# Collaborative Localization of Mobile Users with Bluetooth: Caching and Synchronisation

Alexandre Barreira
Georgia Institute of Technology
abarreira3@gatech.edu

Philipp Sommer
Autonomous Systems Lab
CSIRO ICT Centre
philipp.sommer@csiro.au

Brano Kusy
Autonomous Systems Lab
CSIRO ICT Centre
brano.kusy@csiro.au

Raja Jurdak
Autonomous Systems Lab
CSIRO ICT Centre
raja.jurdak@csiro.au

## ABSTRACT

Location awareness is a key requirement for many pervasive applications. Collaborative localization can improve accuracy and coverage indoors and improve power consumption by duty-cycling GPS outdoors. We use Bluetooth for collaborative localization of mobile personal devices. Specifically, we embed information in Bluetooth device names to improve latency of information exchange between participating nodes. We identify and demonstrate on real hardware two problems in the Bluetooth stack that negatively impact localization accuracy: a) device name caching that introduces significant device-specific delays in transmitting information between nodes, and b) poor accuracy of time synchronization in modern mobile devices. Our solution is to append additional time information to the device name and track time offsets between nodes. We verify experimentally that this helps to both detect outliers and correct for time-synchronization errors and thus mitigate localization errors.

## Keywords

Localization, Bluetooth, Collaboration

## 1. INTRODUCTION

Precise location information serves as a basic building block for many pervasive computing systems enabling services tailored to the current position of mobile users. Location information outdoors is typically obtained from GPS modules on mobile devices, while indoor location information is delivered through specialized location tags that are attached to mobile users. Today's smartphones provide a compelling localization platform as they integrate GPS and multiple communication technologies capable of localizing users indoors (Bluetooth, Wi-Fi, etc). Bluetooth is commonly used for localization due to its pervasiveness in modern office environments [1, 5, 4]. These techniques typically embed location information in Bluetooth device names, which enables mobile devices to triangulate their location by inferring proximity to a number of infrastructure nodes. The reliance on infrastructure nodes, however, limits the utility of indoor Bluetooth localization in areas with sparse coverage of Bluetooth devices.

This paper proposes the use of Bluetooth device names for collaborative localization of mobile devices both indoors and outdoors. Collaborative Bluetooth localization can increase the density and coverage for indoor scenarios, where sharing location information between mobile devices enables devices further away from infrastructure nodes (laptop or desktop PCs) to more accurately determine their location. In outdoor scenarios, collaborative Bluetooth localization enables sharing of GPS location information among multiple smartphones. By splitting the energy burden of operating the GPS modules among multiple nodes, lifetime of individual nodes increases [2].

A key building block for collaborative localization using device names is to ensure that location information is shared between mobile devices in a timely manner. If the transmission or reception of location information is delayed, the mobile node might have changed its location significantly which would lead to inaccurate range estimates. Our experiments show that vendor-specific implementations of Bluetooth communication stack can create unpredictable delays in data reception shared through device names. These delays can be as high as tens of seconds which we conjecture to stem from vendor-specific caching implementations. We propose to embed the current time in device names to detect caching delays. This approach, however, requires all participating devices to be time-synchronized. We observe that modern mobile phones often exhibit time offsets in the order of tens of seconds. These offset persist even when phone are set to automatically synchronise to the cellular network time, as we have observed multi-second clock drifts among different network providers.

We design a simple algorithm for detecting and correcting both the caching delays and time-synchronization errors by learning and tracking pairwise clock offsets between neighboring nodes. Our algorithm works under the assumption

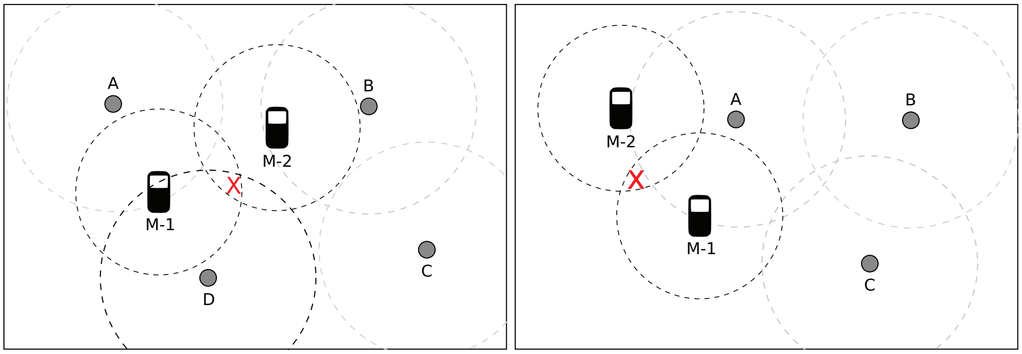


**Figure 1: Example of collaborative localization with infrastructure nodes (A,B,C,D) and mobile users (M-1,M-2). The estimated position of an unknown node is indicated by a cross: Mobile devices M-1 and M-2 improve location accuracy in a scenario with high density of fixed nodes (left). Mobile devices M-1 and M-2 extend the coverage area provided by the infrastructure nodes A, B and C in the low-density case (right).**

that the clock drifts are relatively small over short periods of time and that re-synchronization of devices is a relatively infrequent task. Given the accurate time information, we discover and discard stale location information due to device name caching. We evaluate our algorithm in experiments and show that it reduces the errors in device name timestamps to an average of one second, serving as an enabler for collaborative Bluetooth localization.

## 2. COLLABORATIVE LOCALIZATION

Collaborative localization algorithms rely on information provided by neighboring nodes which is used to improve the accuracy of localization, or to estimate positions in low density scenarios where no infrastructure is available (see Figure 1). The ability to communicate with other nodes allows us to restrict a node's location estimate to an area bounded by the communication range of its neighbors. Ranging techniques (e.g. time-of-flight or received signal strength) can be employed to further narrow down a node's position estimate. After a node has completed processing location information from its neighbors, it will update its position estimate which will then be announced to other nodes. In this position paper, we propose a generic framework for collaborative localization of mobile devices using Bluetooth. Our approach is not limited to a specific localization algorithm, but can be implemented using localization algorithms presented in previous work, e.g. [3] or [6].

### 2.1 Bluetooth based Localization

Most modern computers include a Bluetooth interface to enable wireless connectivity with peripheral devices. The primary way of communicating information over Bluetooth is to establish a connection between devices followed by the actual data exchange. However, the Bluetooth network stack needs to first discover the neighboring devices which leads to delays. As mobile devices might only be within their radio range for a short time, location information is commonly embedded in Bluetooth device names and shared in the discovery phase. Our approach augments the device name with an estimate of the current location (building-specific cartesian coordinates), an estimate of location uncertainty based on the freshness and the algorithm's confidence in the location estimate, and the corresponding timestamp. Nearby

**Algorithm 1** Bluetooth-based collaborative localization

```
loop
    now = getLocalTime()
    devices = bluetoothInquiry()
    for device in devices do
        [rssi, location, timestamp] = getInfo(device)
        syncError = now + getOffset(device) - timestamp
        if (abs(syncError)<threshold) then
            addNeighbor(rssi, location)
        end if
        updateOffset(device, now, timestamp)
    end for
    location = estimateLocation(devices)
    updateBluetoothName(location, timestamp)
end loop
```

devices can use this information together with the signal strength reading contained in Bluetooth packets to estimate their current position, which is then again reflected in the device name. A high level overview of our collaborative localization algorithm is shown in Algorithm 1. Next we give a brief overview of the Bluetooth stack.

**Bluetooth network stack.** During the inquiry phase a device switches between different frequencies in a pseudo-random manner to transmit and listen for the responses from nearby devices. This will result in a list of device addresses in the inquirer's neighborhood. In a second step, remote devices are asked for their device name, which can contain up to 248 characters. Remote device names can be obtained without the need to establish an explicit connection between the two devices. A device inquiry will also report the received signal strength indication (RSSI) and the device class of each found device.

**Bluetooth device name caching.** Disseminating information through device names is particularly sensitive to caching of remote device names in the communication stack at the receiver. A cached version of the remote device name will also contain stale location information, negatively impacting the localization accuracy. Unfortunately, smartphone operating systems provide little control over the un-

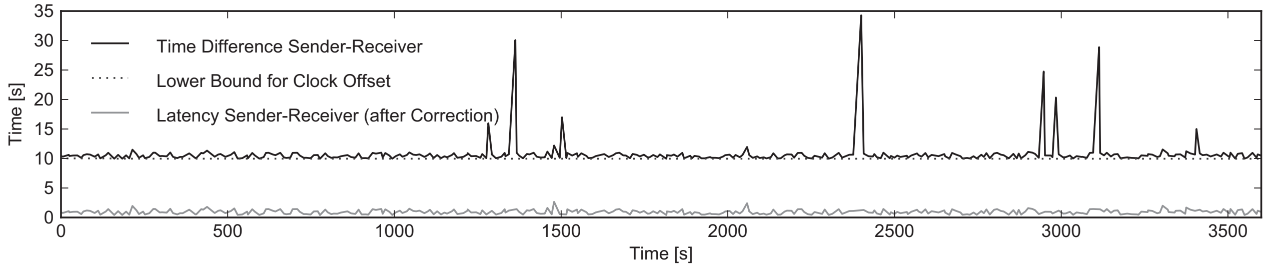


**Figure 2: Time errors observed in data dissemination through Bluetooth device names for the Samsung Nexus S smartphone. The offset of 10 seconds is due to time-synchronization errors. The occasional spikes up of up to 30 seconds are due to device name caching in the Bluetooth stack of Android.**

derlying Bluetooth protocol stack. In particular, a method to flush the cache of remote device names is not available, and therefore, we cannot guarantee that the contents of remote device names are up-to-date. Caching strategies also vary between different versions of the operating system and device models.

**Rejection of cached device names.** We include the current timestamp into the device name. This allows the receiver to estimate the time offset between the remote device and the local clock. Under the assumption that clocks of mobile phones remains stable over short time intervals, we can calculate a lower bound for the time offset. Consequently, a cached device name will result in a significantly larger time offset and can be discarded before passing it on to the localization algorithm. However, such an approach will not be able to mitigate the effect of small delays introduced by the Bluetooth transmission itself. Note that this simple clock offset estimation algorithm also corrects for global time-synchronization errors.

# 3. EVALUATION

We demonstrate the feasibility of our approach using a setup consisting of two Samsung Nexus S phones placed in close proximity to each other. Both phones are running version 2.3.3 of the Android operating system. Every phone continuously updates its Bluetooth device name once every second with the current local time. At the same time, each phone performs periodic Bluetooth device inquiries, which return the MAC addresses, device names, device classes and RSSI readings of nearby devices. For our evaluation, we use the local clocks of the devices which are only loosely synchronized exhibiting a clock offset of 9.5 seconds.

We measure the offset between the timestamp included in the remote device name and the local clock and plot results in Figure 2. The average offset is 11.2 seconds with a standard deviation of 3.2 seconds. At several instances, the measured offset was significantly larger than the average offset, indicating a cached version of the remote device name. We believe that this peaks in Figure 2 occur when the local device has detected a remote device during the inquiry phase but failed to query for an up-to-date device name. Thus, a cached version of the device name is returned to the application. By using a threshold decision, which is based on the minimal received offset, we are able to discard outdated location information included in remote device names before it is fed to the localization algorithm. The resulting average latency for propagating a device name after applying our correction algorithm is roughly 1.0 seconds with a standard deviation of 0.6 seconds.

# 4. CONCLUSIONS

In this position paper, we describe the use of Bluetooth device names to enable collaborative localization for mobile users. We demonstrated that large time delays can be introduced by local caching of remote device names. We developed a filter that can reject this stale data using timing information embedded in device names. Our experimental results showed that the time errors decrease drastically using our algorithm, thus providing accurate and timely location data which is mandatory for collaborative localization algorithms.

# 5. REFERENCES

[1] U. Bandara, M. Hasegawa, M. Inoue, H. Morikawa, and T. Aoyama. Design and implementation of a Bluetooth signal strength based location sensing system. In *Proc. IEEE Radio and Wireless Conference*, 2004.

[2] R. Jurdak, P. Corke, D. Dharman, and G. Salagnac. Adaptive GPS Duty Cycling and Radio Ranging for Energy-efficient Localization. In *Proc. SenSys*, 2010.

[3] X. Li. Collaborative Localization With Received-Signal Strength in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, 56(6):3807–3817, Nov. 2007.

[4] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran. Accurate Mobile Robot Localization in Indoor Environments using Bluetooth. In *Proc. ICRA*, 2010.

[5] M. Zaafir Barahim, M. Razvi Doomun, and N. Joomun. Low-Cost Bluetooth Mobile Positioning for Location-based Application. In *Proc. ICI*, 2007.

[6] P. Zhang and M. Martonosi. LOCALE: Collaborative Localization Estimation for Sparse Mobile Sensor Networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 195–206. IEEE, Apr. 2008.