WiFi Localization as a Network Service

João André Silva* and Maria João Nicolau** and António Duarte Costa*
*Departamento de Informática, Universidade do Minho, Braga, Portugal. Email: joaosilvafama@gmail.com, costa@di.uminho.pt
**Departamento de Sistemas de Informação, Universidade do Minho, Guimarães, Portugal. Email: joao@dsi.uminho.pt

***Abstract*—In last years, many efforts have been made to build indoor positioning systems using different types of signals and different types of position algorithms. Experiences with sound, light, infra-red, Bluetooth and popular IEEE 802.11 network signals have been carried out both by commercial and academic entities. The major goal is to fulfill the lack of GPS coverage inside buildings with a more precise and/or cheaper alternative. WiFi based Localization systems gain special attention in this context due to its huge usage inside all modern facilities.**

Many algorithms have been proposed to estimate the device position within the WiFi network using the signal strengths (RSS) obtained by 802.11 device drivers. RSS values can be registered by each packet received, either by the the user's device or by the network's Access Points. Values can then be compared with a pre-configured map of signal strengths obtained in a calibration (offline) phase.

This paper presents a localization system, where the location is determined exclusively by the network infrastructure. The location computation is based on the signal strength measured at Access Points and sent to a localization server. The localization server determines the location, stores it in a database and accepts requests from authenticated clients. Results obtained in experimental tests are discussed and compared with those published by other authors, allowing us to conclude that the system combines a good accuracy, with the advantage of not requiring any intervention from clients in the localization process. The entire system is designed in order to make locations available as a network service.

I. Introduction

The number of mobile devices is increasing every day. Nowadays almost everyone carries a mobile phone or laptop everywhere, not only to be contactable but also to be connected to the Internet. This led to emergence of new applications and services that take advantage of location information of users. Hence it is relevant to find more efficient ways of tracking devices. The localization accuracy depends on the system used to locate the mobile devices. A system like GPS, offers good results in external environments but it cannot be used for indoor localization. In this case, 802.11 wireless networks may be used.

Tracking devices in 802.11 wireless networks has been subject of research in recent times and has increasing importance due to the proliferation of WiFi networks. A large number of systems using WiFi networks to track mobile devices are implemented in hospitals, offices or warehouses to control stocks. In most of these localization systems, the location is determined by the mobile devices themselves without any intervention of the network infrastructure. However, there may be important advantages if the location is determined by the network infrastructure. The network administrator controls the infrastructure and may prevent any attempt to corrupt the system and react faster and efficiently to any problem that may occur. In addition, it is not necessary to do any change in the mobile device (client side). But greater advantage comes from having location info available as a basic network service. Applications running on any device can query the localization service as they query any other directory service. Device position is by default private information available only to the applications on that device, but can be easily shared with others according to some established access policy.

This paper describes a WiFi based localization system designed to provide locations as a network service. Location is computed by a Localization Server, based on RSS values permanently monitored by the infrastructure Access Points on all packets. Values are sent to a Capture Server that stores them in a online database. Upon request, the Localization Server can compute location of any device at a given time, by comparing the values in the online database with a map of signal strengths done in a calibration (offline) phase. The paper is structured as follows. Next section presents a very short overview of related work. In section III the global system architecture is presented emphasizing how location can be determined exclusively by the infrastructure. The implementation is described in section IV, detailing both the online and offline phases. A few algorithm variants are presented in section V. Results discussed in section VI show that the system has an accuracy comparable to other WiFi Localization Systems, without requiring client intervention.

II. Related Work

RADAR [1] was the first WiFi localization system based on RSS fingerprinting. The entire system was designed based on the assumption that signal strength values measured at the receiving devices can be used to infer their position. Authors first show that signal strength is higher when the receiver is closer to the the transmitter. Then they propose a system that works on two phases. In the first phase, called the offline phase, RSS values are measured at each point of a given set of locations distributed inside the localization area. The information collected in this phase (RSS fingerprint map) is afterwards used in the online phase to estimate a device location. In the online phase, RSS values obtaining by the receiver are compared with those previously observed in the offline phase. The system then uses deterministic methods to infer the device's position.

HERECAST [4] uses a different concept of location. Instead of a precise location using coordinates in a spacial plan, the system presents the location to the users using only common words. The location is obtained using a database of known WiFi Access Points, continuously updated with information provided by the users themselves. For each AP, the database stores the SSID string and the AP location, in human readable language: the place, building, etc. where the AP is located. For a given request, the system searches the AP associated with the user's device in the database and shows its correspondent location. The location refers only to an area matching signal coverage range of the AP.

ARIADNE [3] is similar to RADAR in several ways. Location is given in numerical coordinates on a plant and, like RADAR, the system has two phases: the offline phase and the online phase. ARIADNE however improves the offline phase and reduces the huge amount of work that is usually necessary to create a complete RSS map. Instead of measuring several times, with different conditions, at each point, ARIADNE uses only a single measure. The complete RSS map is then derived using an algorithm that runs on a model of the building. Parameters are obtained by ray tracing and simulation. Since different objects, built on different materials, attenuate signals in different ways, signatures derived by simulation may contain errors. However, in many buildings, collecting all values is not only unpractical, but also sometimes unfeasible.

HORUS [5] makes use of probabilistic models to improve localization accuracy. It is also a WiFi based localization system, that runs on the client's device, and uses RSS information. In order to apply probabilistic techniques, authors store the RSS signature values with information about their distribution.

III. WiFi Localization System

As stated before, in the proposed system, the processing and all the decisions are made by the network infrastructure. The client has a passive role. To be located, the client only needs to send some information to the network, so it can be captured by the APs (Access Points). The APs are responsible for going through the channels in which the network operates capturing all the packets they can get. For privacy reasons, no information is processed besides source MAC address and signal strength. The signal strength found in each packet is extracted and stored in the Capture Database. This phase is usually called the *online* phase.

The system includes a calibration phase which is called *offline* phase. At this phase, a single client runs through the indoor area in order to collect samples of the signal strength at each position. The information collected in this phase is also stored in the *Capture Database*. The *Localization Server* implements a localization algorithm and uses the information stored in the *Capture Database* to determine the location of each device. Typically, the localization algorithm tries to find the minimum distance between the values sent by devices in the *online* phase and the values collected in the *offline* phase. This strategy is usually called *Location Patterning* or *Fingerprinting*.

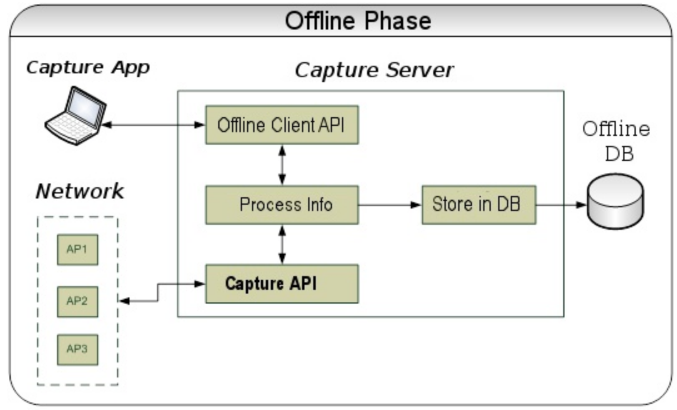

Fig. 1. Offline Phase

IV. Implementation

To evaluate the proposed localization system, a prototype was implemented using three Asus Routers with DD-WRT firmware [2]. C language was used to implement packets capture in routers to save, as far as possible, routers resources. All other components of *Localization System* were implemented using JAVA language. The Capture database was implemented using MySQL.

A. Offline Phase

The offline phase consists mainly in creating a map of signatures based on the signal strength registered in each position defined for the testing area. It was chosen one big laboratory of our department as the testing area ($14m \times 7m$). The three ASUS routers were placed so that in any point of the testing area it is possible to receive signal in good conditions from the three routers. Then, a grid was drawn over the plant of the testing area forming, for this particular case, 13 measuring points. At each point of the map is necessary to collect samples in four different directions: North, South, East and West, which makes 13×4 (52) points. It should be noted that it was necessary to collect a minimum of 50 samples per point in order to calculate a mean for the signal strength value.

To accomplish the offline phase, a collection system was implemented. This system includes four components as shown in Figure 1: the capture application, the capture server, the routers and the capture database (offline part).

The client application was implemented using JAVA language. It is a very simple application intended to be used in a mobile device with network connectivity. At each grid point, the user must introduce the position (on xx axis and on yy axis) and the direction of the mobile device. The client application sends these data together with information about its MAC address to the capture server, so that the capture can be initiated.

The capture server receives this information and sends the MAC address of the mobile device running the capture application, to the routers. There is an application running at the routers, which receives this MAC address and starts the capture. During one minute, each router will capture

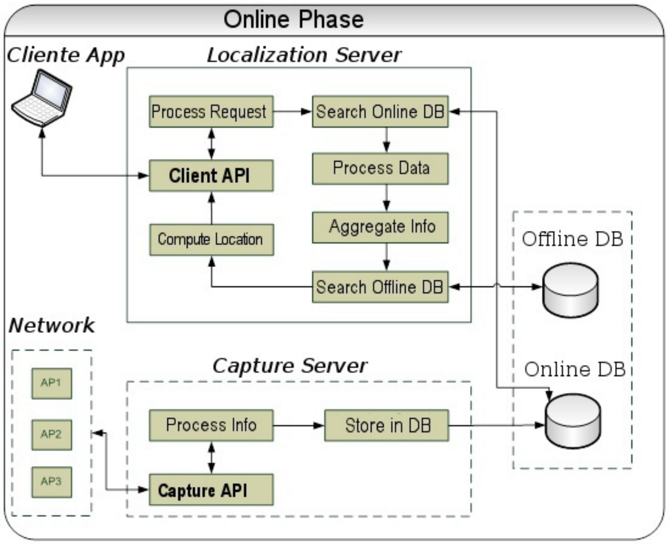

Fig. 2. Online Phase

all the packets sent by this MAC address to the network, extracting the signal strength from the packets header. Ending the capture, the routers send the collected information to the Capture Server. The Capture Server receives and processes this information and stores it in the capture database (offline part).

The database keeps all the samples captured in order to increase the accuracy of the Localization System. A minimum of 50 samples per collect point is needed to be considered a valid measure. After the collection and storage of captured data, it is necessary to summarize the table of data collection, so that each point has a value identifying its location for each access point. Each location in the summary table has a value of signal strength for each access point (RSSI1, RSSI2, RSSI3). Actually, the samples are summarized simply by calculating their average, however a more sophisticated process may be applied in the future. The summary values are then used in the online phase to make the correlation between the average collected in real time and average that was gathered during offline phase.

B. Online Phase

For the development of this phase was necessary to have an application in the routers to capture as much traffic as possible, ie, contrary to what happened in the previous phase, the routers must capture all traffic and not just the packets sent by a specific mobile device. Only non-sensitive packet header fields (MAC address and signal strength) are processed. The Capture Server and the Client Application were also adapted to the requirements of the online phase. A new server called Localization Server that calculates the location of each mobile device was implemented. It uses an algorithm to compare the data collected with the data on the map of signatures created in the offline phase.

The online phase system includes five components as shown in Figure 2: the Client Application, the Capture Server, the Routers, the Database and the Localization Server. The online phase starts the localization process with the routers sending the captured information to the Capture Server of the online phase. The Captures Server in turn, treats the received information and stores it in the Captures Database (online part). The Localization Server receives from the Client Application, the time interval for the process of calculating the location and the MAC address of the target device. After receiving this data the Localization Server initiates the localization process. It asks the online database for information about the signal strength of the target device, and makes an average of all the samples taken. After obtaining the average of signal strengths, the localization server will seek the average of each point in the Offline Database and compares them using a particular algorithm. Throughout this study, multiple algorithms were tried in order to improve the results obtained.

TABLE I
ALGORITHM VARIANTS IMPLEMENTED

	Distance between online (v) and offline (o) vectors
A1	$d(v,o) = \|v_1 - o_1\| + \|v_2 - o_2\| + ... + \|v_n - o_n\|$
A2	$d(v,o) = \sqrt{(v_1 - o_1)^2 + (v_2 - o_2)^2 + ... + (v_n - o_n)^2}$
A3	$d(v,o) = \sqrt{(k_1 \times D_1 + k_2 \times D_2 + ... + k_n \times D_n) \times 100}$ where $D_i = (v_i - o_i)^2$ and $\sum_{i=1}^{n} k_i = 1$ with $k_i \in [0..1]$ (*higher values of D ⇒ **lower** values of K*)
A4	Use A3 formula with different weights. (*higher values of D ⇒ **higher** values of K*)
A5	$d(v,o) = \sqrt{(k_1 \times D_1 + k_2 \times D_2 + ... + k_n \times D_n) \times 100}$ where $D_i = \|v_i - o_i\|$ and $\sum_{i=1}^{n} k_i = 1$ with $k_i \in [0..1]$ *higher values of D ⇒ **lower** values of K*)
A6	Use A5 formula with different weights. (*higher values of D ⇒ **higher** values of K*)

V. ALGORITHMS USED

At first, a major algorithm that mimics the one used by RADAR, was implemented. This algorithm works by computing the distance between the vector of RSS values obtained in online phase $V = (v_{ap1}, v_{ap2}, ..., v_{apn})$ with all RSS vectors stored in offline phase. The result is a sorted list of points, starting with the best match, that is the best candidate location to present to the user. The best match corresponds to the minimum distance found.

A few variants of the base algorithm were also implemented, using different distance formulas. Table I shows a set of variants implemented, including the euclidean distance (A2) used in RADAR. Weight factors were also used. Variants A3 and A5 assume that the nearest APs give more stable RSS values and hence are weighted with higher coefficients. On the contrary, variants A4 and A6 assume that nearest AP values are less discriminatory and are weighted with lower coefficients.

The base algorithm was also modified to give not only the best point (the first in results list) but also the coordinates of the midpoint for the first K results (assuming that k-first results are the k-nearest positions of the actual location).

VI. RESULTS

In the evaluation online phase, 13 physical locations were selected. At each online location, tests were repeated 30 times,

TABLE II
GLOBAL RESULTS CONSIDERING THE NEAREST POINT ONLY

	Average Error Distance (m)	No. of Exact Matches	Max Error (m)
A1	3.471	40	9.144
A2	3.590	37	9.144
A3	3.538	39	9.144
A4	3.557	38	9.144
A5	3.571	35	9.144
A6	3.481	38	9.144

TABLE III
GLOBAL RESULTS CONSIDERING THE MIDPOINT OF THE 7-NEAREST

	Average Error Distance (m)	No. Exact Macthes	No. Err $< 0.5m$	No. Err $< 1.0m$	Max Error (m)
A1'	2.807	0	2	24	6.561
A2'	2.737	0	9	27	6.451
A3'	2.756	0	10	30	6.451
A4'	2.712	0	10	32	6.419
A5'	2.780	0	4	33	6.561
A6'	2.751	0	7	30	6.451

resulting in an overall of 390 tests. For each test all algorithm variants were run and results compared.

Table II shows the results when all algorithms return only the nearest point. This table includes the average error distance (in meters), the no. of exact matches returned and also the maximum error. The error distance is the euclidean distance between the real location tested and the estimated location returned by the algorithm. While average error distances are very similar, better values were observed with A1 variant that uses a Manhattan distance formula, and the worst with A2 that uses Euclidean distance. Maximum error is the same in all cases.

Table III shows the results obtained when all algorithms return the midpoint of the 7-nearest results instead of the nearest point only. The value of 7 was found by testing all possible K-nearest possibilities. The average error distance is lower in all cases, but now A1 shows the worst behavior. Best results were achieved by A4. Maximum error is also lower in all variants. In this case there are no exact matches, but the number of observations with error below $1.0m$ was included, showing that the number of values near a match is in the same range of the exact match results of the nearest point approach.

Figure 3 shows the cumulative distribution function (CDF) of the error distance for the best case algorithm variant A4. Lines show the results for the nearest point and the midpoint approaches. In the first case, about 10% of all observations are exact matches and the maximum error is above $9m$. In 80% of the cases the error is below 5 meters. Results improve significantly in the second case. No exact matches were found, but 10% results have error below 1 meter. The maximum error is near $6.5m$. 70% of the result are below $3m$ and most values are between 2 and 3 meters.

VII. CONCLUSIONS AND FUTURE WORK

A WiFi based localization system was designed and implemented from scratch in order to make locations available as a network service. Location is determined exclusively by the network infrastructure, without requiring or demanding collaboration from the users. In this way, application clients may issue location requests to a localization server, using an appropriate application interface (API). Location replies are given based on pre-established access control policies. A client may have access to its own position and allow/deny accesses to other entities. Privacy and other security concerns are addressed by applying authorization and authentication methods to this service.

The system was evaluated and shows an accuracy that is comparable to other WiFi based localization systems. Average distance error is in 2 to 3 meters, which is equivalent to the values observed in RADAR and ARIADNE. The system can be expanded to include new algorithms. The offline phase needs improvements in order to expand the localization service to the entire building. Demo applications must also be made available to users. It was observed that in some points of the localization area the RSS variations are higher, and this variation must be measured and taken into consideration.

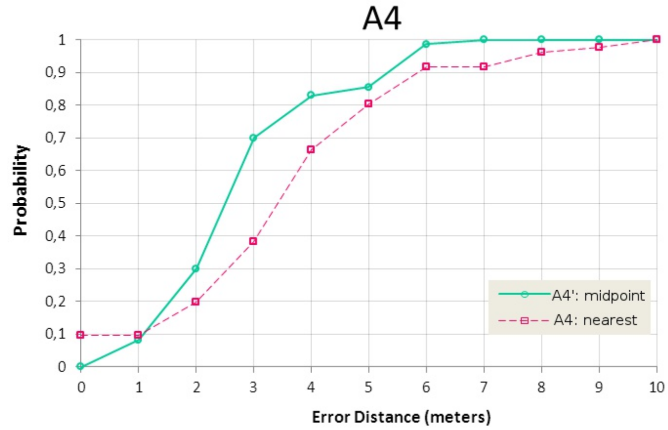

Fig. 3. CDF of the error distance for A4 variant (nearest point vs. 7-nearest midpoint)

REFERENCES

[1] P Bahl and V. N. Padmanabhan, “Radar: an in-building rf-based user location and tracking system,” *Proceedings IEEE INFOCOM 2000*, vol. 2, no. c, pp. 775–784, 2000.

[2] DD-WRT Community. (2011). “What is DD-WRT?”, [Online]. Available: http://www.dd-wrt.com/wiki/index.php/What_is_DD-WRT? (visited on 05/30/2011).

[3] Y. Ji and S. Biaz, “Ariadne: a dynamic indoor signal map construction and localization system,” *Proceedings of MobiSys ’06*, pp. 151–164, 2006.

[4] M. Paciga and H. Lutfiyya, “Herecast: an open infrastructure for location-based services using wi-fi,” *Proceedings of WiMob2005*, vol. 4, pp. 21 –28, 2005.

[5] M. Youssef and A. Agrawala, “The horus location determination system,” *Wireless Networks*, vol. 14, no. 3, pp. 357–374, 2008, ISSN: 10220038.