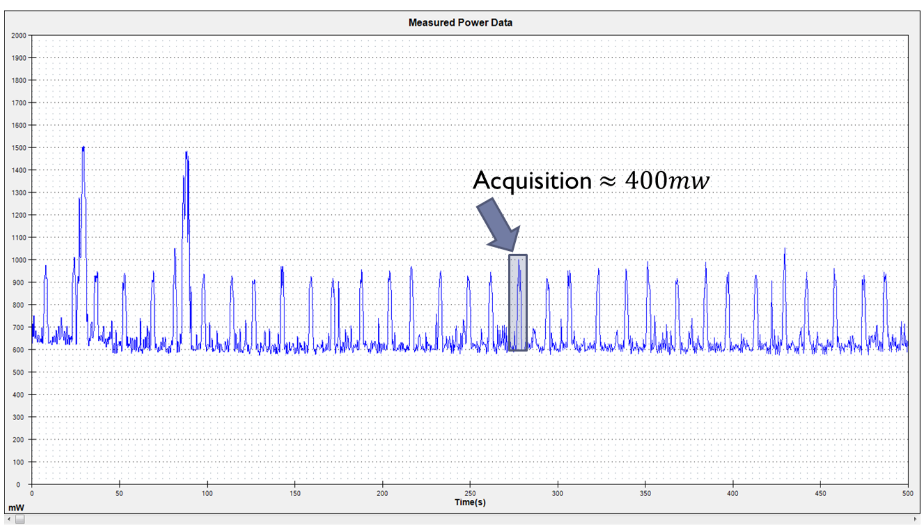

Figure 6: Instantaneous power consumption for acquisition phase

of the GPS device from the same smart-phone when it tries to get a position in outdoor environments. Note that now we are able to identify all the phases, being

[2]Thanks to Miguel Palomera for this data

the acquisition phase according with the first experiment, i.e., about 400 mw, the tracking phase about 700 mw and the position calculation about 1200 mw. For this experiment we query the position many times to force the separation of the phases, thus, we can see many short peaks when the position is being calculated. We assume that those high peaks represent the least-square calculation on the smart-phone's main processor.

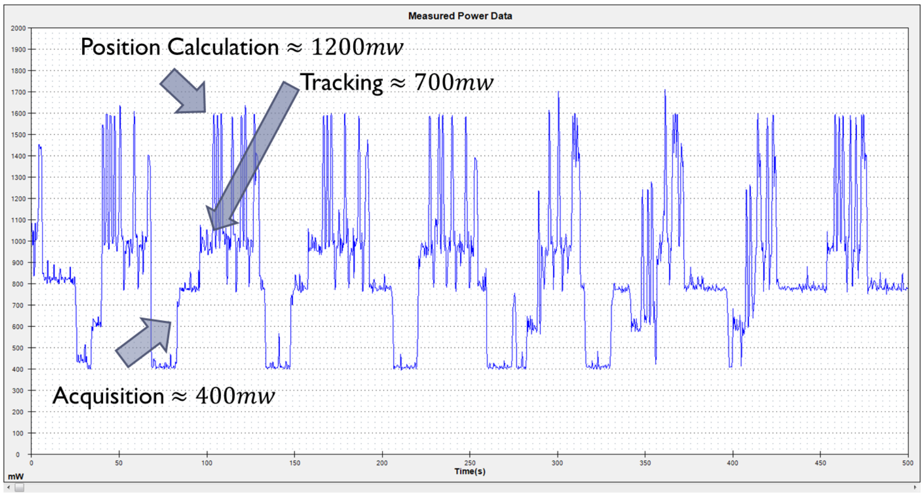

Figure 7: Instantaneous power consumption for acquisition, tracking and position calculation phases

3 Related work

Bearing in mind the problem we are interested in, current solutions presented in the literature tend to explore two different approaches, as following.

Some authors strive to devise novel signal processing and positioning techniques mainly to increase the GPS receiver's accuracy and TTFF. For instance, Sun et al. (2005) present a comprehensive survey on signal processing techniques used in AGPS systems. Akopian & Syrjarinne (2009), Sirola (2001) and van Diggelen (2009) show different approaches on how to model the GPS positioning equations and how to solve them without decode any portion of data when some assistance is provided. Those solutions are useful when the receiver can use assisted data from network, thus, they can decrease the TTTF and allow the use of GPS devices even when data decoding is not possible such as in presence of noise, shadow and incomplete data.

By different approaches, some authors try to establish a smart use of GPS by changing its duty-cycle and using different types of sensors available in current smart-phones. For example, Lin et al. (2010) propose a smart mechanism, namely A-Loc, which automatically determines the dynamic accuracy requirement for mobile search applications. A-Loc chooses the sensor that saves energy while attend the application's accuracy requirements. This approach relies on the idea that location applications do not always need the highest available accuracy. To do so, they use a Bayesian estimation framework to model the location and sensor errors, and thus, choose the most adequate sensor.

Paek et al. (2010) propose a rate adaptive GPS-based positioning for smartphones. They argue that, due the so-called urban canyons problem, even GPS sensor is not able to provide a high accuracy position, thus, the device should be turned off. In those situations, alternatively, they use other location mechanism such as accelerometer, space-time history, cell tower blacklisting and bluetooth to decrease the active time of GPS devices.

Zhuang et al. (2010) present an adaptive location-sensing framework to reduce the use of GPS device in various scenarios. Their framework is based on four design principles: (i) substitution, where alternative location-sensing mechanisms are used instead of GPS device; (ii) supression, where the information provided by other sensor can suppress the use of the GPS device, for instance, if the accelerometer can identify that the user is not moving, the GPS can be turned off; (iii) piggybacking, that synchronizes location request from different applications, and (iv) adaptation that adjust sensing parameters when battery level is low.

Differently from the current proposals as those herein discussed, we are not trying to avoid the use of GPS sensor once it's the more accurate known sensor to provide location service. In this work, we strive to provide a low-power GPS-based location service as described in the next Section.

4 Our proposal

We propose the use of an adaptive GPS device for mobile phone applications, in the sense that the device changes its operation mode in order to meet the application requirements such as those shown in Figure 5.

Thus, differently from the proposals discussed in Section 3, our proposal is focused on GPS-based location only. The basic idea is that the GPS radio should be used for the smallest possible time interval (very small duty-cycle). As we see in Figures 6 and 7, the GPS dwells several seconds performing the acquisition and tracking phases and, thus, drains a large amount of energy. In our proposal, the GPS receiver should perform an assisted acquisition in order to narrow the search space (frequency x code phase) and decrease the time to figure out which satellites are visible as the same way that current AGPS devices are doing. van Diggelen (2009) brings a thorough discussion on acquisition assistance on Chapter 3. Additionally, we use of the technique shown in Section 1.1.3 in order to drive the receiver's radio duty-cycle. Thus, the assistance here aims not only at decreasing the TTFF but mainly at saving energy.

Figure 8 shows the system view and the necessary conditions to this approach. The GPS device receives assistance to fast the acquisition phase and a coarse time. The device should collect the GPS signals and performs the acquisition and tracking phases to estimate the code phase. This acquisition and tracking are performed using only few milliseconds of the signal. Once the acquisition for standalone GPS systems is performed typically using 11 ms of the signal, we used this amount of time as our baseline solution. Thus, the GPS device needs to perform the track phase only in the same amount of the signal and its radio can be turned off as soon as the correspondent amount of data is collected.

After the device estimate the code phase, it sends this sub-millisecond part of the pseudo-ranges to the cloud. The cloud is aware of the cell towers' positions and uses the position of the cell tower that the mobile phone is locked as the initial position for the method described in Section 1.1.3. The cloud also has the ephemeris/almanac of all satellite constellation in order to reconstruct the the user's receiver full pseudo-range by using the Eq. 9. Thus, we avoid lots of digital signal processing and position calculation that are very energy hungry tasks and pay the cost of send a small amount of data to the cloud. In next Section we show the evaluation results of this solution.

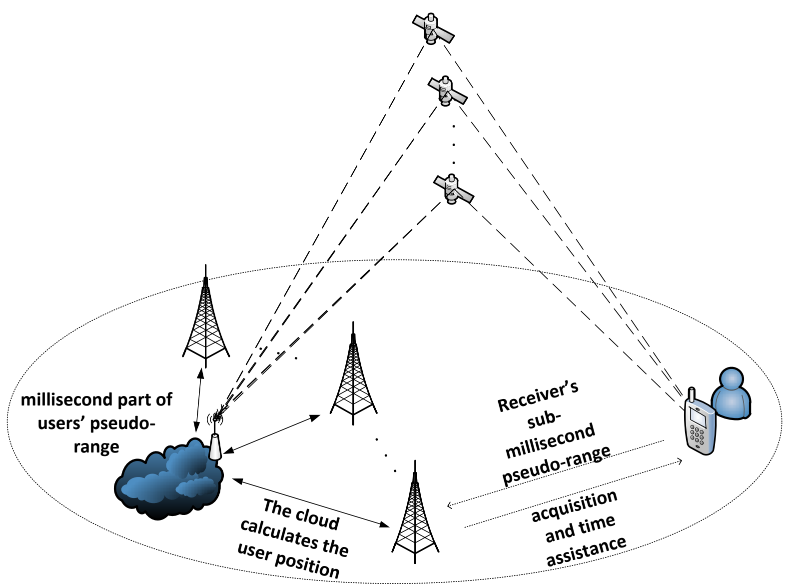

Figure 8: Low-power MS-assisted system overview

4.1 Evaluation

We collect raw GPS data of from 3 different locations by using a SiGe GN3S sampler version 2 dongle from Sparkfun Electronics[3]and we chose one different location to

[3]http://www.sparkfun.com/commerce/product_info.php?products_id=8238

Location	Latitude	Longitude	Precision (m)	# of Satellites
Assumed Cell Tower	47.64168°	−122.14081°	±5.18	9
Shopping Overlake	47.63006°	−122.14433°	±4.57	10
Microsoft Commons	47.64337°	−122.13928°	±4.27	10
Grass Lawn Park	47.66941°	−122.14373°	±4.88	8

Table 2: Ground truth positions

simulate our cell tower position. We use a Garmin GPS Map 60CS[4] to collect the position of those locations and use them as ground truth. Table 2 shows all the positions (latitude and longitude), the precision and the number of satellite in view at the moment we collected the positions.

Figure 9 shows how the locations are geographically distributed and the distances to the assumed cell tower. Observe that all those distances are within the maximum allowed by the technique herein used (see Section 1.1.3) and are plausible for current mobile networks cell size. Next sections show the results of this approach.

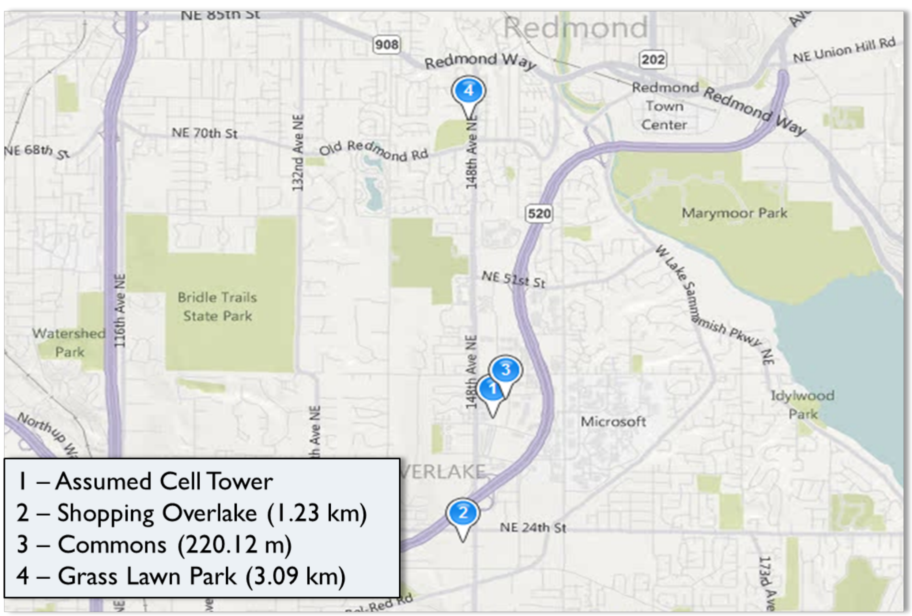

Figure 9: Scenario of evaluation

[4] http://www8.garmin.com/products/gpsmap60cs

4.1.1 Error overview

Next plots show an overview of the behavior of the absolute error when we calculate the position by using the software provided by the book Borre et al. (2006), depicted as GPS, and when we use our proposed method with samples of 11, 50, 100, 300, and 500 ms, namely LEGPS011, LEGPS050, LEGPS100, LEGPS300 and LEGPS500, respectively. As for every analyzed chunk we are able to compute the user's position, we compute the final position by averaging the results of some analyzed chunks. This strategy increases the confidence of the calculations. For our analysis, we consider the maximum of 50 chunks, i.e., when LEGPS is considering, only 11 chunks, but when other strategy is used, 50 chunks are computed.

The results were obtained from traces collected at the locations listed in Table 2, and compared to the ground truth (the position reported by the Garmin MAP 60CS).

Observe that when the original GPS software is used, 80.76% of the samples present an absolute error less than 10%, i.e., within the expected error of a regular GPS system. These results are shown in Figure 10. For these values, there is no significant difference between the software GPS and the Garmin MAP 60CS. Some measures (19.23%) presented more than 10 m of absolute error, being only 7.6% greater than 20 m. This is an indication that the software GPS presented on the book Borre et al. (2006) leads to very similar results than the Garmin MAP 60CS.

As we mentioned before, our solution, that analyzes only a very small portion of the signal, introduces some additional error. This is shown in Figure 10 where we can see that 85.38% of the measures present less than 40 m of error. This plot shows the results for all strategies together, i.e., with different number of chunks of the signal being analyzed.

In Figure 10 we can also observe that for 300 and 500 ms, 76.92% and 69.23% of the measures present the absolute error less than 20 m, respectively. For 11, 50, and 100 ms the error tends to be spread presenting the max value of 78.72 m when 11 ms of the signal is used. The results presented in this section show that the LEGPS solutions can be adapted according to the application requirements. For instance, if the application requires only a city block precision, the use of 11, 50 or 100 ms is affordable. In case of more accurate requirements, the system can use 300 or 500 ms, being 300 a good trade-off of the number of chunks and the precision. The following sections show a more detailed analysis of the absolute error.

4.1.2 Average error in function of the number of satellites

Next plots shows the average of the absolute error in function of the number of satellites. Figure 11 shows that the the number of satellite in view poses a strong influence on the error. Observe that when the receiver can see only five satellites, all strategies present the largest errors and when the number of satellites increases, the error decreases proportionally. This is an expected result for current GPS systems. We can see that, depending on the application requirements, with five satellites we can use more data in order to decrease the error. If the application requires only a city block accuracy, even 11 ms is affordable.

In Figure 12 we can see that the location, i.e., the distance to the assumed

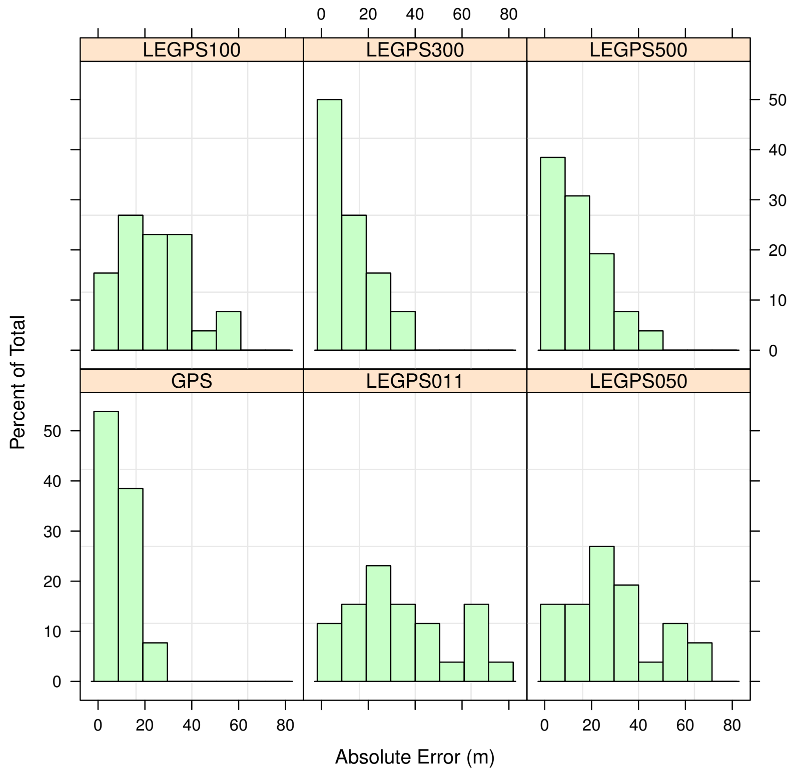

Figure 10: Error histogram for each analyzed strategy

cell tower, does not play an important role in terms of accuracy. Observe that the Grass Lawn Park location is the farthest while Microsoft Commons is the closest. Even though the Grass Lawn Park present the highest errors, it only occurs when only five satellites are in view. Therefore, when the number of satellites in view is six, the results are very similar to the other locations. The missing values (for instance, 7 and 8 satellites for Grass Lawn Park) occurs because sometimes we are not able to find this number of satellites in the correspondent trace. We can see that if the distance to the cell tower is within the maximum allowed to estimate the millisecond part of the pseudo-range correctly, it does not lead to additional error and our solution is affordable in the sense that the cell tower can be used as the initial position in the user's position calculation.

4.1.3 Average error in function of the number of analyzed chunks

Figures 13 and 14 show the behavior of the absolute error in function of the number of analyzed chunks. In Figure 13 we can see, as expected, that the standard GPS

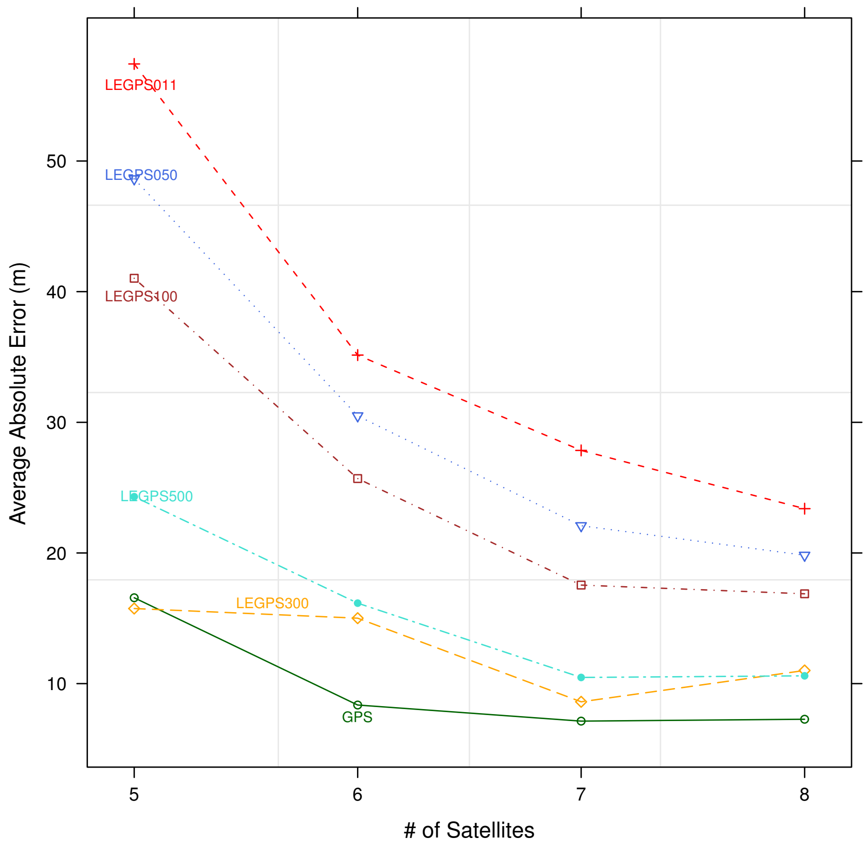

Figure 11: Error in function of the number of satellites

present the better results. The error variation is close to 10 m. LEGPS300 and LEGPS500 strategies present a good performance, close to the standard GPS, with the third quartile presenting values less than 20 m. When the number of chunks decrease, the error increase as well. Observe that for LEGPS011, i.e., our baseline proposal, the third quartile is less than 50 m and the median is about 30 m. This precision is affordable, for example, to applications that require city block level of accuracy.

In Figure 14 we can see the same results but considering the location. We can see that the Grass Lawn Park measures present the largest interval between the first and third quartile. This behavior can be explained due the fact that the number of satellites in view is small, typically 5 or 6, as we can see in Figure 12. All other locations present a very small interval between the first and third quartile, indicating that the measures are consistent. We also observe that the LEGPS011,

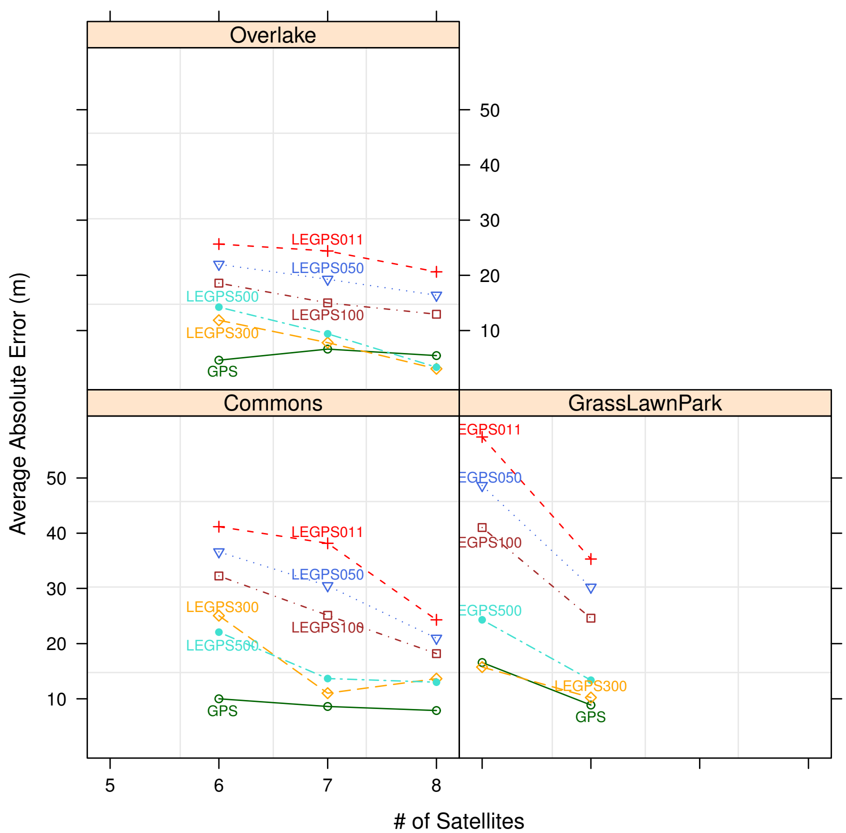

Figure 12: Error in function of the number of satellites and location

typically, presents the largest interval. For the measures that were taken at Shopping Overlake, we can see a very small interval due the large number of satellites in view during the data collection. Even though the number of satellites in view at Microsoft Commons is also large, at Shopping Overlake, it is larger in average, leading to better results.

4.1.4 Average DOP in function of the number of analyzed chunks

An usual measure of the quality of the position reported by a GPS device is the so-called dilution of precision (DPO). This measure indicates the how good the geometry of the satellites in view is to the actual position calculation. For instance, if the satellites are alined or close together, the position cannot be calculated with good precision. Other factor that influences the DOP measure is the presence of

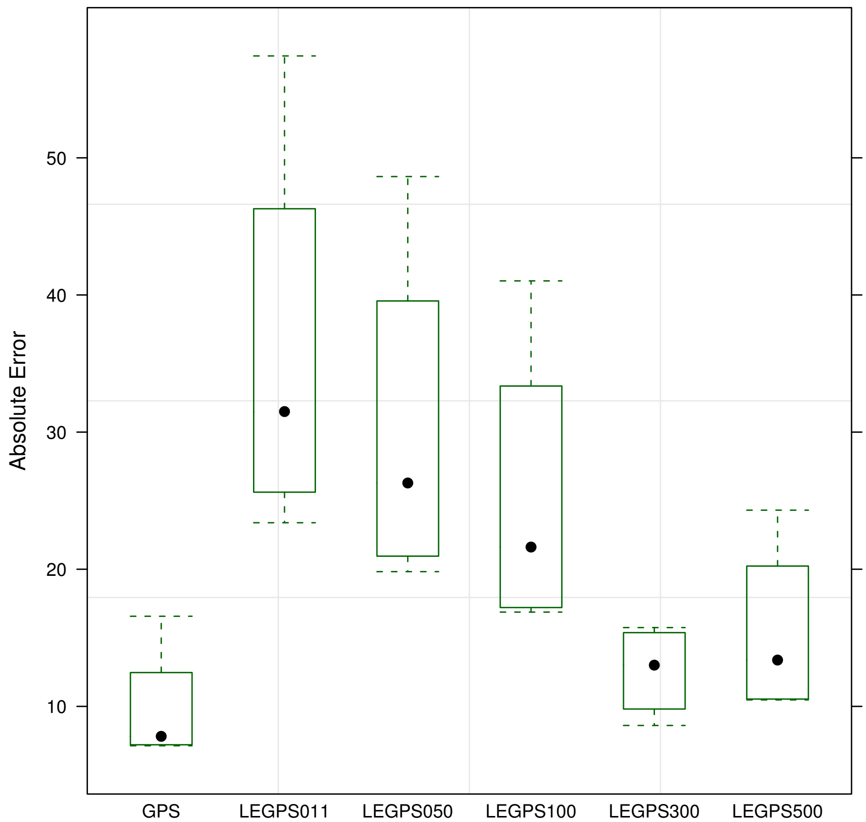

Figure 13: Error in function of the number of chunks

obstructions such as nearby mountains and buildings. DOP measure is calculated by the diagonal elements of the matrix $Q = (H^T H)^{-1}$. Figure 15 shows the average DOP of all techniques herein evaluated. As shown, all LEGPS techniques present the same DOP, that indicates a good geometry and a clear view of sky. All the values are less than 6 and typical values are less than 5, being 6 a moderate DOP and 5 a good one (van Diggelen 2009). In terms of DOP, we can see that all LEGPS strategies lead to a good to moderate quality.

4.1.5 Energy trade-off

In this section we start a discussion of the viability of LEGPS in a wide variety of scenarios. The basic idea is to estimate the energy savings in comparison to the current GPS solutions.

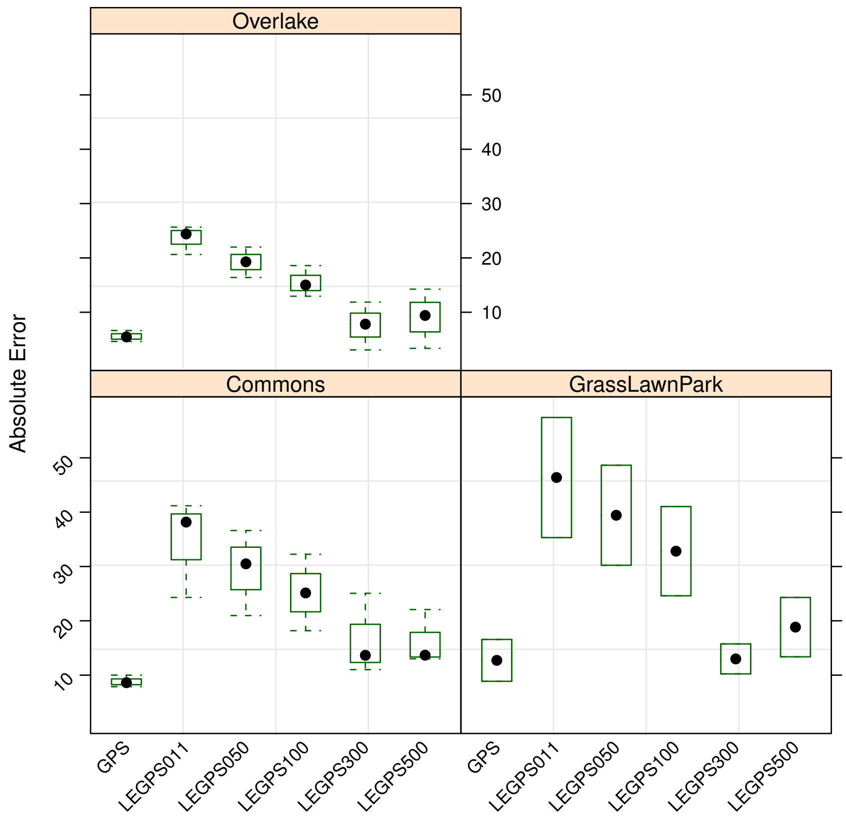

Figure 14: Error in function of the number of chunks and location

Table 3 presents some estimated values of the necessary energy to processing the data and the amount of data to be transmitted. The values of the estimated energy were calculated based on the values obtained from the Figure 7 and are conformable only to the mobile phone herein used. For instance, lets consider the estimation of the LEGPS011. For this calculation, we are considering $\varepsilon_{Processing} = \varepsilon_{Acquisition} + \varepsilon_{Tracking}$ once the position calculation is to be performed on the server side. So, the acquisition and tracking can be fused together and the energy drain can be estimated as $11.10^{-3} \times 700.10^{-3} = 7.7$ mJ for each position query. The remain values present on Table 3 were calculated by similar manner.

Lets again considerer the case of 11 ms and lets compute the amount of data that should be sent trough the network. In this case, we need to send 11×12 doubles ($\varphi^{(k)}$), plus 12 5-bits integers (PRNs), plus 1 double (time), performing a total of 539.5 bytes. If we consider additional computing on the receiver side,

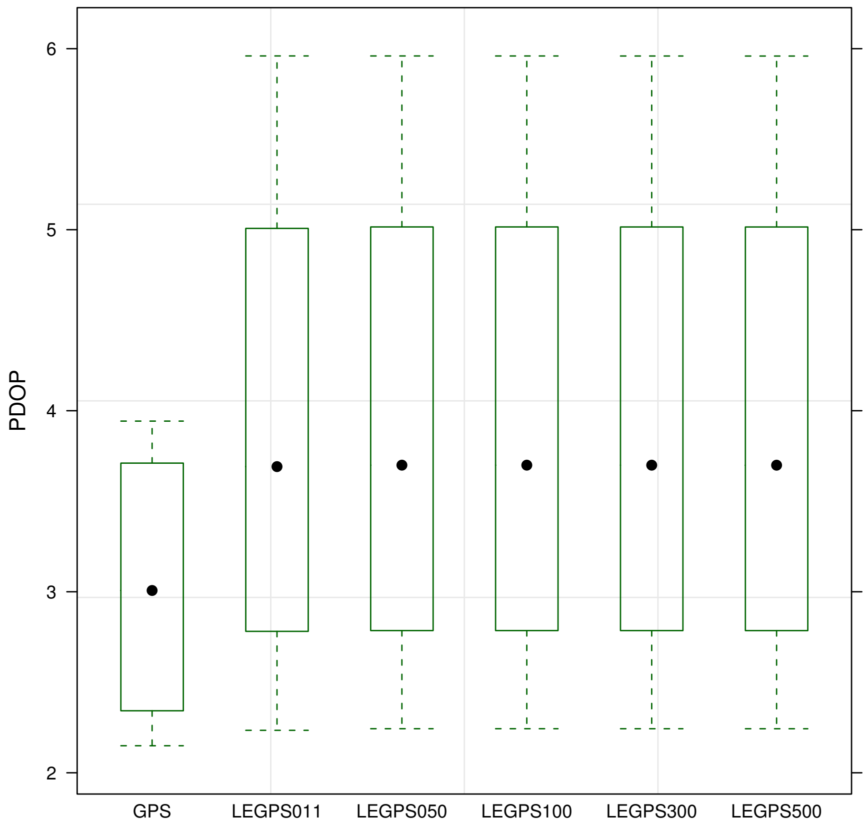

Figure 15: Error in PDOP in function of the number of satellites

we can pre-processing the sub-millisecond pseudo-ranges and and send only one most representative $\varphi^{(k)}$ instead 11 samples, forming a total of 59.5 bytes. As we are considering the maximum of 50 chunks to compute the user's position, the amount of data is the same for strategies LEGPS050, LEGPS100, LEGPS300, and LEGPS500.

With the values shown in Figure 3 we can envision some application scenarios and the energy trade-off, as following:

Standalone: For this scenario, there is no need of real-time position but the receiver can be assisted by the network therefore the cost to send the data trough the network does not pay-off. Thus, the techniques herein discussed are useful to decrease the GPS' duty-cycle but the position itself is calculated on the receiver-side. Additional receiver energy will be drained to compute the least-square method. When real-time data are required, for instance to turn-by-

Table 3: Energy trade-off

Strategy	$\varepsilon_{Processing}$ (mJ)	Transmission Regular (bytes)	Transmission Optimized (bytes)
LEGPS011	7.7	539.5	59.5
LEGPS050	35	$2,411.5$	59.5
LEGPS100	70	$2,411.5$	59.5
LEGPS300	210	$2,411.5$	59.5
LEGPS500	350	$2,411.5$	59.5
AGPS	$7,000$	–	–

turn navigation, the LEGPS needs to tracking the signal continuously and its behavior will be the same as the current AGPS.

Networked: This scenario includes all applications that need to send some data to the network. For instance mobile search and geo photo tagging need to send some data to the cloud (the search query and the picture). For this applications, LEGPS can keep sensing the user's sub-millisecond pseudo-range in a very low duty-cycle and send the data opportunistically. The position data collected by this approach can be used to create an user's position log that can be used as input to many application. For example, mobile search can take advantage of the user's past positions.

5 Final remarks

This document presented a deep study on GPS theory, proposed a new low-energy GPS system, and showed some experiment results to test the proposed solution. We used a software-based GPS lab kit in order to assess our proposal and we got promising results. We show that GPS devices, with adequate assistance, can reduce the time to get the position from $6-10$ s, typically, to some milliseconds (about 11 ms). This approach leads to a large amount of energy saving once we only need to collect some milliseconds of the signal, and thus, the radio needs to be on only during this short period, leading to a very low duty-cycle.

Even though our solution leads to additional position error, we argue that it is affordable for a wide variety of applications. For instance, our solution can be applied to continuously sense the user's position and provide valuable location information to mobile location-based applications such as mobile search, geo photo tagging, and weather forecast, among others. Those applications require city block level precision only, and it is the precision provided from our baseline solution when we use LEGPS with 11 ms of signal and only 5 satellites in view. In case of applications that require more accurate results, our solution can use, for instance, 300 ms of the signal and maintain similar precision as standard GPS devices that use 30 s, or current AGPS devices that use about $6-10$ s.

References

Akopian, D. & Syrjarinne, J. (2009), 'A fast positioning method without navigation data decoding for assisted gps receivers', *IEEE Transactions on Veicular Technology* **58**(8), 4640–4645.

Borre, K., Akos, D. M., Bertelsen, N., Rinder, P. & Jensen, S. H. (2006), *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, Birkhäuser.

Kaplan, E. D. & Hegarty, C. J. (2005), *Understanding GPS: Principles and Applications*, second edn, Artech House.

Lin, K., Kansal, A., Lymberopoulos, D. & Zhao, F. (2010), Energy-accuracy trade-off for continuous mobile device location, *in* 'MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services', ACM, New York, NY, USA, pp. 285–298.

Misra, P. & Enge, P. (2006), *Global Positioning System: Signals, Measurements, and Performance*, Ganga-Jamuna Press.

Paek, J., Kim, J. & Govindan, R. (2010), Energy-efficient rate-adaptive gps-based positioning for smartphones, *in* 'MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services', ACM, New York, NY, USA, pp. 299–314.

Sirola, N. (2001), A method for GPS positioning without current navigation data, Master's thesis, Department of Electrical Engineering. Tampere Univeristy of Technology.

Sun, G., Chen, J., Guo, W. & Liu, K. J. R. (2005), 'Signal processing techniques in network-aided positioning', *IEEE Signal Processing Magazine* pp. 12–23.

Syrjarinne, J. (2000), Time recovery through fusion of inaccurate network timing assistance with gps measurements, *in* 'Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on', Vol. 2, pp. WED5/3 – WED510 vol.2.

van Diggelen, F. (2009), *A-GPS: Assisted GPS, GNSS, and SBAS*, Artech House, Boston/London.

Zhuang, Z., Kim, K.-H. & Singh, J. P. (2010), Improving energy efficiency of location sensing on smartphones, *in* 'MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services', ACM, New York, NY, USA, pp. 315–330.