Fault Tolerant Fingerprint-based Positioning

C. Laoudias, M. P. Michaelides and C. G. Panayiotou

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

8 June 2011

Outline

Outline

Technologies and Measurements

Why WLAN technology? (instead of Ir, Ultrasound, RFID, etc)

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

Why RSS measurements? (instead of AOA/TOA/TDOA)

- Angle and timing measurements require additional hardware
- RSS values are constantly monitored and easily collected

Why fingerprints? (instead of attenuation model)

- Attenuation models are insufficient indoors
- Fingerprints capture the RSS-location dependency and are more robust to signal variations

Motivation of our work

Main focus of fingerprint positioning methods so far has been on reducing the positioning error which is in the order of 2-10m depending on the

- underlying method (deterministic, probabilistic, neural network, etc)
- experimentation parameters (number of fingerprints collected, resolution of the reference locations, density of the APs)

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.

Outline

SNAP Algorithm

Subtract on Negative Add on Positive (SNAP)[1] algorithm

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

Objectives

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

Positioning with Binary Data

SNAP Algorithm

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

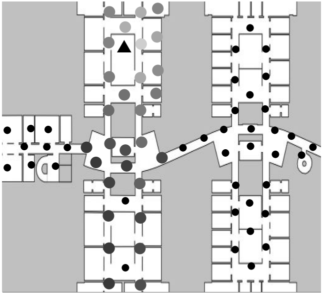
2. *Likelihood Matrix $\mathcal{L}$*

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

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

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)*
   $RoC_j \subseteq L, \ j = 1, \dots, n$

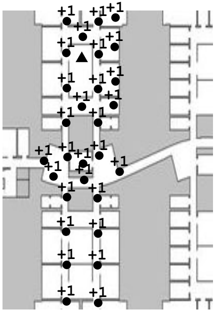
2. *Likelihood Matrix* $\mathcal{L}$

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

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

3. *Location Estimation*

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



Positioning with Binary Data

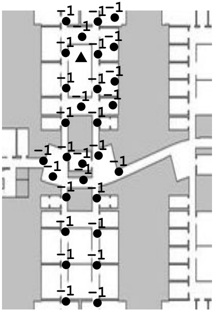
SNAP Algorithm

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

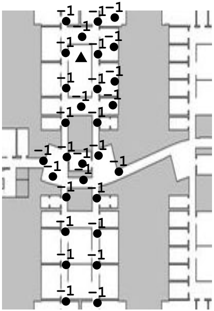
1. *Region of Coverage (RoC)*
   $RoC_j \subseteq L, \; j = 1, \ldots, n$
2. *Likelihood Matrix* $\mathcal{L}$

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

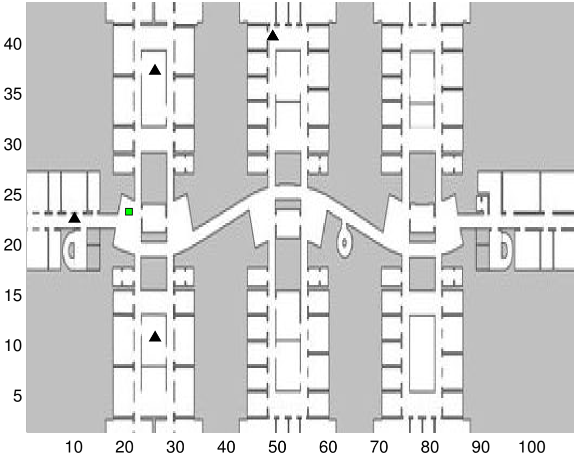
$$LV_i = \sum_{j=1}^{n} \mathcal{L}(i,j)$$

3. *Location Estimation*

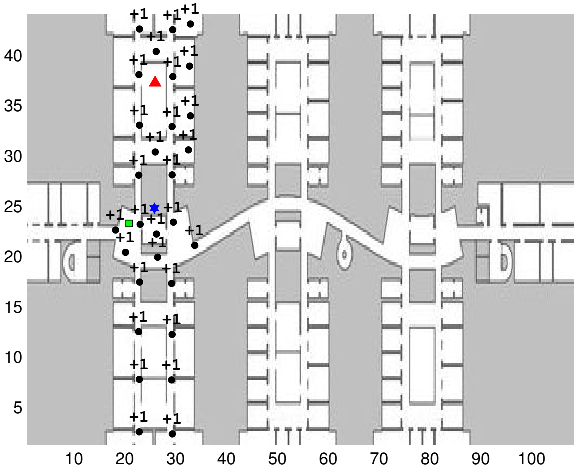
$$\widehat{\ell}(s) = \arg\max_{\ell_i \in L} LV_i$$



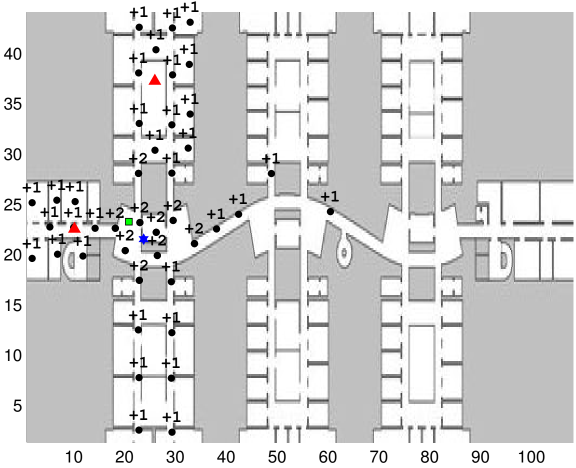
Example application of SNAP



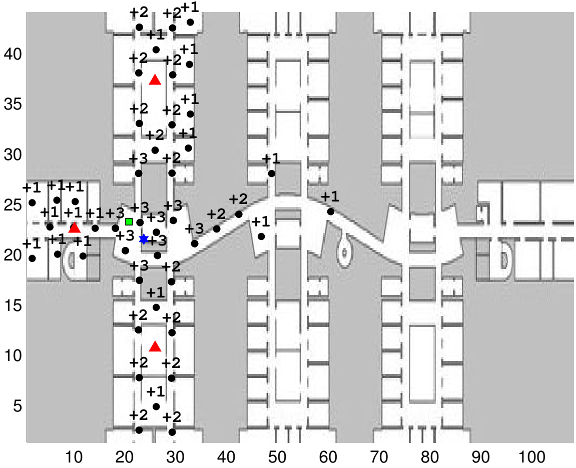
Example application of SNAP



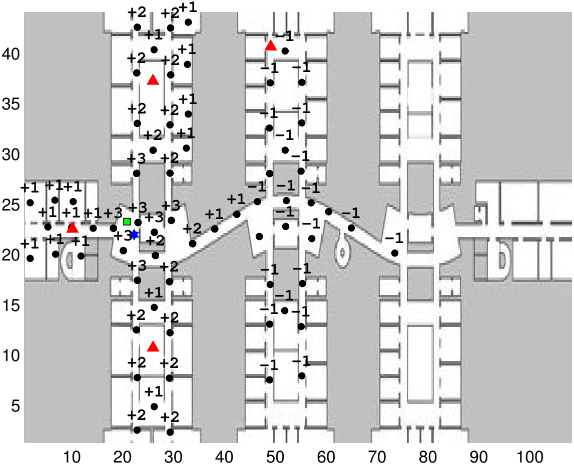
Example application of SNAP



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

SNAPft-z: Improving the Fault Tolerance of SNAPz

AP failures during positioning

A subset of the APs that would otherwise be present in s, are no longer detected and their negative contributions may introduce high errors.

Modified binary SNAP algorithm

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

SNAPft-z algorithm

We incorporate the idea of zones into this modified algorithm to build a fault tolerant SNAP variant.

Outline

Measurement Setup

Experimentation area

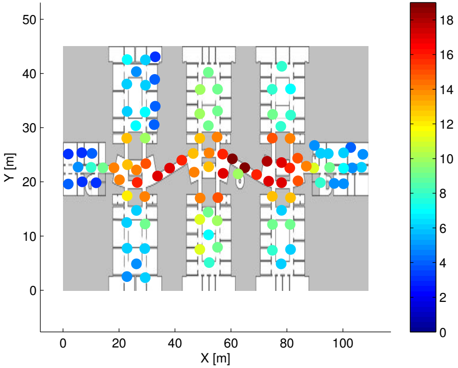
- Area 110x45m on the 2nd floor at VTT Research Center, Finland
- 107 reference locations with 2-3m spacing
- 31 WLAN APs (9.7 APs detected on average)

Training data

- 30 fingerprints per reference location (3210 fingerprints in total

Testing data

- Route of 192 locations sampled 3 times (576 fingerprints in total)



Computational Complexity

Table: Computational Complexity of Positioning Methods

	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
SNAPz	$(n-1)l$	0	0	l	0.49

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

K-Nearest Neighbor (KNN)[2]

- $\widehat{\ell}(s) = \frac{1}{K}\sum_{i=1}^{K} \ell'_i$, $\{\ell'_1, \ldots, \ell'_l\}$ wrt increasing distance $\|\overline{r}_i - s\|$

Minimum Mean Square Error (MMSE)[3]

- $\widehat{\ell}(s) = \sum_{i=1}^{l} \ell_i p(\ell_i|s)$, $p(\ell_i|s) = \frac{p(s|\ell_i)p(\ell_i)}{p(s)}$, $p(s|\ell_i) = \prod_{j=1}^{n} p(s_j|\ell_i)$

Positioning Accuracy

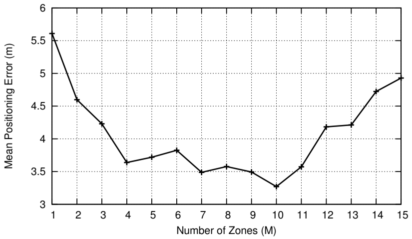


Figure: Accuracy of SNAPz for variable number of zones.

	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
SNAPz	3.64	3.37	2.41	0.06	13.21

Table: Positioning Error in meters

Outline

Fault Models

AP Failure model

Effect

- AP detected in the offline phase is not available during positioning

Feasibility

- Random AP failures or AP shut down temporarily/removed permanently
- Adversary cuts off the power or jams the communication channel

Simulation

- Remove the RSS values of faulty APs in the original test fingerprints

Other Fault Models

False Negative, *False Positive* and *AP Relocation* models[4] that capture the effect of unpredicted failures or malicious attacks.

Fault Tolerance of SNAPz algorithm

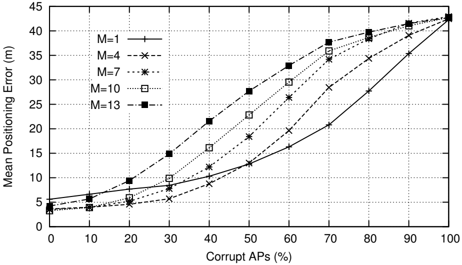
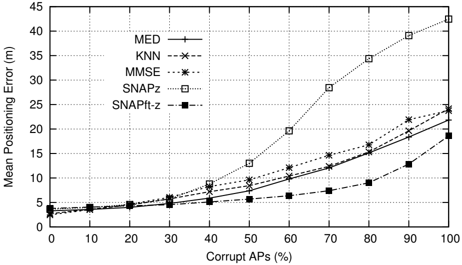


Figure: Variable number of zones

- Under the *AP Failure* model, $M = 4$ for $\leq 50\%$ faulty APs
- $M = 1$ for $> 50\%$ faulty APs
- Using $M > 4$ is not a good option
- $M = 4$ provides a good tradeoff between accuracy and fault tolerance
- Similar behaviour for SNAPz under other fault models
- $M = 4$ is a good option for SNAPft-z as well

Comparison of Positioning Methods



AP Failure model

- Median-based method (MED)[5]
- SNAPz is not resilient to this type of faults
- SNAPft-z exhibits higher fault tolerance
- For 60% faulty APs $\mathcal{E} = 6.38\mathrm{m}$ for SNAPft-z (9.80m, 10.40m, 12.09m and 19.64m for MED, KNN, MMSE and SNAPz)
- Results with other fault models are included in the paper

Outline

Concluding Remarks

- SNAP algorithm with WLAN RSS fingerprints
  - Trade-off between positioning accuracy and computational complexity
  - The proposed SNAPft-z algorithm improves the positioning accuracy of binary SNAP and provides higher resilience to faults
- Future Work
  - Develop a strategy for setting the number of zones M in SNAPft-z algorithm
  - Investigate the actual power savings on Android smartphones

References

1. M. Michaelides and C. Panayiotou, "SNAP: Fault tolerant event location estimation in sensor networks using binary data," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1185–1197, 2009.
2. 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.
3. 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.
4. 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.
5. Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 91–98.

Thank you for your attention

Contact

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