Π10: List of Publications

Journal papers


Conference papers


Workshop papers


Part one: The Statistical Terminal Assisted Mobile Positioning methodology and architecture

C. Laoudias a,*, C.G. Panayiotou a, C. Desiniotis b, J.G. Markoulidakis b, J. Pajunen c, S. Nousiainen c

a University of Cyprus, Department of Electrical and Computer Engineering, 75, Kallipoleos Street, P.O. Box 20537, 1678 Nicosia, Cyprus
b Vodafone-Panafon (Greece), Technology Strategic Planning – R&D Department Tzavella J-3, Halandri, 152 31 Athens, Greece
c VTT Technical Research Center of Finland, VTT Information Technology, P.O. Box 1000, FIN-02044 VTT, Finland

Available online 5 February 2008

Abstract

Statistical Terminal Assisted Mobile Positioning (STAMP) is a methodology that improves the accuracy of existing positioning techniques by exploiting measurements collected at the terminal side. STAMP is setting a unified positioning framework, in which different types of raw network related measurements are employed by multiple positioning techniques in order to derive coarse position estimates. Subsequently, statistical processing is performed to further increase accuracy. STAMP is complementary to satellite positioning systems, while the proposed architecture is highly applicable to User Plane location architectures. Due to its open and modular architecture, new positioning algorithms and post processing techniques can be added in the localization chain, thus supporting effectively a wide variety of Location Based Services (LBS). The implementation of STAMP in a prototype focuses particularly on Quality of Position issues and compatibility with currently available and up-coming standards and communication protocols. Preliminary results using actual network measurements, in both Greece and Finland, reveal the efficiency of STAMP.

© 2008 Elsevier B.V. All rights reserved.

Keywords: Location Based Services; User Plane architecture; Mobile positioning; Statistical processing

1. Introduction

Location Based Services (LBS) enable the provision of enhanced personalized services to the mobile user through the identification of the user’s current position. During the late 90s LBS did not get widely accepted for many reasons such as the lack of standards, low quality content, customer perception issues and inadequate positioning performance. However, nowadays most obstacles have been overcome and the market has become more mature to accept the advent of advanced location oriented applications. This is also the result of the deployment of next generation wireless networks and standardization activities carried out by the 3rd Generation Partnership Project (3GPP) and Open Mobile Alliance (OMA).

A wide variety of positioning techniques has been proposed so far, each one presenting certain advantages, as well as drawbacks. There is a trade off between the accuracy achieved and the initial investment required on the network side, while the support of legacy terminals is also an issue. Therefore, all positioning technology roadmaps begin with low cost and low accuracy techniques, e.g. Cell Identity based and evolve in the long term towards more advanced, accurate and reliable techniques, such as A-GPS. However, A-GPS terminals are still expensive, with limited market penetration and there will be a relatively long period of time for which commercial GSM or GSM/UMTS devices will not be equipped with GPS receivers. Furthermore, many LBS may not even require the high accuracy provided by GPS. Even if GPS equipped
Terminals proliferate in the near future, satellite based techniques will still suffer from time-to-first-fix delay, excessive battery consumption and accuracy degradation related to satellite visibility problems occurring indoors and in urban canyon conditions. Therefore, other mobile positioning solutions with high degree of applicability are in demand to complement satellite positioning systems. Such network-based-terminal-assisted positioning methods are often considered as short term or fallback solutions in hybrid schemes based on A-GPS. The crucial requirement for these methods refers to the ability of direct deployment in commercial cellular networks, without the need for device replacement or hardware modifications at the network side.

STAMP is an advanced localization methodology that is generic and applicable to legacy cellular networks and Beyond 3G (B3G) heterogeneous radio access environments [1], where multiple wireless technologies coexist (e.g. GSM/GPRS, UMTS, WLAN, WiMAX, etc.). The main idea behind STAMP is the utilization of measurements from all available access networks that the Mobile Terminal (MT) performs periodically while in idle mode [2]. An agent installed on the MT captures and stores location related information, such as Received Signal Strength (RSS) values and timing measurements from the primary and all neighboring cells, available during normal operation. At the LBS application initiation the list of measurements is uploaded to the Positioning Server and then exploited through multiple positioning techniques, in order to provide estimates of the history of the terminal’s motion. As a final step, statistical processing with the aid of Kalman filtering provides a more accurate estimate of the current MT position. At the same time tracking capabilities and velocity estimation are also provided. Knowledge of the MT’s speed and direction enables the provision of advanced LBS. STAMP can be considered both as a stand-alone solution or combined with A-GPS in a hybrid scheme.

There are two architectural approaches for the deployment of LBS, namely the Control and User Plane. Control Plane architectures exploit the existing circuit-switched network infrastructure and signaling layer to support the exchange of position related information between the terminal and the network. These architectures were originally designed to enable the provision of voice-centric LBS applications and to support emergency calls [3]. In this case, the positioning process is completely controlled by network operators, who provide the location data to third parties through the central Gateway Mobile Location Center (GMLC), [4]. On the other hand, User Plane location architectures, based on the emerging OMA Secure User Plane Location (SUPL) standards [5], allow the transfer of location related information (e.g. network measurements), through data packets over secure IP connections, independently of the underlying access technology and network infrastructure. Thus, the deployment of LBS by third parties is facilitated and the necessary network modifications to support positioning are minimized. However, an agent is required at the terminal side, that handles the communication exchange. Both architectures may be equally applicable depending on the application type, the profile of the end-user, the initiator of the positioning request (network or terminal) and the terminal capabilities [3,6]. It should be noted that the latest version of OMA User Plane Location Protocol (v.2.0), supports the transfer of multiple historical position related information, thus facilitating the use of STAMP method as the positioning platform in User Plane architectures.

The IST-MOTIVE project [7] provided the opportunity to gain more insight on the practical aspects of implementing STAMP, as part of the Ubiquitous Terminal Assisted Positioning (UTAP) system. The project aims to deliver a positioning system prototype that will serve as a complete platform to enable LBS deployment. In this platform additional positioning techniques can be easily integrated as separate components, while Quality of Positioning (QoP) may be assured based on the application requirements and the specific technique selected to perform localization. QoP is treated as a set of attributes, such as the positioning error and the system response time, associated with a request for the MT’s geographic position. Moreover, position and location management is provided by caching past position estimates, in order to avoid unnecessary calculations. This article provides a detailed description of STAMP methodology as part of the UTAP system, presenting the proposed architecture, individual components and technical details of this positioning framework.

The remainder of this article is structured as follows. Section 2 outlines some well-known positioning algorithms and state of the art localization techniques. In Section 3, the STAMP concept is introduced and some important factors that affect the performance are highlighted. The platform requirements, followed by the proposed system architecture, are detailed in Section 4. Prototype implementation details are presented in Section 5, while Section 6 demonstrates the application of STAMP in some indicative positioning scenarios, based on actual network measurements. Finally, Section 7 provides some concluding remarks and discusses future work related to the UTAP prototype.

2. Overview of mobile positioning techniques

One way to categorize mobile positioning techniques is according to the measurements employed and the underlying processing algorithm. In the first category, localization is achieved by the Angle of Arrival (AOA) of a signal from the terminal to several Base Stations (BS). This method performs best in rural areas, where Line-of-Sight (LOS) paths between the MT and the BSs are prevalent. Furthermore, it requires the use of sophisticated antenna arrays and suffers from multipath propagation due to reflections [8]. The second category comprises techniques that rely on timing information, namely the Time of Arrival
(TOA). TOA information is available in GSM through the Timing Advance (TA) [9] and in UMTS through the Round Trip Time (RTT) parameter [10]. These parameters are mainly used in order to enhance the performance of the pure Cell identity technique. The Uplink TOA (UL-TOA) which is a standardized method for GSM [11], uses an access burst as the measured signal, which is generated by forcing the mobile to perform an asynchronous handover. TOA techniques determine the location of the MT as the intersection of circles derived from TOA measurements from at least three BSs [12]. Timing measurements can also be used by the MT to calculate differences between the signals sent by the nearby BSs, known as Time Difference of Arrival (TDOA) measurements. A TDOA value defines a hyperbola with the two BSs as the foci and at least three hyperbolae are required to uniquely identify the MT position [13]. This technique is also standardized in GSM [11,14] named Enhanced Observed Time Difference (E-OTD) and in UMTS known as Observed Time Difference of Arrival (OTDOA) [15,16].

The final category utilizes RSS measurements, collected periodically as part of the MT’s standard functionality to assist in the handover process. There are two approaches followed in this category. In the first one, a proper propagation model is used to translate RSS values to distances from the respective neighboring BSs. Then through standard trilateration techniques position is determined. The propagation model can be empirical, semi-empirical or statistical and also compensate for the directionality of the antenna’s radiation pattern [17–19]. Moreover, a method presented in [20] does not require a known and accurate propagation model in order to perform positioning. The second approach, known as fingerprint technique, employs a database of fingerprints measured a priori inside the area where the MT is to be located. Each fingerprint contains signal information from the BSs or WLAN Access Points (AP) detected at certain locations. Positioning is performed by matching the signal information of the request fingerprint, received by the MT, to the signal information of the reference fingerprints. The signal information usually refers to RSS measurements from neighboring GSM cells [21]. However, Power Delay Profiles [22] or Received Signal Code Power (RSCP) values [23] of the detected UMTS cells can also be employed. In [24] measurements from multiple networks were combined in order to form the fingerprints. To overcome the cumbersome task of collecting the fingerprints on site, the output from network planning tools has been used in [25] to artificially generate the required fingerprints.

The aforementioned algorithms perform positioning in a static manner, i.e. they make no assumptions regarding the MT’s motion and dynamics. Time processing of noisy measurements, in the form of filtering, can alleviate the positioning errors and increase accuracy by incorporating proper mobility models. Filtering, including the well-studied Kalman filters [26], commonly used in satellite based positioning, can be applied in the raw measurements or in the coarse position estimates domain. In the case of raw measurements, the performance of the underlying positioning algorithm is greatly improved by estimating the system’s state, i.e. the MT’s motion, from noisy observations. Different variants of Bayesian filters [27] have been discussed in the literature, including Kalman Filter [28] and in this issue, Extended Kalman Filter (EKF) [30,31] and Particle Filter [32].

In the case of coarse position estimates derived from any positioning technique, filtering is employed in order to eliminate high positioning errors that are not corresponding to the MT’s dynamics. These are also known as post processing techniques that generate a smoothed location sequence to reflect the MT’s mobility pattern more accurately. Popular post processing techniques include Kalman filters [33–36] and Particle filters [37]. Map matching is also employed as a post processing technique to increase the accuracy by matching the MT’s trajectory to a set of candidate road segments of a digital road map or indoor floor plan [38,39]. In [24] map matching has been used as the final processing step following Kalman filtering.

3. The STAMP methodology

In this section, the STAMP concept is presented, followed by an analysis of the factors that influence the performance of this positioning methodology, in terms of accuracy, storing requirements and communication overhead.

3.1. Overview of STAMP concept

The STAMP concept is illustrated in Fig. 1 for a MT moving through an area covered by multiple radio access networks. While in idle mode the MT periodically stores all available network measurements, thus forming a list of location related information. Each entry in this list, denoted
as the STAMP List, contains the actual measurements and a special field, which indicates the corresponding type of access technology. In this article we consider one type of network measurements, namely, GSM, UMTS and WLAN RSS values. However, this can be extended in order to include all available measurements required by other positioning techniques. Each entry in the STAMP List is also time-stamped and in this way the STAMP List is actually a record of historical information reflecting the MT’s motion. When the STAMP List is full, updating is performed in a sliding window fashion by discarding the oldest and incorporating the current measurement. During standard MT operation in idle mode, a set of parameters is constantly monitored [40,41] in order to support the Network Selection, Cell Selection and Reselection functions. These parameters include the Cell Identity and RSS values from the primary and a set of neighboring cells in GSM/UMTS networks. In a WLAN environment the AP Identity and RSS values are also monitored to ensure optimum performance. Moreover, MTs supporting both GSM and UMTS connectivity, can decide to monitor only the first, the second or both networks according to certain radio coverage conditions. In the future, multi-homed terminals will have the ability to be simultaneously attached to several wireless access networks. Since these monitoring procedures are part of the MT’s standard functionality, adding a component to handle the STAMP List management is the only software modification required at the terminal side.

When an LBS session is established, the STAMP List possibly augmented with additional information available during active mode such as TA for GSM, is uploaded to the Positioning Server; see Fig. 1. Based on the type of collected measurements and the QoP requirements the most appropriate positioning algorithm can be selected, to provide a coarse position estimate for each entry in the STAMP List. Subsequently, filtering is used as a post processing step, to smooth the positioning error in the sequence of position estimates and increase the accuracy of the current MT’s position.

The STAMP methodology is compatible with any positioning technique using network related measurements, already deployed at the operator’s network. New and proprietary techniques, including terminal assisted E-OTD and OTDOA, can also be incorporated. STAMP hides the diversity of techniques making the positioning procedure completely transparent to the end user. It should be noted that STAMP is applicable even in case of a single access technology and/or positioning technique. In [42] the STAMP method has been applied in a GSM network to improve the accuracy provided by the CGI++ positioning technique.

3.2. STAMP efficiency analysis

3.2.1. Quality of coarse position estimates

The efficiency of STAMP methodology depends highly on the accuracy achieved by the underlying positioning technique. Errors are introduced in the coarse position estimates due to specific radio propagation conditions, sensitivity of the measurement circuitry and noise disrupting the raw network measurements. Another source of positioning error is the application of a technique using a finite set of candidate terminal positions with inadequate resolution, such as a grid of reference locations in fingerprint methods that is not dense enough. These factors should be taken into account, in order to enhance the positioning technique, achieve higher accuracy for coarse position estimates and further increase the effect of subsequent statistical processing. For example, if RSS measurements are considered, a standard propagation model such as the Hata model [43], can be used to calculate distances between the MT and neighboring BSs and then determine the terminal location through trilateration. However, a propagation model calibrated to best fit the specific environment in the area of interest, is expected to increase the accuracy of coarse position estimates prior to the application of statistical processing. As a result, the performance of STAMP will be improved.

3.2.2. STAMP List size

Another important factor is the STAMP List size, i.e. the number of measurement entries stored locally at the MT. This is closely related to convergence issues regarding the subsequent use of statistical processing. Therefore, the STAMP List should be long enough, so that an adequate number of coarse position estimates are provided to the statistical processing module. The position and velocity estimation provided by the statistical processing at the final position, i.e. the location where the user initiated the LBS application, depend on the number of historical measurements. This is illustrated in Fig. 2 using simulations. Different mobility scenarios are considered, namely Static, Walking, Driving and Fast Driving assuming a terminal is moving at constant speed of 0, 4, 20 and 40 km/h, respectively. RSS values from neighboring BSs are disturbed by shadow fading, modeled as zero mean Gaussian noise with standard deviation \( \sigma = 8 \text{ dB} \) [44]. Under these conditions the average positioning error provided by CGI++ is 250 m. STAMP is then applied on the coarse location estimates provided by CGI++ to investigate the effect of variable STAMP List size. When STAMP is used, a better estimate for the final position and terminal velocity is achieved for all mobility scenarios (positioning error is reduced to 65–75 m). In case of lower mobility scenarios the performance is slightly worse, especially regarding the estimated velocity; see Fig. 2b. This is due to the effect of systematic errors introduced in CGI++ estimates, which is further analyzed in [2]. Simulation results, obtained for all mobility scenarios, reveal that while the STAMP List size is increased, performance is improved. However, beyond the value of 50–60 samples the improvement is only marginal. Thus, the size should be kept as low as possible in order to avoid excessive memory requirements on the MT and reduce the network overhead imposed by the messages exchanged with the Positioning Server. Each entry
contains a small number of parameters, including the timestamp, the network type and network related measurements. Therefore, the required STAMP List size is not excessive. Consequently, it can be well handled both by the MT, regarding on-site storing requirements and network resources, considering the communication overhead when the STAMP List is transferred.

3.2.3. Sampling Period

The Sampling Period, i.e. the time interval between two consecutive entries in the STAMP List, is another crucial parameter. The proper value depends on the accuracy requirements of the LBS application, the access technology that the measurements are related to, as well as the mobility of the terminal. The effect of the Sampling Period on the positioning error, in terms of the 67% and 95% cumulative distribution function (cdf), for different mobility scenarios using STAMP over CGI++ is depicted in Fig. 3. A valid analysis on the Sampling Period has to take into consideration the terminal speed. For example, setting the value to 5 s may prove sufficient for a terminal moving at walking speed (approximately 4 km/h), as the distance covered between successive samples is 6 m. However, this sampling rate can significantly increase the estimation error for a terminal with a speed of 40 km/h, as 60 m are covered in the sampling interval. In [2] the concept of Sampling Distance has been introduced, defined as the distance the terminal covers between two measurements, in order to evaluate the performance of STAMP.

It is obvious that the Sampling Period should be short enough to allow for accurate positioning in the recent past especially when higher mobility scenarios are under consideration, while a longer value is desirable in order to accommodate low battery consumption requirements. Assuming that increasing the sampling frequency does not greatly affect the power required by the MT, the Sampling Period should be adjusted according to the terminal speed, in order to maximize the positioning accuracy. In this context, two distinct policies may be applied, namely the static and dynamic approach. In the static case, an appropriate constant value is used to cover an expected terminal speed range and provide adequate performance. In the dynamic selection scheme, different mechanisms may be employed in order to estimate the terminal dynamics and adapt the Sampling Period accordingly. For instance, the fluctuation of RSS levels and changes to the primary cell and monitored neighboring cells can be used to infer the state of movement, such as walking, driving or stationary [45]. Alternatively, the Sampling Period can be long while the terminal is in idle mode and become as short as possible during the LBS session, in order to provide a number of dense measurements related to the user’s final location.

For real-time tracking applications the Sampling Period is a very important parameter. Some tracking applications require a high level of accuracy and precision. In this case, the lowest possible value should be used to accurately reflect the MT’s trajectory. However, the Sampling Period has a lower bound value due to terminal limitations, such as the clock cycle and access technology specifications, e.g. in GSM RSS values are updated every 480 ms in active mode. Apart from this, the STAMP methodology can be applied to improve the performance of tracking applications. Additionally, based on the level of latency that the application can tolerate, a variable STAMP List size may be selected. In this scheme only a few samples, referring to the current segment of the terminal route, are continuously uploaded to the Positioning Server. Real-time tracking is accomplished by incorporating the STAMP estimate for the last segment as the initial estimate in the Kalman filter for the next segment in an iterative fashion.

4. Proposed architecture

The following architecture is based on a typical server, i.e. the Positioning Server (PS), which is able to support
the basic functionality of STAMP methodology in a simple and efficient way. This architecture is based on the client-server model and standardized communication protocols are employed. The UTAP architecture is designed according to several requirements and criteria set, in order to comply with STAMP principles. Supporting various features of the PS, that may be implemented based on different protocols and techniques, leads to a modular design. The use of separate components, performing distinct and well-defined tasks facilitates system scalability, functional verification and validation, as well as modification of functionalities or interfaces to be compliant with current protocol versions. Compatibility with existing architectures, standards and communication protocols for transferring available network measurements is crucial, in order to have a framework that easily fits into existing service oriented architectures. The ability to expand the system for supporting up-coming wireless access technologies is an important factor. Easy integration of additional positioning algorithms and/or statistical processing techniques in the future, was also considered. Taking into account that several positioning algorithms are available, QoP provisions are necessary to optimize system performance and meet the positioning requirements of individual LBS applications.

The UTAP system consists of the PS and the UTAP clients. Clients, i.e. position requestors, can be MTs or other external LBS applications. Two distinct scenarios are supported regarding the source of the positioning request that correspond to the Terminal and Network Initiated cases, respectively. These requests may have different QoP requirements in terms of positioning accuracy, Maximum Location Age and Maximum Delay. The first attribute ensures that only the technique providing the required accuracy will be used to perform positioning. The second one is used to allow for a previously estimated position to be returned as the current position, instead of employing a positioning technique. The last one is used to exclude those positioning techniques that are computationally inefficient, even if they achieve the desired level of accuracy. The proposed architecture is depicted in Fig. 4.

At the PS side, the Pre-Processing module handles the establishment and termination of the connection with the client through the Client Handler module, as well as the parsing of information carried within the messages exchanged. If the message that carries position related information is valid, its contents are represented in an appropriate internal message format, which is then passed to the Controller module.

The Controller module controls the information flow within the UTAP system. When a connection is established with the PS, the Controller handles the Network and Terminal Initiated sessions. It holds all session specific data, including unique session ID and session specific timer. It also maintains a Location Cache, where previous position estimates are stored for later reference. If the QoP requirements such as the Maximum Location Age set by the application allow it, a cached position will be returned to the client without employing the Algorithms module to perform actual positioning. The Controller has access to the Privacy and Security Management module in order to ensure the privacy and security of the involved parties. Every positioning request made by external clients is authenticated and authorized, otherwise rejected, based on the specific user profile and settings, as well as emergency and lawful regulations that may apply in the future.

The Algorithms module is responsible for calculating position estimates according to the QoP requirements and the location related information contained in the request. More specifically, the Algorithm Selector loads and initializes all available positioning algorithms and filters out the components that cannot provide a position estimate. This case includes a wrong network type or lack of support for the specific area. Then, the best candidate
algorithm is selected to process the actual location query. The decision is affected by the uploaded location related measurements, e.g. a subset of all available techniques are applicable when RSS values from a GSM network are employed. QoP issues related to the positioning accuracy and computational time are also taken into consideration, when selecting a particular technique. Therefore, some techniques may be filtered out because they exceed the QoP thresholds set in the positioning request. Finally, the estimated position is returned. Regarding GPS enabled devices, the actual coordinates are the location measurements. GPS is considered as an optional positioning technique to be selected, depending on Dilution of Precision (DOP) accuracy parameters. These parameters include the Horizontal, Vertical and Position DOP (HDOP, VDOP, PDOP), which are calculated mathematically from the positions of the usable satellites on the local sky. Thus, WLAN fingerprinting might be selected, for example in a street canyon, if the GPS precision is low. The Algorithm Selector has also the ability to switch among different algorithms, while processing the location related information, to support a hybridization scheme. This is valuable when, for example, the MT is moving from indoors (WLAN fingerprinting) to outdoors (cellular techniques or GPS) and vice versa. The Algorithm Selector is also responsible for feeding the sequence of successive position estimates, derived from a single or multiple positioning techniques, to the Statistical Processing module.

The Algorithm Interface provides the communication interface between the Algorithm Selector and the independent algorithm components. This interface ensures that additional positioning algorithms can be easily integrated in the future, thus providing an open positioning platform. Apart from performing the actual positioning when finally queried, every component should be able to provide a fast initial QoP estimation based on the current measurement. This is necessary in order to assist the Algorithm Selector in the selection process. For example, a RSS based algorithm using trilateration should return higher positioning error as initial QoP prediction, if the BSs included in the current measurement are aligned, resulting in poor geometric conditions.

At the terminal side, the UTAP agent is a software component that is responsible for the STAMP related functionality. It maintains and manages the STAMP List, processes all incoming positioning requests, generates responses and handles all low level API communication to access and collect radio layer measurements. Measurements from all available radio access technologies are stored in the STAMP List, in an appropriate format. It should be mentioned that the UTAP system architecture is consistent with the SUPL architecture specification [46]. The User Plane Location Protocol (ULP) [5] and Mobile Location Protocol (MLP) [47] are the communication protocols employed for transferring the messages created by MTs and LBS applications, respectively. ULP messages also convey privacy and authentication related information. Moreover, time-stamping of measurements is already included in the specifications and the support for multiple historical measurements was submitted to OMA as a Change Request for SUPL and has been accepted [48]. Therefore, the STAMP concept and the proposed architecture are in line with User Plane positioning roadmaps.

5. Prototype implementation

The development of the prototype is based on the UTAP system architecture, presented in Section 4. The
UTAP agent has been implemented in C++ and installed on Symbian OS (Series 60) enabled MTs. The STAMP List management functionality currently supports the collection of RSS measurements from GSM, UMTS and WLAN networks, which are stored locally at the MT. The Sampling Period has a fixed predefined value of one second, however in future implementations it can be reconfigured according to a dynamic sampling policy selected by the UTAP agent. Moreover, the Sampling Period may be decided at the application server side and controlled following the Device Management (DM) procedures specified by OMA [49]. A client simulation program has also been developed to test the communication flow between the SUPL enabled terminal and the PS. Through this command line application a RSS measurement survey file collected for a route is properly encoded and sent to the PS as consecutive positioning requests. Actual location coordinates are also included for calculating the positioning error. In order to add some flexibility, the STAMP List size is a user defined parameter.

All individual components comprising the PS, as depicted in Fig. 4, have been realized. Regarding the positioning algorithm components, the Database Correlation Method (DCM), CGI++ and Common Pilot Channel (CPICH) have been implemented. They all rely on RSS values collected by available wireless access networks, but follow two distinct approaches: fingerprint matching and trilateration. In DCM, positioning is conducted by comparing the signal information of the request fingerprint, measured by the MT to be located, to the signal information of the reference fingerprints in the Database and returning the location, based on the best match [21]. The CGI++ positioning technique, which is also known in the literature as Enhanced Cell Global Identity (ECGI) technique, is based on the Cell IDs and RSS values from the serving and neighboring cells measured at the MT side. Then, an estimate of the distance between the MT and each BS is feasible by using a proper radio propagation model. The cost function for the Least Squares estimation, used to derive the MT’s position, is detailed in [42]. The CPICH technique is similar to CGI++ and employs RSS measurements from UMTS networks. The Statistical Processing module included in this prototype, is based on the MT Position Kalman filter described in [29] in this issue, while the corresponding iterative algorithm is detailed in [33].

The communication protocols exploited in this implementation are a subset of the SUPL family protocols, while some modifications were made in order to support additional functionalities. Since WLAN positioning is not yet fully supported, this feature was added. QoP indicators concerning GPS coordinates, such as HDOP, VDOP and PDOP, were also added to indicate if the GPS position meets the required QoP. The option to include a location tag to the uploaded measurements is also provided, as an additional feature for GPS enabled terminals. This can be helpful, for example when a user is asked to contribute RSS fingerprints from a specific area to the operator’s database for a reward. ULP messages are encoded in XML ensuring compliance with the standards. XML was used instead of Packet Encoding Rules (PER) as the former facilitates development and is easier to be handled with standard libraries, even though it is less efficient in terms of resulting data load. In general, the non-roaming, proxy modes for both Network and Terminal Initiated cases are implemented in the UTAP system. Moreover, the SUPL Location Center (SLC) and the SUPL Positioning Center (SPC) [46] are merged into a single entity.

A management GUI has been designed to control the features of the PS, such as enabling or disabling algorithm components and interactively setting some algorithm specific attributes. The option to depict the resulting position estimates on different layers of digital maps is also provided. This feature is not part of the system’s standard functionality, as the primary goal of UTAP system is the provision of high quality location estimates to external LBS applications. This option is implemented for visualization, verification and testing purposes and is used in the following localization scenarios to demonstrate the effect of STAMP methodology.

6. Localization scenarios

Some indicative scenarios were considered in order to evaluate the UTAP system and verify the functionality of individual modules. At the same time preliminary accuracy results for the positioning algorithms and statistical processing components were obtained. RSS samples have been collected in different routes, including indoor and outdoor scenarios. The STAMP List size was fixed to 30 samples. The estimated positions, as presented in Fig. 5, are plotted on digital maps. Statistics are also reported, including the mean positioning error \( m_r \) and the standard deviation \( \sigma_r \) for the whole route. Finally, the PS response time is calculated for a single SUPL client requesting to be positioned, using 30 samples. The delays introduced by each positioning related task, are tabulated in Table 1. These indicative results reveal that the average delay to provide an estimate of the current user location is 185–935 ms, depending on the positioning technique selected to process each sample. The dominating factor in the PS response time is the delay introduced by the positioning algorithms.

6.1. Indoor positioning

In this scenario, a user walking inside a building is considered. The reference fingerprint database contains 378 WLAN fingerprints that have been collected indoors and outdoors close to the building walls, in a 13Km² area covered by 27 WLAN APs. A WLAN attached mobile device is then used to collect samples, containing RSS measurements from all hearable APs, which are stored in the STAMP List. The DCM algorithm is enabled through the management GUI and successive position estimates are depicted on a floor plan map of the VTT premises located in Espoo, Finland as shown in Fig. 5(a). The true
Fig. 5. Localization scenarios using the Positioning Server management GUI. (a) Indoor scenario. (b) Outdoor scenario. (c) Hybrid scheme scenario.
techniques, available as separate modules. The Gradient Descent sample, while the recursive steepest descent algorithm to provide location estimates in 5 ms/
m dots and blue squares, respectively. When only DCM is locations and DCM position estimates are shown in green dots and blue squares, respectively. When only DCM is used, results show that $m_e = 4.7$ m ($\sigma_e = 3.16$ m). The position estimates derived after the statistical processing of the coarse DCM estimates are denoted with red triangles. In this case, the accuracy provided by the STAMP methodology is improved ($m_e = 3.2$ m), ($\sigma_e = 1.90$ m).

6.2. Outdoor positioning

An outdoor scenario is depicted in Fig. 5(b). A MT within a vehicle is collecting measurements from the commercial GSM network of Vodafone-GR, in an urban area of Athens, Greece covered by 20 BSs. For this route, each sample contains RSS measurements from 6 BSs. The CGI++ component provides coarse position estimates (blue squares), while true positions are shown in green dots. The component has been initialized with the necessary network information, including the Cell IDs, as well as the coordinates, transmit power, height and frequency of each BS which are required by the Hata propagation model. Results show that $m_e = 212$ m and $\sigma_e = 94$ m, when only CGI++ is used. The estimated positions obtained after Kalman filtering, are shown in red triangles. Statistical processing leads to considerable improvement regarding the accuracy achieved ($m_e = 167$ m, $\sigma_e = 78$ m). In this scenario, the average positioning error provided by the plain Cell ID technique is 280 m. The standard Hata propagation model was employed [43], however utilizing a calibrated propagation model is expected to enhance the performance of CGI++. Moreover, calibrating the Kalman filter parameters will further increase the effect of the statistical processing.

6.3. Hybrid positioning

The hybridization concept is illustrated in Fig. 5(c) for a user passing through a building. The same fingerprint data-base, as in Section 6.1, is used. GPS and DCM are the positioning techniques under consideration and the most suitable for each sample is selected, based on the estimated QoP. The true locations, DCM estimates and GPS positions are shown in green dots, blue squares and black x-marks, respectively. In the $AB$ segment, DCM estimates are only available when the user approaches the main building walls. GPS location estimates are heavily affected by the building and some times indicate a route through the walls. Inside the building ($BC$ segment) only DCM estimates are available. Statistical processing is applied on the position estimates derived after hybridization to further increase accuracy (red triangles). When the user moves outdoors ($CD$ segment), DCM estimates are still available and are used for some time, since the GPS receiver does not have a satellite fix. As the user moves away from the building the DCM accuracy degrades and eventually DCM estimates are unavailable. Therefore, GPS estimates are selected. However, it takes time for the GPS estimates to get accurate enough and this also affects the performance of the Kalman filter in the $CD$ segment. Estimates provided by the statistical processing component indicate a sudden hop in the user position and the estimated velocity, shown in red arrows, increases rapidly. If WLAN fingerprints were available for that part of the route, DCM instead of GPS estimates could be used to achieve higher accuracy. Even so, in the hybrid GPS/DCM scheme $m_e = 8.36$ m ($\sigma_e = 7.1$ m), while $m_e = 11.06$ m ($\sigma_e = 8.69$ m) if only GPS is used.

7. Conclusions

In this article, a terminal assisted localization methodology, applicable in current cellular networks, has been presented. STAMP is compatible with any positioning technique, while it enhances the positioning performance and allows the tracking of mobile terminals. The deployment of STAMP implies only additional software modifications at the terminal and network side. The proposed architecture follows the SUPL architecture specifications and has been designed under consideration of existing standards and communication protocols such as ULP and MLP. Furthermore, it is in line with market trends showing that terminal manufacturers will deliver SUPL enabled devices within the next years. Moreover, the modular architecture facilitates fast integration of new and custom positioning algorithm components, providing an open platform for enabling the provision of LBS.

Future work will focus on finalizing and testing the UTAP system prototype, as most parts have already been realized. We plan to investigate and implement additional positioning techniques, such as probabilistic DCM, UMTS RTT and post processing methods, such as Kalman filter variants and map matching. Additionally, we intend to improve the positioning accuracy by calibrating the filter parameters and performing extended trials, both indoors and outdoors. Further steps will concentrate on enhancing

<table>
<thead>
<tr>
<th>Task</th>
<th>Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of the SUPL Client Handler</td>
<td>0.1</td>
</tr>
<tr>
<td>Verification of the SUPL XML message format</td>
<td>3</td>
</tr>
<tr>
<td>Location query message decoding (ULP parser)</td>
<td>4</td>
</tr>
<tr>
<td>Initial QoP estimation for each enabled algorithm</td>
<td>1</td>
</tr>
<tr>
<td>Positioning using CGI++ technique</td>
<td>150 or 900</td>
</tr>
<tr>
<td>Positioning using DCM technique</td>
<td>270</td>
</tr>
<tr>
<td>Statistical processing initialization and execution</td>
<td>25</td>
</tr>
<tr>
<td>Location response message encoding (ULP parser)</td>
<td>2</td>
</tr>
</tbody>
</table>

Results are obtained assuming the PS is running on a Pentium 4 3.6 GHz PC with 1 GB RAM and the STAMP List size is 30 samples.

The CGI++ component employs two different computational techniques, available as separate modules. The Gradient Descent is based on a recursive steepest descent algorithm to provide location estimates in 5 ms/sample, while the Grid approach uses a grid to search exhaustively for the optimum location and requires 30 ms/sample.


The true locations, DCM estimates and GPS positions are shown in green dots, blue squares and black x-marks, respectively. In the $AB$ segment, DCM estimates are only available when the user approaches the main building walls. GPS location estimates are heavily affected by the building and some times indicate a route through the walls. Inside the building ($BC$ segment) only DCM estimates are available. Statistical processing is applied on the position estimates derived after hybridization to further increase accuracy (red triangles). When the user moves outdoors ($CD$ segment), DCM estimates are still available and are used for some time, since the GPS receiver does not have a satellite fix. As the user moves away from the building the DCM accuracy degrades and eventually DCM estimates are unavailable. Therefore, GPS estimates are selected. However, it takes time for the GPS estimates to get accurate enough and this also affects the performance of the Kalman filter in the $CD$ segment. Estimates provided by the statistical processing component indicate a sudden hop in the user position and the estimated velocity, shown in red arrows, increases rapidly. If WLAN fingerprints were available for that part of the route, DCM instead of GPS estimates could be used to achieve higher accuracy. Even so, in the hybrid GPS/DCM scheme $m_e = 8.36$ m ($\sigma_e = 7.1$ m), while $m_e = 11.06$ m ($\sigma_e = 8.69$ m) if only GPS is used.

7. Conclusions

In this article, a terminal assisted localization methodology, applicable in current cellular networks, has been presented. STAMP is compatible with any positioning technique, while it enhances the positioning performance and allows the tracking of mobile terminals. The deployment of STAMP implies only additional software modifications at the terminal and network side. The proposed architecture follows the SUPL architecture specifications and has been designed under consideration of existing standards and communication protocols such as ULP and MLP. Furthermore, it is in line with market trends showing that terminal manufacturers will deliver SUPL enabled devices within the next years. Moreover, the modular architecture facilitates fast integration of new and custom positioning algorithm components, providing an open platform for enabling the provision of LBS.

Future work will focus on finalizing and testing the UTAP system prototype, as most parts have already been realized. We plan to investigate and implement additional positioning techniques, such as probabilistic DCM, UMTS RTT and post processing methods, such as Kalman filter variants and map matching. Additionally, we intend to improve the positioning accuracy by calibrating the filter parameters and performing extended trials, both indoors and outdoors. Further steps will concentrate on enhancing
the QoP provisions and providing uninterrupted positioning, especially in hybrid schemes where different network measurements and multiple positioning techniques are available.

Acknowledgements

This article introduces concepts and technologies deployed within the framework of the project MOTIVE [7], which is partially funded by the European Commission under the 6th framework of the IST program. This work is partly supported by the Cyprus Research Promotion Foundation under contracts ENI\$/0506/59 and ΠΑΗΠΟ/0603/06. The authors thank Reach-U (http://www.reach-u.com) for the provision of digital maps in the context of MOTIVE.

References


[40] TS 43.022 v.7.2.0: Functions related to Mobile Station (MS) in idle mode and group receive mode.

[41] TS 25.304 v.6.8.0: User Equipment procedures in idle mode and procedures for cell reselection in connected mode.


Abstract—The increasing demand for indoor location-based services has motivated the development of positioning methods that exploit the existing wireless network infrastructure. Accuracy is an important requirement, however fault tolerance is also highly desirable in case of failures or malicious attacks. We investigate the fault tolerance of fingerprint-based methods under a variety of fault or attack scenarios. We study the Subtract on Negative Add on Positive (SNAP) algorithm and modify it appropriately for the WLAN setup. Our results indicate that SNAP achieves adequate accuracy with very low computational complexity and exhibits smoother performance degradation in the presence of faults compared to other methods.

I. INTRODUCTION

Indoor positioning of individuals or equipment has attracted research interest over the last decade. This is due to the fact that people tend to spend most of their time in indoor environments, such as office buildings or shopping malls, where satellite based positioning is infeasible. The objective is to provide accurate and reliable location estimates using the available wireless networks in order to enable the development of location oriented services and applications, including in-building guidance, event detection and asset tracking.

Several positioning methods rely on WLANs and exploit Received Signal Strength (RSS) measurements to infer location, owing to the wide availability of WLAN Access Points (AP) and the ease of collecting RSS samples without specialized equipment. Such methods utilize a number of RSS fingerprints collected a priori at some predefined reference locations. Location can then be estimated by finding the best match between the currently measured fingerprint and reference fingerprints [1]–[4]. So far, the focus has been on improving accuracy, however the time required to estimate user location is also very important, because it affects the battery consumption of mobile devices. Moreover, providing adequate accuracy when faults are injected in the positioning system is another requirement that has received little attention.

To this end, we build upon the Subtract on Negative Add on Positive (SNAP) algorithm that was proposed in our previous work [5] to address the problem of event detection in binary sensor networks and demonstrated some desirable properties, such as low complexity and fault tolerance. Firstly, we adapt the SNAP algorithm to the WLAN setup using only information of whether an AP is detected during positioning or not. Secondly, we show how the accuracy of SNAP can be improved by introducing the idea of zones to exploit different RSS levels. This algorithm achieves a level of accuracy that is comparable to other well-known positioning methods, but is considerably simpler and much faster which is desirable for low power mobile devices in order to save valuable energy. Finally, we investigate the fault tolerance of SNAP against other methods for a variety of fault or attack scenarios and present a variant of SNAP that exhibits smooth performance degradation, as the percentage of faulty APs is increased.

In Section II, we provide the details of the SNAP algorithm and describe the SNAP variants. Experimental results in our performance evaluation are presented in Section III, followed by the study on fault tolerance in Section IV. The conclusions and some ideas for future work are discussed in Section V.

II. POSITIONING WITH RSS FINGERPRINTS

A. Definitions

We use a set of predefined reference locations \( \{L : \ell_i = (x_i, y_i), i = 1, \ldots, l\} \) to collect RSS values from \( n \) APs deployed in the area of interest (offline phase). A reference fingerprint \( r_i = [r_{i1}, \ldots, r_{in}]^T \) associated with location \( \ell_i \), is a vector of RSS samples and \( r_{ij} \) denotes the RSS value related to the \( j \)-th AP. Usually, \( r_i \) is averaged over multiple fingerprints collected at \( \ell_i \) to alleviate the effect of noise in RSS measurements and outlier values. During positioning (online phase), we exploit the reference data to obtain a location estimate \( \hat{\ell} \) given a new fingerprint \( s = [s_1, \ldots, s_n]^T \) measured at the unknown location \( \ell \).

We define the Region of Coverage \( RoC_j \subseteq L, j = 1, \ldots, n \) as the subset containing the reference locations where the \( j \)-th AP is detected during the offline phase. For instance, all reference locations in our experimental setup (small black dots) and the locations inside the \( RoC \) of a single AP (larger dots) are depicted in Fig. 1. The AP is located in the top left wing (black triangle) and the grayscale colorbar indicates the mean RSS level from that AP at each location.

![Fig. 1. Example RoC of an AP in our experimental setup.](image-url)
B. Existing Positioning Methods

The Nearest Neighbor method [1] estimates location by minimizing the Euclidean distance between the observed fingerprint during positioning \( s \) and the reference fingerprints \( r_j \), i.e. \( \hat{\ell}(s) = \arg \min_{\ell_i} D_i \), where \( D_i = \sum_{j=1}^{n} (r_{ij} - s_j)^2 \). In the \( K \) Nearest Neighbors (KNN) method location is estimated as the mean of \( K \) reference locations with the shortest distances.

A median-based distance metric, instead of the Euclidean, was proposed in [6] to improve the fault tolerance of the standard KNN method when failures or incorrect RSS readings are present during positioning.

The probabilistic Minimum Mean Square Error (MMSE) method uses the conditional probabilities \( p(\ell_i|s) \) and determines location as \( \hat{\ell}(s) = \sum_{i=1}^{n} \ell_i p(\ell_i|s) \). Applying Bayes rule the problem reduces to estimating \( p(s|\ell_i) \) and assuming that RSS measurements from neighboring APs are independent we get \( p(s|\ell_i) = \prod_{j=1}^{n} p(s_j|\ell_i) \). The Kernel approach [2] uses Gaussian kernels to obtain \( p(s_j|\ell_i) \) from the reference data.

C. Positioning with SNAP using Binary Data

We exploit the available RSS fingerprints by utilizing only binary information, i.e. an AP is either detected in the online phase or not. During positioning, the currently observed fingerprint \( s \) contains RSS values from a subset of the available APs denoted as \( S \). In this case, the SNAP algorithm employs three main components to derive the unknown user location.

1) Region of Coverage (RoC): In the offline phase, we determine the Region of Coverage \( RoC_j \), \( j = 1, \ldots, n \) for all available APs, based on the reference data.

2) Likelihood Matrix \( L \): In the online phase, each element in the \( l \times n \) matrix \( L \) is updated to reflect the contribution of the \( j \)-th AP to the reference locations \( \ell \in RoC_j \). Every AP that is detected in the currently observed fingerprint \( s \) adds a positive one (+1) contribution to the elements of \( L \) that correspond to the locations inside its respective \( RoC \). On the other hand, every AP that is not detected in \( s \) adds a negative one (−1) contribution to the elements of \( L \) that correspond to the locations inside its respective \( RoC \). The positive and negative contributions for a single AP are illustrated in Fig. 2.

Formally, the elements of \( L \) are obtained by

\[
\mathcal{L}(i, j) = \begin{cases} 
+1, & j \in S \text{ AND } \ell_i \in RoC_j \\
-1, & j \notin S \text{ AND } \ell_i \in RoC_j \\
0, & \ell_i \notin RoC_j
\end{cases}
\]

Then, for each location \( \ell_i \), \( i = 1, \ldots, l \) we calculate the likelihood value \( LV_i \) of the user being located at \( \ell_i \) by summing the contributions of all APs

\[
LV_i = \sum_{j=1}^{n} \mathcal{L}(i, j)
\]

3) Location Estimation: The maximum of the likelihood values points to the estimated location given by

\[
\hat{\ell}(s) = \arg \max_{\ell_i \in L} LV_i
\]

In cases where more than one reference locations \( \ell_i \) have the same maximum value \( LV_i \), then the estimated location is the mean of the corresponding locations.

Example: We illustrate the application of SNAP algorithm in a simple scenario; see Fig. 3. We consider only four APs (triangles) in our reference data and the locations that are covered by at least one of these APs are shown in dots. During positioning, the user resides in an unknown location (square) and the observed fingerprint \( s \) contains RSS values from three APs, while the AP installed in the top middle wing is not detected. The three APs that are detected, add a positive one (+1) contribution to the elements of \( L \) that correspond to the locations inside their respective RoC. The AP that is not detected adds a negative one (−1) contribution to the elements of \( L \) that correspond to the locations inside its RoC. The resulting likelihood values, after adding and subtracting the contributions of all APs, are shown in Fig. 3. The user location is estimated (shown with a star) as the mean of the reference locations that have the same maximum likelihood value +3.

D. On the fault tolerance of SNAP

In the SNAP algorithm, an AP contributes to the location estimation whether it is detected in the observed fingerprint \( s \) during positioning or not. This can be very effective in the fault-free case, however in case of faults SNAP may not be able to provide adequate performance. Faults may be caused unintentionally due to unpredicted AP failures, or deliberately

![Fig. 2. Contribution of an AP to the locations inside its RoC.](image1)

![Fig. 3. Example application of the SNAP method.](image2)
by an adversary who aims at compromising the positioning system. For instance, when AP failures occur during positioning, then a subset of the APs that would otherwise be present in \( s \) are no longer detected. Thus, the negative contributions of these APs may introduce high errors in the estimated user location. We propose a SNAP variant, denoted as SNAPf, that mitigates this effect by ignoring the negative contributions of the failed APs. In this case, the elements of \( \mathcal{L} \) are obtained by

\[
\mathcal{L}(i, j) = \begin{cases} 
+1, & j \in S \text{ AND } \ell_i \in \text{RoC}_j \\
0, & \ell_i \notin \text{RoC}_j
\end{cases} \quad (4)
\]

E. Improving Accuracy with RSS levels

The original SNAP algorithm uses only binary information and thus it is not expected to provide a high level of accuracy. We can further improve the performance by taking into account the information about RSS levels. The idea is that if an AP is detected during positioning, then the user is more likely to reside in the locations inside the RoC of the specific AP that have similar RSS values to the observed RSS value.

Based on the reference data we may determine the range of RSS values and let \( \min \) and \( \max \) denote the minimum and maximum RSS values, respectively. We divide this range of RSS values into \( M \) subranges, i.e. non overlapping equally spaced intervals, and the \( m \)-th interval \( Z_m \) is given by

\[
Z_m = [\min + (m - 1)A, \min + mA], \quad m = 1, \ldots, M \quad (5)
\]

where \( A = \frac{\max - \min}{M} \).

We can now define the Zone of Coverage \( \text{ZoC}_{mj} \subseteq \text{RoC}_j \), \( m = 1, \ldots, M \) and \( j = 1, \ldots, n \), as the subset containing the reference locations \( \ell_i \) where the \( j \)-th AP is detected during the collection of reference RSS fingerprints and \( r_{ij} \in Z_m \). The zones \( \text{ZoC}_{mj} \) for all available APs are determined prior to positioning and essentially each \( \text{RoC}_j \) is divided into \( M \) zones so that \( \text{RoC}_j = \bigcup_{m=1}^{M} \text{ZoC}_{mj} \). The modified SNAP algorithm, denoted as SNAPz, incorporates the notion of zones and the elements of \( \mathcal{L} \) are now obtained by

\[
\mathcal{L}(i, j) = \begin{cases} 
+1, & j \in S \text{ AND } \ell_i \in \text{ZoC}_{mj} \\
0, & j \in S \text{ AND } \ell_i \in \left( \bigcup_{k=1}^{m-1} \text{ZoC}_{kj} \right) \\
-1, & j \notin S \text{ AND } \ell_i \in \text{RoC}_j - \bigcup_{k=m}^{M} \text{ZoC}_{kj} \\
0, & \ell_i \notin \text{RoC}_j
\end{cases} \quad (6)
\]

Using this rule, every AP that is detected in the currently observed fingerprint \( s \) adds a positive one (+1) contribution only to those elements of \( \mathcal{L} \) that correspond to locations inside the appropriate zone \( \text{ZoC}_{mj} \). A zero (0) contribution is added to the locations inside the neighboring zones, i.e. \( \text{ZoC}_{(m-1)j} \) and \( \text{ZoC}_{(m+1)j} \), while a negative one (−1) contribution is added to the locations inside the remaining zones. The intuition is that when an AP is detected in the online phase with certain RSS value, then the user resides with high probability in the zone where the reference locations have similar RSS values. Due to the noise disturbing the RSS values the user may be located with some probability in the neighboring zones. Finally, the user is located with low probability in the other zones, where the reference locations have RSS values that are very dissimilar to the observed RSS value. We define \( \text{ZoC}_{0j} = \text{ZoC}_{(M+1)j} = \emptyset \) to handle the boundary conditions for \( m = 1 \) and \( m = M \) in (6).

III. PERFORMANCE EVALUATION

A. Measurement Setup

We collected our reference data in a typical office environment at VTT Technical Research Centre of Finland. We used a smartphone to collect 30 RSS fingerprints at 107 distinct reference locations on the second floor for a total of 3210 reference fingerprints. There are 31 WLAN APs installed in the building and on average 9.7 APs are detected per location. The floorplan of the area and the reference locations are depicted in Fig. 1. We collected testing data by walking over a path that consists of 192 locations. One fingerprint is recorded at each location, and the same path is sampled 3 times.

B. Positioning Accuracy

We examine the performance of SNAPz with respect to the accuracy for varying number of zones. In our setup, we found that \( \min = -101 \text{dBm} \) and \( \max = -34 \text{dBm} \). This range is divided in intervals of size \( \frac{\max - \min}{M} \) and each zone contains the locations with RSS values that fall into the respective interval.

The mean positioning error (\( \mathcal{E} \)) over all test data is plotted in Fig. 4, as a function of the number of zones. If \( M \) is small, then each zone contains many reference locations. Thus, when a zone is activated during positioning, then more locations are likely to be used in the location estimation and the error is increased. Note that in case \( M = 1 \), SNAPz is equivalent to the SNAP algorithm that uses only binary data. On the other hand, if \( M \) is large, then each zone contains only few locations and due to the noise in the RSS values the wrong zone may be activated during positioning, leading to accuracy degradation.

The curve indicates that the highest level of accuracy is achieved for \( M = 10 \) zones, however \( \mathcal{E} \) does not vary significantly for \( 4 \leq M \leq 11 \). For reasons that are related to fault tolerance and will become clear shortly, we select \( M = 4 \) and this value is used for the rest of the experiments. To provide a fair comparison, we identified in a similar fashion that \( K = 3 \) neighbors and kernel width \( \sigma = 7 \) provides the lowest \( \mathcal{E} \) for KNN and MMSE methods, respectively.

![Fig. 4. Positioning accuracy of SNAPz for variable number of zones.](image-url)
In the following, we further investigate the positioning accuracy of SNAPz and the statistics for the positioning error are summarized in Table I. Results indicate that the MMSE method achieves the highest level of accuracy ($E = 2.46$), followed by KNN ($E = 2.70$). SNAPz provides less accurate location estimates and the mean error is increased by around 1m compared to other methods. However, this accuracy degradation is not significant and $E = 3.64$ is adequate for most indoor location-based services and applications.

C. Computational Complexity

We investigate the estimation time of the positioning methods by using a Matlab implementation on an Intel Pentium 4 processor 3.6GHz with 1GB RAM, while the execution times are averaged over 100 runs using the test data. The number of computations and the time required by each method are summarized in Table II. SNAPz does not require heavy computations and one location estimate takes 0.49msec, which is 1.6 and 3.5 times lower compared to KNN and MMSE methods, respectively. Therefore, the SNAPz method can extend the battery life of low power mobile devices, such as smartphones, especially when frequent positioning requests or tracking applications are considered.

B. Performance under Faults

Under the False Positive model, an AP is detected during positioning in locations outside its original RoC. This can happen in case a heavy object, which was previously obstructing the propagation path, has moved and the transmission signals can travel further. An attack scenario that manifests in a similar manner is when a rogue AP is deployed and programmed to replicate an existing AP. This model is simulated by injecting random RSS values to a subset of the APs in 70% of the test locations, where those APs would otherwise be undetected.

The AP Relocation model captures the effect of an AP which is detected during positioning inside an area that is different than the expected one. This may happen when the AP is moved to a new location, e.g. for network operation reasons. An attacker can impersonate a specific AP, while at the same time eliminate the AP signals through jamming. This model is simulated by replacing the RSS signals of the faulty AP in the test data with the values of another randomly selected AP.

IV. Fault Tolerance

We study the performance of positioning methods in case of faults by using some realistic fault models presented in [7].

A. Fault Models

The AP Failure model assumes that several APs used in the offline phase may not be available during positioning. This can be caused by random AP failures, such as power outages. Alternatively, an attacker can cut off the power supply of some APs or use specialized equipment to jam the communication channel. We simulate this model by removing the RSS values of the faulty APs in the original test fingerprints.

The False Negative model assumes that an AP is no longer detected during positioning in some locations inside its original RoC. This can happen accidentally if furniture or equipment is moved, so that the propagation path is blocked and the AP signal cannot be detected in locations where it was previously weak. We simulate this model by ignoring valid RSS readings for a subset of APs in 70% of the test locations (randomly selected), where the APs were previously detected.

Under the False Positive model, an AP is detected during positioning in locations outside its original RoC. This can happen in case a heavy object, which was previously obstructing the propagation path, has moved and the transmission signals can travel further. An attack scenario that manifests in a similar manner is when a rogue AP is deployed and programmed to replicate an existing AP. This model is simulated by injecting random RSS values to a subset of the APs in 70% of the test locations, where those APs would otherwise be undetected.

The AP Relocation model captures the effect of an AP which is detected during positioning inside an area that is different than the expected one. This may happen when the AP is moved to a new location, e.g. for network operation reasons. An attacker can impersonate a specific AP, while at the same time eliminate the AP signals through jamming. This model is simulated by replacing the RSS signals of the faulty AP in the test data with the values of another randomly selected AP.

B. Performance under Faults

We investigate fault tolerance with respect to the performance degradation, as the percentage of faulty APs is increased. We apply the fault models to corrupt the original test data and the results for $E$ are averaged over 20 runs using randomly selected subsets of faulty APs in each run.

The performance of SNAPz under the AP Failure model for varying number of zones is plotted in Fig. 5. In the fault-free case and when less than 50% of APs are corrupted, using $M = 4$ zones provides a high level of accuracy and the performance degrades smoothly. In case more than half of the APs are corrupted, using $M = 1$ zone improves the fault tolerance of SNAPz. Using more zones has a negative effect as $E$ increases rapidly. Similar behaviour was observed for the other fault models. Thus, we use $M = 4$ zones in SNAPz algorithm that is a good tradeoff between accuracy and fault tolerance.

The SNAPft variant can be extended with the use of zones. Interestingly, we found that using $M = 4$ zones is also a good option for SNAPft. Thus, we consider SNAPft ($M = 4$) and the median-based KNN ($K = 3$) approach, denoted as MED.

Under the AP Failure model (Fig. 6a), the MED method performs slightly better than KNN, followed by MMSE method.
The SNAPz algorithm is not resilient to this type of faults and $\mathcal{E}$ increases rapidly as the percentage of faulty APs exceeds 40%. In this case, the SNAPft algorithm proves to be very fault tolerant. For instance, $\mathcal{E} = 6.38m$ for SNAPft when 60% of the APs are corrupted compared to 9.80m, 10.40m, 12.09m and 19.64m for MED, KNN, MMSE and SNAPz, respectively.

Similar behavior is observed when the False Negative model is assumed; see Fig. 6b. SNAPft exhibits high fault tolerance and outperforms the MED algorithm, especially when more than 60% of the APs are corrupted. For SNAPft $\mathcal{E}$ remains below 8m, even when all APs are corrupted. On the other hand, under the False Positive model (Fig. 6c) $\mathcal{E}$ explodes for KNN and MMSE methods. Adequate fault tolerance can be achieved with MED algorithm, however $\mathcal{E}$ grows with higher rate when the percentage of faulty APs increases beyond 50%. The SNAPz and SNAPft algorithms are very fault tolerant and $\mathcal{E}$ is less than 6m for both, even when all APs are affected.

Faults under the AP Relocation model seem to cause similar performance degradation to all positioning methods when less than half of the APs are corrupted, as shown in Fig. 6d. Beyond that point, SNAPft has the same level of fault tolerance with MED and they both perform better compared to KNN, MMSE and SNAPz methods. To summarize, the SNAPft variant with the zones improves significantly the fault tolerance of the SNAPz algorithm, under almost all fault models in our evaluation, and performs better than other positioning methods.

V. CONCLUSION

We introduced an implementation of the SNAP algorithm for WLAN fingerprint-based positioning that is simple and time efficient. This algorithm was modified to improve the accuracy and the resilience to faults. Experimental results indicate that the SNAP variants achieve the desired performance in terms of accuracy and fault tolerance, while maintaining low computational complexity. For future work we plan to implement the SNAP algorithm on mobile devices to study its energy consumption compared to other positioning methods.

ACKNOWLEDGMENT

This work is supported by the Cyprus Research Promotion Foundation. Authors would like to thank P. Kemppi and Y. Li at VTT Technical Research Centre of Finland (www.vtt.fi) for the provision of experimental WLAN RSS data.

REFERENCES


Fault Tolerant Positioning using WLAN Signal Strength Fingerprints

C. Laoudias, M. P. Michaelides and C. G. Panayiotou
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering, University of Cyprus
Kallipoleos 75, P. O. Box 20537, 1678, Nicosia, Cyprus
Email: laoudias@ucy.ac.cy, michalism@ucy.ac.cy, christosp@ucy.ac.cy

Abstract—Accurate and reliable location estimates using wireless networks are important for enabling indoor location oriented services and applications, such as in-building guidance and asset tracking. Providing adequate level of accuracy in case of faults or attacks to the positioning system is equally significant, thus our main interest is on the fault tolerance of positioning methods, rather than the absolute accuracy in the fault-free case. We introduce several fault models to capture the effect of failures in the wireless infrastructure or malicious attacks and discuss how these models can simulate the corruption of signal strength values during positioning. The models are used to investigate the fault tolerance of positioning methods and evaluate them in terms of their performance degradation as the percentage of corrupted signal strength measurements increases. Experimental results using our fault models are also presented.

I. INTRODUCTION

The use of wireless networks to infer the unknown location of individuals or equipment, especially in indoor environments, has attracted research interest over the last decade. This is mainly due to the increasing demand for location-based services and applications, such as in-building guidance, asset tracking in hospitals or warehouses, event detection and autonomous robot navigation. Different positioning technologies have been discussed in the literature including infrared, Bluetooth, Radio-frequency identification (RFID), Ultra-wideband (UWB), ultrasound and Wireless Local Area Network (WLAN); see [1], [2] for an overview of technologies to determine location and commercial positioning systems.

A wide range of positioning methods rely on WLANs, owing to the wide availability of relevant infrastructure. The WLAN Access Points (AP) may belong to the private fully-controlled network of a single operator that provides the positioning service. Alternatively, public WLAN-based positioning systems, such as Skyhook [3], rely on publicly available APs and exploit information about all APs that can be detected in the area of interest. Usually WLAN-based methods exploit Received Signal Strength (RSS) samples from APs, which can be easily collected without specialized equipment. In this context, several approaches utilize a number of RSS fingerprints collected a priori at some predefined reference locations. Location can then be estimated using the currently measured fingerprint to find the best match between the current and reference fingerprints [4]–[8].

The focus of positioning methods so far has been on improving accuracy. In real world, however, WLAN APs can fail or exhibit erroneous behaviour, thus compromising the performance of these methods. For instance, some APs may be unavailable during positioning due to random and unpredictable failures, such as power outages. Fingerprint positioning systems are also susceptible to non-cryptographic attacks that render a set of APs useless or corrupt the expected RSS values by altering the propagation environment. We treat these failures and attacks in a unified framework, because they both inject faults that may lead to performance degradation during positioning, and investigate the fault tolerance of positioning methods. Fault tolerance is an important issue that has received little attention in the literature. Our main contribution is to define realistic fault models, study the performance of positioning algorithms in the presence of faults and motivate future research in this direction.

The rest of the paper is structured as follows. Section II is a survey of previous work on fault or RSS attack models, detection schemes and fault tolerant positioning methods. In Section III, we introduce several fault models that capture the effect of AP malfunctions or malicious attacks during positioning. In Section IV, we study the well-known Nearest Neighbor method. Our measurement setup is described in Section V, followed by the experimental results in Section VI. Finally, the conclusions and some ideas for future work are discussed in Section VII.

II. RELATED WORK

A. Fault and Attack Models

Some early works investigate the performance of positioning algorithms when a single AP is shut down either intentionally or accidentally. Authors in [9] evaluate several NN variants and weighting schemes in a multi-floor area covered by 10 APs to determine their robustness when the AP, that is closest to the mobile device during positioning becomes unavailable. In [10] the effect of eliminating one out of five APs in the position estimation accuracy is studied using Monte Carlo simulations based on IEEE 802.11 channel models. Both works reach the conclusion that NN approaches, especially if more than one neighbors are used, are quite robust to single AP failures.
In [11] an attack is simulated by randomly choosing the RSS readings of one or two out of six APs and multiplying them with a constant. Authors in [12] consider a similar linear attack model which is simulated by perturbing the original RSS values over all APs by a constant attenuation or amplification constant. It was observed that using a real material, such as glass, metal, foil, books etc, causes a constant percentage power loss independent of distance. This type of attacks is easy to launch with low cost materials and at the same time the adversary may control the effect of the attack by selecting the appropriate material [13]. On the other hand, amplification attacks can be performed by deliberately increasing the AP transmit power.

The model employed in [14] assumes that RSS measurements are corrupted by additive Gaussian noise. This is motivated by the standard log-distance signal propagation model [15] that provides the received signal level as a function of the transmitted power, the distance to the transmitter and the path loss exponent. Under this model, an RSS attack is caused by altering the propagation environment and is simulated by adding Gaussian noise to the collected test data.

B. Fault and Attack Detection

Fault tolerant positioning systems could be supported by fault (attack) detection mechanisms that are efficient, i.e. exhibit high detection and low false positive rates. For instance, a detection component could trigger an alert for the security personnel each time there is a fault (attack) indication. Furthermore, the positioning component could also switch to a fault tolerant counterpart in order to mitigate the effect of the fault (attack) and still provide adequate level of accuracy until the problem is resolved.

Attack detection in wireless localization is studied in [13], [16] for a variety of positioning methods, including range-based and RSS fingerprinting, and detection relies on statistical significance testing. For example, in the case of Nearest Neighbor RSS fingerprint positioning, the minimum distance between the fingerprint observed during positioning and the fingerprints in the pre-constructed radio map, denoted as $D_r$, is used as the test statistic. The distribution of the training data contained in the radio map is used to select an appropriate threshold $\tau$ and subsequently an attack is signified during positioning in case $D_r > \tau$. A key observation in this work is that the performance of the proposed detection method is better under signal amplification, compared to attenuation attacks. For probabilistic positioning techniques equivalent test statistics are studied, including the likelihood of the location with the highest value or the sum of the likelihoods over all locations. Both test statistics are found to decrease significantly under attack.

Authors in [17] exploit the communication capabilities among transmitters in a WSN setup based on MicaZ beacon nodes to decide whether there are node failures in the system. In this approach, beacon nodes periodically measure their local neighborhood, defined as the set of other beacon nodes that they can hear. This neighborhood is compared to the original neighborhood, which is measured shortly after the system has been installed. If the intersection between the current and original neighborhoods is large, the system is assumed to be fault-free. On the other hand, if the fraction of failed nodes exceeds some threshold, then failure (or similarly an attack) is detected. However, this approach assumes adequate connectivity between beacon nodes that does not change substantially over time. Moreover, due to the node communication requirement, this approach cannot be directly applied to positioning methods that rely on WLAN AP infrastructure.

C. Fault Tolerant Positioning Methods

The fault tolerance of positioning methods has been mainly explored in the context of WSNs, where node failures can be frequent, and focused particularly on range-based techniques; see [18] for an overview and experimental evaluation. These are also known as multilateration techniques in which location is estimated in a least squares sense using a set of distances from at least three landmarks with known locations. Several types of measurements can be used to obtain the required distances, including Time of Arrival (TOA), Time Difference of Arrival (TDOA) and RSS. In [19] a set of static and mobile hidden base stations are used for secure positioning in range-based systems. The resistance of this class of techniques to distance spoofing attacks, e.g. by altering the RSS level that leads to erroneous distance calculation, is analyzed in [20] and a mechanism for secure positioning, coined verifiable multilateration, is described.

In our previous work [21] we investigated fault tolerance using a network of wireless sensors that make binary observations. Such simple sensors are able to report the presence of an event or intruder, when the measured signal at their location is above a threshold (positive observations), or otherwise remain silent (negative observations). We introduced the Subtract on Negative Add on Positive (SNAP) algorithm that exploits these binary sensor beliefs in order to estimate the event/intruder location in an efficient and fault tolerant manner, even when a large number of sensors report erroneous observations.

As a first step to improving system robustness to RSS-based attacks, authors in [11] suggest increasing the redundancy by using more sensors or APs. Moreover, the effect of outlier APs is reduced with the introduction of a median-based, instead of the Euclidean, distance measure that is applicable in both range-based and fingerprint-based positioning methods. Similarly, in the context of MoteTrack positioning system [17], the Euclidean distance in the NN algorithm is replaced by an adaptive fingerprint distance measure to cater for faulty nodes. Under the presence of faults, the adaptive measure penalizes only RSS values found in the currently observed fingerprint $s$ and not in a reference fingerprint $r$, so as to minimize the errors introduced from failed nodes. In the fault-free case, the algorithm reverts to the standard measure, thus penalizing RSS values from all nodes not found in common between $r$ and $s$.

On a different line, Kushki et al. [14] describe a sensor selection methodology, based on a nonparametric estimate of the Fisher Information, for increasing the resilience of
fingerprinting-based positioning systems to RSS attacks. Essentially, this method selects only a number of reliable APs from the set of available APs to mitigate the attack.

III. FAULT MODELS

In the context of fingerprint-based positioning, we use a set of predefined reference locations \( \{ L : \ell_i = (x_i, y_i), i = 1, \ldots, l \} \) to collect RSS values from \( n \) APs deployed in the area of interest (offline phase). A reference fingerprint \( r_i = [r_{i1}, \ldots, r_{in}]^T \) associated with location \( \ell_i \), is a vector of RSS samples and \( r_{ij} \) denotes the RSS value related to the \( j \)-th AP. Usually, \( r_i \) is averaged over multiple fingerprints collected at \( \ell_i \) to alleviate the effect of noise in RSS measurements and outlier values. During positioning (online phase), we exploit the reference data to obtain a location estimate \( \hat{\ell} \), given a new fingerprint \( s = [s_1, \ldots, s_n]^T \) measured at the unknown location \( \ell \).

In this work, we assume that the reference RSS data collected in the offline phase are not corrupted. This assumption is not restricting because reference data can be validated using security and attack prevention or detection mechanisms [22] prior to deploying the positioning system. Thus, we focus on non-cryptographic RSS-based attacks and failures that may occur during the online phase. In the following we introduce several fault models to capture the effect of AP malfunctions or malicious attacks during positioning. Then, we describe how these new models can be simulated using the original test data to allow extensive evaluation and comparison of fingerprinting algorithms. Each model is followed by a short discussion on the feasibility of the underlying attack or the occurrence probability of the relevant failure, in both private and public WLAN-based positioning systems.

Before describing the fault models let us define the Region of Coverage (RoC) of an AP, as the subset of reference locations where that particular AP is detected during the collection of reference RSS fingerprints. For instance, all reference locations in our experimental setup (small black dots), the locations inside the RoC (larger dots) of the AP named 2CU and the AP location (black triangle) are depicted in Fig. 1, while the grayscale colorbar indicates the mean RSS level from the specific AP at each location; see Section V for more details on the setup.

A. AP Failure Model

First, we consider the case where several APs used in the offline phase are not available during positioning. This can be caused by random AP failures, such as power outages, WLAN system maintenance, AP firmware upgrades etc. Regarding public positioning systems an AP listed in the database may be temporarily shut down or permanently removed by its owner. The latter case constitutes an AP failure during positioning from the user perspective until the database is updated. When an attack is assumed, an adversary can easily cut off the power supply of some APs or use specialized equipment to severely jam the communication channel to make the attacked APs unavailable. Jamming attacks can be easily launched, as reported in [23]. We simulate this model by removing the RSS values of the faulty APs in the original test fingerprints.

B. False Negative Model

In this model, the assumption is that an AP may no longer be detected in some locations inside its original RoC. This can happen accidentally if furniture or equipment is moved, so that the propagation path is blocked and the AP signal cannot be detected in locations where it was previously weak. Public WLAN positioning systems that cover large urban areas outdoors, can also be affected by this type of faults, e.g., construction of a building in the vicinity of that AP. We simulate this model by ignoring valid RSS readings for a set of APs in a number of test fingerprints.

C. False Positive Model

Another scenario is when an AP is detected during positioning in locations outside its original RoC. Contrary to the False Negative model, this can happen unintentionally in case a heavy object or equipment, which was previously obstructing the propagation path, was moved after collecting the reference data. Essentially, the transmission signals can travel further and make that AP hearable in locations outside its original RoC. An attack scenario that manifests in a similar manner is when a rogue AP is deployed and programmed to replicate an existing AP. In this fashion, the corrupted AP is thereafter detected during positioning in locations possibly far beyond its original RoC. The False Positive model is simulated by injecting random RSS values to the test data for a set of APs that would otherwise be undetected in those locations that the respective test fingerprints were collected.

D. AP Relocation Model

Our last model captures the effect of relocating a set of APs and thus a faulty AP is detected during positioning inside an area that can be different than the expected one. This may happen in case that the AP is moved to a new location, e.g., for network operation reasons, and the reference data are not updated by collecting additional fingerprints to cater for the affected areas. In public positioning systems, that expand over several office buildings and WLAN-equipped private properties, this may happen quite often. On the other
hand, an adversary may launch an attack with the same effect by physically relocating an AP. Alternatively, the attacker can impersonate a specific AP (Sybil attack), while at the same time eliminate the AP signals through jamming. We simulate the AP Relocation model by replacing the RSS readings of the corrupted AP in the test data with the values of another randomly selected AP.

AP impersonation can be easily implemented, especially in public positioning systems, because rogue APs can forge the MAC addresses of legitimate APs and transmit at arbitrary power levels within their physical capabilities. Details on the feasibility of impersonation and replication attacks can be found in [24], where the application of these attacks on the Skyhook [3] public WLAN positioning system is reported.

IV. NEAREST NEIGHBOR METHOD

Nearest Neighbor (NN) method estimates location by minimizing a distance metric $D_i$, such as the squared Euclidean distance, between the observed fingerprint during positioning $s$ and the reference fingerprints $r_i$.

$$\hat{l}(s) = \arg \min_{\ell_i} D_i, \quad D_i = \sum_{j=1}^{n} (r_{ij} - s_j)^2.$$  \hspace{1cm} (1)

Essentially, all reference locations are ordered according to $D_i$ and the location $\ell_i$ with the shortest distance between $r_i$ and $s$ in the $n$-dimensional RSS space is returned as the location estimate. The $K$ Nearest Neighbors (KNN) method estimates location as the mean of $K$ reference locations with the shortest distances and has been reported to provide higher level of accuracy compared to NN [4].

In practical implementations, WLAN APs provide only partial coverage in the area of interest and it is not expected that the sets of APs in fingerprints $r_i$ and $s$ will be identical. Thus, handling missing RSS values in one fingerprint or the other is a practical problem. Assuming fault-free positioning, a specific AP found in a fingerprint $r_i$ and not in $s$ can be due to the fact that $s$ is recorded in a location $\ell$ that is far from $\ell_i$. Even if $\ell$ and $\ell_i$ are spatially correlated, $s$ may not contain a RSS reading from that AP because of a transient effect in the WLAN adapter of the mobile device. Alternatively, if faults are also considered, then the missing AP can be the result of a random failure or a malicious attack during positioning.

Our objective is to study the performance of NN method under various fault models in order to get an insight of its inherent fault tolerance. To this end, we define $R_i$ and $S$ as the sets of APs that are present in fingerprints $r_i$ and $s$, respectively. Using these definitions, $D_i$ in (1) can be viewed as a distance metric that comprises the error contributions of three components

$$D_i = \sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in R_i \setminus S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij},$$ \hspace{1cm} (2)

where $d_{ij} = (r_{ij} - s_j)^2$. The first term refers to the intersection of $R_i$ and $S$ and represents the distance with respect to those APs that are common in fingerprints $r_i$ and $s$. The second term employs those APs that are detected in $r_i$ and not in $s$, while the last term increases the distance $D_i$ further by considering those APs that are found in $s$ and not in $r_i$.

When only few APs are common in $r_i$ and $s$, i.e. $|R_i \cap S|$ is small compared to $|R_i \setminus S|$ or $|S \setminus R_i|$, then this is a strong indication that these fingerprints are collected in distant locations. Thus, the second and third terms in (2) should be taken into account to increase $D_i$ and reflect the strong dissimilarity between fingerprints $r_i$ and $s$. This can be handled by using a small constant, e.g. below the sensitivity level of the WLAN adapter, to replace the missing RSS values $s_j$ and $r_{ij}$ in the second and third term of (2), respectively.

The distance metric in (2) is effective in the fault-free case because it penalizes all APs that are not found in common between $r_i$ and $s$. However, when AP faults or RSS attacks are considered, then this metric may not be able to guarantee the required fault tolerance. The effect of faults can be alleviated by using a median-based distance metric [11], instead of the Euclidean metric in (1). In this case, given the observed fingerprint $s$ and the reference fingerprints $r_i$, location is estimated by

$$\hat{l}(s) = \arg \min_{\ell_i} D_i, \quad D_i = \text{med}_{j=1}^{n} (r_{ij} - s_j)^2.$$ \hspace{1cm} (3)

V. MEASUREMENT SETUP

We collected our reference data in a typical modern office environment on the second floor of a three storey building at VTT Technical Research Centre of Finland. The floor consists of eight wings containing offices and meeting rooms connected with corridors. There are 31 Cisco Aironet APs installed throughout the building that use the IEEE 802.11b/g standard. We used a Fujitsu-Siemens Pocket Loox smart phone with Windows Mobile operating system to collect RSS measurements from all APs at 107 distinct reference locations on the second floor. These locations are separated by 2-3 meters and form a grid that covers all public places and meeting rooms.

A total of 3210 reference fingerprints, corresponding to 30 fingerprints per reference location, were collected at the rate of 1 sample/sec. Due to the open plan interior design, the APs can be partially detected on the second floor, and the average number is 9.7 APs per reference location. The floorplan of the experimentation area and the reference locations are depicted in Fig. 2, while the grayscale colorbar indicates the number of APs detected at each location. RSS values range from $-101$ dBm to $-34$ dBm and we used the value $-101$ dBm to handle the missing RSS values in the fingerprints. For testing purposes, we collected additional fingerprints on the second floor by walking at a constant speed over a path that consists of 192 locations. One fingerprint is recorded at each location, and the same path is sampled 3 times for a total of 576 test fingerprints.

VI. EXPERIMENTAL RESULTS

We use the collected RSS data to evaluate the fault tolerance of positioning methods and focus particularly on the NN method. In our setup, we found that using $K = 3$ neighbors
provides better performance in the fault-free case and this value is used throughout the experiments. For comparison, we consider the median-based KNN approach of [11], using $K = 3$ neighbors, denoted as medKNN hereafter. Moreover, we use a version of SNAP algorithm [21] adapted to the WLAN setup by utilizing information of whether an AP is detected during positioning or not. We also evaluate the probabilistic Minimum Mean Square Error (MMSE) positioning algorithm, which is based on the Kernel method described in [6].

The performance of the positioning methods is quantified using the mean positioning error ($M_e$) over all test fingerprints, i.e. the mean distance between the actual and estimated locations pertaining to the test data. We investigate fault tolerance with respect to the performance degradation, as the percentage of faulty (or attacked) APs is increased. Essentially, a method that exhibits smoother performance degradation is more fault tolerant. From another perspective, we may select a desirable upper bound on the performance, e.g. $M_e = 5$ m, and examine what is the percentage of corrupted APs that each positioning method can tolerate. We apply the fault models detailed in Section III to corrupt the original test data and the results for $M_e$ are averaged over 20 runs using randomly selected subsets of faulty APs in each run. For completeness we also consider the RSS attack models discussed in earlier works [12], [14] and study the resilience of the positioning methods.

In the fault-free case, MMSE method provides the best accuracy ($M_e = 2.46$ m), followed by KNN for which $M_e$ is 2.70 m. For medKNN, $M_e$ is 3.30 m which indicates that the median-based metric is not a good option when no faults are present. SNAP method has the worst performance ($M_e = 5.61$ m), however this is expected because SNAP only considers whether an AP is detected or not in the observed fingerprint and does not utilize its RSS level.

A. Performance under the AP Failure model

In Fig. 3a, $M_e$ is plotted as a function of the percentage of APs that have failed. In this scenario, medKNN provides the best performance in terms of fault tolerance, while KNN is only slightly worse. If 50% of the available APs are compromised, then $M_e$ is 7.39 m and 8.42 m for medKNN and KNN, respectively. These are followed by MMSE and SNAP methods for which $M_e$ is 9.61 m and 12.80 m, respectively. SNAP proves to be very sensitive to this type of faults and $M_e$ increases rapidly. On the other hand, medKNN can tolerate up to 30% failed APs, assuming $M_e = 5$ m, followed by KNN and MMSE that can tolerate up to 20% failed APs. Notice that even when 100% of the APs have failed, i.e. no AP is detected in any of the observed fingerprints, each positioning method can still provide a location estimate because we use the value $\pm 101$ dBm to handle the missing RSS values. In this case of course the estimate provided by each method cannot change to reflect the actual traveled path.

B. Performance under the False Positive model

This fault model can be viewed as a moderate case of the AP Failure model, in the sense that faulty APs may not be detected in some locations inside their original RoCs during positioning, but are not totally unavailable. We simulate this scenario by ignoring valid RSS values for a subset of the APs in 70% of the test locations (randomly selected), where the APs were previously detected. The performance of positioning methods for increasing number of faulty APs is illustrated in Fig. 3b. In case 50% of the APs are corrupted, then $M_e$ is 5.14 m, 6.04 m and 6.40 m for the medKNN, KNN and MMSE methods, respectively. For SNAP method results indicate that $M_e = 9.74$ m. Moreover, if the upper bound on the performance is 5 m, then medKNN exhibits higher fault tolerance as it can tolerate up to 45% faulty APs, compared to 35% for the KNN and MMSE methods.

C. Performance under the False Negative model

This fault model has exactly the opposite effect, i.e. an AP is detected during positioning in locations outside its original RoC. We simulate this scenario by injecting realistic random RSS values to a subset of the APs in 70% of the test locations (randomly selected), where those APs would otherwise be undetected. KNN and MMSE methods exhibit similar behaviour and their performance degrades rapidly as the percentage of faulty APs increases; see Fig. 3c. For instance, when 50% of the APs are affected then $M_e$ is around 15.50 m for both KNN and MMSE. In this case, medKNN provides the best performance ($M_e = 5.51$ m), followed by the SNAP method ($M_e = 7.93$ m). The medKNN method can tolerate 45% faulty APs, compared to only 15% for KNN and MMSE, while keeping the mean error below 5 m.

An interesting observation is that medKNN provides similar performance and has the same level of fault tolerance for both the False Positive and False Negative models. In contrast, if the False Positive model is assumed, KNN and MMSE methods prove to be significantly less fault tolerant compared to the False Negative model.

D. Performance under the AP relocation model

Faults injected according to the AP relocation model seem to cause similar performance degradation to all positioning methods. Experimental results under this fault model are
illustrated in Fig. 3d. All methods perform equally well when less than 30% of the APs are compromised. However, if the percentage of faulty APs exceeds 40%, then the medKNN method provides slightly better performance. Specifically, if half of the APs are corrupted, then $M_e$ is 9.20 m, 11.46 m, 11.70 m and 10.70 m for medKNN, KNN, MMSE and SNAP methods, respectively. Results indicate that medKNN, KNN and MMSE can tolerate around 25% corrupted APs without violating the 5 m upper bound on the performance.

E. Performance under the Linear Attack model

The Linear Attack model\cite{12} captures the effect of perturbing the RSS values during positioning by a constant factor. Interestingly, we found that KNN proves to be the most fault tolerant method, followed by MMSE, for both attenuation and amplification attacks. If the test RSS values are attenuated by 20 dBm for 50% of the APs, then $M_e = 5.15$ m and $M_e = 5.55$ m for KNN and MMSE, compared to $M_e = 7.03$ m for medKNN, as shown in Fig. 4a. Moreover, KNN and MMSE can tolerate up to 45% corrupted APs, compared to 30% for medKNN, in case $M_e = 5$ m is acceptable. The performance of SNAP degrades smoothly and when the percentage of corrupt APs increases beyond 60%, SNAP proves to be more fault tolerant compared to medKNN.

If RSS values are amplified by 20 dBm, then $M_e$ remains below 5 m for both KNN and MMSE methods even when all APs are corrupted; see Fig. 4b. SNAP also retains a high level of performance. On the other hand, medKNN method performs poorly and $M_e$ increases sharply, especially as the percentage of corrupted APs increases beyond 50%, for both attenuation and amplification attacks. For instance, under the attenuation attack model, $M_e$ increases at a rate of 7.5% when less than half of the APs are corrupted, while the rate is approximately 20% when the percentage of corrupted APs exceeds 50%.

F. Performance under the Additive Gaussian Noise model

In our experiments, this model is simulated by adding Gaussian noise with variance $\sigma_n$ to the original test RSS data. When $\sigma_n = 10$ dBm, there is only marginal performance degradation for KNN, and MMSE methods. For these methods results indicate that $M_e$ is increased around 1 m when all APs are corrupted compared to the fault-free case, contrary to medKNN for which $M_e$ is doubled; see Fig. 4c. For higher noise variance ($\sigma_n = 20$ dBm), performance degrades faster for KNN and MMSE, however $M_e$ remains below 6 m even when all APs are corrupted. For medKNN method, $M_e$ increases rapidly when more than half of the APs are corrupted, as shown in Fig. 4d. For instance, when 80% of the APs are corrupted $M_e = 7.94$ m for medKNN, compared to $M_e = 5.27$ m and $M_e = 5.42$ m for KNN and MMSE, respectively. When the percentage of faulty APs increases beyond 70%, then medKNN is outperformed by SNAP method as well.

G. Discussion

Our experimental results indicate that there is not a single positioning method that provides overall a high level of fault tolerance. It seems that various types of faults or RSS attack strategies require different approaches to maintain an adequate level of positioning performance. The SNAP method exhibits smooth performance degradation under almost all fault/attack models, however it is not a good candidate solution when few APs are corrupted, because the positioning error is rather high in the fault-free case. The probabilistic MMSE method exhibits slightly worse performance compared to KNN method and does not provide higher fault tolerance in any scenario.

The medKNN method is fault tolerant in some cases and provides the best performance under the False Negative, False Positive and AP Relocation fault models. However, its performance degrades significantly in case more than half of the APs are corrupted, especially under the RSS attack models, due to the median-based distance metric. On the other hand, the standard KNN method is very robust to RSS attacks; see Fig. 4. Moreover, KNN outperforms medKNN in the fault-free case. However, KNN does not perform equally well when the fault models introduced in Section III are considered. This implies that we could build an adaptive KNN method that selects either the Euclidean or the median distance metric. In this case, a detection mechanism is required in order to decide the type of fault (attack) and then trigger KNN algorithm to switch to the appropriate metric. This is part of our ongoing research on fault tolerant positioning.

Another direction is to modify the Euclidean metric of the KNN method in order to improve its fault tolerance. The distance metric in (2) is effective in the fault-free case, however when AP faults or RSS attacks are considered, then this metric may not be able to provide an adequate level of positioning accuracy. For instance, when AP failures occur during positioning, then a subset of the APs that would otherwise be present in the observed fingerprint $s$, are no longer detected. In this case, $|R_i \setminus S|$ becomes larger and errors in distances $D_i$ are increased due to the second term in (2). This leads to the wrong ordering of candidate locations and introduces high errors in the estimated location. Thus, this term could be removed from the distance metric in (2) to ignore faulty APs in $R_i \setminus S$.

VII. Conclusion

Our focus is on the fault tolerance of positioning methods, rather than the absolute accuracy in the fault-free case. AP malfunctions or malicious attacks lead to faults that manifest themselves in a similar fashion by corrupting RSS values during positioning. To this end, we introduced several models for the evaluation of RSS fingerprinting methods under the presence of faults or attacks. We investigated the fault tolerance of certain positioning methods and presented experimental results using real RSS measurements.

Our future work includes the modification of the distance metric in the standard KNN method in order to build KNN variants that will be tolerant to specific faults. We also plan to study and develop robust detection schemes to decide the type of fault (attack) and select the appropriate distance metric. Our objective is to build an adaptive KNN method that will
Fig. 3. Performance evaluation under various fault/attack models.

(a) AP Failure model.
(b) False Negative model.
(c) False Positive model.
(d) AP Relocation model.

Fig. 4. Performance evaluation under RSS attack models: (a) and (b) constant noise, (c) and (d) additive Gaussian noise.

(a) Linear Attack model with constant attenuation (−20 dBm).
(b) Linear Attack model with constant amplification (+20 dBm).
(c) Additive Gaussian Noise model ($\sigma_n = 10$ dBm).
(d) Additive Gaussian Noise model ($\sigma_n = 20$ dBm).
be fault tolerant and perform adequately under different types of faults or RSS attacks.

ACKNOWLEDGMENT

This work is supported by the Cyprus Research Promotion Foundation. Authors would like to thank P. Kemppi and Y. Li at VTT Technical Research Centre of Finland (www.vtt.fi) for the provision of experimental WLAN RSS data.

REFERENCES

Localization using Radial Basis Function Networks and Signal Strength Fingerprints in WLAN

C. Laoudias∗, P. Kemppi†, C. G. Panayiotou∗
∗KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering, University of Cyprus
Kallipoleos 75, P.O. Box 20537, 1678, Nicosia, Cyprus
Email: {laoudias, christos}@ucy.ac.cy
†VTT Technical Research Centre of Finland
Vuorimiehentie 3, P.O. Box 1000, FIN-02044, Espoo, Finland
Email: Paul.Kemppi@vtt.fi

Abstract—Fingerprinting localization techniques provide reliable location estimates and enable the development of location aware applications especially for indoor environments, where satellite based positioning is infeasible. In our approach we utilize Received Signal Strength (RSS) fingerprints collected in known locations and employ a Radial Basis Function (RBF) neural network to approximate the function that maps fingerprints to location coordinates. We present a clustering scheme to reduce the size and computational complexity of the RBF architecture and demonstrate the applicability of this approach in a real-world WLAN setup. Experimental results indicate that the RBF based method is an efficient approach to the location determination problem that outperforms existing techniques in terms of the positioning error.

I. INTRODUCTION

Localization techniques enable the provision of location information regarding people, mobile devices and equipment. Estimating location accurately is a challenge especially inside buildings, where satellite-based positioning is not applicable due to the severe attenuation or blockage of satellite signals. Positioning accuracy is the key issue to effectively support advanced indoor location aware services. Indicative applications include in-building guidance, asset tracking in hospitals or warehouses and autonomous robot navigation.

A wide variety of localization techniques based on radio signal propagation models have been studied [1] and several types of measurements that relate the position of a mobile device to the known positions of fixed transmitters can be utilized. These measurements include Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA) and Received Signal Strength (RSS). Even though relatively accurate propagation models are available for open areas, the presence of non line-of-sight paths between the receiver and the transmitter introduces additional measurement noise. Especially indoors, multipath conditions are common and lead to further accuracy degradation. An overview of technologies for wireless indoor location systems is provided in [2].

Fingerprinting techniques address the issue of multipath propagation by utilizing fingerprints collected a priori in the entire area of interest and stored in a database. These fingerprints are associated with a set of predefined reference points and contain location related information, such as RSS measurements or Power Delay Profile (PDP) parameters. The unknown location is estimated by using the currently measured fingerprint to find the best match in the database. Matching is based on a distance measure between the current and collected fingerprints or on probability distributions.

Artificial Neural Networks (ANN) [3] provide an alternative solution to the location determination problem. ANNs comprise a number of non-linear transformation units, i.e. neurons and a sufficiently large set of free parameters, i.e. interconnection weights. In this context, localization can be viewed as a function approximation problem. The ANN exploits the fingerprint database in order to approximate the function that maps the fingerprints from the high dimensional signal space to coordinates in the plane by interpolating the collected data. Our approach is based on a special class of ANNs, called Radial Basis Function (RBF) networks and we utilize location fingerprints that contain RSS measurements from several Access Points (AP) available in WLAN. The contribution of this work is the presentation of an efficient localization method based on RBF networks. We apply a clustering scheme to reduce the network size and computational overhead during the localization process. Experimental results indicate that the proposed method provides more accurate location estimates compared to other approaches. Moreover, the underlying RBF architecture is scalable and can be easily applied to different WLAN setups, in which variable number of APs, reference points or fingerprints may be available.

The rest of the paper is structured as follows. Previous work related to indoor localization using RSS fingerprints is discussed in Section II. The proposed method based on RBF networks is detailed in Section III. In Section IV we present the WLAN experimental setup used for our performance evaluation, followed by the results regarding the positioning accuracy. Finally, Section V provides the conclusions and discusses some ideas for future work.

II. RELATED WORK

Several solutions to the location determination problem using RSS fingerprints have been studied in the literature. These
approaches differ in the underlying localization algorithm, however they all rely on a RSS map that covers the entire area of interest. During the measurement campaign we use a set of predefined reference points $\ell_i = (x_i, y_i)$, $i = 1, \ldots, L$ to collect RSS values from $n$ neighboring APs. A reference fingerprint $s = [s_1, \ldots, s_n]^T$ is a vector of RSS samples and $s_j$ denotes the RSS value related to the $j$-th AP. A series of reference fingerprints $s(\ell_i, m)$, $i = 1, \ldots, L$ and $m = 1, \ldots, M$, is collected at each reference point and stored in a database followed by the physical coordinates $(x_i, y_i)$. In the localization step we exploit the reference data in order to obtain a location estimate $\hat{\ell}$ given a new fingerprint $s' = [s_{1}' , \ldots , s_n']^T$ measured at the unknown location $\ell$.

In the deterministic techniques location is estimated by minimizing an error function, e.g. the Euclidean distance between $\ell'$ and the reference fingerprints in the database. Localization time can be greatly reduced by preprocessing the reference fingerprints collected at $\ell_i$ to obtain the mean of the combined fingerprints $\bar{s}(\ell_i) = \frac{1}{M} \sum_{m=1}^{M} s(\ell_i, m)$. The Nearest Neighbor algorithm proposed in [4], [5] is based on this approach. The nearest neighbor is essentially the location with the shortest distance from $\ell'$ in the $n$-dimensional signal space. The $K$ Nearest Neighbors (KNN) variant estimates location as the centroid of $K$ locations with the shortest distances. The Database Correlation Method (DCM) introduces an additional term in the error function to penalize missing RSS values in the fingerprints [6].

On the other hand, probabilistic techniques estimate the unknown location by calculating the conditional probabilities $p(\ell_i | s')$. Applying Bayes rule the problem reduces to estimating $p(s' | \ell_i)$. Assuming that RSS measurements from neighboring APs are independent we get $p(s' | \ell_i) = \prod_{j=1}^{n} p(s_j' | \ell_i)$. The Kernel method introduced in [7] and the Histogram method presented in [7], [8] may be used to estimate $p(s_j' | \ell_i)$ from the reference data. Probabilistic techniques are reported to achieve higher positioning accuracy at the expense of increased computational complexity.

In the case of ANNs, the objective is to approximate the function that maps RSS fingerprints to locations. The reference fingerprints and corresponding coordinates $(x_i, y_i)$ are employed to train the network and adjust the weights accordingly. Subsequently, a location estimate is obtained when a new fingerprint $s'$ is presented to the network inputs. Recently, RBF networks have been discussed for indoor localization in Wireless Sensor Networks (WSN) by utilizing distance measurements [9] or series of successive RSS fingerprints [10]. Positioning techniques based on ANNs have also been applied to areas where WLAN infrastructure is available. Authors in [11] propose a Multi Layer Perceptron (MLP) network to perform localization using RSS measurements. A Generalized Regression Neural Network (GRNN) architecture, which is a RBF-type network with slightly different output layer, is evaluated in [12] using RSS values from three transmitters.

Our main contribution is that we provide well defined ways for setting the RBF parameters. The resulting RBF is small enough, easy to train, has good localization performance and avoids over-fitting. In [9], a small scale $(3 \times 3m)$ test bed with low noise and line-of-sight propagation conditions is used, and the standard RBF (sRBF) is reported to perform better than other algorithms. However, sRBF is prone to over-fitting and we show that in a realistic WLAN environment with noisy RSS measurements the proposed clustered RBF outperforms sRBF. In [10] the size of the RBF network is decided experimentally and even worse, it ends up being excessive compared to our approach. The MLP network [11] is significantly more difficult to train and must be retrained if new data become available. In addition, the size of the network needs to be determined experimentally. Finally, the GRNN method [12] uses a predefined way for setting the network weights, but loses some of the flexibility that other approaches have and thus it has lower performance.

### III. Radial Basis Function Networks

#### A. Function Approximation with RBF

In general, RBF networks are ANNs that have an input layer, a single hidden layer with non-linear radial, i.e. distance based, basis functions and an output layer. The architecture of a fully connected RBF network is depicted in Fig. 1. The input vector $x$ is provided as input to all radial basis functions and the output $f(x)$ is given by

$$f(x) = \sum_{i=1}^{C} w_i \varphi(||x - c_i||)$$

where $||x - c_i||$ is the Euclidean distance between $x$ and the $n$-dimensional basis function center $c_i$. The number of basis functions is $C$ and $w_i$ are the network weights. Usually the Gaussian radial basis function is used, i.e. $\varphi(||x - c||) = \exp (- \beta||x - c||^2)$. We can use RBF networks to approximate any continuous function by fitting the values of the function $f(x_i) = b_i$, $i = 1, \ldots, C$ at known points $x_i$. We set the centers of the basis functions equal to $x_i$ in (1) and then determine the weights by using $b_i$ and solving the system of linear equations. A RBF approximation example in one dimension is illustrated in Fig. 2.

#### B. Localization Method

In the context of indoor localization using fingerprints the RBF has $n$ inputs, corresponding to the RSS measurements.

![Fig. 1. Architecture of a Radial Basis Function network.](image-url)
from all \( n \) available APs and two outputs representing the 2-dimensional coordinates. Given a RSS fingerprint \( s = [s_1, \ldots, s_n]^T \) measured at location \( \ell = (x, y) \) the output of the RBF network may be expressed as a weighted sum of normalized Gaussian basis functions

\[
\ell(s) = \sum_{k=1}^{C} w_k u(\|s - c_k\|) = \sum_{k=1}^{C} w_k \frac{\varphi(\|s - c_k\|)}{\sum_{j=1}^{C} \varphi(\|s - c_j\|)}
\]

(2)

where \( w_k \) are 2-dimensional weights. Note that we have used normalized basis functions, since they provide improved performance compared to unnormalized functions, especially as the input dimensionality increases. The RBF network is used in order to approximate the function that maps RSS fingerprints to locations in physical space. The parameters \( c_k, \beta \) and \( w_k \) may be determined to obtain a good approximation by optimizing the fit between (2) and the reference data. Thus, we form the following set of equations

\[
\ell_i = \sum_{k=1}^{C} w_k u(\|s(\xi, m) - c_k\|), \ i = 1, \ldots, L, \ m = 1, \ldots, M
\]

(3)

We calculate \( w_k \) by solving the system of linear equations based on (3) using the reference fingerprints in the database and the corresponding coordinates. Subsequently, the weights \( w_k \) are used during localization to obtain a location estimate \( \hat{\ell} \) given a new fingerprint \( s' \) according to

\[
\hat{\ell}(s') = \sum_{k=1}^{C} w_k u(\|s' - c_k\|).
\]

(4)

Selecting the proper value for \( C \) and choosing the centers \( c_k \), is not trivial and affects the performance of the RBF network. In the standard RBF (sRBF) architecture each reference fingerprint defines a center \( c_k \) so that the total number of basis functions is \( C = L \cdot M \). In this case, the linear system has a unique solution and the sRBF design guarantees exact fitting for all reference data. However, this architecture has high memory requirements since all reference fingerprints, used as centers for the basis functions, are required to perform localization. Moreover, computational complexity is high both for the calculation of \( w_k \) and location estimation.

Several center selection schemes may be applied to reduce \( C \) and design more efficient RBF architectures. For instance, we may sample \( C < L \cdot M \) centers randomly among the reference fingerprints or select the desired number of centers by using the Orthogonal Least Squares learning algorithm [13]. An alternative solution is to obtain a good value for \( C \) experimentally, as in [10]. However, these approaches may be time consuming and/or do not always ensure the best data fitting. We adopt a clustering method to dramatically reduce \( C \), while maintaining adequate localization performance. In the proposed clustered RBF (cRBF) architecture the centers are set equal to the mean value fingerprints \( \bar{s}(\xi_i), \ i = 1, \ldots, L \).

In this case, \( C = L \) and the weights \( w_k \) are calculated in a least squares sense by solving the overdetermined system of equations based on (3).

The \( \beta \) parameter is also important in the cRBF approach since it determines the accuracy of the fit between the function approximation in (2) and the reference data. The value of \( \beta \) specifies the width of the Gaussian basis functions and allows their sensitivity to be adjusted. Decreasing \( \beta \) leads to wider basis functions such that there exists more overlap among them. Therefore, when a new fingerprint \( s' \) is present several basis functions will give fairly large outputs, leading to a more accurate location estimate. On the other hand, when \( \beta \) is larger reference data are fitted more sharply. In this case, the errors between the cRBF model and the reference data are minimized, however highly inaccurate location estimates may be obtained during localization. The appropriate \( \beta \) value is usually selected experimentally based on the reference data and can be further fine-tuned if testing data are available. However, this approach can be time consuming and a new value must be selected in case additional reference data are collected, e.g. when new reference points are used to cover more rooms. In the cRBF method a heuristic is used to set the width \( \beta \) according to

\[
\beta = \frac{1}{2d_{\max}}
\]

(5)

where \( d_{\max} = \max \|c_i - c_j\| \) for \( i, j = 1, \ldots, L \). The intuition is that when the distance among centers in the \( n \)-dimensional signal space increases, the value of \( \beta \) is reduced to ensure that the basis functions still overlap enough to produce accurate location estimates. With this scheme the value of \( \beta \) can be easily adjusted to provide high level of accuracy when a variable number of reference points, reference fingerprints or APs is used; see the experimental results in Section IV-B.

C. ANN Implementation Comparison

The cRBF method is more efficient compared to other ANN based methods. The reduced network size ensures that the weights \( w_k \) are obtained faster compared to the sRBF design. Moreover, the memory overhead for storing the basis function centers and weights, as well as the number of operations required for estimating a location, are reduced by a factor...
of $M$. Therefore, the proposed method based on cRBF is attractive when localization is performed on a mobile device with limited memory and processing power.

The GRNN design is similar to sRBF and each reference fingerprint defines a basis function center in the architecture. However, contrary to sRBF, the weights are set equal to the coordinates of the respective reference points in (4). GRNN calculates the distance between the fingerprint $s'$ and the centers, thus interpolating the location estimate $\hat{\mathbf{r}}$ according to the fingerprint database. In this approach no time is spent to obtain $w_k$, but the memory requirements and computational complexity during localization are the same with sRBF.

The MLP network uses the sigmoidal function $f(x) = 1 / (1 + \exp(-x))$ in the hidden layer. This function can have non-zero outputs over a large region of the input space, as opposed to Gaussian radial basis functions that respond to relatively small regions. Therefore, only a small set of weights is required to fit the reference data and memory overhead as well as localization time are significantly reduced. The major drawbacks in this approach is that the size of the MLP network can only be decided experimentally and long training time is needed to calculate the weights with the back propagation gradient descent learning algorithm. MLP training suffers from local minima, while in RBF networks linearity propagation gradient descent learning algorithm. MLP training time is needed to calculate the weights with the back propagation gradient descent learning algorithm. MLP training suffers from local minima, while in RBF networks linearity ensures that optimum weight values are easily found. Another disadvantage of the MLP is that it must be retrained in case additional reference fingerprints are collected, as opposed to RBF networks where retraining time can be greatly reduced by using appropriate matrix operations.

IV. PERFORMANCE EVALUATION

A. Measurement Setup

The localization trial was carried out in a typical office environment at the premises of VTT Technical Research Centre. The size of the 3-storey building is roughly 110m×45m. Each floor consists of 8 wings containing office rooms, open plan offices and meeting rooms connected with corridors. The measurement campaign was conducted in the second floor using 107 distinct reference points. These points are located 2-3 meters apart from each other and form a grid that covers all the public spaces and meeting rooms. There are 10 Cisco Aironet APs installed in this floor that use the IEEE802.11b/g standard. The reference points are depicted in Fig. 3 and the color scheme denotes the number of hearable APs at each reference point. We used a Fujitsu-Siemens Pocket Loox smart phone to collect RSS measurements with 1dBm resolution due to hardware limitations. Typical RSS values range from -101dBm to -34dBm in close proximity to an AP. We used a small constant (-110dBm) to handle the missing RSS values in the fingerprints. A total of 3210 reference fingerprints were collected, corresponding to 30 fingerprints per reference point and stored in the database with the physical coordinates. Fingerprints were also collected by following a predefined route that consists of 192 locations. One fingerprint is recorded at each location and the same route is sampled 3 times.

![Fig. 3. Reference points and number of hearable APs.](image)

B. Experimental Results

We evaluate the performance of the cRBF architecture in terms of the positioning error, defined as the Euclidean distance between the actual and estimated location. All fingerprints collected by sampling the same route multiple times are utilized to obtain accuracy results. Unless otherwise noted, fingerprints contain RSS values from $n = 10$ WLAN APs and all reference fingerprints, i.e. $L = 107$ and $M = 30$, are stored in the database and used by the localization methods.

We have implemented the MLP\(^1\), GRNN, sRBF and cRBF architectures, as well as the deterministic DCM algorithm and the probabilistic method (PROB). For DCM we used the 2 Nearest Neighbor approach, while PROB is based on the Kernel method described in [7]. Accuracy results are summarized in Table I and cRBF has the best localization performance according to the mean and median error. The standard deviation (Std) of the error is also low. This is followed by PROB, GRNN and DCM. The PROB method slightly improves accuracy in the given setup compared to DCM and does not justify the computational overhead. MLP and sRBF provide less accurate location estimates. These results indicate that cRBF is a promising localization method that performs better than other ANN approaches.

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>sRBF</th>
<th>cRBF</th>
<th>GRNN</th>
<th>DCM</th>
<th>PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>Max</td>
<td>23.7</td>
<td>24.0</td>
<td>13.1</td>
<td>17.2</td>
<td>21.4</td>
<td>15.8</td>
</tr>
<tr>
<td>Mean</td>
<td>4.6</td>
<td>4.6</td>
<td>3.4</td>
<td>3.9</td>
<td>4.0</td>
<td>3.8</td>
</tr>
<tr>
<td>Median</td>
<td>3.8</td>
<td>3.6</td>
<td>3.0</td>
<td>3.5</td>
<td>3.5</td>
<td>3.6</td>
</tr>
<tr>
<td>Std</td>
<td>3.7</td>
<td>3.6</td>
<td>2.2</td>
<td>2.5</td>
<td>2.8</td>
<td>2.4</td>
</tr>
</tbody>
</table>

\(^1\)The MLP network has ten inputs, two outputs and the implementation that gave the best results in the given setup consists of a single hidden layer with 30 sigmoidal functions.
The positioning error in the cRBF location estimates with respect to the value of $\beta$ is plotted in Fig. 4. If $\beta$ is large the basis functions are very narrow and reference data are fitted more sharply. The resulting implementation minimizes the error between the cRBF approximation and the reference data, however when a new fingerprint is presented only few basis functions respond and the estimates can be inaccurate. On the other hand, if $\beta$ is small the basis functions are wide and the reference data are fitted very smoothly. In this case, each basis function is essentially responding in the same, large region of the input space and the cRBF performance during localization is degraded. The curve in Fig. 4 indicates that a proper value may be selected in order to minimize the mean error in the location estimates. The heuristic introduced in Section III-B, based on the maximum distance among centers in the input space, ensures that the basis functions have the appropriate width. This is used in our experiments to set the value of $\beta$ accordingly.

In the following we further investigate the performance of the proposed cRBF method compared to DCM. The actual and estimated locations, obtained with cRBF and DCM, are depicted in Fig. 5 for a single route. The cRBF location estimates are very accurate and reflect the actual traveled route. The Cumulative Distribution Function (CDF) of the positioning error is plotted in Fig. 6. Accuracy of less than 5m is achieved for 80% of the cRBF estimates, while the mean error is 3.4m for cRBF and 4.0m for DCM. Approximately 5% of the DCM estimates have positioning error higher than 10m and the maximum error is 13.1m for cRBF and DCM, respectively. The cRBF method can alleviate high positioning errors in the location estimates and outperforms DCM in terms of estimation accuracy.

Collecting reference data to build the RSS map is a cumbersome task and it is a major restriction in fingerprint based localization techniques. There is a trade off regarding the time spent on data collection and the positioning accuracy. We address this issue by using a variable number of reference points or fingerprints and investigate the performance of cRBF method. First, we increase the number of reference points $L$, which are selected to cover the whole floor. This affects the size of the cRBF architecture and the localization performance is illustrated in Fig. 7. In all cases, cRBF achieves higher accuracy compared to DCM. For $L = 60$ the mean error in the cRBF estimates falls below 4m and when $L = 80$ the accuracy is approximately 3.5m. Increasing the value of $L$ further does not provide significant improvement.

The mean error achieved by DCM and cRBF, with respect to the number of reference fingerprints $M$, is depicted in Fig. 8. Even when only 5 or 10 fingerprints are collected at every reference point, the cRBF performs better and the mean error is 3.7m. By increasing the value of $M$ the improvement observed in the DCM accuracy is marginal. On the contrary, the performance of cRBF is improved by using more fingerprints and the error is about 3.4m if 25 fingerprints are used. This can be justified as follows. The number of reference fingerprints $M$ affects the center of each Gaussian basis function, which is set equal to the mean value fingerprint $\pi(\ell_i)$. The cRBF method is more sensitive to the value of distance between the new fingerprint $s'$ and each $\pi(\ell_i)$ compared to the DCM algorithm, due to the exponential basis function. Thus, when more fingerprints are available per reference point the centers are closer to their actual values and cRBF performs better.

Results in the previous experiments show that the proposed method can provide adequate level of accuracy without excessive data collection time requirements. The number of reference points can be reduced and few fingerprints can be collected without significant performance degradation. On the other hand, the number of APs in the area of interest affects the size of the input layer in the cRBF architecture. We use a variable number of APs $n$, which are selected to provide good coverage for the whole route; see Fig. 9. As expected, by increasing $n$ the positioning error is decreased since the fingerprints contain RSS measurements from more APs. By using 6-7 APs the positioning error is about 4m. The number of APs is related to the dimensionality of the signal space and the cRBF architecture can map RSS fingerprints to location coordinates more accurately.

V. CONCLUSIONS

The proposed cRBF algorithm is a promising fingerprint based technique that provides more accurate location estimates compared to other localization methods. Experimental results indicate that adequate level of accuracy is achieved even with limited number of reference points, collected fingerprints or installed APs. The proposed method is scalable and can be easily applied to other environments. The heuristic used to set the value of $\beta$ ensures that under different conditions the cRBF design interpolates the reference data smoothly and performs well during localization.

Our future work includes the application of the cRBF approach to other WLAN environments in order to verify the localization performance. We also plan to integrate this method into an indoor positioning platform. We emphasize that the proposed cRBF is a ANN design which is simple to train and is very efficient after training, thus it can be easily incorporated into real time applications. Both, network and user-plane positioning architectures can be used. In a network-based architecture, a mobile terminal sends the currently observed RSS fingerprint to the positioning server which responds with the location coordinates. However, we envision a system where
the previously collected fingerprints will be used to train the cRBF off-line and determine the weights. Once a terminal enters an area, it receives from a server this small set of weights and using the currently measured fingerprints it can localize and track itself independently thereafter.

ACKNOWLEDGMENT

This work is partly supported by the Cyprus Research Promotion Foundation under contract ENIΣΧ/0506/59 and the European Science Foundation (ESF) in the framework of the Middleware for Network Eccentric and Mobile Applications (MiNEMA) activity.

REFERENCES


Indoor Localization using Neural Networks with Location Fingerprints

Christos Laoudias\textsuperscript{1}, Demetrios G. Eliades\textsuperscript{1}, Paul Kemppi\textsuperscript{2}, Christos G. Panayiotou\textsuperscript{1}, and Marios M. Polycarpou\textsuperscript{1}

\textsuperscript{1} KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering, University of Cyprus
Kallipoleos 75, P.O. Box 20537, 1678, Nicosia, Cyprus
\{laoudias,eldemet,christosp,mpolycar\}@ucy.ac.cy

\textsuperscript{2} VTT Technical Research Centre of Finland
Vuorimiehentie 3, P.O. Box 1000, FIN-02044, Espoo, Finland
Paul.Kemppi@vtt.fi

Abstract. Reliable localization techniques applicable to indoor environments are essential for the development of advanced location aware applications. We rely on WLAN infrastructure and exploit location related information, such as the Received Signal Strength (RSS) measurements, to estimate the unknown terminal location. We adopt Artificial Neural Networks (ANN) as a function approximation approach to map vectors of RSS samples, known as location fingerprints, to coordinates on the plane. We present an efficient algorithm based on Radial Basis Function (RBF) networks and describe a data clustering method to reduce the network size. The proposed algorithm is practical and scalable, while the experimental results indicate that it outperforms existing techniques in terms of the positioning error.

Key words: Localization; WLAN; Fingerprinting; Received Signal Strength; Radial Basis Function Networks

1 Introduction

Localization techniques are used in order to determine the position of people, mobile devices and equipment. The provision of reliable location estimates is a challenge, especially indoors where satellite-based positioning is infeasible. Positioning accuracy is the key issue to effectively support indoor location aware services. Indicative applications include in-building guidance, asset tracking in hospitals or warehouses and autonomous robot navigation.

A wide variety of localization techniques have been discussed in the literature and can be categorized according to the type of measurements employed in the underlying positioning algorithm. Location is estimated with angle, timing or signal strength measurements from a number of transmitters with known locations by utilizing radio signal propagation models [1]. However, the presence of non line-of-sight paths between the receiver and the transmitter can cause accuracy degradation. Especially indoors, where multipath conditions are prevalent,
model inaccuracies may lead to high positioning errors; see [2] for an overview of technologies for wireless indoor location systems.

Localization performance in indoor environments can be improved by utilizing a premeasured map of Received Signal Strength (RSS) measurements. In this case, a set of predefined locations is associated with vectors containing RSS values from neighboring transmitters. These vectors, referred to as location fingerprints, are collected offline and stored in a database followed by the location coordinates. The unknown location can then be estimated on line from the current RSS fingerprint by finding the best match in the database. Matching is based on a distance measure between the current and collected fingerprints or on probability distributions [3–5].

Artificial Neural Networks (ANN) have been proposed as a solution to the location determination problem [6–10]. We adopt ANNs in a function approximation scheme to map RSS fingerprints in the high dimensional input signal space to locations in the physical space. In the envisioned indoor localization system data collected offline is used to train the ANN. Subsequently, when a mobile device running a location-based application enters a building, it receives the parameters of the trained ANN and is enabled to localize itself by using the currently measured RSS fingerprints. As a result, it is desirable to have an ANN-based algorithm that is computationally efficient and requires a small set of parameters in order to keep the communication cost low. Moreover, the ANN needs to be easily retrained in case the information in the database is outdated or new data become available. In this context, we evaluate different ANN models namely the Multi Layer Perceptron (MLP), Radial Basis Function (RBF) and Generalized Regression Neural Network (GRNN) for the implementation of the localization method.

The rest of this paper is structured as follows. In Section 2 the problem is defined and we briefly describe the related work in this area. In Section 3 we present the WLAN experimental setup used to conduct the measurements. The ANN designs and the proposed method based on RBF are detailed in Section 4. In Section 5 the positioning accuracy results are presented and we discuss the advantages and drawbacks of the ANN implementations. Finally, Section 6 provides concluding remarks and discusses future work.

2 Indoor Localization Overview

2.1 Problem Formulation

We introduce the theoretical framework for localization techniques based on fingerprints, assuming a WLAN infrastructure and availability of RSS measurements from neighboring Access Points (AP). Let $D \subseteq \mathbb{R}^2$ be a 2-dimensional physical space denoting the area of interest. We define the finite set of locations $\mathcal{L} \subseteq D$ known as reference points, where $\mathcal{L} = \{ \ell_i \in D | \ell_i = (x_i, y_i), i = 1, \ldots, L \}$. At each location $\ell_i \in \mathcal{L}$ a mobile device is used to collect RSS measurements from $n$ neighboring APs. Thus, we form an $n$-dimensional input space denoted
Indoor Localization using Neural Networks 3

by $S$. A reference fingerprint $s \in S$ is a vector of RSS measurements collected at location $\ell_i$, i.e. $s = [s_1, \ldots, s_n]^T$ and $s_j$ denotes the RSS value related to the $j$-th AP. The reference points, can be placed over a uniform grid to cover the entire area with the desired resolution. However, the grid is usually non uniform and sparse due to building walls, furniture and other objects that limit the area where measurements can be performed. At each reference point $\ell_i \in L$ we collect a series of fingerprints $s(\ell_i, m)$, $m = 1, \ldots, M$ and thus the database contains $R = L \cdot M$ fingerprints followed by the respective location coordinates.

We also define the mean value fingerprint $s(\ell_i) = \frac{1}{M} \sum_{m=1}^{M} s(\ell_i, m)$. During localization the goal is to obtain an estimate denoted as $\hat{\ell}$, given a fingerprint $s' = [s'_1, \ldots, s'_n]^T$ that is measured at the unknown location.

### 2.2 Fingerprinting Techniques

Several approaches have been discussed for indoor localization using RSS fingerprints that are briefly described next.

In the deterministic approach, $\hat{\ell}$ is obtained by minimizing a given norm of the difference between $s'$ and the reference fingerprints. The Nearest Neighbor method introduced in [3], assumes the Euclidean distance as the optimization criterion and thus $\hat{\ell}(s') = \arg \min_{\ell_i} \|s' - \bar{s}(\ell_i)\|_2^2$. In this case $\hat{\ell} \in L$. The $K$ Nearest Neighbors (KNN) variant [3] determines $\hat{\ell} \in D$ as the centroid of the $K$ locations with the shortest distances between $s'$ and the mean value fingerprints. Weighted versions of the KNN algorithm have also been proposed.

From a probabilistic point of view, location is determined by calculating the conditional probabilities $p(\ell_i|s')$, $\forall \ell_i \in L$. Then, the estimated location $\hat{\ell} \in L$ may be obtained by $\hat{\ell}(s') = \arg \max_{\ell_i} p(\ell_i|s')$, as in [4]. Alternatively, authors in [5] calculate the expected value of the location variable $\ell$, i.e. $\hat{\ell}(s') = \mathbb{E}[\ell|s'] = \sum_{i=1}^{L} \ell_i p(\ell_i|s')$, in order to obtain the Minimum Mean Square Error (MMSE) estimate $\hat{\ell} \in D$. By application of Bayes rule the problem reduces to calculating $p(s'|\ell_i)$. Assuming that RSS measurements from neighboring APs are independent we get $p(s'|\ell_i) = \prod_{j=1}^{n} p(s'_j|\ell_i)$. Different methods have been proposed to estimate $p(s'|\ell_i)$ by utilizing the fingerprints in the database, namely the Kernel and Histogram methods [4, 5]. In general, probabilistic techniques achieve higher positioning accuracy compared to the deterministic ones, at the expense of increased computational complexity.

### 2.3 Artificial Neural Network Approaches

In the context of ANNs, localization can be viewed as a classification problem. Each reference point defines a class and in this case the output of the ANN is one of the reference points $\ell_i \in L$. Authors in [6] extend this approach by using a $L \times 1$ vector output for the network. The vector output provides the probability of $s'$ belonging to each class and $\hat{\ell} \in D$ is obtained as the weighted mean of the respective reference points.
Alternatively, estimating current location can be viewed as a function approximation problem. The objective is to find a mapping $F(s) : S \rightarrow D$ of RSS fingerprints onto locations in the physical space. Recently, RBF networks have been discussed for localization in Wireless Sensor Networks (WSN). In [7] distance measurements from three beacon nodes, instead of RSS fingerprints, are utilized to evaluate the performance of RBF networks and compare to MLP and Recurrent Neural Networks (RNN) architectures. In [8] location is estimated with a RBF network using RSS measurements, however the number of neurons in the hidden layer is decided experimentally and can be very high thus increasing the computational cost. Positioning techniques based on ANNs have also been applied to areas where WLAN infrastructure is available. Authors in [9] propose a MLP architecture with a single hidden layer to perform localization using RSS measurements from three WLAN APs. A GRNN architecture, which is a RBF-type network with slightly different output layer, is proposed in [10] to determine location using RSS values from three transmitters.

In this paper we evaluate MLP, RBF and GRNN architectures using RSS measurements from ten WLAN APs. We focus on RBF networks and discuss a clustering method to reduce the size of the hidden layer and improve the computational complexity. In addition, this approach alleviates some of the overtraining problems of standard RBF networks. Experimental results indicate that the proposed clustered RBF design outperforms the deterministic approach and provides higher level of accuracy compared to MLP and GRNN.

3 Experimental Setup

The localization trial was carried out in a typical modern office environment at the premises of VTT Technical Research Centre in Espoo, Finland. The measurement campaign was conducted in the second floor of the 3-storey building, where $n = 10$ Cisco Aironet APs that use the IEEE802.11b/g standard are installed. We developed a Site Survey software that utilizes a floorplan map to mark $L = 107$ distinct reference points located 2-3 meters apart from each other in order to cover all public spaces and meeting rooms; see Fig. 1. RSS samples were collected with 1dBm resolution by using a WLAN-enabled smart phone. This resolution, though accurate enough for some applications, it introduces some “quantization” error since two locations that are very close to each other cannot be distinguished. This resolution depicts the lower bound of the error that any localization technique (using RSS measurements) can achieve. Typical RSS values range from -101dBm to -34dBm in close proximity to an AP. In case an AP was not hearable at a reference point, a small constant (-110dBm) was used to handle the missing RSS values in the fingerprints.

We have measured 30 fingerprints per reference point and selected randomly $M = 25$ out of these, corresponding to a total of $R = 2675$ fingerprints, which are stored in the database. This is our training set, while the remaining 5 fingerprints per reference point are kept as a test set for the performance evaluation of the ANN architectures described in Section 4. Additional fingerprints were
also collected independently of the training set during a separate measurement campaign to form a second test set by following a predefined route that consists of 192 locations. One fingerprint was recorded at each location and the same route was sampled 3 times.

4 Artificial Neural Network Architectures

4.1 Multi Layer Perceptron (MLP)

The fully connected MLP network has ten inputs, corresponding to the RSS measurements from all available APs, while the output linear layer has two neurons representing the location coordinates \((x, y)\). We use the sigmoidal transfer function for neurons in the single hidden layer. The size of the hidden layer was decided experimentally trying to keep it as small as possible, while preserving an adequate level of positioning accuracy. Specifically, we reserved 20% of the training fingerprints as a validation set and the network that achieved the best performance on this set was selected. The synaptic weights \(w\) were determined with the standard back propagation algorithm and the validation set was used as an early stopping method to avoid overfitting the training data. We have also investigated the use of a separate single output MLP network for each coordinate, \(x\) and \(y\). We employed the same validation procedure in order to decide the network sizes, however the performance of this combination of two MLP designs on the validation set was degraded. The MLP architecture considered in Section 5 has 20 and 2 neurons in the hidden and output layers, respectively. We point out that training of the MLP is rather time consuming and the network must be retrained in case new data becomes available.

4.2 Radial Basis Function (RBF)

We examine a fully connected RBF network to approximate \(F(s) : S \rightarrow D\) and use the normalized Gaussian function for neurons in a single hidden layer. The
network has ten inputs and two outputs. Given a fingerprint $s'$, the estimated location $\hat{\ell}$ is given by

$$\hat{\ell}(s') = F(s') = \frac{\sum_{k=1}^{C} w_k \varphi(||s' - c_k||)}{\sum_{j=1}^{C} \varphi(||s' - c_j||)}$$

where $\varphi(||s' - c_k||) = \exp\left(-\beta ||s' - c_k||^2\right)$. The number of neurons in the hidden layer is $C$, $c_k$ is the 10-dimensional center for neuron $k$, and $w_k$ are the 2-dimensional weights for the linear output layer. The value of $\beta$ must be appropriately selected to ensure that the Gaussian basis functions are wide enough and the resulting RBF architecture implements a smooth approximation $F(s)$.

In the GRNN architecture each reference fingerprint defines a center $c_k$, i.e. $C = R$. The weights $w_k$ in (1) are set equal to the coordinates of the respective reference points and in that sense $\hat{\ell}$ is the weighted average of the reference points whose fingerprints are closest to $s'$.

However, the weights $w_k$ can be determined in order to optimize the fit between $F(s)$ and the reference data. Thus, one may select the centers $c_k$ and the width $\beta$ and then form the following set of equations

$$(x_i, y_i) = \sum_{k=1}^{C} w_k \varphi(||s_i - c_k||), \quad i = 1, \cdots, L \text{ and } m = 1, \cdots, M$$

In the standard RBF network (sRBF) each reference fingerprint defines the center of a neuron. In this case, the system of linear equations based on (2) can be written in matrix form as $Uw = d$, where $U = \{u(||s_j - c_i||) | (j, i) = 1, \cdots, R\}$ and $u(\cdot)$ is the normalized Gaussian basis function given in (1). Matrix $d$ contains the coordinates of the reference points and the weights are easily obtained by $w = U^{-1}d$.

The number of neurons in the hidden layer can be reduced dramatically by application of a clustering technique on the reference fingerprints. In the clustered RBF (cRBF) architecture each center is set equal to the mean value fingerprint $\bar{s}(\ell_i)$. Thus, $C = L$ and the weights are calculated in a least squares sense by solving the overdetermined system of equations based on (2). The minimum-norm solution for the weight vector is $w = U^{+}d$, where $U^{+}$ is the pseudoinverse of matrix $U$ defined as $(U^{T}U)^{-1}U^{T}$.

The sRBF design guarantees exact fitting for reference data at the expense of increased hidden layer size. Moreover, it is well known that sRBF is prone to overfitting and exhibits inadequate generalization capabilities. In [7] it is reported that sRBF outperforms cRBF for the localization problem and the positioning error is in the order of few cm. However, in that work the evaluation was conducted in a small scale (3×3 m) experimental test bed with line-of-sight conditions using low noise distance measurements from 3 beacon nodes. Under realistic propagation conditions the accuracy of sRBF is degraded. We consider the sRBF design in our evaluation to verify that when noisy RSS measurements collected in a real-life WLAN environment are utilized, its performance is poor compared to the clustered counterpart.
5 Results and Discussion

The MLP, sRBF, cRBF and GRNN designs are compared in terms of the positioning error, defined as the Euclidean distance between the actual and estimated location. We have implemented the deterministic KNN localization method [3] and use it as baseline for our evaluation. In our experimental setup, the value $K = 2$ provides the lowest positioning error and is therefore selected for the rest of the experiments.

5.1 Test Case 1

The first test set comprises 535 RSS fingerprints in total, i.e. 5 test vectors per reference point and has the same statistical distribution of positions as the training set. Table 1 summarizes the accuracy results. The MLP, GRNN and the proposed cRBF architectures are equivalent regarding the mean and median positioning error. The results also indicate that the error in half of the location estimates derived with the KNN algorithm is below 1.9m. This is lower compared to MLP, GRNN and cRBF, however a considerable fraction of the KNN estimates exhibit error higher than 10m leading to the same level of accuracy as far as the mean error is concerned. As expected the sRBF design achieves the highest level of accuracy for the given test set. Note that during the data collection process the RSS level in some reference points may not vary much for certain APs and duplicate fingerprints are recorded. Therefore, there is a high probability that exactly the same fingerprint is present in both the training and test sets. The sRBF design guarantees exact fitting for training data and for this reason the median error is zero. However, sRBF is prone to overfitting and its poor generalization capabilities are depicted in the maximum positioning error; even moderate deviation from the training fingerprints leads to significant accuracy degradation.

5.2 Test Case 2

The RSS fingerprints in the second test set are measured by walking inside the area of interest. Note that most of the unknown locations do not coincide with any reference point. Location estimates obtained with the MLP network for a single route are depicted in Fig. 2 (dots), while the black line denotes the actual route. The estimated locations for the same route using the sRBF and cRBF designs are illustrated in Fig. 3 and Fig. 4, respectively. The increased number of neurons in the sRBF network results in worse localization performance and the estimates do not reflect the traveled route. The sRBF network is overtrained and has essentially learned the noise in the reference fingerprints. Even when a smaller value is used for $\beta$, in order to increase the width of the Gaussian function and create a smoother approximation $F(s)$, sRBF fails to accurately locate the user when new fingerprints are presented to the network. GRNN location estimates are depicted in Fig. 5 and higher accuracy is achieved compared to MLP and sRBF networks.
Positioning error statistics pertaining to the second test set that contains all fingerprints collected after sampling the same route 3 times are tabulated in Table 2. The cRBF design has the best localization performance according to the mean and median error. The standard deviation (Std) of the error is also low and cRBF is the only network that outperforms the KNN algorithm. This is followed by the GRNN architecture that achieves the same level of accuracy as KNN. The sRBF design provides less accurate location estimates compared to cRBF, GRNN and KNN. Finally, the MLP network exhibits the worst performance for the given test set and the maximum error is surprisingly high.

The MLP has very low memory requirements for storing the network weights and biases and essentially the fingerprint database is compressed into a small set of parameters. Moreover, the MLP is the least computationally intensive, due to the small number of neurons. In the sRBF and GRNN designs, the weights and all reference fingerprints are required to perform localization, while they
Table 1: Test Case 1.

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>sRBF</th>
<th>cRBF</th>
<th>GRNN</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Max</td>
<td>10.0</td>
<td>14.3</td>
<td>9.1</td>
<td>8.2</td>
<td>12.2</td>
</tr>
<tr>
<td>Mean</td>
<td>2.7</td>
<td>1.9</td>
<td>2.6</td>
<td>2.7</td>
<td>2.5</td>
</tr>
<tr>
<td>Median</td>
<td>2.4</td>
<td>0.0</td>
<td>2.3</td>
<td>2.4</td>
<td>1.9</td>
</tr>
<tr>
<td>Std</td>
<td>1.7</td>
<td>3.0</td>
<td>1.8</td>
<td>1.8</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Table 2: Test Case 2.

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>sRBF</th>
<th>cRBF</th>
<th>GRNN</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Max</td>
<td>29.4</td>
<td>24.0</td>
<td>13.1</td>
<td>17.2</td>
<td>21.4</td>
</tr>
<tr>
<td>Mean</td>
<td>5.3</td>
<td>4.6</td>
<td>3.4</td>
<td>3.9</td>
<td>4.0</td>
</tr>
<tr>
<td>Median</td>
<td>4.2</td>
<td>3.6</td>
<td>3.0</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Std</td>
<td>4.4</td>
<td>3.6</td>
<td>2.2</td>
<td>2.5</td>
<td>2.8</td>
</tr>
</tbody>
</table>

exhibit longer estimation time compared to MLP due to the increased network size. Problems related to storage memory and localization time can be alleviated by adopting the cRBF architecture. Nowadays, the memory and computational overhead of all these ANN architectures can be well handled by high-end mobile devices. However, the transmission overhead to communicate the ANN parameters to the device through the WLAN is significant, thus rendering the MLP and the proposed cRBF designs the best candidate solutions.

The practicality and scalability of each ANN architecture are also critical issues. For instance, the MLP requires long training time, while the back propagation algorithm suffers from local minima and does not guarantee optimum weight values. Moreover, the MLP must be retrained in case additional fingerprints are collected at new reference points to cover more rooms. Another disadvantage is that the size of the MLP can only be decided experimentally and it is not clear how the MLP will scale for different number of inputs, e.g. using measurements from less than 10 APs. On the other hand, the cRBF network can be trained faster by solving a linear system of equations, while linearity ensures that optimum weight values are found. The structure and size of the neural network can be decided in a principled manner when the cRBF design is used, thus increasing its applicability to other environments. In case new reference fingerprints are available the size of the cRBF network is easily decided, while retraining time can be greatly reduced by using appropriate matrix operations.

6 Conclusions

We have evaluated several ANN designs to perform indoor localization by exploiting RSS fingerprints collected in a typical office environment. We rely on WLAN infrastructure to minimize the deployment cost since no specialized equipment is required. The proposed cRBF algorithm is a promising solution to the location determination problem that can be easily scaled and applied to other indoor environments with WLAN coverage. The mobile device needs to receive only a small number of parameters through the WLAN in order to start locating itself inside the building. Moreover, experimental results indicate that the cRBF achieves higher level of accuracy compared to the sRBF, MLP and GRNN designs, as well as the deterministic KNN algorithm.

Future work will focus on further improving the cRBF approach by using a variable selection procedure in order to limit the area where the user may reside...
and determine which APs to use in the localization process. We also plan to use an appropriate network regularization method and variable $\beta$ values in the Gaussian basis functions, based on the distribution of centers in the multidimensional signal space, to achieve higher accuracy.

Acknowledgments. This work is partly supported by the Cyprus Research Promotion Foundation under contract ENIΣΧ/0506/59 and the European Science Foundation (ESF) in the framework of the Middleware for Network Eccentric and Mobile Applications (MiNEMA) activity.

References

Ubiquitous Terminal Assisted Positioning Prototype

C. Laoudias*, C. Desiniotis†, J. Pajunen‡, S. Nousiainen‡, C. Panayiotou* and J. G. Markoulidakis†

*University of Cyprus, Department of Electrical and Computer Engineering
Email: laoudias@ucy.ac.cy, christosp@ucy.ac.cy
†Vodafone-Panafon Greece, Technology Strategic Planning - R&D Department
Email: christer@telecom.ntua.gr, Yannis.Markoulidakis@vodafone.com
‡VTT Technical Research Center of Finland, VTT Information Technology
Email: Juuso.Pajunen@vtt.fi, Sami.Nousiainen@vtt.fi

Abstract—Statistical Terminal Assisted Mobile Positioning is a methodology that enhances the performance of existing localization techniques, by exploiting historical measurements collected at the terminal side. We propose a unified positioning framework in which different types of network related measurements may be employed by multiple techniques to derive coarse position estimates, while filtering is applied as a post processing step to further increase accuracy. The proposed architecture is applicable to User Plane location architectures, while its open and modular design allows easy integration of new positioning algorithms and post processing techniques. The implementation of this concept in Ubiquitous Terminal Assisted Positioning prototype concentrates particularly on Quality of Position issues and compatibility with currently available standards and communication protocols.

I. INTRODUCTION

Location Based Services (LBS) include personalized applications offered to the mobile user through the identification of his/her current location. During the late 90’s LBS did not get widely accepted for many reasons such as the lack of standards, poor content quality, low network throughput, customer perception issues and inadequate tracking performance. However, nowadays most obstacles have been overcome, while the market has become more mature to accept the advent of advanced applications. This is also the result of the deployment of next generation wireless networks and standardization activities carried out by the 3rd Generation Partnership Project (3GPP) and Open Mobile Alliance (OMA).

So far, a wide variety of positioning techniques has been proposed, which are characterized by a trade-off regarding the accuracy achieved, the applicability to legacy terminals and the relevant deployment costs. These techniques employ signal strength and angle or time of arrival measurements to perform positioning and are often considered as short term choices or fallback solutions in hybrid schemes based on GPS; see [1] for an overview. Further to the most common positioning techniques, more sophisticated methods appear in the literature. They rely on time processing of noisy measurements in the form of filtering, which can alleviate high positioning errors by incorporating proper mobility models. Filtering including the well studied Kalman filters [2], can be applied on the raw measurements or on a series of coarse position estimates derived through standard techniques. In the case of raw measurements, the performance of the underlying positioning algorithm is greatly improved by estimating the terminal’s motion, from noisy observations. Different variants of Bayesian filters [3] have been discussed in the literature, including Extended Kalman Filter (EKF) [4] and Particle Filter [5]. In the case of coarse position estimates, filtering is employed as a post processing technique that generates a smoothed location sequence to reflect the terminal’s mobility pattern more accurately. Popular post processing techniques include Kalman Filter [6], [7] and Particle Filter [8].

Statistical Terminal Assisted Mobile Positioning (STAMP) is an advanced localization methodology that is generic and applicable to legacy cellular networks and to Beyond 3G (B3G) heterogeneous environments, where multiple Radio Access Technologies (RAT) such as GSM, UMTS and WLAN coexist. The main idea of STAMP is the utilization of historical position estimates derived by any positioning technique [9]. Location related information, such as Received Signal Strength (RSS) values and timing measurements from the available access network, are collected and stored during the standard operation of the Mobile Terminal (MT). When an LBS session is established the list of measurements is uploaded to the Positioning Server and then exploited through multiple positioning techniques to provide estimates of the history of the MT’s motion. As a final step statistical processing, with the aid of Kalman filtering, is employed to derive a more accurate estimate of the current MT location. At the same time tracking capabilities and velocity estimation may also be provided.

The practical aspects of implementing STAMP, as part of the Ubiquitous Terminal Assisted Positioning (UTAP) system, are investigated in the context of MOTIVE project [10]. The project aims to deliver a positioning system prototype that will serve as a complete platform to enable LBS deployment. The proposed architecture is compatible with User Plane architectures, based on the emerging Secure User Plane Location (SUPL) standards defined by OMA [11]. Location related information collected by the MT is transferred through data packets over secure IP connections, independently of the access technology and network infrastructure, thus minimizing network modifications. However, an agent is required at the MT side to handle the communication exchange. Additional positioning techniques can be easily integrated into this platform as add-on components. Quality of Position (QoP) is provided according to the application requirements and the technique selected to perform localization. QoP is treated as a...
set of attributes, such as the accuracy of the position estimate and the system response time, associated with a request for the MT’s location.

This paper provides a description of the UTAP system prototype. The abbreviations, commonly used in literature, as well as the OMA related acronyms are tabulated at the end of this Section. The STAMP concept is introduced in Section II, followed by the platform requirements and the proposed architecture in Section III. The UTAP system prototype and the current implementation status are presented in Section IV. Section V outlines the results of applying STAMP in commercial GSM and WLAN networks with some indicative positioning scenarios. Finally, Section VI provides concluding remarks and discusses future work related to the UTAP system.

### Acronym List

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CGI++</td>
<td>Cell Global Identity + received signal strength measurements</td>
</tr>
<tr>
<td>CPICH</td>
<td>Common Pilot Channel</td>
</tr>
<tr>
<td>DCM</td>
<td>Database Correlation Method</td>
</tr>
<tr>
<td>DM</td>
<td>Device Management</td>
</tr>
<tr>
<td>DOP</td>
<td>Dilution of Precision</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communication</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Service</td>
</tr>
<tr>
<td>MLP</td>
<td>Mobile Location Protocol</td>
</tr>
<tr>
<td>MT</td>
<td>Mobile Terminal</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>PS</td>
<td>Positioning Server</td>
</tr>
<tr>
<td>QoP</td>
<td>Quality of Position</td>
</tr>
<tr>
<td>RAT</td>
<td>Radio Access Technology</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>STAMP</td>
<td>Statistical Terminal Assisted Mobile Positioning</td>
</tr>
<tr>
<td>SUPL</td>
<td>Secure User Plane Location</td>
</tr>
<tr>
<td>ULP</td>
<td>Userplane Location Protocol</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UTAP</td>
<td>Ubiquitous Terminal Assisted Positioning</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>

---

### II. OVERVIEW OF STAMP CONCEPT

The STAMP concept is depicted in Fig.1 for a MT moving through an area covered by multiple RATs. The MT periodically stores all available network measurements while in idle mode, thus forming a list of location related information, denoted as the STAMP List. Each entry in the STAMP List contains a vector with the actual measurements and a special field, which indicates the corresponding type of access technology. In this paper GSM, UMTS and WLAN RSS measurements are considered. However, this can be extended in order to include available measurements required by any positioning technique. All entries are time-stamped, thus rendering the STAMP List, a record of historical information reflecting the MT’s motion. When the STAMP List is full, updating is performed in a sliding window fashion by discarding the oldest and incorporating the current measurement vector. A set of parameters is constantly monitored, as part of the MT’s standard operation in idle mode [12], [13] in order to assist the Network Selection, Cell Selection and Reselection functions. These parameters include the Cell Identity and RSS values from the primary and a set of neighboring cells in GSM/UMTS networks. In a WLAN environment the AP MAC Address and RSS values may also be monitored to ensure optimum performance. Moreover, according to radio coverage conditions MTs supporting both GSM and UMTS connectivity, may monitor only the first, the second or both networks. Multi-homed terminals available in the near future will have the ability to be simultaneously attached to several wireless access networks. Since these monitoring procedures are part of the MT’s standard functionality, adding a module to handle the STAMP List management is the only software modification required at the terminal side.

At the LBS application initiation, the STAMP List is uploaded to the Positioning Server; see Fig.1. The STAMP List may also be augmented with additional information available during active mode, such as Timing Advance (TA) for GSM or Round Trip Time (RTT) for UMTS. Based on the type of collected measurements and the QoP requirements the most appropriate positioning technique can be selected, to provide a coarse position estimate for each entry in the STAMP List. Subsequently, statistical processing in the form of filtering, is used as a post processing step, to smooth the positioning error in the sequence of position estimates and increase the accuracy of the current MT’s position. For instance, Bayesian filtering in the form of Kalman or particle filters can be employed.

An important factor that affects the performance of the method is the STAMP List size, i.e. the number of measurement entries stored locally at the MT. This is closely related to convergence issues regarding the subsequent use of statistical processing. Therefore, the STAMP List should be long enough, so that an adequate number of coarse position estimates are provided to the Kalman filter. On the other hand, the size should be kept as low as possible to avoid excessive memory requirements on the MT and reduce the network overhead.
imposed by the messages exchanged. In [14], simulation analysis revealed that acceptable accuracy can be achieved if the STAMP List size is 40. This can be well handled both by the MT, regarding on-site storing requirements and network resources, considering the communication overhead.

The Sampling Period, i.e. the time interval between two consecutive entries in the STAMP List, is another crucial parameter. The proper value depends on the accuracy requirements of the LBS application, as well as the access technology that the measurements are related to. The Sampling Period should be short enough to allow for accurate positioning in the recent past, while a longer value is desirable to accommodate low battery consumption requirements. This trade-off has been further analyzed in [14], using different mobility scenarios in order to investigate the effect of user speed.

STAMP is compatible with any technique that exploits network related measurements. New and proprietary techniques, including terminal assisted E-OTD, OTDOA or fingerprint methods, can be incorporated. STAMP hides the diversity of techniques making the localization procedure completely transparent to the end user. It should be noted that STAMP is applicable even in case of a single RAT and/or positioning technique. In [9], STAMP has been applied in a GSM network to improve the accuracy of the CGH++ technique.

III. PROPOSED ARCHITECTURE

The UTAP system comprises the Positioning Server (PS) and the UTAP clients. Clients, i.e. position requestors, can be MTs or any other external LBS applications. Two distinct scenarios are supported regarding the source of the positioning request that correspond to the Terminal and Network Initiated cases, respectively. The proposed UTAP system architecture supports the basic functionality and satisfies the requirements of STAMP methodology. It is based on a typical client-server model as depicted in Fig.2.

At the PS side, a hierarchical modular design is followed. In order to facilitate system scalability, functional verification and validation, as well as modification of interfaces to be compliant with current protocol versions, the use of separate components, performing distinct and well-defined tasks, was chosen.

Compatibility with existing architectures and standardized communication protocols for transferring available network related measurements is also important, in order to have a platform that fits into existing service oriented architectures. The ability to expand the system to support upcoming wireless access technologies, such as WiMAX, is another key factor. Easy integration of additional positioning algorithms and/or statistical processing techniques was also considered. Taking into account that several positioning algorithms are available, QoP provisions are necessary to optimize system performance in terms of response time and meet the positioning requirements of individual LBS applications. For instance, positioning requests may have different QoP requirements in terms of positioning accuracy, Maximum Location Age and Maximum Delay. The first attribute ensures that only the technique providing the required accuracy will be employed. The second one is used to allow for a previously estimated position to be returned as the current position. The last one is used to exclude those positioning techniques that are computationally inefficient, even if they achieve the desired level of accuracy. Following the high level design principles, the components that facilitate the operation of the PS are presented below.

The Pre-Processing module handles the establishment and termination of the connection with the client through the Client Handler module, as well as the parsing of information carried within the messages exchanged. If the message that carries position related information is valid, its contents are represented in an appropriate internal message format. Subsequently, this is passed to the Controller module that controls the information flow within the PS. When a connection is established, the Controller handles the Network and Terminal Initiated sessions. It holds all session specific data, including unique session id and session specific timer. A Location Cache, in which previous position estimates are stored for later reference, is also maintained and updated. If the QoP requirements, such as the Maximum Location Age set by the application allow it, a cached position will be returned to the client without triggering the Algorithms module to perform actual positioning. The Controller has access to the Privacy & Security Management module to ensure the privacy and security of the involved parties. Every positioning request made by external clients is authenticated and authorized, otherwise rejected, based on the specific user profile and settings, as well as emergency and lawful regulations that may apply in the future.

The Algorithms module is responsible for delivering position estimates according to the QoP requirements and the location related information contained in the request. More specifically, the Algorithm Selector loads and initializes all available positioning algorithms at the PS start-up. When a positioning request is processed, the algorithm components that cannot provide a position estimate are filtered out. This case includes a wrong network type or lack of support for the specific area. Then, the most suitable algorithm is selected to process the actual location query. The decision is affected by the uploaded location related measurements, e.g. a subset of all available techniques are applicable when RSS values from
a GSM network are employed. In order to select a specific technique, QoP issues related to the positioning accuracy and computational time are also taken into consideration. Therefore, some techniques may be filtered out because they exceed the QoP thresholds set in the positioning request. Finally, the estimated position is returned. Regarding GPS enabled devices, the GPS coordinates are the location measurements. GPS is considered an optional positioning technique to be selected, depending on Dilution of Precision (DOP) accuracy parameters. These parameters include the Horizontal, Vertical and Position DOP (HDOP, VDOP, PDOP), which are calculated mathematically from the positions of the usable satellites. Thus, WLAN fingerprinting may be selected, for example in a street canyon, if the GPS precision is low.

The Algorithm Selector has also the ability to switch among different algorithms, while processing the location related information, to support a hybridization scheme. This is valuable when, for example, the MT is moving from indoors (WLAN fingerprinting) to outdoors (cellular techniques or GPS) and vice versa. This scenario is presented in Section V-C with real measurements. The series of estimated positions, derived from a single or multiple techniques after processing the STAMP List, is then forwarded to the Statistical Processing module.

The Algorithm Interface provides the communication interface between the Algorithm Selector and the independent algorithm components. This interface ensures that additional positioning algorithms can be easily integrated in the future. Apart from performing the actual positioning when finally queried, every component should be able to provide a fast initial QoP estimate based on the current measurement. This is necessary to assist the Algorithm Selector in the selection process. For example, a RSS based algorithm using trilateration should return higher positioning error as initial QoP prediction, if the Base Stations (BS) included in the current measurement are aligned, resulting in poor geometric conditions.

The Statistical Processing module implements a post processing technique that filters the series of coarse position estimates according to the positioning error provided by each positioning method. This final step alleviates high errors, while smoothing the sequence of coarse position estimates. Thus, higher accuracy is achieved regarding the current terminal location. Accuracy enhancement is illustrated in Sections V-A and V-B with real measurements for an indoor and an outdoor moving scenario, respectively.

At the terminal side, the UTAP agent is a software component that is responsible for the STAMP related functionality. It maintains and manages the STAMP List, processes all incoming positioning requests, generates responses and handles all low level API communication to access and collect radio layer measurements. Measurements from all available RATs are stored in the STAMP List, in an appropriate format.

It should be noted that the UTAP system architecture is consistent with OMA SUPL specifications [11]. The UTAP agent upgrades the device to a SUPL enabled terminal. The Userplane Location Protocol (ULP) [15] and Mobile Location Protocol (MLP) [16] are the communication protocols employed for transferring the messages created by MTs and LBS applications to the PS, respectively. Time-stamping of measurements is included in the specifications, while the support for multiple historical measurements was submitted to OMA as a Change Request for SUPL and has been accepted [17]. Therefore, the STAMP concept and the proposed architecture are compliant with User Plane positioning roadmaps.

IV. Prototype Implementation

The development of the prototype is based on the UTAP system architecture, presented in Section III. The UTAP agent has been implemented in C++ and installed on Symbian OS (Series 60) enabled MTs. The STAMP List management functionality currently supports the collection of RSS values from GSM, UMTS and WLAN networks that are stored locally at the MT. The Sampling Period has a fixed value of one second, however in future implementations it can be configured dynamically following the OMA Device Management (DM) procedures [18]. A client simulation program has also been developed to test the communication flow between the SUPL enabled terminal and the PS. Through this command line application a RSS measurement survey file collected for a route is properly encoded and sent to the PS as consecutive positioning requests. Actual location coordinates are also included for calculating the positioning error. In order to add some flexibility, the STAMP List size is a user defined parameter.

All individual components comprising the PS, as depicted in Fig.2, have been realized. Regarding the positioning algorithm components, the Database Correlation Method (DCM) [19], CGI++ [20] and Common Pilot Channel (CPICH) have been implemented and integrated into the PS. They all rely on RSS values collected by available wireless access networks, but follow two distinct approaches; fingerprint matching and trilateration. The CPICH technique is similar to CGI++ and employs RSS measurements from UMTS networks. The Statistical Processing module implemented in this prototype, is based on the Kalman filter iterative algorithm described in [6]. The coarse position estimates \((x, y)\), provided by the positioning algorithms, are treated as measurements disturbed by zero mean Gaussian noise with covariance matrix \(R = \sigma_R I_2\). The parameter \(\sigma_R\) reflects the mean positioning error introduced in the estimates when a specific technique is used. In that sense, \(\sigma_R\) is easily obtained by performing real measurements in the area of interest and calculating the average error of each positioning algorithm offline.

A management GUI has been designed to control the features of the PS, such as enabling or disabling algorithm components and interactively setting some algorithm specific attributes. The option to depict the resulting position estimates on different layers of digital maps is also provided; see the localization scenarios in Section V. This feature is not part of the system’s standard functionality, as the primary goal of UTAP system is the provision of high quality location estimates to external LBS applications. This option is implemented for visualization, verification and testing purposes.
A subset of the SUPL family protocols was exploited in this implementation, while some modifications were made to support additional functionalities. Since WLAN positioning is not yet fully supported, this feature was added. GPS QoP indicators such us HDOP, VDOP and PDOP, were also added to indicate whether the GPS position meets the required QoP. The option to include a location tag to the uploaded measurements is also provided, as an additional feature for GPS enabled terminals. This can be helpful, for example when a user is asked to contribute RSS fingerprints from a specific area to the operator’s database for a reward. ULP messages are encoded in XML and exchanged over TCP/IP connections. In general, the non-roaming, proxy modes for both Network and Terminal Initiated cases are implemented. Moreover, the SUPL Location Center (SLC) and the SUPL Positioning Center (SPC) [11] are merged into a single entity.

V. LOCALIZATION SCENARIOS

In order to verify the functionality of individual modules comprising the UTAP system, some indicative scenarios were considered. At the same time preliminary accuracy results for the positioning algorithms and statistical processing components were obtained. RSS samples have been collected in different routes, including indoor and outdoor scenarios. The STAMP List size was fixed to 30 samples. The estimated positions, as presented in Fig.3, are plotted on digital maps. Accuracy results are reported including the mean positioning error $m_e$ and the standard deviation $\sigma_e$ for the whole route. It should be mentioned that a thorough evaluation of the prototype and analysis of the results is out of the scope of this paper. As a future step, we will assess the UTAP system through extensive trials, examine the system performance and analyze the accuracy results obtained under different conditions.

A. Indoor Positioning

In this scenario, a user walking inside a building is considered. In order to build the reference fingerprint database, 378 WLAN fingerprints have been collected indoors and outdoors close to the building walls in a $13K \text{ m}^2$ area covered by 27 WLAN APs. Samples containing RSS measurements from all hearable APs are then collected by a WLAN attached mobile device and stored in the STAMP List. The DCM algorithm is enabled through the management GUI and successive position estimates are depicted on a floor plan map of the VTT premises located in Espoo, Finland as shown in Fig.3a. The true locations and DCM position estimates are shown in green and blue dots, respectively. When only DCM is used results show that $m_e = 4.7m$ ($\sigma_e = 3.16m$). Red dots denote the position estimates derived after the statistical processing of the DCM estimates. In this case, the positioning error is slightly improved ($m_e = 3.2m$, $\sigma_e = 1.90m$).

B. Outdoor Positioning

An outdoor scenario is depicted in Fig.3b. A MT within a vehicle is collecting measurements from the commercial GSM network of Vodafone-GR, in an urban area of Athens, Greece covered by 20 BSs. For this route, each sample contains RSS measurements from 6 BSs. The CGI++ component provides coarse position estimates (blue dots), while true positions are shown in green dots. The component has been initialized with the necessary network information, including the Cell IDs, as well as the coordinates, transmit power, height and frequency of each BS which are required by the Hata propagation model. Results show that $m_e = 212m$ and $\sigma_e = 94m$, when only CGI++ is used. The estimated positions obtained after Kalman filtering, are shown in red dots. Statistical processing leads
to considerable improvement regarding the accuracy achieved ($m_e = 167\text{m}, \sigma_e = 78\text{m}$). In this scenario, the average positioning error provided by the plain Cell ID technique is 280m. The standard Hata propagation model was employed [21], however utilizing a calibrated propagation model is expected to enhance the performance of CGI++. Moreover, calibrating the filter parameters will further increase the effect of the statistical processing.

C. Hybridization & Ubiquitous Positioning

In Fig.3c the hybridization concept is illustrated for a user passing through a building. The same fingerprint database, as in Section V-A, is used. GPS and DCM are the positioning techniques under consideration and the most suitable for each sample is selected, based on the estimated QoP. The true locations, DCM estimates and GPS positions are shown in green, blue and yellow dots, respectively. In the AB segment, DCM estimates are only available when the user approaches the main building walls. GPS location estimates are heavily affected by the building and some times indicate a route through the walls. Inside the building (BC segment) only DCM estimates are available. Statistical processing is applied on the estimates derived after hybridization to further increase accuracy (red dots). When the user moves outdoors (CD segment), DCM accuracy degrades and it takes time for the GPS estimates to get accurate enough. This also affects the performance of the Kalman filter and the estimated velocity, shown in red arrows, increases rapidly. If WLAN fingerprints were available for that part of the route the achieved accuracy would be higher. Even so, in the hybrid GPS/DCM scheme $m_e = 8.36\text{m} (\sigma_e = 7.1\text{m})$, while $m_e = 11.06\text{m} (\sigma_e = 8.69\text{m})$ if only GPS is used.

VI. Conclusions

In this paper, a terminal assisted localization methodology, applicable in existing wireless networks and positioning techniques, has been presented. The deployment of STAMP implies only additional software modifications at the terminal and network side. The proposed architecture follows the SUPL architecture specifications and has been designed under consideration of existing standards and communication protocols, such as ULP and MLP. Furthermore, it is in line with market trends showing that terminal manufacturers will deliver SUPL enabled devices within the next years. Moreover, the modular architecture provides an open platform that facilitates fast integration of new and custom positioning techniques.

Future work will focus on fine-tuning and testing the end to end communication in the UTAP system in order to examine and analyze the system performance and stability. We plan to test the system in a commercial UMTS network and implement additional positioning techniques, such as probabilistic DCM and post processing methods, such as Kalman filter variants and map-matching. We will also evaluate the UTAP system in real life conditions by investigating the effect of user speed and density of infrastructure. Further steps will concentrate on enhancing the QoP provisions and providing uninterrupted positioning, especially in hybrid schemes where different measurements and multiple positioning techniques are available.

ACKNOWLEDGMENT

This paper introduces concepts and technologies deployed within the framework of the MOTIVE project [10], which is partially funded by the European Commission under the IST-FP6 program. This work is partly supported by the Cyprus Research Promotion Foundation under contracts ΠΛΗΡΟ/0603/06 and ΕΝΙΣΧ/0506/59. The authors would like to thank Reach-U (http://www.reach-u.com) for the provision of digital maps in the context of MOTIVE.

REFERENCES

[12] “TS 43.022 v.7.2.0: Functions related to Mobile Station (MS) in idle mode and group receive mode,” 3GPP.
Abstract—Provision of accurate and reliable location estimates is the key issue for the proliferation of indoor location oriented services and applications. Our positioning method is based on Radial Basis Function (RBF) networks and we exploit Received Signal Strength (RSS) measurements from several WLAN Access Points (AP). We incorporate the RSS covariance matrix into the estimation method and couple that with a methodology that indicates which APs can be ignored during positioning without sacrificing accuracy. We evaluate the RBF method in a real-life setup, and experimental results suggest that the proposed approach performs well.

I. INTRODUCTION

Indoor positioning is challenging due to the complex propagation environment and unpredictable time varying conditions, such as the presence of people or equipment. Improving the localization accuracy and the robustness to these environmental factors is expected to increase the interest in location aware applications, such as indoor guidance and asset tracking.

Different positioning technologies have been discussed in the literature including infrared, Bluetooth, RFID, UWB, ultrasound and WLAN. Several positioning methods rely on WLANs, mainly due to the wide availability of relevant infrastructure in indoor environments. These methods exploit Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA) and Received Signal Strength (RSS) measurements from Access Points (AP) to infer the unknown user location. An overview of technologies to determine location and a survey of commercial positioning systems is provided in [1], [2].

In the context of WLAN positioning, RSS measurements are usually preferred, because they can be easily collected without the need for specialized and expensive equipment. Indoor radio propagation models have been used to transform RSS values into distances from at least three relevant APs in order to determine the unknown user location. However, this approach has some limitations, mainly due to the multipath effect that renders the use of standard log-distance propagation models inadequate. Another problem is that the exact locations of the APs are required, and such information is either not available or hard to obtain. Fingerprinting methods address both issues by utilizing RSS fingerprints collected a priori at some predefined reference points in the area of interest. Location can then be estimated using the currently measured fingerprint to find the best match between the current and collected fingerprints.

In our approach, we employ a type of Artificial Neural Networks (ANN) called Radial Basis Function (RBF) networks and use the collected reference data to build a mapping between the RSS fingerprints and location coordinates. We investigate the performance of the RBF positioning method, in case the RSS covariance matrix is used. Moreover, variable selection is addressed, and we discuss a suitable cross-validation methodology to decide which input variables, i.e. APs, can be eliminated for dimensionality reduction.

The rest of the paper is structured as follows. Previous work related to indoor positioning using RSS fingerprints and AP selection methods is discussed in Section II. The proposed method based on RBF networks is detailed in Section III. In Section IV, we present the WLAN experimental setup used for our performance evaluation, followed by the results regarding the positioning accuracy. Finally, Section V provides the conclusions and discusses some ideas for future work.

II. RELATED WORK

A. WLAN RSS-based positioning

Several approaches utilize a number of RSS fingerprints, collected a priori at some reference points. In the deterministic case, location is estimated as the average of $K$ Nearest Neighbors (KNN) [3], i.e. reference points with the shortest distance between the observed and mean RSS fingerprint at each point. Probabilistic methods determine location by using estimates of probability density functions (PDF). Kernel-based techniques or the histogram density estimate can be used as nonparametric approximations of the required PDFs [4]. Other methods rely on ANNs, including the Multi Layer Perceptron (MLP) [5], [6] or Support Vector Machines (SVM).

In the proposed RBF method, location is expressed as the weighted sum of Gaussian radial, i.e. distance-based, functions and we use the RSS covariance matrix in distance calculation. Weights can be determined easily from the training data using linear algebra. In the positioning phase, the RBF network
outputs a location estimate given the currently observed fingerprint.

B. AP Selection Methods

A mobile device may detect a large number of APs due to the ubiquitous coverage of several WLANs that have been deployed independently. However, some APs may provide correlated RSS samples that introduce bias in the location estimates. Moreover, using all available APs has an impact on the computational complexity of the positioning algorithm. On the other hand, ignoring specific APs can lead to accuracy degradation, if the RSS values from these APs affect the ability of the positioning algorithm to discriminate between neighboring reference points. Therefore, the objective of AP selection schemes is to identify a subset of APs that will be used during positioning without compromising performance.

In the SkyLoc fingerprint-based floor positioning system [7], forward selection and backward elimination have been used to select the appropriate GSM Base Stations. In the context of WLAN RSS positioning, Youssef et al. choose the APs with the highest RSS value in the observed fingerprint. The intuition is that these APs provide the highest probability of coverage over time. However, it has been observed experimentally that the variance of RSS measurements at a given location increases with its mean power [8], [9] and thus selecting the strongest APs may not always provide the best accuracy. In [10] the APs that are more capable of distinguishing the reference points, are included in the subset, and the selection process is based on the Information Gain criterion. Authors in [11] rely on divergence measures, such as the Bhattacharyya distance or the Information potential, to deal with correlations between signals from APs and obtain a small subset. In [12], selection is based on the discrimination score of each AP calculated with the Fisher criterion.

In this work we use cross validation to select the appropriate APs. These kinds of techniques, including the leave-one-out and generalized cross validation, are able to predict the performance of a trained model when new unobserved inputs are available. In this way, available APs can be ordered based on their ability to describe the input-output relation, and then we can select only the most important to consider during positioning.

III. POSITIONING ALGORITHM

A set of reference points \( \ell_i = (x_i, y_i), i = 1, \ldots, L \) is used to measure RSS fingerprints \( s = [s_1, \ldots, s_n]^T \) from \( n \) APs, where \( s_j \) denotes the RSS value related to the \( j \)-th AP. A series of fingerprints \( s(\ell_i, m), i = 1, \ldots, L \) and \( m = 1, \ldots, M \), are collected at each reference point and associated with the physical coordinates \((x_i, y_i)\). Thus, our training set contains \( N = L \cdot M \) fingerprints denoted as \( s_i, i = 1, \ldots, N \).

In our method, the RBF network has \( n \) inputs, corresponding to RSS values from all \( n \) APs and two outputs representing the coordinates. Given a fingerprint \( s \) measured at location \( \ell = (x, y) \), the output of the RBF network may be expressed as the weighted sum of normalized Gaussian basis functions

\[
\ell(s) = \sum_{i=1}^{L} w_i u(s, c_i) \\
u(s, c_i) = \frac{\varphi(||s - c_i||)}{\sum_{j=1}^{L} \varphi(||s - c_j||)}
\]

where \( \varphi(||s - c||) = \exp \left(-\frac{1}{2} ||s - c||^2 \right) \) and \( w_i = [w_i^x \ w_i^y] \) are 2-dimensional weights. We set each RBF center \( c_i \) equal to the mean value fingerprint at each reference point \( \bar{s}(\ell_i) \) that is defined as

\[
\bar{s}(\ell_i) = \frac{1}{M} \sum_{m=1}^{M} s(\ell_i, m), \ i = 1, \ldots, L.
\]

We may determine the unknown weights using the reference data by solving the overdetermined system of linear equations

\[
\ell_i = \sum_{j=1}^{L} w_j u(s(\ell_i, m), c_j), \ i = 1, \ldots, L, \ m = 1, \ldots, M.
\]

These equations can be written in matrix form as two linear systems \( Uw^t = d^t \), \( t \in \{x, y\} \) that represent a separate RBF network for each location coordinate, where

\[
U = \begin{bmatrix}
u(s(\ell_1, 1), c_1) & u(s(\ell_1, 2), c_1) & \cdots & u(s(\ell_1, L), c_1) \\
u(s(\ell_2, 1), c_1) & u(s(\ell_2, 2), c_1) & \cdots & u(s(\ell_2, L), c_1) \\
\vdots & \vdots & \ddots & \vdots \\
u(s(\ell_M, 1), c_1) & u(s(\ell_M, 2), c_1) & \cdots & u(s(\ell_M, L), c_1)
\end{bmatrix}
\]

\[
w^t = [w^t_1, w^t_2, \ldots, w^t_L]^T
\]

\[
d^t = [d^t_1, d^t_2, \ldots, d^t_N]^T
\]

Each row in the \( N \times L \) matrix \( U \) contains the responses of the basis functions to a particular fingerprint, while \( w^t \) are the \( L \times 1 \) unknown weights and \( d^t \) are the \( N \times 1 \) outputs for each coordinate. The respective weights are calculated in a least squares sense by \( w^t = U^+d^t \), where \( U^+ \) is the pseudoinverse defined as \( U^+ = (U^TU)^{-1}U^T \).

Subsequently, the weights are used to derive the estimated location, given the currently observed fingerprint \( s' = [s'_1, \ldots, s'_n]^T \)

\[
\hat{\ell}(s') = \sum_{i=1}^{L} w_i u(s', c_i).
\]

A. Distance Calculation

The euclidean norm can be used to calculate the distance between the input fingerprint and RBF centers in the RBF network. However, individual fingerprint elements may differ regarding the distribution of RSS. Thus, we consider a weighted norm as distance measure and use the following set of basis functions that represent multivariate Gaussian distributions with mean \( c_i \) and single common covariance matrix \( \Sigma \).

\[
\varphi(||s - c||) = \exp \left(-\frac{1}{2} (s - c)^T \Sigma^{-1} (s - c) \right), \ i = 1, \ldots, L.
\]
The covariance matrix determines the receptive field of the basis functions [13] that affects the performance of the RBF positioning method, because it has an influence on the subset of the RSS input space for which each basis function has a fairly large output. In our previous work [14], we considered $\Sigma = \sigma^2 I$, where $\sigma^2$ is a common variance for all $n$ APs and proposed a heuristic to select an appropriate value for $\sigma^2$ according to the maximum distance among RBF centers.

Another option is to use a diagonal covariance matrix

$$\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$$

(10)

where $\sigma_k^2$ is the sample variance of the $k$-th AP estimated from the collected reference data. In this way, we are still under the assumption that RSS values from different APs are independent, and we exploit the RSS variance in our AP selection methodology that is described in the following. Note that a non-diagonal covariance matrix could be used to consider correlations between APs. However, when a WLAN is deployed inside a building, non-overlapping transmission channels are usually preferred in order to minimize interference between neighboring APs. Thus, the independency assumption is valid and has been verified experimentally in [8].

B. Cross validation AP Selection

In typical WLAN setups, there are several APs installed throughout the building that can be detected and used for positioning. Thus, the input dimensionality of the RBF networks in our method is increased. This is not a significant limitation regarding the calculation of the RBF weights, because the weights are determined once off-line. However, computational complexity during positioning is increased, which can affect real-time positioning in case low processing power mobile devices are considered.

In this work, we address the need for the joint design of AP selection and distance calculation. We use a single common norm weighting matrix $\Sigma^{-1}$ by setting

$$\Sigma = \text{diag}(\alpha_1\sigma_1^2, \alpha_2\sigma_2^2, \ldots, \alpha_n\sigma_n^2).$$

(11)

Vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ can be considered as a vector of scaling parameters that indicate the importance of each input variable, i.e. AP. Small value of a specific $\alpha_k$ suggests that the respective input is significant, and even small changes are reflected to the output of the RBF network. On the other hand, if $\alpha_k$ is large, then the output does not vary much with respect to the RSS level of that AP. Thus, we may use parameters $\alpha$ to obtain an ordering of the available APs based on their significance.

We use the Generalized Cross Validation (GCV) method to estimate $\alpha$; see [13] for details. In this context, vector $\alpha$ is selected to minimize the objective function given by

$$V(\alpha) = \frac{1}{N} \| (I - A(\alpha))d \|^2$$

$$A(\alpha) = U(U^TU)^{-1}U^T$$

(12)

Matrix $A$ depends implicitly on $\alpha$, because the covariance matrix given in (11) is used for distance calculation. The output of the location coordinate $x$ or $y$ is denoted by $d$ and $\text{tr}(\cdot)$ denotes the trace of a matrix. Notice that (12) relies only on the training data. Following this procedure, we obtain two sets of parameters $\alpha^t$, $t \in \{x, y\}$ that scale the inputs in each RBF network appropriately. Subsequently, we can choose the subset of APs that are important for the performance of our positioning algorithm and remove the remaining for dimensionality reduction.

IV. EXPERIMENTAL EVALUATION

A. Measurement Setup

The performance evaluation was done using data collected in a typical office environment on the second floor of a three storey building at VTT Technical Research Centre of Finland. The floor consists of eight wings containing offices and meeting rooms connected with corridors. The floorplan of the experimentation area and the locations of the reference points are depicted in Fig. 1. We used a Fujitsu-Siemens Pocket Loox smart phone with Windows Mobile operating system to collect RSS measurements at 107 reference points. These points are separated by 2-3 meters and form a grid that covers all public spaces and meeting rooms.

A total of 3210 reference fingerprints, corresponding to 30 fingerprints per reference point, were collected at the rate of 1 sample/sec. We use all 31 available WLAN APs that are installed throughout the building. Due to the open plan interior design, these APs can be partially detected on the second floor, and the average number is 9.7 APs per reference point. RSS values range from -101dBm to -34dBm and we used a small constant to handle the missing RSS values in the fingerprints. For testing purposes, fingerprints were also collected by walking at a constant speed over a path that consists of 192 locations. One fingerprint is recorded at each location, and the same path is sampled 3 times.

B. Parameter tuning

We reserved the collected fingerprints from one route as validation data and used samples from the other two routes as
The conditional probability boring APs are independent, we get.

\[ p = \min \{ \alpha^t \} \]

By application of Bayes rule, the problem reduces to calculating

\[ \log p(s|\ell, M) = \log p(s|\ell) - \log \sum_{s^*} p(s^*|\ell, M) \]

The KNN variant determines location as the average of \( K \) reference data, and parameter \( w \) determines the kernel width.

\[ \hat{\ell}(s') = \arg \min_{\ell} \| s' - \pi(\ell) \|^2. \]  

(14)

The mean error pertaining to the validation set for KNN method using variable number of neighbors \( K \) is plotted in Fig. 2. The mean error regarding the test set is also included to show that tuning \( K \) based on the validation data can provide acceptable and usually near-optimal performance when the test data are considered. The plot indicates that \( K = 3 \) provides the lowest mean error for the given setup, and this value is used thereafter.

In the probabilistic method \( 4 \), denoted as KERNEL, the unknown location is estimated as the expected value of the location variable \( \ell \)

\[ \hat{\ell}(s') = \mathbb{E}[\ell|s'] = \sum_{i=1}^{L} \ell_ip(\ell_i|s'). \]  

(15)

By application of Bayes rule, the problem reduces to calculating \( p(s'|\ell_i) \) and assuming that RSS measurements from neighboring APs are independent, we get \( p(s'|\ell_i) = \prod_{j=1}^{n_i} p(s'_j|\ell_i) \).

The conditional probability \( p(s'_j|\ell_i) \) is estimated as the average of \( M \) equally weighted Gaussian kernel functions. Each kernel is centered on \( s_j(\ell_i, M) \), \( m = 1, \cdots, M \) according to the reference data, and parameter \( w \) determines the kernel width.

The mean positioning error as a function of the kernel width is depicted in Fig. 3, and \( w = 7 \) provides the lowest error with respect to the validation data set.

For the RBF method, a Genetic Algorithm was used to minimize (12) and obtain several solutions for the scaling parameters \( \alpha^t \). We used the validation data to select the best \( \alpha^t \), \( t \in \{ x, y \} \) and for this solution \( \alpha_x \in [0.18 \, 1.36] \), while \( \alpha_y \in [0.13 \, 2.07] \). Sorting the scaling parameters indicates which APs are not so important to describe the RSS-position relation. Those APs that have high scaling values can be excluded during positioning without significantly affecting the performance of the RBF method. For the RBF network that corresponds to each location coordinate \( x \) or \( y \), we take the ordering of APs with respect to increasing scaling values. The regulating parameter \( 0 \leq b \leq 1 \) denotes the percentage of APs that will be used for position estimation.

The positioning error as a function of \( b \) is plotted in Fig. 4. The mean error regarding the validation set is 3.1m in case the 16 most important APs (\( b = 0.5 \)) are used in each RBF network. Note that these two subsets do not necessarily contain the same APs, because the ordering is based on the individual scaling parameters \( \alpha^t \). Using less than 12 APs (\( b < 0.4 \)) leads to further accuracy degradation, because the selected APs are not enough to provide good coverage throughout the experimentation area. Results on the validation set reveal that if we set \( b = 0.7 \), then the RBF method provides the same level of accuracy compared to KNN and KERNEL methods.

C. Results

We use the test data, collected by sampling the same path twice, to evaluate the performance of the proposed
RBF method. First, we employ RSS measurements from all available APs to examine the level of accuracy provided by the RBF method, when the diagonal covariance matrix $\Sigma = \text{diag}(\sigma_k^2), k = 1, \ldots, 31$ is used in distance calculation. Then, we investigate the effect on the performance in case we use a subset of APs based on the scaling factors calculated in Section IV-B.

The actual locations (shown with dark grey line) and the RBF estimates for each test set are depicted on the floorplan map in Fig. 5 and Fig. 6, respectively. The user follows a path that covers different wings in the floor, and the estimated locations reflect the traveled trajectory.

The RBF method is compared to KERNEL and KNN methods in terms of the positioning error, defined as the Euclidean distance between the actual and estimated location. The Cumulative Distribution Function (CDF) of the positioning error for the first test set is plotted in Fig. 7. The error regarding the 67th percentile is 2.6m, 2.9m and 3.0m for RBF, KERNEL and KNN methods, respectively. When the 95th percentile is considered, the error is 5.1m, 5.7m and 5.5m. Similar performance is achieved for the second test set, as shown in Fig. 8. In this case, the positioning error is increased, but the RBF estimates are still more accurate compared to other methods.

Results pertaining to both test sets are summarized in Table I and indicate that the RBF method improves the positioning accuracy with respect to the mean and median error. Moreover, the RBF method provides the best performance when the maximum positioning error is considered.

<table>
<thead>
<tr>
<th>POSITIONING ERROR [M]</th>
<th>Mean</th>
<th>Median</th>
<th>67% CDF</th>
<th>95% CDF</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>2.5</td>
<td>2.3</td>
<td>2.9</td>
<td>5.4</td>
<td>7.4</td>
</tr>
<tr>
<td>KERNEL</td>
<td>2.6</td>
<td>2.4</td>
<td>3.1</td>
<td>5.3</td>
<td>8.4</td>
</tr>
<tr>
<td>KNN</td>
<td>2.7</td>
<td>2.5</td>
<td>3.3</td>
<td>5.7</td>
<td>8.6</td>
</tr>
</tbody>
</table>

The RBF approach outperforms the other methods in our evaluation when RSS values from all available APs are employed. However, our second objective is to achieve a level of accuracy that is comparable to other methods, while reducing the number of APs required during positioning. The CDF of the positioning error pertaining to both test sets is plotted in Fig. 9, assuming that $b = 0.7$ for the RBF method. In this case, the RBF networks utilize RSS measurements from 22 APs contrary to KNN and KERNEL methods that exploit 31 APs.
devices, such as PDAs, are considered. To extend the battery life, especially when low power mobile devices are used, a small subset of the available APs can indeed be beneficial to minimize the computational complexity during positioning and thus estimation time can be reduced by using the scaling parameters to identify which APs should be ignored during positioning. Within WLAN, the variance of RSS is very high and at the same time different subsets of APs may be more appropriate for positioning, depending on the region that the user resides. As a next step, we plan to consider an adaptive approach in order to further improve the performance of the RBF method.

The mean error is 2.7m, 2.6m and 2.7m for RBF, KERNEL and KNN methods, respectively. Accuracy results reveal that we can use a fraction of the APs that are spread in the area of interest, without compromising the performance of the RBF method.

The reduction in the number of APs that are used in the RBF networks, results in significant savings with respect to the computational time during positioning. One position estimation using the RBF method takes approximately 0.33msec (mean), for a Matlab implementation on an Intel Pentium 4 processor 3.6GHz with 1GB RAM. We assume that the RBF network weights are calculated offline, all 31 APs are used for positioning, and the execution times are averaged over 100 runs using the data from both test sets. If RSS measurements from 22 APs are used, then the mean calculation time per location estimate is decreased by 24% (0.25msec). We also highlight that in large public places the number of detected APs can be much higher than 31. Therefore, using a small subset of the available APs can indeed be beneficial to minimize the computational complexity during positioning and extend the battery life, especially when low power mobile devices, such as PDAs, are considered.

V. CONCLUSION

A RBF-based positioning method that exploits WLAN RSS fingerprints is presented. The proposed method uses the RSS covariance matrix in distance calculation and is very efficient, because the weights and scaling parameters of the RBF network are calculated once during training and are used thereafter to estimate the unknown user location. Moreover, computational complexity and thus estimation time can be reduced by using the scaling parameters to identify which APs should be ignored during positioning.

Within WLAN, the variance of RSS is very high and at the same time different subsets of APs may be more appropriate for positioning, depending on the region that the user resides. As a next step, we plan to consider an adaptive approach in order to further improve the performance of the RBF method. In this approach, reference points that are spatially related to the user location are selected dynamically, and the covariance matrix is re-estimated in real-time. Then, an AP selection methodology can be applied. In this fashion, we may adjust the size of RBF networks and re-calculate the weights on-line by exploiting only relevant reference data, instead of calculating them off-line based on all available reference fingerprints.

ACKNOWLEDGMENT

This work is partly supported by the Cyprus Research Promotion Foundation under contract ENIΣΧ/0506/59.

REFERENCES

Indoor Positioning in WLAN using Radial Basis Function Networks with Received Signal Strength Fingerprints

Christos Laoudias and Christos G. Panayiotou
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering, University of Cyprus, Nicosia
laoudias@ucy.ac.cy, christosp@ucy.ac.cy

Abstract

Positioning techniques enable the provision of location information regarding people, mobile devices and equipment. Estimating location accurately is a challenge especially inside buildings, where satellite-based positioning is not applicable due to the severe attenuation or blockage of satellite signals. Positioning accuracy is the key issue to effectively support advanced indoor location aware services. Indicative applications include in-building guidance, asset tracking in hospitals or warehouses and autonomous robot navigation.

Different positioning technologies have been discussed in the literature including infrared, Bluetooth, RFID, UWB, ultrasound and WLAN. Several positioning methods rely on WLANs, mainly due to the wide availability of relevant infrastructure in indoor environments. These methods exploit Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA) and Received Signal Strength (RSS) measurements from Access Points (AP) to infer the unknown user location.

In the context of WLAN positioning, RSS measurements are usually preferred, because they can be easily collected without the need for specialized and expensive equipment. Indoor radio propagation models have been used to transform RSS values into distances from at least three relevant APs in order to determine user location through multilateration. However, this approach has some limitations, mainly due to the multipath effect that renders the use of standard log-distance propagation models inadequate. Another problem is that the exact locations of the APs are required, and such information may not be available or is hard to obtain.

Fingerprinting methods address both issues by utilizing RSS fingerprints collected a priori at some predefined reference points in the area of interest. Location can then be estimated using the currently measured fingerprint to find the best match among the reference fingerprints. Matching is based on a distance measure between the current and reference fingerprints or on probability distributions.

In our approach, we employ Radial Basis Function (RBF) networks and use the collected reference data to build a mapping between the RSS fingerprints and location coordinates. We present an efficient RBF-based positioning method and provide well defined ways for setting the RBF parameters. We investigate the performance of the RBF-based method as a function of the number of available APs, reference points or fingerprints and compare it to some well known approaches.

Corresponding author:
Christos Laoudias
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering, University of Cyprus, Nicosia
laoudias@ucy.ac.cy