Demo: Cross-Technology Broadcast Communication between Off-The-Shelf Wi-Fi, BLE, and IEEE 802.15.4 Devices

Hannah Brunner[†], Rainer Hofmann[†], Markus Schuß[†], Jakob Link[‡], Matthias Hollick[‡],
Carlo Alberto Boano[†], and Kay Römer[†]

[†]Institute of Technical Informatics, Graz University of Technology, Austria
[‡]Secure Mobile Networking Lab, Darmstadt University of Technology, Germany

hannah.brunner@student.tugraz.at, {rainer.hofmann, markus.schuss, cboano, roemer}@tugraz.at
{jlink,mhollick}@seemoo.tu-darmstadt.de

Abstract

In this demo, we showcase a generic cross-technology-communication scheme allowing bidirectional communication between off-the-shelf IoT devices operating in the 2.4 GHz band. In particular, we make use of and extend the X-Burst framework to enable communication between devices embedding a Wi-Fi, BLE, or IEEE 802.15.4 radio. X-Burst encodes data in the duration of energy bursts by transmitting legitimate frames with different payload length. Devices with incompatible physical layer, but operating on overlapping channels, can detect the energy bursts and decode information by sampling the received signal strength at a high frequency. As the transmission of frames with variable size and energy detection are features available in most off-the-shelf IoT devices, X-Burst is not technology-specific and allows to *broadcast* cross-technology frames to multiple devices using diverse technologies *simultaneously*.

1 Motivation

Cross-technology communication (CTC) has recently emerged as a valuable technique to allow a direct interaction between wireless devices with incompatible physical layer (PHY). The ability to communicate *directly* with nearby appliances allows wireless devices to autonomously coordinate frequency usage and minimize cross-technology interference, as well as to synchronize their clocks without the need of expensive and inflexible gateways. CTC is therefore increasingly attractive, given the heterogeneity of IoT devices and technologies, as well as the growing congestion of the RF spectrum. The latter is particularly severe in the 2.4 GHz license-free band, crowded, among others, by a plethora of BLE, IEEE 802.15.4, and Wi-Fi devices.

Existing work on CTC has mostly focused on demonstrating that a data exchange across various technologies is possible and on achieving a high throughput [3]. However, most of the existing CTC schemes support only unidirectional communication [4], or make use of software-defined radios to enable a cross-technology data exchange [1].

In this demo, we leverage X-Burst [2] to allow a bidirectional exchange between the three most ubiquitous technologies using the 2.4 GHz band. Specifically, we add the Raspberry Pi 3B+ to the platforms supported by X-Burst, thus enabling off-the-shelf Wi-Fi, BLE, and IEEE 802.15.4 devices to *broadcast CTC frames to each other simultaneously*.

2 Principle

X-Burst is a generic CTC framework enabling data exchange between constrained IoT devices with incompatible PHY. X-Burst uses packet-level modulation to transmit and receive CTC frames, i.e., it exploits properties such as the frame duration, the interval between frames, or the energy-level with which they are sent to convey information between heterogeneous devices. CTC frames can be decoded by means of energy detection, i.e., by performing a high-frequency sampling of the received signal strength (RSS). This approach is more generic than PHY emulation and allows a device to broadcast CTC frames to multiple devices employing diverse wireless standards simultaneously [2].

In X-Burst, the implementation of CTC functionality (e.g., the encoding and decoding of symbols, as well as the assembly/disassembly of frames) is separated from platform-specific details using a hardware abstraction layer (HAL), as shown in Fig. 1. This ensures a high portability of the framework: the HAL of each platform simply needs to expose how to (i) generate bursts of different length, (ii) sample the RSS, and (iii) fine-tune the radio's transmission power.

We have previously integrated X-Burst into the Contiki operating system, and supported several off-the-shelf IoT platforms embedding BLE or IEEE 802.15.4 radios, such as the TI CC2650 LaunchPad, Zolertia Firefly, and TelosB nodes [2]. To enable a CTC between these platforms and an off-the-shelf Wi-Fi device, we have now also implemented and ported X-Burst to the popular Raspberry Pi 3B+.

Since the Raspberry Pi 3B+, as most Wi-Fi devices, does not expose support for frame injection and RSS sam-

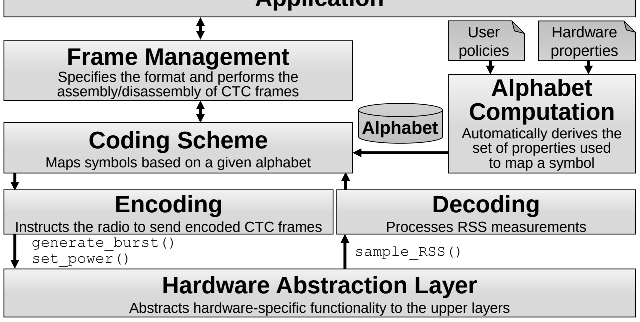
Application
User policies
Hardware properties
Frame Management
Specifies the format and performs the assembly/disassembly of CTC frames
Alphabet Computation
Automatically derives the set of properties used to map a symbol
Alphabet
Coding Scheme
Maps symbols based on a given alphabet
Encoding
Instructs the radio to send encoded CTC frames
Decoding
Processes RSS measurements
generate_burst()
set_power()
sample_RSS()
Hardware Abstraction Layer
Abstracts hardware-specific functionality to the upper layers

Figure 1. X-Burst's architecture (adapted from [2]).

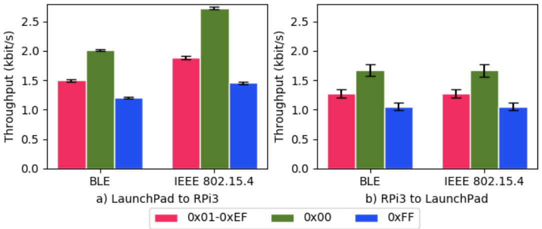

Figure 2. Data throughput between TI CC2650 LaunchPad and Raspberry Pi 3B+ for different payload content.

pling by default, a firmware modification is required. The `BCM43455c0` radio used on the Raspberry Pi 3B+/4B series has recently been reverse-engineered, and it is now possible to replace code down to individual instructions within the firmware and exploit the unused memory to implement new functionality using the Nexmon patching framework [5].

JamLab-NG [6] is built on top of Nexmon and extends its frame injection functionality by disabling the clear channel assessment and by avoiding other sources of entropy such as the operating system's network stack. We use JamLab-NG's `jelly` tool to create energy bursts by injecting frames of custom length at a fixed transmission speed without the need to connect to an access point. We have further extended JamLab-NG's low-level interface from the kernel to the firmware (`ioctls`) to trigger an existing energy detection function within the `BCM43455c0` and return its result to a userland application through the kernel's network stack.

To enable a cross-technology data exchange, we map data symbols into energy bursts of pre-defined duration. In our implementation, we make use of a 2-bit coding scheme and specify four burst durations, namely: 224, 416, 608, and 800 μs. These values are chosen based on the properties of the employed hardware platforms (such as the RSS sampling frequency and time granularity), such that every device performing CTC is able to correctly distinguish two different energy burst durations by means of RSS sampling [2].

3 Demonstration

To showcase our work, we set up a demo as illustrated in Fig. 3. We make use of four off-the-shelf IoT platforms supporting X-Burst, namely: TI CC2650 LaunchPad (BLE), Raspberry Pi 3B+ (Wi-Fi), Zolertia Firefly (IEEE 802.15.4), and TelosB mote (IEEE 802.15.4). Each device is equipped with four LEDs of different colors, where each color is associated to a device (e.g., red→Firefly; green→TelosB). Each device is also equipped with three buttons, two of which allow to turn on/off the LED associated to that specific device by initiating the transmission of a broadcast CTC frame. The third button allows to initiate the transmission of several CTC broadcast frames back-to-back in order to compute the throughput to all nearby devices. All communications are monitored, logged, and displayed using a laptop connected via USB to each device, as well as a PicoScope, so to gain a more detailed insight about X-Burst's encoding process.

In preparation for the demo, we evaluate the throughput experimentally between a Raspberry Pi 3B+ and a TI CC2650 LaunchPad in both BLE and IEEE 802.15.4 mode.

Fig. 2 shows our results: the TI CC2650 LaunchPad and the Raspberry Pi 3B+ can exchange CTC messages at up to 2.5 kbit/s, depending on the payload content (values such as '`0x00`' are encoded in shorter durations than '`0xff`' and hence transmitted faster) as well as the technology used. The relative differences in throughput between the various technologies are due to the different radio preparation time of each platform (i.e., the time elapsed between the transmission of two consecutive frames).

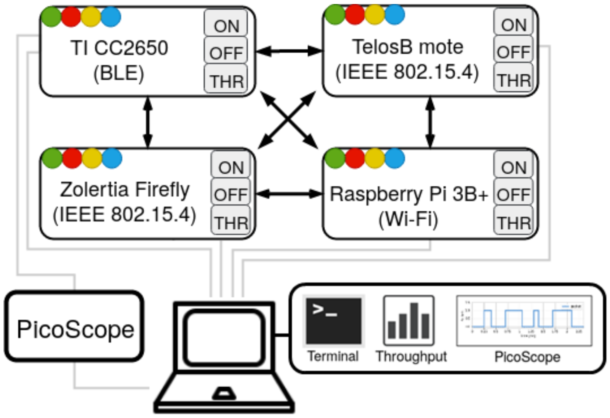

Figure 3. Demonstration setup: several BLE, Wi-Fi, and IEEE 802.15.4 devices broadcast CTC frames to control each other's LEDs. Real-time info about the throughput and the exchanged CTC frames is displayed on a laptop.

Acknowledgments

This work has been performed within the LEAD project "Dependable Internet of Things in Adverse Environments" funded by Graz University of Technology and in the context of the LOEWE centre emergenCITY. This work was partially supported by the FFG-funded Pro^{2}Future COMET center (contract no. 6112792) and by the SCOTT project. SCOTT (`http://www.scott-project.eu`) has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement no. 737422. This joint undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway. SCOTT is also funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between May'17 and April'20.

4 References

[1] Z. Chi et al. B2W2: N-way Concurrent Communication for IoT Devices. In *Proc. of the 14th SenSys Conf.* ACM, 2015.

[2] R. Hofmann, C. A. Boano, and K. Römer. X-Burst: Enabling Multi-Platform Cross-Technology Communication between Constrained IoT Devices. In *Proc. of the 16th SECON Conf.* IEEE, 2019.

[3] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He. BlueBee: a 10,000x Faster Cross-Technology Communication via PHY Emulation. In *Proc. of the 15th SenSys Conf.* ACM, 2017.

[4] Z. Li et al. WEBee: Physical-Layer Cross-Technology Communication via Emulation. In *Proc. of the 23rd MobiCom Conf.* ACM, 2017.

[5] M. Schulz, D. Wegemer, and M. Hollick. The Nexmon Firmware Analysis and Modification Framework: Empowering Researchers to Enhance Wi-Fi Devices. *Computer Communications*, 129(1), 2018.

[6] M. Schuß et al. JamLab-NG: Benchmarking Low-Power Wireless Protocols under Controllable and Repeatable Wi-Fi Interference. In *Proc. of the 16th EWSN Conf.* Junction Publ., 2019.