Positioning in Indoor Environments using WLAN Received Signal Strength Fingerprints

Christos Laoudias

Department of Electrical and Computer Engineering
KIOS Research Center for Intelligent Systems and Networks
University of Cyprus
Nicosia, Cyprus

9 February 2011

Outline

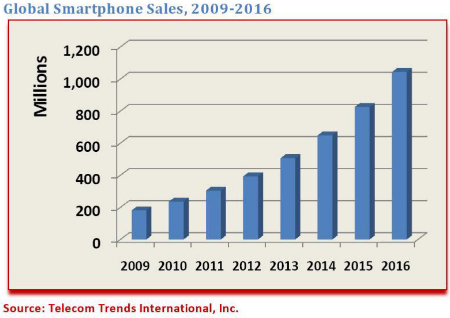
Outline

Why Indoor Positioning?

- People spend most of their time indoors, e.g. shopping malls, libraries, airports, university campuses
- Massive availability of mobile devices with wireless connectivity
- Satellite-based geolocation, e.g. GPS, is infeasible indoors
- Interest in indoor location-aware applications, e.g. in-building guidance, asset tracking, event detection

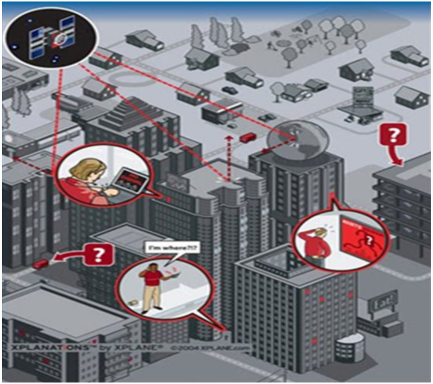
Why Indoor Positioning?

- People spend most of their time indoors, e.g. shopping malls, libraries, airports, university campuses
- Massive availability of mobile devices with wireless connectivity
- Satellite-based geolocation, e.g. GPS, is infeasible indoors
- Interest in indoor location-aware applications, e.g. in-building guidance, asset tracking, event detection



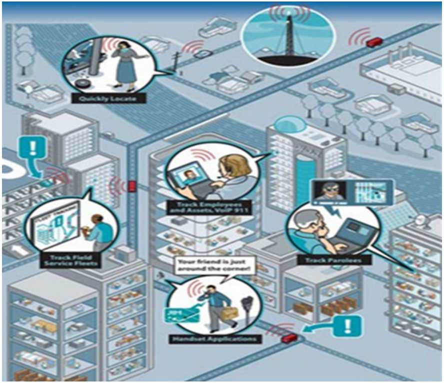
Why Indoor Positioning?

- People spend most of their time indoors, e.g. shopping malls, libraries, airports, university campuses
- Massive availability of mobile devices with wireless connectivity
- Satellite-based geolocation, e.g. GPS, is infeasible indoors
- Interest in indoor location-aware applications, e.g. in-building guidance, asset tracking, event detection



Why Indoor Positioning?

- People spend most of their time indoors, e.g. shopping malls, libraries, airports, university campuses
- Massive availability of mobile devices with wireless connectivity
- Satellite-based geolocation, e.g. GPS, is infeasible indoors
- Interest in indoor location-aware applications, e.g. in-building guidance, asset tracking, event detection



Indoor Applications

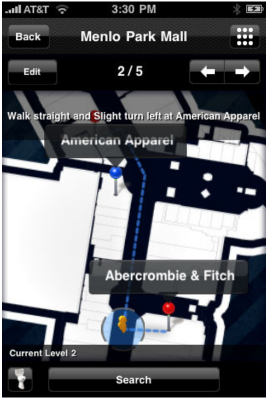


Figure: FastMall

Figure: Aisle411

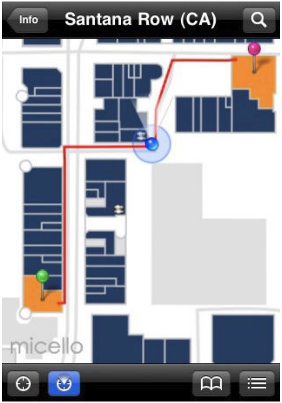


Figure: Micello

Indoor Applications

Figure: Point Inside (a mall)

Figure: Point Inside (an airport)

Indoor Applications

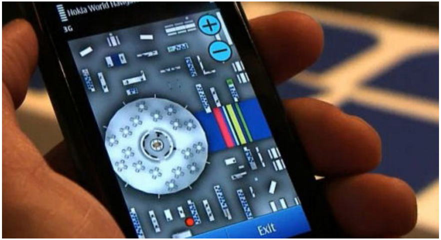


Figure: Nokia World Indoor Navigator

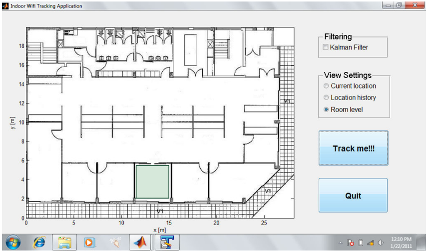
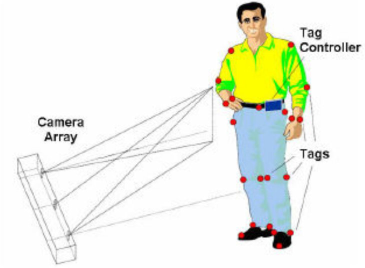


Figure: Indoor WiFi Tracker

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

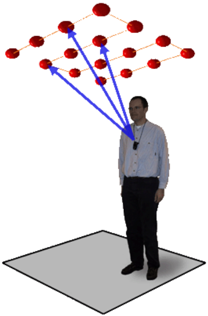


Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

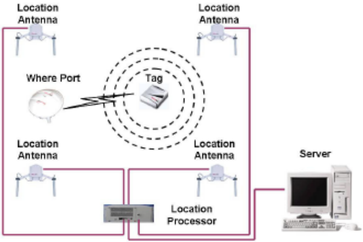


Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)



Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

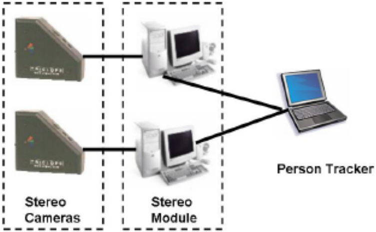
- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

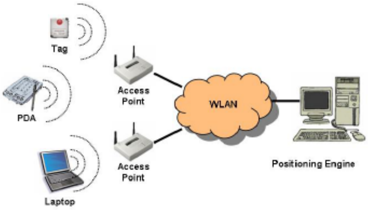


Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

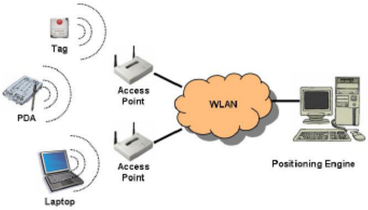


Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Technologies for Indoor Positioning

- IR (e.g. Firefly)
- Ultrasound (e.g. Active Bat, Cricket)
- RFID (e.g. WhereNet)
- UWB (e.g. Ubisense)
- Cameras (e.g. Easy Living)
- WLAN (e.g. Ekahau)

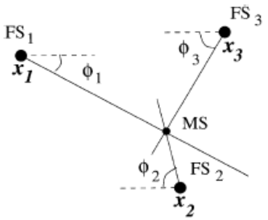


Why WLAN technology?

- Ubiquitous deployment of WLAN infrastructure (APs)
- Most mobile devices are equipped with WLAN adapters

Measurements and Algorithms

- Angle of Arrival (AOA)
- Time of Arrival (TOA)
  - $\tau_i = \frac{d_i}{c}$
- Time Difference of Arrival (TDOA)
  - $\rho_{i,j} = \frac{d_i - d_j}{c}$
- Received Signal Strength (RSS)
  - $rss_i = K - 10n \log d_i$ [dBm]

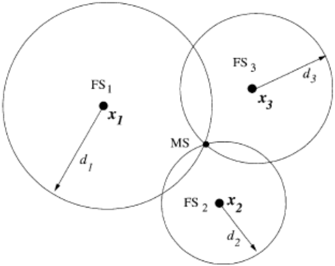


Why RSS measurements?

- AOA/TOA/TDOA measurements require additional hardware
- RSS values are constantly monitored and easily collected

Measurements and Algorithms

- Angle of Arrival (AOA)
- Time of Arrival (TOA)
  - $\tau_i = \frac{d_i}{c}$
- Time Difference of Arrival (TDOA)
  - $\rho_{i,j} = \frac{d_i - d_j}{c}$
- Received Signal Strength (RSS)
  - $rss_i = K - 10n \log d_i$ [dBm]

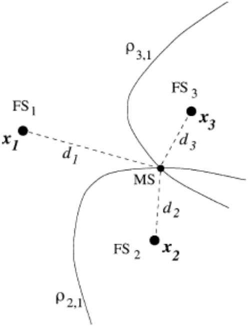


Why RSS measurements?

- AOA/TOA/TDOA measurements require additional hardware
- RSS values are constantly monitored and easily collected

Measurements and Algorithms

- Angle of Arrival (AOA)
- Time of Arrival (TOA)
  - $\tau_i = \frac{d_i}{c}$
- Time Difference of Arrival (TDOA)
  - $\rho_{i,j} = \frac{d_i - d_j}{c}$
- Received Signal Strength (RSS)
  - $rss_i = K - 10n \log d_i$ [dBm]

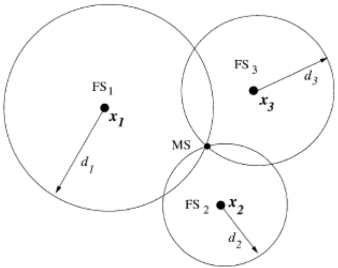


Why RSS measurements?

- AOA/TOA/TDOA measurements require additional hardware
- RSS values are constantly monitored and easily collected

Measurements and Algorithms

- Angle of Arrival (AOA)
- Time of Arrival (TOA)
  - $\tau_i = \frac{d_i}{c}$
- Time Difference of Arrival (TDOA)
  - $\rho_{i,j} = \frac{d_i - d_j}{c}$
- Received Signal Strength (RSS)
  - $rss_i = K - 10n \log d_i$ [dBm]

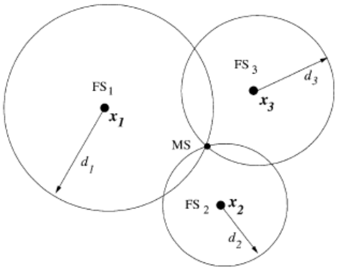


Why RSS measurements?

- AOA/TOA/TDOA measurements require additional hardware
- RSS values are constantly monitored and easily collected

Measurements and Algorithms

- Angle of Arrival (AOA)
- Time of Arrival (TOA)
  - $\tau_i = \frac{d_i}{c}$
- Time Difference of Arrival (TDOA)
  - $\rho_{i,j} = \frac{d_i - d_j}{c}$
- Received Signal Strength (RSS)
  - $rss_i = K - 10n \log d_i$ [dBm]



Why RSS measurements?

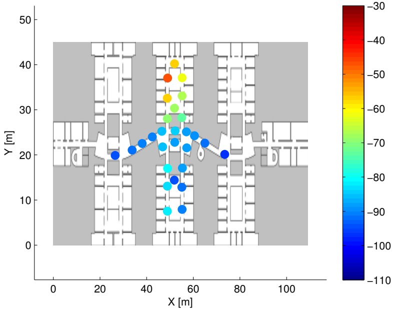
- AOA/TOA/TDOA measurements require additional hardware
- RSS values are constantly monitored and easily collected

Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations

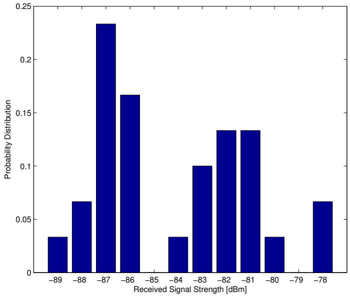
Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations



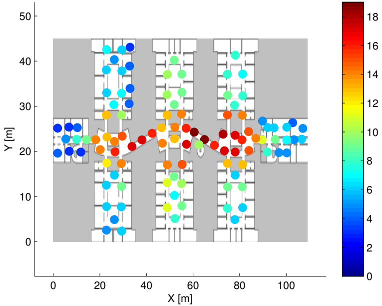
Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations



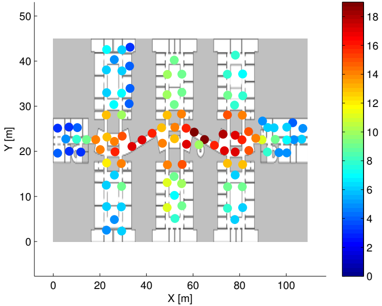
Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations



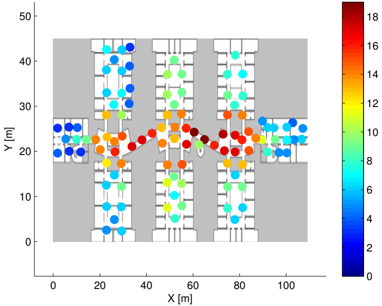
Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations

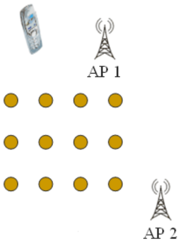


Attenuation Models vs Fingerprints

- Attenuation models are insufficient indoors
  - Complex propagation conditions (multipath, shadowing) due to walls and ceilings
  - RSS value fluctuates over time at a given location
  - Variable # of detected APs
  - Unpredictable factors (people moving, doors, humidity)
- Fingerprints
  - Capture the RSS-location dependency
  - More robust to signal variations

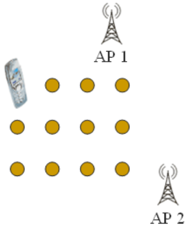


Fingerprint-based Positioning



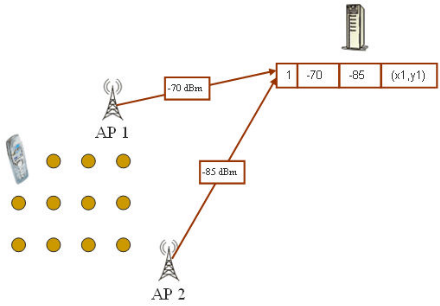
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t)$, $t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



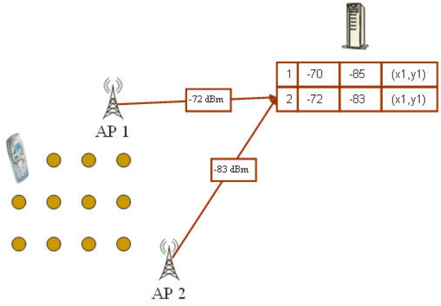
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t), \; t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints $r^k, \; k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



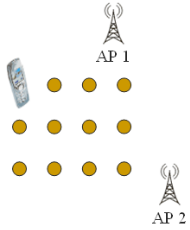
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t), \ t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints $r^k, \ k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \dots, r_{in}]^T$
  - Series $r_i(t), \ t = 1, \dots, T$
  - Training set contains $N = l \cdot T$ fingerprints $r^k, \ k = 1, \dots, N$
  - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \dots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

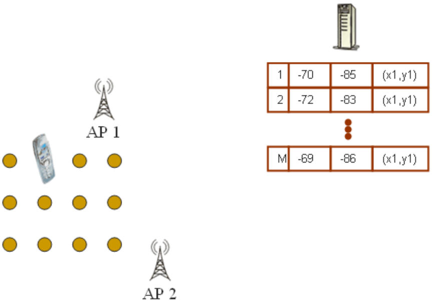
Fingerprint-based Positioning



1	-70	-85	(x1,y1)
2	-72	-83	(x1,y1)
⋮			
M	-69	-86	(x1,y1)

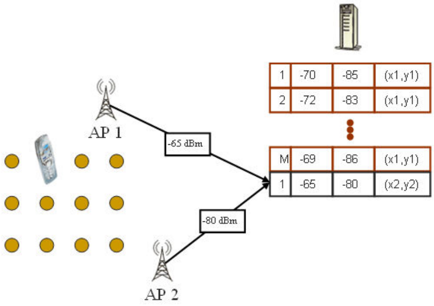
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \dots, r_{in}]^T$
  - Series $r_i(t), \;\; t = 1, \dots, T$
  - Training set contains $N = l \cdot T$ fingerprints $r^k, \;\; k = 1, \dots, N$
  - Averaging $\overline{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \dots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



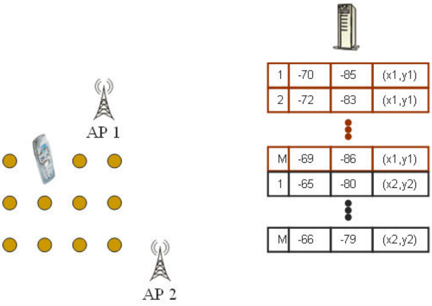
- **Offline phase:** Build RSS radio map
    - n APs deployed in the area
    - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
    - Series $r_i(t)$, $t = 1, \ldots, T$
    - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
    - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
    - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
    - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



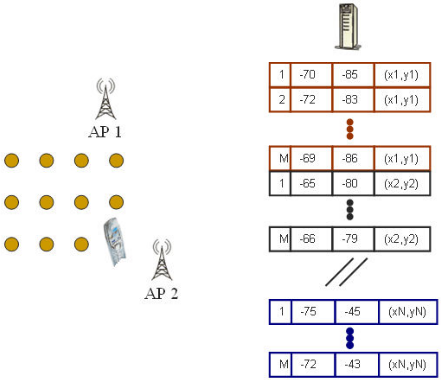
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t)$, $t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



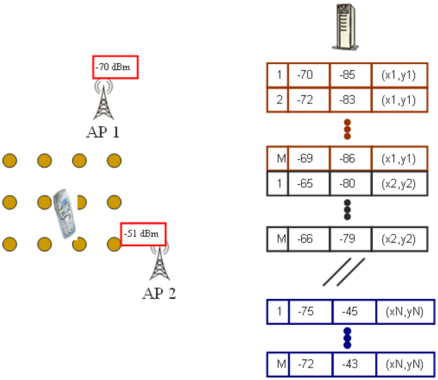
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t), \;\; t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints $r^k, \;\; k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



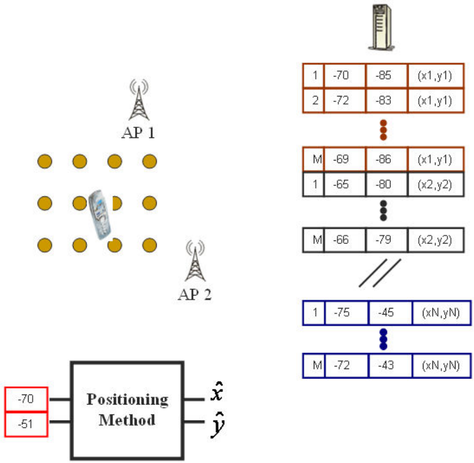
- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t)$, $t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t)$, $t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Fingerprint-based Positioning



- **Offline phase:** Build RSS radio map
  - n APs deployed in the area
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Series $r_i(t)$, $t = 1, \ldots, T$
  - Training set contains $N = l \cdot T$ fingerprints r^k, $k = 1, \ldots, N$
  - Averaging $\overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_i(t)$
- **Online phase:** Positioning
  - Fingerprint $s = [s_1, \ldots, s_n]^T$ is observed
  - Obtain an estimate $\widehat{\ell}$ using the radio map

Deterministic Approach

Deterministic positioning methods

Location is estimated as a convex combination of the reference locations ℓ_i by using the K locations with the shortest distances between $\overline{r}_i$ and s.

$$\widehat{\ell} = \sum_{i=1}^{K} \frac{w_i}{\sum_{j=1}^{K} w_j} \ell'_i \quad (1)$$

where $\{\ell'_1, \ldots, \ell'_l\}$ denotes the ordering of reference locations with respect to increasing distance $\|\overline{r}_i - s\|$.

K-Nearest Neighbor (KNN) variants

- NN: $K = 1$
- KNN: $K \neq 1$, $w_i = \frac{1}{K}$
- Weighted KNN: $K \neq 1$, $w_i = \frac{1}{\|\overline{r}_i - s\|}$

Probabilistic Approach

Probabilistic positioning methods

Location ℓ is treated as a random vector that can be estimated by calculating the conditional probabilities $p(\ell_i|s)$ (*posterior*) given s.

$$p(\ell_i|s) = \frac{p(s|\ell_i)p(\ell_i)}{p(s)} = \frac{p(s|\ell_i)p(\ell_i)}{\sum_{i=1}^{l} p(s|\ell_i)p(\ell_i)} \quad (2)$$

$$p(s|\ell_i) = \prod_{j=1}^{n} p(s_j|\ell_i) \quad (3)$$

where $p(s|\ell_i)$ is the *likelihood*, $p(\ell_i)$ is the *prior* and $p(s)$ is a constant.

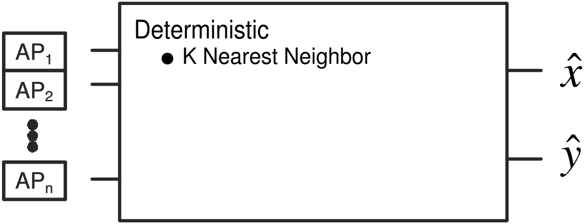
Positioning variants

- Maximum Likelihood (ML): $\widehat{\ell} = \arg\max_{\ell_i} p(s|\ell_i)$
- Maximum A Posteriori (MAP): $\widehat{\ell} = \arg\max_{\ell_i} p(s|\ell_i)p(\ell_i)$
- Minimum Mean Square Error (MMSE): $\widehat{\ell} = \mathbf{E}[\ell|s] = \sum_{i=1}^{l} \ell_i p(\ell_i|s)$

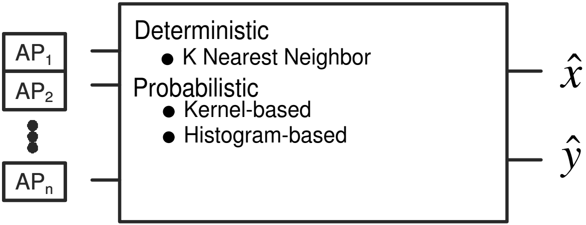
Outline

RBF-based Positioning Method

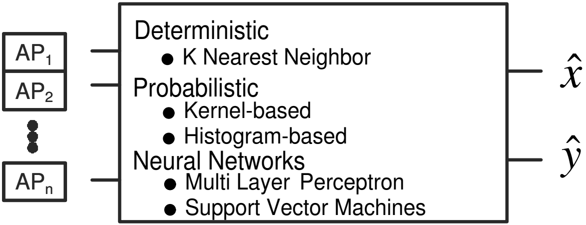
RBF-based Positioning Method



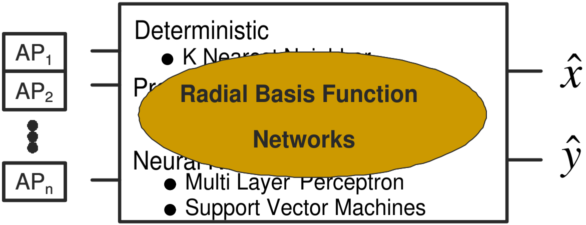
RBF-based Positioning Method



RBF-based Positioning Method



RBF-based Positioning Method



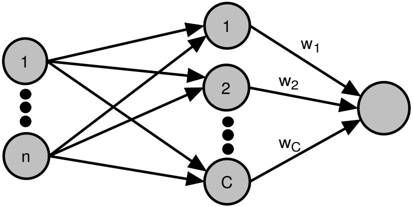
RBF-based Positioning Method I

Data Regression

$$\ell(s) = \sum_{i=1}^{C} w_i u(s, c_i)$$

where $u(s, c_i) = \frac{\varphi(\|s - c_i\|)}{\sum_{j=1}^{C} \varphi(\|s - c_j\|)}$

- C: number of centers
- c_i: n-dimensional center
- $\varphi(\|s - c\|) = \exp\left(-\frac{1}{2}\|s - c\|^2\right)$
- w_i: 2-dimensional weights



RBF-based Positioning Method II

Training (offline)

System of linear equations using the $N = I \cdot T$ reference fingerprints

$$\ell_i = \sum_{j=1}^{C} w_j u\big(r_i(t), c_j\big), \;\; i = 1, \ldots, I, \;\; t = 1, \ldots, T \tag{4}$$

Matrix form $\mathbf{U}\mathbf{w} = \mathbf{d}$

- $\mathbf{U} \in R^{N \times C}$: each row contains the responses to a particular fingerprint
- $\mathbf{w} \in R^{C \times 2}$: unknown weights
- $\mathbf{d} \in R^{N \times 2}$: outputs that represent the location coordinates

The weights can be easily determined through linear algebra.

RBF-based Positioning Method III

Positioning (online)

$$\widehat{\ell}(s) = \sum_{j=1}^{C} w_j u(s, c_j) \tag{5}$$

Center Selection

standard RBF (sRBF)

- $C = N$, i.e. $c_i = r^i, \ i = 1, \dots, N$
- $w = U^{-1}d$
  - High memory requirements
  - Computational complexity (weight calculation and positioning)
  - Prone to overfitting

clustered RBF (cRBF)

- $C = I$, i.e. $c_i = \overline{r}_i, \ i = 1, \dots, I$
- $w = U^{+}d, \ U^{+} = (U^T U)^{-1} U^T$
  - Better than selecting $C < N$ centers randomly or experimentally or by using a center selection algorithm (e.g. OLS)
  - Computationally efficient due to the compact size
  - Better generalization

Distance Calculation

Set of basis functions

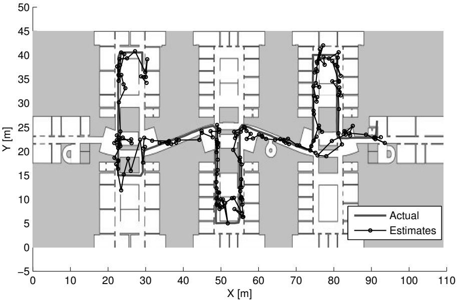
$$\varphi(\|s - c_j\|) = \exp\Big(-\frac{1}{2}(s - c_j)^T \Sigma^{-1} (s - c_j)\Big), \; j = 1, \ldots, C.$$

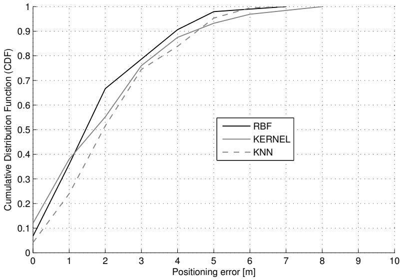
- $\Sigma = \sigma^2 I$, where σ^2 is a common variance (width) for all n APs
  - Select σ^2 experimentally and fine-tune with validation data
  - Use a heuristic so that $\sigma^2 \propto d_{max}$, where $d_{max} = \max \|c_i - c_j\| \text{ for } i, j = 1, \ldots, C$
- $\Sigma = diag(\sigma_1^2, \; \sigma_2^2, \; \ldots, \; \sigma_n^2)$
  - σ_k^2 is the sample variance of the k-th AP
  - Can be used to build an AP selection methodology for dimensionality reduction
- A non-diagonal covariance matrix Σ does not work well in practice, because the RSS values from neighboring APs are independent

Properties of the cRBF Positioning Method

- Reduced network size
  - Unknown weights are fast and easy to compute
  - Low memory requirements for storing few centers and weights
  - Low computational complexity during positioning
- Practicality & Scalability
  - Retraining time for new data is reduced with appropriate matrix operations (e.g. MLP has to be trained from scratch)
  - Network size is decided in a principled manner (e.g. MLP size is selected experimentally)
  - Easily scaled to other setups with different number of APs, reference locations or fingerprints

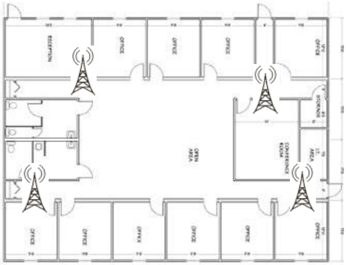
Experimental Results





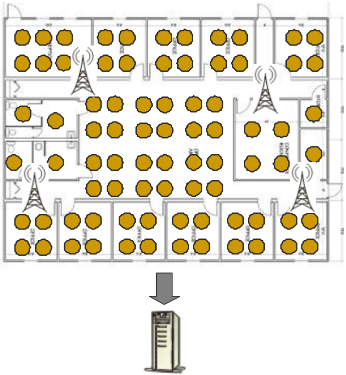
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



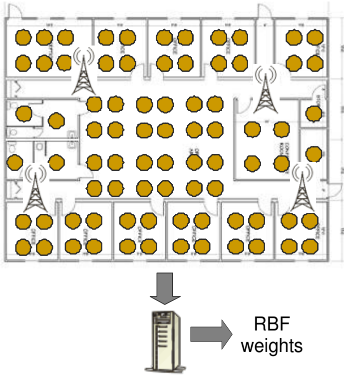
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



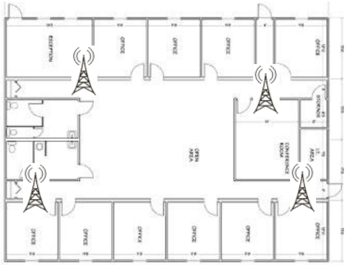
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



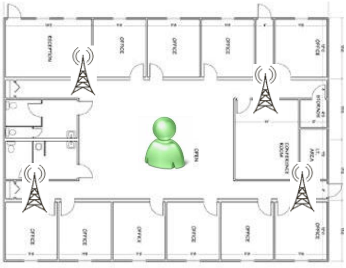
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



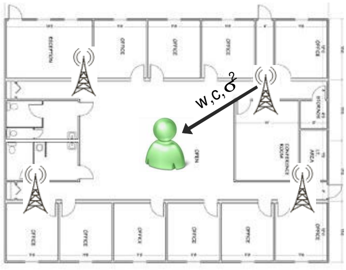
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



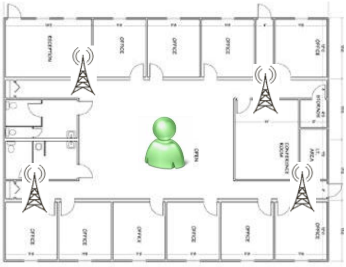
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



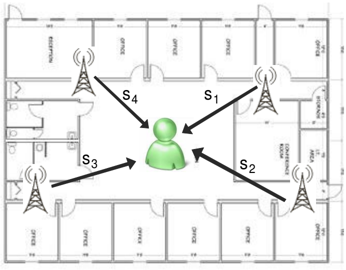
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



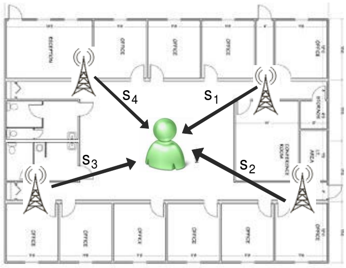
Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



Indoor Positioning System

- **Offline phase**
  1. Collect and store reference fingerprints
  2. Train RBF to determine network weights
- **Online phase**
  1. Transmit a small set of parameters
  2. Use the observed fingerprint to self-locate
- **Properties**
  1. Reduced start-up time
  2. Low communication overhead
  3. Privacy and Security



Outline

SNAP Algorithm

Subtract on Negative Add on Positive (SNAP) algorithm

- Event detection in binary sensor networks
- Low computational complexity and fault tolerance

Objective

- Adapt the SNAP algorithm to the WLAN setup
- Enhance the performance in terms of **fault tolerance** and **accuracy**

Methodology

- Modify the original SNAP algorithm to use WLAN RSS fingerprints
- Examine the **fault tolerance** of SNAP using our fault models
- Improve the **accuracy** by exploiting the RSS levels in the fingerprints

Positioning with Binary Data

SNAP Algorithm

1. *Region of Coverage (RoC)*
   $RoC_j \subseteq L,\ j = 1, \ldots, n$

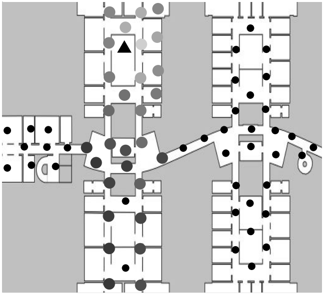
2. *Likelihood Matrix $\mathcal{L}$*

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

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

3. *Location Estimation*

$$\widehat{\ell}(s) = \arg\max_{\ell_i \in L} LV_i$$



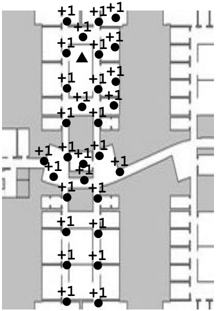
Positioning with Binary Data

SNAP Algorithm

1. *Region of Coverage (RoC)*
   $RoC_j \subseteq L, \; j = 1, \ldots, n$
2. *Likelihood Matrix* $\mathcal{L}$

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

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



3. *Location Estimation*

$$\widehat{\ell}(s) = \arg\max_{\ell_i \in L} LV_i$$

Positioning with Binary Data

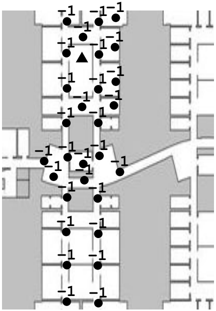
SNAP Algorithm

1. *Region of Coverage (RoC)*
   $RoC_j \subseteq L,\ j = 1, \ldots, n$

2. *Likelihood Matrix* $\mathcal{L}$

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

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



3. *Location Estimation*

$$\widehat{\ell}(s) = \arg\max_{\ell_i \in L} LV_i$$

Positioning with Binary Data

SNAP Algorithm

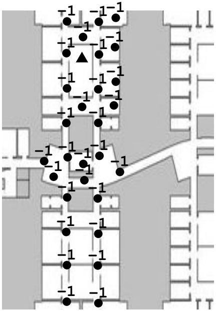
1. *Region of Coverage (RoC)*
   $RoC_j \subseteq L,\ j = 1, \ldots, n$
2. *Likelihood Matrix $\mathcal{L}$*

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

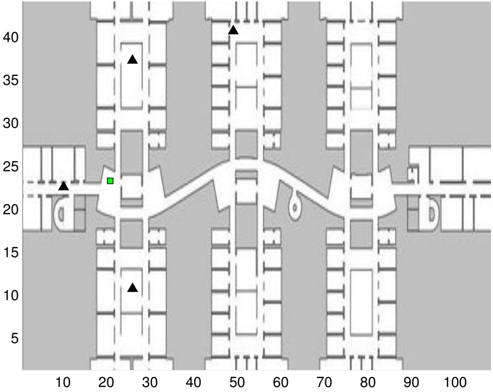
$$LV_i = \sum_{j=1}^{n} \mathcal{L}(i,j)$$

3. *Location Estimation*

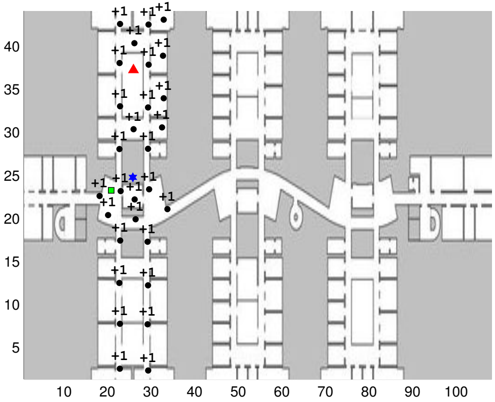
$$\widehat{\ell}(s) = \arg\max_{\ell_i \in L} LV_i$$



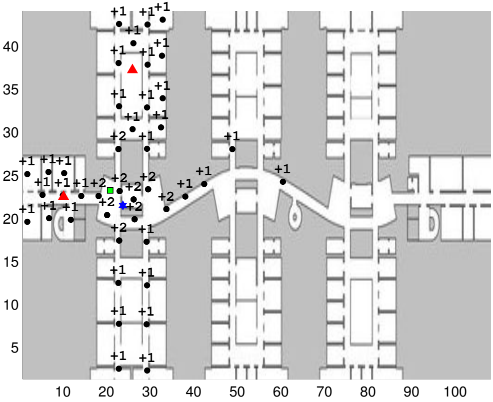
Example application of SNAP



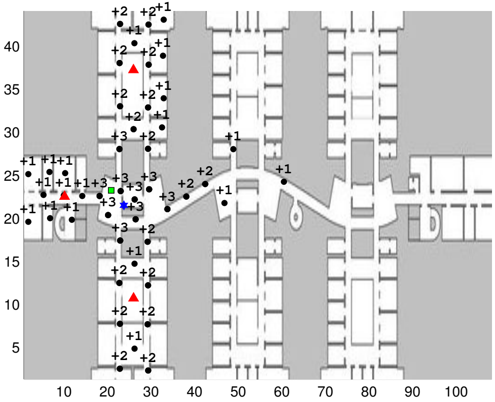
Example application of SNAP



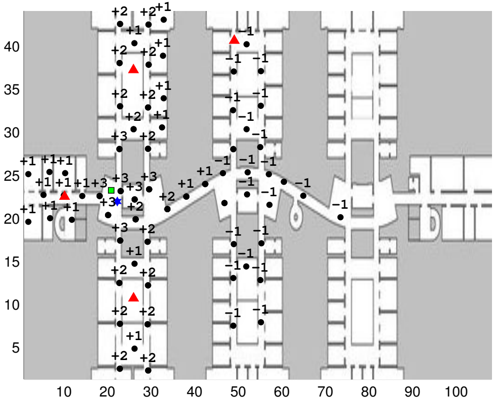
Example application of SNAP



Example application of SNAP



Example application of SNAP



SNAPz: Improving the Accuracy of SNAP I

Idea

If an AP is detected, then the user is more likely to reside in the locations inside the *RoC* that have similar RSS values to the observed RSS value.

Zone of Coverage (ZoC)

$$Z_m = \left[min + (m-1)\frac{max - min}{M},\ min + m\frac{max - min}{M}\right],\ m = 1, \ldots, M$$

- $ZoC_{mj} \subseteq RoC_j$, $m = 1, \ldots, M$ and $j = 1, \ldots, n$
- $\{ZoC_{mj} : \ell_i | \overline{r}_{ij} \in Z_m,\ i = 1, \ldots, l\}$
- $RoC_j = \bigcup_{m=1}^{M} ZoC_{mj}$

SNAPz: Improving the Accuracy of SNAP II

SNAPz algorithm

$$\mathcal{L}(i,j) = \begin{cases} +1, & j \in S \text{ AND } \ell_i \in ZoC_{mj} \\ 0, & j \in S \text{ AND } \ell_i \in ZoC_{(m-1)j} \cup ZoC_{(m+1)j} \\ -1, & j \in S \text{ AND } \ell_i \in RoC_j - \bigcup_{k=m-1}^{m+1} ZoC_{kj} \\ -1, & j \notin S \text{ AND } \ell_i \in RoC_j \\ 0, & \ell_i \notin RoC_j \end{cases}$$

If an AP is detected with certain RSS value, then the user resides

- with high probability in the zone where the reference locations have similar RSS values
- with some probability in the neighboring zones
- with low probability in the remaining zones

Experimental Results I

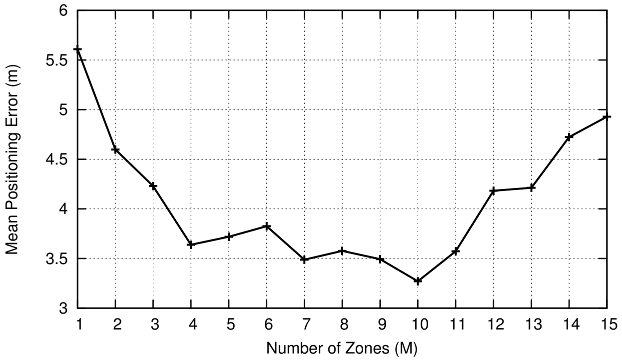


Figure: Performance of SNAPz for varying number of zones.

Experimental Results II

Table: Positioning Error in meters

	Mean	Median	Std	Min	Max
KNN	2.70	2.39	1.61	0.16	8.78
MMSE	2.46	2.18	1.63	0.09	8.99
cRBF	2.38	2.07	1.51	0.08	7.87
SNAPz	3.64	3.37	2.41	0.06	13.21

Table: Computational Complexity

	additions	multiplications	exp	sorts	time (msec)
KNN	$(2n-1)l$	nl	0	l	1.25
MMSE	$(2n+3)l-3$	$(2n+4)l$	nl	0	2.18
cRBF	$(2n+2)l-3$	$(n+3)l$	l	0	1.73
SNAPz	$(n-1)l$	0	0	l	0.49

l: # of reference locations, n: # of APs, sorts: # of floats to be sorted

Outline

Concluding Remarks

- Introduction to indoor positioning and fingerprint methods
- Fingerprint positioning method based on RBF networks
  - High level of accuracy, scalable and applicable in different WLAN setups
  - Positioning system based on the proposed RBF method
- SNAP algorithm with WLAN RSS fingerprints
  - Trade-off between positioning accuracy and computational complexity
  - Investigate the actual power savings on mobile devices

Open Research Issues

Main focus of fingerprint positioning methods so far has been on reducing the positioning error.

Computational Complexity

Time required to estimate location is important, because it affects the battery life of low power mobile devices.

Fault Tolerance

It is desirable to provide smooth performance degradation in the presence of faults, due to unpredicted failures or malicious attacks.

Heterogeneous Devices

Maintain an adequate level of accuracy for various types of devices (different WLAN adapters), without collecting device-specific fingerprints.

Open Research Issues

Main focus of fingerprint positioning methods so far has been on reducing the positioning error.

Computational Complexity

Time required to estimate location is important, because it affects the battery life of low power mobile devices.

Fault Tolerance

It is desirable to provide smooth performance degradation in the presence of faults, due to unpredicted failures or malicious attacks.

Heterogeneous Devices

Maintain an adequate level of accuracy for various types of devices (different WLAN adapters), without collecting device-specific fingerprints.

Open Research Issues

Main focus of fingerprint positioning methods so far has been on reducing the positioning error.

Computational Complexity

Time required to estimate location is important, because it affects the battery life of low power mobile devices.

Fault Tolerance

It is desirable to provide smooth performance degradation in the presence of faults, due to unpredicted failures or malicious attacks.

Heterogeneous Devices

Maintain an adequate level of accuracy for various types of devices (different WLAN adapters), without collecting device-specific fingerprints.

Open Research Issues

Main focus of fingerprint positioning methods so far has been on reducing the positioning error.

Computational Complexity

Time required to estimate location is important, because it affects the battery life of low power mobile devices.

Fault Tolerance

It is desirable to provide smooth performance degradation in the presence of faults, due to unpredicted failures or malicious attacks.

Heterogeneous Devices

Maintain an adequate level of accuracy for various types of devices (different WLAN adapters), without collecting device-specific fingerprints.

References

1. K. Pahlavan, X. Li, and J. Makela, "Indoor geolocation science and technology," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 112–118, 2002.
2. H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.
3. M. Kjærgaard, "A taxonomy for radio location fingerprinting," in *3rd international conference on Location-and context-awareness*. Springer-Verlag, 2007, pp. 139–156.
4. Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.
5. P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *IEEE International Conference on Computer Communications INFOCOM*, vol. 2, 2000, pp. 775–784.
6. T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, Jul. 2002.
7. **C. Laoudias**, M. P. Michaelides, and C. G. Panayiotou, "Fault Tolerant Fingerprint-based Positioning," in *IEEE International Conference on Communications (ICC)*, 2011. [accepted]
8. **C. Laoudias**, M. P. Michaelides, and C. G. Panayiotou, "Fault tolerant positioning using WLAN signal strength fingerprints," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010, pp. 1–8.
9. M. P. Michaelides, **C. Laoudias**, C. G. Panayiotou, "Fault Tolerant Detection and Tracking of Multiple Sources in WSNs using Binary Data," in *48th IEEE Conference on Decision and Control (CDC)*, 2009, pp.3769-3774.
10. **C. Laoudias**, P. Kemppi, C. G. Panayiotou, "Localization using radial basis function networks and signal strength fingerprints in WLAN," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2009, pp. 1–6.
11. **C. Laoudias**, D. Eliades, P. Kemppi, C. Panayiotou, M. Polycarpou, "Indoor localization using neural networks with location fingerprints," in *Artificial Neural Networks – ICANN*. Springer, 2009, pp. 954–963.
12. **C. Laoudias**, C. G. Panayiotou, "Indoor Positioning in WLAN using Radial Basis Function Networks with Received Signal Strength Fingerprints," in *3rd Cyprus Workshop on Signal Processing and Informatics (CWSPI)*, 2010.
13. **C. Laoudias**, C. G. Panayiotou, P. Kemppi, "On the RBF-based positioning using WLAN signal strength fingerprints," in *7th Workshop on Positioning Navigation and Communication (WPNC)*, 2010, pp. 93–98.

Thank you for your attention