IoTBench - Past, present, and future of a community-driven benchmarking initiative

Romain Jacob and Markus Schüß
on behalf of IoTBench

CPS-IoTBench Workshop
April 15, 2019
How does your algorithm compare to the state-of-the-art?

< insert your research object here >
How does your algorithm compare to the state-of-the-art?
method

< insert your research object here >

low-power wireless protocol

“We need a benchmark for IoT networking.”
The Vision
“Okay, this protocol is really cool. Let’s see how well it perform...”
Send your binary for evaluation

Execute the evaluation automatically
Send your binary for evaluation

Execute the evaluation automatically

Receive your evaluation results
Send your binary for evaluation

Store tested binaries (for future repetitions)

Execute the evaluation automatically

Receive your evaluation results

© 2019 IoTBench | All rights reserved  www.iotbench.ethz.ch
Send your binary for evaluation

Store tested binaries (for future repetitions)

Execute the evaluation automatically

Store evaluation results

Receive your evaluation results

Compare results obtained so far
Send your binary for evaluation

Store tested binaries (for future repetitions)

Execute the evaluation automatically

Store evaluation results

Receive your evaluation results

Compare results obtained so far
The Problem(s)
Why is it **so difficult** to compare (low-power) wireless protocols?

Many different test settings

![Graph showing inter-packet interval (s) vs. testbed size (# nodes). The graph includes a legend with the note: Periodic data collection only.]
Why is it so difficult to compare (low-power) wireless protocols?

Many different test settings

Experiments are not reproducible
not comparable

Only 16.5% wireless networking papers provide enough info to only attempt to reproduce the results [1]

Why is it so difficult to compare (low-power) wireless protocols?

Many different test settings

Experiments are not reproducible
not comparable

No reference results available
How to address these challenges?

Many different test settings

Experiments are not reproducible
     not comparable

No reference results available
How to address these challenges?

Many different test settings

Formalized test configurations

Experiments are not reproducible
not comparable

No reference results available
How to address these challenges?

Many different test settings

Experiments are not reproducible
not comparable

No reference results available

Formalized test configurations

Experimental methodology

Definition of repeatability

Comparison methodology
How to address these challenges?

- Many different test settings
- Experiments are not reproducible
- No reference results available
- Formalized test configurations
- Experimental methodology
- Definition of repeatability
- Comparison methodology
- Benchmark problems
- Common experimental infrastructure
Comparison of the SotA

Protocol database

Test Configuration
- Formalized test scenario
- Common experimental infrastructure
- Benchmark problems

Experiments
- Evaluation methodology

Performance results
- Definition of repeatability

Repeatable?
- Yes
- No

Comparison of the SotA
- Comparison methodology
Getting there

The EWSN Dependability Competition
EWSN Dependability Competition
Goal: quantitatively compare the performance of low-power wireless systems
EWSN Dependability Competition

Step 1: define a common test scenario
EWSN Dependability Competition
Step 2: define performance metrics to enable comparison

Do solutions allow a reliable, timely, and energy-efficient communication?

Three evaluation metrics
- Number of messages delivered correctly
- End-to-end latency
- Total power consumption of all nodes
EWSN Dependability Competition

Step 3: define the benchmark problem(s)

Select your parameters e.g. 32 Bytes every 5 Seconds

D-Cube

Web-interface

Execute the evaluation automatically

Testbed Infrastructure

Store evaluation results

Send your binary for evaluation

Receive your evaluation results

Compare results obtained so far

Select your parameters e.g. 32 Bytes every 5 Seconds
EWSN Dependability Competition
Step 3: define the benchmark problem(s)

Select your parameters
e.g. 32 Bytes every 5 Seconds

D-Cube

Web-interface

Send your binary for evaluation

Testbed Infrastructure

Execute the evaluation automatically

Receive your evaluation results

Create Job

Name

Description
Super secret optimization

Duration
600 Seconds

Competition Category
Category 1: Data collection
Node Layout 3

Traffic Load
30000 Milliseconds
64 Bytes

Jamming type
Level 3

Capture serial
On
Binary Patching

Choose File
broken.hex

Close Create
EWSN Dependability Competition

Step 3: define the benchmark problem(s)

Select your parameters e.g. 32 Bytes every 5 Seconds

D-Cube

Web-interface

Send your binary for evaluation

Execute the evaluation automatically

Testbed Infrastructure

Store evaluation results

Receive your evaluation results

Compare results obtained so far

Testbed Infrastructure

Web-interface

Send your binary for evaluation

Execute the evaluation automatically

Store evaluation results

Receive your evaluation results

Compare results obtained so far
EWSN Dependability Competition

Step 3: define the benchmark problem(s)

Select your parameters e.g. 32 Bytes every 5 Seconds

Inject knowledge about the benchmark problem

Execute the evaluation automatically

Store evaluation results

Compare results obtained so far

Send your binary for evaluation

Receive your evaluation results

D-Cube

Web-interface

Testbed Infrastructure

Inject knowledge about the benchmark problem

Testbed Infrastructure

Store evaluation results

Compare results obtained so far

Select your parameters e.g. 32 Bytes every 5 Seconds

Inject knowledge about the benchmark problem

Execute the evaluation automatically

Store evaluation results

Compare results obtained so far

Send your binary for evaluation

Receive your evaluation results
EWSN Dependability Competition

Step 3: define the benchmark problem(s)

Select your parameters
e.g. 32 Bytes every 5

D-Cube

Inject knowledge about
the benchmark problem

Execute the evaluation

typedef struct
{
  uint8_t traffic_pattern;  // 0:unused, 1:p2p, 2:p2mp, 3:mp2p, 4: mp2mp
  uint8_t source_id[TB_NUMNODES];  // Only source_id[0] is used for p2p/p2mp
  uint8_t destination_id[TB_NUMNODES];  // Only destination_id[0] is used for p2p/mp2p
  uint8_t msg_length;  // Message length in bytes in/to EEPROM
  uint8_t msg_offsetH;  // Message offset in bytes in EEPROM (high byte)
  uint8_t msg_offsetL;  // Message offset in bytes in EEPROM (low byte)
  uint32_t periodicity;  // Period in ms (0 indicates aperiodic traffic)
  uint32_t aperiodic_upper_bound;  // Upper bound for aperiodic traffic in ms
  uint32_t aperiodic_lower_bound;  // Lower bound for aperiodic traffic in ms
} pattern_t;

Compare results
obtained so far
EWSN Dependability Competition
Step 3: define the benchmark problem(s)

Select your parameters e.g. 32 Bytes every 5 Seconds

D-Cube
Web-interface
Send your binary for evaluation
Receive your evaluation results

Inject knowledge about the benchmark problem
Execute the evaluation automatically

Testbed Infrastructure
Store evaluation results

Compare results obtained so far
D-Cube
Low-cost Benchmarking infrastructure

The physical component of the infrastructure

Build on top of off-the-shelf hardware
Raspberry Pi + open-source addon PCB

Focus on easy deployment

~ 100€ for the box, excluding the node
D-Cube
Low-cost Benchmarking infrastructure

Parameters can be modified on a per experiment basis (fully automated)

Supports the generation of stimuli
Unix-style single purpose applications
D-Cube
Low-cost Benchmarking infrastructure

Parameters can be modified on a per experiment basis (fully automated)

Supports the generation of stimuli
Unix-style single purpose applications
D-Cube
Low-cost Benchmarking infrastructure

Parameters can be modified on a per experiment basis (fully automated)

Supports the generation of stimuli
Unix-style single purpose applications

Parameters
- Traffic Pattern and Node Identities
- Traffic Load
- System Parameters
- Experiment Parameters
- Environmental Parameters

Firmware

Testbed Infrastructure
D-Cube
Low-cost Benchmarking infrastructure

Parameters can be modified on a per experiment basis (fully automated)

Supports the generation of stimuli
Unix-style single purpose applications
D-Cube
Low-cost Benchmarking infrastructure

Parameters can be modified on a per experiment basis (fully automated)

Supports the generation of stimuli
Unix-style single purpose applications
JamLab-NG
Repeatable interference generation

To compare results, running experiments in an “office” is insufficient

Raspberry Pi used in D-Cube comes with a build-in Wi-Fi card
Firmware is modified to generate interference on the Wi-Fi card itself

Complete control would require a testbed devoid of any RF interference
JamLab-NG
Repeatable interference generation

With ad-hoc Wi-Fi traffic

With JamLab-NG (Confiture)
D-Cube
Low-cost Benchmarking infrastructure

Testbed performs the measurements
- Does not affect the target node
- Real-time monitoring
- Energy, I/O with timestamp, node communication
D-Cube
Low-cost Benchmarking infrastructure

Computation of Performance metrics
- Reduction of dimensionality
- Are computed after completion
- Using python + pandas for easy analysis
- Used for comparison of results
**D-Cube**

Low-cost Benchmarking infrastructure

**Computation of Performance metrics**
- Reduction of dimensionality
- Are computed after completion
- Using python + pandas for easy analysis
- Used for comparison of results

... And who wins?
EWSN Dependability Competition
Who wins the competition?

2019 Data Collection
DeCoT+ (Academia)

2019 Dissemination
BigBangBus (Industry)
EWSN Dependability Competition
Comparison of performance metrics

Leaderboard

- Public version during the preparation
- Final version after evaluation phase
- Comparison for a single combination of parameters
EWSN Dependability Competition
Comparison of performance metrics

Leaderboard

- Public version during the preparation
- Final version after evaluation phase
- Comparison for a single combination of parameters
EWSN Dependability Competition
Comparison of performance metrics

Leaderboard
- Public version during the preparation
- Final version after evaluation phase
- Comparison for a single combination of parameters

https://iti-testbed.tugraz.at/ewsn2019/leaderboard/
EWSN Dependability Competition
Comparison of performance metrics

Heatmap

Comparing the results of individual Benchmark Problems
Insight into overall performance

Legend
From left to right, for each message length: aperiodic, periodic 5s, periodic 30s
Looking ahead

On-going and future work
Comparison of the SotA

Test Configuration
- Formalized test scenario
- Common experimental infrastructure
- Benchmark problems

Experiments
- Evaluation methodology

Performance results
- Definition of repeatability
- Repeatable?
- Yes
- No

Protocol database

Comparison methodology
Test configuration =
Test scenario + Test environment

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Traffic type</th>
<th>Traffic type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period</td>
<td>Period</td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td>Payload</td>
</tr>
<tr>
<td></td>
<td>Number of sources</td>
<td>Number of sources</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Environment</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Platform</td>
</tr>
<tr>
<td></td>
<td>Frequency band</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

tiny.cc/TestConfig
The description framework serves to describe **profiles** and benchmark problems.

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Test Environment</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic type</td>
<td>Number of nodes</td>
<td>Terminating</td>
</tr>
<tr>
<td>Period</td>
<td></td>
<td>Short</td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td>Number of sources</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Traffic type</td>
<td>Platform</td>
<td>20</td>
</tr>
<tr>
<td>Period</td>
<td>Frequency band</td>
<td>TelosB</td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Number of sources</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
The description framework serves to describe **profiles** and benchmark problems.

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Traffic type</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period</td>
<td>Terminating</td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td>Short</td>
</tr>
<tr>
<td></td>
<td>Number of sources</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Environment</th>
<th>Number of nodes</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Platform</td>
<td>TelosB</td>
</tr>
<tr>
<td></td>
<td>Frequency band</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Describe your own setup.
The description framework serves to describe profiles and benchmark problems

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Traffic type</th>
<th>Period</th>
<th>Payload</th>
<th>Number of sources</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Environment</td>
<td>Number of nodes</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Platform</td>
<td>TelosB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Frequency band</td>
<td>2.4 GHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Profile</td>
<td>Terminating</td>
<td>40 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Short</td>
<td>100 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>16 B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benchmark pbm</td>
<td></td>
<td>Graz/Layout3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The description framework serves to describe profiles and benchmark problems.

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Traffic type</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period</td>
<td>Terminating</td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td>Short</td>
</tr>
<tr>
<td></td>
<td>Number of sources</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Environment</th>
<th>Period</th>
<th>Payload</th>
<th>Number of sources</th>
<th>Number of nodes</th>
<th>Platform</th>
<th>Frequency band</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graz_Layout3</td>
<td>40 s</td>
<td>100 ms</td>
<td>16 B</td>
<td>20</td>
<td>TelosB</td>
<td>2.4 GHz</td>
</tr>
</tbody>
</table>

Benchmark pbm

Exact setup (public)
Towards a Methodology for Experimental Evaluation in Low-Power Wireless Networking

Includes material from Hanspeter Schmid and Alex Huber
Comparison of the SotA

Formalized test scenario
Common experimental infrastructure
Benchmark problems

Protocol database

Experiments
Evaluation methodology

Performance results
Definition of repeatability

Repeatable?
No
Yes

Comparison of the SotA
Comparison methodology
2\textsuperscript{nd} Workshop on Benchmarking Cyber-Physical Networks and Internet of Things (CPS-IoTBench)
April 15, 2019 | Montréal | Canada

![Diagram of the benchmarking process]

- **Test Configuration**
  - Formalized test scenario
  - Common experimental infrastructure
  - Benchmark problems

- **Experiments**
  - Evaluation methodology

- **Performance results**
  - Definition of repeatability

- **Protocol database**

- **Repeatability check**
  - Yes
  - No

- **Comparison of the SotA**
  - Comparison methodology

---

© 2019 IoTBench | All rights reserved
Comparison of the SotA

Formalized test scenario

Common experimental infrastructure

Benchmark problems

Protocol database

Experiments

Evaluation methodology

Performance results

Definition of repeatability

Repeatable?

Yes

No

Comparison of the SotA

Comparison methodology
 Evaluation methodology

Definition of repeatability

Repeatable? Yes

Yes

No

Comparison of the SotA

Comparison methodology
Send your binary for evaluation

Store tested binaries (for future repetitions)

Execute the evaluation automatically

Store evaluation results

Receive your evaluation results

Compare results obtained so far
We need:

- Standardized metrics for the evaluations
- Standardized test scenarios
- Central repository
- Common interface to the different testbeds
- Test environments

Benchmarks:

- Scenarios description
- Protocols binary
- Evaluation results

Testbeds:

- Simulators
We need **You!**
IoTBench - Past, present, and future of a community-driven benchmarking initiative

Join us and Get involved!

www.iotbench.ethz.ch

@iot_bench

Icons from thenounproject.com