Low Energy and Sufficiently Accurate Localization for Non-Smartphones

Kuldeep Yadav, Vinayak Naik, Amarjeet Singh, and Pushpendra Singh

Indraprastha Institute of Information Technology, Delhi, India

***Abstract*—**

Location-aware applications are steadily gaining popularity across the world. However lack of GPS in low-end programmable phones ($<$ $100) and general absence of Wi-Fi infrastructure in developing countries prevents users of low-end phones (majority of population in developing countries) from using such applications as their phones can not get location. GSM-based approaches, such as using Cell IDs, have been developed as they do not require specific hardware on phone and need no additional infrastructural support. However, Cell ID based approaches require access to a comprehensive database of Cell IDs which does not exist in developing countries and its growth is not promising either.

In this paper, we present a novel GSM-based approach of using Cell Broadcast Service (CBS) messages for localization. Unlike Cell ID based approach, our approach does not depend on comprehensive database and can run on programmable low end phones. We demonstrate the effectiveness of our approach on data collected in New Delhi, India. We further propose two space-time history based algorithms to improve upon the localization accuracy of our baseline CBS approach. The proposed algorithms provide up to 35% improvement in accuracy over the baseline method. We test our algorithms for two different cell phone operators and show that the algorithms perform consistently better. At the end, we present potential location-aware applications which can be built using CBS based localization.

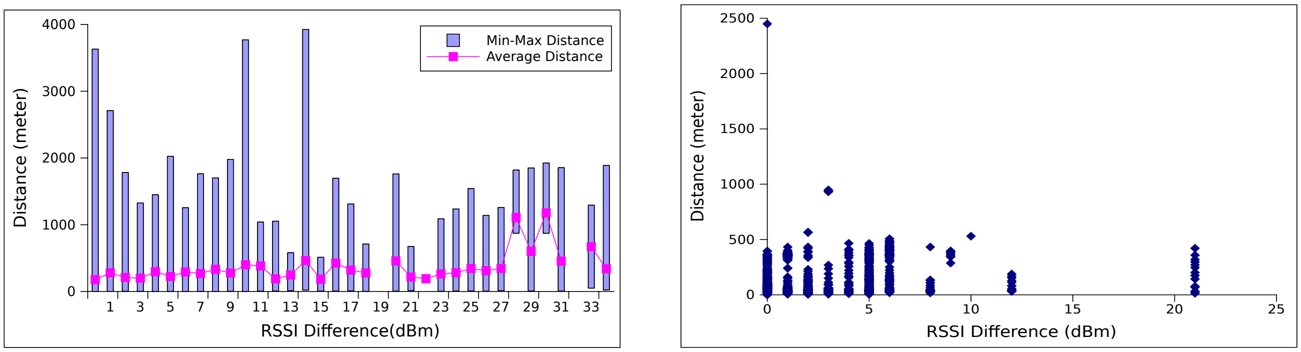
I. Introduction

User location has been an integral part of user context in delivering context-aware services such as navigation, activity recognition, local business search, and friend finder services etc. Interestingly, all context aware services do not require same level of accuracy for current location. For instance, navigation applications require high level of accuracy (10 meter) whereas if one has to share location with online social networks, required location accuracy could be in hundreds of meter. Many technologies/approaches are available to measure user's current location on mobile phone. The majorly used ones are following:

1) **GPS (Global Positioning System)**: Highly accurate (app. 10-100 meter) satellite based approach and most common for high-end phones. However, it is high energy consuming, requires special hardware, and only works outdoors.
2) **WiFi based Positioning** : A perceptual map of wireless APs identifier with respective signal strengths and approximate location is created by wardriving and stored in a database. The mobile phone queries this database to estimate current location. Though it can work indoors but it is also high energy consuming and requires special hardware besides needing the Wi-Fi infrastructure which does not exist in majority of countries.
3) **GSM based Positioning** : There are two kinds of GSM positioning approaches: Base station assisted and Independent. Base station assisted approaches require installation of sophisticated component on base station and hence require assistance from operator. Base station independent GSM positioning approach is based on Cell ID, where like WiFi based positioning system, a perceptual map of GSM Cell towers is created using war-driving and this is queried to estimate the current location of the phone.
   This method does not require any extra hardware and easily works on phones having GSM connection (app. 85% of phones). This approach have low accuracy than GPS and WiFi (app. 100-1500 meters) which in turn depends on many factors like coverage of Cell ID database, Cell ID density in an area, visibility of Cell IDs on phone etc. as we will discuss later in the paper.

Mobile phones having GPS and Wi-Fi capabilities are costly, so a large number of phones do not have them. It is predicted that for the next five years, over 50% of the phones will not have GPS [5]. Also in most part of the world, WiFi network is not available. Apart from cost, mobile phones are highly energy constrained and continous use of GPS and WiFi drains the battery very quickly. For the class of applications that do not require fine grained location accuracy, Cell ID based GSM localization is better suited due it its wide availability and low power consumption. In fact for low-end phones (without GPS/Wi-Fi capability) as found majorly in developing countries, this is best suited [17]. However, it has following limitations which need to be overcome:

1) According to GSM standards, a phone can receive signals from 7 different Cell towers [16]. But most of the phones can access (using APIs) to only one Cell tower to which the phone is currently connected [13]. Due to this, Cell ID based approach offers a coarse grained accuracy. In this paper, we will focus only on single Cell ID based localization.
2) For Cell ID based localization, perceptual map (Cell ID database) has to be created by wardriving. Wardriving is very costly because it is practicaly impossible to cover each and every street of a country to create database of Cell Ids. There are few crowd-sourcing based open source Cell ID database, like Open Cell ID, which have only few entries and they too become obsolete due to lack of participation.



(a) In whole dataset, Min-Max bars representing minimum and maximum distance between two position into same cell ID. For instance, for a difference of 14 dBM in RSSI, distance between those points can range from 0 to 4000 meter

(b) Only for single cell ID, X-Y scatter plot for showing variance in RSSI difference vs distance between two positions into same Cell ID. For instance, two points with RSSI difference 21 dBM can have a distance of 0 to 500 meter

Fig. 1: RSSI analysis on self collected dataset

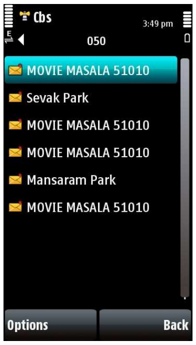

Cbs 3:49 pm
050
MOVIE MASALA 51010
Sevak Park
MOVIE MASALA 51010
MOVIE MASALA 51010
Mansaram Park
MOVIE MASALA 51010

Fig. 2: A native application in Nokia Symbian phone displaying last six received CBS Messages

We propose using Cell Broadcast Service (CBS) messages to provide localization for low end phones. CBS is a GSM standard where nearby Cell towers broadcast their locality name. A phone can receive CBS messages from only one Cell tower to which it is currently connected. Our proposed scheme removes the necessity of building Cell ID database and can suport location aware services which do not require fine grained accuracy. Figure 2 shows a native application in Nokia Symbian phone displaying last six received CBS Messages.

The primary contributions of the paper are as follows:

1) To our knowledge, ours is the first study to propose and evaluate CBS messages to provide localization for low-end (those without accelerometer and GPS) mobile phone users
   a) Architecture of a system that uses CBS messages for localization
   b) Identification of challenges in realizing working system
2) *TimeWeighted* and *FrequencyWeighted* algorithms, theoretically suited for fast and slow movements respectively, to improve localization accuracy over a baseline approach,
   a) Evaluation of accuracy of the algorithms using 58 traces of real data for two different operators
   b) Improvement of up to 35% in accuracy over baseline approach
3) Potential applications that can work on the accuracy provided by our approach of using CBS messages for localization

The paper is organized as follows. Section II describes related work and problem definition. Section III presents CBS based localization architecture and Section IV discusses challenges associated with it. Section V presents two algorithms, which enhance accuracy of CBS-based localization over baseline approach. In Section VI, we present evaluation of proposed algorithms using collected data. We present potential applications, whose demand for location accuracy is satisfied by our CBS-based approach, in Section VII. Finally, we conclude in Section VIII.

II. Background and Problem Formulation

Prior work related to GSM-based localization can be divided into two categories: (A) Cell ID-based Approaches and (B) Fingerprinting-based approaches.

A. *Cell ID-based Approaches*

In this approach, Cell IDs are fetched using phone APIs, and looked up in an existing database to provide localization. To the best of our knowledge, none of the mobile phone operators reveal exact location of the Cell towers. Hence, Cell tower location is approximated using crowd sourcing/war driving data, which could be several hundred meter away from its actual location. If there are multiple visible Cell IDs, the approaches compute some function, e.g. centroid, of all the geo-coordinates (latitude and longitude) obtained from the database.

As discussed above, there are limitations on how much visibility phone APIs provide to third party developers for accessing Cell IDs. Many of prior works assume that phone APIs provide access to multiple Cell IDs, as far as seven, at a time [16]. However, Ramesh et al [4] tested on Symbian OS for N95 and Android OS for GI phone and found that it gives access to only one Cell ID to which the phone is currently connected. On the same line, Nurmi et al [15] have found that all the Nokia S60 and Nokia 900 also give access to only one Cell ID. This significantly reduces accuracy of the localization as compared that obtained had there been access to seven Cell

IDs. Google Mobile Maps'(GMM) My Location [1] app works on a single Cell ID-based approach, where it provides a median localization error of 656.37 meter for a rural area and 503.89 meter for an urban area, as tested by [3]. The localization error depends on density of cell towers. Since in urban areas density of cell towers is high, so this method will give good location accuracy.

As identified in Section I, it is hard to get a comprehensive database of cell IDs. There are some proprietary databases, such as one used by GMM, which are not publicly shared. There exist open source initiatives, e.g. OpenCellID [2] and Cell Spotting [3], which build their database using crowd-sourcing. To check the coverage of open source cell ID databases, we selected two widely used operators in New Delhi. We call them X and Y for anonymity. On our self collected dataset of Cell IDs for operators X, we have found that out of 252 cell IDs, OpenCellID contained only 65. For operator Y, the number was only 21 out of 164 as shown in Table I. We cannot find out comprehensiveness of the GMM as it is not publicly available.

Crowd-sourcing for building cell ID database seems to be in-effective due to (A) lack of incentives as people need to incur airtime charges for contributing to the databases and (B) lack of GPS-enabled phones in developing countries.

Operator	No of cell IDs	Found on OpenCellID	%
X	252	65	31%
Y	164	21	13%

TABLE I: Success rate of Open Cell ID (most extensive open source database of cell IDs) on our dataset collected in New Delhi region

B. Fingerprinting-based Approaches

In this approach, RSSI (Received Signal Strength Indication) is also collected along with Cell IDs during war-driving. Typically, a fingerprint constitutes Cell IDs, their associated RSSI, and GPS locations that are represented in a vector form. For this approach, database size is larger and more effort is needed during war-driving. During the tracking phase, Cell ID(s), associated RSSI, and stored vector space of fingerprints is searched using KNN (K Nearest Neighbor) to approximate coordinates. Here, KNN uses euclidean distance in RSSI space as a metric to find closest stored fingerprint [16]. This approach gives accuracy better than the cell ID-based approaches since granularity of stored information is more. However, it requires more storage and computation power.

Continuous war-driving effort is required in this approach because signal strength keeps on fluctuating due to changes in physical environment. It works good when there is visibility of seven cell towers and their respective RSSIs. Recent results demonstrates that RSSI measure from single cell tower is not a good measure to calculate movement [4].

An RSSI difference is the absolute change in the RSSI, for a given Cell ID, when user moves from one location to another. In our database, we had 24064 unique RSSI difference values from 410 unique cell IDs. We plot maximum, minimum, and average distances for each RSSI difference. As seen in Figure 1a, the average difference is almost constant for RSSI difference ranging from 1 to 9 dBm. We zoom in on one cell ID and plot the data (Refer Figure 1b). We see the similar behavior for RSSI difference ranging from 1 to 6 dBm. This concludes that RSSI is not a good measure for GSM-based localization as one observes similar RSSI values between two points with large physical distance between them.

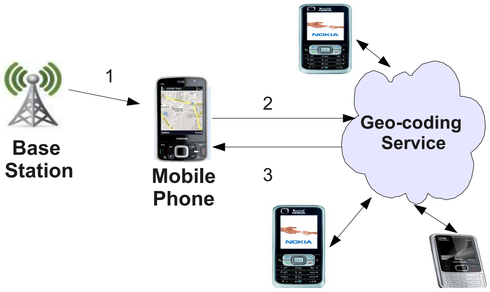


Fig. 3: Architecture of CBS based Localization System

III. Architecture of CBS-based Localization and Pilot Collection of Data

The CBS messages are broadcast by Cell towers to all the phones in its range [2]. CBS is defined in the phase II of GSM standard 3.49 [1]. The users need not pay airtime charges to receive CBS messages, even while roaming outside of their home area. The CBS messages are commonly used to broadcast information about weather forecast, landmarks/area names, news, announcement by governments, etc. All this information can be broadcast simultaneously on different channels. A cell tower typically broadcasts the locality/landmark name, where it is located. Channel 50 is reserved for broadcasting location/area names. Most of the phones come with built-in APIs to capture CBS messages.

A. Architecture of CBS-based Localization Scheme

In this subsection, we describe architecture of our proposed solution to use landmark names in the CBS messages to find users' locations. Figure 3 shows architecture of our working CBS-based approach. The data flows as follows, depicted by numbered arrows in the figure:

1. GSM base station broadcasts CBS messages, each containing a CBS string mentioning location name or advertisement. The messages are received by our application running on the phone.

2. If the message content is location, then phone checks for geo-coordinates of the landmark in its local cache. If it is not available, the application makes a request to cloud-based geo-coding service.

3. In reply to the geo-coding request, the geo-coding service returns geo-coordinates of that location to the phone. The application adds it to local cache of the phone. Cloud based

[1] http://www.google.com/mobile/maps/
[2] www.opencellid.org
[3] www.cellspotting.com

geo-coding service is likely to get request from many phones, using which it builds a cache of all location names with their geo-coordinates. Phones can download this global cache proactively to avoid frequent requests to the cloud.

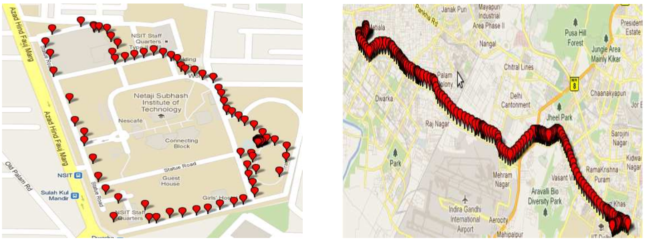
Above described approach is the most basic way of estimating a user's location using CBS messages and called as baseline approach. Baseline approach is identical to Cell ID approach described in Section II-A.

B. *Pilot Collection of Data*

To characterize accuracy of CBS localization approach, we collected CBS messages for operators X and Y in an urban setting of New Delhi, India. Five volunteers ran our data collection application for three month. We have collected this data to measure accuracy of the baseline approach and possibly design new algorithms to improve upon that accuracy.

Our data collection application is written in J2ME. We have tested it on Nokia S60 and Nokia S40 phones. Though we have collected data using Nokia phones, we have found that nearly all Java-enabled phones provide APIs to receive CBS messages. For example, the phones from Samsung, Sony Ericsson, Black Berry, etc work fine but their APIs to get other location information like Cell ID differs since each provider gives proprietary APIs to access information. Our application can run on all these platforms with minor modifications.

The application collects CBS messages on channel 50, records the following details at the time of reception of a message timestamp of receipt of the messages, cell ID, MCC (Mobile Country code), MNC (Mobile Network Code), and GPS coordinates (if GPS is available on the phone). Volunteers were given choices to start and stop application at any point of time. After collecting each trace, participants can tag their activity as walking, traveling, or both. Application provided two methods to upload the trace – (a) using phone's data connection and (b) transferring it to PC first and then uploading it using PC's Internet connection. Figure 4 shows two traces collected by a volunteer; walking traces is about 3 km long and collected around the campus while traveling trace is around 23 km collected while going from campus to home.



(a) Walking trace of about 3 km around the campus (b) Travelling trace of about 23 km from campus to home

Fig. 4: Representation of a walking and traveling trace from our dataset

Nearly half of our traces did not had GPS coordinates due to volunteers being in indoor. For consistency purpose, we have only considered the traces which had GPS values nearly all the time in this paper. We list out some of the statistics about the dataset in Table II. We analyze the collected data in the next section and list out challenges in using CBS message for localization.

State	X	Y	Combined(X+Y)	Avg Duration (Minutes)
Travelling	27	12	10	46
Walking	12	7	7	65

TABLE II: Number of travelling and walking traces in CBS dataset across two different operators X and Y

IV. Challenges in CBS based Localization

Data from our pilot study brought forth non-trivial challenges that require addressing before even the baseline approach can be used effectively. We addressed some of these challenges in our prior work [9], [10] but this paper presents comprehensive analysis with a bigger dataset.

A. *Filtering of Advertisement Messages*

CBS messages contain advertisements in addition to location names. It is essential to filter out these advertisements. In case of X, we found that number of advertisements differ among operators X and Y, as shown in Table III.

Operator	Total CBS Messages	Advertisements(%)
X	3106	48%
Y	1173	60.53%

TABLE III: Percentage of Advertisement CBS Messages in our Dataset collected for operator X and Y

It was observed that advertisements contain some common patterns such as special characters ('*','#','%','@') or continuous digits like ('55050'). Using these two discriminators, we designed a regular expression to on-line filter all the advertisements at the phone itself [10]. We got 100% accuracy in filtering advertisements when the regular expression was applied off-line to 4279 CBS messages in our dataset.

B. *Geocoding of Landmark Names*

As per our architecture, CBS location messages need to be geo-coded using a geo-coding service like Google Maps. Among all the on-line maps services, we found Google Maps to be most effective, and we have used it for all the experiments in this paper. We obtained 143 unique CBS location names in our dataset, among which 30% of location names could not be geo-coded by Google Maps at first. We call them false negatives. Primary reasons for occurrence of false negatives are the following:

1) Location names may exist differently (in the geo-coding service), e.g. there could be a spelling difference, use of short hand abbreviations, or with a completely different name. For example, 'Matiyala' and 'Matyala', 'Uttam Nagar' and 'Uttam Ngr', 'Dwarka Sec-3' and 'Sec-3 Dwarka'.
2) There is no publicly available extensive GIS database.

We have employed following approaches on the location names, that could not be geo-coded directly by Google Maps:

1) Sanitizing Location Names : To resolve the ambiguity present in location names, we do a pre-processing of landmark names before sending them to the geo-coding service. Pre-processing algorithm apply following steps to sanitize the CBS location names:
    a) Replace special ’-’ character by a space, so ’Dwarka Sec-02’, ’Dwarka Sec-2’ and ’Sec-2-Dwarka’ are converted to ’Dwarka Sec 02’, ’Dwarka Sec 2’ and ’Sec 2 Dwarka’ respectively.
    b) Numerical characters in the location name are separated out from surrounding text characters e.g. converting ’Dwarka Sec2’ to ’Dwarka Sec 2’.
    c) After removing special characters from location name, search for popular abbreviations in location names like ’NGR’, ’SEC’, ’VHR’ etc., and replace then with its full form like ’NGR’ for ’Nagar’ followed by a space. We have manually populated this mapping table from the location names.

    After pre-processing by the above algorithm, Google Maps service was able to geo-code nearly 50% of the false negatives. Other 50% of the names were not present on map service or existed with a different name. For instance, the location ’Dwarka Mor’ exists on Google Maps and can be geo-coded, but same location with a different name ’Kakrola Mor’ does not exist on Google Maps.
2) Use of on-line map based business search services: For the location names which are completely missing from digital maps or exists with a different name, we took help of the data present in on-line map-based business search services like Google local search. These business name are often collected through crowd-sourcing, so many of the location names (not found on Google maps otherwise) were present in business names. After retrieving business names, we applied K-means to approximate geo-coordinates for a location. However, currently we cound not verify a location’s geo-coordinates automatically and we leave it to future work.

We believe that a common algorithm that can work for all the names is hard to achieve due to non-standard nomenclature for CBS messages and poor GIS database (specially in developing countries). However, it is still a one-time task to geo-code the names which are not automatically geo-coded by any service and requires much less effort than the wardriving task used by Cell ID-based approaches.

C. Inaccuracy of Geo-coding Services

We are using geo-coding services, e.g. Google Maps, for finding geo-coordinates of the location names. There are inherent errors within these services e.g. a location called “Dwarka Sec-3” (a neighborhood in Delhi) is not mapped with geo-coordinates representing the central location of that neighborhood. Such errors vary from one location name to another and get introduced in the result. From our dataset, we got an aggregate of 143 unique location names from CBS messages. During each of the trace, we also collected the GPS coordinates and mapped them to the corresponding landmark name received at that instant. Since there were multiple GPS coordinates mapped to a single location name, we took an average of all the GPS coordinates collected for a given location name and define that as the calculated GPS coordinate for the corresponding location name. We then calculate the error, in the localization, for a location name as the distance between the calculated GPS coordinates and the geo-coordinates returned by the geo-coding service for that location name.

Figure 5 shows a bar graph of distribution of error in terms of percentage of the landmarks names. Out of 143 landmark names, 16% of the names could not be geo-coded. About 58% of the names which were successfully geo-coded, geo-coding error is more than 600m. This motivate us to build algorithms which can reduce the error produced by geo-coding.

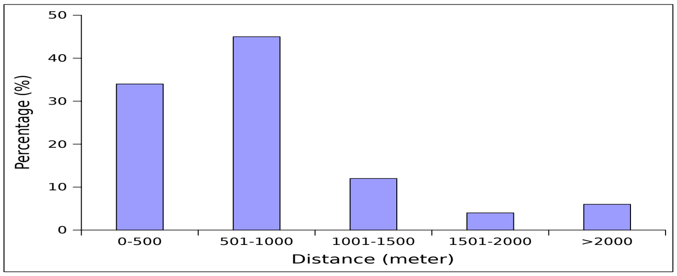


Fig. 5: Distribution of error from inaccurate Geo-coding services. For 58% of the names, error is more than 600m.

D. Heterogeneity in Landmark Names Among Operators

Similar to cell ID-based approaches, CBS-based approaches also suffer from operator heterogeneity i.e. broadcasted CBS location names for a particular place can differ among operators which may affect localization accuracy. In Section VI, we will show impact of operator heterogeneity on accuracy of localization by analyzing results from experiments with different operators. The challenge is to tolerate this heterogeneity.

V. Algorithms To Improve Localization Accuracy

Baseline CBS based localization takes the most recently received CBS message’s geo-coordinates to approximate the location of a user. Baseline approach does not always give good results due to two inherent errors: one which is caused by geo-coding service (described in Section IV-C) and other due to the fact that CBS location names may be far away from user’s actual location. A key insight towards reducing the impact of these errors is that we are not taking into account history of the location names visited by the mobile user.

To account for location history, we form a vector of location names received by user’s phone in the past. When the user is stationary, the phone often receives multiple distinct location names as it can associate with different cell towers at different time instances. These location names sometimes may include locations which are far away in real world from user’s current location. However, the frequency of such location names is much smaller than frequency of location names which are in close proximity to the current location in real world. We hypothesize that this frequency difference is a factor of distance between Cell Tower and user. Therefore a weighted average based approach where the weights given to each landmark name is dependent on the frequency of number of received messages with the corresponding location name (in a given time window) will intuitively work well for improving the localization accuracy. We call this approach *FrequencyWeighted* in the following.

For a slow moving user, since the conditions are similar to a static user, the *FrequencyWeighted* approach should ideally provide better localization accuracy. However, a fast moving user will probably be in the range of a cell tower for a short duration and hence will receive a small number of (often only a single) CBS messages with the corresponding location name. However, it may happen that the currently received location name corresponds to a location in real world that is ahead on the path of the user while the previously received location name was behind on the path of the user (a typical case when the landmark name is received immediately on crossing the cell boundary). Therefore, weighted average of the received landmark names with higher weight given to those that are received most recently, will intuitively improve the localization accuracy. We call this approach *TimeWeighted* in the following.

A. TimeWeighted *Algorithm*

Assuming that CBS messages are received at more than a certain minimum rate, once every λ minutes, *TimeWeighted* algorithm considers all the received CBS messages in the past to calculate the current location of the user. In other words, whenever there is a long gap (more than λ minutes) in the reception, the algorithm forgets past history of messages and starts accumulating new history. The pseudocode of *TimeWeighted* algorithm is given in Algorithm 1. At the first location instance, the calculated location is same as the current geo-coordinates because there is no history available. Thereafter, the calculated location is the average of the current location and previously calculated location. As a result, the weight of previous location messages decreases exponentially with time.

B. FrequencyWeighted *Algorithm*

The algorithm takes a fixed time window in the immediate past, δ minutes, and looks at only the received messages in that time window. As discussed above, this algorithm caters to improved localization accuracy in the case of static or slow moving scenario. The *FrequencyWeighted* algorithm takes similar inputs as that of *TimeWeighted* algorithm, and a time window duration δ as described in Algorithm 2. The algorithm first extracts all the location geo-coordinates, in time window δ behind the current time, with their corresponding frequency of occurrence. It computes weighted average from the extracted location geo-coordinates to come up with calculated coordinates. Time window parameter δ needs to be tuned as a high value of δ could consider old location names and a low value could unnecessarily discard recent location names. We have developed a service for mobile phones to implement both of above algorithms. It receives all CBS location messages and stores them in a location vector. All these locations are geo-coded using Google Maps API. Whenever any application needs current location of the user, the service takes location vector as an input and returns calculated coordinates.

```
Algorithm: TimeWeighted
Input: Location Vector(LocVector) containing CBS location name,
        reception time stamp(ReceptionTime), GeoCoordinates[Lat,Lon]
        and a time out interval $\lambda$ minutes
Output: Approximate Location Coordinates ie
          EstimatedCoordinates[Lat,Lon]
begin
    Index=0;
    RunningCoordinates= LocVector[Index].Geo-Coordinates;
    Index = Index + 1;
    while Index $<$ LocVector.Size do
        TimeDifference = LocVector[Index].ReceptionTime -
        LocVector[Index-1].ReceptionTime;
        if TimeDifference $< \lambda$ then
            RunningCoordinates = (RunningCoordinates +
            LocVector[Index].Geo-Coordinates)/2;
        else
            RunningCoordinates= LocVector[Index].Geo-Coordinates;
        end
        Index = Index + 1;
    end
    EstimatedCoordinates = RunningCoordinates;
    return EstimatedCoordinates;
end
```

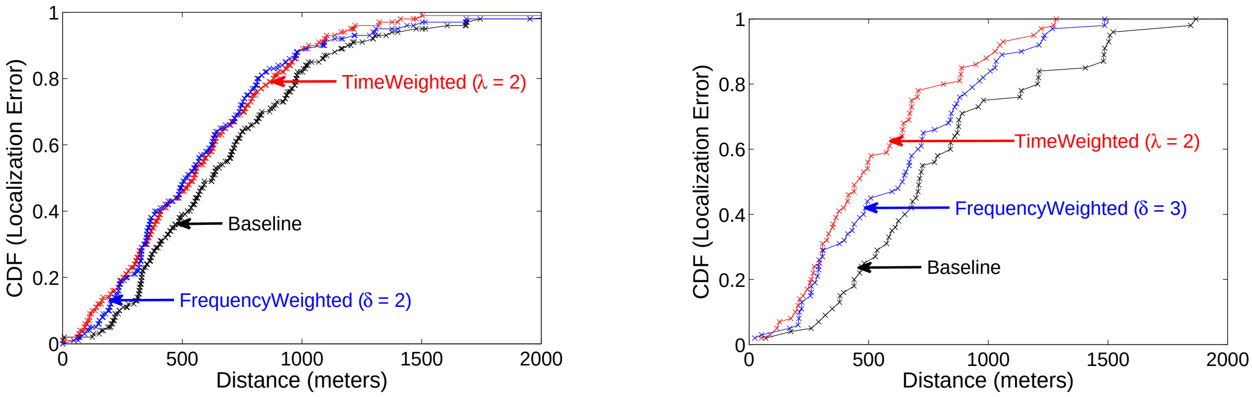
Algorithm 1: Pseudocode of *TimeWeighted* Algorithm

It is important to note that our approach (aimed for low-end phones) cannot assume any means, e.g. accelerometer or GPS, to measure the speed of the user and accordingly adapt the averaging policy for improved localization. We therefore compare the two approaches - *FrequencyWeighted* and *TimeWeighted* with the baseline approach empirically for cases with slow and fast user speed.

VI. Evaluation of The Algorithms' Accuracy

We now describe the empirical evaluation of the two algorithms, *FrequencyWeighted* and *TimeWeighted* , explained in the previous section, using our self collected real-world dataset. We used point-based localization approach as a baseline for comparison. The point-based localization approach estimates mobile user's location based only on the last received CBS message. This approach is identical to the one used by cell ID based localization approach, including service providers like Google as described in Section II-A. We use localization error as our evaluation metric which is distance between actual location (GPS Coordinates) and predicted location (CBS based approach). For simplicity purpose, we will discuss only one operator's result (referred to as operator Y). However, at the end of Section VI-A and Section VI-B, we also present results for operator X.

As hypothesized earlier, the accuracy of the algorithms could depend on the speed of travel. Hence, we collected traces for two different motions of walking and traveling. We define walking as movement at an average speed of 3.5 Km/h and traveling as movement at average speed of 30 Km/h.



(a) Comparison for traveling traces

(b) Comparison for walking traces

Fig. 6: CDF plots for *TimeWeighted* and *FrequencyWeighted* algorithms w.r.t to Baseline for operator Y

```
Algorithm:FrequencyWeighted
Input: Location Vector(LocVector) containing CBS location name,
       reception time stamp(ReceptionTime), GeoCoordinates[Lat,Lon],
       time window parameter $\delta$ and empty vector TWVector[Location,
       Geo-coordinates, Frequency]
Output: Approximate Location Coordinates ie
        EstimatedCoordinates[Lat,Lon]
begin
    Index=LocationVector.Size;
    EndTime = LocationVector[Index].ReceptionTime;
    StartTime = EndTime - $\delta$;
    Index = Index-1;
    while Index $> 0$ do
        ReceptionTime = LocationVector[Index].ReceptionTime;
        Location = LocationVector[Index].LocationName;
        if (ReceptionTime $>$ StartTime) AND (ReceptionTime $<$
        EndTime) then
            if Location is in TWVector then
                Update the Frequency;
            else
                Add Location with its Geo-Coordinates to TWVector;
            end
        end
        Index = Index - 1;
    end
    Compute weighted average on all the locations($L_1, L_2, ..., L_n$)
    with their frequency($f_1, f_2, ...., f_n$) present in TWVector;
    return EstimatedCoordinates;
end
```

Algorithm 2: Pseudocode of *FrequencyWeighted* Algorithm

A. Traveling Traces

Let us first analyze the effect of varying input parameters on the performance of two algorithms. For *TimeWeighted* algorithm, λ is a time-out parameter, which is necessary to forget old history. Empirically, we found optimum λ to be 2 minutes since it gave the least median localization error for all the traveling traces. We, therefore, have used λ as 2 minutes for evaluating the performance of *TimeWeighted* algorithm. For *FrequencyWeighted* algorithm, parameter δ is used to fix the time window within which it considers the received CBS messages to perform weighted average. Empirically, we found optimum δ to be 2 minutes for traveling traces since it gave the least median localization error for all of traveling traces. We, therefore, have used δ=2 for evaluating the performance of *FrequencyWeighted* algorithm.

Figure 6a compares the CDF of localization error for *TimeWeighted* and *FrequencyWeighted* algorithm with the baseline approach. Both the *TimeWeighted* and *FrequencyWeighted* algorithms perform consistently better than baseline. The improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 12% and 16% respectively, as shown in Table IV.

Let us discuss some intuition for performance of the two algorithms for traveling case. Typical rate of arrival of CBS message is 1 per minute. With λ fixed to 2 minutes, and average speed of traveling trace as 30 Km/h, if no CBS message is received for 2 minutes, the user has approximately moved by 1 Km from the location of previously received CBS message. It is therefore better for *TimeWeighted* algorithm to discard the history of CBS messages than to consider them for future calculation of localization. Similarly, with δ fixed to 2 minutes, *FrequencyWeighted* algorithm will only consider CBS messages received within a distance of 1 Km for calculation of localization, giving weights based on frequency of each CBS message received. This will mostly translate to average of two distinct CBS messages received in the 2 minute interval. Therefore, in case of traveling trace with correspondingly fixed parameter values, the two algorithms differ in that *FrequencyWeighted* algorithm never considers any CBS message outside the 2 minute window while the *TimeWeighted* algorithm gives any message outside the 2 minute window a small weight in case there is no time out in received rate of CBS messages. Soon after the time out, for the first 2 minutes, calculated localization for the two algorithms will be same.

Traces	**Baseline**	***TimeWeighted***	***FrequencyWeighted***
Travelling	621.40	549.82	521.52
Walking	712.94	462.54	644.85

TABLE IV: Operator Y : Median localization error comparison of *TimeWeighted* and *FrequencyWeighted* algorithms with baseline for walking and traveling traces

For operator X, both algorithms perform equally good as compared to baseline. The improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 10% and 11% respectively, as shown in

Traces	**Baseline**	***TimeWeighted***	***FrequencyWeighted***
Travelling	688.2	618.29	615.14
Walking	466.69	382.8	386.56

TABLE V: Operator X : Median localization error comparison of *TimeWeighted* and *FrequencyWeighted* algorithms with baseline for walking and traveling traces

Table V.

B. Walking Traces

In walking traces, there is no such instance where CBS location messages were not received for a significant amount of time. However, we still kept λ equal to 2 minutes for *TimeWeighted* algorithm to maintain uniformity across both traveling and walking traces. For the case of *FrequencyWeighted* algorithm, we again empirically calculated the most optimal value of δ that came out to be 3 minutes. Intuitively, higher value of δ, as compared to the case of traveling traces, is justified since longer history of CBS location messages will be useful due to lower speed.

Figure 6b show the CDF plot of *TimeWeighted* and *FrequencyWeighted* algorithm performance as compared to baseline for walking traces. As shown in Table IV, overall *TimeWeighted* and *FrequencyWeighted* give an accuracy improvement of approximately 35% and 10% respectively over the baseline approach.

Intuitively, we had hypothesized *FrequencyWeighted* algorithm to provide higher localization accuracy than *TimeWeighted* algorithm for walking traces (as also discussed in Section V). However, empirical study showed otherwise. By observing the collected data, we found out that the walking traces contained a lot of location names, that were farther located, 1200-1500 meters, from cellphone's actual location. This noise, particularly, gets added by the geo-coding service and presence of distant location names, which are among the challenges mentioned in Section IV. Effect of this noise can also be seen in terms of higher point-based localization error for walking traces (712.94 m) as compared to traveling traces (621.4 m).

Although *FrequencyWeighted* algorithm is hypothesized to have better accuracy for walking traces but, if the message containing distant location name is repeated within the δ time interval, it will have significant effect on the location computed by *FrequencyWeighted* algorithm (with fixed δ). On the other hand, for *TimeWeighted* algorithm, when such a CBS message with distant landmark name is received most recently, the calculated location get inaccurate. However, as the time progresses the weight of the CBS message with distant location is reduced and the corresponding inaccuracy in calculated location also reduces.

We also conclude that our initial assumption that fast and slow motion patterns would demand different approaches for improved localization was empirically found incorrect on our collected data. As shown here, *TimeWeighted* algorithm that was hypothesized to handle fast motion suffices for slow motion as well since it tolerates the noise added by the geocoding service for real data. However, we believe that the localization accuracy may vary across different environments. Therefore, an approach that can adapt based on accurate location input known intermittently from an oracle (in physical world through the GPS coordinates from intermittently turned on GPS or from a GPS enabled phone in close proximity) will reduce error in localization accuracy significantly.

For operator X, baseline accuracy was good due to good quality of landmarks. Improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 18% and 17% respectively, as shown in Table V.

C. Impact of Operator Heterogeneity on Accuracy

We observed that different operators provide different location names as well as with different time interval (broadcast cycle). We analyze the impact of operator heterogeneity on localization accuracy by collecting walking and traveling traces with two different phones, each having a operator X and Y. These traces were collected simultaneously and for the same geographic path described in Table II.

Table VI show the median localization error for the three different approaches across two different operators. Although the individual errors are different for each operator, we observe that *TimeWeighted* algorithm consistently performs better for both the operators. This empirically confirms with our finding that *TimeWeighted* algorithm is able to tolerate different broadcast cycle of operators and .

Algorithm	**Walking**		**Traveling**	
	X	Y	X	Y
Baseline	670.71	641.08	530.87	712.94
TimeWeighted	577.11	581.38	318.5	462.54
FrequencyWeighted	562.41	529.82	343.07	644.85

TABLE VI: Median localization error comparison of different algorithms

VII. Potential Applications of CBS-based Localization

There are wide range of location based applications which can be built using CBS messages. Hereby, we list some of them which we have already developed.

A. Activity classification

Activity recognition using mobile phones can enable wide range of context aware applications and there are already some efforts to do that using GSM based localization. We have noted in our data collection that CBS messages' rate of reception (message received per minute) is higher in walking traces than that in traveling as represented in Table VII. We have classified mobility with speed of about 3Km/hr as walking and about 30Km/hr as traveling. At an average, number of CBS messages (includes location names and advertisements) received per minute is higher than two in walking where as it is lower than two in traveling traces. Using CBS message reception rate as a measure, we were able to binary classification with 100% accuracy over a session. The session duration should be equal to or greater than five minutes of time. Accuracy of minute level activity classification was about 70%, which is due to unpredictable behavior of CBS message reception during traveling.

We believe that this kind of less granular activity classification could be easily used in applications like PEIR which just needs to know state of the user over some time intervals.

Activity	**Duration in Minute**	**CBS Msg /Minute**	**Location CBS /Minute**	**Avg Speed (km/h)**
Walking	124	2.40	1.77	3.14
Walking	47	2.06	1.04	4.10
Traveling	16	1.62	0.94	30.89
Traveling	25	1.64	0.85	31.78

TABLE VII: CBS reception rate comparison among traveling and walking traces

B. Location Sharing and Local Search

Growing ubiquity of location enabled smartphones prompted people to share current location with their friends. However, these servics are limited to mostly smartphones which uses GPS for getting current location and GPRS for communication. Using CBS-based localization system, we have built a location sharing service using Facebook. We have given two communication mediums i.e. SMS and GPRS. Since most of the people uses bulk SMS packs, it is preferred medium for many users to send their location to Facebook as well as query other friend's current location. We are also building an mobile application which can be used with twitter to publish location specific tweets which can used to build local trends.

Most of people in developing countries like India does not use digital maps for navigation and searching local businesses [11]. They usually take help of others to get an idea about directions from place A to place B which is mostly landmark oriented. We have built a local search application where current location is estimated by CBS location messages and it will fetch relevant entries from local business database in vicinity of current location. Mostly, there geographic distance is used as a metric for vicinity which requires all the locations in local business database to have geo-coordinates. Unfortunately, many places does not have rich listing of local businesses with geo-coordinates and it is mostly represented by landmark names. In those cases, simple string matching is done to find location names from the database. For an instance, for a local search query like ”Is there any Pizza Hut near Dwarka?”, we have to search for pizza hut in locations Dwarka as well as nearby locations to give richer results to user. From CBS location message, we build vicinity graph which can enable such queries.

VIII. Conclusion and Future Work

In this paper, we proposed CBS-based localization, which falls in the class of GSM-based approaches. However, our approach removes the necessity of war-driving or building a Cell ID database from GSM based localization. Hence, CBS-based localization is a promising solution for non-smartphones and provides them opportunity to access location based services without any extra infrastructure.The localization accuracy provided by a baseline solution is low due to geo-coding noise as well as CBS locations may be far from actual locations. Our algorithms *TimeWeighted* and *FrequencyWeighted* reduces impact of these errors by taking space time history. From empirical evaluation, we have found out that *TimeWeighted* can work for both walking and traveling traces across two different operators. Since most of low end phones does not have much processing power, our algorithms does not pose any special requirements at backend or phone client and can be easily deployed in real world. Also, proposed algorithms can work on enhancing accuracy of Cell ID based localization without any change.

We have already built some real world applications using CBS based localization and will provide APIs so that application developers can use it in their applications. In future, we plan to combine CBS based localization approach with GPS to reduce energy consumption by periodically sampling GPS. Also, we are building a model from the collected data and theoretically investigate whether using *TimeWeighted* is optimal.

References

[1] http://cell-broadcast.blogspot.com/2005/11/history-and-importance-of-cell.html

[2] http://www.gsmhelpdesk.nl/en/helpdesk/helpdesk.php?id=57

[3] Mohamed Ibrahim and Moustafa Youssef,A Hidden Markov Model for Localization Using Low-End GSM Cell Phones, http://arxiv.org/abs/1010.3411.

[4] Jeongyeup Paek, Joongheon Kim, Ramesh Govindan, Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones, MobiSys'10.

[5] GPS and Mobile Handsets. http://www.berginsight.com/ReportPDF/ProductSheet/bi-gps4-ps.pdf.

[6] Constandache, I., Choudhury, R. R., and Rhee, I. 2010. Towards mobile phone localization without war-driving. In Proceedings of the 29th Conference on information Communications (San Diego, California, USA, March 14 - 19, 2010). IEEE Press, Piscataway, NJ, 2321-2329.

[7] Robert P. Biuk-Aghai, GSM-Based Provider-Independent Positioning Method, Location Asia 2007, Hong Kong, China, 13-14 September 2007..

[8] Jain, S., Ghosh, R.K., Shyamsundar, R.K., Engineering location based pathfinding on Indian road networks over low end mobile phones, Communication Systems and Networks 2010, pp.1-9, Jan 5-9, 2010.

[9] Yadav, K., Naik, V., Singh, A., Singh, P., Kumaraguru, P., and Chandra, U. 2010. Challenges and novelties while using mobile phones as ICT devices for Indian masses: short paper, NSDR'10.

[10] Kuldeep Yadav, Vinayak Naik, Pushpendra Singh, and Amarjeet Singh. 2010. Alternative localization approach for mobile phones without GPS. In Middleware '10 Posters and Demos Track (Middleware Posters '10).

[11] Arun Kumar, Dipanjan Chakraborty, Himanshu Chauhan, Sheetal Agarwal, Nitendra Rajput, FOLKSOMAPS: Towards Community Driven Intelligent Maps for Developing Regions, in Third International Conference on Information and Communication Technologies and Development (ICTD), Doha, Qatar, April 2009.

[12] Yilin Zhao, Mobile phone location determination and its impact on intelligent transportation systems," , IEEE Transactions on Intelligent Transportation Systems , vol.1, no.1, pp.55-64, Mar 2000.

[13] Paek, J., Kim, J., and Govindan, R. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. Mobisys'10.

[14] Telecom Regulatory Authority of India : http://www.trai.gov.in/WriteReadData/trai/upload/ PressReleases/756/pr7sep.pdf

[15] Petteri Nurmi et al, A Grid-Based Algorithm for On-Device GSM Positioning, UbiComp 2010.

[16] Practical Metropolitan-Scale Positioning for GSM Phones, UbiComp 2006.

[17] Arvind Thiagarajan and Lenin S. Ravindranath and Hari Balakrishnan and Samuel Madden and Lewis Girod, Accurate, Low-Energy Trajectory Mapping for Mobile Devices, NSDI'11.