

Smart location-awareness and -triggers on mobile devices with geofence technology

Fredrik Englund

24-3-2015

Royal institute of Technology (KTH) Kista, Sweden.

School of information and communication (ICT).
Examiner and supervisor: Johan Montelius.

Acknowledgements

I would like to thank my supervisor, Johan Montelius, for the valuable advice and support he has given me in execution and writing of this report.

I would also like to thank Nathaniel Taylor for valuable advice and support with my measurement set-ups and for lending and analyzing the results from the oscilloscope.

Finally I would like to thank The Mobile Life for giving me this thesis, resources and expertise in Android to guide me in this thesis work.

Abstract

Today we have many mobile applications that use location services. With the increasing rate of applications per phone and the fact that each phone has a limited amount of power, it becomes more and more important that we see the battery as the limited resource it truly is and that we should start to think about how we are spending it. Therefore this thesis will investigate how much power the location services actually need to work properly. In addition to the power consumption this thesis will also investigate how accurate each location service is so that we can make a statement about what we get for the amount of battery consumed.

Sammanfattning

Idag så finns det många mobilapplikationer som använder sig av lokaliseringstjänster. Idag när kvoten av applikationer per mobil ökar stadigt och det faktum att mobiler endast har en begränsad batterikapasitet, så blir det mer och mer viktigt att vi verkligen ser batteriet som en begränsad resurs som vi ska vara sparsamma med. Därför kommer detta examensarbete att handla om hur mycket batterikapasitet som lokaliseringstjänster verkligen behöver använda för att fungera på ett bra sätt. Detta examensarbete kommer också att undersöka vilken precision varje service har så att vi kan bestämma på ett ungefärligt sätt vad vi får för vilken batterikonsumtion.

Contents

Title Page	i
Acknowledgements	ii
Abstract	iii
Sammanfattning	iv
Table of contents	v
List of figures	viii
1 Introduction	1
1.1 Problem definition	1
1.1.1 Questions	1
1.1.2 Main problem	2
1.2 Method	2
1.2.1 Accuracy	2
1.2.2 Battery	2
1.2.3 Tests	3
1.2.4 Tools	3
1.3 Delimitation	3
1.3.1 Assumptions and platform	3
1.3.2 Android specific delimitation	4
1.3.3 Server interaction delimitation	4
1.4 Results	4
1.4.1 Goals	4
1.4.2 Conclusion	5
2 Related work concerning battery consumption and accuracy in Android	6
2.1 Pre-work	6
2.2 Close related work	6
3 Background	7
3.1 Android Geofence basics	7
3.1.1 Geofence	7
3.1.2 Geopush	7
3.1.3 Geotigger	8
3.1.4 Geofence transitions	8

3.2	Android basics	8
3.2.1	Google Cloud Messaging service (GCM)	8
3.2.2	Service	9
3.2.3	Intent Service	9
3.2.4	Async Task	9
3.2.5	Process	9
3.2.6	Threads	9
3.2.7	Push Message	9
3.2.8	Google Cloud Messaging service	10
3.3	Positioning Services	10
3.3.1	WPS	10
3.3.2	GPS	10
3.3.3	Cellular towers	10
3.3.4	Fused provider	11
4	World and demands	12
4.1	Method delimitation	12
4.2	Accuracy	12
4.3	Battery	13
4.4	Assumptions	14
5	Implementation	16
5.1	Software	16
5.2	Accuracy	16
5.3	Battery	16
5.4	Main application	17
5.5	Test application	18
6	Tests	19
6.1	Accuracy test	19
6.1.1	Step size	19
6.1.2	Accuracy test simulation	19
6.1.3	Fused accuracy test	20
6.1.4	GPS accuracy test	22
6.1.5	WiFi accuracy test	23
6.2	Battery test	25
6.2.1	Comparison	26
6.2.2	Fused Battery test	27
6.2.3	GPS Battery test	29
6.2.4	WiFi Battery test	30

7 Evaluation	33
7.1 Accuracy	33
7.1.1 Geofence Border	33
7.1.2 Solution one	33
7.1.3 Solution two	33
7.1.4 Geofence size	34
7.1.5 Comparison	34
7.2 Battery	35
7.2.1 WiFi	35
7.2.2 GPS	36
7.2.3 Fused	36
7.2.4 Comparison	37
8 Conclusion	40
8.1 Accuracy	40
8.1.1 Geofence border	40
8.1.2 Location provider	40
8.2 Battery	41
8.2.1 Update frequency	41
8.2.2 User friendly	42
8.2.3 My choice	42
9 Future work	44
10 References	45

List of Figures

1	Geofence	7
2	Steps	19
3	Simulations	20
4	Fused accuracy test	21
5	fig: GPS accuracy test	22
6	WiFi accuracy test	24
7	Battery setup	25
8	Oscilloscope	25
9	GPS dry run in mA	26
10	WiFi and Fused dry run in mA	27
11	Fused low amount of WiFi area, current in mA	28
12	Fused medium amount of WiFi area, current in mA	28
13	Fused high amount of WiFi area, current in mA	29
14	GPS low amount of WiFi area, current in mA	29
15	GPS medium amount of WiFi area, current in mA	30
16	GPS high amount of WiFi area, current in mA	30
17	Fused low amount of WiFi area, current in mA	31
18	Fused medium amount of WiFi area, current in mA	31
19	Fused high amount of WiFi area, current in mA	32

1 Introduction

Even during the early days of mobility, mobile operators could send us somewhat geolocation-aware messages via SMS, by triangulating cell tower positions. Then came smartphones with GPS built in, and data plans so that everyone had Internet connections, as well as the ability for application developers to send push notifications to applications installed on a device, even if the application was closed or running in background.

Since then marketers and others have tried to figure out how to best make use of this location-awareness for smartphone users, and mobile operating systems (OS) have been enhanced to introduce new ways of triggering actions to users based on the current location.

One early solution was to update back-end (database on a server) with locations of specific devices at certain intervals, then use those saved locations as a selection criteria when sending push notifications to end users. This always had a lag between where the user is at this very moment and where the server thinks the user is. That is, pseudo real time at best. Therefore push notifications had to be sent to a geographically large enough area so that the user had not already left it when receiving the push. And saving geolocations more frequently would just drain battery life and in addition it has privacy implications.

With the relatively new geofence technology this has changed. Via back-end the application can be served with one or many geographical areas, “geofences”, that should trigger certain actions. Most often this action is a pop-up message, which to the end user is perceived as a normal push notification. But since the user doesn’t need to send its location to a server this becomes more of a monitoring service rather than a push service. This could make the user take back its privacy when looking at location awareness. But there are still other questions that needs to be asked. How much battery does these services consume? What accuracy can you get for that battery consumption? How far can we rely on a geofence transition?

1.1 Problem definition

1.1.1 Questions

I want to investigate this new way of doing geopush. My investigation should be able to answer these questions.

- Are there geofence standards that can be used across platforms?

- If I can't use geofences as standards, what is the best data format to use?
- Pros and cons of geofence compared to other solutions?
- Implications for end users and privacy policies?
- What is the battery consumption for each geofence provider?
- Which accuracy can I achieve for each geofence provider?

1.1.2 Main problem

The main problem statement for this thesis is, based on geofence coordinates, how I can get geographically location-aware push notifications/messages to end users, without lag and in as real time as possible, without having the application opened? That is, enabling location-awareness in the background, that can trigger relevant geopush messages, while optimizing battery usage and being as sensitive as possible to the privacy of end users.

The main optimization is that of the battery but I can't optimize the life time so much that I lose the functionality of the program. In most cases when I am discussing geofences that means that I can't optimize life time so much that I loose to much accuracy of my position. This problem creates a new question how much accuracy am I willing to lose? The answer of the last question has to be depend on how much battery consumption I earn from that loss of accuracy.

1.2 Method

1.2.1 Accuracy

The only reason why I am interested in the accuracy is to be able to make ruff comparisons. I will print enter and exit transitions on a map as circles and use the accuracy as the radius of the printed circle. I will use a black circle to mark and enter transitions and a white circle for exit transitions. Both the enter and exit transition are semi transparent and will therefore create a grey area if and where they overlap.

1.2.2 Battery

I will measure the battery consumption by plugging in a multimeter between my phone and its battery. After that I will connect the multimeter to a laptop that will continuously store the values given by the multimeter. Note

that the program that stores the current values will calculate a mean every second and supply that as the value. I will then create a graph for each test to display the battery consumption. During the tests I will only have the most necessary settings turned on. For example when using the WiFi provider I will have the WiFi and the mobile data turned on since the phone will have to ask Googles servers where the WiFi that I can currently see are, but when I switch to the GPS only provider then I will turn off both WiFi and mobile data.

1.2.3 Tests

There will be three unique tests conducted in three different areas. The different tests are the three different location providers given in Android. The providers are the WiFi provider, the GPS provider and the Fused provider which used both GPS and WiFi. The three different areas are defined by the density of WiFi in them. Low WiFi area (no WiFi), medium WiFi area which is a normal house area and finally the High WiFi area which is an area of multistorey houses.

I will setup geofences which I will walk through before I do any test in an area so that each test in the same area will use the same geofences.

1.2.4 Tools

I will run all the tests on a large smartphone Samsung Galaxy Note III. I will use a laptop and a multimeter. Except for some cables those are all the hardware that I will need. I will need two different software programs for this study. The first program is an Android application that asks for the current locations and that will check if I am entering or exiting geofences. It should also be possible to print this information in a Google map so that I can compare the accuracy. Secondly I need a sampling program for the multimeter.

1.3 Delimitation

1.3.1 Assumptions and platform

For the purpose of this thesis, whatever longitude and latitude I get from the applications SDK, I will assume is correct. And for the practical implementation of the prototype application, only the Android platform is considered, although geofence also exists in iOS. However, the back-end API implementation serving the geofence coordinates is generic and should be able to be used from any platform.

1.3.2 Android specific delimitation

All services and tasks are running in the same process. The only advantage I would get if I were to run them in separate processes is that it would make it hard for bad code to control the whole program. The draw backs of having all services in the same process are that before switching between services I would need to do a full context switch. And since I want as real time as possible I have discarded the thought of running threads in different processes.

Since I want to minimize battery consumption I want the background process to be as idle as possible while still performing at maximal performance. For this reason I have chosen to use the Intent Service class as the main background service since as long as it doesn't receive any intents it will remain idle. The geofence will be registered with the OS and geotriggers will be sent as intents to the intent service.

I also use a service that the intent service uses as a Async task. The reason why it isn't a Async task is that when I have started the application I will need it to function as a service. So as long as there is no application started by a user the service will be used as a Async task. But as soon as the program is started it will function as a real service. Note that for it to function as a service it needs to be bound to an application.

1.3.3 Server interaction delimitation

Since I want this program in as real time as possible all logic, if I am in a geofence or not, should be done on the phone and not on a server. This means that the server interaction will be limited to register clients for GCM pushes, fetching of geofences and the actual GCM pushes.

1.4 Results

1.4.1 Goals

- An API with a test application that uses the real application.
- A background application that triggers and registers geofences from local storage and can communicate with a server when first retrieving the geofences.
- Documentation of the geofence protocol used (indicating if the protocol can be used across platforms).
- Battery consumption tests which includes accuracy.

- Documentation of privacy concerns.

1.4.2 Conclusion

Both the WiFi only and the Fused provider does really well in my tests so the choice between them depends on the environment, how fast the information needs to be provided and how large margins you have. The Fused provider uses the GPS as a backup and will give more constant information and usually more precise information but will be a lot more battery consuming compared to the WiFi only provider. The GPS only provider is less reliable than the Fused and uses about as much power consumption. Note that if I have enough WiFi around us the WiFi only provider gives a really good accuracy and the WiFi only does really well in the battery consumption tests.

2 Related work concerning battery consumption and accuracy in Android

2.1 Pre-work

There has been a lot of work done in the field of location awareness for mobile applications. Even as early as 1998[7] there are some thesis papers that investigates the possibility of using cell towers and GPS systems to triangulate our current position. Today most thesis papers are investigating how to correctly find a clients position indoors where the cell towers and GPS systems can't reach the clients [1][3]. Most of these thesis papers are focusing purely on security[5][6] or accuracy.

2.2 Close related work

The paper *Location Based Pre-caching and Network Coding in Smart Content Distribution* [4] is the first one that I have found that has tried to investigate the battery consumption. They investigated how much battery it took while having a few geofences and a pre-cashing service. The pre-cashing service consists of two zones, one small zone (*information zone*) where you utilize information and a larger zone (*pre-cashing zone*), that encapsulate the small zone, where you download the information you will utilize in the *information zone*. The main difference between my work and theirs is that I will only do the battery consumption tests on geofences and no other application logic. My goal will be to minimize the battery consumption compared to the goal of investigating if pre-cashing is possible. Their work in battery consumption is more a test if it is an acceptable battery consumption instead of battery consumption optimization. I will also focus on the different ways of achieving a position and their power consumption. This means that I will have more accurate measurements and I will focus on getting positions which just enough application logic for it to be useful. I am probably going to have different results since I will have more accurate data and I do not include downloading of large information files.

3 Background

There are a few concepts that someone not working in the field might need more knowledge about before reading this report. Therefore this chapter will give short explanations about those concepts. I will presume that the reader knows these concepts in the chapters following the Background chapter.

3.1 Android Geofence basics

3.1.1 Geofence

Geofence is defined in the form of a area that uses longitude, latitude and some form of area description. A geofence is not only defined as a circle but in many cases are only available as a circle. I will use only circles from now on so when I write gefoence I will mean a circular geofence. Another important aspect of the geofence is that it must be a representation of an area in the real world. We can see four examples of geofences in figure 1. The difference in color means only that the grey ones are from the server and the green ones I have created in the application.

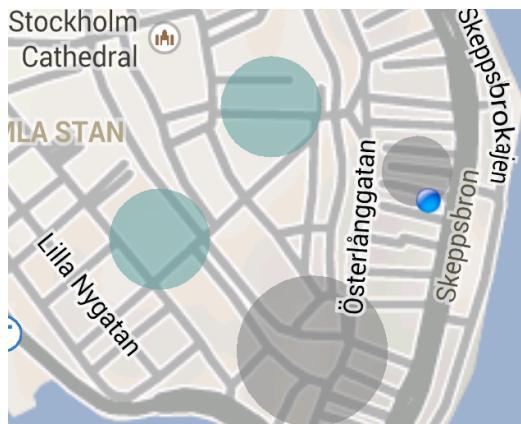


Figure 1: Geofence

3.1.2 Geopush

Geopush is short for geofence push. A geopush notification is a push message that contains information that is dependent on a geofence location. This means that a geofence location transition must trigger the geofence push. Geopushes can come directly from a server, as it was done in the past. Technically a geopush can be simulated by the OS without the need to get it from a server but when reading online many will assume that if you use geopush

that the message comes directly from a server. But since I am trying to optimize battery and realtime that push notification should come from the OS itself. I will instead send the instructions of what to do through push notifications and then let the OS to all the geofence calculation and sending the geofence information to the application.

3.1.3 Geotrigger

Geotrigger is short for geofence trigger and is almost the same as a geopush notification but refers to the state change rather than the message to the user which means a geotrigger triggers a geopush notification. Important to know for this thesis is that a geotrigger in my application will be triggered locally on the phone and not on a server.

3.1.4 Geofence transitions

There are three different transitions that can occur.

- Enter transition; this means that most of the area that our given position and its accuracy spans are inside the geofence, and last time I checked I was outside the geofence. In other words I have entered a geofence.
- Exit transition; this means that most of the area that our given position and its accuracy spans are outside the geofence, and last time I checked I was inside the geofence. In other words I have exited a geofence.
- Dwell transition; this means that most of the area that our given position and its accuracy spans are inside the geofence, and last time I checked I was inside the geofence. In other words I am dwelling inside a geofence.

3.2 Android basics

3.2.1 Google Cloud Messaging service (GCM)

By connecting my application to the GCM server we can get a GCM ID which we can relay to my server. My server can then use the ID to send push messages to my device and thereby minimizing the polling my device needs to do. The GCM messages doesn't provide first in first out semantics but when sending geofences we do not need to consider this.

3.2.2 Service

An Android Service is a background program (no GUI). Unless the service is bound to an application and the service doesn't terminate itself the service will run for as long as the OS allows(theoretically it could be forever since the Android OS will only kill a service if it is in need of some resource the service uses).

3.2.3 Intent Service

An Android service that runs in the background(no GUI). The intent service will be waiting for intents to receive before executing. Once they are done with the last intent it will go back to "sleeping" mode. This means that the service will only run after received intents. Intents can be sent from a application, another service or if we register broadcast-receivers from the OS.

3.2.4 Async Task

A Async task like the services will not run on the main thread. The main difference is that an Async Task will do a method once and then terminate. They can be considered as a short term service.

3.2.5 Process

A process is a collection of shared data and a "pool" of threads. This means that different processes doesn't share any data while threads share some data (threads in the same process).

3.2.6 Threads

A thread is the smallest unit of processing that can be scheduled by an operating system. Each program isn't restricted to one thread, usually a program has multiple threads. Different threads share the same process resources if they are in the same process. One example of process resources are data and open files. Since a thread includes a program counter the thread essentially decides what to execute next.

3.2.7 Push Message

A push message is a message that a server sends to a client. It is called push since the server forces the message on the client, the client itself never has to check the server if there are any new messages.

3.2.8 Google Cloud Messaging service

GCM or Google Cloud Messaging service is a service maintained by Google. It receives messages and delivers them to devices through push messages. To be able to receive a message from a GCM server you first need to register your device at the GCM servers. The GCM servers will respond with your unique ID. Then you can give that ID to other servers or services that will send different messages to the GCM servers with your ID. The Servers will then lookup your ID and use it together with the rest of the data you sent when you registered your device to send a push message, containing the message the server/service sent to the GCM server, to you.

3.3 Positioning Services

3.3.1 WPS

WPS (WiFi based Positioning System). The WiFi is considered as a known location (either by connecting and asking the router or by looking into a database of known WiFi). The distance from the router is based on the signal strength from the WiFi. By combining the above two methods we can get a ruff position from any WiFi. We can also use fingerprinting to increase accuracy. For further reference read about WiFi locations in [1] [2] [3].

3.3.2 GPS

GPS (Global positioning system) uses satellites to produce a location. The accuracy depends on how many satellites we can find. We need at least four satellites to be able to get GPS coordinates. This is usually considered the most accurate of the positioning systems but also the one that consumes most battery. Before being able to use the GPS systems we need to locate the GPS satellites, this can take some time and power. Since we need to locate satellites we can't be indoors while using GPS. Indoor positioning is almost always impossible while using the GPS systems since they need a clear line of sight to the device.

3.3.3 Cellular towers

Cellular positioning uses base stations to triangulate a position. We need at least three different base stations to get a location and the location gained is usually a ruff position. Cellular triangulation is usually considered the worst way to calculate a position since it gives a bad accuracy.

3.3.4 Fused provider

Fused provider uses all the above positioning technologies to give a fast, high accuracy and with as low battery consumption as possible. It may use combinations of the above technologies or just one depending on the circumstances. For example if we can find a good enough location using only WiFi then it will not use the GPS but if we only get an approximate location from the WiFi it may use a few satellites to increase the accuracy. Exactly how it provides the location is controlled by the Android OS and how it chooses what resources to use is, as far as I know a Google secret.

4 World and demands

4.1 Method delimitation

The built in battery statistics in the Android system only measures active applications which is useless for this thesis since the only active part of the developed application is the result displayed on the Google map, the real work that the application does is done in the background. The battery statistics would also be unhelpful since it is inaccurate (measures percentage of battery) and they battery statistics is measured and calculated on the same device it is measuring and calculating on, which means that each measurement and calculation will add to the total power consumed.

I'd prefer to measure from the battery directly compared to measure from the USB cable since this will eliminate the possibility of a second current.

Now before I actually decide to use a multimeter to measure the current, I need to make sure that the current flows out in even pulses that aren't too fast for my multimeter. This will be done by connecting my cellphone to a oscilloscope.

4.2 Accuracy

I want the OS to do as much as possible since it will be able to use other applications data as well. For example if I am currently using Google Maps my program should not need to update any locations at all since Google Maps already supply the OS with my current location. Therefore I need to check with the OS is if I am really in a geofence or not. I will use a fake GPS provider to simulate a walking route on the map, the fake GPS provider has a high accuracy. By using the fake GPS provider I will gain one important aspect when checking how sure the OS is that I are actually in a geofence. When using the fake GPS I can move my position straight towards a geofence with small steps. I will also be able to set the accuracy to the same value all the time. This means that as soon as I get a geofence enter transition it will be the first possible enter transition, and as soon as I get an exit transition it will be the first possible exit transition.

Since this thesis focuses on battery consumption am I only interested in the accuracy so far that I can make a comparison and see ruffly what we gain or lose in accuracy. If we can't see the difference with our eyes only then the difference is not worth knowing for this thesis.

4.3 Battery

The Background IntentService service will be responsible for providing accuracy when entering or exiting a geofence. When I am not entering or exiting a geofence I will assume that the accuracy is not good enough or that the location aren't close enough for a transition when testing the battery consumption.

My main idea for testing the battery consumption is to connect a multimeter in between the battery and the phone to measure the current going out of the battery. During a visit to the Samsung support they have told me that this method is not possible since the battery always needs to be plugged in to make the phone start. In more detail the battery has 4 "pins" that needs to be connected to the phone at all time. Two of them are normal positive and negative where the last two are a separate positive and negative for different components that needs different current or voltage. These components are for example the antenna of the cellphone. If there are different voltages or currents and what other components are connected to these last two pins was a company secret or for some other reason not allowed to be shared with me.

I was however given another possible solution which was to cut open a USB cable and connect a multimeter to the positive cable of the USB cable. They connect this USB cable to a power source and my phone, thereby setting the phone in a charging state. To get the charging state I was also supposed to add a resistance in between the fourth and fifth pin of the USB cable. Once the phone is in the charging state the phone should try to draw all current from the USB cable before using the battery and there by make me able to measure the current through the USB cable.

I would have preferred to use the first method rather than the second method since in the second method if I need more current than the USB cable currently can give me then it will take some from the battery. This makes the last method possibly flawed. In addition to that we must remember that the tests will be mobile tests. With that I mean that I will walk around outdoors during my tests which would probably make my power source my laptop and a laptop doesn't give such a high current unless we force the voltage to become five volt with one of the speed charger adapters that are on the market.

The third option in measuring the current from the device is to use a power meter application. This should be considered the worst option since we are measuring a device output with the same device. This means that the power measure tool will affect how much power it is measuring just by taking the measurements. We could make a standard test run where only

the power measurement tool is running and then subtract this value from the real measured value. But that is not optimal either since the measurement of only the measurement tool will probably vary through time and thereby we will have to create a mean value to use as a subtraction which will in the end give us a incorrect result.

There are other programs as well which doesn't measure the actual power consumption but instead measures the time that an application uses different components. This also includes start up times for each component. Once all the data has been gathered the application will lookup the standard power consumption for each component and thereby together with time and start up time calculate the power used by the application. The big problem with this method is that we need to rely on standard values given from the manufacturers of the cellphones to give us correct values. If a manufacturer should ever manipulate any data it should be these values since they are complicated to measure and a small change or mistake may lead to better statistics once they sell the device. Another problem with this method is that it uses standard values which in them selves will not be as accurate as the other measurement methods.

The best solution is the first one which is the one I used in the end. I connect a few cables to the four pins on the phone and then connect the other end of the cables to the current pin in the cellphone and thereby give me a cable to measure on.

4.4 Assumptions

When testing battery consumption I will assume the worst case scenario. If there are enough other applications asking for the locations then this program will get their location updates as well which means that it doesn't need to look for the GPS satellites itself. The worst case scenario in this scenario is that it is only my application that wants location updates.

I will also assume that the user wants as little communication with the server as possible since this increases the privacy and it will also lower the demand on the WiFi/mobile data services. The mobile data and WiFi can take a lot of battery so keeping as much as possible in memory will lower the battery consumption.

The worst case scenario here would be if at the start of the test the background services would needs to do an update to get all the geofences from the server. This isn't realistic for most programs since they should be pre bundled with geofences or at the first start update these geofences. In the case of testing the power consumption of different location settings the download part of the geofences aren't as interesting. So from now on when I

say the worst case scenario will be that the application is the only application running on the device and therefore will need to locate GPS satellites as soon as it is started.

Even on a new Android device there are some background processes that runs in the background like push and standard mail services, we also have update services (for the OS) and many more. Most of them you can terminate by going into application manager and look under the running tab choose one and click on the disable button. I will also assume that the background processes that I can't terminate before a run like the Android OS, use up a constant current.

When I walk the routes that I have specified there will be three iterations on each route. I will assume that I walk the exact same path but not with the same speed. The speed will change the time it takes for me to walk through each route but I will not measure the total current drawn but instead the mean, the maximums and minimums of each iteration.

5 Implementation

5.1 Software

A test program and background services was developed for the Android OS with a play 2 back-end server. This test program is used to test that the program works, not to gather statistics except that it will show us the position and accuracy of a geofence transition.

5.2 Accuracy

All my current geofences are displayed using a Google map. Once I enter a geofence I will draw a circle with the radius with the same length as our accuracy, the center of the circle will be the guessed transition coordinates. I will also use a black circle for entering transitions and a white circle of exiting transitions. This gives us an image that I can easily compare different tests with.

Before I test the battery consumption I will do an accuracy test which will involve the test application. I will simulate locations like it was an ideal run. The simulated locations will be close to each other and will always move towards the center of a geofence. In real life this is usually not even possible, even if I where aiming for a geofence, since there usually are man made and natural obstacles in the way. But this way will let me analyze the geofence behavior with minimal influence of location positioning and step size.

Since I don't actually write out the exact numbers this can be argued that it is imprecise, but I'm not interested in the actual precision of each geofence transition I'm just interested if it is an acceptable accuracy or an unacceptable accuracy.

When doing the real tests I will use the same accuracy testing but then I will not simulate the locations but instead get them from one of the location providers.

5.3 Battery

Then I will connect my multimeter in between the battery and it's slot in the cellphone, measuring on the positive cable. There will be four cables. Two normal positive and negative and two special for components that needs dedicated current (like the antenna). I will check if the two dedicated supply a constant current while using positioning services and then measure from the first positive cable.

I will connect the cellphone to a computer and measure the battery consumption using a few developer programs. I will then compare an idle phone with a phone that has my services running on it in different modes and thereby get how much my services drains the battery.

There are three different scenarios I will consider.

- The first scenario will be like a large city with a high probability of multiple WiFi in each skyscraper.
- The second scenario will be equal to a residential area where it would probably be a maximum of one WiFi in each house.
- The last scenario will be an area completely without WiFi.

In each one of these areas I will have to walk the same area with the same geofences but with different components activated.

- The first will be with only GPS activated.
- The second will be with only WiFi activated.
- The last will be with both WiFi and GPS activated, this is to test the Fused provider.

5.4 Main application

The main application is actually an Intent Service which can call a service (used as a AsyncTask unless a program has been bound to it) for registration of location updates. If the service is bound to the application it will automatically send the geofences to the application. The Intent Service itself will receive enter and exit geofence transitions as well as boot intents and timer intents. When receiving a timer intent it will tell the service to update. When receiving a boot intent the background service starts. If I receive enter or exit transition intents the intent service will send notifications to the notification bar.

The service that the intent service calls fetches updates from a server when it is called by the Intent service. It also registers the client at the GoogleCloudMessaging (GCM) service and informs my play 2 back-end of its GCM ID that we just gain from registering at GCM service. All registration of which geofences the application should monitor is done in this service.

5.5 Test application

I also got a test application that uses Google maps where we can see my geofences and add new local ones. This application must extend the GeofenceApplicationInterface. I've made it like this so that in future work we can use any desired application to bind to the service. The Google map also gets updated with position and accuracy as soon as we receive an enter or exit transition.

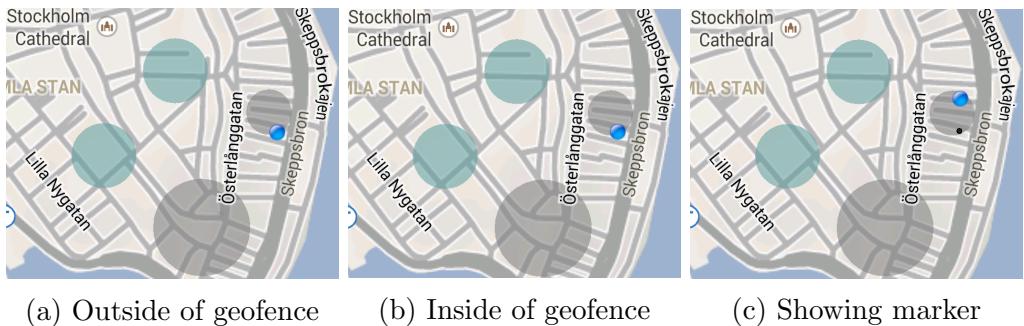
6 Tests

6.1 Accuracy test

6.1.1 Step size

By the word *step size* I mean the length between two location updates.

Since I use a fake GPS provider to do the walking for me, in the first Accuracy test simulation, it is hard to estimate the *step size*, since the simulator don't show the latitude or longitude but instead shows us where we are on a map. Therefore I will first show three pictures that will show the *step size*. Figure 2a is outside the geofence. Figure 2b is inside the geofence and figure 2c shows the marker made by my test program (the radius of the black dot is the accuracy we get from the fake GPS provider). The blue dot is our current position, it's a dot and not an arrow since the program isn't sure in which direction we are moving. Considering that the black dot has a radius of 1 meter then the steps taken each time should be in the range of 1-3 meters.



(a) Outside of geofence

(b) Inside of geofence

(c) Showing marker

Figure 2: Steps

6.1.2 Accuracy test simulation

For the real simulations I will lower the accuracy so that we can see the circles and how they overlap the geofences better. Depending on how large steps you take (could be compared to update frequency) we will be more or less close to the border of the geofences. If we could overlap on both sides of the geofence border, that would be of great interest for this study. In simulation two I tried to use as small steps as possible. Black dots represents an enter event and white dots represents an exit event.



Figure 3: Simulations

We can see in the simulation figure 3a and figure 3b that both the entry and exit events seems to be able to overlap both sides of the geofence border. A ruff estimate is that at least 80% of the accuracy circle must be on the right side of the geofence border.

6.1.3 Fused accuracy test

In this section I will walk with both GPS and WiFi enabled through three different areas. The first area should have loads of WiFi and multistorey houses. This should encourage the WiFi part to take a greater role. The second area should be something like a residential area with a sparse amount of WiFi and single-storey houses and the last should be like a graveyard or a park with no WiFi at all.

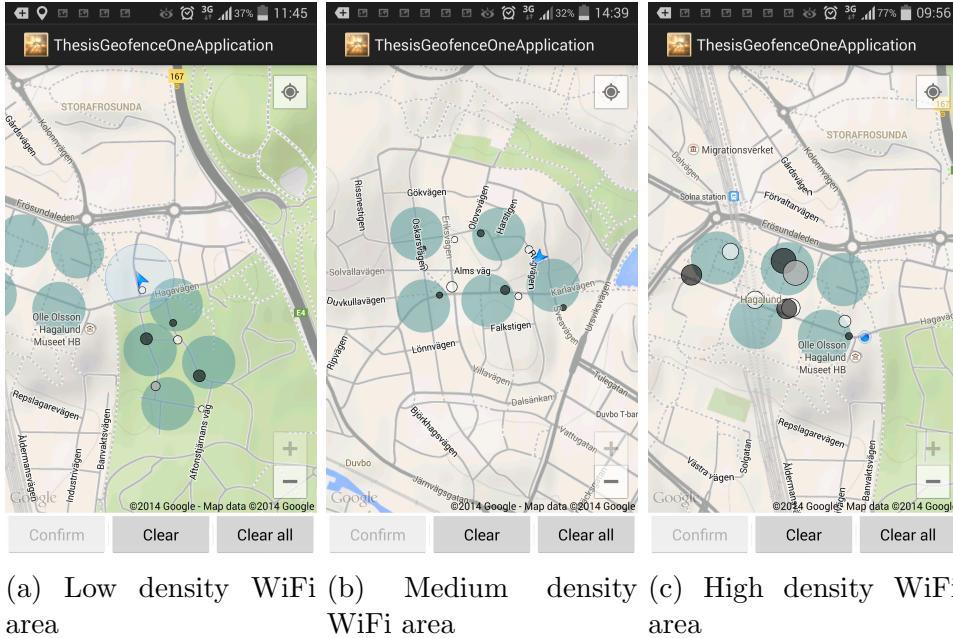


Figure 4: Fused accuracy test

We can see some really interesting observations in these images. If we start by looking at figure 4a we can see that even if we get geofences sometimes it takes a long time to get an exit transition. In the second geofence from the top (in 4a) we see an enter transition (black circle) isn't directly at the border and it's exit transition is in another geofence (the one at the bottom). We should also notice that the exit transition is on top of the enter transition for that geofence making the circle grey instead of black or white.

If we move on to figure 4b we can see that the transitions are much closer to the borders.

Now lets look at the last figure 4c. This one is really interesting since it doesn't behave as one would presume. We see that the top right geofence doesn't have an enter transition and therefore not an exit transition either. We can also see that the geofence transitions have a lot worse accuracy. If we look at the lower left geofence and the upper left geofence (in figure 4a) we see that the enter transitions aren't really even in the geofences. When I walked around in the high density I saw that my position jumped sometimes, usually giving me a change in accuracy. I think that the OS has gotten two different positions and one is inside (one from WiFi and one from the GPS) but the OS hasn't had time to report both locations to my program. I should also include that with multistorey houses all around me the GPS satellites

may be obscured and therefore making the Fused provider rely more on the WiFi.

6.1.4 GPS accuracy test

In this section I will walk with only GPS enabled. I will go through the same three areas as in the Fused accuracy test.

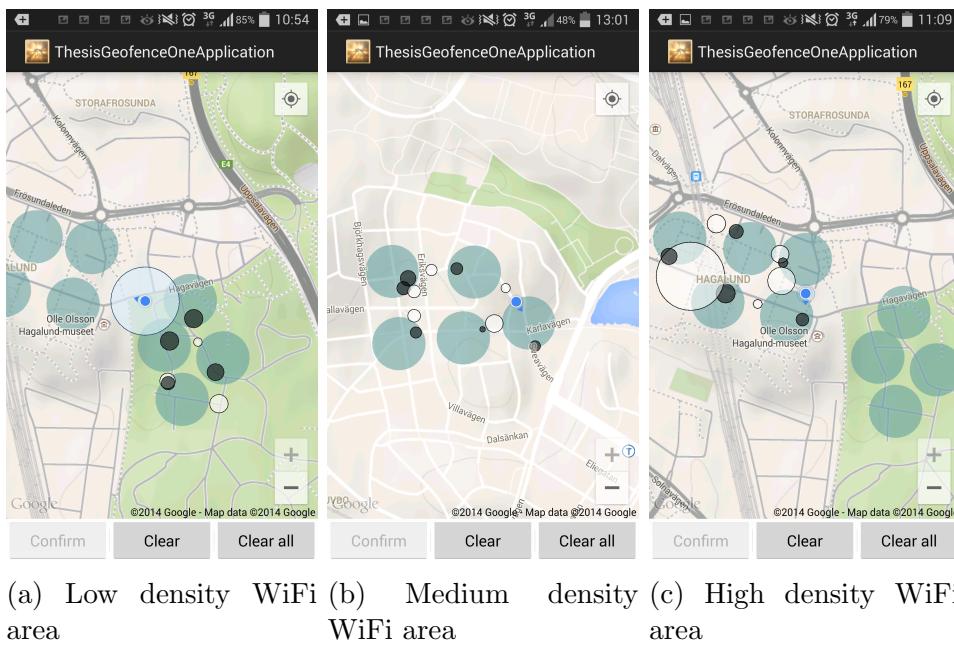


Figure 5: fig: GPS accuracy test

First lets look at the figure 5a. We can see that we got a transition on all the geofences and that almost all is about the same size except for two of them. One is extremely large and one is really small. You can see the large one in the top left corner and the small one is a white area between the two rightmost geofences. The small one just means that we have located many satellites and the large one means that we have found really few satellites. We should also mention how close most of the transitions are to the borders. Except for one enter transition in the most left of the geofences almost all are touching the border.

If we look at figure 5b we can see that all is about the same size but we have one interesting phenomena in the top left corner. There are two enters transition (black dots) and one exit transition (white dot). Up there the GPS worked strangely. Instead of following the path I walked (up and then right in the picture) it was drawn to the right directly when I entered

the geofence. It went right so long that I got an exit transition and a few seconds after that snapped/jumped back to my correct position where I got another enter transition. We should note that all of the transitions are right on the border of the geofences except for two exit transitions.

Lets take a look at figure 5c. The size of the geofences vary the most here not just from this provider but from all the other providers as well. I suspect that the multistorey houses interferes with the GPS somehow. Just as we did in the figure 5a I got a really large exit transition on the left side. In that area I lost the GPS tracking completely for a few seconds during the run. We can also see that even though most geofences are close to the border we can find three transitions that is not touching the border. Neither in figure 5b or in figure 5a could we find so many late transitions.

Overall the GPS triangulates our position quite good but what we do not notice in the test runs is how jumpy the accuracy felt when I was doing the runs. It can easily jump for 100 meter radius down to 10 meter radius in seconds. Most of us are used to the GPS systems in cars and know how accurate they are but we must remember that cellphone GPS's have a great disadvantage which is that a car can't or should not run off the roads which means that a car GPS can triangulate and then move the position to the closest road and most of the time that will just make the guess more accurate. The cellphone can't do that since we do not always follow the roads when we walk. Almost all the time no matter how large or small the accuracy was, with the exception of 5c, I was almost always in the center of the guessed area.

6.1.5 WiFi accuracy test

In this section I will walk with only WiFi enabled. I will walk through all the previous described areas. One might think that it isn't interesting to see how well the WiFi only location provider deals with no WiFi around and in case off accuracy it isn't but in the case of battery consumption it is interesting to see how much the provider consumes when there is nothing to do.

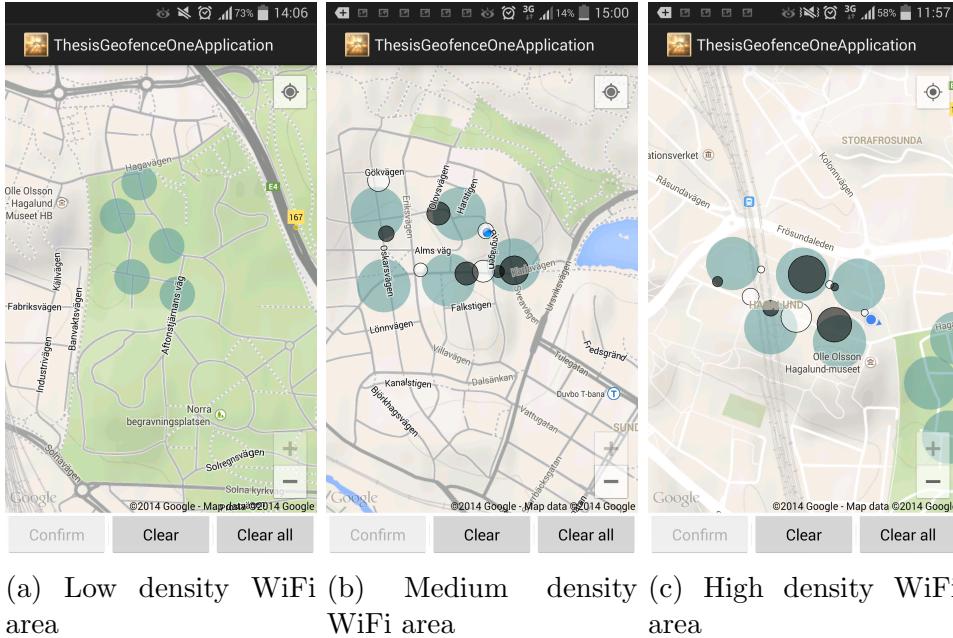


Figure 6: WiFi accuracy test

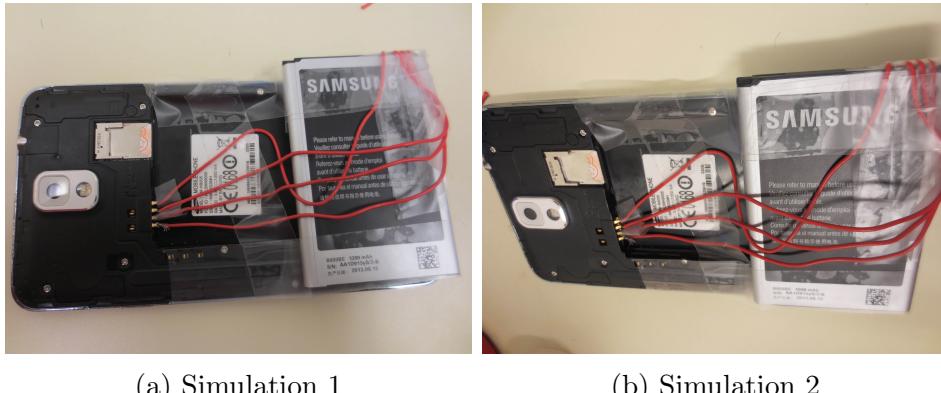
First of all note that in figure 6a there are no WiFi and therefore we obviously do not get any transitions nor location updates. We can see in the image that we don't even have a current position since we see no blue dot or arrow indicating where we are.

If we look at figure 6b we notice that one area has no enter nor exit transition. We can also see that there are two enter transitions in the area furthest to the right. That is because I walked in a circle and ended the test run in that area. The size of the accuracy seems to variate a lot. We probably get a quite good accuracy if we are between two houses and see two WiFi where as the larger accuracy are most likely areas where we see only one WiFi.

If we last take a look at figure 6c we can see that sometimes the accuracy is great, a lot better than in any of the two other and sometimes they are much worse. That they are a lot better sometimes is expected since there are many more WiFi to use to calculate our position. The answer to why some transitions are so large is not so simple. It could be that the multistorey houses lowers the accuracy of the GPS so that when someone using the Fused provider and reports positions of WiFi to Google the same WiFi setup gets a lot of different coordinates. It could also be that the known WiFi that we see overlap more or less exactly each other since all are in the same building and therefore doesn't add any or a little data for our calculations.

6.2 Battery test

I will walk with a laptop connected to the cellphone, which shall measure the power consumption with different settings, through different areas, the areas shall have different properties that will in the end affect the battery consumption. Below you can see my setup for the multimeter in between the battery and the cellphone. That is figure 7a and figure 7b.

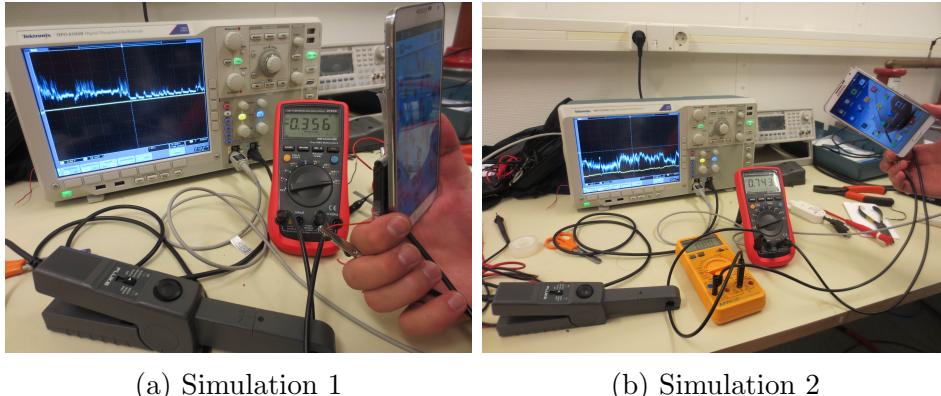


(a) Simulation 1

(b) Simulation 2

Figure 7: Battery setup

First we connected the phone and battery to an oscilloscope.



(a) Simulation 1

(b) Simulation 2

Figure 8: Oscilloscope

Once connected we compared the values from the multimeter with the values from the oscilloscope and realized that they where almost the same. We double checked with another multimeter. Since the multimeter uses true RMS the tops of the spikes in the current will be a bit to high. This problem

will be fixed once we connect the multimeter to the computer. The computer takes all the samples it gets during a second and stores the mean value.

If you look at the battery it connects to four different pins (top right in 7b). Two of them are the positive and two are negative. This means that there are two pairs and the one that is the main power for the phone (the one that gives the current for the software like OS and applications) I will be measuring from. Lets call that pair the normal pair. The normal pair gives the current for most of the components running on the phone. We tested both pairs while activating location services and saw only a difference in current on the normal pair. The extra pair gives current to the antenna among other things. Exactly what it gives current to was not revealed to me when I asked Samsung, it might be a company secret and since we did not see any relevant change in current from the extra pair during this test I haven't researched it more.

6.2.1 Comparison

To be able to see what the location features contribute to the graphs we need to see a few graphs without any location service doing anything. I've done two "dry runs". The first (figure 9) is with all the same settings and services running but with the WiFi turned off and the mobile data turned off as it is when we have only the GPS running. The second (figure 10) is with the same settings and services running but with WiFi and mobile data turned on, as it is when we run either the Fused or the WiFi tests.

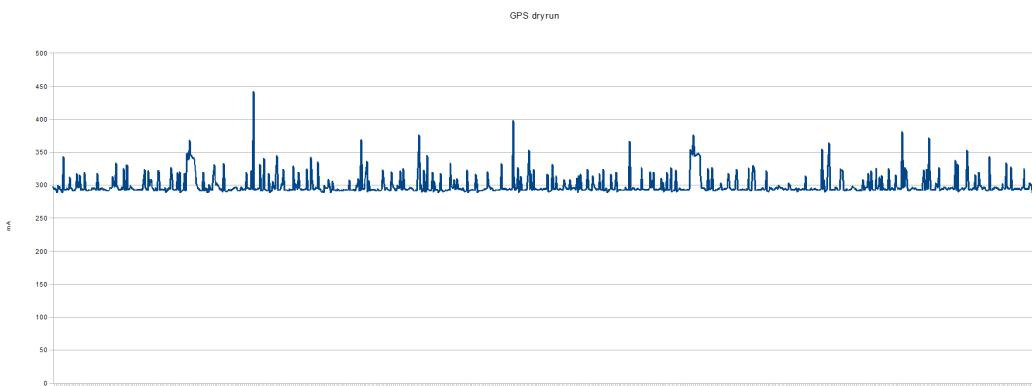


Figure 9: GPS dry run in mA

Mean value is about 300.

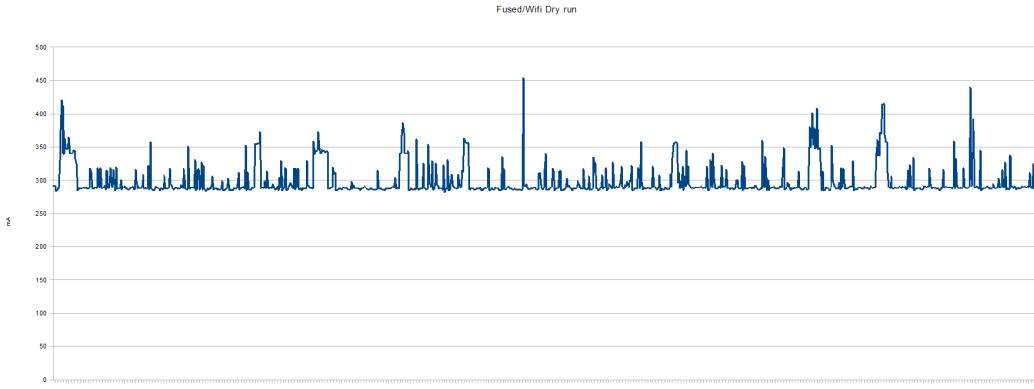


Figure 10: WiFi and Fused dry run in mA

Mean value is 300.

In both cases we see that there are a few irregular spikes of usage. If we remove these we can see that the current moves between 300 and 350 mA and that the less components that are active the less variation we get. The highest spikes goes up to 450mA but considering the length of the test and the length of these spikes then these spikes are uncommon.

6.2.2 Fused Battery test

In this section I will walk with both GPS and WiFi enabled through three different areas. The first area should have loads of WiFi and multistorey houses. This should encourage the WiFi part to take a greater role. The second area should be something like a residential area with a sparse amount of WiFi and one-storey houses, and last I'm going to go through a area with no WiFi at all. This is interesting since we want to know if the Fused provider will take measurable amount of more power than the GPS only provider when there are no WiFi around.

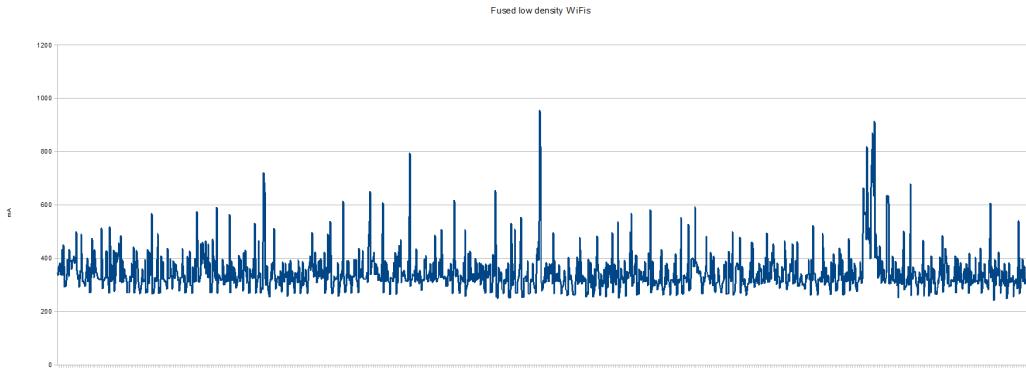


Figure 11: Fused low amount of WiFi area, current in mA

We can see in this figure 11 that the normal spikes are between 400 mA and 600 mA. The mean value is 350 .

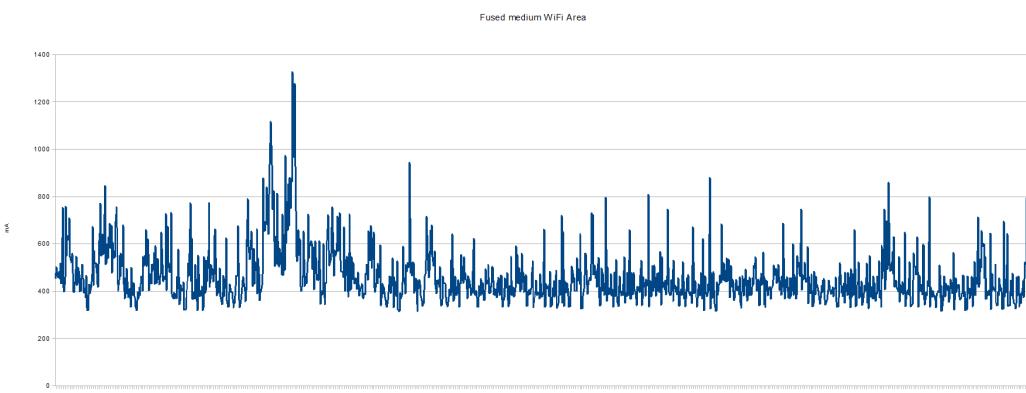


Figure 12: Fused medium amount of WiFi area, current in mA

We can see in this figure 12 that the normal spikes are between 600mA and 800mA. The mean value is 460.

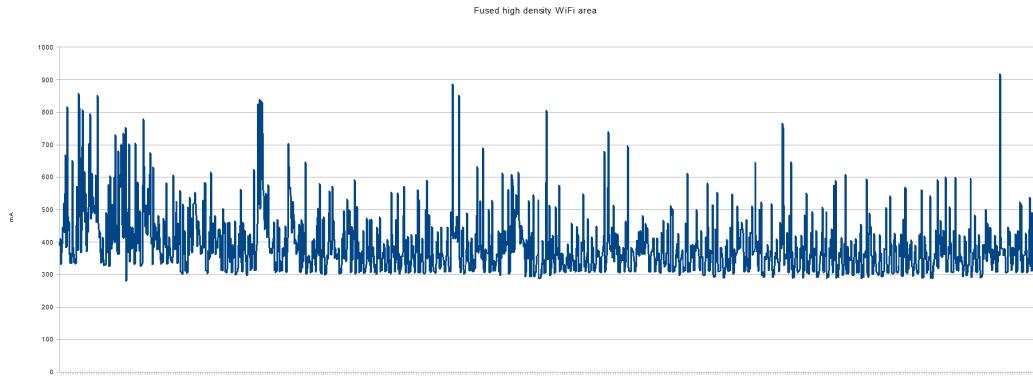


Figure 13: Fused high amount of WiFi area, current in mA

We can see in this figure 13 that the normal spikes are between 600 mA and 800 mA. The mean value is 610.

6.2.3 GPS Battery test

In this section I will walk with only GPS enabled. I will go through the same three areas in the Fused accuracy test.

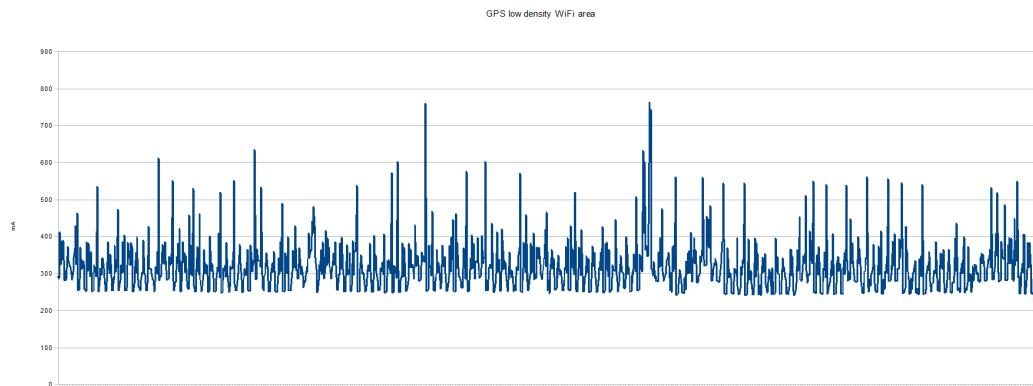


Figure 14: GPS low amount of WiFi area, current in mA

We can see in this figure 14 that the normal spikes are around 400mA to 600mA. The mean value is 470.

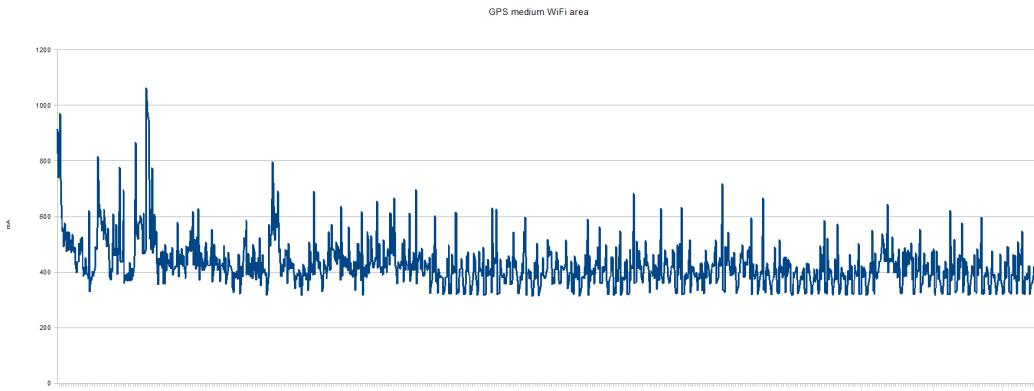


Figure 15: GPS medium amount of WiFi area, current in mA

We can see in this figure 15 that the normal spikes are around 400mA to 600mA. The mean value is 520.

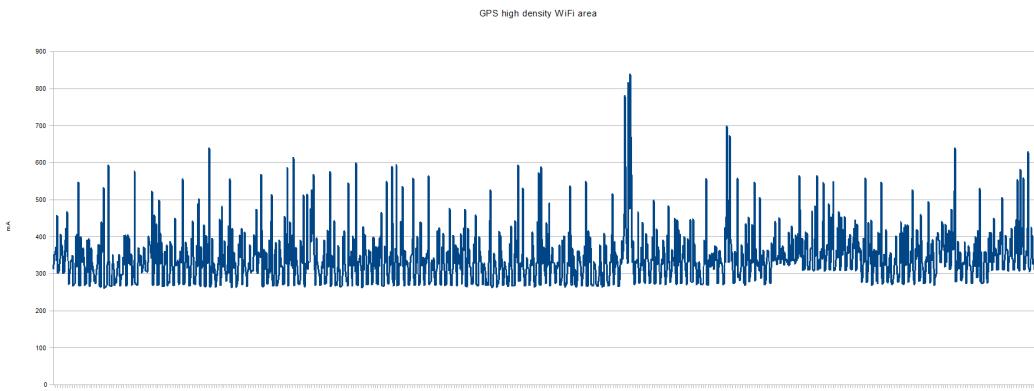


Figure 16: GPS high amount of WiFi area, current in mA

We can see in the figure 16 that the normal spikes are around 400mA to 600mA. The mean value is 340.

6.2.4 WiFi Battery test

In this section I will walk with WiFi and mobile data enabled. When we only have a WiFi enabled you might think that it is not interesting to go through the area which contains no WiFi since we already know that we will not get any location data. But even as we walk and will find no WiFi we still have to scan for WiFi and it is interesting to know how much power is used to scan for WiFi. This means that I will go through the same three areas as in both Fused Battery test and GPS Battery test.

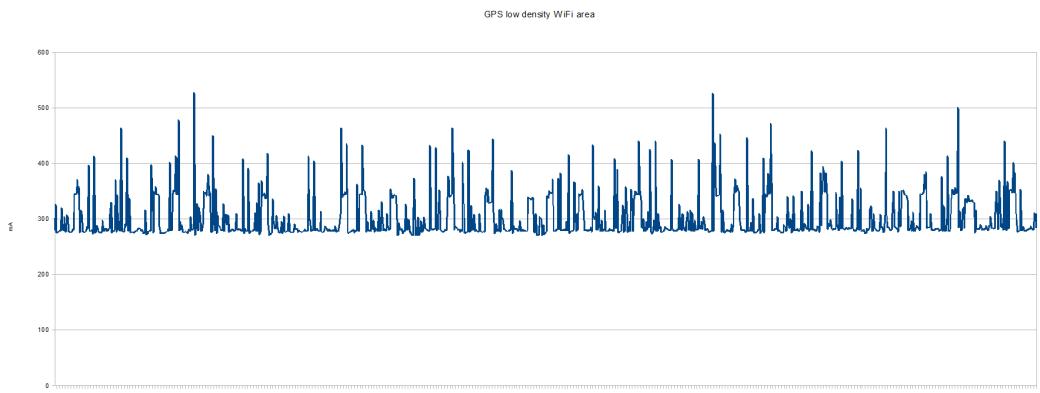


Figure 17: Fused low amount of WiFi area, current in mA

We can see in this figure 17 that the normal spikes are around 600mA to 800mA. The mean value is 300.

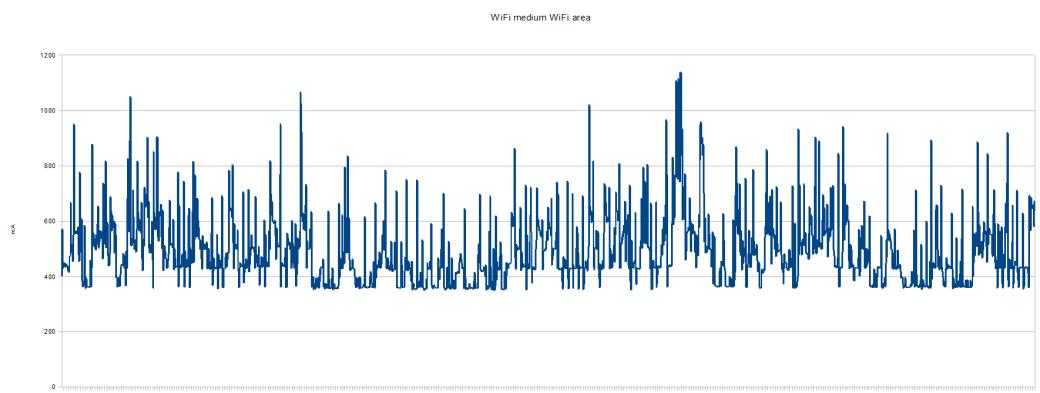


Figure 18: Fused medium amount of WiFi area, current in mA

We can see in this figure 18 that the normal spikes are around 600mA to 800mA. The mean value is 490.

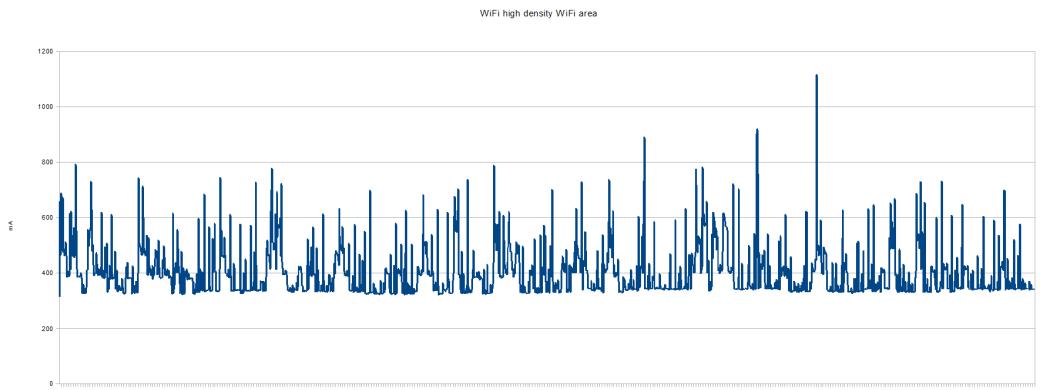


Figure 19: Fused high amount of WiFi area, current in mA

We can see in this figure 19 that the normal spikes are around 600mA to 800mA. The mean value is 400.

7 Evaluation

7.1 Accuracy

7.1.1 Geofence Border

We can see from the tests in the chapter *Test* section *Accuracy* and subsection *Accuracy test simulation* that when I enter a geofence we aren't necessarily a 100 percentage sure that we are inside the geofence. We could possibly say that I am close enough, that it doesn't matter, but then we haven't thought about what might happen a moment later. I might get an exit transition directly after my enter transition due to changes in accuracy and after that a new enter transition. This means that if I walk close to the border I might be getting multiple transitions for the same geofence in a short time period, this is almost never intended.

7.1.2 Solution one

One method of avoiding this problem is to say that once exited a geofence you may not enter it again for X amount of time. The main problem with this would be if we worked close to the geofence which means that we would be close to the border of the geofence the whole day. So after time X has passed we get the same problem again. One possibility of using geofences is to send deals to the client, a deal might not be available to the same device once or twice a week. If we have some of these limitations this solution would work.

7.1.3 Solution two

Another is to use two different geofences for exiting and entering transitions. The exit geofence can have a radius of 10 meters wider than the enter geofence. This would create a buffer-zone of 10 meters which we have to pass to both enter and exit the geofence (which now consists of two geofences). The main problem with this solution is that we are only allowed to monitor 100 geofences per application in Android and last time I checked only 20 geofences per application in iOS. Using this method would mean that we cut the amount of possible geofences that we can monitor at one given time in half. Fifty geofences for Android can be too few for many of the applications and 10 geofences for iOS are too few for most applications. Another problem with this is the size of the buffer-zone. If we have an accuracy much larger than the buffer-zone then the zone might as well not have existed. This means that we either set a maximum accuracy or use a dynamic buffer

zone that is dependent on the accuracy. Having a dynamic buffer zone is not preferable since if the accuracy becomes really low then the buffer zone might be too large for the geofence. Having a minimum accuracy level could work well with a static buffer zone.

7.1.4 Geofence size

Another aspect of this is the possible sizes of the geofence. If we have an accuracy of 10 meters and roughly 80 percentage of our possible location must be inside the geofence, then we can say that there is no possible way that we can enter a geofence smaller than 8 meters. We can possibly even say that it is extremely unlikely that we will enter a geofence that is 10 meters in diameter and even if we manage to enter it we would have to walk almost into the middle of the geofence before getting a trigger. Therefore we should be wary of using small geofences. In the iOS version there are or at least used to be a minimum size of 100 meters. This might be a little too large for my taste but I would not recommend anything less than 30 meters and would prefer to have at least 50 meters radius on each geofence.

7.1.5 Comparison

The choice of which location provider to use affects the accuracy in two different ways. First the accuracy of each given location will increase or decrease relatively to the other providers depending on the number of WiFi and how many GPS satellites we can find. The second way that the accuracy would change is the number of locations that the provider will give.

- When we look at the case of WiFi only it will feel like discrete events instead of continuously flowing events. With that I mean that if we look at a map we will not see ourselves glide from point A to point B, it would more look like we are taking jumps from point A to point B and the number of jumps will be dependent on the number of WiFi around us. The more WiFi the more jumps and therefore the shorter length between each jump.
- By looking at the GPS only we can say that as long as we have enough satellites it will look like we are gliding from point A to point B but as soon as we have less than three satellites we will stop getting locations completely. To get a location we need to be able to solve all the variables and those are latitude, longitude, height over the ocean, time and since we need one satellite for each variable we need at least four satellites to get any data. If we have four satellites and lose one satellite

we stop getting data. Say that we walk for 20 seconds before gaining our forth satellite then we will after those 20 seconds get a jump to the correct location.

- When we analyse the Fused provider we can see that mostly it looks like we glide from point A to point B but suddenly our position can make small jumps. This is probably when the Fused provider makes use of WiFi and can more accurately position ourselves. It can also be that we lost to many satellites like described above.

From a geofence perspective it doesn't matter if we glide or jump between locations, what matters are how fast a geofence transition will be reported. If we jump from point A to B instead of gliding, we can almost certainly say that the jumping movement will report the transition slower since we have to get further in into the geofence before receiving the location that creates the transition. If we have too small geofences and/or too large jumps we might even completely miss the transition. In many possible uses for geofences we have different stores, museums or other attractions where the owners could install own WiFi and then set the geofence a little larger than the WiFi range to make sure that we can't miss the geofence. This way seeing the WiFi would be the same as being inside the geofence. Note that many will probably want to give up the gliding motion if the jumps are small enough.

7.2 Battery

Mean value minus the mean value of the dry runs for each provider is displayed below.

	Fused	WiFi	GPS
Low density WiFi area	46	2	171
Medium density WiFi area	158	191	220
High density WiFi area	313	100	40

7.2.1 WiFi

As expected we see that the WiFi with low density WiFi in the area is almost zero. Since there are no WiFi to scan we use no extra current. If we compare the medium density area to the high density are in the WiFi column we see that the medium WiFi area has used more current than the high density. This is probably the case since in the high density area loads of WiFi comes at the same time as we approach a multistorey house but in the medium density area we get more continuously WiFi since the houses with the WiFi

are closer than the multistorey houses were. This means that if the WiFi is going to work the current will probably be in between the medium density value and the high density value. The low density value will I discard in the comparison later since we did not get any geotriggers from it, the only interesting thing about it is that if we don't have any WiFi around us this method will not use any unnecessary power.

The tests have been running on a device with an update frequency of five seconds. This means that we will try to get a location each five seconds. The power consumption should be linear with the update frequency in this case which means that if we increase the frequency to 30 seconds then the average power consumption should be lowered to 1/5 of the current average power consumption.

7.2.2 GPS

Now if we look at the GPS column we can see that the medium and low density is quite close together around 200 mA but the high density area is really low. With lots of multistorey houses around me the GPS had difficulty finding the satellites which is probably the reason why it uses so little current. To be sure more tests might have to be made. It could be that this is a special place that for some reason doesn't find the satellites that good but comparing them don't give us that useful information since the high WiFi density area didn't give us as good accuracy as we want from a GPS. We should note that the more satellites we find the more power the GPS location provider seem to consume.

The tests have been running on a device with an update frequency of five seconds. This means that we will try to get a location each five seconds. The power consumption should be linear with the update frequency unless we need to find the GPS satellites again. This means that if we increase the frequency to 30 seconds then the average power consumption should be lowered to 1/5 of the current average power consumption. Unless we need to find the satellites again as opposed to just asking the satellites for the needed data. The risk of having to search for satellites again increases with the distance travelled and the time spent not getting a position.

7.2.3 Fused

The Fused provider are a high accuracy location provider which means that it will use all the components it can to get as good accuracy as possible. The low density area should use the GPS almost all the time since it will not find any WiFi. Since we saw that the WiFi only measurement on the low

density WiFi area used almost no power, we can conclude that the Fused low density measurement should be about the same values as the GPS low density measurement. This is far from the case. One observation I did during the tests of the Fused runs was that sometimes the positng would jump a little bit. My guess is that it switches between the different components to get a position from all, but it might not be using all of them at the same time. This means when we switch to WiFi the power consumption goes down. Another possibility is that we did not need to search for the satellites as much as we did in the GPS only measurement.

The tests have been running on a device with an update frequency of five seconds. This means that we will try to get a location each five seconds. The power consumption should be linear with the update frequency unless we need to find the GPS satellites again. In this case which means that if we increase the frequency to 30 seconds then the average power consumption should be lowered to 1/5 of the current average power consumption, given that we do not need to find the satellites again. The risk of having to search for satellites again increases with the distance traveled and the time spent not getting a position. Fused provider is a combination of both the other two location providers so if we actually have to find a new satellite it will not affect the average as much as in the GPS only part.

7.2.4 Comparison

Considering that the low density WiFi area is almost always uninteresting since there is almost always at least one WiFi active somewhere during the route you are walking, unless you are creating an application for monitoring large gardens or cemeterys. If we remove the GPS only method in the high density WiFi area, which seems suspicious, and keep the GPS only in the low density WiFi area, since it doesn't matter that there are no WiFi for the GPS only measurements, then we can see the following table for average power consumption per second.

Fused	WiFi	GPS
225	150	200

If we look at the data then we can see that the GPS is quite constant around the 200 mA per second but the WiFi varies a bit more depending on how many WiFi are around and the Fused seem to vary even more. This means that the Fused just isn't the one with the highest average value but also with the highest worst case consumption. The WiFi is the one with the lowest average value but also it's worst case scenario is the same as the average value of the GPS.

If we look at the mean value from the dry runs the program itself takes around 300mA per second. If we must have a location the lowest value is 150 and adding 50 or 75 more mA per second isn't that much (50mA more for GPS and 75mA more for Fused). But we should say that the 300mA per second is with a large screen and full brightness on. The main battery consumption thief on a phone with no application running is the brightness so if we use a small phone and lower the brightness it's probably not impossible to get the power consumption down to 100mA. If we consider that the phone only takes 100mA and we add the 150 for the WiFi we get 250 mA per second. Adding 50 or 75 mA per second more will then much more significant. It will drain the phone about 30 percentage more. And turning on the most power saving location service will increase the consumption with 60 percentage.

Now if we look at a phone that doesn't have any extra services running and its screen is off, we can get down to 10 mA per second. Usually if we just install one extra application this goes up. But if we have a phone that is using only 10 mA per second then by just turning on the most power saving location service we will increase the current consumption by 1500 percentage. But we should remember that this is for an update frequency of 5 seconds. If we just change it to 30 seconds there will be a vast difference.

There are also a few extra questions. The WiFi location services are supposed to be the most power saving one of them all, which it also is, but compared to an idle phone there aren't a vast difference. One assumption I did during this test was that the user was walking around using the location services. But what happens if we stand still? The GPS would not need to find the satellites over and over again since we aren't moving but the WiFi only location provider how would it work in this case? My guess is that since we already have scanned the area it might be smart enough so that we would not need to do so many more checks. It would just need to check the id of the WiFi around us and realize we have not moved. This might eliminate the need for using the mobile data, which in turn would lower the power consumption a lot. If we are really lucky we might get closer to the power consumption on the low density WiFi area. If this is true and your phone is idle you might not even notice that the location services are on.

When we talk about the WiFi only location provider as a good solution we must remember that for it to work properly the WiFi location must be known by Google servers. This means that someone must have used the Fused provider and walked there and reported the positions. For geofences there are some solutions to this problem. All the geofences must be located on a server which means that the programmers know or can see where they are. If we for example develop an application for a company that sells gas, the company employees can then once a month just turn on their Fused

provider during their shift to report positions. If on the other hand we have a restaurant in the middle of a large city my guess is that the WiFi are already reported since there is always someone that uses Google maps or something like it to find their way. When we aren't looking at geofences but a map we demand higher accuracy which is what we get from the Fused location service provider.

8 Conclusion

8.1 Accuracy

8.1.1 Geofence border

Considering the two possible solutions discussed in the chapter Discussion subsection Accuracy I believe we have to choose the solution that we may not enter a geofence again after time X has passed since we usually need as many geofences as we can possibly get.

If the amount of possible geofences to monitor increases we probably want to switch over to the other solution (two geofences with a buffer zone to enter and exit).

The choice of which method to use should be application dependent. If we need less than five geofences then the two geofence method will be the best. It could be as simple as that we want different content or information depending on our current geographical area, which means that we are only interested in knowing our current area and when we exit it. On the other hand we might want to have a geofence per gas station in as large area as possible.

For this thesis the small overlap of the geofence border isn't an issue. I will not doubt an enter or an exit transition to be true but again depending on what the application should do we might want higher accuracy rate, this could be easily achieved by using the two geofence methods described in the discussion.

It should be known that currently the minimum size of an iOS geofence is 100 meters. In Android there aren't a minimum size but we probably should not use geofences smaller than 50 meters. Of course smaller geofences will work but if we get our accuracy from a WLAN then our accuracy probably will be in the range of 10 meters radius, which would in turn mean that we can't enter a geofence with a width smaller than 10 meters radius. And a geofence with a width of 20 meters radius we will have to walk half way into its center before getting an enter transition. And if we get a rough WLAN position it might even be as long as 30 meters radius!

8.1.2 Location provider

Even though the location provider settings isn't something the programmer will choose the choice of which one the user chooses will affect the accuracy in different ways. Since the WiFi only and sometimes even the Fused provider are prone to look like our current location are jumping between locations instead of gliding leads us to want even larger geofences if these are involved.

The Fused provider will probably only look like it is jumping when correcting its previous location so we could argue that we should not consider the jumps it makes as jumps in this case. Increasing the update frequency would not solve this phenomena since the problem is that a set of WiFi indicate a discreet location. Only by turning on the GPS will we be able to get a continuous flow of locations.

When looking at the pure accuracy of each provider the Fused provider usually has the best accuracy. The GPS provider usually gets a quite good accuracy. If we have a lot of WiFi the WiFi only provider can give us a really good accuracy as well but it is really unreliable. If we want to use small geofences the discreet behaviour of the WiFi providers may be a large issue even though in the Fused provider we also have the GPS we might get jumps that jumps over a small geofence. Even though in the Fused provider the jumps are corrections of the current position it might not be a behaviour we want. We might rather loose accuracy and delay a correct position if we get a continuous flow so that all geofences can be triggered.

I started by saying that the developer can't control which provider is used but in the case the application needs a high accuracy we can always prompt the user and ask him or her to use the higher accuracy provider.

8.2 Battery

8.2.1 Update frequency

If we have our phone active and we have a large phone the location services will not take so much more power, we might just loose a third of our battery. But if we have our services running in the background with a high frequency we will lose extremely much more battery. If we decrease the frequency we might get the phenomena that we always need to check for the GPS satellites, which will take some extra power, but otherwise lowering the frequency of the updates should be linear to the power consumption. In my tests I've had an update frequency of 5 seconds, which is a really fast update frequency. Lowering it to 30 seconds should not do so much for the accuracy of a walking person, but consider that you are driving a car instead in 100km/h then a 30 seconds update frequency will be a vast difference in the update frequency. In fact if you travel 100 km/h and you have an update frequency of 30 seconds then you will be able to travel 830 meters in between each update. One of the later versions of the location provider has a probability guess that indicates with which probability we are walking, driving a car or bicycling. Using this guess and the fact that if the user currently has the screen on to change the update frequency should make geofences accurate and not to much power

consuming. Another factor is the geofence size, if we have geofences that are in the size of 10 km then of course even while driving a car with an updating frequency of 30 seconds will be enough.

So the actual update frequency should be calculated from the size of the geofence, travelling method or rather the possible speeds we travel in and what type of provider the user is using. From a power saving modes point of view I would say that we need at least one update per geofence radius or per geofence diameter. If we use any of the other two I would say at least two updates per geofence radius size.

The developer of an application should also consider making the update frequency dynamic to the distance of the closest geofence. If we are walking and the closest geofence are 10 km away we might not need a location update for another 30 minutes. If we on the other hand are 10 meters away we might want an update in the range of 5-10 seconds. The developer should also consider the possibility that the users work or home are close to a geofence which if we implemented the above method might get a lot of unnecessary updates. The developer should consider implementing a stand still or back-off function. Like if we would have gotten the same position the last five updates we might increase the time to the next location update.

8.2.2 User friendly

Another thing to think about is if the user has the high accuracy (Fused), power saving (WiFi) or the GPS only provider activated. If the user is using the power saving mode then the developer probably should lower the frequency since the user most likely has a problem with the battery or for some other reason don't want to spend so much battery at the moment.

It might be so that when we are standing still the WiFi only location service provider might drop a lot of its power consumption. This means that it might only use power when we are moving and thereby save a lot of power since most people work in a small area. We should then remember that the WiFi only service provider only works once the location has been reported to Googles servers. Which means that the company developing the applications should consider a way to report the locations without the need for the clients to do so.

8.2.3 My choice

Over all I would recommend the WiFi only or the Fused provider for location services. If we need more power choose the WiFi only otherwise choose the Fused provider. The GPS only provider isn't saving that much power but

might loose a lot of accuracy. In other words I would personally choose the WiFi Only provider and if I want to use a application that needs more accuracy like Google maps I would switch to Fused provider. I would in other words never use the GPS only provider but discarding it completely would be wrong since it has one benefit that I haven't mentioned. Compared to the Fused and WiFi only providers the GPS only provider doesn't give or receive any information from Googles servers. This is good from a security perspective since Google can't gather any statistics about you and you do not need to use the mobile data. To not use the mobile/WiFi data is good in cases where you know that you have some but almost no internet connection. Examples of this could be in the country house or far out on the ocean.

9 Future work

First of all I have discarded the iOS completely since it would have been a to much for me to include it in this thesis. It's GPS might use a lot more or less power I can not now until I do some tests on an iPhone. Secondly it would be interesting to see how the different location providers react on a still environment (especially the WiFi only and possibly the Fused provider). A third interesting comparison would be to see how the Fused and GPS only providers handles areas where we can get a GPS connection but no mobile data.

10 References

- 1 "Indoor Location Detection using WLAN", *Anqi Luo & Lei Ge*, Master of Science Thesis TRITA-ICT-EX-2009:209, KTH.
- 2 "Location aware communication", *Johan Eckerström*, Master of Science Thesis urn:nbn:se:kth:diva-93110, KTH
- 3 "Location aware web access", *Jenny Charvandeh* Master of Science Thesis TRITA-ICT-EX-2009:121, KTH
- 4 "Location Based Pre-caching and Network Coding in Smart Content Distribution", *Bo Zhang & Luis Delgado Romera* Master of Science Thesis XR-EE-KT 2013:004, KTH
- 5 "Protecting Location-Data Against Inference Attacks Using Pre-Defined at Personas", *Amir Hossein Chini Foroushan* Master of Science Thesis TRITA-ICT-EX-2011:258, KTH
- 6 "Reliability of Location Determination Systems", *Shuang Lei* Master of Science Thesis TRITA-ICT-EX-2009:109, KTH
- 7 "Making the communication network aware of the user's location", *Christian Jansson* Master of Science Thesis urn:nbn:se:kth:diva-96522, KTH