Bandwidth and Storage Reduction of Radio Maps for Offline WLAN Positioning

Laura WIROLA[1], Lauri WIROLA[2], Robert PICHÉ[1]
[1]Tampere University of Technology, Tampere, Finland, [2]HERE, Finland

***Abstract*—Most of the existing mobile device positioning methods require data connectivity, i.e. they work in the mobile-assisted, or online mode. However, this consumes energy, induces transmission costs and results in unnecessarily long time-to-first-fix. These issues can be alleviated using mobile-based, or offline, mode. In this mode the device carries a subset of the global radio map in memory for fast positioning without data connection. The challenge of this approach is the large size of the offline radio map that needs to be downloaded, stored and updated periodically in the mobile device. This paper presents a method to find the significant APs in the global radio map and proposes using only those in offline positioning in order to compress the size of the required offline radio map. We also propose a method to further compress the size of the offline radio map by hashing the globally unique AP BSSIDs into locally unique shortened BSSIDs. We test the proposed methods with real-world data.**

I. Introduction

Today WLAN-based positioning is by far the leading non-GNSS method for indoor and urban positioning. The obvious advantage of the WLAN-based methods is the globally existing infrastructure both in terms of access points and device population. Public buildings (shopping malls, offices, schools) as well as homes are filled with WLAN Access Points (APs). In addition, practically all modern mobile devices include a WLAN radio.

WLAN-based methods require up-to-date reference data for adequate positioning availability and accuracy. In order to realize this, current commercial WLAN-based positioning services are based on crowd-sourced AP databases containing AP coordinates and other related attributes. These databases are called Radio Maps (RMs). The major challenge with this approach is the high dynamics of the WLAN AP environment. The WLAN landscape is not centrally managed, as opposed to cellular networks, because anyone can have their own WLAN AP and may move it from one location to another without notice. Therefore, there is a major burden in collecting data to keep up with changes and to update the database on a continuous basis.

WLAN-based location systems may be divided into two categories based on where the position calculation is performed. In *server-based*, or mobile-assisted, positioning the mobile device sends measurements to a remote server for position estimation and the server provides the position estimate to the device. This is also called online positioning and requires the device to have connectivity whenever the device wants to position itself. In a *device-based*, or mobile-based, positioning the mobile device carries a local copy of the global RM and the positioning calculation takes place in the end user device. Usually the local copy is a subset of the global RM in the positioning server. This subset may contain positioning assistance data e.g. for a single city or just a small area in which the user typically moves. This mode can also be referred to offline positioning, because connectivity between the device and the server is not needed for every position request.

A. Discussion on online and offline positioning

Online and offline positioning both have their own advantages and disadvantages in terms of accuracy, availability, time-to-first-fix, privacy, security, authentication, data consumption and server resources. This section discusses these aspects in the context of online and offline positioning.

1) Time-to-first-fix: This is one of the major items affecting the location services user experience [1]. In online positioning the major part of the time-to-first-fix results from setting up the data connection and round-trip time required for sending the measurements to and receiving the response from the server. Especially, WLAN-based navigation requires high update rate with zero delay and lag, which are practically impossible requirements to fulfill with online positioning, because network latencies may be high and unpredictable. In contrast, in offline positioning the terminal can keep itself location-aware all the time reducing the time-to-first-fix to negligible.

2) Privacy: Privacy is an ever more important issue in operating positioning services. There are three aspects to privacy. The first aspect is that the device can *authenticate* the remote party with whom location-related information is being exchanged. The second aspect is that the communications between the client and the server is *secure*. The third aspect is the *privacy* of the data, i.e. that the client can trust the remote party to only use clients location information for the agreed purposes.

The first two aspects can be effectively solved today by standard SSL/TLS techniques (Secure Sockets Layer / Transport Layer Security) [2]. TLS allows asymmetric cryptography for authentication and, after the key exchange, symmetric encryption of the data transfer. In practice this means that the client is aware to whom it is talking and that the sensitive information transferred between the client and the server cannot be eavesdropped. If further security is required, the architecture may implement client-side certificates. However, this solution induces an extra burden of key management. The third item, privacy, is the most problematic because it cannot be resolved purely with technological means — it is based on

trust, which can be broken either deliberately (misuse of data by the service provider) or non-intentionally in case of, e.g., server hacking.

It is important to acknowledge that the aforementioned privacy issues are inherently linked to online positioning services, in which the location information is being transferred between clients and servers. However, offline positioning is largely immune to these problems. When downloading the offline RM it suffices for the client just to send an approximate location to the server in order to get the RM information for the surroundings. In addition, communication with the server happens quite rarely, maybe only on a monthly basis. Moreover, the data provided by the server to the client is the same for all the clients, i.e. all the clients in the same area receive the same data. Thus eavesdropping the data channel does not provide any valuable information. Finally, as the location information that the client needs to provide to the server is approximate and is exchanged quite rarely, the server will not have a capability to recreate the location track of the client.

3) Accuracy and availability: Generally speaking accuracy and availability of offline and online positioning can be similar in the context of static position estimation. However, offline positioning can over time provide better accuracy and availability. Firstly, as the device carries its own RM in offline case, the device is able to filter the time series of measurements or position estimates continuously and the resulting accuracy of position estimation can thus be better than with single-shot online positioning requests. There are also circumstances in which the data connection to the server is not available (network availability, congestion, quality) or not allowed by the user (especially in roaming conditions). In such conditions only offline positioning is available. Therefore, it can also be asserted that, broadly speaking, availability of offline positioning can be higher than that of online positioning.

4) Data transmission considerations: Online positioning requires transferring data each time the device needs to be positioned. Although the amount of data of a single positioning request is small, the cumulative data consumption may be very high. With offline positioning the data consumption is quite predictable — the RM needs to be downloaded once and refreshed periodically.

Having said that, the large size of the RM that the device has to carry in order to provide offline positioning is a special challenge of this approach. To illustrate, suppose that an RM contains only the locations of APs. In urban areas a realistic assumption is that there is on average one AP every 10 m^2 especially because APs are usually located in several floors in office buildings. This corresponds to 10^5 APs per square kilometer. An RM covering a large region, such as a city, may hence have millions of APs. Thus it is clear that the RM as such is too large to download, store and update in a mobile device and should be compressed in order to achieve acceptable RM size in terms of bytes. Note that as such the byte consumption might not be an issue, but as the RM needs to be refreshed frequently, the cumulative data consumption can become unnecessarily high also in this use case.

5) Server resource considerations: Lastly, online positioning is very demanding in terms of server resources. Each positioning transaction requires a SSL/TLS connection, of which termination is very resource-hungry, and may become the performance bottleneck. Also, each request will result in a database lookup for, e.g. AP coordinates. Again, the database access may be become the bottleneck in high-performance services with hundreds of millions of users. Obviously, all of these bottlenecks can be alleviated by introducing more servers, improved load-balancing as well as introducing high-performance storage. However, this also increases the service operation expenses. Thus it is clear that offloading the online positioning servers results in advantages both in the service provider as well as user sides.

B. Case for Offline RMs

The previous discussion shows that the offline positioning offers some advantages over online positioning. In the rest of this paper we concentrate on showing how the size of the offline RM can be reduced enough so that the aforementioned bandwidth and storage problems are no longer issues.

Our approach is based on identifying the most significant APs in the RM and including only those in the offline RM. The idea of detecting the most significant APs stems from the fact that in big offices and public buildings there are usually several APs in the same location. These APs thus have the same coverage areas and are almost always detected simultaneously in an AP scan. In fact, from the positioning perspective this information is redundant and sometimes may even lead to degraded positioning performance [3].

Our approach uses fingerprint data not only to create an RM but also to detect the most significant APs in the RM. We define the most significant APs to be the smallest set of APs such that each of the fingerprints received in history could be served. Thus we consider each fingerprint to be a positioning request and we assume that the past positioning requests (fingerprints) correlate with the future positioning requests. We propose using only the most significant APs in the offline RM.

We propose two different methods to detect the most significant APs, a continuous mode and a batch mode. Batch mode uses fingerprint data that is collected over a pre-defined time window to find the most significant APs during that period. The continuous mode updates the significance data constantly as new fingerprints arrive. The set of significant APs found by the continuous mode is generally not as small as the set found by the batch mode, but on other hand the continuous mode is easier and cheaper to implement.

We also present a method to further compress the size of the RM by compressing the globally unique BSSIDs of WLAN APs (MAC address of WLAN AP). We assume that if the offline RM contains only a subset of the RM in the server, the compressed BSSIDs would still be locally unique BSSIDs. In addition, we evaluate the possible ambiguities that are caused by the BSSID compression in a real environment.

In order to evaluate the positioning performance of the reduced offline RM, we evaluate the accuracy, availability and consistency of the position estimates. Now, removing APs is a good compression method for offline RMs, but as always in compression, the tradeoff is that we lose in positioning performance. However, in contrast with much of the literature that is concerned with the small-scale systems, we assert that the most important performance metrics for large-scale positioning systems are time-to-first-fix and availability [1]. Therefore, we are willing to give up some accuracy, but obviously not too much, in exchange for a smaller offline RM that produces fast time-to-first-fix and maintains the availability. In any case the accuracy of crowd-sourced WLAN positioning service is in the order of tens of meters so increasing mean error by, say, 10 meters will not be significant. Many use cases, such as fetching local weather forecast or finding a restaurant in the neighborhood, do not require a meter-level accuracy but function perfectly well with, say, 50-m accurate positioning methods. Such accuracy is roughly the expected mean accuracy of the large-scale crowd-sourced WLAN-based positioning services. Therefore, as the positioning technology under consideration has anyhow an inherent error of several tens of meters, losing another ten meters of accuracy is not an issue. Having said that we also assert that whatever the accuracy is, the position estimate needs to be consistent, i.e. the associated uncertainty estimate must reflect true error. Thus we monitor consistency very carefully in addition to availability and, to lesser extent, accuracy.

II. Previous work

Some radio map compression techniques have been proposed in the literature earlier. The proposed techniques may be categorized into two different approaches. The first approach is to reduce the size of the RM in the fingerprint domain. These methods include reducing the dimensionality of the fingerprints or reducing the number of fingerprints in the database. The second approach is to compress the fingerprint data into statistical or physical models such as coverage area models or path loss models.

In the first category Ledlie [4] reduces the dimensionality of the fingerprints by assigning weights to each AP in the context of a single fingerprint. Weights are determined by signal strength values and AP observation frequencies. The APs with weights below a predefined threshold value are removed from a fingerprint. This approach finds the most important APs in each space, for example in a single room. Thus the same APs may have high importance in one space, but may be removed from a fingerprint collected in some other space.

Laitinen et al. [3] study various different AP significance measures that can be used to retain only a subset of APs in a single fingerprint. Results show that by removing redundant APs from fingerprints the positioning accuracy is better than with the original fingerprint data given that a proper significance metric is used. However, deciding which one to use is shown not to be a trivial problem.

In order to reduce the number of fingerprints in an RM, Arya et al. [5] try simple methods that map fingerprints to the spatially closest grid points. The per-AP average of the signal strength measurements in each grid point is set as the new signal strength measurement for each AP at the grid point. Publications [6] and [7] propose more sophisticated methods that use different clustering methods to combine a set of fingerprints into a single fingerprint. These clustering methods take into account also the radio layer measurements (e.g. signal strength values). A Block-based Weighted Clustering technique is proposed in [7] to compress the radio database. The technique is evaluated in a GSM cellular network and the results show that although the size of the compressed database is only 20% of the original the accuracy of position estimate remains the same. These results show that there is plenty of redundant information in the original fingerprint data.

The second category is to compress the fingerprint data into statistical or physical models. In these approaches the RM consists of model parameters instead of the collected fingerprints. The simplest method is to reduce the database size by constructing a statistical model of the coverage area of an AP or any other node in a wireless network [8–10]. In general, a coverage area is a 2D region in which the node is observable. However, due to the statistical approach and collection methodology (crowd-sourcing), a coverage area presents actually the distribution of the users that contribute the learning data. In the cited publications coverage areas are modeled as probability distributions whose parameters can be described using mean and covariance and can be visualized as an ellipse. Only five parameters need to be stored for each coverage area making the radio map very small in terms of storage consumption. A Gaussian distribution for coverage area models is proposed in [8, 9], while [10] proposes a Student-t distribution which is known to be more robust for outliers. These methods do not use signal strength information for the model generation. Thus the method is relatively robust against signal strength fluctuations, but does not achieve as good positioning accuracy as traditional fingerprinting methods.

In order to take the signal strength information into account in coverage area estimation, in [11] fingerprints are grouped based on signal strength values and coverage areas are independently fit to the different groups. In positioning phase only the coverage area that corresponds with the signal strength measurements is chosen for each heard AP. In [12] the fingerprint data is compressed into 2-dimensional Gaussian Mixture Models. This may be considered to be a more sophisticated method compared to a single Gaussian component and gives more realistic coverage area estimates than a single Gaussian, especially for multi-modal distributions.

Fingerprint data may also be compressed into pathloss models that take into account the physical radio propagation model. Nurminen et al. [13, 14] propose a Bayesian method to estimate pathloss model parameters. In this approach only the fitted pathloss model parameters are stored in the RM. They state that positioning accuracy is comparable to traditional

fingerprinting methods.

In general, the more sophisticated the model is, the better the positioning accuracy. However, at the same time the methods also become more sensitive to environmental characteristics. For instance, while in rural areas a simple two-parameter pathloss model suffices, in urban environment with complex propagation environment such a simple model may be totally inadequate. Also, more complex methods typically also have more configuration parameters for which it may be difficult to find a global optimum.

One very significant benefit of the model-based approaches is the predictable size of the RM. As mentioned before, when modeling an AP using a coverage area, only five integers are required (center point and ellipse parameters). Thus the RM size is proportional to the AP count with modest loss of accuracy. By contrast, in case of a fingerprint database the growth of the database may be uncontrolled, but at the same time accuracy does not suffer. In general, as noted in [15], the accuracy and bandwidth consumption correlate.

Finally, some publications propose methods to select only a subset of the observed APs [16–19]. These methods include MaxMean, which selects APs based on average received signal strength value and InfoGain, which attempts to remove those APs that add the least information to the position estimate. In addition, methods such as Principal Component Analysis, Discrete Cosine Transformation and Independent Component Analysis have been proposed for the AP selection. However, in all of these methods the AP reduction is made only in the positioning phase after the positioning data (fingerprints, path loss models, coverage areas etc.) have already been downloaded to the mobile device. The motivation in these papers has thus been to rather reduce computational complexity and power consumption than to reduce the actual size of the RM.

Interestingly, most of the reviewed publications only consider the changes in positioning accuracy, when the RM gets compressed using various methods. However, practically none of the publications study changes in availability and consistency, which we consider as important characteristics as accuracy.

III. Detecting the most significant APs

Busy environments such as shopping malls, offices and university campuses typically have lots of WLAN capacity installed. This means that at any given location multiple APs (even up to several hundred) can be observed. Moreover, the controlled deployments by IT departments often use WLAN transmitters that provide service with multiple BSSIDs. Usually the signal environment and coverage of such APs is the same and from the positioning perspective such extra APs do not add any extra information [3]. When the size of the RM needs to be limited, the natural choice is to find the redundant APs from the RM and retain only the most significant ones. We propose two methods to detect these most important APs using fingerprint data. Section III-A proposes a batch mode and Section III-B proposes a continuous mode. The assumption is that there is a remote server that has the capability to process the fingerprint data as well as to learn and save the significance data for each of the observed APs. The significance data may be for example a binary flag that indicates the most significant APs.

A. Batch mode

In the batch mode the fingerprints that have been collected over a certain period of time (such as a week or a month) are analyzed in order to find the most significant APs. Each fingerprint consists of a set of APs that are observed at a certain location.

Assume that we have n fingerprints, $F = (F_1, F_2, \ldots, F_n)$, and in this fingerprint set altogether m unique APs, denote as $AP = (AP_1, AP_2, \ldots, AP_m)$. The information from the fingerprints is used to create a binary array M of size $n \times m$ in which each row corresponds to a fingerprint and each column corresponds to an AP. Now, $\mathrm{M}_{ij} = 1$ if AP_j is observed in fingerprint F_i, otherwise $\mathrm{M}_{ij} = 0$. The array M contains only the binary observation information. No other information, such as received signal strength values, is used.

Detection of the significant APs can be formulated as a problem known in combinatorics as the set covering problem. The batch mode algorithm is a greedy algorithm that makes a locally optimal AP selection at each iteration and is known to be the best possible polynomial time approximation for the set cover problem. To detect the most significant APs, do:

1) Calculate the sum of every column:
$$S = (\sum_{i=1}^{n} \mathrm{M}_{i1}, \sum_{i=1}^{n} \mathrm{M}_{i2}, \ldots, \sum_{i=1}^{n} \mathrm{M}_{im_t})$$
2) Find
$$S_k = \max(S)$$
and choose AP_k to be a significant AP
3) Delete the rows in M where $\mathrm{M}_{r_i k} = 1$.
4) Delete column k from M
5) If M is empty, stop, else go back to 1

In this algorithm, the number of columns, m_t, in array M changes in every iteration. In the first iteration $m_t = m$ and in the following iterations $m_t < m$

Figure 1 shows an example of AP reduction in the batch mode. Array M contains five APs ($m = 5$) and five fingerprints ($n = 5$). Now columns 3, 4 and 5 have the same column sum, so AP_3 is chosen to be a significant AP (note that equally well AP_4 or AP_5 could have been chosen). Because AP_3 appeared in fingerprints F_1, F_2, F_5, delete rows $\{1, 2, 5\}$ from the array M, then delete column 3. After the reduction the array M is 2-by-4. In the resulting array M columns 3 and 4 (now corresponding to AP_4 and AP_5) have the same column sum and AP_4 gets chosen as a significant AP (again equally well AP_5 could have been chosen). The reduction step removes the rows $\{1, 2\}$ (now corresponding to fingerprints F_3 and F_4) and thus the algorithm stops. In the end AP_3 and AP_4 are significant APs as those two are sufficient to position all the considered FPs.

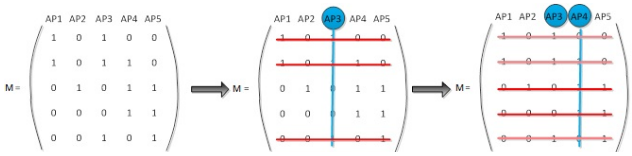

Fig. 1. An example of the AP selection in batch mode

The batch mode returns a set of indices $k = [k_1, \ldots, k_p]$ such that the sub-array $\mathrm{M}(:, k)$ does not have any zero rows. In other words, a set of APs such that every fingerprint in the history contains at least one AP from the set. With this set of APs all the considered fingerprints can be located. The previous example shows that the batch mode does not give a unique set of APs and thus the set of APs may not always be the optimal set of AP.

The batch mode requires that the server must collect and save the fingerprint data over some period of time before being able to determine the significant APs. Storing such data may be a privacy risk. Also the dimension of the array M may become excessively large making the data processing difficult or even impossible. Thus the continuous mode, which is presented next, is more suitable for real life implementation.

B. Continuous mode

In the continuous mode the server processes the fingerprint data constantly as new fingerprints arrive. To find the significant APs, process the fingerprints one at a time as follows:

1) Get the significance data for all of the APs that are observed in the fingerprint
2) If none of the APs in the fingerprint are significant, select some or all the APs in the fingerprint to be a significant AP
3) Continue to next fingerprint

As specified, if at least one of the APs in the fingerprint is already marked as a significant, the received fingerprint could have already been positioned with that and nothing needs to be done. In the contrary case, i.e. if none of the APs are significant, some or all of the APs in the fingerprint need to be marked as significant APs so that there will be at least one significant AP that allows locating the fingerprint.

It remains to decide how to choose the significant APs in case none of the APs have prior significance information. One option is to set the significance flag to all of the APs in the fingerprint. However, this causes the number of significant APs to grow fast. Other choices are also possible such as picking one or more APs randomly. However, the actual choice of the method is not critical here. In our test we choose the AP with the highest received signal strength value. The AP with the strongest received signal strength value could be considered as the most important AP for that single fingerprint.

The advantage of the continuous mode is that the fingerprints may be processed as they arrive and need not be stored for an extended period before processing, which is important from the privacy point-of-view. This also allows uncoupling processes related to the learning of the significant APs and the offline RM generation. The offline RM can be generated anytime simply by selecting the APs that have been detected to be significant over a certain period of time. The drawback of the continuous mode is that, in general, the set of significant APs is larger than the set that is found using the batch mode, that returns near-optimal set of APs.

C. Time considerations

The AP radio environment changes constantly as APs get moved and new ones are introduced. Hence for the best performance the RM needs to be updated constantly. Similarly, the set of significant APs changes in time. Thus it is important to update the significance data periodically. Such consideration is taken into account in the batch mode in a natural manner. The fingerprint data could be collected, e.g., for a month, after which it is used to detect the significant APs to be included to the next release of the offline RM. In the continuous mode there could be a moving time window, say, one month. This means that whenever a new fingerprint arrives, the system checks whether any of the APs in the fingerprint has been significant during the last month. This way a fresh offline RM could be generated anytime and will contain the significant APs from the previous month.

Another question is how often the offline RM should be updated to the mobile device. A minor change in the set of significant APs does not affect accuracy and availability significantly. Therefore, the system should wait until there are enough changes before releasing a new version of offline RM for download, because every update to the offline RM consumes data and server resources.

IV. Compressing BSSIDs of APs

The globally administered WLAN AP Basic Service Set Identifications (BSSIDs, that is the MAC address of the physical radio) are, by definition, globally unique. The BSSID is a 6-byte number and in case of a globally-administered BSSID the upper three bytes is the IEEE-granted Organization Unique Identifier (OUI) and the lower three bytes is the OUI-specific part. BSSID conflicts can be prevented this way. However, there are also locally-administered BSSIDs, which may clash, but they can be recognized from the BSSID (the 2nd least-significant bit of the most significant byte encodes whether the BSSID is locally or globally administered).

Six bytes for an AP ID is a major component in the offline RM byte consumption. Thus, it is tempting to attempt reducing this byte consumption.

Cryptographic hash function is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string. Hash functions are used in, for example, message integrity verification and password verification [20]. We propose that BSSIDs in the offline RM could be compressed by hashing the 6-byte globally unique BSSIDs into 2-byte locally unique BSSIDs.

Hashing the 6-byte BSSID to 2-byte BSSID means that multiple BSSIDs are mapped to the same 2-byte value. In case all the possible 6-byte BSSIDs (range $[0, 2^{48} - 1]$) are evenly

mapped to the 2-byte values (range [0, $2^{16}-1$ = 65535]), there will be 2^{32} BSSIDs mapped to each hash value. In general this would cause ambiguities in positioning phase so that in the database there may be several matches (several locations, models) for each observed AP. However, we assume that the offline RMs carries data for a small well-defined area (such as a city or a state) so that in this space the compressed BSSIDs are still practically unique.

If ambiguities occur, their impact can be mitigated easily by simple outlier detection, for example by comparing the AP location with the locations of the other observed APs. Figure 2 shows an exemplary situation with the white stars being APs that have a locally unique ID in the offline RM. For one compressed ID of the observed APs there are four different locations in the RM. Those locations are marked with grey stars. However, it is easy to determine which location is the correct one, because only one of the locations is close to the other AP locations. In addition, other sources of information can be used in the ambiguity mitigation. In the case of filtering, the previous position estimate gives prior information based on which location can be chosen. Also, if position information from other sources is available, such as coarse cellular position estimate, it can be used to determine the correct AP location. In difficult situations APs with ambiguities could be ignored from position calculation if there is at least one AP with a known location.

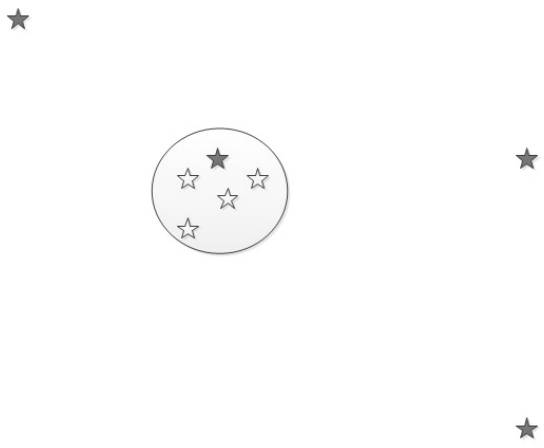

Fig. 2. An example of resolving ambiguities

In this work we use CRC-16 ITU-T algorithm to compress BSSIDs. Cyclic Redundancy Check (CRC) is an error detection code and is primarily used in data communications to determine whether an error has occurred within a large block or stream of information bytes [21]. CRC-based methods add a fixed-length checksum value at the end of the message. The CRC checksum calculation needs a generator polynomial that is used to divide the original message [22]. CRC-16 ITU-T algorithm uses the polynomial

$$x^{16} + x^{12} + x^5 + 1$$

to divide the message. In this work we use this polynomial to divide the original BSSID and use the 2-byte remainder as the compressed BSSID. We use CRC because the algorithm is easy to implement and CRC checksums are known to have minimal overall collision probabilities. This means that the algorithm maps the 6-byte BSSIDs evenly to 2-byte BSSIDs which imply as few ambiguities as possible. Figure 3 gives an example of the previous property. The test data is 25 million real-world unique BSSIDs that are hashed with the CRC-16 ITU-T algorithm. In the optimal case the 25 M BSSIDs would be evenly distributed in the range [0, 65535] of the hash function domain, i.e. 387 BSSIDs would map to each value of the range. Figure 3 shows that the BSSIDs get distributed fairly evenly throughout the whole range. The number of BSSIDs mapped to each value varies between 305 and 468 with the mean of 387, as desired.

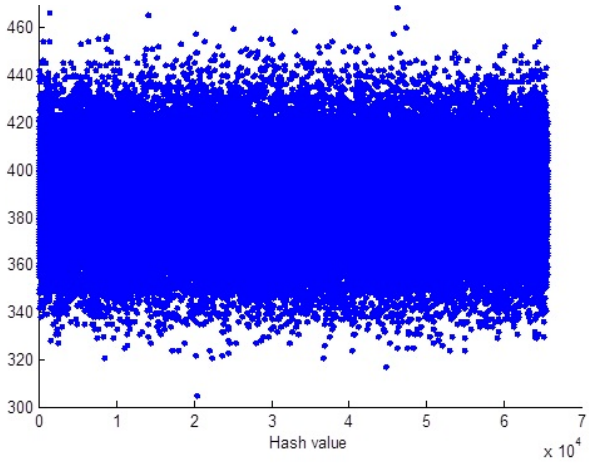

Fig. 3. An example of the distribution of mapped BSSIDs

The CRC algorithms are, obviously, not the only possible choices. The choice of the algorithm may depend for example on the size limitations of the RM or positioning performance requirements. It is obvious that if the size of an RM is the limiting factor, the BSSIDs could be compressed more. This, however, implies more ambiguities. If the positioning performance is the limiting factor, the compression rate for BSSIDs could be smaller, e.g. by using a three-byte hash value, and thus the amount of ambiguities would be smaller.

V. Experimental results

The proposed technique is tested in Wroclaw city center, Poland. The test data consists of 34 445 fingerprints that are used to find the most significant APs and build the (reduced) RM. Altogether 8222 APs were detected during the training phase. In addition, 433 fingerprints (different from learning data) are used to evaluate the positioning performance of the offline RM. The test area is approximately 1.5 km $\times$ 2.5 km. The fingerprints are compressed into coverage area models, as presented in [8] and [9]. The coverage area -based positioning methods do not give the best possible positioning accuracy, but as mentioned in Section I-B, we rather concentrate on changes in availability than on positioning accuracy. In sections V-A and V-B we present the experimental results on how the

positioning performance changes with the offline RM as the AP count gets reduced and BSSIDs of the APs are compressed, respectively.

A. AP reduction

Table I shows the results for availability, positioning accuracy and consistency with five different RMs and the number of APs that was present in each RM. The first RM is the original RM that contains all the APs that were observed in the training phase, referred to as fullRM. The second RM contains the significant APs found using the batch mode, referred to as batchRM. The third RM contains the significant APs found using the continuous mode by selecting all the APs to be significant if none of them were already a significant one, referred to as contAllRM. The fourth RM contains the significant APs found using continuous mode by selecting only the strongest AP to be the significant one if none of them were already a significant one, referred to as contStrongRM. Finally, in the fifth RM 9% of the original APs are selected randomly, referred to as randomRM. Thus the number of APs in randomRM is more or less the same as in contStrongRM. RandomRM is used to verify that the proposed scheme for systematic detection of the significant APs is better than simply selecting the APs to the offline RM randomly.

The Availability column in Table I describes how many of the 433 test fingerprints yielded position with the different RMs. The Error column shows the mean error in meters with respect to the GPS-based position for the successful cases. The column Consistency shows in how many cases the true position lies inside the 68% confidence ellipse. A positioning method is described in [23] to be consistent if the percentage is greater or equal to 68%. The last column in Table I shows the number of APs in each RM and percentage of APs compared to the fullRM.

TABLE I
NUMERICAL RESULTS

RM type	Availability [%]	Error [m]	Consistency [%]	Number of APs
fullRM	94	65	21	8222
batchRM	91	82	54	546 (7%)
contAllRM	93	69	35	2108 (26%)
contStrongRM	91	75	50	702 (9%)
randomRM	69	74	53	777 (9%)

The results show, as expected, that the batchRM results in the smallest set of APs. The number of APs in the RM is only 7% of the original and the availability drops only three percentage points, from 94% to 91%. The mean error increases 17 m. This was, however, expected and the accuracy drop is not too significant for the considered use cases. When finding the significant APs by using the fingerprints as proposed in Section III, the result is that the APs with largest coverage areas are chosen to be the significant ones, because they are observed the most often. As the positioning accuracy is worse the larger the coverage area, the result is expected.

Regarding the continuous mode, the results show that the contAllRM achieves essentially the same availability and accuracy as the full RM with only one quarter of the original APs. This can be considered to be an excellent result. Moreover, consistency is much higher with the contAllRM than with the fullRM. While the consistency improvement may be unintuitive, the explanation lies in the characteristics of the used positioning algorithm. The uncertainty estimate produced by the algorithm combines the coverage area sizes in the inverted domain so that the smaller coverage areas dominate. The uncertainty estimate is always smaller than the smallest coverage area and the size decreases as $1/\sqrt{n}$ as more APs are taken into account. Thus, the smaller amount of APs in positioning (with the fullRM the average number of APs used in positioning is 14, with contAllRM seven and with the rest just two) the better the consistency. In general, the position algorithm does not fulfill the consistent condition proposed in [23] with any of the RMs.

Finally, the most interesting method, contStrongRM performs very well. The availability, accuracy and consistency penalty are negligible and the resulting reduced RM has only 9% of the original APs.

To conclude, the results show that using fingerprints to find the significant APs is a powerful technique. The availability with randomRM is only 69%, whereas with the other RMs the availability is over 90%. Based on the relative complexity of the different methods as well as the results shown in Table I, the contStrongRM is the most promising technique for real life implementation. It is the easiest to implement and yields better positioning availability and accuracy than batchRM. However, even though it cannot find the totally smallest set of significant APs, the number of APs is still very low compared to the fullRM, only 9% of the fullRM.

B. BSSID compression

In this section we evaluate how the compression of the BSSIDs impacts the positioning performance. There are 8222 globally unique 6-byte BSSIDs in the fullRM and 7819 2-byte IDs after compressing all the BSSIDs. Therefore, there are only a few ambiguities.

Looking from the positioning request perspective, with the fullRM the number of positioning cases in which ambiguities occur is 334 out of 433. In most of the cases the ambiguities could be resolved by the simple outlier detection that is illustrated in Figure 2. This method is based on simply picking the APs that are within a predetermined distance from the median of the AP locations. Moreover, in case there are at least two APs with unique location in the RM, the median of those is used as the reference point. Note that the number of ambiguities is lower when there are fewer APs in the RM. For example, contStrongRM has 702 globally unique BSSIDs and after compression the number of unique APs is 695. In the positioning trials ambiguities occur only in 124 cases out of 433. Therefore, the AP reduction and BSSID compression techniques can be used together very effectively.

The most difficult situations occur when in the fingerprint there are solely APs with multiple locations after the BSSID compression. In this case it may be difficult to find a good

set of APs that are close to each other. If this happens, one option is to ignore all the APs from the position calculation or to take all the APs into the position calculation. The first option, however, decreases the availability and the second option increases the positioning error. This can be considered to be a service-level design choice.

Figures 4 and 5 summarize the availability and accuracy results, respectively, with fullRM, batchRM, contAllRM, contStrongRM and randomRM with and without BSSID compression. For all of the cases the reference is the dark blue bar denoted as Original RM without the BSSID compression. For the RMs with compressed IDs there are three different scenarios: no outlier detection at all, outlier detection as described earlier with all the APs used in difficult situations ("take all") and outlier detection as described earlier with difficult cases are simply dropped ("ignore all").

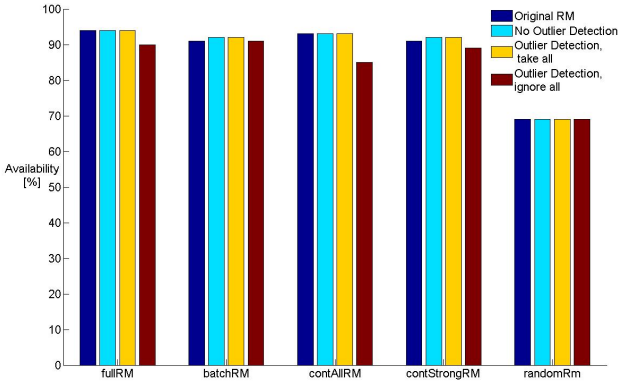

Fig. 4. Availability with RMs with compressed BSSIDs

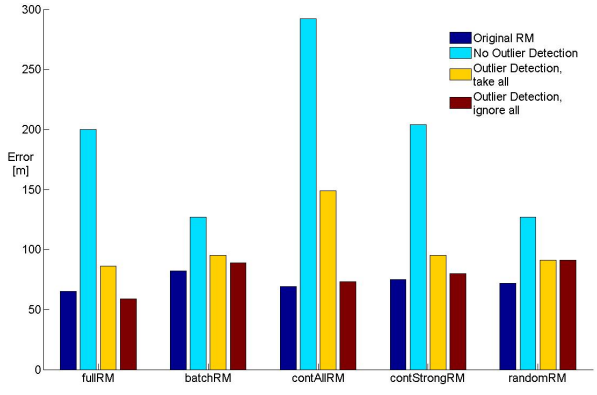

Fig. 5. Positioning error with RMs with compressed BSSIDs

The results show that the availability is practically the same across all the outlier detection scenarios for each RM type except for the obvious drop, when difficult cases are simply ignored. Interestingly, in some cases availability is even slightly better with the RM with compressed IDs than with the original RM. Unfortunately, in these cases the position estimate is typically wrong, which can be seen as large increase in mean errors in Figure 5 for the scenarios without outlier detection (cyan) and take all approach in difficult cases (yellow).

Clearly, from the error perspective a conservative outlier detection mechanism is desired. However, the mechanism does not need to be too complicated. Even simple outlier detection in case of ambiguities improves the positioning accuracy with only a slight impact on availability. As discussed in Section IV, the ambiguities could be reduced even further if some supportive positioning information would be available. Also, more advanced clustering techniques could be used for outlier detection. However, the results show that the simple method used here is sufficient for the majority of the cases and thus the use of more complex methods is not justified.

In general, considering that with the BSSID compression the byte-consumption of the RM is further reduced to almost one third of the original, the resulting loss in availability and accuracy can be considered acceptable. With contStrongRM, which is the recommended option in Section V-A, the availability with the RM with full and compressed BSSIDs is 91% and 89%, respectively, even if location is not provided in difficult cases. The corresponding mean errors are 75 m and 79 m, respectively, meaning that the accuracy loss is acceptable.

To conclude, as shown in Table I the availability and mean error with the fullRM with full BSSIDs is 94% and 65 m, respectively. As shown, by selecting only 9% of the original APs (contStrongRM method) and by further compressing the size of the RM by compressing the BSSIDs, the availability drops only 5 percentage points and the mean error increases only by 14 meters. Again, these penalties can be considered acceptable in applications using crowd-sourced WLAN-based positioning.

VI. Conclusion

This paper presents a method for generating an RM that is suitable for offline positioning, i.e. it is small in size, but still provides adequate positioning quality. The method uses WLAN fingerprint data to find the significant APs that are then included in the offline RM. We present different methods to select the most significant APs and propose using only those in the offline RM. We also present a method to further compress the RM by hashing the BSSIDs into non-unique IDs. However, we assume that if the offline RM contains only a small region, such as a city, the rate of collisions in the compressed BSSIDs is manageable. We evaluate the availability, accuracy and consistency of these methods with real WLAN data.

The results show that the RM can easily be compressed by selecting only a subset of APs in the offline RM. Choosing the most significant APs to the offline RM is superior as compared to choosing the similar size AP subset randomly. Results show that roughly the same positioning availability and accuracy as with the original RM can be achieved by selecting only 10% of the original APs wisely. Results also show that the BSSID compression is an efficient method to compress the offline RM further in terms of bytes. Some ambiguities occur in our

test cases, but those were easily mitigated by a simple outlier detection method.

Acknowledgment

This research was funded by HERE, a Nokia Business. The authors are grateful to Mikko Blomqvist and Dr. Tech. Jari Syrjärinne of HERE for their support and advice.

References

[1] L. Wirola, T. Laine, and J. Syrjärinne, "Mass-market requirements for indoor positioning and indoor navigation," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Zurich, Switzerland, September 2010.

[2] "RFC 6176," March 2011.

[3] E. Laitinen, E. S. Lohan, J. Talvitie, and S. Shrestha, "Access point significance measures in WLAN-based location," in *Proceedings of the 9th Workshop on Positioning, Navigation and Communication 2012 (WPNC'10)*, Dresden, Germany, March 2012.

[4] J. Ledlie, "Method and apparatus for on-device positioning using compressed fingerprint archives," Patent WO 2011/067 466, June 9, 2011.

[5] A. Arya, P. Godlewski, and P. Mellé, "Performance analysis of outdoor localization systems based on RSS fingerprinting," in *Proceedings of the 6th international conference on Symposium on Wireless Communication Systems (ISWCS'09)*, Piscataway, NJ, USA, 2009.

[6] A. Arya, P. Godlewski, and P. Melle, "A hierarchical clustering technique for radio map compression in location fingerprinting systems," in *Proceedings of the 71th IEEE Vehicular Technology Conference, VTC Spring 2010*, Taipei, Taiwan, May 2010.

[7] A. Arya, P. Godlewski, M. Campedel, and G. du Chene, "Radio database compression for accurate energy-efficient localization in fingerprinting systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1368–1379, 2013.

[8] L. Koski, R. Piché, V. Kaseva, S. Ali-Löytty, and M. Hännikäinen, "Positioning with coverage area estimates generated from location fingerprints," in *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010 (WPNC'10)*, Dresden, Germany, March 2010.

[9] L. Koski, T. Perälä, and R. Piché, "Indoor positioning using WLAN coverage area estimates," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Zurich, Switzerland, September 2010.

[10] R. Piché, "Robust estimation of a reception region from location fingerprints," in *International Conference on Localization and GNSS*, Tampere, Finland, June 2011.

[11] M. Raitoharju, M. Dashti, S. Ali-Löytty, and R. Piché, "Positioning with multilevel coverage area models," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, Sydney, Australia, November 2012.

[12] K. Kaji and N. Kawauchi, "Design and implementation of Wifi indoor localization based on Gaussian mixture model and particle filter," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, Sydney, Australia, November 2012.

[13] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, Sydney, Australia, November 2012.

[14] ——, "Statistical path loss parameter estimation and positioning using RSS measurements," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS2012)*, October 2012.

[15] L. Wirola, "Studies on location technology standards evolution in wireless networks," Ph.D. dissertation, Tampere University of Technology, 2010.

[16] S.-H. Fang, T.-N. Lin, and P.-C. Lin, "Location fingerprinting in a decorrelated space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 685–691, 2008.

[17] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Los Alamitos, CA, USA, 2003.

[18] Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 7, pp. 877–888, 2006.

[19] S.-H. Fang and T.-N. Lin, "Principal component localization in indoor WLAN environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, 2012.

[20] M. Cochran and U. of Colorado at Boulder. Computer Science, *Cryptographic Hash Functions*. University of Colorado at Boulder, 2008.

[21] L. Null and J. Lobur, *The Essentials of Computer Organization and Architecture*. Jones & Bartlett Learning, 2010.

[22] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," in *Proceedings of the IRE*, January 1961.

[23] S. Ali-Löytty, N. Sirola, and R. Piché, "Consistency of three Kalman filter extensions in hybrid navigation," *European Journal of Navigation*, vol. 4, no. 1, pp. 33–40, February 2006.