A. Artifact Appendix

A.1 Abstract

This artifact shows the main software section of BRIDGE, a solution that leverages an additional trigger node (referred to as TRIGGER) to make the direction finding feature compatible with all Bluetooth devices. We provide our program to control the TRIGGER, as well as the process of receiving the corresponding IQ data.

A.2 Artifact check-list (meta-information)

- **Algorithm: We present our synchronization process via nesting packet and its corresponding packet flow implementation.**
- **Program: A GNU Radio program controlling the TRIGGER is provided.**
- **Run-time environment: Our codes are implemented and tested on Ubuntu 20.04.**
- **Hardware: Our codes are tested on a 4-core CPU laptop. One USRP N210 and locators are also required.**
- **Execution: Around 15 minutes for a complete data sampling period.**
- **Output: For a successful localization data collection, corresponding IQ samples will be generated from the locators after implementing the BRIDGE program.**
- **Experiments: Please set up the experiment environment based on the requirements in Installation and set up the locators appropriately.**
- **How much disk space required (approximately)?: This artifact is less than 1GB in total (including the deployment of locators).**
- **How much time is needed to prepare workflow (approximately)?: The time for preparing the workflow should be less than 1 hour.**
- **How much time is needed to complete experiments (approximately)?: The full experiments should be completed in less than an hour for one target.**
- **Publicly available?: No**

A.3 Description

A.3.1 How to access

The whole codes can be downloaded from GitHub, which also describes the preparation for locators:
https://github.com/Zrt98/Bridge.git

A.3.2 Hardware dependencies

- CPU: No explicit requirements, but should be able to run the required software.
- RAM: At least 4 GB
- GPU: No explicit requirements.
- Transceiver: Software-Defined Radios (SDR) with a full duplex. We use a USRP N210.

A.3.3 Software dependencies

Linux (Ubuntu) based operating system that is able to run GNU Radio. We use Ubuntu 20.04 in our experiments.

Requirements:

The detailed package requirements are described on our github repository.

We recommend to use UHD on Linux with a higher version than 3.10.0.0.

A.4 Installation

Step 1. Create the UHD environment.

The detailed process to build and install the UHD and GNU Radio on Linux can be referred to the official guidance:
https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux

Step 2. Install our BRIDGE program.

```
mkdir build
cd build
cmake ..
make
```

Step 3. Deploy and prepare for the locators.

The locators can be established based on the guide:
https://www.silabs.com/documents/public/quick-start-guides/qsg175-direction-finding-solution-quick-start-guide.pdf

Detailed process can be followed on our GitHub site.

A.5 Experiment workflow

A.5.1 Target Identification

To run the following BRIDGE program, the MAC address of the target is needed in advance. For devices with a fixed MAC address (e.g., “AABBCCDDEEFF”), just note its MAC address for later use.

For BLE devices with randomized address, the device name can be used to detect the current MAC address with Bluetooth apps (e.g., Simplicity Connect Mobile App on Android). After detecting an advertising packet with the specific device name, keep the current MAC address (will last for around 15 minutes before next randomization).

A.5.2 BRIDGE implementation

For target with a fixed MAC address simply run

```
cd build
./fix.sh FFEEDDCCBBAA
```

For devices with a randomized MAC address, run

```
cd build
./random.sh FFEEDDCCBBAA
```

keep in mind to update the latest MAC address (every 10-15 minutes) and re-run “random.sh” under a long-time data sampling.

A.6 Evaluation and expected results

The expected output is to successfully receive and generate a series of IQ samples from the locator. The MQTT explorer can be used to monitor the IQ sampling process. Example results are also provided on our Github page.

A.7 Notes

A.8 Methodology

Submission, reviewing and badging methodology:

- https://www.acm.org/publications/policies/artifact-review-an
- https://cTuning.org/ae