Device Signal Strength Self-Calibration using Histograms

Christos Laoudias[*], Robert Piché[†] and Christos Panayiotou[*]

[*] KIOS Research Center for Intelligent Systems and Networks, University of Cyprus
[†] Tampere University of Technology, Tampere, Finland

Outline

Introduction

Device Calibration

Self-Calibration

Performance Evaluation

Conclusions

Motivation of our work

- RSS is intended for determining the signal quality and not for positioning purposes

Motivation of our work

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - Each vendor's implementation is limited up to $\mathrm{RSSI}_{\max}$
  - RSSI is mapped to power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging

Motivation of our work

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - Each vendor's implementation is limited up to $\mathrm{RSSI}_{\max}$
  - RSSI is mapped to power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging
- Using a new device for positioning is feasible, but the RSS values are not compatible with the radiomap

Motivation of our work

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - Each vendor's implementation is limited up to $\mathrm{RSSI}_{\max}$
  - RSSI is mapped to power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging
- Using a new device for positioning is feasible, but the RSS values are not compatible with the radiomap
- Best accuracy is guaranteed only if the user carries the same device during positioning, otherwise *calibration* is required

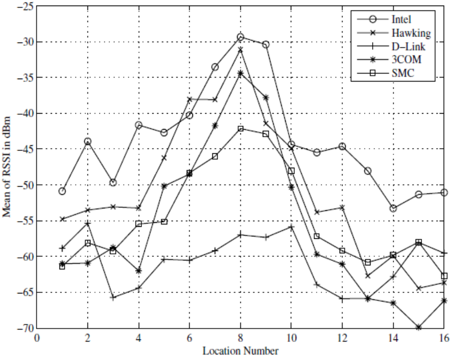
KIOS Center for Intelligent Systems & Networks

Motivation of our work

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - Each vendor's implementation is limited up to $\mathrm{RSSI}_{\max}$
  - RSSI is mapped to power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging
- Using a new device for positioning is feasible, but the RSS values are not compatible with the radiomap
- Best accuracy is guaranteed only if the user carries the same device during positioning, otherwise *calibration* is required
- Existing calibration methods do not fit well in real-time positioning scenarios

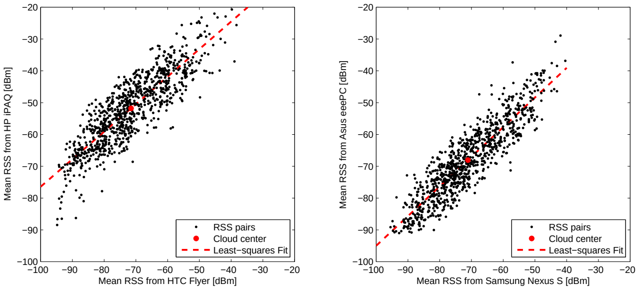
Device Diversity

Vendor	Model	Chipset	Max (dBm)	Min (dBm)	Range
3COM	3CRUSB10075	unknown	+10	-94	104
D-Link	AirPlus DWL-650+	Texas Instrument	-50	-100	50
SMC	EZ Connect SMC2635W	ADMTek	-14	-82	68
Hawking Technology	HWC54G Rev.R	Prism GT	0	-75	75
Intel	PRO/Wireless 2200BG	Intel	-10	-84	74



Source: K. Kaemarungsi (2006)

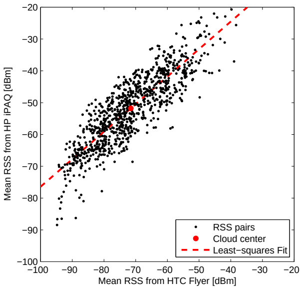
Good News: Linearity between RSS values

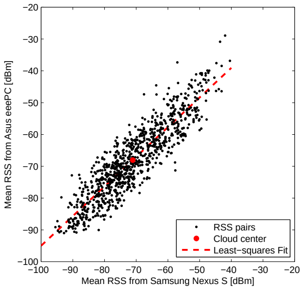


- **Manual Calibration**: Collect several colocated RSS pairs at *known* locations and estimate the linear coefficients through least squares

$$\bar{r}_{ij}^{(2)} = \alpha_{12}\bar{r}_{ij}^{(1)} + \beta_{12}$$

Good News: Linearity between RSS values





- **Manual Calibration**: Collect several colocated RSS pairs at *known* locations and estimate the linear coefficients through least squares

$$\bar{r}_{ij}^{(2)} = \alpha_{12}\bar{r}_{ij}^{(1)} + \beta_{12}$$

- Limited Applicability: (i) User needs to be familiar with the indoor area and (ii) a considerable data collection effort is required

Can we do it more efficiently?

Objectives

- Fully automatic approach with short calibration time
- Runs concurrently with positioning while the user walks around
- No user intervention or tedious data collection

Idea

- Perform device self-calibration on-the-fly using histograms of RSS values observed simultaneously with positioning

RSS Histograms

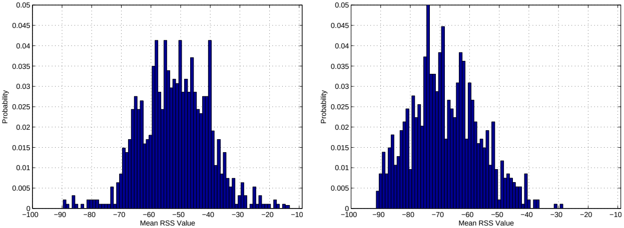


Figure: HP iPAQ (left) and Asus eeePC (right)

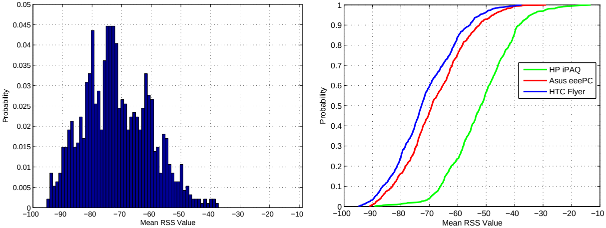
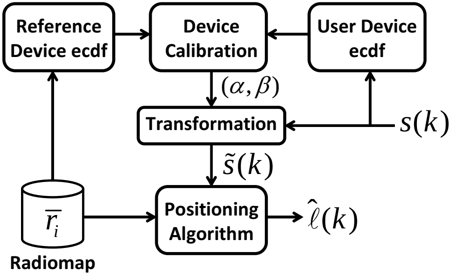
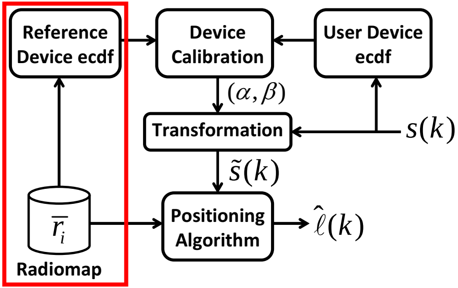


Figure: HTC Flyer (left) and Empirical cdfs (right)

Self-Calibration Method

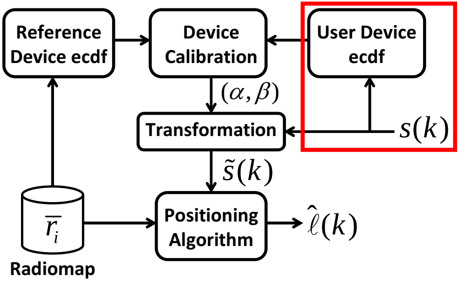


Self-Calibration Method



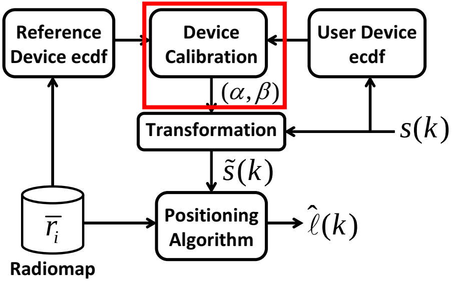
1. Create the ecdf of the reference device from the radiomap

Self-Calibration Method



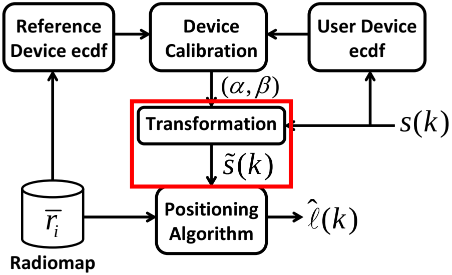
1. Create the ecdf of the reference device from the radiomap
2. Create and update the ecdf of the new device by using $s(k)$

Self-Calibration Method



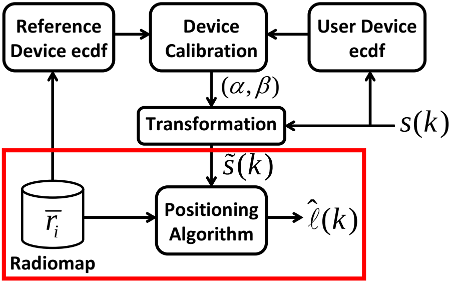
1. Create the ecdf of the reference device from the radiomap
2. Create and update the ecdf of the new device by using $s(k)$
3. Fit a linear mapping between the reference and new device to obtain (α, β) by using "representative" ecdf values

Self-Calibration Method



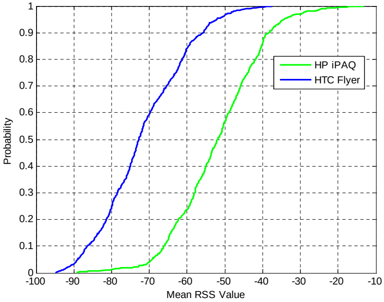
1. Create the ecdf of the reference device from the radiomap
2. Create and update the ecdf of the new device by using $s(k)$
3. Fit a linear mapping between the reference and new device to obtain (α, β) by using "representative" ecdf values
4. Transform the observed RSS values with $\tilde{s}_j(k) = \alpha s_j(k) + \beta$

Self-Calibration Method



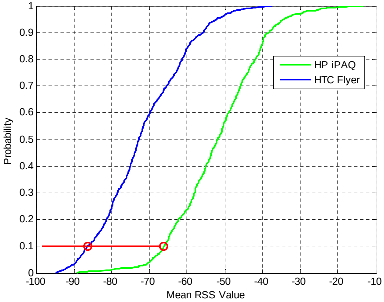
1. Create the ecdf of the reference device from the radiomap
2. Create and update the ecdf of the new device by using $s(k)$
3. Fit a linear mapping between the reference and new device to obtain (α, β) by using "representative" ecdf values
4. Transform the observed RSS values with $\tilde{s}_j(k) = \alpha s_j(k) + \beta$
5. Estimate location $\hat{\ell}(k)$ with any fingerprint-based algorithm

Inverse ecdf Linear fitting



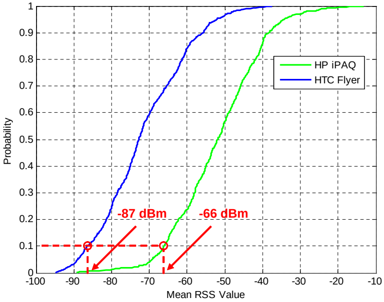
- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ are the ecdfs of the reference and user device
- $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta, \;\; y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Inverse ecdf Linear fitting



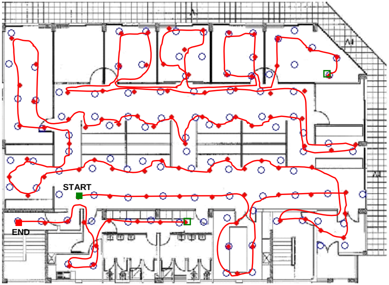
- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ are the ecdfs of the reference and user device
- $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta,\ \ y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Inverse ecdf Linear fitting



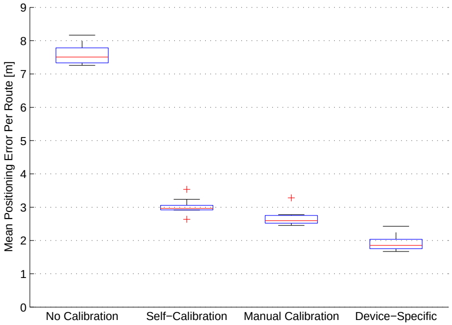
- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ are the ecdfs of the reference and user device
- $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta, \ y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Experimental Setup



- 560 m^2 office, 9 WiFi APs, 5 devices (1 HP iPAQ PDA, 1 Asus eeePC laptop, 1 HTC Flyer Android tablet, 2 Android smartphones)
- **Training Data:** 105 reference locations, 20 fingerprints per location (2100 in total) with each device for comparison
- **Testing Data:** Route with 2 segments, 96 test locations, 1 fingerprint per location, route sampled 10 times

Experimental Results – 10 Routes



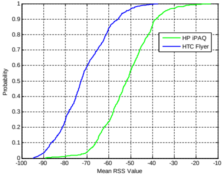
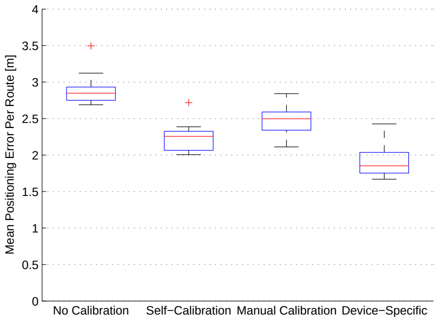


Figure: HTC Flyer user with HP iPAQ radiomap

Experimental Results – 10 Routes



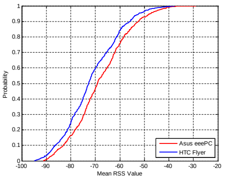
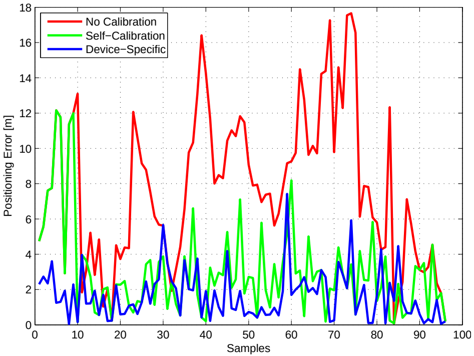


Figure: HTC Flyer user with Asus eeePC radiomap

Experimental Results – Single Route



- iPAQ radiomap with Flyer user-carried device
- For the first 10 sec the device is uncalibrated and accuracy is not adequate
- Beyond that point, the device is automatically calibrated and accuracy is greatly improved

Results with pairwise device combinations

Table: Median of the mean error $\bar{\epsilon}$ [m], with and without calibration.

	iPAQ	**eeePC**	**Flyer**	**Desire**	**Nexus S**
iPAQ	2.7	2.8 (6.6)	3.0 (7.5)	2.9 (8.4)	2.6 (7.7)
eeePC	2.8 (4.4)	2.3	2.3 (2.8)	2.6 (3.5)	2.5 (2.9)
Flyer	3.2 (5.9)	2.6 (3.0)	1.9	2.1 (2.3)	2.6 (2.7)
Desire	3.4 (6.1)	2.8 (3.2)	2.5 (2.5)	2.4	2.5 (2.6)
Nexus S	3.0 (6.2)	2.6 (2.8)	2.7 (2.7)	2.4 (2.5)	2.3

- All 5 devices used as a reference (row) and test device (column)
- Mean positioning error using device self-calibration (results without calibration shown in parentheses)
- The diagonal cells report the accuracy when the reference and test devices are the same (i.e. device-specific radiomap is used)
- Self-calibration improves the accuracy for all device pairs

Concluding Remarks

Device diversity is one of the reasons that hinders the proliferation of RSS-based positioning systems.

Our Contributions

- Low-complexity, yet effective method that allows any mobile device to be self-calibrated
- Automatic calibration is attained shortly after the user has started positioning

Future Work

- Application in larger scale setups featuring non uniform WiFi AP layouts (possible skewness of the RSS histograms)
- Integrate with our **Airplace** indoor positioning platform developed for Android smartphones

`http://www2.ucy.ac.cy/~laoudias/pages/platform.html`

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

RSS Difference Approach

Assume that a mobile device resides at a location ℓ, which is covered by 2 WiFi APs, namely AP_1 and AP_2. The RSS values recorded by the device are given by

$$\begin{aligned} RSS_1 &= A - 10\gamma \log_{10} d_1 + X_1 \\ RSS_2 &= A - 10\gamma \log_{10} d_2 + X_2 \end{aligned}$$

where d_i, $i = 1, 2$ is the distance from the i-th AP, while $X_1, X_2 \sim \mathcal{N}(0, \sigma_n^2)$ are independent Gaussian noise components disturbing the RSS values.
Taking the difference of these RSS values, denoted as $RSSD_{12}$, gives

$$RSSD_{12} = RSS_1 - RSS_2 = 10\gamma \log_{10} \frac{d_2}{d_1} + X'$$

where $X' \sim \mathcal{N}(0, 2\sigma_n^2)$ is the linear combination of X_1, X_2.

Inverse ecdf Least Squares Fitting

If $\mathbf{u}$ is a continuous random variable and $\mathbf{y} = f(\mathbf{u})$ with monotonically increasing f then $f = F_{\mathbf{y}}^{-1} \circ F_{\mathbf{u}}$. In particular, the inverse cdf ordered pairs

$$\{(u_i, y_i) = (F_{\mathbf{u}}^{-1}(q_i), F_{\mathbf{y}}^{-1}(q_i)) : q_i \in \{0.1, \dots, 0.9\}\}$$

lie on the curve $y = f(u)$.

Proof:
We have

$$\begin{aligned} F_{\mathbf{u}}(u) &= P(\mathbf{u} \leq u) = P(f(\mathbf{u}) \leq f(u)) = \\ &= P(\mathbf{y} \leq f(u)) = F_{\mathbf{y}}(f(u)). \end{aligned}$$

Applying $F_{\mathbf{y}}^{-1}$ to both sides gives the identity $f = F_{\mathbf{y}}^{-1} \circ F_{\mathbf{u}}$. Also, the components of the inverse cdf ordered pairs satisfy

$$y_i = F_{\mathbf{y}}^{-1}(q_i) = F_{\mathbf{y}}^{-1}(F_{\mathbf{u}}(u_i)) = f(u_i).$$