ABSTRACT

Individual localization services seldom satisfy the requirements for accurate, robust, punctual, and seamless location information. To enhance the provisioning, a set of challenges has to be addressed, pertaining to handover, fusion, and integration of individual localization services. In the following, we advertise the Standardized Localization Service (SLSR), a middleware architecture for achieving those goals. We instantiate the SLSR in an office-like indoor environment and evaluate its performance in a specifically designed testbed for evaluation of indoor localization approaches. Our results typify the contributions of different functional components envisioned in the SLSR on its overall performance.

CSC CONCEPTS

• Networks ➔ Network architectures; Location based services; Network experimentation;

ACM Reference format:

1 INTRODUCTION

Location information of mobile devices is the main input to location-based services and a valuable source of context information in wireless networks. Location information should be seamless, accurate, promptly available, and robust. However, individual localization solutions are not seamlessly deployed and can provide satisfactory performance only in some types of environments [6].

For promptly, seamlessly, and robustly achieving the desired accuracy, there is a need for fusion, handover, and integration of different sources of location information, which is currently missing [1]. To fill this gap, in this work we advertise the Standardized Localization Service (SLSR), a localization service architecture that enables standardized provisioning of location information to the applications irrespective of the services that are providing this information. To this end, we provide an overview of different functional properties of the SLSR. We further evaluate the performance of the SLSR in a testbed environment designed for performance benchmarking of indoor localization solutions. In the evaluation, we aim at showing that each functional property of the SLSR improves its overall performance.

2 OVERVIEW

The Location-Based Application (LBA) is a component of the SLSR, defined as the entity that requires location information of a mobile device. The central entity of the SLSR is the Integrated Location Service (ILS), serving as a setting for fusion, integration, and handover of entities capable of providing location information of a device, i.e. Services for Location Information Provisioning (SLIPs). Assuming that the user is mobile and serviced by different SLIPs, the ILS becomes a proxy with respect to the LBAs’ requirements.

A functional overview of the SLSR is given in Figure 1 and in details discussed in [5]. The process starts by different LBAs requesting location information of certain features from the ILS. The features are desired accuracy and latency of location information, where location information can be requested once, or either periodically or on an event for a certain duration. These requests are received at a listener for requests for location information.

The requests are then forwarded to a long-term request interpreter, whose function is to translate a request for periodic provisioning or for provisioning on events into repeated individual requests for location information. Each request is then potentially resolved with cached location information in case such information is not stale and if it is attributed with an acceptable accuracy for addressing the request. Similarly, each request is possibly resolved with mapping between location information types. All requests resolved by either cached or mapped location information are immediately reported back to the corresponding LBAs.

All unresolved requests are used as an input to an algorithm for selection of SLIPs. In addition, provisioning features of available SLIPs are also used as an input to the algorithm. These features of provisioning are obtained through the discovery of SLIPs and their features, which is initiated by the ILS. Upon request, all available SLIPs estimate and report their provisioning features to the
ILS. Selection of SLIPs to be invoked is therefore based on the requirements from LBAs and provisioning features of available SLIPs. The selection can be subject to various optimization in relation to relevant metrics, i.e. accuracy of location information, and latency and power consumption of provisioning.

The selected SLIPs are in the following step invoked for generating location information. Location information from different SLIPs are received at a listener for location information. Upon receiving each location information, the ILS decides, based on the requirements from an LBA, if this information is forwarded to the LBA. Adversely, the location information can be fused with location information from other SLIPs before its reporting. This fusion process enhances the accuracy of reported location information, while trading-off latency of provisioning, since a certain delay is introduced by waiting for multiple SLIPs to report their location information. The reported information is additionally used for updating cached location information.

### 3 EVALUATION

In its prototypical implementation, the SLSR is realized as a set of Python 2.7-based daemon processes. Messaging between the distributed components of the SLSR is supported by a single-writer log-based paradigm administered by the Global Data Plane (GDP). For the instantiation of its prototypical implementation we used the TWIST testbed, which is an office environment in its usual usage. For the instantiation of SLIPs we selected a set of WiFi Received Signal Strength Indicator (RSSI) fingerprinting-based localization solutions. Two fingerprinting-based solutions have been instantiated due to their practical convenience. However, it is straightforward to instantiate any other type of localization solution under the control of the ILS, as long as the instantiated solution is compliant to the interface specified by the ILS, as discussed in [3]. By leveraging two fingerprinting algorithms and varying the number of training points for fingerprinting as indicated in Figure 2, we achieved different provisioning features of the instantiated SLIP. Requests for location information from LBAs, specified by their accuracy and latency features, duration and type of provisioning, have been specified in a way that mimics a usual person’s tracking scenario, location-based light and thermostat switching scenarios, location-aided horizontal WiFi handover mechanism, etc.

The testbed features a mobility platform capable of autonomously moving a device to be localized over a predefined trajectory [2]. In its positioning in the testbed space, the platform achieves less than 10 cm in average localization error, hence the locations reported over by the platform over the trajectory (Figure 2) were used as ground truths for evaluating the SLSR. Metrics used for performance evaluation include satisfaction of requirements for accuracy and latency of location information provisioning and the total power consumption when the platform is used as the mobility platform.

Table 1 summarizes the evaluation results. The "Basic" type includes only basic functionalities of the ILS, i.e. reception of requests, invocation of SLIPs, reception of location estimates, their fusion and reporting to the LBAs. The "Caching/mapping" type additionally includes caching and mapping functionalities. As visible from the table, introducing the mapping and caching capabilities dramatically reduces the total consumed power, while also highly benefiting both accuracy and latency requirements satisfaction. The "Long-term" type assumes that requests for periodic or on an event provisioning for a certain duration can be specified in one request. This type further benefits the latency of location information provisioning, since it removes the latency overhead of repeatedly sending a request by an LBA. The "Dynamic" type uses a dynamic specification of the latency and accuracy features of provisioning of SLIPs. This type benefits the requirements satisfaction because a more accurate selection of SLIPs can be performed. In conclusion, the supplemental functionalities are reasonable additions to the SLSR because their inclusion improves the SLSR's performance.

<table>
<thead>
<tr>
<th>Type</th>
<th>Total # of requirements</th>
<th>Accuracy satisfaction</th>
<th>Latency satisfaction</th>
<th>Consumed power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>665</td>
<td>234</td>
<td>199</td>
<td>2033</td>
</tr>
<tr>
<td>Caching/mapping</td>
<td>665</td>
<td>361</td>
<td>511</td>
<td>1225</td>
</tr>
<tr>
<td>Long-term</td>
<td>665</td>
<td>371</td>
<td>542</td>
<td>1171</td>
</tr>
<tr>
<td>Dynamic</td>
<td>665</td>
<td>471</td>
<td>537</td>
<td>1376</td>
</tr>
</tbody>
</table>

### DISCLAIMER

The purpose of this document is to advertise the SLSR. A proposal for standardization of the interaction between the components of the SLSR is given in [3]. An in-depth overview of the design, implementation, and evaluation of the SLSR is provided in [5]. Selection of SLIPs based on the requirements from LBAs and provisioning features of the available SLIPs is discussed in [4].

### ACKNOWLEDGMENTS

This work received funding from the European Horizon 2020 Programme in the scope of the EU eWINE Project (Nr. 688116).

### REFERENCES

SLSR: Standardized Localization Service
Filip Lemic, Vlado Handziski, Anatolij Zubow, Adam Wolisz
Telecommunication Networks Group, Technische Universität Berlin, Germany

Goals
- Seamless handover and fusion of different localization services;
- Straightforward addition of new localization services;
- In an unified way enabling portability of location-based applications;

Contributions
- A flexible middle-ware localization service architecture;
- A proposal for standardized interaction between architectural components;
- SLSR: a prototypical implementation of such an architecture;

Example Usage - Horizontal Handover
"Heterogeneous requirements for location information (e.g. once/periodically/on event, flexible accuracy/latency/duration, ...)
1. RSS decrease → an "app" is triggered → the app requests location:
   request_location("2D", "1m accuracy", "1s", "movement")
2. The user is still far from handover locations → but, the user is moving → request location again for the case the user moves for more than 3 m:
   request_location(., "on event 3m", "duration 30sec")
3. The user is close to a handover location → request periodic provisioning:
   request_location(., "periodically", "duration 30sec")
4. The handover occurred → check if the user is progressing towards locations where no handover is needed:
   request_location("movement")
5. Yes, s/he is! → we don’t need location information for a while:

High-level Architecture
- Request_location: type, 2D/3D, accuracy, period, on event, duration, movement;
- request_context: map and mapping between location information types;

Standardized Localization Interface (SLI)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify policy</td>
<td>Specify trade-off policy</td>
<td>LA→SLI</td>
</tr>
<tr>
<td>Request location</td>
<td>Request location information</td>
<td>LA→SLI</td>
</tr>
<tr>
<td>Report location</td>
<td>Report location information</td>
<td>IL→LA</td>
</tr>
<tr>
<td>Request renewal</td>
<td>Request provisioning renewal</td>
<td>LA→SLI</td>
</tr>
<tr>
<td>Request context</td>
<td>Request context of location information</td>
<td>LA→IL</td>
</tr>
</tbody>
</table>

Evaluation
- Fingerprinting-based services;
- Varying algorithms/training set densities → different provisioning features;
- Basic - basic functionalities of the integrated location service;
- Caching & mapping - caching and mapping functionalities introduced;
- Long-term interpretation - pushing intelligence to the integrated location service;
- Dynamic - "God-view" on provisioning features;

Acknowledgments
This work has received funding from the EU eWINE Project (No. 688116).