CS/RADAR: Indoor Location Discovery and Tracking

Dan Gibson and Jake Adriaens
University of Wisconsin-Madison Department of Electrical and Computer Engineering
{degibson, jtadriaens}@wisc.edu

Abstract

Recent work done at Microsoft and elsewhere suggests that the concept of RF-based location and tracking (RADAR) is useful from the perspective of user location and tracking in local-area networks. RADAR operates by sampling signal strengths from some number of nearby transmitters. These samples are compared against samples of known locations, stored in an RF-based map.

We have replicated a portion of the Microsoft RADAR system, and speculate on possible adaptations of the RADAR concept for use in ubiquitous computing scenarios and wireless sensor networks. We have extended the RADAR concept to include a client/server model, which significantly reduces storage and processing requirements of the mobile device.

1 Introduction

Recent work done at Microsoft and elsewhere suggests that the concept of RF-based location and tracking (RADAR) is useful from the perspective of user location and tracking in local-area, wireless-based networks. Ability to locate mobile users (nodes) allows a system to tailor itself to changing network topology and to provide services based on user location. For instance, a network print request from mobile user could be automatically forwarded to the nearest public printer without requiring the user to manually determine which printer is closest to his/her location.

RADAR is also useful from the perspective of ubiquitous computing. Locations determined from RADAR-equipped sensor nodes could be used to dynamically control room temperatures, lighting conditions, or air circulation based on the number of users in a particular area. It is because of this omnipresent-monitoring property that RADAR is also useful to wireless sensor networks: it is conceivable that a RADAR-like system could be combined with data-dissemination protocols to determine which nodes in networks may be eligible to enter a sleep state to conserve power based on proximity of mobile, RADAR-equipped data sinks [Kim]. That is, given a geographic routing scheme, RADAR could be used to quantify a given node's routing usefulness.

Unfortunately, RADAR requires an RF-map of the sensing field, usually generated (manually or mathematically) prior to deployment. For each mobile sink or node to store and process a large RF-map is an unreasonable request for a wireless sensor network platform. Hence, we have developed a client / server RADAR system, (CS/RADAR) which permits mobile nodes to use RADAR without knowledge of local RF conditions and without significant computation.

2 Related Work

The RADAR concept is a product of Bahl et. al. at Microsoft Research. Much of the work presented here is intended to replicate some of the results presented in "A Software System for Locating Mobile Users: Design, Evaluation, and Lessons". RADAR is a representative of location discovery algorithms using RSSI (Received Signal Strength Indicator) -based or simpler connectivity-based strategies. Among these (but not exclusively limited to) are Daedalus [Balakrishnan] and the Duress Alarm Location System [Christ].

RSSI-based algorithms need not use radio-frequency information as in the case of RADAR—indeed schemes exist using infared signals [Want], and many other media could be used in a RADAR-like system (eg. ocean currents).

Our work is primarily intended to replicate the work done by Microsoft Research. It differs only in that we present RADAR as a network service. Like the work done at Microsoft, our implementations use no RADAR-specific hardware and form an overlay of an existing network structure.

3 Assumptions

Use of RADAR as a location discovery mechanism in mobile computing in the context of a wireless network assumes that:

- An RF-map to which samples of the current RF conditions will be compared exists, either pre-mapped or mathematically generated.
- Transmitters (base stations) used to generate the RF-map are fixed in position and number, and do not experience periods of non-functionality; they emit a constant-amplitude radio signal.
- Mobile users/nodes are able to sample the RSSI of each base station within communication range.
- RF conditions do not change due to environmental factors (e.g. opening doors, cordless phones, microwave ovens, etc.).

In addition to the assumptions of general wireless networking, to adapt RADAR to use in wireless sensor networks (WSNs) further assumes that:

- Acquiring the needed information for RF-sampling is not costly in terms of power or computing.
- The RF-map may be accessed without incurring cost of large storage or computational requirements, and the cost of RADAR in terms of power is low.

The concept of CS/RADAR in WSNs requires a similar set of assumptions. Namely, we assume:

- Some number of nodes have fixed positions—these nodes are used to generate an initial RF-map of the sensing field.
- At least one base station exists where the RF-map may be stored. This base station is presumed to have greatly increased storage and computational capability.
- Some underlying communication infrastructure exists to allow communication between mobile nodes/sinks and base stations.

Note that it is not necessary that the position of each node in the sensing field be known for RADAR to be useful—they need only generate RF traffic. The positions of other non-mobile nodes are not used in RADAR-based location.

4 CS/RADAR Operation

In a RADAR-based location system, a mobile node or sink generates a sample of the RF-conditions at its location each time it wishes to determine its location. This sample contains an RSSI value for each node in communication range of the mobile node. For many communication strategies, the generation of this sample does not require any extra effort on the part of the mobile node/sink; connectivity-based information would already be gathered in most routing schemes, and some indication of signal strength is often available from RF hardware (an RSSI register in 802.11b implementations, for instance). It is even possible for a mobile sink/node to estimate an approximate signal-to-noise ratio (SNR) by considering the number of lost packets in a given time frame. The sample may include SNR or other parameters, such as noise level or channels in use. The sample is intended to represent the RF characteristics endemic to a particular locale, regardless of how a sample is represented.

Once a sample has been generated, the sample is compared to similar samples in the RF-map describing the region. The exact method of comparison is implementation-specific, and greatly affects the accuracy and granularity of a location divined in this way. In general, the simplest classes of algorithms compare samples against known position/sample pairs and report the mobile node's location as the position of the position/sample pair that most closely matches the current sample. More advanced algorithms attempt to interpolate position by employing cost-based heuristics and averaging of multiple position/sample pairs. Yet a third approach involves keeping a history of the last N positions of a node. Interpolation and path-based heuristics are used in conjunction with this location history to produce a likely current position.

CS/RADAR differs from the above scheme in that the RF-map is not stored locally at the mobile node. Instead, the mobile node submits

its sample via the underlying network infrastructure to a base station, which advertises a CS/RADAR positioning service. This base station performs the necessary algorithm(s) to estimate the mobile node's position, and returns that information to the node, again employing the underlying routing scheme. This separation of sampling from position divination is advantageous in WSNs for a number of reasons:

- Mobile devices need not store an RF-map locally.
- No device is privy to the information in the RF-map, which could be used maliciously in some environments.
- Client / server models allow individual nodes to be queried for their positions in a simple manner.
- Locale-switching does not require a device to acquire a new RF-map; the node need only communicate with a different server.

The client/server model lends itself to many networking situations, and WSNs in general, simply because of the divisibility of the RADAR concept into the sample / compare stages.

5 Location Discovery

We implemented five algorithms for location discovery. The first of these algorithms we called "match by audible nodes" (MBAN). MBAN determines position by weighting a location based on the number of base stations from the node sample that match the base stations in the map sample. As an example, assume we took a sample and were able to hear base stations A and B. In our map assume location 1 (L1) hears base stations A, B and C, while location 2 (L2) hears base stations A, C, and D. In this example L1 would receive a weight of 2 and L2 would receive a weight of 1. This indicates that L1 is the more likely position.

The second algorithm is called "choose the smallest MBAN" (CTSM). The purpose of CTSM was to fix a problematic area in MBAN. The specific issue that arose is illustrated here: Assume you node sample hears A and B. Now assume the map has L1 with A and C, and L2 with B, C and D. In this case both L1 and L2 would receive a weight of one. Through experimentation we found position L1 is much more likely in this case because the probability of not detecting 1 node that should be withing communication range are much greater than the odds of not hearing 2 nodes. To address this issue, CTSM breaks ties between locations with the greatest weight in MBAN by choosing the location that has the least amount of base stations heard (among the tied solutions).

The third location discovery algorithm we implemented showed the best results in predicting position. It has been dubbed "CTSM plus signal strength" (CPSS), and is denoted "M3" in (6). We developed CPSS in order to increase the prediction resolution of CTSM. CPSS works by first running CTSM, and taking all the locations that tie as potential candidate locations. CPSS further narrows down the candidate pool, considering the difference in measured signal strength to the signal strength recorded in the map. CPSS then choses the location with the least difference, as this is the most likely position.

The fourth and fifth location discovery algorithms were not very successful in predicting locations. These algorithms were interpolating position and guessing position based on the last known position (history-based, as described above). We tried many different implementations of these algorithms but were unsuccessful on improving any of the previous three algorithms.

Interpolation was based on the best three candidates for our position returned from the CPSS method. Once we had these three candidates we took a weighted average of the coordinates from the three candidates as our position. Typically, these candidate postions were far from each other, and yielded poor results. We believe that with the resolution we were trying to achieve there was not enough difference in IEEE 802.11b RF-characteristics over small distances in our map. This caused the second and third best solutions to be more like random points from a large area that we could be in, rather than actual likely points we were at.

Using history to weight possible positions was unsuccessful for one main reason: When predicting positions it was common for the guessed position to deviate slightly from your actual position. When this deviation was opposite from the direction we were actually traveling, weighting history would cause it to be unlikely to make the jump from the bad guess to the ever-more-distant current (true) position. Using history was, however, successful in eliminating the large jumps we see occasionally with CPSS. It also worked well if you watched your position as you where moving, and when it got off track you went back to where it was predicting; this would allow the algorithm to track correctly again. We believe some fine tuning of the weighting parameters might be able to be done to fix the issue, but it is not a good idea to use in practice: once the prediction starts going wrong, it rarely corrects itself.

6 Methodology

We implemented CS/RADAR on the third floor of a three-story building. We chose the east wing of the building because we believed its layout would produce distinct RF characteristics throughout the area. We relied on an in-place system of IEEE 802.11b access points to produce the sampled conditions; the positions of these access points were not known.

Our implementation of CS/RADAR was written for Linux and Windows 2000 environments; these environments were chosen for their academic usefulness and GUI capability; CS/RADAR could be implemented on virtually any platform.

We performed our experiments in two phases: in the first phase we generated RF-maps of our target area with varying sample densities. These maps were generated empirically by sampling many positions in our target area and storing the position/sample pairs for later use in RF-maps. The second phase called for a number of traversals of our target area, employing different location algorithm, RF-map density and movement speed combinations.

6.1 Mapping

Three RF-maps were generated during the mapping portion of this project. Maps `sparse`, `moderate`, and `dense` have increasing levels of sample density. Samples in `sparse` were generated taking only one sample per room, and allowing one sample at the intersection of each hallway. The `moderate` map includes all samples in the `sparse` map, three samples spaced equidistant between all `sparse` hallway samples, and three to four additional samples in each mapped room. Map `dense` contains roughly one sample per square yard of mapped area, though does not necessarily include the same samples as found in `sparse` or `moderate`.

To reduce the effect of antennae orientation, all samples in this phase were taken facing the same compass direction. Additionally, we allowed a fixed time to elapse between arriving at a location and sampling to allow transient effects specific to our implementation's sampling method to disappear.

The effectiveness of these maps in combination with the CPSS algorithm discussed above is discussed in section 6, below.

6.2 Pathing

A single path through our mapped area was chosen for testing of CS/RADAR. We evaluated each of our three sampling densities with two sampling schemes: 1-Sample and 3-Sample Majority Vote. All pathing experiments were performed with only a single mobile node, for manpower constraints only. The CS/RADAR system could allow multiple mobile nodes with minimal additional overhead.

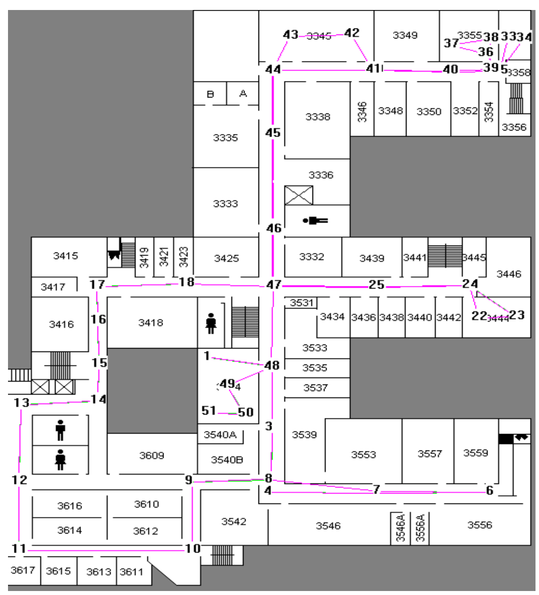

Fig. 1 Evaluation Path

The testing path was selected to attempt to challenge the location discovery mechanism with architectural trials as well as to provide regions where location divination would be relatively straightforward. We opted to include many hallways, individual rooms, and staircases—especially windowed areas—to maximize the diversity of our testing environment.

7 Performance/Results

7.1 Evaluation Criteria

The obvious criterion for a location discovery system is the discrepancy between the indicated location and the true location. We have used this difference as our primary evaluation criteria, though we have included others, specific to indoor location discovery techniques:

- Room-correctness: A sample is room-correct if the indicated position corresponds to the same room as the true position.
- Closest-point correct: A sample is closest-point correct if the algorithm returns the best possible match to a given location. Some algorithms only return positions that were mapped during the mapping phase—it is impossible for these algorithms to be exactly correct in most cases.
- Hit: A sample is a "hit" if the true location corresponds to a mapped location and the algorithm returned that location as the indicated position.
- Miss: A sample is a "miss" if the true location corresponds to a mapped location, but the algorithm returned a different location as the indicated position.

7.2 Single Sample Performance

We evaluated each of our generated maps (`sparse`, `moderate`, and `dense`) using only a single sample of RF-conditions at each point on our selected path. The values in the graphs and tables correspond to units of normalized distance—true distance (in feet) is approximately one-third of the normalized distance.

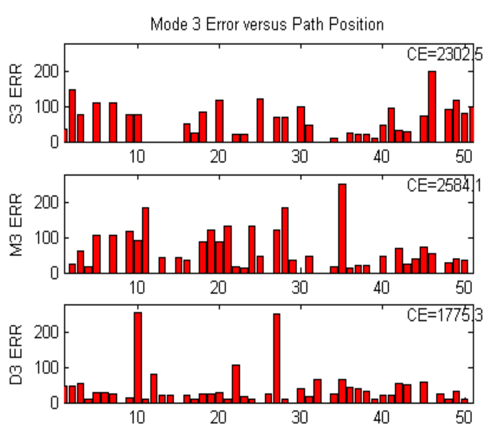

Fig 2. Single Sample Error (Distance) as a function of position (51 points / path)

	Map Density		
	`sparse`	`moderate`	`dense`
Average	45.1	50.7	34.8
Median	25.0	35.1	22.8
Std. Dev.	49.1	57.2	50.2
Cum. Sum.	2302	2584	1775
Rel.Cum.Err.	14%	16%	11%

Table 1. Single Sample Performance (Distance)

	Map Density		
	`sparse`	`moderate`	`dense`
Closest Room	86%	96%	94%
Closest Point	69%	49%	29%
Miss	6%	49%	71%

Table 2. Single Sample Performance (Indoor-specific)

As is evident from the data, CS/RADAR's single-sample performance suffers from both glitching effects (`dense` map, position 10 for instance) and noise-like errors. Even these seemingly high cumulative errors (2584 NDU ≈ 860 feet!) are small compared to the total distance traveled in the course of the entire path length (just over 16,000 NDU). The relative cumulative error in Table 1 shows the cumulative distance error divided by the entire path length.

Considering only distance as a metric, there is little performance difference between the runs using the `sparse` and `moderate` maps. The average and cumulative errors are comparable for both densities—only `dense` significantly outperforms the other maps, if only by a modest 30%.

When considering the other metrics, the difference between the mapping strategies becomes more evident. All three of the maps generated produced good results at determining room-level granularity, but the number of closest-point samples declines sharply as sample density increases. Simultaneously, the rate of "misses" increases dramatically.

7.3 3-Sample Majority Performance

For the `moderate` map, we repeated our experiment allowing three samples per location (or more, if needed for majority voting) to determine the indicated position. We hoped that this modification to the sampling scheme would reduce the effects of RF-variation and eliminate the “glitches” as seen in the single-sample experiments.

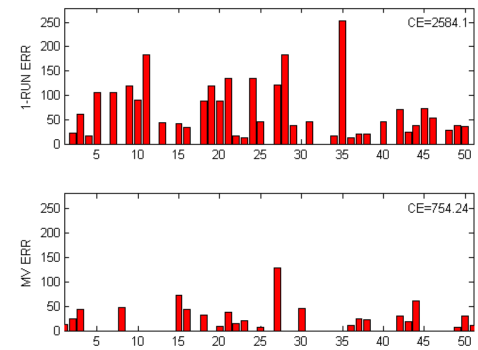

Fig. 3. `Moderate` sample error for single-sample (top) and 3-sample majority vote (bottom)

	Sampling Scheme	
	Single-sample	3-sample MV
Average	50.7	14.9
Median	35.1	1.0
Std. Dev.	57.2	24.2
Cum. Sum.	2584	760
Rel.Cum.Err.	16%	4.7%

Table 3. 3-sample performance (distance)

	Sampling Scheme	
	Single-sample	3-sample MV
Closest Room	96%	94%
Closest Point	49%	75%
Miss	49%	6%

Table 4. 3-sample performance (Indoor-specific)

Figure 3 and Tables 4 and 5 reflect the effect of using a multi-sample majority-vote scheme to determine position. Total distance error across the entire path length is reduced fourfold, and the median error is very small; many predictions are accurate to a few feet. The average distance error is reduced 70%, and the distribution of error is clustered more tightly about the median.

Considering the indoor-specific criteria, the majority-vote scheme combines the miss rate and closest-point rate of the `sparse` map with the room-level accuracy of the `moderate` and `dense` maps.

7 Improvements

There are a few improvements we could make to CS/RADAR that became clear during the testing phases of our design. One of the limiting factors of our implementation is the wireless hardware we are using. 802.11b wireless networking cards vary in quality. It is quite common for the signal strength measured in one card to be different than what is measured from another card, even of the same brand and model. It is also common for cards to have different ranges so they may detect different wireless access points at the same location. Detecting position is dependent on this in our algorithms, so in order to make the system more generalized, a calibration function would help greatly. If the software could be calibrated to take into account the characteristics of different wireless cards an RF-map generated by very sensitive card would be useful for a whole range of wireless hardware.

Another useful extension of CS/RADAR would be adding the notion of intermittent base stations. We often ran into problems when we were at the edge of a base station’s range: At times the node could here the base station while other times it could not. If the base station was detected when mapping the area but not during location discovery, or vice versa, CS/RADAR would generally predict a position very far from our actual location. If a method could be devised of marking base stations at a given position as intermittent, we could penalize less for sampling differently than the map at that position, and would therefore be more likely to predict the correct position.

Adding a reverse lookup of a position would be another useful feature. Finding RF-coverage of an area would be simple if we were to add a function that would give the RF-characteristics of any point that was asked of it. With the time already taken to measure RF-characteristics all over the target area, it would not add any overhead to query the map to check that sufficient base station coverage exists.

A downfall of this system is generating the RF-map. Currently, this is done by manually sampling every point that is needed in the map. This is quite time consuming and repetitive. Deriving a way to generate these maps automatically would greatly increase the usefulness of CS/RADAR. There is a large cost associated with generating the map initially and regenerating it if any of the RF-characteristics of the area change. Automating this process would allow CS/RADAR to be much more useful in dynamic environments.

8 Conclusions

RADAR works well for location discovery under existing static 802.11 wireless networks. Indoors, it provides room-level location discovery granularity and does not require any extra hardware on top of the existing wireless infrastructure. However, RADAR is subject to RF noise in the area as well as a glitching effect when at the edge of signal range. We have shown using a majority-voting scheme can reduce these effects though. It also performs poorly in environments where the base stations may change location or transmitting characteristics.

The concept of RADAR can be extended to other sensible medium besides RF as well. The only requirements being that the characteristics of the medium vary with position but remain static over time. The characteristics need not vary predictably with position, but predictability would allow automatic generation of condition maps. It is also possible that the characteristics of the medium vary with time, so long as they do it in a predictable manner, or can be remapped easily and quickly.

References

Bahl, P., Balachandran, A., and Padmanabhan, V. "A Software System for Locating Mobile Users: Design, Evaluation, and Lessons." http://www.ece.wisc.edu/~megerian/ece902/radar.pdf

Bahl, P., and Padmanabhan, V. "RADAR: An In-Building RF-based User Location and Tracking System." In Proc. IEEE INFOCOM (Tel-Aviv, Israel, Mar. 2000)

Balakrishnan, H., Padmanabhan, V., Seshan, S., Stemm, M., Amir, E., and Katz, R., "TCP Improvements for Heterogeneous Networks: The Daedalus Approach." http://citeseer.ist.psu.edu/balakrishnan97tcp.html

Christ, T., and Godwin, P., "A Prison Guard Duress Alarm Location System." IEEE 1993, CH3372-0/93 http://localization.millennium.berkeley.edu/papers/radio/A/prison/Guard/Duress/Alarm/Location/System.pdf

Doherty, L., Pister, K., and Ghaoui, L. "Convex position estimation in wireless sensor networks." In Proc. IEEE INFOCOM (April 2001)

Kim, H., Abdelzaher, T., Kwon, W. "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks." SenSys'03, November 5–7, 2003

Schilit, B., and Theimer, M. "Disseminationg Active Map Information to Mobile Hosts." IEEE Network (Sep/Oct 1994), 22-32

Want, R., Hopper, A., Falcao, V. and Gibbons, J. "The Active Badge Location System." ACM Transactions on Information Systems 10, 1 (January 1992), 91-102

Werb, J., and Lanzl, C. "Designing a positioning system for finding things and people indoors." IEEE Spectrum 35, 9 (September 1998), 71-78.