Grant agreement for:
Collaborative project, Small and medium-scale focused research project (STREP)

Deliverable D3.4:
Library of Algorithms for Wireless Sensor Networks

Project acronym: WISEBED
Project full title: Wireless Sensor Network Testbeds
Grant agreement no.: 224460

Responsible Partner: TUBS
Report Preparation Date: May 31, 2011
Contents

1 Introduction 4

2 OS Facets 6
   2.1 WP2 OSA ................................................................. 6
   2.2 Contiki ................................................................. 7
   2.3 TinyOS ................................................................. 8
   2.4 iSense ................................................................. 8
   2.5 ScatterWeb2 ......................................................... 9
   2.6 Shawn ................................................................. 10
   2.7 Linux ................................................................. 11
   2.8 Feuerware ......................................................... 11
   2.9 TriSOS ................................................................. 12
   2.10 iOS ................................................................. 12
   2.11 Android ......................................................... 13

3 Library of Algorithms for Wireless Sensor Networks 14
   3.1 Routing algorithms (20) ......................................... 15
   3.2 Clustering (12) .................................................. 17
   3.3 Time Synchronization (5) ..................................... 19
   3.4 MAC Layer (5) ................................................ 20
   3.5 Localization (8) ................................................ 21
   3.6 Energy Saving Schemes (6) ................................. 22
   3.7 Security (9) ................................................... 23
   3.8 Graph Algorithms (7) ....................................... 24
   3.9 Target Tracking (2) ......................................... 25
4 Conclusion

References
1 Introduction

The overall goal of WP3 is to design and implement an algorithm library for heterogeneous sensor networks, the Wiselib. While the goals merely state to have a large number of algorithm implementations available to the general public, we have added an extra step that should substantially increase the usefulness and sustainability of the library: We have developed a flexible and efficient framework based on C++ templates, allowing for algorithm development for heterogeneous networks. A first version was developed during WISEBED’s first year. In the second year, it was extended and refined, and more algorithms were added. During the third year, we stabilized existing and extended the list of supported platforms, even integrated mobile devices such as Android and iPhone. The list of algorithms was also extended to nearly 80, with contributions from extern partners and other EU projects.

The architecture of the Wiselib was presented at EWSN 2010 [8]. We do not repeat design paradigms and technical descriptions in this deliverable, but rather refer to this publication.

Currently there are 78 algorithms in 12 categories in the Wiselib. It consists of three separate distributions:

“Incubation” contains algorithm implementations that do not use the generic C++ framework. They only compile on specific platforms or for specific simulators. Algorithms in this library are usually placed here for one of the following reasons:

- They are in a evaluation stage to be later ported to the C++ framework.
- They work under constraints that forbid using the framework (such as MAC layer algorithms, which often require direct hardware access).
- They are written in the context of other WISEBED-related research, where applicability to that research is of higher importance than compatibility to the C++ framework.

“Testing” algorithms use and extend the C++ framework. This distribution follows the “Release Early, Release Often” principle of Open Source. Every WISEBED partner can add algorithms, even if they undergo frequent changes.

“Stable” is a distribution consisting of algorithms that are tested for compatibility with the design principles of the C++ framework, and are known to run on different platforms. Adding algorithms here is restricted to the major Wiselib contributors—RACTI, TUBS, and UPC. These are allowed select mature algorithms from Testing and promote them to Stable.
In publications, only Testing and Stable are referred to as “the Wiselib”. This allows for clearer presentation, as we can focus on the major selling point of cross-platform development.

All three distributions will be released to the public under Open Source licenses. Large portions can already be accessed from the Wiselib website\(^1\). Some Incubation algorithms are available from other sources, such as the websites of their authors.

\(^1\)http://www.wiselib.org
2 OS Facets

The OS facets are the connection to the underlying OS, abstracting hardware functionality for the algorithms. After three years, we have seven facets, with three derived concepts for specialized functionality. There are 11 software platforms supported, running on more than 10 different hardware platforms.

Table 1 provides an overview of all facets and the corresponding support in the various software platforms.

![Table of OS Facets](image)

Figure 1: Support for various operating systems/firmwares in the Wiselib. Red marked facets are new in year 3.

2.1 WP2 OSA

**OS Facet**

*Contributor(s):* TUBS, ULANC  
*Status:* Fully supported  
*Target:* WP2 OSA
| **Radio Facet** | **Contributor(s):** TUBS, ULANC | **Status:** Works, proof of concept | **Target:** WP2 OSA |
| **Timer Facet** | **Contributor(s):** TUBS, ULANC | **Status:** Works, proof of concept | **Target:** WP2 OSA |
| **Debug/Logging Facet** | **Contributor(s):** TUBS, ULANC | **Status:** Works, proof of concept | **Target:** WP2 OSA |

### 2.2 Contiki

| **OS Facet** | **Contributor(s):** TUBS | **Status:** Fully supported | **Target:** Contiki |
| **Radio Facet** | **Contributor(s):** TUBS, UZL | **Status:** Fully supported | **Target:** Contiki |
| **Extended Data Radio Facet** | **Contributor(s):** TUBS, UZL | **Status:** Fully supported | **Target:** Contiki |
| **Timer Facet** | **Contributor(s):** TUBS | **Status:** Fully supported | **Target:** Contiki |
| **Debug/Logging Facet** | **Contributor(s):** TUBS | **Status:** Fully supported | **Target:** Contiki |
| **Clock Facet** | **Contributor(s):** TUBS | **Status:** Fully supported | **Target:** Contiki |
| **Serial Communication Facet** | **Contributor(s):** TUBS | **Status:** Fully supported | **Target:** Contiki |
2.3 TinyOS

**OS Facet**
- **Contributor(s):** TUBS, TUD
- **Status:** Fully supported
- **Target:** TinyOS

**Radio Facet**
- **Contributor(s):** TUBS, TUD, UZL
- **Status:** Fully supported
- **Target:** TinyOS

**Extended Data Radio Facet**
- **Contributor(s):** TUBS, TUD, UZL
- **Status:** Fully supported
- **Target:** TinyOS

**Timer Facet**
- **Contributor(s):** TUBS, TUD
- **Status:** Fully supported
- **Target:** TinyOS

**Debug/Logging Facet**
- **Contributor(s):** TUBS
- **Status:** Fully supported
- **Target:** TinyOS

**Clock Facet**
- **Contributor(s):** TUBS, UZL
- **Status:** Fully supported
- **Target:** TinyOS

**Serial Communication Facet**
- **Contributor(s):** TUBS
- **Status:** Fully supported
- **Target:** TinyOS

2.4 iSense

**OS Facet**
- **Contributor(s):** RACTI, TUBS, UPC, UZL
- **Status:** Fully supported
- **Target:** iSense

**Radio Facet**
- **Contributor(s):** RACTI, TUBS, UPC, UZL
- **Status:** Fully supported
- **Target:** iSense

**Extended Data Radio Facet**
<table>
<thead>
<tr>
<th>Facet</th>
<th>Contributor(s)</th>
<th>Status</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Transmission Power Radio Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Timer Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Debug/Logging Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Clock Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Settable Clock Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Serial Communication Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
<tr>
<td>Random Facet</td>
<td>RACTI, TUBS, UPC, UZL</td>
<td>Fully supported</td>
<td>iSense</td>
</tr>
</tbody>
</table>

### 2.5 ScatterWeb2

<table>
<thead>
<tr>
<th>Facet</th>
<th>Contributor(s)</th>
<th>Status</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Facet</td>
<td>TUBS</td>
<td>Fully supported</td>
<td>ScatterWeb2</td>
</tr>
<tr>
<td>Radio Facet</td>
<td>TUBS</td>
<td>Works, proof of concept</td>
<td>ScatterWeb2</td>
</tr>
<tr>
<td>Timer Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.6 Shawn

OS Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Radio Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Extended Data Radio Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Variable Transmission Power Radio Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Timer Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Debug/Logging Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Clock Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Serial Communication Facet
Contributor(s): RACTI, TUBS, UPC
Status: Fully supported
Target: Shawn

Random Facet
2.7 Linux

OS Facet
- **Contributor(s):** RACTI, TUBS, UPC
- **Status:** Fully supported
- **Target:** Linux

Radio Facet
- **Contributor(s):** Contribution: FRONTS
- **Status:** Fully supported
- **Target:** Linux

Timer Facet
- **Contributor(s):** Contribution: FRONTS
- **Status:** Fully supported
- **Target:** Linux

Debug/Logging Facet
- **Contributor(s):** Contribution: FRONTS
- **Status:** Fully supported
- **Target:** Linux

Clock Facet
- **Contributor(s):** Contribution: FRONTS
- **Status:** Fully supported
- **Target:** Linux

Random Facet
- **Contributor(s):** Contribution: FRONTS
- **Status:** Fully supported
- **Target:** Linux

2.8 Feuerware

OS Facet
- **Contributor(s):** FUB, TUBS
- **Status:** Fully supported
- **Target:** Feuerware

Radio Facet
- **Contributor(s):** FUB, TUBS
- **Status:** Fully supported
- **Target:** Feuerware

Timer Facet
2.9 TriSOS

OS Facet
- **Contributor(s):** Contribution: G-Lab
- **Status:** Fully supported
- **Target:** TriSOS

Radio Facet
- **Contributor(s):** Contribution: G-Lab
- **Status:** Fully supported
- **Target:** TriSOS

Timer Facet
- **Contributor(s):** Contribution: G-Lab
- **Status:** Fully supported
- **Target:** TriSOS

Debug/Logging Facet
- **Contributor(s):** Contribution: G-Lab
- **Status:** Fully supported
- **Target:** TriSOS

Clock Facet
- **Contributor(s):** Contribution: G-Lab
- **Status:** Fully supported
- **Target:** TriSOS

2.10 iOS

OS Facet
- **Contributor(s):** TUBS
- **Status:** Fully supported
- **Target:** iOS

Radio Facet
- **Contributor(s):** TUBS
- **Status:** Fully supported
- **Target:** iOS

Timer Facet
2.11 Android

OS Facet
Contributor(s): RACTI
Status: Fully supported
Target: Android

Radio Facet
Contributor(s): RACTI
Status: Fully supported
Target: Android

Timer Facet
Contributor(s): RACTI
Status: Fully supported
Target: Android

Debug/Logging Facet
Contributor(s): RACTI
Status: Fully supported
Target: Android
3 Library of Algorithms for Wireless Sensor Networks

This section describes the current state of algorithm implementations in the Wiselib. After three years we have 12 algorithm categories:

1. routing algorithms,
2. clustering algorithms,
3. time synchronization protocols,
4. MAC layer protocols,
5. localization algorithms,
6. energy saving schemes,
7. security,
8. graph algorithms,
9. target tracking,
10. data dissemination,
11. neighborhood discovery, and
12. data collection.

In the following sections, we report on the individual algorithms. For each, we describe

- Whether it is the implementation of original WISEBED research (“WISEBED algorithm”), or an algorithm from the literature (“Implementation”). In addition, some algorithms were also contributed by other EU-Projects (“Contribution”).
- The algorithm category. For some, a sub-category is listed (for example, “Security / Cryptographic Algorithms” under Security).
- The distribution. For algorithms in Incubation, we also describe the platform for which they were developed.
- The release status. Algorithms that are available to the general public are marked “Public”. Some algorithms are available only for WISEBED partners, these are marked “Restricted”. This is usually done when publications need to be accepted at conferences or journals before the algorithm can be made public, to prevent third parties from accessing unpublished research.
- The WISEBED partner(s) that contributed the algorithm to the Wiselib.
3.1 Routing algorithms (20)

Destination-Sequenced Distance-Vector Routing (DSDV) (Implementation)
Category: Routing
Wiselib Distribution: Stable
Release Status: Public
Contributed by: TUBS
References: [42]

Dynamic Source Routing (DSR) (Implementation)
Category: Routing
Wiselib Distribution: Stable
Release Status: Public
Contributed by: TUBS
References: [29]

Tree Routing (Implementation)
Category: Routing
Wiselib Distribution: Stable
Release Status: Public
Contributed by: TUBS
References: [46]

Flooding (Implementation)
Category: Routing
Wiselib Distribution: Stable
Release Status: Public
Contributed by: TUBS
References: [1]

Topology Control Based Routing (WISEBED Algorithm)
Category: Routing
Wiselib Distribution: Testing
Release Status: Public
Contributed by: UPC
References: -
Description: This is a Routing algorithm that establishes routing using any Topology Control algorithm. It exhibits how Wiselib algorithms can be templatized by other algorithms.

TORA (Implementation)
Category: Routing
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: [41]

AODV (Implementation)
Category: Routing
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: [43]
**Static (Contribution)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Public
- **Contributed by:** G-LAB
- **References:** -

**Lazy (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Public
- **Contributed by:** UNIGE
- **References:** -

**Secure Routing (WISEBED Algorithm)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Public
- **Contributed by:** RACTI
- **References:** -

**Description:** Combine any routing algorithm with any security algorithm to setup encrypted message transfer.

**Greedy (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Public
- **Contributed by:** UNIGE
- **References:** [53]

**Optimized Link State Routing (OLSR) (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Restricted
- **Contributed by:** UBERN
- **References:** [14]

**DYnamic Manet On-demand (DYMO) (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Testing
- **Release Status:** Restricted
- **Contributed by:** UBERN
- **References:** [12]

**APSR (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Incubation (Shawn)
- **Release Status:** Public
- **Contributed by:** RACTI
- **References:** [5]

**GPSR (Implementation)**

- **Category:** Routing
- **Wiselib Distribution:** Incubation (Shawn)
- **Release Status:** Public
- **Contributed by:** RACTI
- **References:** [30]
### Face (Implementation)

**Category:** Routing  
**Wiselib Distribution:** Incubation (Shawn)  
**Release Status:** Public  
**Contributed by:** UNIGE  
**References:** [11]

### GFG (Implementation)

**Category:** Routing  
**Wiselib Distribution:** Incubation (Shawn)  
**Release Status:** Public  
**Contributed by:** UNIGE  
**References:** [21]

### GRIC (Implementation)

**Category:** Routing  
**Wiselib Distribution:** Incubation (Shawn)  
**Release Status:** Public  
**Contributed by:** UNIGE  
**References:** [45]

### PAMPA (Implementation)

**Category:** Routing  
**Wiselib Distribution:** Incubation (LorienOS)  
**Release Status:** Public  
**Contributed by:** ULANC  
**References:** -

### EfficientRouting (Implementation)

**Category:** Routing  
**Wiselib Distribution:** Incubation  
**Release Status:** Public  
**Contributed by:** UNIGE  
**References:** [25]

### 3.2 Clustering (12)

### BFS (Implementation)

**Category:** Clustering  
**Wiselib Distribution:** Testing  
**Release Status:** Public  
**Contributed by:** RACTI  
**References:** [36]

### DFS (Implementation)

**Category:** Clustering  
**Wiselib Distribution:** Testing  
**Release Status:** Public  
**Contributed by:** RACTI  
**References:** [50]

### HDL (Implementation)

**Category:** Clustering  
**Wiselib Distribution:** Testing  
**Release Status:** Public  
**Contributed by:** RACTI  
**References:** [5]
<table>
<thead>
<tr>
<th>MinMaxD (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEACH (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Moca (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Highway Clustering (Contribution)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BGU Clustering (Contribution)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LCA (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WCA (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEEN (Implementation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category:</strong></td>
</tr>
<tr>
<td><strong>Wiselib Distribution:</strong></td>
</tr>
<tr>
<td><strong>Release Status:</strong></td>
</tr>
<tr>
<td><strong>Contributed by:</strong></td>
</tr>
<tr>
<td><strong>References:</strong></td>
</tr>
</tbody>
</table>
3.3 Time Synchronization (5)

Flash Mob Organization (WISEBED Algorithm)

Category: Time Synchronization
Wiselib Distribution: Incubation (iSense, Atmel ATmega48)
Release Status: Public

Description: Time synchronization algorithm that allows the organization of an event without sharing global time basis. Any node can initiate an event at any point in time.

LTS (Implementation)

Category: Time Synchronization
Wiselib Distribution: Testing
Release Status: Public

References: [51]

TPSN (Implementation)

Category: Time Synchronization
Wiselib Distribution: Testing
Release Status: Public

References: [22]

HRTS (Implementation)

Category: Time Synchronization
Wiselib Distribution: Testing
Release Status: Public

References: [15]

RBS (Implementation)

Category: Time Synchronization
Wiselib Distribution: Testing
Release Status: Public

References: [18]
3.4 MAC Layer (5)

**Wisebed Virtual Link support (WISEBED Algorithm)**

*Category:* MAC Layer  
*Wiselib Distribution:* Testing  
*Contributed by:* RACTI, TUBS, ULANC, UZL  
*Release Status:* Public  
*References:* [7]

*Description:* Enable message transfer between two nodes which are not able to communicate directly over their radio. Instead, messages are sent over connected PCs and then injected to the application, which can not differentiate between these virtual links and real physical ones.

**S-MAC (Implementation)**

*Category:* MAC Layer  
*Wiselib Distribution:* Incubation (ScatterWeb)  
*Contributed by:* UBERN  
*Release Status:* Restricted  
*References:* [55]

**WiseMAC (Implementation)**

*Category:* MAC Layer  
*Wiselib Distribution:* Incubation (ScatterWeb)  
*Contributed by:* UBERN  
*Release Status:* Restricted  
*References:* [17]

**MaxMAC (WISEBED Algorithm)**

*Category:* MAC Layer  
*Wiselib Distribution:* Incubation (ScatterWeb)  
*Contributed by:* UBERN  
*Release Status:* Restricted  
*References:* [27]

*Description:* MaxMAC, is an Energy-Efficient Medium Access Control protocol (recently published at EWSN 2010) that targets at achieving maximal adaptivity with respect to throughput and latency. By adaptively tuning essential parameters at run-time, the protocol reaches the throughput and latency of energy-unconstrained CSMA, while still exhibiting a high energy-efficiency in periods of sparse traffic.

**AREA-MAC (Implementation)**

*Category:* MAC Layer  
*Wiselib Distribution:* Incubation (Feuerware)  
*Contributed by:* FUB  
*Release Status:* Restricted  
*References:* -
# 3.5 Localization (8)

## Triangulation (WISEBED Algorithm)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>TUBS</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>

**Description:** Localization algorithm based on geometrical triangulations. There are no Euclidean coordinates used—each node belongs to one or more triangles.

## Ad-hoc Positioning (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>TUBS, UZL</td>
</tr>
<tr>
<td>References</td>
<td>[39, 31]</td>
</tr>
</tbody>
</table>

## N-hop Multilateration (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>TUBS, UZL</td>
</tr>
<tr>
<td>References</td>
<td>[49, 31]</td>
</tr>
</tbody>
</table>

## Robust Positioning (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>TUBS, UZL</td>
</tr>
<tr>
<td>References</td>
<td>[48, 31]</td>
</tr>
</tbody>
</table>

## GPS-free Positioning (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>TUBS</td>
</tr>
<tr>
<td>References</td>
<td>[52]</td>
</tr>
</tbody>
</table>

## Greedy Localization (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>UNIGE</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>

## Emission Inhibition Localization (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Incubation (Shawn)</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>UNIGE</td>
</tr>
<tr>
<td>References</td>
<td>[44]</td>
</tr>
</tbody>
</table>
Virtual Raw Anchor Coordinates (Implementation)

Category: Localization
Wiselib Distribution: Incubation
Release Status: Public
Contributed by: UNIGE
References: [25]

3.6 Energy Saving Schemes (6)

LMST (Implementation)

Category: Energy Saving Schemes / Topology Control
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UPC
References: [35]

FLSS (Implementation)

Category: Energy Saving Schemes / Topology Control
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UPC
References: [34]

K-NEIGH (Implementation)

Category: Energy Saving Schemes / Topology Control
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UPC
References: [10]

XTC (Implementation)

Category: Energy Saving Schemes / Topology Control
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UPC
References: [54]

CBTC (Implementation)

Category: Energy Saving Schemes / Topology Control
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UPC
References: [33]

Duty Cycling based on Ant Behavior (Implementation)

Category: Energy Saving Schemes / Duty Cycling
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: TUBS, UPC
References: [24]
3.7 Security (9)

AES (Implementation)
Category: Security / Cryptographic Algorithms
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: [20]

ECIES (Implementation)
Category: Security / Cryptographic Algorithms
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: [20]

SHA1 (Implementation)
Category: Security / Cryptographic Algorithms
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: [20]

HARPS (Implementation)
Category: Security / Cryptographic Algorithms
Wiselib Distribution: Testing
Release Status: Public
Contributed by: UZL
References: -

SecureHDL (Implementation)
Category: Security / Group Key Establishment
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: -

SecureDFS (Implementation)
Category: Security / Group Key Establishment
Wiselib Distribution: Testing
Release Status: Public
Contributed by: RACTI
References: -

FRONTS GKE (Contribution)
Category: Security / Group Key Establishment
Wiselib Distribution: Testing
Release Status: Public
Contributed by: FRONTS
References: -

Diffie-Hellman Key-Exchange Algorithm (Implementation)
Category: Security / Cryptographic Algorithms
Wiselib Distribution: Testing
Release Status: Restricted
Contributed by: UBERN
References: [47]
### Eschenauer-Gligor Key-Management algorithm (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Security / Cryptographic Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Restricted</td>
</tr>
<tr>
<td>Contributed by</td>
<td>UBERN</td>
</tr>
<tr>
<td>References</td>
<td>[19]</td>
</tr>
</tbody>
</table>

### 3.8 Graph Algorithms (7)

#### DDFS (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Fundamental Graph Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>UPC</td>
</tr>
<tr>
<td>References</td>
<td>[4]</td>
</tr>
</tbody>
</table>

#### DBFS (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Fundamental Graph Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>UPC</td>
</tr>
<tr>
<td>References</td>
<td>[56]</td>
</tr>
</tbody>
</table>

#### TwoHop (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>RACTI</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Judged (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>RACTI</td>
</tr>
<tr>
<td>References</td>
<td>[40]</td>
</tr>
</tbody>
</table>

#### MultiJudged (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>RACTI</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>

#### ParMultiJudged (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>RACTI</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Rand Coloring (Implementation)

<table>
<thead>
<tr>
<th>Category</th>
<th>Graph Algorithms / Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiselib Distribution</td>
<td>Testing</td>
</tr>
<tr>
<td>Release Status</td>
<td>Public</td>
</tr>
<tr>
<td>Contributed by</td>
<td>RACTI</td>
</tr>
<tr>
<td>References</td>
<td>-</td>
</tr>
</tbody>
</table>
3.9 Target Tracking (2)

Passive and Lightweight Target Tracking (Implementation)

Category: Target Tracking
Wiselib Distribution: Testing
Release Status: Public
Contributed by: UNIGE
References: [38]

Private Tracking (Contribution)

Category: Target Tracking
Wiselib Distribution: Testing
Release Status: Public
Contributed by: FRONTS
References: [38]

3.10 Data Dissemination (2)

Code Distribution (Implementation)

Category: Data Dissemination
Wiselib Distribution: Incubation
Release Status: Public
Contributed by: ULANC
References: [26]

Data Propagation with Guaranteed Delivery for Mobile Networks (Implementation)

Category: Data Dissemination
Wiselib Distribution: Incubation
Release Status: Public
Contributed by: UNIGE
References: [3]

3.11 Neighborhood Discovery (1)

Echo (Contribution)

Category: Neighborhood Discovery
Wiselib Distribution: Testing
Release Status: Public
Contributed by: FRONTS
References: -

3.12 Data Collection (1)

SWAT Service (Contribution)

Category: Data
Wiselib Distribution: Testing
Release Status: Public
Contributed by: G-Lab
References: -
3.13 Summary

Finally, we summarize the various contributions from the WISEBED partners. Table 1 shows the algorithm counts for the 12 categories. The contributions per site are shown in Table 2—note that due to collaboration in algorithm development the total amount of contributions is greater than the number of algorithms. Finally, Table shows the size of the Wiselib distributions (Stable, Testing, Incubation).

<table>
<thead>
<tr>
<th>Category</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing</td>
<td>20</td>
</tr>
<tr>
<td>Clustering</td>
<td>12</td>
</tr>
<tr>
<td>Time Synchronization</td>
<td>5</td>
</tr>
<tr>
<td>MAC Layer</td>
<td>5</td>
</tr>
<tr>
<td>Localization</td>
<td>8</td>
</tr>
<tr>
<td>Energy Saving Schemes</td>
<td>6</td>
</tr>
<tr>
<td>Security</td>
<td>9</td>
</tr>
<tr>
<td>Graph Algorithms</td>
<td>7</td>
</tr>
<tr>
<td>Target Tracking</td>
<td>2</td>
</tr>
<tr>
<td>Data Dissemination</td>
<td>2</td>
</tr>
<tr>
<td>Neighborhood Discovery</td>
<td>1</td>
</tr>
<tr>
<td>Data Collection</td>
<td>1</td>
</tr>
<tr>
<td><strong>12 Categories</strong></td>
<td><strong>78 Algorithms</strong></td>
</tr>
</tbody>
</table>

Table 1: Total amount of algorithms in the Wiselib.

<table>
<thead>
<tr>
<th>Site</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUB</td>
<td>1</td>
</tr>
<tr>
<td>RACTI</td>
<td>26</td>
</tr>
<tr>
<td>TUBS</td>
<td>12</td>
</tr>
<tr>
<td>UBERN</td>
<td>7</td>
</tr>
<tr>
<td>ULANC</td>
<td>3</td>
</tr>
<tr>
<td>UNIGE</td>
<td>12</td>
</tr>
<tr>
<td>UPC</td>
<td>12</td>
</tr>
<tr>
<td>UZL</td>
<td>5</td>
</tr>
<tr>
<td>External</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2: Contributions by site.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>4</td>
</tr>
<tr>
<td>Testing</td>
<td>56</td>
</tr>
<tr>
<td>Incubation</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 3: Number of algorithms in distributions.
4 Conclusion

During the three years of the Wisebed project, we designed and implemented a generic algorithm library for heterogeneous sensor networks, the Wiselib [8]. In the end of the project, it consists of 78 algorithms, running on 11 software platforms (sensor node operating systems, simulators, mobile devices), involving more than 10 hardware platforms—from tiny 8-bit systems up to powerful nodes such as the iMote2 or mobiles.

All partners contributed code to the Wiselib, by also collaborating in algorithm development and especially stabilization of OS facets. Even more, the Wiselib was used and extended by external partners, such as the EU-project FRONTS. Hence, we expect the Wiselib to be used over the end of the Wisebed project, since there are already matured plans in other projects, e.g. the EU-project SPITFIRE or the German project WSNLAB.
References


