KIOS Center for Intelligent Systems & Networks

Health Monitoring of WLAN Localization Infrastructure using Smartphone Inertial Sensors

Raja Umair Haider[†], **Christos Laoudias**[*] and Christos Panayiotou[*]

[†]Department of Electrical Engineering, Linköpings Universitet, Linköping, Sweden
[*]KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Cyprus

Supported by the Cyprus Research Promotion Foundation under Grant ΤΠΕ/ΟΡΙΖΟ/0609(ΒΕ)/06

Outline

Motivation & Objective

Motivation

- Location Fingerprinting (LF) is a popular method for WLAN localization owing to the ubiquity of Access Points (AP)
- LF is very sensitive to changes in the AP deployment, thus any fault in these APs can severely degrade accuracy
- **Example:** An AP may become unavailable during positioning due to equipment failure or if an attacker unplugs it

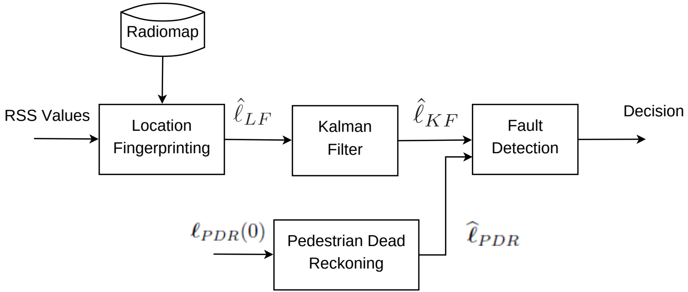
Objective

Develop an automated solution to monitor the health status of the WLAN infrastructure for fault detection and mitigation

Main Idea

Exploit two parallel location streams, one coming from LF and the other computed with onboard inertial sensors to detect AP faults

Health Monitoring System



- The LF module uses the RSS values observed while walking to estimate the unknown user location
- The KF module uses an appropriate mobility model to smooth the current LF location estimate
- The PDR module implements an infrastructure-free approach that uses sensory data for user tracking
- KF and PDR location streams are provided as inputs to the FD module that signifies the presence of an AP fault or not

Location Fingerprinting Module

Offline phase – Radiomap construction

Location	AP_1	AP_2	AP_3	AP_4
ℓ_1	-30	-70	NaN	-58
ℓ_2	-49	NaN	-65	-65
ℓ_3	-70	-30	NaN	-80
ℓ_4	-80	NaN	NaN	-70
ℓ_5	-65	NaN	-49	-49

Online phase – Positioning

- The positioning fingerprint is processed with the probabilistic MMSE algorithm to determine location
- NaN values are replaced with a low RSS value to penalize missing APs in the radiomap and positioning fingerprints
- If an AP is not available due to a fault its value will be replaced by this low RSS value, thus introducing errors

Kalman Filter Module

- LF estimates may sometimes deviate from the actual user location due to noise in the RSS measurements
- KF processes the current LF estimate $\widehat{\ell}_{LF} = (\widehat{x}_{LF}, \widehat{y}_{LF})$ for smoothing and removing outlier locations
- KF implementation
  - Assumes a constant speed user mobility model
  - **Prediction step:** Projects the previous filter estimate ahead in time
  - **Correction step:** Uses the LF estimate to update the prediction
  - The filter output $\hat{\ell}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$ reflects better the travelled path

Pedestrian Dead Reckoning Module

PDR approach

- Step detection and step heading determination
- Step counting and location projection using fixed step length S and the measured heading angle θ_k

$$\hat{x}_k = \hat{x}_{k-1} + S\sin(\theta_k) \quad (1)$$
$$\hat{y}_k = \hat{y}_{k-1} + S\cos(\theta_k) \quad (2)$$

PDR implementation

- Custom application developed on HTC Desire smartphone to collect data from the accelerometer and orientation sensors
- Accelerometer was sampled at 40 Hz and orientation sensor was sampled at 10 Hz
- Step counting through peak detection on the filtered total acceleration signal

Fault Detection Module

Main Idea

FD module takes as inputs the location streams from the KF and PDR modules, and decides whether there is a faulty AP or not

Detection Mechanism

- Calculate mean positioning residual $\bar{\epsilon}$ using successive location estimates over a section of the travelled path

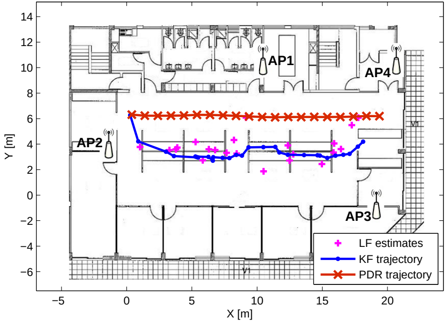
$$\bar{\epsilon} = \frac{1}{K}\sum_{k=1}^{K}\epsilon_k, \quad \epsilon_k = \|\hat{\ell}_{KF}(t_k) - \hat{\ell}_{PDR}(t_k)\| \tag{3}$$

- Compare $\bar{\epsilon}$ to an appropriately selected threshold γ and if $\bar{\epsilon} \geq \gamma$, the presence of an AP fault is signified

Intuition

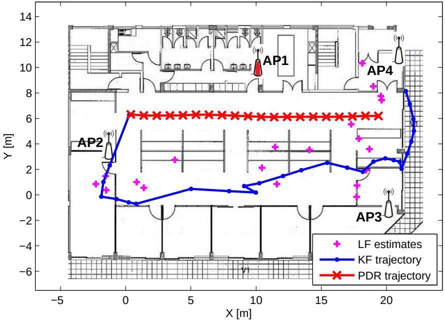
If one or more APs have failed during positioning, then high errors will be introduced in the LF estimates and the residual error between the KF and PDR trajectories will be increased

Concept Demonstration



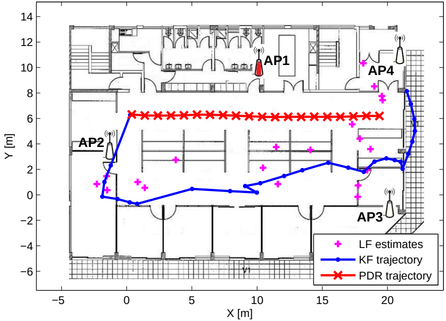
- **Fault-free case:** PDR trajectory is accurate, while the KF travelled path includes some erroneous locations, due to the noise in the RSS values that degrade the LF estimates

Concept Demonstration



- **Fault-free case:** PDR trajectory is accurate, while the KF travelled path includes some erroneous locations, due to the noise in the RSS values that degrade the LF estimates
- **Faulty case:** If AP_1 is faulty, then the KF trajectory is severely affected and seems to be repelled by the faulty AP

Concept Demonstration



- **Fault-free case:** PDR trajectory is accurate, while the KF travelled path includes some erroneous locations, due to the noise in the RSS values that degrade the LF estimates
- **Faulty case:** If AP_1 is faulty, then the KF trajectory is severely affected and seems to be repelled by the faulty AP
- This observation leads to a threshold-based detection scheme to distinguish the fault-free from the faulty AP scenario

Threshold Selection

Objectives

The threshold value is crucial and there is a trade-off to guarantee high detection rate, while reducing false detections

- Setting γ at a low value, essentially means that we consider the KF estimated trajectory to be very accurate and this could lead to false alarms in the fault-free case
- If γ is set to a high value, then a faulty AP might go undetected

Practical issues

Physically turning off the APs and then measuring a sample path to calculate $\bar{\epsilon}$ is not practical, due to access restrictions and size of target areas

Solution

Manipulate some real-life positioning data for injecting *artificial* AP faults to aid threshold selection

Artificial Fault Injection

- Collect positioning data from the surrounding APs under normal conditions by walking along a predefined path
- Compute the mean positioning residual in the fault-free case $\bar{\epsilon}_{ff}$
- Manipulate the fault-free positioning data so that the RSS values corresponding to a faulty AP are replaced by NaN values to indicate a missing AP
- These missing RSS values introduce errors in the LF computations, leading to erroneous estimates when the original RSS radiomap is used
- This way, a single AP fault is injected without physically unplugging or removing it
- Compute the mean positioning residual $\bar{\epsilon}_{faulty}$ in the case of an artificial fault and select a threshold value that satisfies

$$\bar{\epsilon}_{ff} < \gamma < \bar{\epsilon}_{faulty} \quad (4)$$

Threshold Selection Algorithm & Guideline

Algorithm

1. Collect LF and PDR positioning data under fault-free conditions along a route in the target area
2. Calculate $\bar{\epsilon}$ between the KF and PDR position estimates over the whole route
3. Use the fault-free positioning data to create different datasets by artificially injecting single AP faults
4. Repeat step 2 for each faulty AP and calculate $\bar{\epsilon}_{faulty}$ using the corresponding faulty AP positioning dataset
5. Select the threshold using $\bar{\epsilon}_{ff} < \gamma < \bar{\epsilon}_{faulty}$, where $\bar{\epsilon}_{faulty}$ is the minimum residual value across all faulty datasets

Guideline

A route should be selected such that most APs in the target area have strong RSS values in that route (i.e., if an AP is far from the route it provides weak RSS values in the fault-free case that makes it hard to capture when it fails)

Related Work & Setup

Related fault detection algorithm

- **Idea:** If the distance between two successive LF estimates exceeds a threshold, then a fault is detected
- **Intuition:** The user is not expected to travel a long distance between the previous and current LF estimates
- **Drawback:** LF estimates between subsequent user locations can be very inaccurate, thus increasing false detections

Comparison setup

We modified this policy to compare with the proposed scheme using a sample path with 28 distinct locations covered by 4 APs

Threshold selection in our setup

Using artificial fault injection $\overline{\epsilon} \in [5.5, 7.9]$ for all different single AP faults, thus our threshold was set to $\gamma = 5\,\mathrm{m}$.

Results

False Detection Rate

- For 5 test walks along the sample path with no AP faults we found that $\bar{\epsilon} \in [3.1, 4.2]$
- The proposed scheme had no false detections because the threshold $\gamma = 5\,\text{m}$ is not exceeded for any test walk
- The other algorithm had 60% false detections due to some highly erroneous LF estimates caused by noise and outliers

Correct Detection Rate

- AP_1, AP_2 and AP_3 were manually turned off and 5 new test walks were sampled in the presence of a real single AP fault
- $\bar{\epsilon} \in [4.7, 9.1]$ for all faulty AP scenarios
- The proposed scheme achieves [80% 100% 80%] correct detection rate compared to [80% 20% 40%]

Concluding Remarks

- Traditionally, PDR has been used as a complementary technology to enhance the performance of LF systems
- We demonstrated how it can be used in parallel with LF to detect AP faults (transparently from the user)
- Preliminary results from our proof-of-concept prototype are promising, but further experiments are required
- Possible applications
  - **Fault Detection:** Trigger alarm for security and/or maintenance personnel to take action
  - **Fault mitigation:** Revert to fault tolerant LF algorithms to prevent high localization errors

Future Work

- Investigate multiple and other types of faults (e.g., AP relocation) and fault isolation (i.e., identify the faulty AP)
- Address PDR limitations on smartphones
- Examine alternative detection methods, e.g., pattern-based

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

Extra Slides

Kalman Filter: Implementation Details

$$\bar{\mathbf{a}}_k = \mathbf{\Phi}\mathbf{a}_{k-1} \quad (5)$$

$$\bar{\mathbf{P}}_k = \mathbf{\Phi}\mathbf{P}_{k-1}\mathbf{\Phi}^T + \mathbf{\Gamma}\mathbf{Q}\mathbf{\Gamma}^T \quad (6)$$

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{M}^T\left(\mathbf{M}\bar{\mathbf{P}}_k\mathbf{M}^T + \mathbf{R}\right)^{-1} \quad (7)$$

$$\mathbf{a}_k = \bar{\mathbf{a}}_k + \mathbf{K}_k\left(\mathbf{b}_k - \mathbf{M}\bar{\mathbf{a}}_k)\right) \quad (8)$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}\mathbf{M}\right)\bar{\mathbf{P}}_k, \quad (9)$$

- $\mathbf{a}_k = [x(t_k)\ y(t_k)\ u_x(t_k)\ u_y(t_k)]^T$ is the vector of the user location $[x(t_k)\ y(t_k)]^T$ and velocity $[u_x(t_k)\ u_y(t_k)]$ at time t_k
- $\bar{\mathbf{a}}_k$ denotes the one-step ahead prediction of the filter
- $\mathbf{P}$ is the error covariance matrix, $\mathbf{K}_k$ is the KF gain, $\mathbf{b}_k$ is the LF estimate $\widehat{\ell}_{LF}$ at time t_k and $\mathbf{I}$ denotes the identity matrix
- $\mathbf{R} = \sigma_R^2\mathbf{I}$ and $\mathbf{Q} = \sigma_Q^2\mathbf{I}$ are the measurement and process noise covariance matrices, where σ_R^2 and σ_Q^2 represent the uncertainties in the LF estimates and the mobility model

False vs Correct Detection Rate

- Our evaluation shows that AP_1 and AP_3 have 20% probability of missing an AP fault, when $\gamma = 5\,\mathrm{m}$
- A lower threshold value, e.g. $\gamma = 4\,\mathrm{m}$, leads to higher fault detection rate
- This comes at the expense of 40% false detections in the fault-free case, compared with no false detections when $\gamma = 5\,\mathrm{m}$
- The threshold should be selected in light of this trade-off and if we can tolerate some false detections, then we can be more conservative