When Assistance Becomes Dependence: Characterizing the Costs and Inefficiencies of A-GPS

Narseo Vallina-Rodriguez[a]
Jon Crowcroft
name.surname@cl.cam.ac.uk

Alessandro Finamore[b]

surname@tlc.polito.it

Yan Grunenberger[c]
Konstantina Papagiannaki
yan/dina@tid.es

[a]Computer Lab, University of Cambridge, UK
[b]Politecnico di Torino, Torino, Italy
[c]Telefonica Research, Barcelona, Spain

Location based services are a vital component of the mobile ecosystem. Among all the location technologies used behind the scenes, A-GPS (Assisted-GPS) is considered to be the most accurate. Unlike standalone GPS systems, A-GPS uses network support to speed up position fix. However, it can be a dangerous strategy due to varying cell conditions which may impair performance, sometimes potentially neglecting the expected benefits of the original design. We present the characterization of the accuracy, location acquisition speed, energy cost, and network dependency of the state of the art A-GPS receivers shipped in popular mobile devices. Our analysis is based on active measurements, an exhaustive on-device analysis, and cellular traffic traces processing. The results reveals a number of inefficiencies as a result of the strong dependence on the cellular network to obtain assisting data, implementation, and integration problems.

I. Introduction

One of the engines for the success of smartphones has been the breadth of mobile applications, bootstrapped by a rich set of platform-dependent APIs that enabled innovation and creativity by third party developers. Location APIs have played a leading role among them. For example, beside classic navigation and map services, mobile apps can suggest points of interest, locate friends, provide targeted ads and perform *geo-tagging*, i.e. associate location information to user content such as photos.

Location APIs operate by processing data both from the embedded sensors and external data fetched from the network connection to provide coordinates or human readable information like the street name. They rely on two technologies: Network Positioning System (NPS) and Assisted-GPS (A-GPS). In the first case, the location is retrieved by measuring the signal level of surrounding transmitters (Wi-Fi access point or cellular towers) to compute its location by performing a lookup on a set of remote databases. NPS technologies have the ability to provide short Time To First Fix (TTFF, i.e. time necessary to compute a location) at a limited energy cost but they are not an optimal solution for navigation and other apps requiring high accuracy. On the other hand, GPS is the best technology for accuracy but it can require minutes to fix the position depending on the signal conditions.

Sensor	Accuracy (m)	Energy	TTFF (s)	Indoors Support
Cellular	100-5000	Mid	1-3	✓
Wifi	25-200	Mid	1-3	✓
GPS	1-50	High	5-120	Near windows
A-GPS	1-100	Very High	2-60	Near windows

Table 1: Comparison of location technologies in modern mobile devices.

This makes it not compatible with modern mobile apps where users typically perform just a single location fix for *check-in* (e.g. Foursquare), or geo-tagging services. Instead, A-GPS combines GPS capabilities with assisting data provided through the cellular network to help the receiver to speed up synchronization of the GPS signal and the position fix. Table 1 compares the characteristics of these technologies reporting the accuracy error in fixing the position, the TTFF, the energy consumed and the ability to sense location indoors.

While indoors localization techniques and the performance of regular GPS chipsets have been well studied in the literature, little is known on the true performance of A-GPS and the assisting infrastructure. To the best of our knowledge, this is the first study focused on investigating the whole A-GPS system in modern smartphones. We carefully analyse how A-GPS receivers operate and behave on multiple devices and on the cellular network. We also cover

the assisting infrastructure and how the current design impacts on the whole system. We use a comprehensive approach that relies on *i)* active experiments to identify different A-GPS implementations in modern smartphones, *ii)* a set of on-device analysis to assess the accuracy, speed and energy cost of acquiring a location, and *iii)* a characterization of A-GPS systems using real cellular traffic traces from more than three million users. In more detail, our work reveals a number of interesting and previously unreported findings:

- A limited number of protocols provide assisting data. Although standardized protocols like OMA SUPL specification are available, many of them are chipset-specific. The whole assisting infrastructure is generally hosted in North America. This design adds significant latency for operation in other places of the world for mobile content distribution (Sec. III).

- While accuracy and speed of acquisition are within the range of the expected performance of A-GPS systems, the energy evaluation of A-GPS receivers reveals a positive correlation between energy overhead and the network dependence (Sec. III).

- A-GPS traffic has virtually no impact on the network in terms of signaling and traffic volume. However, it is frequently re-downloaded despite its cacheability. Nevertheless, the control-plane signaling can negatively affect the performance of A-GPS receivers (Sec. V).

Despite the fact that A-GPS was originally designed to operate more efficiently and faster than GPS, this work reveals that the optimization benefits are bounded to the implementation effort on the device itself and the OS integration. The impact of the cellular network and the support infrastructure are quite neglected parameters in current system implementations. We believe that this work could bring some light and research interest to the optimisation of A-GPS technologies, perhaps requiring the addition of new functionalities and features.

II.   From GPS to A-GPS : Principles

GPS is a geolocation system designed to work at worldwide scale. Started in the 1970s, it is based on a fleet of satellites constantly broadcasting data frames at a very low rate (50bps) to receivers on the earth's surface, allowing reception with a very low level of signal. They transmit on the same frequency band (L1

band at 1575 MHz) using CDMA-like spread spectrum techniques and a pseudo random sequence. Each satellite transmits frames composed of *i)* accurate timing information generated by an atomic clock embedded into the satellite itself, *ii)* its precise orbital information used to compute its location which remains valid for up to 4 hours (the *ephemeris*), and *iii)* ionospheric conditions and the operating status of the whole system (the *almanac*).

Before decoding the ephemeris to compute its location, any GPS receiver needs to acquire the satellite's signals and identify the visible satellites through a frequency and code-delay search[1]. Once a satellite is locked, the receiver can decode the data sent in the frames received at a slow pace[2] to compute its location. When this information has been obtained from enough satellites, the receiver can compute its own position using tri-lateration techniques. The whole process can take several minutes. This is known as *cold start* and can take even longer if there are bit errors in frame reception. However, satellite re-acquisition is faster than acquisition. In other words, if the receiver was recently active, the ephemeris, time and last-known position (i.e, a priori position), are still relatively accurate and allow the time required by the frequency/code-delay search to be reduced, enabling a position fix without having to fully decode GPS frames transmitted by the satellites. This is known as *warm start* and it can take about 30 seconds. Even better, it is possible that the last known position and the clock offset are valid so the receiver is able to acquire GPS signals faster. This is known as *hot start*.

GPS receivers were initially designed for periods of continuous navigation (order of hours) with a relatively short fixing time (order of minutes). However, a low TTFF on a mobile device is vital for the user experience. A-GPS takes advantage of the benefits of re-acquisition by providing assistance data in the form of a coarse location estimation, time and orbital information which is fetched from an online server using a reliable link such as cellular networks. In particular, the orbital information is usually provided in the form of an extended ephemeris which is obtained and synthesised by a network of GPS receivers deployed worldwide. Consequently, it has global and longer temporal validity as it uses forecast techniques to pre-

---

[1]Due to satellites' motion and Doppler effect, it is possible to have up to 8.4 kHz of frequency shift (receiver's motion is negligible at pedestrian speed).

[2]Each GPS frame is broadcast every 30 seconds, and they are composed of five sub-frames containing the ephemeris (transmitted every 30 seconds), almanac (every 12.5 min as it is fragmented in several frames) and time (every 6 seconds).
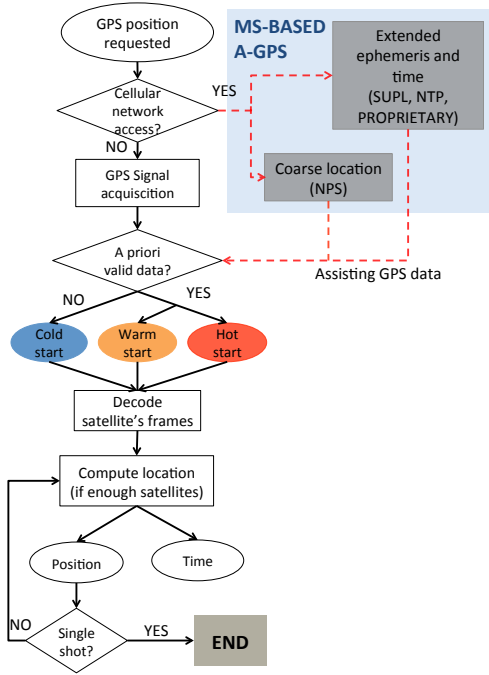
Figure 1: Autonomous GPS and MSB location (user-plane) A-GPS logic. The assisting data (gray boxes and dot lines) is downloaded as a parallel process to the normal GPS flow (white boxes).

dict the satellite's orbit. As opposed to standalone GPS receivers, A-GPS improves receivers' sensitivity, allowing them to work sometimes even indoors as reported by Kjærgaard et al. [2010]. However, there are variations from one chipset to another. Depending on whether the location computation is offloaded or computed locally, A-GPS technologies can be classified as:

- *Mobile Station Based (MSB) - Location computation is local.* The A-GPS receiver (i.e. *Mobile Station*) obtains extended ephemeris, almanac, time and coarse location from a remote server. The mobile device acquires GPS signals and computes its position.

- *Mobile Station Assisted (MSA) - Location computation is offloaded.* The server provides the mobile device with the expected code delays and Doppler values. The mobile device acquires raw satellite's signal and sends them to a remote server which computes the location and returns the results to the mobile device.

A-GPS technologies can be also distinguished with respect to the channels used to provide assistance to the mobile stations. In the *control plane*, assistance data is based on 3GPP control standards such as GSM-RRLP.
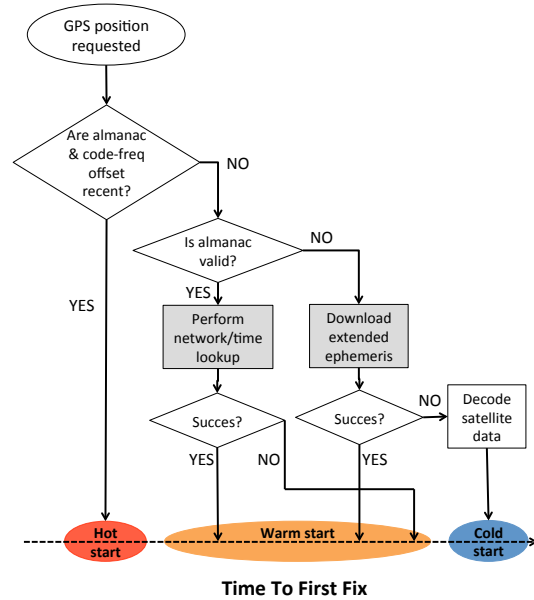

Figure 2: A-GPS logic according to packet capture.

In the *user plane*, the assistance data is encapsulated over the IP network. A remarkable example is OMA SUPL protocol. Hybrid solutions are also possible, but most modern operating systems configure the A-GPS receivers as user-plane and MSB. Their basic theoretical logic is presented in Fig. 1. The download of assistance data runs in parallel with the mainstream GPS acquisition process.

III. A-GPS technologies

Most of the smartphone platforms provide analogous features in terms of location services to the developer and the end user. However, the final experience and their openness may vary depending on the level of abstraction offered by the operating system and chipset manufacturer. In order to get a bigger picture, we completed a series of active experiments on a set of Android and iPhone devices that can accurately represent the market as of first quarter 2012 in the UK [3]. The characteristics of these devices are detailed in Table 2.

In order to collect and monitor the download of assistance data over the cellular interface, we run tcpdump in the background while these operations are being performed. Despite important OS and hardware operational differences, a common generic behaviour has been extracted from their traffic pattern as described in Fig. 2. It helped us to realise that

---

[3]We use geo-tagged Flickr images, a popular photo service, as an arbitrary index of popularity of these phoneshttp://www.flickr.com/cameras/

Phone	Google Nexus One	Google Nexus S	iPhone 3G	Samsung Galaxy S2	iPhone 4S
GPS Chipset	Qualcomm QSD8250	Broadcom 4751	Broadcom 4750	SiRF star IV	Qualcomm MDM6610
Type	MSB	MSB	MSB	MSB, MSA or pure GPS	MSB
Traffic classes	gpsOne	Bcom-LS + SUPL	LTO	SUPL	gpsOne

Table 2: Mobile devices used for active experiments.

Class	Connection	Hostname	IPs	Object	Whois
LTO (Broadcom)	HTTP	iphone-wu.apple.com	17.254.32.16	lto2.dat	Apple
gpsOne (Qualcomm)	HTTP	xtra1.gpsonextra.net xtra2.gpsonextra.net xtra3.gpsonextra.net	216.187.118.44 both 69.90.74.197	xtra.bin	Peer 1 Hosting
gpsOne (iPhone4, Qualcomm)	HTTPS	iphone-ld.apple.com	variable	xtra2.bin	Akamai
SUPL (Google)	TCP 7275/7276 (SSL)	supl.google.com	74.125.x.192	-	Google
Bcom-LS (Broadcom)	raw socket port7275	bcmls2.glpals.com bcmls.glpals.com	64.210.203.195 216.34.140.195	-	Broadcom

Table 3: Characteristics of A-GPS traffic classes.

control-plane latency found in cellular networks can impair performance, and to identify the following traffic classes:

gpsOne: Qualcomm proprietary technology. It fetches an extended ephemeris (*xtra.bin* file) through HTTP. The service is provided by three hostnames `xtraN.gpsonextra.net`, where (N = 1,2,3) corresponding to two IPs in the Peer 1 Hosting network. A load balancing scheme is used to handle the requests at DNS level (`xtra2.gpsonextra.net` can assume both IPs while the other two are statically mapped). This technology can be found in some Android phones as well as in the latest iPhone versions (from 4S). In the latter case, assistance is provided by Apple servers[4] rather than Qualcomm ones.

Long Term Orbit (LTO): Broadcom proprietary technology (similar to gpsOne) used in earlier iPhone devices (3G, 3GS and 4). A system daemon (*locationd*) performs a HTTP connection to *iphone-wu.apple.com* in order to fetch the *lto2.dat* file[5]. This data corresponds to chipset specific and satellite data which is updated a few times a day (i.e, the extended ephemeris). Despite the quasi-static nature of this type of content which makes it compatible to be served through CDN's, the support infrastructure is fully centralized: *iphone-wu.apple.com* resolves into a single Apple server located in the US.

TCP/7275, TCP/7276 (SSL): SUPL and customized vendor protocols. Google SUPL *supl.google.com* service is present in every single Android device. The service can use SSL (port 7276). Although other mi-

nor services provide SUPL support on port 7275, We found that the majority of the traffic on this port is related to *bcmls.glpals.com* and *bcmls2.glpals.com*. These hostnames are two Broadcom specific servers for proprietary assistance using an identified TCP protocol, that we refer to as Bcom-LS (Broadcom Location Services).

Table 3 summarizes the characteristics of the four classes of traffic identified for these popular devices. For most of the technologies, a few servers, mainly located in North-America are providing A-GPS assistance for *the entire world*. A cross-check from other vantage points in different countries (France, Italy, Spain, UK, US) corroborates that observation. Furthermore, DNS records are not properly adapted for a large-scale deployment. For instance, DNS records for gpsOne present a TTL of one day, mapped to only three IPs, all hosted in North America under the same autonomous system. Similarly, Google SUPL servers resolve to a single physical IP, despite having a CNAME record with a short TTL. Finally, Bcom-LS on port 7275 also presents a very short TTL (1 min) that maps to a single IP in the US (without round robin) serving the whole world. This makes the whole assisting infrastructure highly vulnerable in case of failure, but it is changing with newer releases. Newer Apple devices (from iOS 4) rely on Akamai services to increase the resilience and reduce latency. We further observed two more types of traffic potentially related to A-GPS. First, we noticed that NTP requests are performed in certain chipsets each time the A-GPS receiver is used but it can be attributed to other operating system services to correct the time. Secondly, iPhone devices use SSL to perform NPS operations.

---

[4] `https://iphone-ld.apple.com/xtra/xtra2.bin`

[5] `http://iphone-wu.apple.com/7day/v2/latest/lto2.dat`

Summary:

A-GPS technologies shipped on modern smartphones differ from one device to another. Despite the presence of a standard protocol such as OMA SUPL, the A-GPS ecosystem is clearly using a small set of proprietary protocols. Furthermore, the assistance infrastructure for the entire-world is generally localised in North America, making A-GPS vulnerable to failures. In Sec. IV, we characterize the performance and energy consumption of two popular A-GPS chipsets trying to find inefficiencies in the way these technologies are integrated on mobile systems. We choose Android devices due to their openness.

IV. Characterizing A-GPS performance

Whenever an Android application requests an A-GPS fix, the Java Location API is invoked (*LocationManager*). The Location Manager is responsible for accessing the low C/C++ layer (Android HAL layer) via the *GpsLocationProvider*. That provides access to the low level GPS functions from the chipset. The open source nature of Android allows tracing the behaviour of the Location Manager and other classes involved in downloading the assistance data (e.g. *GpsXtraDownloader* and *SntpClient* for Qualcomm chipsets).

Handsets such as the Nexus S integrate the A-GPS module in a completely different way. We have identified a binary daemon (*gpsd*) running in the background that acts as a middleware layer as shown in Fig. 3 (the gray elements represent the basic components expected on any Android device). Such a design modifies the standard work-flow by introducing a software bypass between the A-GPS logic and the cellular interface (blue arrow on the figure). Furthermore, a dump of the symbols of the *gpsd* uncovered additional features such as NTP calls, SSL and SUPL functions for NPS, and calls to other sensors such as the magnetometer and accelerometer. This allows the A-GPS daemon to directly access the network functions *without* consulting the Android subsystem. This causes an additional energy overhead as the OS may perform such actions in parallel.

In this section, we conduct a series of experiments designed to measure three metrics we consider as representative of the A-GPS performance. We select exclusively terminals endorsed by Google such as the Nexus One and Nexus S. According to the specifications, these two handsets are running respectively a Qualcomm and a Broadcom A-GPS chipset, hence it covers two of the main manufacturers identified in the
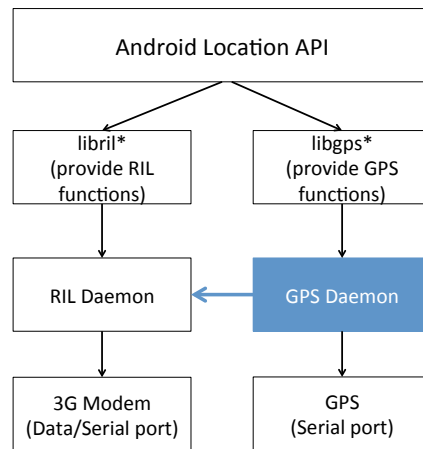

Figure 3: Software architecture of A-GPS on the Nexus S.

previous section. The experiment is designed to see some differences in their behaviour and performance at the following levels:

1. *Time to first fix or TTFF (seconds)*: time elapsed between the instant in which the location engine receives a location request until the request is fulfilled. This is designed to address the usability of A-GPS from a user perspective.

2. *Accuracy (metres)*: difference between the true location and that returned by A-GPS. It measures the quality of the position fix.

3. *Current draw (mA)*: amount of current expended in acquiring a GPS location. It quantifies the energy cost of using A-GPS technology.

IV.A. Methodology

As already reported by Kjærgaard et al. [2010], the performance of A-GPS depends on both handset design and on its context. For that reason, we performed the experiments in three different environments (all of them with clear sky conditions) as described in Table 5. The scenarios were selected to cover realistic situations in which A-GPS performance may be compromised.

For statistical significance and given the sudden changes that can occur on the radio channels, TTFF and accuracy experiments are repeated 25 times. We placed the devices 10 cm apart from each other on a surface for practical reasons, in the same position and facing the same direction while they were sensing location simultaneously. The *"ground truth"* is estimated as the average fix between two Nexus One fixing A-GPS locations for a long time and filtering out

Device	Location	TTFF Cold start (s)		TTFF Warm start (s)	
		Median	98 Percentile	Median	98 Percentile
Nexus S	Open sky	10.1	28.0	9.1	18.1
	U. canyon	34.8	143.9	13.1	50.5
	U. park	6.9	14.0	7.1	13.1
Nexus One	Open sky	4.1	28.1	4.0	6.1
	U. canyon	10.1	86.1	4.1	11.1
	U. park	3.1	10.1	3.1	8.1

Table 4: Summary of the TTFF per device in three different environments.

Location	Sats
Open sky. Metal columns used for street lightning were present. Concrete ground	11
Urban canyon. Narrow stairs between concrete walls to limit viewed sky. Concrete ground	7
Urban park. Park with relatively dense tree canopy. Earth ground	12

Table 5: Description of the experiment locations and maximum number of satellites locked

Figure 4: Violin plot and box plot of the TTFF in three different locations for a Nexus S and a Nexus One

outliers. This methodology has been widely used in previous experiments such as the study by Kjærgaard et al. [2010]. As Doppler effect at pedestrian speed is negligible, whether the devices are static or moving does not affect the results.

Estimating power consumption is harder to quantify than accuracy and TTFF. We measure the current drawn by the handset using a Monsoon's PowerMonitor. Any unnecessary hardware module is off. The energy evaluation was done on a controlled environment with clear sky view. To synchronize the traces obtained from the mobile device with the power monitor ones, we generate a current peak pulse by turning on and off energy-intense resources (e.g, camera flash) as to identify when the experiment begins, using USB disconnection events to trigger the data collection.

IV.B. TTFF

Ideally a cold start in A-GPS receivers should look like an autonomous GPS warm start. The violin plot[6] shown in Fig. 4 corroborates that statement as the TTFF is usually below 30 seconds for both handsets instead of minutes as for typical autonomous GPS receivers as reported by Wing et al. [2005]. However, despite having network access to obtain assistance data, contextual factors still play an important role on the performance of A-GPS receivers. As a consequence, the TTFF becomes more unpredictable if environmental and network conditions are

not favourable as in urban canyons, where the TTFF in a cold start presents a heavy tailed distribution requiring up to 100 seconds to perform a fix in this environment.

Table 4 summarizes the results obtained for the devices under study in the three environments analyzed. As expected, being on a warm start helps to reduce the TTFF. The 98th percentile decreases notably for both devices independently of the environment. Overall, the Nexus S is likely to have a higher TTFF if compared to the Nexus One. In fact, while the experiments have been carried with all devices lying on a surface, we noticed that the TTFF for a Nexus S is improved when the device is lifted up or held on a hand as the antenna is located on the back. Consequently, its reception is negatively affected when the device is acquiring GPS signals[7]. Impact of the body position and gesture on some A-GPS receivers has been already studied by Blunck et al. [2011].

IV.C. Accuracy

Fig. 5 shows the density of location fixes on a 2D surface obtained during the experiment done to evaluate the TTFF. As we can see, the accuracy during a warm start (blue points) is generally better than in a cold start (red points). The reason is that the receiver is

---

[6]A violin plot is a combination of a box plot and a kernel density plot that also shows the probability density.

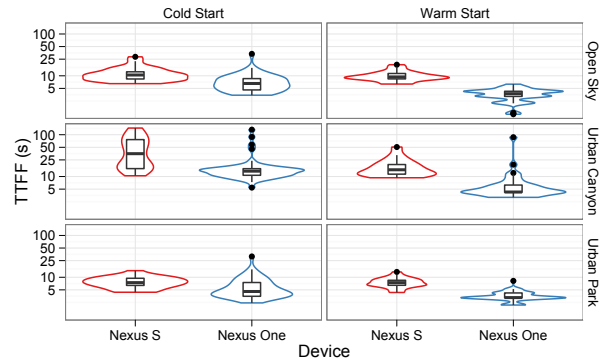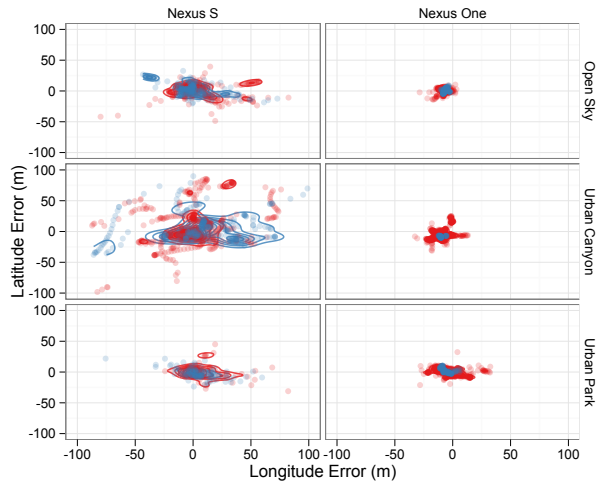[7]http://www.ifixit.com/Teardown/ Nexus-S-Teardown/4365/2

Figure 5: Locations fixes and 2D density estimates in open sky, urban canyon and urban park environments. Red points represent the location fixes on a cold start whilst blue points represent warm start ones.
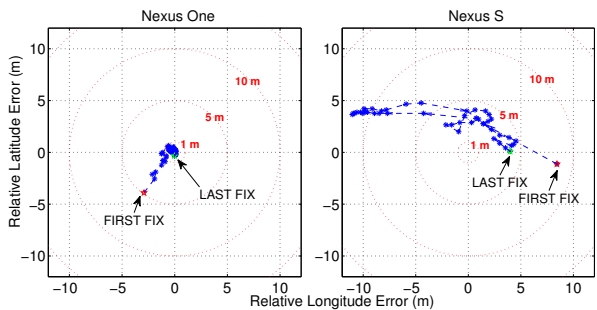

Figure 6: Trajectory shown by the successive fixes during 15 minutes for a typical location. The devices were static.

able to acquire more GPS signals, having more freedom about which satellites to use in order to compute its location. This decision is typically based on signal strength and the expected position of the satellites as studied by Hadaller [2008].

Consequently, most of the location fixes obtained on a warm start are within the region of interest that contains 95% of the fixes for any platform and environment, as the receiver can select the signals from the most adequate satellites. Fig. 6 shows the consecutive fixes reported by a *static* device for 15 minutes. In presence of fluctuations on the time of flight and the clock offset caused by a poor reception or even reflections on the radio signals, consecutive location fixes can jump from one place to another causing trajectories of several tens of meters that can be observed across all the scenarios since the first location fix.
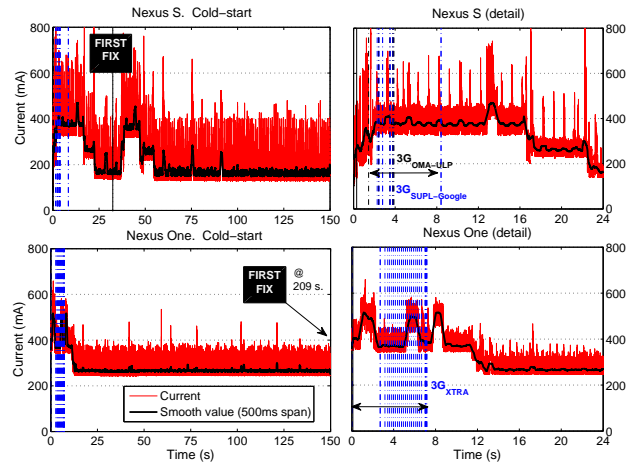

Figure 7: Power consumption during cold start for the Nexus S (top) and the Nexus One (bottom). Left plot details 150 seconds while on the right only the first 24 seconds The dashed lines indicate network traffic while the solid black line marks the first location fix.

IV.D. The energy cost of the network dependency

The studies by Zhang et al. [2010], and by Pathak et al. [2012] report A-GPS as one of the most energy-intensive resources on a mobile system. As opposed to simplified measurements based on power models obtained from linear regression techniques such as PowerTutor, we can obtain detailed information about the costs of performing different tasks such as turning on the A-GPS receiver, downloading assistance data over the cellular network, and fixing location.

Assistance data is vital to reduce the TTFF and improve the sensitivity of the receiver. However, using the cellular interface to obtain these data imposes an energy-overhead on the mobile client. The works by Qian et al. [2010], and Balasubramanian et al. [2009] are just two examples among the many studies that analyzed the characteristics of the current cellular stack and the tied relationship with the battery life in mobile devices. Depending on the volume and the frequency of the traffic, and according to the inactivity timers of the RNC states, the mobile device will exhibit different power levels on its radio interface which define its power consumption, latency, and throughput.

Fig. 7 shows the current consumed (y axis) by the A-GPS modules shipped on a Nexus S (top row) and a Nexus One (bottom) for a cold start, beginning when the application requests a location fix to the *Location-Manager*. The figures on the left show the current draw during 150 seconds whilst the initial 24 seconds are detailed on the right side. The vertical lines cor-
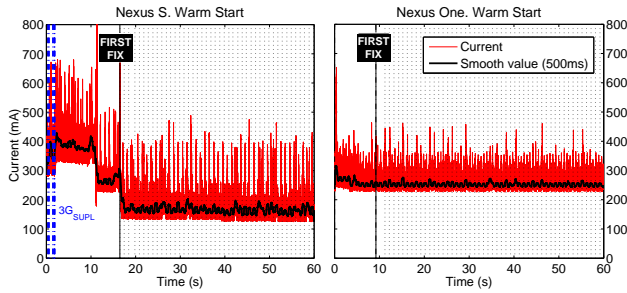
Figure 8: Power consumption during a warm start for a Nexus S (left) and a Nexus One (right) during a minute. The dotted black lines represent new location fixes and the blue dashed ones acquisition of assistance data. We can see that the Nexus One does not require acquiring assistance data on a warm start.

respond to requests for assistance data, actions that happen when the A-GPS receiver is turned on for at least 15 seconds. In these power measurements it is possible to clearly identify the energy overhead attributed to downloading assistance data: 401.9 mA and 338.6 mA on average for the Nexus One and the Nexus S respectively, including CPU costs and RRC inactivity timeouts. As we can see, relying on cellular networks adds a significant energy overhead on the mobile client, which is positively correlated with the network conditions and the RNC inactivity timeouts defined by the network operator. Once the cellular interface is set IDLE (or in paging mode), the current draw for both chipsets decreases to 266.7 and 167.9 mA approximately respectively.

Fig. 8 shows the current draw on a warm start. In this case, the A-GPS module should not depend that much on assistance data (a valid copy should be already available on the device) but the Nexus S still imposes an energy overhead caused by the network activity. Despite the lower current draw for the Nexus S when the cellular interface is IDLE, its apparent dependency on cellular traffic implies higher energy consumption with respect to Nexus One. This is clearly shown in the violin plot represented in Fig. 9 showing the distribution of the energy (in mAh) required to fix a location per device, and GPS state. Considering the network traces collected for the phones listed in Table 2, we can map this to the fact that the Nexus S combines two A-GPS traffic classes: the Broadcom specific on port 7275, followed by SUPL traffic to Google. The combination of these technologies tends to reduce the power efficiency of this device, especially for single shot location requests.
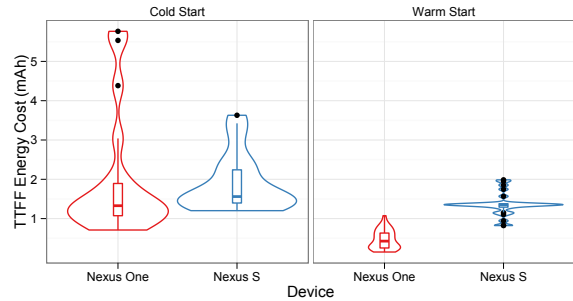

Figure 9: Energy consumed until the TTFF (i.e. single-shot fix) for a Nexus One and a Nexus S in a cold (left) and a warm start (right).

Summary:

A-GPS does not necessarily improve accuracy but it reduces notably GPS's TTFF at the expense of a higher current draw attributed to fetching assistance data over the cellular link. Our analysis revealed performance differences across mobile handsets, mainly due to integration issues that can double the energy consumption until the first fix. The case of the Nexus S illustrates the difficulties for hardware manufacturer to implement innovative schemes that were not initially planned in the original framework (in this case, Android). Based on Sec. III findings, we define a rule-set that will be used in Sec. V against a large data set of cellular traffic traces. This analysis will allow us to better understand the way millions of mobile handsets obtain assistance in the wild over a day.

V. One day of A-GPS traffic

In this section we present a characterization of A-GPS traffic observed in a large European mobile operator. We refer to a data set of mobile traffic collected on August 13th, 2011 at 8 vantage points which share the load of the entire country. It contains 1.7 billion connections corresponding to 22 TB of volume downloaded by more than 3 millions of mobile subscribers. The monitoring activity at each vantage point is reported in a set of text log files. Each entry in the logs contains a set of standard information, such as IPs, ports, number of bytes downloaded, and some other HTTP specific information, such as content type, HTTP user agent, and HTTP response code. Given that NTP can be generated by other background services and applications, it is not included in the network study as it can add uncontrolled noise. Likewise, the SSL connections found in iOS are also filtered out as they seem to be related with NPS technologies.

Each user is identified with an anonymized unique ID and each line of the logs corresponds to a different

(a) Comparing O.S.

Device	Fl[k]	%	Vol[MB]	%	Usr[k]	%
iPhone	792	0.075	29108	0.198	309	14.33
Android	260	0.141	2583	0.116	78	14.87
Total	1051	0.085	31692	0.187	387	14.44

(b) Breakdown Android A-GPS

A-GPS class	Fl[k]	(%)	Vol[MB]	(%)	Usr[k]	(%)
gpsOne	24	9.04	801	31.02	16	21.30
SUPL	124	47.79	93	3.60	52	66.93
Bcom-LS	112	43.17	1689	65.39	15	20.28
Total	260	100	2583	100	83	108.51

Table 6: A-GPS data set.

TCP connection performed by an user. We identified 11 different types of mobile operating systems in the data set but Android and iPhone devices account for 96% of both volume and flows. iPhone is the most popular platform type accounting for 70.7% of the subscribers as opposed to 17.3% of Android devices, generating 83.5% and 12.5% of the traffic volume respectively. As a consequence, for the remaining part of this section we focus solely on iPhone and Android devices. Unfortunately, as the data set accounts exclusively for one day of traffic, an extended and long-term characterization of this traffic is out of the scope of this work. However, we aim to provide an overview of the main properties of A-GPS traffic, highlighting macroscopic effects also affecting the user experience. We characterize each class of assisting traffic, investigating the volumes generated and the time elapsed between subsequent requests done by each device.

V.A. Volumes

Table 6 reports the amount of A-GPS traffic extracted from the cellular log using the set of rules obtained in Sec. IV. Table 6(a) compares the amount of A-GPS traffic with respect to the device OS considering number of flows, bytes, and mobile subscribers (percentages are related to the amount of A-GPS traffic for each OS).

As iPhone adopted only LTO at the time of the data set (iPhone 4S and iPhone5 came later on the market), Table 6(b) breaks down the A-GPS traffic per assisting protocol for Android. Google SUPL is the most common one, being responsible for the higher number of flows but the smallest number of bytes. We observed that 8.51% of devices using SUPL where using also one of the other two classes. We link this observation to the fact that SUPL configuration is directly provided in the Android Open Source Project and its up to the A-GPS implementation to use it or not.

Fig. 10(a) reports the CDF of the total number of flows generated by each user during the whole day for the four classes of traffic. While only 2.87% of the users have more than 10 flows in the day for LTO, gpsOne and SUPL, 3.47% of the users generate more than 30 flows in the day towards the two Bcom-LS servers. Fig. 10(b) reports the CDF of the total number of bytes downloaded in the day by each user. SUPL volume is 10 times smaller than the other three classes. LTO and gpsOne traffic distributions are very similar as their requests are composed of messages of nearly 40 KB thus forcing a RNC promotion to a dedicated channel. LTO traffic tail is heavier due to the higher number of flows requested as seen in Fig. 10(a), whereas Bcom-LS presents a bimodal distribution (requests of 40KB and 1KB). Given their similarities, it is reasonable to presume that the content delivered is similar while the protocol is different.

Based on the number of users requesting assisting data, we can state that Broadcom dominates the market as they are shipped in pre-iPhone 4 devices as shown in Table 6(a). However, different technologies coexists in the Android ecosystem. An analysis based on the `UserAgent` field of the HTTP requests reveals that Qualcomm is the leading vendor in Android with 62.27% of the devices. Fig. 13 reports a tree map of both vendors and type device model popularity based on the number of Android devices generating A-GPS traffic during the day.

V.B. Inter request time

A-GPS data consists of different type of content, from satellite info to cell IDs and time. This data, especially satellite info (e.g, LTO and gpsOne), is highly cacheable, and has a temporal validity of several days as described by Frank Van Diggelen [2009]. However, between 14.65% and 54.47% of devices downloaded such data more than once a day, causing unnecessary energy and signaling costs.

Fig. 10(c) reports the inter request time of consecutive requests performed by each device for the four classes of A-GPS traffic. Unexpectedly, results show the presence of re-downloads within one minute. In particular, 70% of consecutive requests on iPhone devices are interleaved by less than 30 seconds. This is in part related to the presence of some outliers which obtain the content hundreds of times in the day. However, results do not change when the same measurement is performed considering only devices with a maximum of five downloads/day. A deeper analysis of the traffic reveals that these users were legitimately using A-GPS for location-based applications, such as Google Maps. On the other hand, the gpsOne dis-

(a) CDF of total number of flows per user.

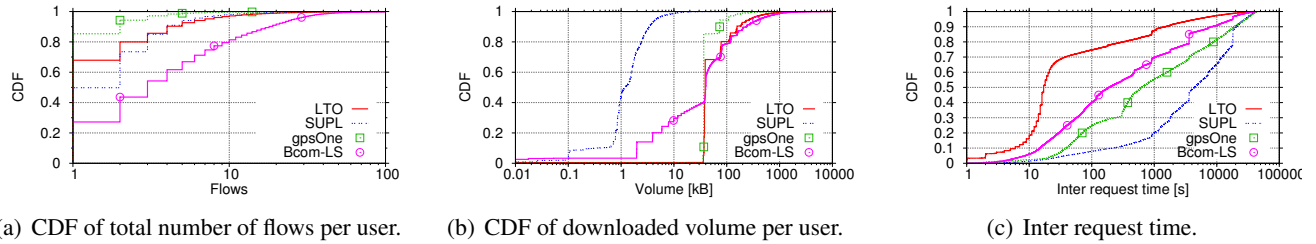(b) CDF of downloaded volume per user.

(c) Inter request time.

Figure 10: A-GPS traffic analysis for devices with more than a flow per day.
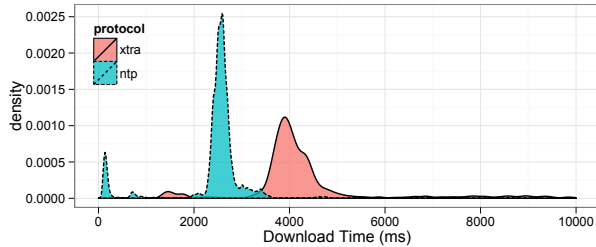
Figure 11: Distribution of the download time for the gpsOne *xtra.bin* and NTP RTT over 3G.

Figure 12: Impact of network latency on the accuracy of 1000 NTP lookups. Android's SNTP client error increases linearly with network latency.

tribution presents a sharp knee around 300 seconds, possibly corresponding to a timeout. For SUPL and Bcom-LS servers, the curves are smoother with two timeouts at one and five hours. 8% and 40% of the consecutive requests happen in less than 100 sec for SUPL and Bcom-LS respectively, thus forcing a larger number of network promotions.

V.C. The impact of cellular networks latency

The control-plane latency required to allocate a radio channel in cellular networks implies that the time required to reach the servers is non-negligible. We run experiments on one of the phones to quantify its impact. First, we measured the RTT of performing 1000 NTP lookup on the national NTP pool and downloading the extended ephemeris for gpsOne on a 3G interface every 30 seconds. The download time histogram for both types of assisting data is shown in Fig. 11. The same effect can be seen in the case of LTO. As feared, the impact of powering up a radio to download a resource from a remote IP (possibly located in another country) has a non negligible download time over 3G which varies from 2 to 4 seconds. The energy expenditure also increases with higher RTTs.

Furthermore, we evaluated the impact of the usage of NTP over the 3G network with variable RTT (including the control-plane latency). Fig. 12 shows the time error (y axis) as a function of the RTT (x axis). As we can see, the NTP error increases linearly with network latency. In that case, the benefits for faster
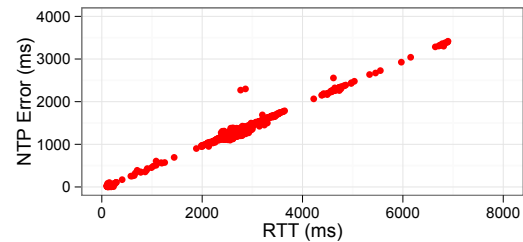
satellite acquisition and location fixing achieved by time synchronization are likely to be affected by the potential errors caused by the network latency and the download time.

Summary:

The small amount of data downloaded and the number of flows observed for assisting GPS do not impact on the network resources of the hosting operator, as well as the data plan of the user. However, the strong dependency on the network may sometimes impair performance due to network latency (mainly control-plane latency) while increasing the energy cost of location operations. The analysis revealed chipset integration inefficiencies such as lack of caching. Although the *extended ephemeris* has a nominal validity of two weeks, multiple re-downloads happen within a one minute time-frame.

VI. Re-thinking A-GPS

In the previous sections, we have seen the limitations of A-GPS mainly due to the dependency on cellular networks and the strong layering on the OS design. With the knowledge obtained from the previous sections, we will discuss different research areas and problems that can improve the way assistance is performed both on the client and the support architecture.

Figure 13: Tree map of the vendor popularity for Android devices. Numbers in brackets indicate the share of users identified for each Android device on the cellular network dataset.

On the client:

Location drivers can use networking resources and sensors autonomously without any coordination with the OS. This behavior is unexpected, especially on embedded system, where a good integration of the different modules is key to achieve a good performance. A more strict integration with the OS can provide important energy savings by avoiding redundant and unnecessary access to such power-hungry resources.

The network usage when A-GPS is operative has a non negligible energy expenditure, especially in the use of single-shot applications with *check-in* and *geo-tagging* capabilities. Although, most of the assistance data is highly cacheable, neither the OS nor the location APIs incorporate caching capabilities. As an example, the extended ephemeris could be corrected on the handsets by porting some of the logic used to generate the files to the end-points whenever new satellite frames are received.

Even time can be cached. A centralised time server can use periods of connectivity (e.g, by piggybacking RNC promotions caused by other applications) to perform NTP lookups. It can help to correct the clock-drift, as well as the latency error shown in Fig. 12, serving immediately time to applications. The OS can exploit its central role to monitor the staleness of the local data and pre-fetch it whenever satisfactory conditions occur. This has two immediate benefits: *i)* it allows reducing the usage of the cellular network even for a *cold-start* with A-GPS operations, and *ii)* it allows the A-GPS receiver to have valid data as soon as possible, reducing the negative impact of the high latency (including the control-plane one) of cellular networks when fetching assistance data as shown Fig. 11. However, identifying when and how data must be pre-fetched is key to optimise the whole system as it can increase the volume of mobile traffic and energy costs.

On the assisting infrastructure:

The supporting infrastructure - in terms of redundancy of DNS records, TTL of these records and servers redundancy, and geographical distribution - is not properly adapted for the requirements of a world-wide A-GPS deployment with millions of mobile devices depending from it. We have shown that only a couple of servers were used for the entire A-GPS network support on the analyzed country. Exploiting CDNs for assistance can provide important benefits, as well as incorporating assistance on network components such as Wi-Fi APs or even neighbouring devices.

VII.   Related Work

Paek et al. [2010] looked at the performance of standalone GPS receivers in urban and rural scenarios. Hadaller [2008] analysed the impact of receivers' mobility, whereas Ramos et al. [2011] studied how to improve the acquisition phase at the signal processing phase. Little research effort was focused on a holystic analysis of A-GPS technologies. Nevertheless, Kjærgaard et al. [2010] analyzed A-GPS performance indoors, whereas Blunck et al. [2011] characterized the impact of antenna design and user gesture on the accuracy and performance of the receiver. The later article also proposed a pure GPS signal classification method to discriminate the environment at the cost of keeping the A-GPS receiver active.

Nevertheless, there is a broad literature about energy efficiency for location sensing and mobile smartphones as surveyed by Vallina-Rodriguez and Crowcroft [2012]. In particular, Paek et al. [2011], Zhuang et al. [2010], Lin et al. [2010], and Kjaergaard et al. [2009] studied how to combine different location sensors such as A-GPS, NPS, compass and accelerometer to reduce the energy consumption in navigation modes. They used forecasting, substitution, suppression, piggybacking and adaptation techniques for outdoors localisation. Liu et al. [2012] proposed an *MSA* technique based on offloading some calculations to the cloud. With few milliseconds of raw satellite's data, it can estimate the device's past locations by exploiting information from public, online databases. Unfortunately, it is still dependent on network access.

VIII.   Conclusion

In this paper, we presented the first full characterization of A-GPS technologies along five different axis: *i)* technology and protocols used (Section III), *ii)* TTFF, accuracy, and energy consumption (Section IV), and *iii)* network dependency (Section V). For that, we used active experiments on modern A-GPS modules as well as passive experiments run on cellular network traces obtained from a major European carrier. The results reveal that A-GPS technology exhibits a severe variability in its performance and energy consumption despite offering reasonable performance in terms of TTFF and accuracy. The main reasons behind are related to the way the chipsets use cellular networks to obtain assistance and how they are integrated in the OS. We notice that the patterns for fetching data do not match with the supposed temporal validity (and cacheability) of the assistance data, adding unnecessary transmissions over the energy-costly cellular interface. We discuss several possibilities for improvement at different levels in order to guarantee a better performance (in terms of TTFF), energy consumption (reduce network dependency), integration (better coupling between the mobile network capabilities and the A-GPS modules), and reliability (adding caching and pre-fetching capabilities for assistance data on the handsets, and increase the security of the assisting infrastructure).

References

N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *ACM IMC*, 2009.

M. Blunck, M. B. Kjærgaard, and T. Skjødeberg Toftegaard. Sensing and classifying impairments of GPS reception on mobile devices. In *Pervasive*, 2011.

Frank Van Diggelen. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

D. Hadaller. Mitigating GPS Error in Mobile Environments. *Technical Report. University of Waterloo*, CS-2008-13, July 2008.

M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer. EnTracked: energy-efficient robust position tracking for mobile devices. In *ACM Mobisys*, 2009.

M. B. Kjærgaard, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, and K. Grønbæk. Indoor Positioning Using GPS Revisited. In *Pervasive Computing*, 2010.

K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy trade-off for continuous mobile device location. In *ACM MobiSys*, 2010.

J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. F. Loureiro, and Q. Wang. Energy efficient GPS sensing with cloud offloading. In *ACM SenSys*, 2012.

OMA SUPL specification. OMA SUPL. http://www.openmobilealliance.org/Technical/release_program/supl_v2_0.aspx.

J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *ACM MobiSys*, 2010.

J. Paek, K. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using Cell-ID sequence matching. In *ACM MobiSys*, 2011.

A. Pathak, C. Hu, and M. Zhang. Where is the energy spent inside my app? Fine Grained Energy Accounting on Smartphones with Eprof . In *ACM EuroSys*, 2012.

F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3G networks. In *ACM IMC*, 2010.

H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal. LEAP: a low energy assisted GPS for trajectory-based services. In *ACM UbiComp*, 2011.

N. Vallina-Rodriguez and J. Crowcroft. Energy Management Techniques in Modern Mobile Handsets. *Communications Surveys Tutorials, IEEE*, 2012.

M. G. Wing, E. Aaron, and L. D. Kellogg. Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability. In *Journal of Forestry*, June 2005.

L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *CODES/ISSS*, 2010.

Z. Zhuang, K. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *ACM Mobisys*, 2010.