Evaluating Bluetooth Low Energy for IoT

Jonathan Fürst*, Kaifei Chen†, Hyung-Sin Kim†, Philippe Bonnet*
IT University of Copenhagen*, UC Berkeley†

CPSBench, Porto
April 10, 2018
jonf@itu.dk
IoT applications rely on BLE for **local, energy-efficient** data exchange between **smartphones** and **resource constrained peripherals**.

- Most **smartphones** are **equipped** with BLE
- Great **API support** on **Android** and **iOS**.

→ Sets BLE apart from other low power wireless technologies such as **ZigBee** or **Thread**.
Example: Smart Light Actuation Time

Key Requirement: Small latency of advertisements.
Example: Smart Light Actuation Time

BLE Advertisement Latency. Advertising interval is set to 1280 ms while smartphones scan in the default balanced mode. Smartphones are placed in 2 m distance. We perform 20 repetitions.
1. Introduction
2. BLE Hardware and Software Abstractions
3. Opening the Black Box
4. Conclusion and Open Questions
BLE Hardware and Software Abstractions
Tracing BLE down the **Software and Hardware Abstractions.**

Smartphone OS (Android)

- **Application Framework APIs**
- **Main Bluetooth Process packages/apps/Bluetooth**
- **Hardware Abstraction Layer (HAL)**
- **Bluetooth Driver**

**Binder IPC**

**Bluetooth, WiFi Signals**

**Casing** ↔ **Antenna** ↔ **Frontend (amplifier, directional coupler)** ↔ **SoC (WiFi, Bluetooth, FM)**

**Hardware Implementation**
Hypothesis (1): BLE is accessed ‘indirectly’, through several (partly hidden) abstraction layers and proprietary drivers, which leads to unpredictable BLE behavior across smartphone models with different OS versions.
Tracing BLE down the **Software** and **Hardware** Abstractions.

Smartphone OS (Android)

- Application Framework APIs: `android.bluetooth`
- Binder IPC
- Main Bluetooth Process: `packages/apps/Bluetooth`
- Java Native Interface (JNI)
- Hardware Abstraction Layer (HAL)
- Bluetooth Driver

---

**Bluetooth, WiFi Signals**

- Casing
- Antenna
- Frontend (amplifier, directional coupler)
- SoC (WiFi, Bluetooth, FM)

**Hardware Implementation**
Hypotheses

**Hypothesis (1):** BLE is accessed ‘indirectly’, through several (partly hidden) abstraction layers and proprietary drivers, which leads to unpredictable BLE behavior across smartphone models with different OS versions.

**Hypothesis (2):** SoC and BLE chip implementations are two major factors that lead to different BLE performance.

**Hypothesis (3):** Hardware components, such as amplifier, antenna, and cover, impact BLE performance.
Opening the Black Box
Opening the Black Box

BLEva Overview
Opening the Black Box

Experimental Setup

- 10 different Android models from 2013 to 2016.
- Factory state before experimentation.
- Dedicated WiFi network for communication.

<table>
<thead>
<tr>
<th>BLE Chip</th>
<th>Phone Models and API Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCN3620</td>
<td>Motorola Moto G (API 22), Moto E2 (API 23)</td>
</tr>
<tr>
<td>WCN3660</td>
<td>LG Nexus 4 (API 22), Asus Nexus 7 (2013) (API 23)</td>
</tr>
<tr>
<td>BCM4339</td>
<td>LG Nexus 5 (API 23), LG G3 (API 21)</td>
</tr>
<tr>
<td>BCM4356</td>
<td>Motorola Nexus 6 (API 24), HTC One M9 (API 23)</td>
</tr>
<tr>
<td>QCA6174</td>
<td>LG Nexus 5X (API 25)</td>
</tr>
<tr>
<td>BCM4330</td>
<td>Samsung Galaxy S3 (API 19)</td>
</tr>
</tbody>
</table>
Advertising Packet Reception Ratio (PRR) for the Same Smartphone on Various OS Versions

→ OS versions (and BLE driver implementations) significantly impact BLE performance.
Impact of OS/SoC Implementations

---

**Advertisement PRR** of Phone Models:

→ SoC significantly impacts PRR. Phones with the same SoC exhibit similar performance, except for LG G3 because of highly customized OS.
**Advertisement Latency** of Phone Models:

→ Advertising latency varies with both SoC vendor and SoC model of a same vendor.
Distance vs. Mean RSSI of Phone Models:

→ The impact of the smartphone model nullifies the impact of distance on RSSI in many cases.
Conclusion and Open Questions
Conclusion and Open Questions

Conclusion

**Hypothesis (1):** BLE is accessed ‘indirectly’, through several (partly hidden) abstraction layers and proprietary drivers, which leads to unpredictable BLE behavior across smartphone models with different OS versions.

**Hypothesis (2):** SoC and BLE chip implementations are two major factors that lead to different BLE performance.

**Hypothesis (3):** Hardware components, such as amplifier, antenna, and cover, impact BLE performance.

→ **Experiments support the two first hypotheses.**
Open Questions

- What should be the API abstraction level on Android/iOS?
- Impact on design and implementation of embedded BLE systems?
- What does it take for applications to become “BLE- conscious”?

BLEva Repository: [https://github.com/jf87/BLEva](https://github.com/jf87/BLEva)

Thank you!
Additional Slides
Write Latency of Phone Models:

→ Write latency is SoC-bound, but varies significantly across SoCs.
Mean RSSI value at each of four measurement times
Opening the Black Box
Impact of OS/SoC/Hardware

Every RSSI sample value during the **whole experiment period**.