Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Fault Detection and Mitigation in WLAN RSS Fingerprint-based Positioning

Christos Laoudias, Michalis Michaelides and Christos Panayiotou

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

Outline

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Introduction

Nearest Neighbor Algorithm

Hybrid Positioning Algorithm

Experimental Evaluation

Conclusions

Center for Intelligent Systems & Networks

Motivation of our work

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main focus of fingerprint positioning algorithms has been on reducing the positioning error which ranges between 2-10m depending on the

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

Motivation of our work

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main focus of fingerprint positioning algorithms has been on reducing the positioning error which ranges between 2-10m depending on the

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

Fault Tolerance

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

Motivation of our work

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main focus of fingerprint positioning algorithms has been on reducing the positioning error which ranges between 2-10m depending on the

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

Fault Tolerance

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

Assumption

The RSS data collected in the offline phase is not corrupted and we focus on AP failures and non-cryptographic RSS attacks that may occur during positioning.

AP Failure model

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Effect

- APs detected in the offline phase are not available during positioning

Feasibility

- Unpredicted AP failures, e.g. power outage, WLAN system maintenance, AP firmware upgrade etc
- AP shut down temporarily or removed permanently (public WLAN systems)
- Adversary cuts off the power supply or severely jams the communication channel

Simulation

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

Measurement Setup

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

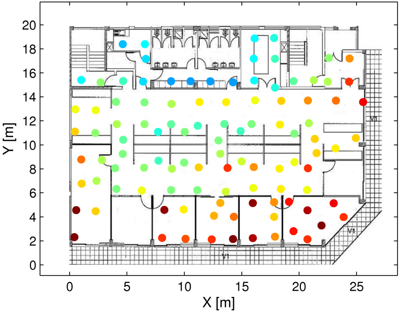
- Area $560\mathrm{m}^2$ at KIOS Research Center, Cyprus
- 73 WLAN APs (9 local, 64 neighboring)
- HP iPAQ hw6915 PDA

Training data

- 105 reference locations, 40 fingerprints per location (4200 in total)

Testing data

- 96 test locations, 20 fingerprints per location (1920 in total)



Nearest Neighbor Algorithm

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Location Estimation

$$\widehat{\ell}(s) = \arg\min_{\ell_i \in L} D_i, \quad D_i = \sqrt{\sum_{j=1}^{n} \left(\bar{r}_{ij} - s_j\right)^2}$$

Fault Detection

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main Idea

- Exploit the distances D_i that are already computed to decide whether fingerprint s is corrupt or not
- The value of a distance-based fault indicator will violate a certain 'fault-free' threshold

Fault Detection

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main Idea

- Exploit the distances D_i that are already computed to decide whether fingerprint s is corrupt or not
- The value of a distance-based fault indicator will violate a certain 'fault-free' threshold

Proposed Fault Indicator

- Sum of distances to the K nearest neighbors $D_{sum}^{(K)}$

Fault Detection

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Main Idea

- Exploit the distances D_i that are already computed to decide whether fingerprint s is corrupt or not
- The value of a distance-based fault indicator will violate a certain 'fault-free' threshold

Proposed Fault Indicator

- Sum of distances to the K nearest neighbors $D_{sum}^{(K)}$

Fault Detection Steps

- Select an appropriate threshold γ based on the distribution of the fault indicator $D_{sum}^{(K)}$ in the fault-free case
- Fault is detected during positioning if $D_{sum}^{(K)} > \gamma$ for the currently observed fingerprint

Fault Detection in practice

Introduction
- Motivation
- Fault Model
- Measurement Setup

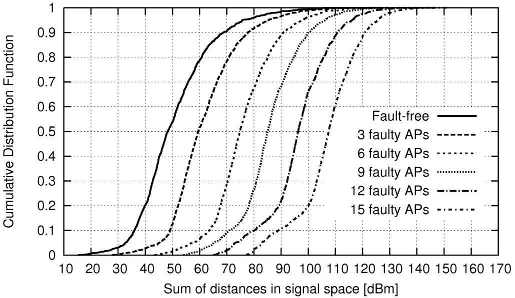
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

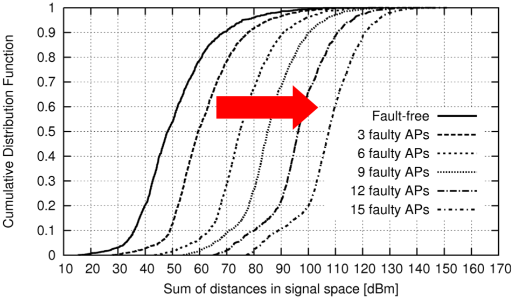
Conclusions
- Concluding Remarks

- As the number of faulty APs is increased the CDF curve of $D_{sum}^{(2)}$ is shifted to the right



Fault Detection in practice

- As the number of faulty APs is increased the CDF curve of $D_{sum}^{(2)}$ is shifted to the right



Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Fault Detection in practice

Introduction
- Motivation
- Fault Model
- Measurement Setup

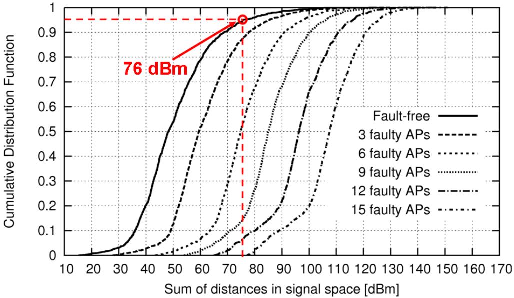
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- As the number of faulty APs is increased the CDF curve of $D_{sum}^{(2)}$ is shifted to the right
- $D_{sum}^{(2)} < 76dBm$ for 95% of time, thus $\gamma = 76dBm$ (5% false detections are acceptable)



Fault Detection in practice

Introduction
- Motivation
- Fault Model
- Measurement Setup

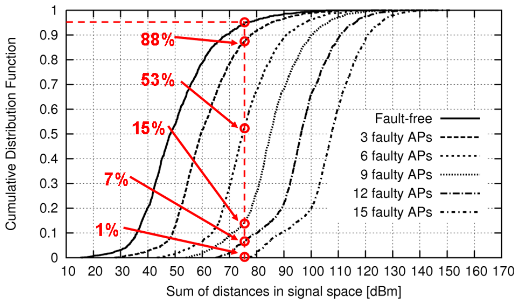
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

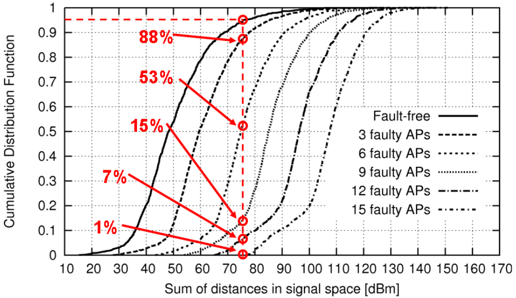
Conclusions
- Concluding Remarks

- As the number of faulty APs is increased the CDF curve of $D_{sum}^{(2)}$ is shifted to the right
- $D_{sum}^{(2)} < 76dBm$ for 95% of time, thus $\gamma = 76dBm$ (5% false detections are acceptable)
- This corresponds to the 88^{th}, 53^{th}, 15^{th}, 7^{th}, 1^{st} percentile as faulty APs increase from 3 to 15



Fault Detection in practice

- As the number of faulty APs is increased the CDF curve of $D_{sum}^{(2)}$ is shifted to the right
- $D_{sum}^{(2)} < 76dBm$ for 95% of time, thus $\gamma = 76dBm$ (5% false detections are acceptable)
- This corresponds to the 88^{th}, 53^{th}, 15^{th}, 7^{th}, 1^{st} percentile as faulty APs increase from 3 to 15
- 12%, 47%, 85%, 93%, 99% correct detections are expected



Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

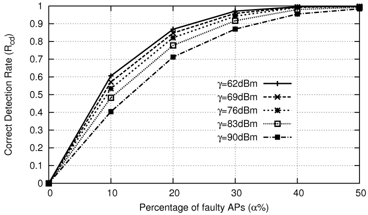
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}



$R_{cd} - R_{fd}$ Trade off

Center for Intelligent Systems & Networks

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

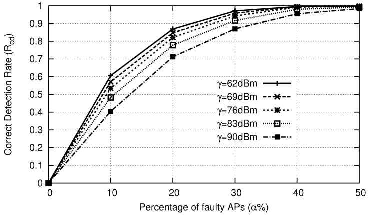
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

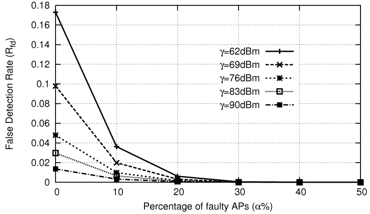
Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}





$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

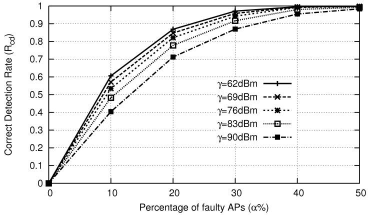
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

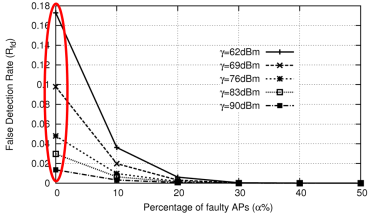
Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$





$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

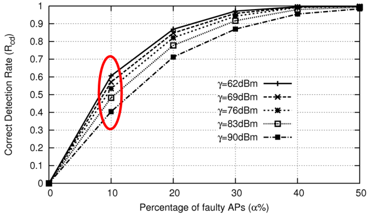
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

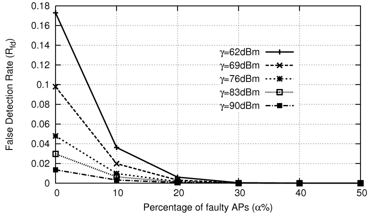
Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%,\ R_{cd} < 0.6\ \forall\gamma$





$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

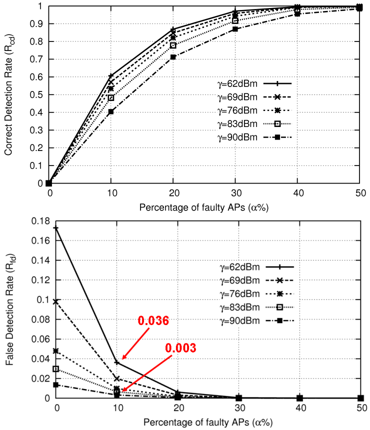
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%,\ R_{cd} < 0.6\ \forall \gamma$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{fd} \downarrow$



$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

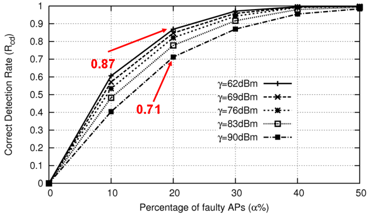
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

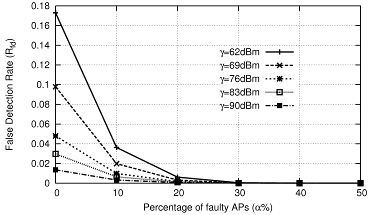
Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%,\ R_{cd} < 0.6\ \forall\gamma$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{fd} \downarrow$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{cd} \downarrow$





$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

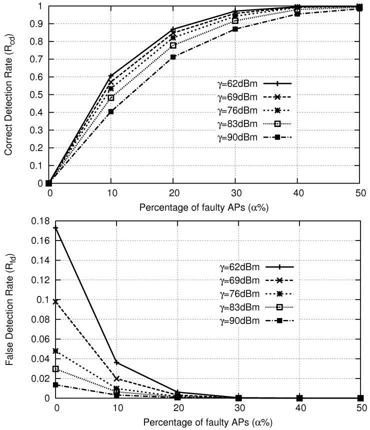
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%, \ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%, \ R_{cd} < 0.6 \ \forall \gamma$
- $\alpha > 0\%, \ \gamma \uparrow \Rightarrow R_{fd} \downarrow$
- $\alpha > 0\%, \ \gamma \uparrow \Rightarrow R_{cd} \downarrow$
- $\gamma = 76dBm$ is a good option



$R_{cd} - R_{fd}$ Trade off

Experimental Evaluation

Introduction
- Motivation
- Fault Model
- Measurement Setup

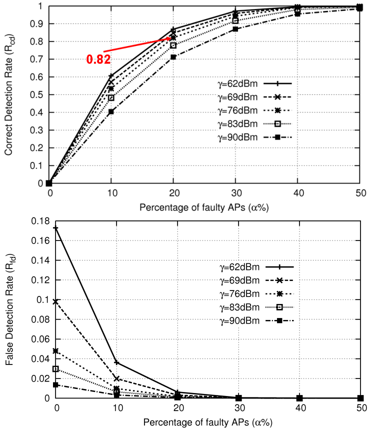
Nearest Neighbor Algorithm
- **Fault Detection**
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%,\ R_{cd} < 0.6\ \forall \gamma$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{fd} \downarrow$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{cd} \downarrow$
- $\gamma = 76dBm$ is a good option
  - High R_{cd} when $\alpha \uparrow$



$R_{cd} - R_{fd}$ Trade off

Introduction
- Motivation
- Fault Model
- Measurement Setup

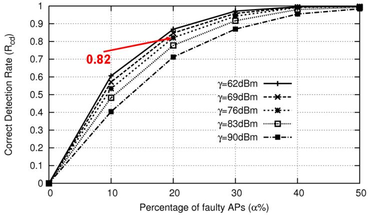
Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

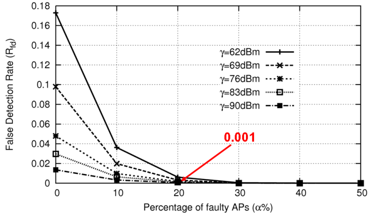
Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

- Correct Detections Rate R_{cd}
- False Detections Rate R_{fd}
- $\alpha = 0\%,\ \gamma \downarrow \Rightarrow R_{fd} \uparrow$
- $\alpha \leq 10\%,\ R_{cd} < 0.6\ \forall\gamma$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{fd} \downarrow$
- $\alpha > 0\%,\ \gamma \uparrow \Rightarrow R_{cd} \downarrow$
- $\gamma = 76dBm$ is a good option
  - High R_{cd} when $\alpha \uparrow$
  - Low R_{fd} when $\alpha \downarrow$





$R_{cd} - R_{fd}$ Trade off

Fault Tolerance

$$\widehat{\ell}(s) = \arg\min_{\ell_i \in L} D_i, \quad D_i = \sqrt{\sum_{j=1}^{n} \left(\overline{r}_{ij} - s_j\right)^2} \qquad (1)$$

Distance Metric

$$D_i = \sqrt{\sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in R_i \setminus S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij}}, \quad d_{ij} = \left(\overline{r}_{ij} - s_j\right)^2 \qquad (2)$$

R_i and S are the subsets of APs that are present in $\overline{r}_i$ and s.

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Fault Tolerance

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

$$\widehat{\ell}(s) = \arg\min_{\ell_i \in L} D_i, \quad D_i = \sqrt{\sum_{j=1}^{n} \left(\bar{r}_{ij} - s_j\right)^2} \qquad (1)$$

Distance Metric

$$D_i = \sqrt{\sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in R_i \setminus S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij}}, \quad d_{ij} = \left(\bar{r}_{ij} - s_j\right)^2 \qquad (2)$$

R_i and S are the subsets of APs that are present in $\bar{r}_i$ and s.

- Effective in the fault-free case because all APs not found in common between $\bar{r}_i$ and s are penalized

Fault Tolerance

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

$$\widehat{\ell}(s) = \arg\min_{\ell_i \in L} D_i, \quad D_i = \sqrt{\sum_{j=1}^{n} \left(\overline{r}_{ij} - s_j\right)^2} \tag{1}$$

Distance Metric

$$D_i = \sqrt{\sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in R_i \setminus S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij}}, \quad d_{ij} = \left(\overline{r}_{ij} - s_j\right)^2 \tag{2}$$

R_i and S are the subsets of APs that are present in $\overline{r}_i$ and s.

- Effective in the fault-free case because all APs not found in common between $\overline{r}_i$ and s are penalized
- What happens in case of faults?

Fault Tolerance

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

$$\widehat{\ell}(s) = \arg\min_{\ell_i \in L} D_i, \quad D_i = \sqrt{\sum_{j=1}^{n} \left(\overline{r}_{ij} - s_j\right)^2} \qquad (1)$$

Distance Metric

$$D_i = \sqrt{\sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in R_i \setminus S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij}}, \quad d_{ij} = \left(\overline{r}_{ij} - s_j\right)^2 \qquad (2)$$

R_i and S are the subsets of APs that are present in $\overline{r}_i$ and s.

- Effective in the fault-free case because all APs not found in common between $\overline{r}_i$ and s are penalized
- What happens in case of faults?

Modified Distance Metric

$$D_i' = \sqrt{\sum_{j \in R_i \cap S} d_{ij} + \sum_{j \in S \setminus R_i} d_{ij}} \qquad (3)$$

Hybrid Positioning Algorithm

General Idea

- Incorporate our fault detection mechanism
- Employ the Modified Distance Metric if faults are present

The Hybrid Positioning Algorithm

1. **RSS Distance Calculation:** Use (2) to calculate the RSS distances D_i between the currently observed fingerprint and all the fingerprints in the radio map.
2. **Fault Indicator Computation:** Compute the fault indicator $D_{sum}^{(K)}$ using the distances D_i from the K Nearest Neighbors.
3. **Location Estimation:** If the condition $D_{sum}^{(K)} > \gamma$ is satisfied, then calculate the respective RSS distances D'_i with (3) and estimate location $\widehat{\ell}(s)$; else use the distances D_i calculated in step 1 to determine location.

Methodology

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Metrics

- *Performance Degradation*: mean positioning error ($\mathcal{E}$) vs percentage of faulty APs
- *Fault Tolerance*: percentage of faulty APs tolerated so that $\mathcal{E} \leq ub$ (e.g. $ub = 5\mathrm{m}$)

Existing Positioning Algorithms

- KNN that uses the standard distance metric (2)
- Probabilistic Minimum Mean Square Error (MMSE)

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

- The median-based KNN variant (MED)

$$\widehat{\ell}(s) = \arg\min_{\ell_i} D_i, \quad D_i = \mathrm{med}_{j=1}^{n} \left(r_{ij} - s_j\right)^2$$

Center for Intelligent Systems & Networks

Results at KIOS with 9 APs

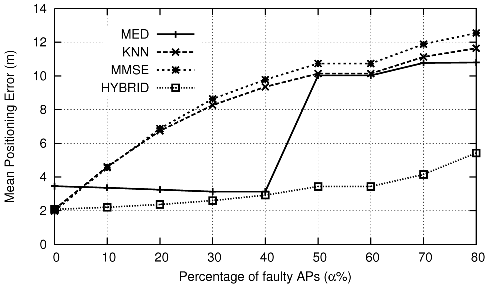
Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- **Results**

Conclusions
- Concluding Remarks



- For KNN and MMSE $\mathcal{E}$ degrades sharply when $\alpha > 10\%$
- HYBRID and MED exhibit similar fault tolerance in case $\alpha \leq 40\%$
- For the HYBRID algorithm $\mathcal{E} = 2.07m$ in the fault-free case, while for MED $\mathcal{E} = 3.45m$
- For MED $\mathcal{E}$ explodes when $\alpha \geq 50\%$ (requires that at least half of the APs provide uncorrupted RSS values)

Results at KIOS with 9 APs

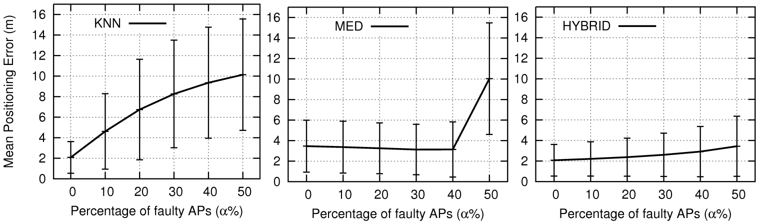
Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks



- When $\alpha = 50\%$, for KNN $\mathcal{E}$ is increased by 8m compared to the fault-free case ($std = 5.5m$)
- For HYBRID $\mathcal{E}$ is only increased by 0.85m when α grows up to 50% ($std = 2.44m$)

Results at KIOS with 73 APs

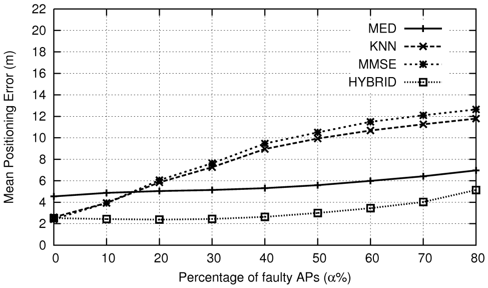
Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks



- KNN and MMSE perform poorly when $\alpha > 20\%$
- HYBRID is extremely fault tolerant: when $\alpha = 50\%$, $\mathcal{E} = 3.0m$ compared to 6.0m (MED), 9.9m (KNN) and 10.5m (MMSE)
- If $\mathcal{E} = 5.0m$ is acceptable, HYBRID can tolerate 80% faulty APs, compared to 30% (MED) and only 10% (KNN, MMSE)

Results at KIOS with 73 APs

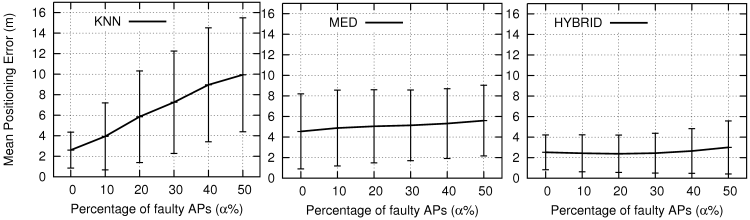
Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks



- For KNN, $\mathcal{E}$ is increased by 7.3m when $\alpha = 50\%$ compared to the fault-free case
- For MED $\mathcal{E}$ is only increased by 1m when $\alpha = 50\%$ and $std = 3.5m$, however it is still outperformed by HYBRID
- For HYBRID $\mathcal{E}$ is only increased by 0.5m when $\alpha = 50\%$ and std remains below 2.6m

Concluding Remarks

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Our Contributions

- Focus on the **Fault Tolerance** of fingerprint-based positioning algorithms, instead of absolute positioning error
- Developed a **robust fault detection** scheme to signify faults
- Introduced a **Hybrid** algorithm that combines the fault detection mechanism with a modified Euclidean distance metric
- Experimental results indicate improved fault tolerance compared to existing algorithms

Future Work

- Apply to different types of faults (e.g. AP relocation)
- Extend our approach to probabilistic fingerprint-based algorithms (e.g. effect of faults on the maximum probability)

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

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

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Extra slides

KOIOS Center for Intelligent Systems & Networks

Effectiveness of the Modified Metric

Location	AP1	AP2	AP3	AP4	AP5	AP6
ℓ_1	-55	-70	-63	-78	NaN	-81
ℓ_2	-67	-87	NaN	-47	-66	-43
ℓ_3	-44	-65	-50	NaN	-52	-87
ℓ_4	NaN	-45	-83	-59	-60	-51
ℓ_5	-48	-69	-58	-83	-59	NaN
ℓ_6	-39	NaN	-68	-76	NaN	-55

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Effectiveness of the Modified Metric

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- **Concluding Remarks**

Location	AP1	AP2	AP3	AP4	AP5	AP6
ℓ_1	−55	−70	−63	−78	*NaN*	−81
ℓ_2	−67	−87	*NaN*	−47	−66	−43
ℓ_3	−44	−65	−50	*NaN*	−52	−87
ℓ_4	*NaN*	−45	−83	−59	−60	−51
ℓ_5	−48	−69	−58	−83	−59	*NaN*
ℓ_6	−39	*NaN*	−68	−76	*NaN*	−55

Fault-free Case

- Observed fingerprint: $s = [-48, -61, -48, NaN, -44, -80]$
- Using (2) or (3) we obtain the ordering $\{\ell_3, \ell_5, \ell_1, \ell_6, \ell_4, \ell_2\}$

Center for Intelligent Systems & Networks

Effectiveness of the Modified Metric

Introduction
- Motivation
- Fault Model
- Measurement Setup

Nearest Neighbor Algorithm
- Fault Detection
- Fault Tolerance

Hybrid Positioning Algorithm

Experimental Evaluation
- Results

Conclusions
- Concluding Remarks

Location	**AP1**	**AP2**	**AP3**	**AP4**	**AP5**	**AP6**
ℓ_1	−55	−70	−63	−78	*NaN*	−81
ℓ_2	−67	−87	*NaN*	−47	−66	−43
ℓ_3	−44	−65	−50	*NaN*	−52	−87
ℓ_4	*NaN*	−45	−83	−59	−60	−51
ℓ_5	−48	−69	−58	−83	−59	*NaN*
ℓ_6	−39	*NaN*	−68	−76	*NaN*	−55

Fault-free Case

- Observed fingerprint: $s = [-48, -61, -48, NaN, -44, -80]$
- Using (2) or (3) we obtain the ordering $\{\ell_3, \ell_5, \ell_1, \ell_6, \ell_4, \ell_2\}$

Failures in AP1 and AP5

- Corrupt fingerprint: $\tilde{s} = [\mathbf{NaN}, -61, -48, NaN, \mathbf{NaN}, -80]$
- Using (2) we obtain the wrong ordering $\{\ell_1, \ell_5, \ell_3, \ell_4, \ell_6, \ell_2\}$
- Using the Modified Metric (3) the correct ordering is preserved