Demo Abstract: Virtual Experimental Evaluation of RF-based Indoor Localization Algorithms

Filip Lemic*, Vlado Handziski*, Niklas Wirström†, Tom Van Haute‡,
Eli De Poorter‡, Thiemo Voigt†, Adam Wolisz*
*Telecommunication Networks Group (TKN), Technical University of Berlin (TUB)
†Swedish Institute of Computer Science (SICS)
‡Department of Information Technology (INTEC), Ghent University - iMinds

***Abstract*—This demonstration presents a set of services for streamlined experimental evaluation and benchmarking of RF-based indoor localization algorithms using previously collected raw measurements. The platform consists of an online service for storing and managing raw indoor localization data collected in a set of extensive experiments. The platform also integrates a cloud-based service for calculation of a standardized set of metrics for characterizing the performance of indoor localization algorithms. To simplify the access to the above services, we also offer a set of Software Development Kits (SDKs) for their use from Python and MATLAB. Experimenters are able to "link" the platform to their indoor localization algorithms, use previously collected data to evaluate the performance of their algorithms and calculate a set of metrics for characterizing their performance.**

I. Introduction

Indoor localization algorithms are usually benchmarked in different environments and scenarios, mostly using different hardware. Thus, even with the usage of a standardized set of metrics the results from experimental valuation are not easily comparable. In addition, experimental benchmarking of indoor localization algorithms is labor, time and cost expensive.

Within the EVARILOS project [1], we address these drawbacks by providing a set of online services for experimental benchmarking of Radio Frequency (RF)-based indoor localization algorithms without the overhead of running real measurement campaigns. Any algorithm can be evaluated and compared with other ones by running it on exactly the same raw datasets, where the raw data is a typical low-level input to RF-based localization algorithms like Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), etc. The focus on raw input data differentiates our approach from the one taken in related efforts like VirTIL [2], which exports already processed range (distance) values as the basic datum. For the purpose of virtual evaluation we provide two online services: a service for access and management of database of raw localization data collected in extensive measurement campaigns and a service for calculation of an extensive set of metrics for characterizing the performance of indoor localization algorithms. The platform also includes two SDKs, for the Python and MATLAB programming languages, providing functions for easy interaction with the above introduced services.

II. Platform Overview

The overview of the platform is given in Figure 1. The service for managing the raw data is responsible for storing and making available measurement datasets collected in experimental campaigns. A detailed description of the service and its functions is given in [3]. The measurements are stored together with the locations where they are taken, annotated with the locations of the transmitting devices, metadata describing the environment and hardware used for the collection. The service provides a publicly available Application Programming Interface (API) for managing the stored data, where the user can "browse" the stored datasets and select the desired one. We further provide the visualization tool that enables users to easily visualize the collected information stored in the provided measurement collections.

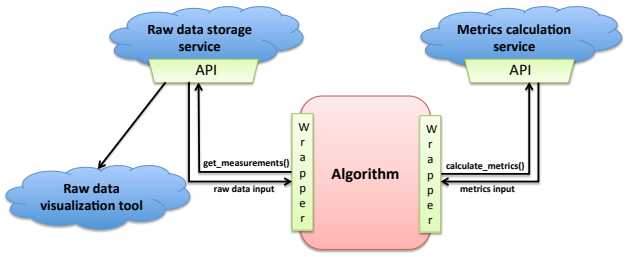


Fig. 1. Overview of the platform

The platform further consists of a set of software SDKs developed for Python and MATLAB programming languages, which are selected because they are widely used for prototyping various kinds of software algorithms, including those for indoor localization. The user can use the wrappers to fetch the desired measurements through a single function call. The user is then able to input the fetched data to the algorithm to be evaluated. The output of the algorithm, i.e. the estimated location can, together with the ground-truth coordinate where the measurement was taken, be sent to a cloud service for calculation of the performance metrics using a single function call provided by the wrappers. In that way, the user is able to easily evaluate the performance of its algorithm, using the experimentally collected data as the input and receiving a set

of standardized metrics as the output of the procedure [4]. A snapshot of the raw data is given with Listing 1. The data consists of RSSI measurements, accompanying metadata (timestamp, transmitter ID, run number, etc.) and locations of transmitting and receiving devices.

Listing 1. Raw data format

```
{
receiver_id: "MacBook Pro",
run_nr: 13,
timestamp_utc: 1373126790,
sender_id: "tplink08",
sender_bssid: "64:70:02:3e:aa:11",
rssi: -42,
channel: "11",
receiver_location: {
    room_label: "FT226",
    coordinate_z: 9.53,
    coordinate_y: 1.67,
    coordinate_x: 23.9},
sender_location: {
        room_label: "FT226",
    coordinate_z: 10.9,
        coordinate_y: 0.7,
        coordinate_x: 31},
}
```

The SDKs wrap the interaction with the cloud service for the data storage using a simple API shown in Listing 2. Using the "get_measurements" command, the experimenter is able to fetch the data from an experiment or a specific measurement. With the "filtering" command, it is possible to filter the fetched data based on desired parameters, such as number of measurements, wireless channel or transmitting device. Finally, using the "calculate_metrics" call, the experimenter is able to obtain the standardized set of the performance metrics.

Listing 2. Python API overview

```
def get_measurements(database, experiment, measurement);
def filtering(measurement, num_meas, channel, sender);
def calculate_metrics(data);
```

As the API shows, the platform offers a set of services for easy "scoping and filtering" this data, so that experimenters can selectively ask for specific record at a set of location coordinates and for a given technology and then get this dataset in an efficient way. This approach, in comparison to plain "downloading" of measurement traces, offers several benefits. Experimental raw datasets for evaluation of indoor localization algorithms can be very large. Especially for "universal" data sets that can be used for evaluation of different localization algorithms, the aim is to collect data at high spatial sampling densities and using diverse hardware equipment. At the same time, any particular algorithm would likely use only a small subset of this data in a given evaluation campaign. Approach of disseminating the whole raw data sets as "downloadable" files is thus very inefficient and leaves to the user the burden of finding the nuggets of relevant data from the whole dataset. The alternative that we offer, an online service for management of this data, is much more convenient for the users.

III. Demo Description

In this demonstration we show how the service for storage of the raw data from indoor localization benchmarking experiments can be accessed and how one can "browse" the available data collections. We also present the functionalities of the visualization tool and how it can be used to easily access the raw data and the metadata related to each data collection. Finally, we show the fetching and filtering capabilities of the SDKs for Python and MATLAB, how the data can be used by a simple WiFi-based fingerprinting algorithm and how the metrics can be calculated with one function call by interacting with the online service.

The above features will be shown on the basis of a dataset collected in the TWIST tesbted [5]. It contains multiple collections of IEEE 802.11 beacon packets RSSI values from APs distributed in locations depicted as blue squares in Figure 2. The dataset also contains collections of beacon packets from IEEE 802.15.4 nodes deployed on positions depicted with dots.

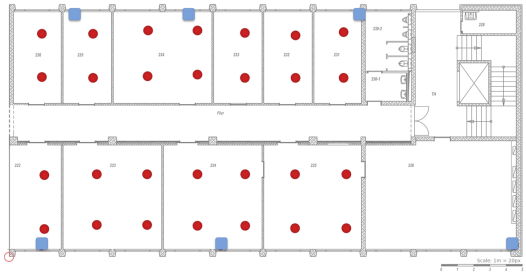


Fig. 2. Transmitting nodes locations in the testbed

IV. Conclusion and Future Work

This work demonstrates a set of tools and measurements collections that can easily be used for experimental benchmarking of IEEE 802.11 and IEEE 802.15.4 RSSI-based indoor localization algorithms, without a need for a testbed and all complexities and costs that usage of testbed introduces. Future work will be focused on collections of different types of data, such as Time of Arrival and Angle of Arrival (AoA).

Acknowledgments

This work has been partially funded by the European Commission (FP7-ICT-FIRE) within the project EVARILOS (grant No. 317989). The author Filip Lemic was partially supported by DAAD (German Academic Exchange Service).

References

[1] *Project EVARILOS*, 2013. [Online]. Available: www.evarilos.eu.

[2] S. Schmitt *et al.*, "A Virtual Indoor Localization Testbed for Wireless Sensor Networks," in *SECON'13*, 2013.

[3] F. Lemic and V. Handziski, "Data Management Services for Evaluation of RF-based Indoor Localization," Telecommunication Networks Group, Tech. Rep. TKN-14-002, 2014.

[4] F. Lemic, "Service for Calculation of Performance Metrics of Indoor Localization Benchmarking Experiments," Telecommunication Networks Group, Tech. Rep. TKN-14-003, 2014.

[5] V. Handziski *et al.*, "TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Network," in *RealMAN'06*, 2006.

Virtual Experimental Evaluation of RF-based Indoor Localization Algorithms

Filip Lemic, Vlado Handziski, Adam Wolisz
Telecommunication Networks Group (TKN)
Technische Universität Berlin (TUB)
{lemic,handziski,wolisz}@tkn.tu-berlin.de

Niklas Wirström, Thiemo Voigt
Swedish Institute of Computer Science (SICS)
{niwi,thiemo}@sics.se

Tom Van Haute, Eli De Poorter
Department of Information Technology (INTEC)
Ghent University - iMinds
{tom.vanhaute,eli.depoorter}@intec.ugent.be

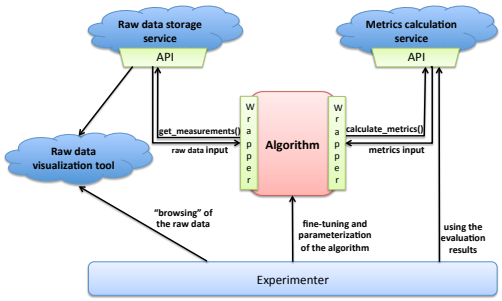
Introduction

Benchmarking of indoor localization algorithms:

- Indoor localization algorithms are usually benchmarked in different environments and scenarios, mostly using different hardware.
- The metrics used for characterizing the performance of the algorithms are usually not standardized and thus hardly comparable.
- Experimental benchmarking of indoor localization algorithms is labor, time and cost expensive.

We address these drawbacks by providing a set of online services for experimental benchmarking of Radio Frequency (RF)-based indoor localization algorithms without a need of a local testbed infrastructure.

Platform Overview

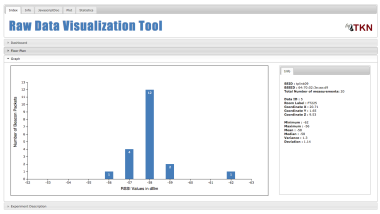


Raw Data Storage Service

Raw data storage service stores previously collected measurements and provides a **publicly available Application Programming Interface (API) for managing the stored data. Additionally, the user can "browse" stored datasets** and select a desired one through the **Raw data visualization tool**. The measurements are stored together with:

- Locations where they are taken,
- Annotated with the locations of the transmitting devices,
- Metadata describing the environment and hardware used for the collection.

```
receiver_id: "MacBook Pro",
run_nr: 15,
timestamp_utc: 1373123492,
sender_id: "tplink08",
sender_bssid: "64:70:02:3e:aa:11",
receiver_location: {
    room_label: "FT226",
    coordinate_z: 9.53,
    coordinate_y: 1.67,
    coordinate_x: 30.28
},
sender_location: {
    room_label: "FT226",
    coordinate_z: 10.9,
    coordinate_y: 0.7,
    coordinate_x: 31
},
rssi: -33,
channel: "11"
```



Metrics Calculation Service

The platform integrates a **cloud-based service for calculation of a standardized set of metrics for characterizing the performance** of indoor localization algorithms:

- Geometrical (point) accuracy of indoor localization,
- Room level accuracy of indoor localization,
- Latency of location estimation,
- Power consumption of a localization device.

```
message Metrics {
        optional double error_average = 1;          // Average localization error
        optional double error_median = 2;           // Median localization error
        optional double error_std = 3;              // Standard deviation of localization error
        optional double error_min = 4;              // Minimum localization error
        optional double error_max = 5;              // Maximum localization error
        optional double room_error = 6;             // Room level accuracy
        optional double latency_average = 7;        // Average latency
        optional double latency_median = 8;         // Median latency
        optional double latency_std = 9;            // Standard deviation of latency
        optional double latency_min = 10;           // Minimum latency
        optional double latency_max = 11;           // Maximum latency
        optional double power_average = 12;         // Average power consumption
        optional double power_median = 13;          // Median power consumption
        optional double power_std = 14;             // Standard deviation of power consumption
        optional double power_min = 15;             // Minimum power consumption
        optional double power_max = 16;             // Maximum power consumption
}
```

Software Libraries

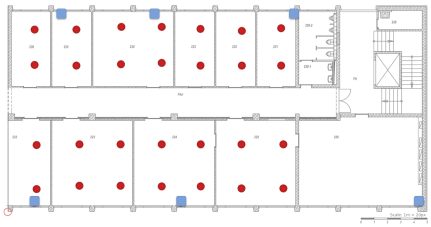
We provide a **set of services for easy "scoping and filtering" this data for Python and MATLAB programming languages**, so that experimenters can selectively ask for specific record at a set of location coordinates and for a given technology and then get this dataset in an efficient way.

```
def get_measurements(database, experiment, measurement);
def filtering(measurement, num_meas, channel, sender);
def calculate_metrics(data);
```

Measurement Collections

The current dataset consists of:

- Collections of IEEE 802.11 beacon packets measured at different locations in our testbed.
- Collections of beacon packets transmitted by the IEEE 802.15.4 TelosB nodes in our testbed.



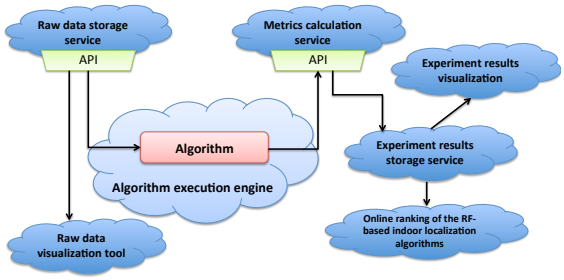
Demo Description

In this demonstration we show:

- How the service for storage of the raw data from indoor localization benchmarking experiments can be accessed and how one can "browse" the available data collections.
- The functionalities of the visualization tool and how it can be used to easily access the data statistics and the metadata related to each data collection.
- The fetching and filtering capabilities of the Software Development Kits (SDKs) for Python and MATLAB.
- How the fetched data can be used by a simple WiFi-based fingerprinting algorithm.
- How the metrics can be calculated with one function call by interacting with the online service for calculation of the performance metrics.

Future Work

- Additional collections of measurements, e.g. Time of Arrival (ToA) and Angle of Arrival (AoA).
- Online ranking and publicly available evaluation results for different indoor localization algorithms in different evaluation scenarios.



Acknowledgments

This work has been partially funded by the European Commission (FP7-ICT-FIRE) within the project EVARILOS (grant No. 317989). The author Filip Lemic was partially supported by DAAD (German Academic Exchange Service).

EWSN 2015
The 12th European Conference on Wireless Sensor Networks
February 9-11, 2015.
CISTER, Porto, Portugal