RFReact: A Real-time Capable and Channel-aware Jamming Platform

Matthias Wilhelm, Ivan Martinovic, Jens B. Schmitt, and Vincent Lenders‡
Disco Labs, TU Kaiserslautern, Germany ‡Armasuisse, Switzerland
{wilhelm,martinovic,jschmitt}@cs.uni-kl.de vincent.lenders@armasuisse.ch

ABSTRACT

In this demo we present RFReact, an USRP2-based jamming platform that enables selective and reactive jamming. Built-in RF analyzers detect signals of interest and trigger transmissions of jamming waveforms. RFReact benefits from the full access to physical layer information due to its software-defined radio implementation: it is agnostic to technology standards and readily adaptable to various applications. Examples of such applications are the controlled and repeatable generation of RF interference to experimentally evaluate the robustness of protocols, the evaluation of reactive jamming strategies and possible countermeasures, and the generation of precisely timed transmissions for experiments. We demonstrate that RFReact is both versatile and precise with a reactive jammer that can demodulate large parts of 802.15.4 packets, decide whether to jam them or not, and execute the decisions while the packets are still on the air.

1. MOTIVATION

Recently, the physical layer of wireless communications and its properties received an increasing interest in the networking community in general, and in the wireless security community in particular. Especially the effects of interference on network performance, both intentional (RF jamming) and unintentional, and possible countermeasures are an active area of research. Experimental evaluation of interfence is especially valuable as channel models are often not capturing the behavior of signal propagation with the required level of accuracy. To assist researchers in this important task, we provide RFReact to the community, a reliable and versatile platform for RF experimentation.

As RFReact is build on top of the widely used software-defined radio (SDR) platform USRP2 it can analyze received signals in software and trigger transmissions of arbitrary waveforms on the medium. In this aspect, it is similar to host-based systems using GNU Radio where a PC is used to analyze and generate RF signals. However, the delay from a signal's presence on the channel to the reaction by the host-based system is too large for time-sensitive applications like reactive jamming. With RFReact, we moved all essential components closer to the hardware to minimize these delays. Our experiments in recent work [1] show that the system reacts fast enough to jam packets currently on the air, with a delay in the order of microseconds, instead of milliseconds as in the host-based architecture. We argue that such short reaction times are essential to achieve repeatable and controllable experiments.

The design of RFReact focuses on flexibility and extensibility. For example, additional RF analyzers can be added to detect a wider range of signals on the wireless medium. In this demo, we show the possibilities of our concept with a full 802.15.4 demodulator. Using this, RFReact is able to selectively jam packets based on header information and payload, e.g., restricting the jamming to ACKs only, to beacons advertising a chosen network or to data frames carrying a specific sender address.

2. APPLICATIONS

By adding custom detectors, RFReact is adaptable to various applications and to different technologies such as IEEE 802.11; we summarize some of our current work and new ideas (with a slight bias on reactive jamming). However, researchers are likely to come up with further applications.

Jamming strategies. The level of channel-awareness offered by RFReact enables the implementation and experimental evaluation of smart jammers. Such a jammer focuses on the destruction of high-value packets to maximize the impact of its attack. With demodulation, this jamming decision can be done while packets are still in transmission. Such experiments help to gauge the effects of such attacks in realistic settings.

Jamming detection. With the availability of smart jammers, the detection of jamming is crucial to initiate countermeasures and to ensure reliable operation in the presence of such jammers. RFReact can make a contribution for the evaluation of jamming detection methods in realistic scenarios and with realistic capabilities of attackers.

Jamming mitigation. The robustness of anti-jamming techniques implemented by hardware devices can be evaluated by experimentation. Our implementation offers a great timing precision, such that the effect of jamming bursts to arbitrary parts of a signal can be quantified in a repeatable way.

Active wireless adversaries. RFReact helps to evaluate injection or message manipulation attacks that require a tight time synchronization with a signal, e.g., to jam a packet and inject a malicious packet of the attacker's choice with minimal time difference or even to overshadow parts of transmissions in order to manipulate the content of a packet without affecting the timing. Along this line, RFReact can help to understand the

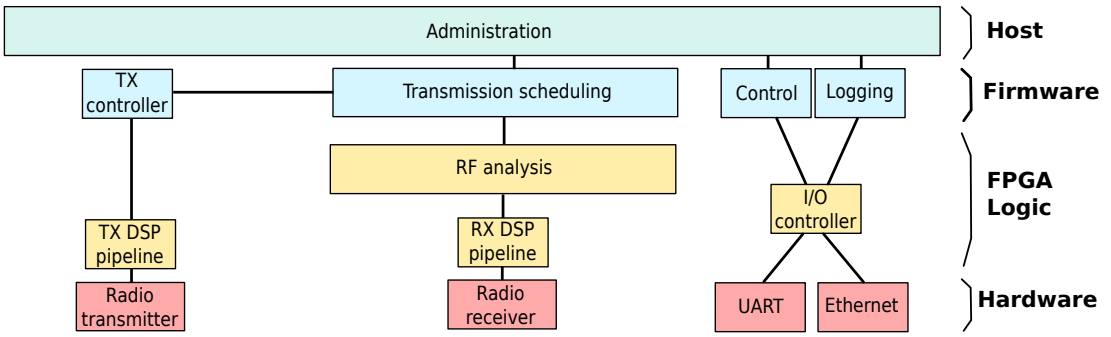

Figure 1: System design of RFReact, showing the functional components and their allocation to the USRP2's resources.

capabilities of realistic attackers, e.g., when considering a wireless Dolev-Yao adversarial model.

Controlled interference. Related to jamming, RFReact can generate arbitrary interference patterns with fine-grained control, without the requirement to change the implementation under test. For example, a scenario with 70 % loss of ACK frames is easily defined and generated at a central point with RFReact, without the need to change the hardware drivers on all devices in the experiment to enforce the experiment's requirements.

The USRP2 is compatible to a large number of daughterboards, making the whole spectrum from DC to 5.8 GHz reachable for experimentation with RFReact.

3. RFREACT'S DESIGN

We implemented RFReact by modifying the vanilla software system that comes with the USRP2. This system provides the sending and receiving functionality of samples, and communication facilities to the host via UDP and UART. It is divided into four tiers: *(i)* the hardware tier with radio chips that sample the RF medium, *(ii)* an FPGA, *(iii)* a soft-microcontroller implemented in FPGA logic and is used to control the device, and *(iv)* a host computer that interconnects via Ethernet to receive and process the samples, as well as for configuration.

RFReact uses several of these tiers on the USRP2 for its components (also refer to Fig. 1):

RF analyzers: The RF analyzers detect features of interest on the medium. They are implemented in FPGA logic as they required high performance and deterministic timing. Thus, the samples are used where they are generated, cutting down the latency. An example of this is our IEEE 802.15.4 demodulator. Several of such analyzer modules can be implemented in parallel to enable signal analysis along several features.

Transmission scheduling. Transmissions can be scheduled in firmware based on time or on channel information gathered by the RF analyzers. Its code runs on the microcontroller, combining a short reaction time and easy development, as the firmware code is written in C. In this module the output of the RF analyzers is processed.

TX controller. The TX waveforms can be chosen arbitrarily. They are either precomputed on the host PC and sent to the USRP2 via UDP, or generated in firmware or in FPGA logic, depending on the desired complexity of the signal and the tolerable delay.

Administration. The control and administration functionality resides on a host PC, which can communicate with the USRP2 either via serial connection or UDP. With this mechanism, the configuration of the scheduling logic can be changed, new waveforms inserted, and statistics gathered during runtime.

3.1 Extending RFReact

In our design, we emphasized the ability to adapt and extend RFReact to different applications. RF analyzers can be added to search for new features of interest on the wireless channel. As input, such an analyzer receives complex samples from the radio frontend to apply their signal processing on. As output, interrupts are used to interact with the firmware for event notification and a bus interface is used to exchange arbitrary data between the analyzer and the rest of the system.

The transmission scheduling module is part of the microcontroller's interrupt controller, and is able to request additional information on a detected event via the bus interface. This information is then used to make scheduling decisions, specifying the behavior of RFReact.

Defining waveforms to be transmitted by the system is a matter of giving a sequence of complex samples that represent the waveform. In this way, modulated data can be transmitted to communicate with the rest of the network.

4. FURTHER INFORMATION

For more information on RFReact and to acquire the necessary resources, please visit `http://disco.cs.uni-kl.de` or contact the main author. We welcome your feedback on the RFReact platform and any information on its application.

5. REFERENCES

[1] Matthias Wilhelm, Ivan Martinovic, Jens B. Schmitt, and Vincent Lenders. Reactive Jamming in Wireless Networks: How Realistic is the Threat? In *Proceedings of the ACM Conference on Wireless Network Security (WiSec 2011)*. ACM, June 2011.