A Platform for the Evaluation of Fingerprint Positioning Algorithms on Android Smartphones

C. Laoudias†, G. Constantinou‡, M. Constantinides‡, S. Nicolaou‡, D. Zeinalipour-Yazti‡ and C. G. Panayiotou†
†KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Email: {laoudias, christosp}@ucy.ac.cy
‡Department of Computer Science, University of Cyprus, Email: {gconst02, mconst02, nicolaou.silouanos, dzeina}@cs.ucy.ac.cy

***Abstract*—We present an Android smartphone platform for the evaluation of RSS fingerprint-based algorithms. Our system follows a mobile-based network-assisted architecture to eliminate the communication overhead and respect user privacy. When a user enters our university campus a smartphone client conducts a single communication with a dedicated server to receive the RSS radiomap and algorithm-specific parameters. The client is thereafter able to position itself independently. We have developed a RSS Logger application to facilitate the construction of the radiomap, which can then be downloaded from our Distribution Server. The positioning application is implemented on the mobile devices and we have integrated several fingerprint methods.**

I. Introduction

Nowadays, the vast majority of mobile devices, feature ubiquitous Internet connectivity through wireless or cellular networks, while according to recent statistics people tend to spend 80-90% of their time in indoors environments. These two facts have revitalized the interest in indoor location-aware applications that require alternative solutions for the provision of accurate and reliable location estimates because satellite-based positioning, e.g. GPS, is unavailable inside buildings. This has motivated the development of positioning algorithms that rely on existing WLAN infrastructure and exploit Received Signal Strength (RSS) fingerprints to determine location, owing to the wide availability of WLAN Access Points (AP) and the ease of collecting RSS samples without specialized equipment; see [1] for a detailed survey.

So far, the focus has been on improving accuracy and the proposed methods are usually evaluated only in terms of the accuracy in small-scale setups. However, the time required to estimate user location is equally important as it may violate the real-time constraints of an application. Moreover, low power consumption is another critical requirement to preserve valuable energy. Yet, both the estimation time and the battery depletion during positioning have received little attention, especially in the context of mobile devices. Our contribution is a mobile platform, which will be open to the research community, for the evaluation of fingerprint methods.

II. Android Positioning Platform

The prototype system is developed on Android smartphones and follows a mobile-based network-assisted architecture to eliminate the communication overhead and respect user privacy. For example, if a user enters our university campus, which is covered by tens of WLAN APs, he/she receives the RSS radiomap and a small set of parameters that are required for the functionality of the positioning methods. All methods are implemented entirely on the device, thus enabling the user to self-locate independently thereafter by using the radio map and the currently observed RSS fingerprint.

We have implemented several methods, such as the K-Nearest Neighbor (KNN) [2] and the probabilistic Maximum A Posteriori (MAP) [3] methods. Our platform additionally supports two state-of-the-art methods developed in-house, namely the Radial Basis Function (RBF) networks [4] and the Subtract on Negative Add on Positive (SNAP) [5] approaches.

A. RSS Logger Application

The *RSS Logger* shown in Fig. 1 (left) is an application developed around the Android RSS API for scanning and recording a user-defined number of samples (Fig. 1, middle) in specific locations at predefined intervals. These samples contain the MAC addresses and RSS levels (in dBm) of all neighboring APs, as well as the coordinates of the location where the user initiated the recording. The collected training data are stored locally in a log file (Fig. 1, right) and the user can contribute them to our positioning system.

The *RSS Logger* is used to collect RSS samples in outdoor and indoor environments. In the outdoor case, the ground truth location (Lat, Long) is obtained from GPS. In the indoor case, e.g. inside the campus buildings where GPS is unavailable, a floorplan map opens in the *RSS Logger* interface (with zooming capabilities) and the user may set the ground truth location in local coordinates (x, y) by clicking on the map.

B. Radiomap Distribution Server

Our server is mainly responsible for the construction and distribution of the RSS radiomap to the clients. To create the radiomap file, the server parses all available RSS log files and calculates the mean RSS value per MAC address by averaging over all samples collected at each distinct location. Finally, the averaged values are stored in the radiomap file so that each line corresponds to the mean RSS values at a specific location and each column to the respective MAC address.

Another interesting feature in our server implementation is a methodology for fine tuning algorithm-specific parameters. For instance, selecting an appropriate value for K in the KNN method is not trivial and may significantly affect the performance of the method. We have adopted an effective approach to handle this issue by using the *RSS Logger* to collect additional RSS samples for validation. Then, the KNN algorithm runs on the server side with different values of K and the appropriate value is selected in order to minimize the mean positioning error pertaining to the validation data. In this fashion, our server first builds the RSS radiomap and then selects appropriate values for the parameters.

C. Find Me Application

Our positioning application (Fig. 2) is a client that runs on Android smartphones and connects to the server in order to download the radiomap and algorithm-specific parameters using a text protocol (Fig. 2, left). The user may select any of the available methods in the configuration panel (Fig. 2, middle) and the client is then able to position itself independently.

For instance, the result of a positioning request initiated outdoors is illustrated in Fig. 2 (right); the estimated location (blue pin) is plotted on Google Maps along with the actual GPS location (yellow pin), while the red disc designates the positioning error. The user can further assess the accuracy achieved by returning back to the main screen where the positioning error is displayed (5.85m in this scenario). Alternatively, if the user resides inside a building then the estimated location is plotted on the floorplan map.

III. Demonstration Settings

For the demo at the conference venue we will use 5 HTC Desire smartphone devices running Android 2.1 (Eclair). These devices are equipped with a Qualcomm Snapdragon QSD 8250 1 GHz processor and provide 512 MB of Flash ROM, as well as 512 MB of RAM. The server will be running on an Asus eeePC netbook with Ubuntu Linux, while the HTC devices will connect to the server through the available WLAN hotspot. We will use a projector along with a display export utility to present the interactions on a smartphone directly on a wall, so that attendees will be able to follow the interactions.

A. Demonstration Scenarios

Interactive: First, we will hand out the 5 smartphones to selected participants and ask them to collect a few RSS samples inside the conference venue and contribute them to the system in order to experience the RSS Logger application (our team will have collected adequate samples before the demo to cover the whole area). Next, the participants may start positioning themselves using the currently observed RSS fingerprint by pressing the **Find Me** button (Online Mode). The estimated location will be displayed on the floorplan map of the venue. In this interactive scenario, our goal is to demonstrate the efficiency of our platform and the effectiveness of the underlying positioning algorithms in practice.

Trace-driven: We will also demonstrate how our platform can be utilized to evaluate different fingerprint positioning methods on smartphones under the same conditions. We have implemented an Offline Mode feature, thus offering the option to simulate a large number of successive positioning requests (i.e. a file with several RSS fingerprints collected at different locations can be stored on the flash media, as opposed to be collected in real time). This setting is rather useful for

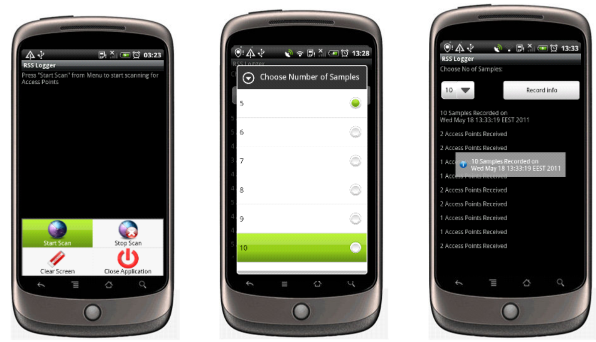

Fig. 1. Screenshots of the *RSS Logger* prototype application.

Fig. 2. Screenshots of the *Find Me* prototype application.

loading recorded test fingerprints and the conference attendees will have the opportunity to assess the performance of several methods in terms of the following criteria:

- **Execution Time:** Measure the average time required in practice to perform positioning on smartphones
- **Positioning Accuracy:** Obtain statistics (e.g. mean value and standard deviation) for the positioning error
- **Power Consumption:** Investigate the battery depletion by using the PowerTutor utility [6] to measure the actual power consumption on smartphones during positioning

Acknowledgment

This work is supported by the Cyprus Research Promotion Foundation and in part by the fifth author's Startup Grant, funded by the University of Cyprus between 2010-2011.

References

[1] M. Kjærgaard, "A taxonomy for radio location fingerprinting," in *LoCA*, 2007, pp. 139–156.
[2] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *IEEE INFOCOM*, 2000, pp. 775–784.
[3] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *ACM MobiSys*, 2005, pp. 205–218.
[4] C. Laoudias, P. Kemppi, and C. G. Panayiotou, "Localization using radial basis function networks and signal strength fingerprints in WLAN," in *IEEE GLOBECOM*, 2009, pp. 1–6.
[5] C. Laoudias, M. P. Michaelides, and C. G. Panayiotou, "Fault tolerant fingerprint-based positioning," in *IEEE ICC*, 2011.
[6] PowerTutor: A Power Monitor for Android-Based Mobile Platforms http://powertutor.org/.