International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

CO-GPS: Energy Economical GPS tracking with IoT

A.Manasa (M.Tech.) & M.Krishna (Associate Professor)
Department Of Ece, [1]kshatriya College Of Engineering, Chepur, Ts, (503224), India
anugandulamanasa12@gamil.com[1] mudarakolla@gmail.com[2]

Abstract
Location is a fundamental service for mobile computing. Typical GPS receivers, although widely available for navigation purposes, may consume too much energy to be useful for many applications. Observing that in many sensing scenarios, the location information can be post-processed when the data is uploaded to a server, we design a Cloud-Offloaded GPS (CO-GPS) solution that allows a sensing device to aggressively duty-cycle its GPS receiver and log just enough raw GPS signal for post-processing. Leveraging publicly available information such as GNSS satellite ephemeris and an Earth elevation database, a cloud service can derive good quality GPS locations from a few milliseconds of raw data. Using our design of a portable sensing device platform called CLEON, we evaluate the accuracy and efficiency of the solution. Compared to more than 30 seconds of heavy signal processing on standalone GPS receivers, we can achieve three orders of magnitude lower energy consumption per location tagging.
Keywords: *Location, Assisted GPS, Cloud-offloading, Coarse-time Navigation*

1. INTRODUCTION

Localization is a fundamental service in mobility. In outdoor applications such as wildlife tracking [22], [20], participatory environmental sensing [15], and personal health and wellness applications, GPS is the most common location sensor. GPS receiving, although becoming increasingly ubiquitous and lower in cost, is processingintensive and energy-consuming. Take ZebraNet sensor nodes [22] as an example. On average, one GPS location fix requires turning on the GPS chip for more than 25 seconds at 462mW power consumption, which dominates its energy budget. As a result, the unit is equipped with a 540-gram solar cell array and a 287-gram 2A-h lithium-ion battery in order to support one GPS position reading every 3 minutes. Power generation and storage accounts for over 70% of the sensor unit's total weight of 1151 grams. Similarly, in wearable consumer devices such as fitness trackers, high energy consumption from GPS receivers mean bulkier devices and low battery life.

The global positioning system (GPS) is the most pervasive technology that provides this fundamental service. The ubiquity of GPS has, henceforth, grown beyond billions of smart phones to embedded devices for enabling many novel outdoor applications across several domains. GPS receivers have, therefore, become more versatile in terms of cost, size and weight; but are still demanding in energy usage. It is an artifact of the computationally intensive GPS receiving operation, which accounts for more than 80% of the total energy expenditure of the sensing platform on which it is coupled [1]–[3].

The high energy profile is a combined effect of two primary factors. First, the GPS Ephemeris data, which contains the time and satellite trajectory information, are sent by the satellites at a very low data rate of 50 bps. As a result, a standalone GPS receiver needs to be turned on for up to 30 seconds in order to receive a complete data packet from the satellite. Second, the task of identifying and tracking the visible satellites, decoding their navigational information and performing the least-square calculation involves a significant amount of processing. It gets 1Prasant Misra∗ was a postdoctoral fellow, and Wen Hu† was a visiting researcher at SICS Swedish ICT in Stockholm during the course of this work. more intensive due to the weak GPS signal strengths (about 20 dB below the noise level); and Doppler frequency shifts (≈ 4.2 kHz) caused by the satellite motion (and receiver movement on the ground) [4]. As a consequence, the first factor makes it difficult to duty-cycle the GPS receiver for saving energy; while the second necessity introduces the need for a sophisticated CPU for complicated computations. The existing state-of-the-art methodology of obtaining a GPS position fix is, therefore, not suitable for many mobile sensing applications such as livestock [5] and wildlife monitoring [6]; which need non-intrusive position tracking support on resource constrained platforms (such as sensor nodes) for long durations.

3) The satellites move at high speed. When a GPS chip is turned off completely for more than a few minutes, the previous code phases and Doppler information are no longer useful, and the device must spend substantial energy to re-acquire the satellites.

4) Post-processing and least-square calculation require a powerful CPU.

In this paper, we address the problem of energy consumption in GPS receiving by splitting the GPS location sensing into a device part and a cloud part. We take advantage of several key observations.

- Many mobile sensing applications are delaytolerant. Instead of determining the location at the time that each data sample is collected, we can compute the locations off-line after the data is uploaded to a server. This is quite different from the turn-byturn navigation scenario that most standalone GPS devices are designed for. The benefit is even

more significant if the data uploading energy is amortized over many data samples.

- Much of the information necessary to compute the location of a GPS receiver is available on line. For example, NASA publishes satellite ephemeris through its web services, so the device does not have to stay on long enough to decode them locally from satellite signals. The only information that the device must provide is a rough notion of time, the set of visible satellites, and the "code phase" information from each visible satellite, as we explain in section 2.
- Code phases can be derived from any millisecond of satellite signal. If we can derive location without decoding any data from the satellites, there is significant opportunity to duty-cycle the receiver.
- In comparison to the constraints on processing power and energy consumption, storage is relatively cheap to put on sensor devices, so we can liberally store raw GPS intermediate frequency (IF) signals together with sensor data. For example, at an IF of 4MHz, sampling a one-millisecond signal at 2-bit resolution and 16MHz rate results in 4KB raw data.

Due to the split between local and cloud processing, the device only needs to run for a few milliseconds at a time to collect enough GPS IF signals and tag them with a rough time stamp. A cloud service can then process the signals off-line, leveraging its much greater processing power, online ephemeris, and geographical information to disambiguate the signals and to determine the location of the receiver. We call this approach Cloud-Offloaded GPS (CO-GPS).

The CO-GPS idea is built on top of a GPS receiving approach called Coarse-Time Navigation (CTN) [21]. While CTN is used for quickly estimating the first location lock (measured as Time To First Fix, or TTFF), we are the first to articulate and quantify its energy saving benefits. Furthermore, we find ways to relax the condition on knowing a reference location that is close to the true location, and maintaining a real-time clock that is synchronized to the satellite clock. As a result, COGPS receivers can have an extremely short duty cycle for long-running tracking applications.

This paper extends [10] by investigating ways to remove satellite detection outliers, which are more likely to be false positives due to weak signal strength and short signal length. As a result, through our empirical evaluation over 1500 real GPS traces, median location error drops from 30m to 12m, and more than 85% of samples have less than 30m error. Furthermore, we removed the dependency on relatively energy-consuming WWVBbased time synchronization, and leverage time stamps resolved from GPS signals themselves to progressively time stamp samples in data traces. A node only needs to be time synchronized once at the beginning of its deployment.

We built a sensor node, called CLEON, based on the CO-GPS principle using a GPS receiving front end chip MAX2769 and a MSP430 microcontroller. We also built and deployed CO-GPS location resolution web service on Windows Azure. Through performance measurements, we show that it takes as little as 0.4mJ to collect enough data to compute a GPS location, in comparison to the order of 1J for GPS sensing on mobile phones or 300mJ for u-blox Max-7 GPS module In other words, with a pair of AA batteries (2Ah), CLEON can theoretically sustain continuous GPS logging (at 1s/sample granularity) for 1.5 years.

To make this paper self-contained, the rest of the paper is organized as follows. In section 2, we first give an overview of how a typical GPS receiver processes satellite signals, in order to motivate our solution. Section 3 describes the principle of CO-GPS. In section 4, we discuss the implementation of the cloud side services. We evaluate the performance of CO-GPS and its parameters using real GPS traces in section 5. Finally, section 6 presents the design of the CLEON sensor node and evaluates its energy consumption in GPS receiving.

COARSE-TIME NAVIGATION

A GPS receiver computes its location by measuring the distance from the receiver to multiple GNSS satellites (also called space vehicles, or SVs for short). Ultimately, it needs to infer three pieces of information: • A set of visible SVs and their current trajectories. The current trajectory parameters, called ephemeris, are sent from the satellites every 30 seconds. • A precise time T when the GPS signals left the satellites. • The distances from the receiver to each SV at time T , often called the pseudoranges. Typically, these are obtained by processing the signals and data packets sent from the satellites. With them, a receiver can use least-square (LS) minimization to estimate its location. Due to page limits, we will not give details of GPS or Assisted GPS (A-GPS) receiving. Interested readers can refer to [8]. We will directly describe the Coarse-Time Navigation principle underlying CO-GPS. Like standalone GPS receivers, CTN starts with the acquisition process, where the received satellite signals are correlated with known 1023-bit GPS Gold codes (aka Coarse/Acquisition codes, one for each satellite). A C/A code repeats every millisecond. Because of the relative motion between a satellite and the receiver, the received signals will have a Doppler shift from the transmission frequency (L1 = 1:575GHz). So the receiver has to search in both Doppler and code phase dimensions to find the correlation peak. The code phase is the duration between the time that the receiver starts processing a sample to the beginning of a C/A code in that sample. At the speed of light, it takes about 50 to 80 ms for the

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

GPS signal to propagate from the satellite to the receiver. Let T0 be the time that a first bit of a C/A code leaves the satellite, and Tr be the time that the receiver receives the same bit. Then, the propagation delay Tr − T0 has two parts, the number of full milliseconds (NMS), and

2.BACKGROUND AND RELATED WORK

This section describes literature work on location determination techniques and how the energy efficiency in smartphones can be improved and battery lifetime can be improved. The literature work can be categorized into different sections as described below:

A. Global Positioning System (GPS)

GPS is a satellite based positioning and navigation system which is used to locate GPS receiver: (i) Almost Sixteen Monitor stations are distributed around the world that collect transmissions from the GPS satellites continuously and send them to the Master control station. (ii) The Master control station, which are 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)Thirty satellites orbit Earth twice a day. These satellites continuously and simultaneously broadcast the Navigation message. (v) A mobile device (GPS receiver) determines its position information from Navigation messages received from the satellites.
Each Navigation message consist of five sub frames which are of 300 bits long and are transmitted over a very slow link such as 50bps, one complete transmission ma take 30 seconds. The first frame (sub frame 1 in Fig. 2) consist of timing information, the next two frames (sub frame 2 & 3) consist of ephemeris data which refers to the satellite's precise orbit and is valid for about 30minutes. The last two frames (sub frame 4 & 5) consist of almanac data which refers to satellite's coarse orbit and status information of all satellites, it may be valid up to several months. The whole almanac data may take about 25 frames or at least 12.2 minutes as there are number of satellites around the world. GPS receiver may see only few satellite but still it needs 12.2 minutes because it has to tune to each frequency/codes. When a mobility device is started or if it does not have location information already stored, TTFF (timeto-first-fix) would be at least 12.2 minutes because it has to identify its current position (Cold start).But normally, mobile devices have some localization data received from GPS in recent past which make it possible to search only subset of satellites and receive navigation messages. TTFF (i.e warm start) would require several seconds to minutes to obtain navigation message.

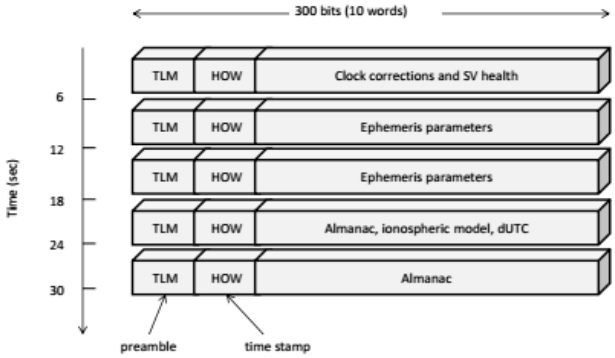


Fig. 1: Frame format of GPS Navigation message

B. Assisted GPS (A-GPS)

A-GPS technique was developed to minimize TTFF, which was important aspect for smartphone applications. A smartphone device with A-GPS feature can receive GPS localization data via cellular service. The mobile device position estimate is provided to A-GPS servers by cell towers. Now on receiving the position estimate of mobile device, the A-GPS server evaluates the satellites visible to mobile device, and sends GPS localization data to the mobile device. Once GPS initial data is received by mobile system it can switch to the real data once available. Though A-GPS minimizes the TTFF it has several shortcomings. First, the process of fetching initial GPS positioning data from cellular network could consume greater energy than Raw GPS itself [6, 8, 9]. Secondly, instead of free GPS data it makes use of costly cellular data. Third, A-GPS technique depends on OS design layering and cellular networking [7, 8]. Fourth, mobile device location privacy is compromised by reaching the networked A-GPS servers.

2.1 GPS Receiving Overview

A GPS receiver computes its location by measuring the distance from the receiver to multiple GNSS satellites (also called space vehicles, or SVs for short). Ultimately, it needs to infer three pieces of information: A precise time T ; A set of visible SVs and their locations at time T ; The distances from the receiver to each SV at time T , often called the pseudoranges. Typically, these are obtained from processing the signals and data packets sent from the satellites. With them, a receiver can use least-square (LS) minimization to estimate its loca-tion. To make this paper self-contained and to motivate our so-lution, we give a brief (and much simplified) description of the GPS receiving process. We start with standalone GPS re-ceiving and then discuss Assisted GPS (A-GPS) and in par-ticular a

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

technique called coarse-time navigation. For a more formal treatment of the principles of GPS and A-GPS, please refer to [12, 27].

2.2 GPS Signals

There are 31 (plus one for redundancy) GNSS satellites in the sky, each orbiting the Earth about two cycles a day. A set of ground stations monitor the satellites' trajectory and health, and send the satellite parameters to the satellites. These parameters include two kinds of trajectory informa-tion: the almanac, which contains the coarse orbit and status information, and the ephemeris, which contains the precise values of the satellite's trajectory. All satellites are time-synchronized to within a few microseconds, and after clock correction, their time stamps can be synchronized within a few nanoseconds. The satellites simultaneously and continuously broad-cast time and orbit information through CDMA signals at L1 =1:575 GHz towards the Earth. The bit-rate of data pack-ets is a mere 50 bps, but the bits are modulated with a higher frequency (1MHz) signal for detecting propagation delays. A full data packet from a satellite broadcast is 30 seconds long, containing 5 six-second-long frames, as shown in Fig-ure 1. A frame always starts with a preamble (called Teleme-try Word, or TLM) and a time stamp (called Handover Word, or HOW). Each data packet contains the ephemeris of the transmitting satellite and the almanac of all satellites. In other words, a precise time stamp can be decoded every 6 seconds, and the high accuracy satellite trajectory can be de-coded every 30 seconds. This low data rate explains why a standalone GPS, as seen in vehicles, can take up to a minute to acquire its first location, a metric called time to first fix (TTFF). The ephemeris information is constantly updated by the ground stations. In theory, the ephemeris data included in the SV broadcast is only valid for 30 minutes.

While the precise time and satellite locations are decoded from the packets, the pseudorange from each SV to the receiver is obtained using much lower-level signal processing techniques. For that, we need to understand GPS signal modulation.

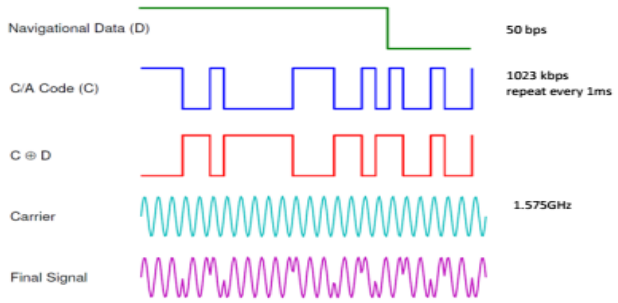


Figure 3. An illustration of GPS signal modulation scheme.

As illustrated in Figure 2, each satellite encodes its signal (CDMA encoded) using a satellite-specific coarse/acquisition (C/A) code of length 1023 chips at 1023 kbps. Thus, the C/A code repeats every millisecond, resulting in 20 repetitions of the C/A code for each data bit sent. The purpose of the C/A code is to allow a receiver to identify the sending satellite and to estimate how long it took the signal to propagate from that satellite to the receiver. Typically, the GPS signals take from 64 to 89 milliseconds to travel from a satellite to the Earth's surface. Since light travels at 300 km/ms, in order to obtain an accurate distance measurement, the receiver must estimate the signal propaga-tion delay to the microsecond level. The millisecond (NMS) and sub-millisecond (subMS) parts of the propagation time are derived very differently: the NMS is decoded from the packet frames, while the subMS propagation time is detected at the C/A code level using correlations

2.3 Acquisition

When a GPS receiver first starts up, it needs to detect what satellites are in view. This is done by detecting the presence of the corresponding C/A codes in the received signal, typ-ically by correlating the signal with each known C/A code template. Since the C/A codes are designed to be orthogonal to each other, a visible satellite will show a spike in the correlation results, and an invisible satellite will not cause any detectable spike. A challenge in acquiring satellites is the Doppler frequency shift caused by the motion of the satellite and by any movement of the receiver on the ground. For example, a rising GPS satellite can move at up to 800m/s towards a receiver, causing a frequency shift of L1 · 800=c = 4:2kHz, where c is the speed of light.

A shift of the same magnitude occurs in the opposite direction for a setting satellite. To reliably compute a correction under this shift, the receiver must generate the C/A code within 500Hz of the shifted frequency. Therefore, in the frequency dimension, the receiver needs to search up to 18 bins. Most GPS receivers use 25 to 40 frequency bins to accommodate local receiver motion and to provide better receiver sensitivity. After compensating for the Doppler frequency shifts, the receiver must determine code phase delays. Because the receiver does not have a clock synchronized with the satellite, and because the signal propagation delay can be affected by atmospheric conditions, the receiver must search over the delay dimension. The receiver usually over samples the 1023 bps C/A code.

Assuming that the receiver samples the baseband signal at 8 MHz, in a brute force way, the receiver will search 8184 code phase positions to find the best correlation peak. Figure 3 shows an example of

such correlation result in the frequency and code phase search space that indicates a successful satellite acquisition. Code phases change over time as the satellites and the device on the ground move. In continuous operation, GPS receivers use a tracking mode to adjust previously acquired Doppler frequency shifts and code phases to the new ones. This is a relatively inexpensive process, using feedback loops. So, once a GPS produces its first location fix, subsequent location estimates become fast. However, once the GPS receiver stops tracking, the utility of previously known Doppler shifts and code phases diminishes quickly. Typically, after 30 seconds of non-tracking, the GPS receiver has to start all over again.

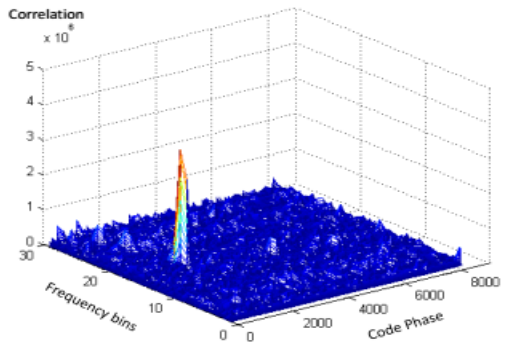


Figure 4: An example of acquisition result

Acquisition is an expensive operation as it must search through 30+ frequency bins times 8,000+ code phase possibilities for every single satellite. If the receiver has some prior information about the satellites, it may be able to do less work. Examples are:

• **Cold Start.** When the receiver has no prior knowledge of the satellites and its own location, it has to search the entire space. Usually, GPS receivers do not buffer the raw data to perform the search, rather they perform one code phase search per millisecond as the signal streams in. Although there are hardware correlators that perform acquisition in parallel, it still takes a few seconds to acquire one satellite. This is one of the main reasons for the slow initial position fix and high energy consumption for standalone GPS devices.

• **Warm Start.** When the receiver has a previous lock to the satellites, it can start from the previous Doppler shift and code phases and search around them. In general, if the previous lock is less than 30 seconds old, a warm start can quickly find the new lock. Otherwise, the receiver has to revert to the cold start process.

• **Hot Start.** When the previous satellite locks are within a second, the receiver can skip the acquisition process and start directly from tracking to refine the Doppler and code phases. In this mode, all information that the receiver needs is already in place. This is the reason that in mobile phones, "continuous GPS sampling" is defined as one location sample per second.

• **A-GPS.** There are multiple ways that an infrastructure can help the GPS receiver start up faster. In particular, in the Mobile-Station Based A-GPS (or AGPSB) mode, the infrastructure provides the up-to-date ephemeris data so that the GPS receiver does not have to decode them from the satellite signals. The first successful decoding of HOW is enough to provide a location fix. In this case, the TTFF is usually around 6 seconds. In the Mobile-Station Assisted A-GPS (or AGPSA) mode, the infrastructure is given the estimated location, so it can provide initial values for Doppler and code phase searches. This allows the receiver to jump directly into the warm start process.

2.3 Location Calculation

An important output of satellite acquisition and tracking processes is the code phase produced by the correlation peaks. It gives the sub-millisecond level propagation delay. If the receiver has decoded the satellite time stamps (HOW), it knows the time that the signals have left the satellites. Then, it can add these sub-millisecond delays to obtain the whole propagation delay and thus the pseudoranges. With correct tracking, the receiver can decode the packets sent by the SVs. In general, without assistance information, the receiver needs to decode SV ephemeris every 30 minutes (its valid time span) and time stamps every 6 seconds. Decoding is energy consuming since it has to run tracking continuously for the packet duration in order to receive all the bits. With A-GPS, the receiver is not required to decode ephemeris, but it must still decode HOW.

Next, given ephemeris, the propagation delays obtained from code phases, and HOW, the GPS receiver performs position calculation using constraint optimization techniques such as Least Squares minimization. Usually, the local clock at the receiver does not know the precise satellite time, so it is treated as one variable in the minimization solver. With the receiver's latitude, longitude, altitude, and the precise satellite time as optimization variables, typical receivers must have at least 4 SVs in view.

Notice that once the satellites are acquired, distance measurements, and thus the location of the receiver, can be estimated every millisecond. A typical GPS receiver will average over multiple Least Square solutions to further reduce noise and improve location accuracy.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

Finally, we discuss how much data is necessary for satellite acquisition. Since the baseband C/A code repeats every millisecond, in the ideal case, 1ms of data is enough for satellite acquisition. Assume a 8 MHz baseband sampling frequency, the minimum amount of data needed for acquisition is 8*1023 = 8184 samples. For most GPS receiver chips, each sample is two bits (one bit sign and one bit magnitude), thus the storage requirement for 1ms of baseband data is 2046 bytes. One corner case that deserves special attention is the bit transition in the middle of the 1ms signal. Since the C/A code is used to modulate the data packets at 50bps, for every 20ms, there is a possibility of a bit transition. If the transition is in the middle of the 1ms sample, then the acquisition of the corresponding satellite will fail. So, in practice, 2ms is more reliable for satellite acquisition.

2.4 Coarse-Time Navigation

The above discussion assumes that the receiver is standalone with no assistance information. A-GPS receivers receive assistance information from servers to improve their TTFF. Typically, the assistance information includes the ephemeris data so the receivers do not have to decode them from the satellite signals. Some A-GPS approaches also provide Doppler shift and code phase guesses to the receiver so their acquisition searches do not start blindly.

One A-GPS mechanism called Coarse-Time Navigation (CTN) is particularly relevant to this paper. With this method, the receiver does not require the timestamp (HOW) decoded from the satellite. Instead, it only needs a coarse time reference and treats common clock bias (i.e. the difference between the receiver clock and the ideal satellite clock) as a variable in Least Square minimization. This method is first described in [27] and further exploited in [23] to reduce mobile phone GPS energy consumption.

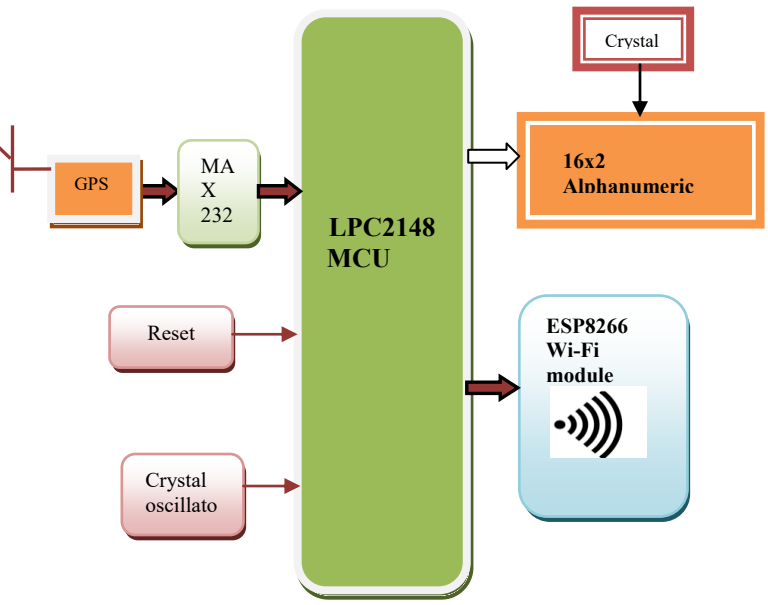
Without decoding the HOW, the receiver cannot synchronize to the satellites' transmission times. So, a key idea in using CTN is to leverage a nearby landmark to estimate NMS. Since light travels at 300 km/ms, two locations within 150km of each other will have the same millisecond part of the propagation delay, rounded to the nearest integer. For mobile phones, it is natural and convenient to use cell tower locations as the landmarks [23]. A cell tower usually covers a radius less than 10km. Although the millisecond part of the pseudorange can be estimated by using the landmark location, the lack of a synchronized clock presents an additional challenge. Since the

3. Design of Proposed Hardware System

CO-GPS DESIGN

The design of Cloud-Offloaded GPS (CO-GPS) leverages the CTN principle but removes the dependency on nearby landmarks. For embedded sensors without cellular connections that are expected to have high mobility over their lifetimes, it is not always possible to provide nearby landmarks. Our key idea is to leverage the computing resources in the cloud to generate a number of candidate landmarks and then use other geographical constraints to filter out the wrong solutions. In this section, we assume that the device is reasonably synchronized with a global clock. We will relax this condition later. When the device needs to sense its location, it simply turns on the GPS receiving front end and records a few milliseconds of GPS signal 2. Our goal is to derive the receiver location offline solely from the short signal and the coarse time stamp.

BLOCK DIAGRAM



3.1 Shadow Locations

The CO-GPS solution assumes a simple flow of information. In addition to application-specific functionality such as sensing, the device has three main components for location sensing — a GPS receiving module, a time synchronization module, and a data

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

storage space. In this section, we assume that the device is reasonably synchronized with a global clock, which we will elaborate upon further in section 5.2. When the device needs to sense its location, it simply turns on the GPS receiving front end and records a few milliseconds of GPS baseband signal. As we discussed in the previous section, we require at least 2ms worth of data to avoid possible bit boundaries. The challenge of deriving receiver location with no reference landmark is the possible outliers, which we call shadow locations. Figure 3.1 illustrate the reason behind it using two satellites. Here, we model the pseudoranges from each SV as a set of waves, each 1 light-ms apart. Clearly, these waves intersect at multiple locations. Since we do not know the exact millisecond part of the propagation delay, all intersections, A;B;C;D;::: are feasible solutions, even though only one of them is real. When more satellites are visible, more constraints are added to the triangulation, which helps resolve the ambiguity. However, the number of satellites alone is not enough.

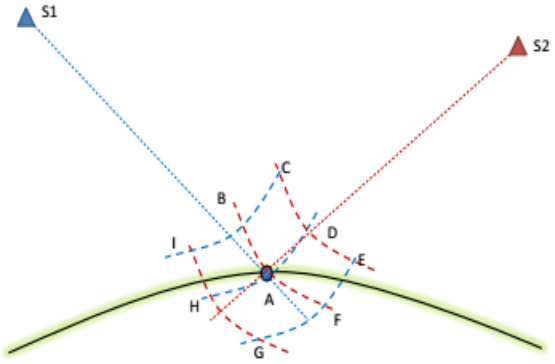


Figure 5: An illustration of multiple feasible solutions under NMS ambiguity

To illustrate this effect, we take a 1ms real GPS trace and apply CTN with an array of landmarks across the globe. There are 6 satellites in view. The landmarks are generated by dividing the latitude and longitude with a 1o resolution around the globe. In other words, we picked 180 ×360 = 64800 landmarks with adjacent distance up to 111km on the equator. Figure 5 shows the total of 166 converged points.

Figure 6: All converged solutions for an example data trace, without landmark knowledge.

3.2 Guessing Reference Locations

The first step in eliminating shadow locations is to reduce the number of possible landmark guesses. Of course, if we know the past location of the sensor and can assume that it has not moved more than 150km between the samples, then we can use the past location as the landmark. However, in the bootstrap process, or when the time difference between readings is large enough to allow movement greater than 150km, we have to assume no prior knowledge of the location of the sensor. Following the acquisition process, we can obtain the set of visible satellites and the frequency bin (identifying the Doppler shift) for each satellite. Knowing the signals' transmission frequency, the Doppler shifts tell us the relative velocity between the satellites and the receiver. If we know the absolute velocities of the satellites, which can be obtained from the ephemeris, then we can derive each angle between the satellite and the receiver, which defines a set of intersecting cones.

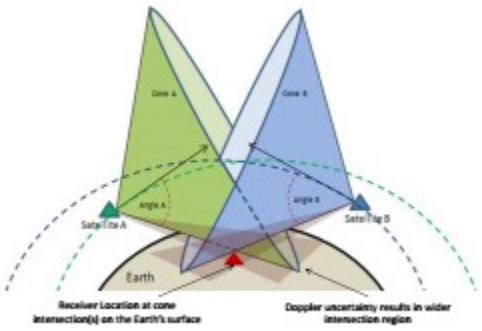


Figure 7: Two satellites A and B at the time of signal reception, the tangents to their orbits, and the angles calculated from the Doppler shifts at the receiver.

3.3 Solution Pruning

Due to the landmark guessing errors, the landmarks alone cannot rule out all shadow locations. Because a

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

light-ms is 300km, the elevation of a shadow location is likely to be far away from the Earth's surface. For example, Figure 7 shows the number of possible solutions when we limit the elevation to be within the [-500, 8000] meter range.

Figure 8: All converged solutions for an example data trace, limited to the [-500, 8000] meter elevation range.

3.4 Accuracy Considerations

A key design consideration of CO-GPS is the tradeoff between accuracy and energy expense. GPS signals are very weak when they reach the Earth's surface, and they suffer from multipath errors and obstruction by objects. Typical GPS receivers use long signal durations and tracking loops to overcome the low signal quality and to improve location accuracy progressively. Notice that the longer the signal is, the more robust the correlation spikes. This is the right thing to do for standalone GPS, since they need to subsequently decode the packet content, which requires good signal quality. However, sampling and storing large quantities of raw data brings energy and storage challenges to embedded sensor devices.

In CO-GPS, the only things we acquire from the signal are the code phases and Doppler shifts. The timing and ephemeris data are all derived off line. Because we do not decode signals, we can use much shorter signal lengths. An additional advantage of this is that it increases the likelihood of detecting satellites that are only intermittently visible, and whose signals fade out over the course of a longer signal sample.

The accuracy of time stamps is another concern. CTN can tolerate a certain amount of time stamp error, since it treats common time bias as another optimization variable. However, when this time error is too big, the least-squares process may not converge, or may converge to a wrong value. From the energy-efficiency perspective, maintaining a highly accurate global clock prevents the device from sleeping for long periods of time. To address the clock synchronization challenge, we can rely on clock radio [19] in different parts of the world. Although technical details vary, the mechanism is the same: a low power radio receiver can tune in to a low frequency band (60kHz in US) to receive atomic clock synchronization signals. The signals are not always available, depending on the receiver location, but is guaranteed to be present for a few hours every day in the entire continental US. Real-time clocks can be used to keep clock drift under a desired threshold between synchronizations. Obviously, the more accurate a real-time clock is, the longer it can sustain correct values between synchronization, and the less energy the device needs to spend on receiving the WWVB signals.

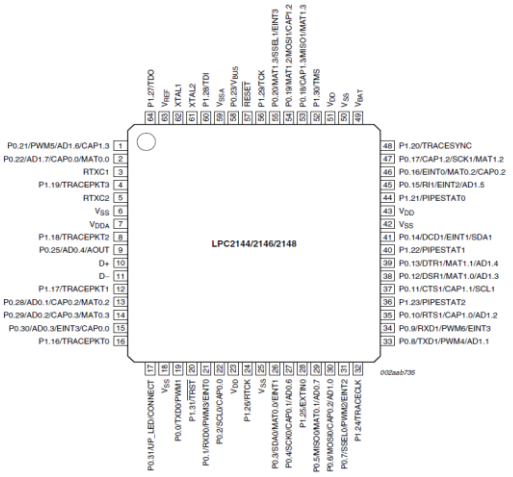
ARM7 MICROCONTROLLER

ARM is an acronym for advanced RISC machine and is manufactured by Phillips. ARM7 is based on reduced instruction set computing architecture. ARM7 is most successful and widely used controller family in embedded system applications. The advantage of low power consumption and low cost increases the range of applications from portable devices to almost all embedded electronic market.

It is preloaded with many in-built features and peripherals making it more efficient and reliable choice for an high end application developer. It also supports both 32-bit and 16-bit instructions via ARM and THUMB instruction set. LPC 21XX series of microcontroller are based on ARM 7 TDMI – S architecture.LPC stands for Low Power Consumption, because for the reason it have different voltages for operation and not like other controllers where the entire controller (CPU + peripherals of controller operate at +5V Vcc).

Pin Diagram

ARM7 LPC2148 microcontroller is a 64 pin dual-in package. There are basically 2 ports in LPC2148, Port0 and Port1. Port0 has 32 pins reserved for it. And Port1 has 16 pins. So total it comes to 32+16 = 48 pins. If it were really 2 ports then the number of port pins should have been 32 + 32 = 64.



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

Fig 9: Pin Diagram of LPC2148

3.5 Web Services

The cloud portion of CO-GPS has two main responsibilities: to update and maintain the ephemeris database, and to compute receiver locations given GPS raw data. We implemented these services on a cloud computing platform, Windows Azure, to achieve high availability and scalability.

3.5.1 Ephemeris Service

There are at least three sources of GNSS satellite ephemeris data sources on the web: • NGS: The National Geodetic Survey of National Oceanic and Atmospheric Administration (NOAA)2 publishes GPS satellite orbits in three types:

Final (igs): The final orbits take into account all possible sources that may affect satellite trajectory. It usually takes NGS a few weeks to process and retrospect all inputs and to make igs available online.

Rapid (igr): The rapid orbits are at least one day behind the current time. Most factors that affect satellite trajectories are taken into account, but not all.

Ultra-rapid (igu): The ultra-rapid orbits are predicted from known satellite trajectories into the near future. NGS's ultra-rapid orbits are published four times a day. Ephemeris published from NGS is in 15 minute intervals, and contains the location and clock correction for each GPS satellite.

NGA: The National Geospatial-Intelligence Agency also publishes an independent GPS orbit data3. This data source, in addition to the satellite positions and clock correction at 5-minute epochs, also contains the velocity vector and the clock drift rate for each satellite. While it may take GNS 12-14 days to produce the final ephemeris, the NGA final ephemeris (called Precise) is usually available with a 2-day latency. In addition, NGA provides up to 7 days of predicted ephemeris, which contains the current ephemeris, but is less accurate.

JPL: While NGS and NGAservicesarefree, NASAJPL provides a paid service called Global Differential GPS (GDGPS) system4. It contains real-time ephemeris (position and clock correction only) updated every minute. Our current implementation uses a combination of NGA and NGS data in the following order. We use NGA Precise as much as possible for historical dates. When NGA Precise is not available, we use NGS Rapids to the most recent date. After that, we use NGS Ultra-Rapids for real-time and near real-time location queries. We implemented a "monitor role" in Azure that periodically fetches data files from NGA and

NGS. Then we perform a polynomial interpolation among the 5 or 15 minute sampling points to obtain a set of parameters [10]. Thus, we have a function that given a satellite and a time stamp returns its location and clock correction at that time. We further differentiate the polynomial to obtain satellite velocity at an arbitrary time. **3.5.2 Location Service**

Putting everything together, the CO-GPS back-end web service must perform the following steps, as shown in Figure 8.

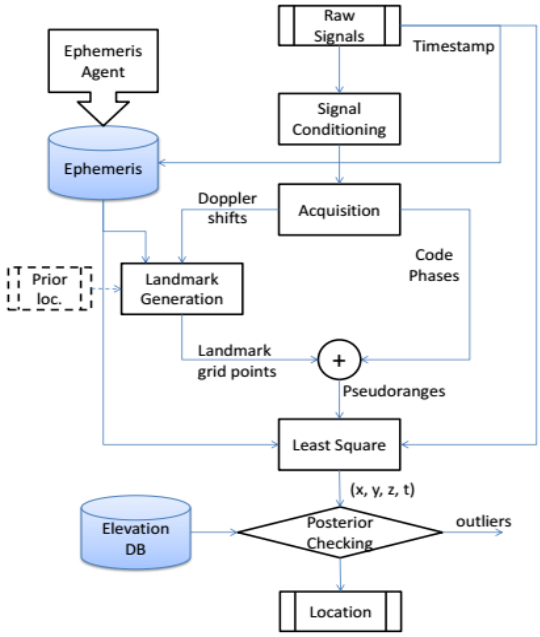


Figure 10. The flow of CO-GPS back-end web service.

4 Evaluation

We evaluate the quality and limitations of the CO-GPS approach using raw GPS samples. This evaluation uses about 100 sets of raw GPS data taken from six different locations in both the northern and southern hemispheres of Earth. We used a SiGe GN3S v3 sampler dongle6, which gives us the flexibility of varying the sample length for evaluation. Our data set contains the baseband GPS signal sampled between 10 to 60 seconds.

To measure the server-side performance, we use a quad-core PC with Intel Xeon 3520 @ 2.66GHz and 6GB mem-ory. The acquisition process takes about 2.6 seconds to fin-ish. Variations in signal length have a linear effects on the execution time mainly due to computing correlations in the frequency domain. Once the code phases are obtained, CTN and least square processes take less than 300ms to calculate the actual location. So if in a bootstrapping process we need to test 10 landmark hypotheses, the total execution time on the server side is less than 6 seconds. For fair comparison, as the ground truth for these samples, we use a software-defined GPS package, called Soft-GNSS [4], to calculate the receiver's positions from the traces. In all data traces, Soft-GNSS achieves Geometric Dilution of Pre-cision (GDOP) values below 6, which is considered

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

at least good. The results produced by Soft-GNSS have expected errors less than 20 m.

When considering the values presented here it is im-portant to note just how different the CO-GPS approach is from a standalone GPS implementation when the same sig-nal trace is used. In addition to regular GPS error sources, CO-GPS adds the following possible sources of error: (i) the position is calculated by using code phase from samples that are closer to each other than in regular GPS; therefore they are more likely to suffer from transient noise that spans over multiple samples, (ii) CO-GPS does not use the lock loops (PLL and DLL) that are implemented in the tracking steps in regular GPS; only the code phase and Doppler fre-quency estimated in the acquisition step are used, which may contribute to less accurate results, and (iii) the use of CTN technique adds additional potential error, especially when the number of satellites is low.

4.1 Acquisition Quality

Since the goal of CO-GPS is to achieve the best possi-ble energy efficiency in GPS sensing, we first evaluated how much data to use and the best duty-cycle strategy to employ to determine an appropriate trade-off between accuracy and low energy use. One of the key parameters that improve sin-gle location calculation is the number of satellites that can be acquired. So, we first vary the parameters to check the ac-quisition quality. A potential method to improve accuracy is to use multiple chunks within each GPS location fix. That is, the receiver wakes up multiple times within a short period and collects chunks of raw samples.

Then the backend service average among the location results from each chunk to obtain the fi-nal fix. As illustrated in Figure 9, the chunk and gap parame-ters define the duty cycling of the receiver when sensing one location. Table 1 shows the three scenarios we evaluated to determine the appropriate amount of data to use to estimate the receiver's position. Each row is a different combination of the number of chunks, the chunk duration in milliseconds, and the time gap (or sleep period) between each chunk.

Table 1. Scenarios of Evaluation

	# of chunks	chunk length (ms)	gap length (ms)
1	1	{2, 4, 6, 8, 10}	0
2	{1, 2, 3, 4, 5}	2	0
3	5	2	{0, 10, 50, 100}

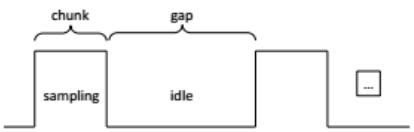
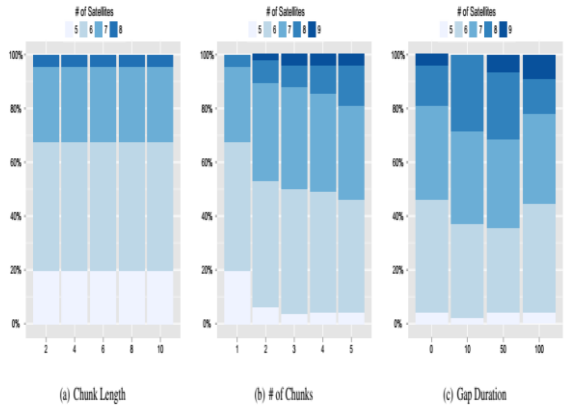


Figure 11: Duty cycling in experimental evaluation. After an idle period (called a gap), the receiver collects a chunk of raw data.

We extract a set of chunks of at least 2 ms duration and average the position outcomes derived from them. When the chunk length is longer than 2 ms, we use the first 2 ms for acquisition and the rest for the tracking loop to refine the code phase and Doppler results.



4.2 Overall Location Accuracy

Figure 12 presents graphs of satellite visibility and error metrics. Figure 12(a) shows the number of visible satellites recognized when each location request contained 5 chunks of 2 ms with a 50 ms gap between them. To improve the location accuracy, we use each chunk independently and then average the result location to obtain the final location. For our data, the receiver often has between 6 and 8 satel-lites in view, with 7 being the most frequent count obtained (Figure 12(a)). We see that in general, the more satellites in view, the more accurate the individual location fixes are (Fig-ure 12(b)). As CO-GPS uses CTN, at least five satellites are required, and the accuracy is greatly improved when more satellites are available. Figure 12(c) compares the the er-ror histogram when we use the single chunk approach, (i.e., when we process only the chunk that yields the largest num-ber of visible satellites and discard the other 4 chunks) ver-sus the error histogram from averaging results from multi-ple chunk.

We observe that when we use multiple chunks, the results have a smaller variance and thus are more accu-rate.The significant improvements are because we are using more information to calculate the position while the samples are independent. Table 2 presents some statistics of the abso-lute error corresponding to the single and multiple chunk ap-proaches. Observe that the mean error is about 20% smaller when the multiple chunk approach is adopted. Figure 13 visualizes some outcomes of CO-GPS's loca-tion estimation for the 6 locations we evaluated. We plot a circle of radius 100 m around the ground truth, represented by a push pin, to

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

give the sense of a block level accuracy (accurate to within a city block). As we can see, on average CO-GPS can achieve < 35m location accuracy with 10ms of data.

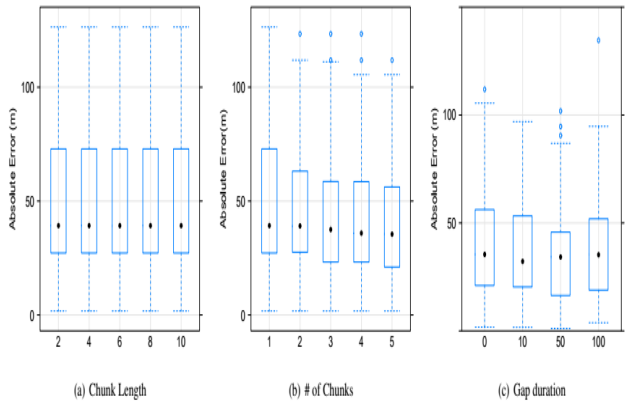


Figure 11: Location error distribution in various experiment settings when single chunk is used for location calculation. In (b), we select the "best" chunk according to the number of satellites in view.

4.3 Time Sensitivity

Two pieces of information are required in order to obtain good results when CTN navigation is adopted: the initial position, and the time stamp corresponding to the moment that the GPS signal was collected. In order to fine-tune CO-GPS's time synchronization mechanism, we evaluated how the error changes as the time drift increases. This is a key parameter that influences how tight the time synchronization must be on a CO-GPS implementation. Figure 14 shows the error boxplots when the time drift increases from 0 up to 300 s. Observe that the error does not change significantly when the time drift varies from 0 to 60 s. After that, the error increases sharply, eventually reaching 106 m. This is due to the fact that under bad ini-tial conditions, CTN navigation is not able to estimate the pseudorange millisecond part properly. Thus, as light travels about 3:105 m/ms, errors of the same order of magnitude are expected due to integer rollover.

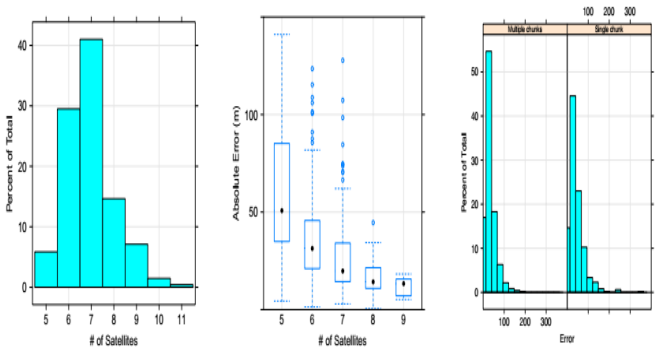


Figure 12: Overall location accuracy distribution.

5. Platform Implementation

Based on the CO-GPS principle and web service, we built a GPS sensor platform, code-named CLEO7. It is a reference design for low power embedded sensing nodes that can log time and the GPS signals received by a mobile object at a high sampling rate. This reference platform consists of a GPS receiver (Maxim MAX2769), a microcontroller (TI MCU-MSP430F5338), a WWVB receiver module for time synchronization, and a serial flash chip for storage and some glue logic. The 8Mbit flash enables storing up to 1000 GPS sample points. The goal of the design is to demonstrate the low energy consumption per GPS sample. In addition, the platform also has a solar cell, a thin-film Micro-Energy cell battery, and a Hi-Jack[13] inspired audio communication port. For the purpose of this paper, the energy evaluation is on the GPS sensing part.

5.1 GPS Sensing

We selected the MAX2769 for the GPS receiver chip due to the relatively low power consumption (18mA in ac-tive mode), support for multiple GNSS standards (GPS, GLONASS, and Galileo), and the simple receiver design with fewer external components. MAX2769 includes a ra-dio front end, RF down converter, and an ADC that gen-erates a bit stream of the sampled down converted RF sig-nal. We used the pre-configured mode 2 (c.f. MAX2769 datasheet[17]) of the receiver which uses a 2 bit ADC to gen-erate 3 output signals : I0, I1, and GPS data clock. Each of these generates data at 16.368 Mbits/s. Capturing the GPS data at this high data rate directly will require the microcontroller to operate at a very high fre-quency. For example, even a DMA based transfer requires 5 clock cycles on the MSP430, requiring a ' 80MHz clock at the microcontroller. This is beyond the operating frequency of the MSP430. Microcontrollers operating with that high frequency will lead to huge power consumption in compar-ison. Hence,

we first use a serial-to-parallel converter glue logic to reduce the microcontroller-side data transfer rate.

5.2 Time Synchronization using WWVB

Correct time stamping is a fundamental requirement to make CO-GPS work. To achieve low-power, high-accuracy time synchronization, we borrowed the approach of using WWVB signals, which is previously exploited in [6]. A CME6005 WWVB module offers universal time signal de-coding from dedicated radio stations around the globe, such as WWVB (in US), DCF77 (in Europe), JJY (in Japan), BPC (in China) etc. Experimental evaluations in [6] have shown that at 2,400 km away from the WWVB station, the signal is available 47% of the time indoors and 75% of the time outdoors, while at 700 km away from the DCF77 station, the signal is available 97% of the time indoors. The time synchronization can achieve an accuracy of 3.9ms (indoor) or 4.3ms (outdoor). Furthermore, the power consumption of CME6005 is low, consuming less than 100µA in active mode and 0.03µA in shutdown mode.

To further reduce energy consumption, CLEO uses a real-time clock (RTC) in the MCU to manage time between syn-chronizations with WWVB. A normal 32,768 Hz crystal for RTC has an accuracy of 20ppm, which may induce a maxi-mum of 1.728s drift over the course of a day. In the previ-ous section, we have validated that CO-GPS can tolerate up to 60s time drift without increasing the location calculation error. This means that CLEO only needs to perform time synchronization once a month, which greatly reduces the en-ergy consumption for receiving WWVB signals. Although the receiver itself only consumes 100µA active power, the MCU needs to be active to decode the time synchronization messages. Assuming that the device needs to run continu-ously for 12 hours to successfully receive the WWVB signal, amortizing the cost over 30 days will result in only 30µA av-erage power consumption. Further reduction of WWVB sig-

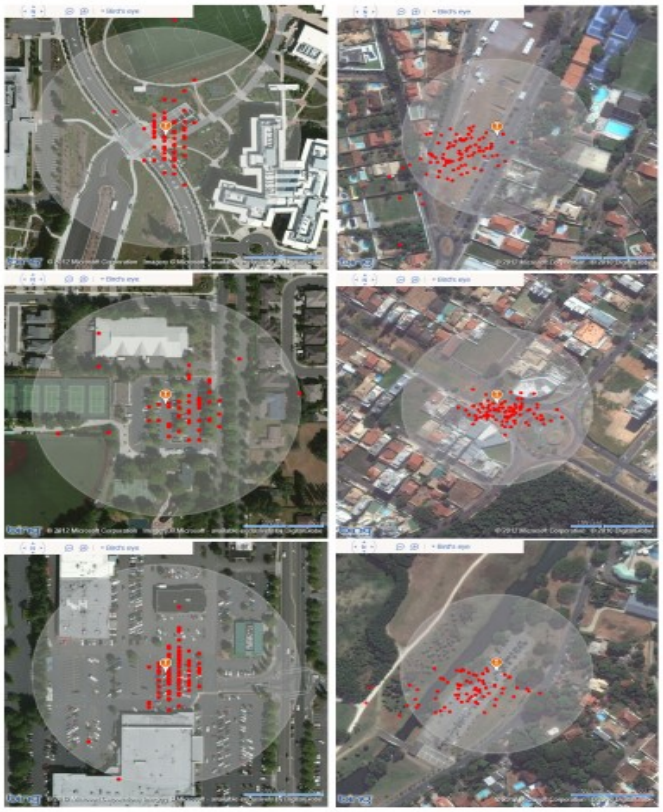


Figure 13: Overall results from 6 locations. The shadow is 100m in diameter. We see that there are bias errors in some cases.

5.3 Energy Consumption Evaluation

We use several techniques to control the power consump-tion of the platform. First, we turn off the power to modules with high leakage currents and make use of different power modes. We completely turn off the GPS receiver and the glue logic by using the Enable pin of the dedicated voltage regulator. We use glue logic with low-leakage power off ca-pability to reduce the leakage current between the glue logic and the microcontroller.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

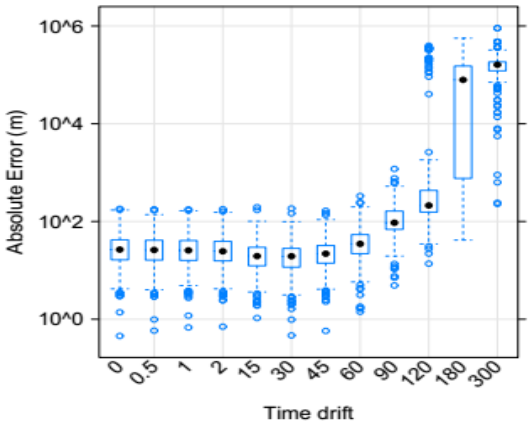


Figure 14: Errors due to time drift (in seconds).

When not active, we operate the mi-crocontroller at the LP3 sleep mode, with only the real time clock active. Second, we use 3 different clock domains on the Micro-controller to reduce the average power consumption. Mi-crocontroller DMA module operates at 12MHz to support GPS data rate, as well as for high speed burst writes to flash. Since the microcontroller CPU is only responsible for setting up of data transfers between GPS and flash modules through the internal RAM, and for decoding low-frequency WWVB data, the CPU core uses a ' 2MHz internal low-accuracy clock. We use a low-power 32kHz real-time clock for main-taining system time.

To measure the power consumption of the CLEO plat-form, which runs at 3V , we connect its power supply through two resistors (15W and 100kW) and short circuit the larger resistor at system power up and during certain operating modes; we measure the voltage drop across these resistors to calculate the current. The measured current draw at different operating modes is shown in Table 3. The average current for GPS receiving is significantly higher than the average oper-ating current (27mA) due to the large in-rush current spike to charge the capacitors. There are opportunities to further optimize this factor. Figure 17 shows an active working cycle for sampling and storing 2ms of GPS signal. The process starts from an idle state. It turns on the GPS receiving module for 2ms, and spends 28:8ms to write them into the flash chip. The total energy consumption of the process is 3V [1:5mA 28:8ms+ 42mA 2:2ms] = 0:407mJ. By comparison, an A-GPS on mobile phones takes about 1J for the first location fix. We achieve a gain of more than three orders of magnitude in device-side energy efficiency.

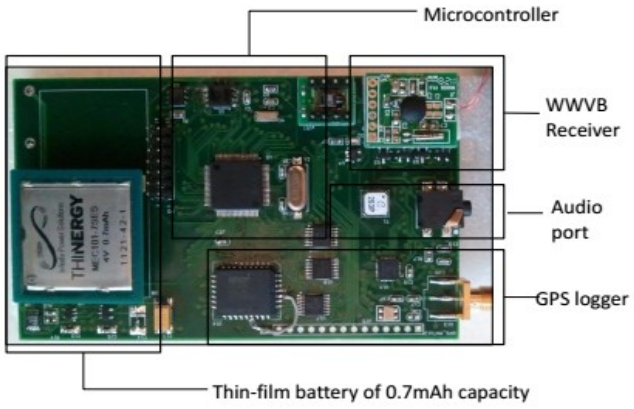


Figure 15: CLEO hardware platform and description.

ADVANTAGES

- Monitors all risks and threats
- Alert message to cell telephone for far flung facts
- Sophisticated safety

APPLICATIONS

- Security, Remote tracking, Transportation and logistics

CONCLUSION

Motivated by the possibility of offloading GPS processing to the cloud, we propose a novel embedded GPS sensing approach called CO-GPS. By using a coarsetime navigation technique and leveraging information that is already available on the web, such as satellite ephemeris, we show that 2ms of raw GPS signals is enough to obtain a location fix. By averaging multiple such short chunks over a short period of time, CO-GPS can on average achieve < 20m location accuracy using 10ms of raw data (40kB). Without the need to do satellite acquisition, tracking and decoding, the GPS receiver can be very simple and aggressively duty cycled. We built an experimental platform using a GPS front end, a serial to parallel conversion circuit, a microcontroller and external storage. On this platform, sensing a GPS location takes more than orders of magnitude less energy than selfcontained GPS modules. The initial success of CO-GPS motivates us to extend the work further. We will exploit various compression techniques, especially those based on compressive sensing principles, to further reduce the storage requirements. We plan to release the hardware reference design and make the LEAP web services available to research communities.

Shadow Locations

The challenge of deriving receiver location with no reference landmark is the possible outliers, which we call shadow locations. Figure 1 illustrates this concern using two satellites. Here, we model the pseudoranges from each SV as a set of waves, each 1 light-m apart. Clearly, these waves intersect at multiple locations. Since we do not know the exact millisecond part of the propagation delay, all intersections, A; B; C; D; ::: are feasible solutions, even though only one of them is the correct location. When more satellites are visible, more constraints are added to the triangulation, which helps resolve the ambiguity. However, a larger number of satellites alone is not enough. To illustrate this empirically, we take a 1ms raw GPS trace and apply CTN with an array of landmarks across the globe. There are 6 satellites in view. The landmarks are generated by dividing the latitude and longitude with a 1o resolution around the globe. In other words, we picked 180 × 360 = 64; 800 landmarks with adjacent distance up to 111km on the equator. Figure 2 shows the total of 166 converged points.

REFERENCES

1. “Embedded and real time projects system” by Dr.kvk.prasad ,Black book
2. “Embedded System” By Raj Kamal fourth edition
3. “8052 Microcontroller And Embedded Systems” By Ali Mazzidi
4. “8051 micro controller” by Ak.ayala second edition
5. “micro processor and interfacing” by A.k.ray fourth edition
6. “ micro controllers and assembly language” by dougles .Vhall forth edition
7. Electronics for you (magazine)
8. Electrikindia (magazine)
9. Electronics bazar (magazine) [1]. IEEE PAPERS

. "Power Supply Design Principles by “Ben Schramm”.
www.pscpower.com
http://www.onsemi.com
http://electrical-engineering-portal.com