A Novel Software Defined Radio for Practical, Mobile Crowdsourced Spectrum Sensing

Phillip Smith[1,*], Anh Luong[2], Shamik Sarkar[1], Harsimran Singh[1], Aarti Singh[3], Neal Patwari[3], Sneha Kasera[1], Kurt Derr[4]
[1]University of Utah, [2]Sandia National Laboratories, [3]Washington University in St. Louis, [4]Idaho National Laboratory
[*]Corresponding Author: phillip.smith@utah.edu

Abstract—Software defined radios (SDRs) are often used in the experimental evaluation of next-generation wireless technologies. While crowdsourced spectrum monitoring is an important component of future spectrum-agile technologies, there is no clear way to test it in the real world, i.e., with hundreds of users each carrying an SDR while uploading data to a cloud-based controller. Current fully functional SDRs are bulky, with components connected via wires, and last at most hours on a single battery charge. To address these needs, we design and develop a compact, portable, untethered, and inexpensive SDR we call *Sitara*. Our SDR interfaces with a mobile device over Bluetooth 5 and can function standalone or as a client to a central command and control server. It transmits and receives common waveforms, uploads IQ samples or processed receiver data through a mobile device to a server for remote processing and performs spectrum sensing functions. We present results from a user study involving more than 100 participants to evaluate Sitara in a hypothetical large-scale crowdsourced spectrum monitoring application. We also present a comparative analysis of Sitara to related crowdsensing systems with a particular emphasis on the role of incentives and user participation.

Index Terms—Mobile Systems, Spectrum Monitoring, Wireless Networks, Crowdsourcing, Software-Defined Radio

1 Introduction

Future mobile wireless advancements will continue a trend of increasing densification, distribution and coordination, and spectrum-agile operation [32], [33], [37]. The performance of these new technologies depends not only on the mobility of individual users with respect to base stations, but also users' mobility with respect to each other. Ideally, to quantify performance experimentally, one would run a large-scale distributed wireless experiment with tens or hundreds of software-defined radios (SDR) programmed to deploy/test a new technology, while individual volunteers each carry these SDRs with them during their normal daily activities. Such an experiment would allow technologies to be tested with users' real-world mobilities, including temporal, spatial, and person-to-person correlations, rather than in artificial testbed or simulation environments that implicitly or explicitly assume independence and stationarity.

For researchers to be able to run such experiments, the SDR must be *truly portable*, so that a volunteer participant is not burdened by the carrying of the device, and in fact does not ordinarily notice it. Furthermore, the hardware must be *low cost* to enable measurements with hundreds of volunteers. We define portable to mean capable of operating for extended periods of time without an external power supply, small enough to carry without encumbering the user — small enough to easily fit into a pocket — and *not tethered* to wires, cables, or external connectors. Its energy consumption must allow it to last as long as most smartphones so that volunteers can charge it on the same schedule as their mobile device. In terms of cost, a researcher should be able to purchase a set of 100 on a standard grant, which would translate to around US $50 or less per device.

Unfortunately, no existing SDR meets the requirements of performing such experiments. Recent products include devices such as the Kickstarter-funded "portable SDR" (PSDR), RTL-SDR, HackRF, LimeSDR, TinySDR and Ettus USRP radios. While each of these possess useful capabilities, all fall short in one area or another for large scale mobile experiments. The most suitable among these would be the Ettus USRP E312 or TinySDR, both battery-powered portable SDRs [14], [21]. The E312, however, is far too large to fit into a pocket and it must be tethered; it would still require a cable connection to a mobile device or laptop to provide a control channel. Additionally, the steep price of the E312 would present a practical limit to large scale distributed experiments. The TinySDR was developed independently around the same time as Sitara and offers lower idle power consumption but higher for transmit/receive (TX/RX) states, along with higher bandwidth and more flexible TX waveforms. Consequently, its function is purposed more as an internet of things (IoT) endpoint device than a mobile, high duty-cycle crowdsource node [21]. The RTL-SDR is low cost, but requires an external processor, to which it must be connected to by USB cable and, most importantly, has no transmit capability. Most true SDRs, capable of digital processing of radio frequency (RF) samples and RX/TX, cost at least US $100 in the most basic form and quickly reach 10 times that cost for more sophisticated offerings. Such SDRs quickly become cost-prohibitive for large-scale experiments, and present a significant limitation shared by others [9].

Current mobile phones, while packed with cellular, Wi-fi and Bluetooth radios, do not always provide the flexibility to make arbitrary changes across layers which researchers want to explore, despite their suitability for some specific applications in localization and sensing [10], [42], [44]. Differences in chipsets, firmware implementation, protocols and carrier-imposed restrictions preclude uniform or arbitrary access and control of the underlying hardware. Indeed, a recent change to the Android API which limited Wi-fi scan rates reportedly crippled some network tools [5]. Even with unrestricted access to the hardware, such operation could disrupt data services and inconvenience the volunteer — something we wish to avoid.

These convenience factors, incentives and purposes of a crowdsourced system can have a direct impact on adoption rate and, ultimately, the success of such a system. In order to better appreciate these factors, we compare other systems which place different requirements (or burdens) upon users to obtain useful data or provide services.

1.1 A Novel Software-Defined Radio

Our primary contribution is a novel open-source device and cloud framework aimed at enabling large-scale experimental research in mobile dynamic spectrum access, propagation modeling, distributed and coordinated reception, and localization. Our device, called *Sitara*, is a truly portable software-defined radio. It is especially suited for distributed, crowdsourced experiments. It is designed to have a battery life of up to a week on a single charge, to be smaller than a credit card, and to cost less than existing fully-featured SDRs. Sitara is convenient for volunteers to carry and is accessible to a broad set of researchers. We anticipate this to be particularly useful for scenarios in which simultaneous, near real-time, geographically distributed narrow-band RF measurements are desired. We demonstrate how the Sitara can become a valuable tool, quickly amassing measurement inputs for models and providing insights to inform decisions for wireless research.

1.2 Achieving True Portability

The aim of achieving a compact, cordless, energy efficient device constrains key design decisions. The inconvenience of frequent charging and limited space for batteries make the power requirements of field-programmable gate arrays (FPGAs) commonly used in other SDR solutions unfeasible. For a device to be practical for crowdsourcing it must also be convenient for volunteers to carry, which means we cannot connect it via cable to their smartphone, and little to no interaction should be required from the volunteer. With the recent availability of Bluetooth 5 devices and the ubiquity of smartphones, we arrive at the solution presented here: a low power transceiver paired with a Bluetooth interface. By pairing with the volunteer's phone, we can piggyback on the phone's WiFi or cellular connection to communicate with a remote server, as well as its location service.

But how can we avoid the large cost and size of most fully-functional SDRs? We apply a lesson from the RTL-SDR, which re-purposed a mass-produced digital video receiver (the RTL2832U) for its ability to output complex-baseband (IQ) samples. We use the Texas Instruments (TI) CC1200 transceiver which, although not designed as an SDR transceiver, has an IQ sample feature as well as transmit capability. This transceiver supports operation below 1 GHz. For experiments presented here, we optimize the RF chain and limit, in firmware, the frequency of operation to 902-928 MHz in the industrial scientific medical (ISM) band. The transceiver is easily adapted to other sub-1GHz ISM bands. The Sitara complements the CC1200 with a Nordic Semiconductor nRF52840 system on a chip (SoC) and supporting circuitry. This SoC contains a processor (CPU) and Bluetooth stack used for processing commands and communicating with a mobile device/gateway. The Sitara supports reading from and writing to arbitrary registers on the CC1200 radio, tuning radio frequency, measuring RSSI, continuously capturing IQ samples, sample capture on carrier-sense, frequency phase lock, and transmission and reception of messages using various modulations. [1]

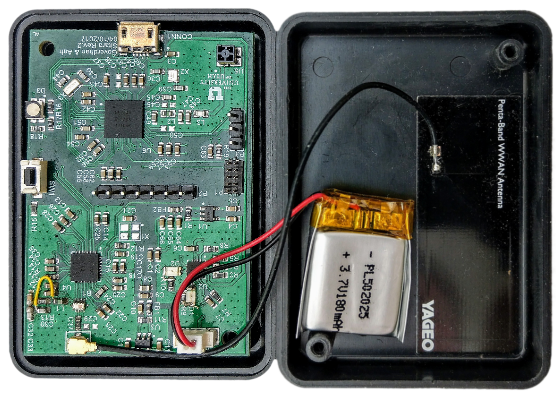

Fig. 1: Sitara PCB with antenna and rechargeable battery housed in an ABS plastic enclosure

1.3 Balancing Power and Throughput

The power budget is always an area of concern on mobile devices. We target low-cost, low power components while introducing the challenge of maintaining high sample throughput on hardware originally designed for intermittent, bursty operation; continuously operating the SoC's CPU alone would deplete our initially specified battery in a matter of hours. We address this problem with an architecture maximizing efficiency by exploiting hardware peripherals to maintain a high data rate while minimizing CPU activity. To prove useful as an SDR, we must maintain a uniform sampling rate for IQ data. This requires solving a number of problems to achieve a tight coordination between transceiver data acquisition, sample processing and the Bluetooth radio.

In general terms, we achieve this by minimizing processing overhead in software and optimizing parameters for Bluetooth transmission. Among the Bluetooth features that make this possible are Low-Energy Data Packet Length Extension introduced in Bluetooth 4.2 and the optional 2 Mb/s bit rate, LE 2M PHY, introduced in Bluetooth 5 [8]. Because the Bluetooth stack is implemented as a "SoftDevice", a

1. Though we do not make use of this functionality in the use cases presented here, Sitara radios can transmit messages between one another using the inherent capabilities of the CC1200 radio.

precompiled binary image, which runs on the single ARM core, there is inherent contention for the CPU's resources. Any timing anomalies occurring while servicing interrupts by the SoftDevice result in a critical fault. Consequently, the SoftDevice must be given interrupt priority, resulting in non-deterministic timing for servicing other interrupts such as sample capture. Our solution overcomes these challenges to provide continuous sample capture over the serial peripheral interface (SPI) and only requires CPU intervention to rotate between receive buffers. The result of our efforts is a maximum, hardware-limited, continuous sampling rate up to 104 kS/s across the SPI interface and Bluetooth data rates exceeding 1Mb/s. The maximum sampling rate across the SPI interface effectively limits our receiver bandwidth to 52 kHz for IQ sampling.

1.4 Cloud-based Command and Control Server

In addition to the Sitara, we develop a mobile application and command and control server interface allowing hundreds of devices to operate in coordination, as shown in Fig. 2. The server provides a convenient web-based GUI for live monitoring and control of connected clients (Fig. 3) and processing of historical measurement data. This allows monitoring real-time measurements in a distributed, mobile environment or delayed logging and upload for later analysis. Accessible records contain RSSI measurements, IQ samples, location, time and device ID. These capabilities enable passive crowdsourced measurements using remote control, or to function as a standalone SDR, controlled wirelessly through a user's mobile device. This is similar to prior efforts [9], but focuses on custom tailoring of experiments for a fine degree of control, rather than optimizing for one specific application.

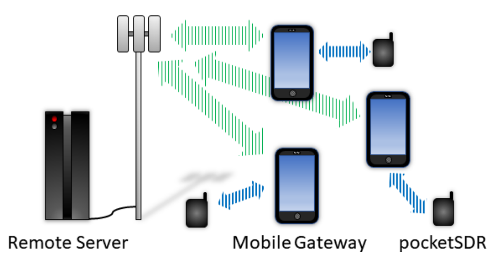

Fig. 2: Sitara backhaul system includes the Bluetooth connection (blue) between the Sitara and mobile gateway, and the WiFi or cellular connection (green) between the mobile device and the remote cloud server.

In this paper we review the design and implementation previously presented [35], walking through solutions to some of the challenges we encountered. We characterize the performance of the Sitara then present experimental results from the following usage scenarios to illustrate the utility of our SDR:

- Transmitter localization using RSSI measurements
- Crowdsourced measurements using multiple concurrent participants, suitable for spectrum monitoring or RF propagation modeling
- Server-side demodulation from IQ sample captures of a 2-FSK transmission

We then discuss related crowdsourced systems and present participant-centered results obtained from surveys and interviews of more than 100 participants to gain an understanding of incentives. We conclude by combining our results in a discussion for future areas of research related to our crowdsourced measurement approach.

2 Design and Implementation

In this section, we examine some of the technical challenges and design decisions during the development of the Sitara, beginning at the hardware component level and then continuing with the firmware development. We also briefly discuss the software development associated with the mobile gateway and server applications.

2.1 Component Choice

In order to develop a low cost, low power solution we look at transceivers capable of RF digital sample output. After considering many options we tended toward wireless transceivers such as the TI CC1200, Atmel RF-233, AT86RF215, Atmel AT86RF215IQ, and Silicon Labs EFR32FG. Among these, interface options and operational frequencies lead us to the CC1200 which tunes to frequency bands between 137 MHz and 950 MHz. The CC1200 is energy efficient and allows raw IQ samples to be exported while still operating over a wide enough frequency range to prove useful. The RF network can be optimized for a target frequency and bandwidth. In this paper, Sitara uses a 915 MHz balun, which we designed for an operating frequency between 902 to 928 MHz. In this hardware configuration, Sitara can operate in lower sub-1GHz ISM bands with minor (three source lines of code) changes to the firmware and reduced gain. To achieve operation in the 2.4GHz range, a different transceiver would be necessary; by choosing a suitable part from the family of related low-power TI transceivers, firmware changes would be minimal.

The CC1200 transceiver operates using a SPI interface to read and write data and control registers, respectively. General-purpose IO (GPIO) connections between the CC1200 and SoC allow interrupt-driven functions such as IQ sample acquisition and RF power level triggering. We adapt the register configurations for optimal spectrum monitoring. The CC1200 provides 3 registers (17 bits total) of magnitude and 2 registers (10 bits total) of angle measurements from the output of its coordinate rotation digital computer (CORDIC) algorithm.

We choose the nRF52840 because it was one of the first available low-power Bluetooth 5 SoCs with a well-supported SDK. Additionally, the ARM Cortex-M4 within the SoC provides a floating point unit (FPU) which is necessary for some SDR applications. The RF output from the nRF52840 is connected to a 2.4 GHz 3dBi SMD chip antenna. While chip antennas are inefficient, the use case is to have a very short Bluetooth link, for example a volunteer might carry both devices in the same handbag, or in two different pockets. Such short links can be reliable even with the antenna loss as we can see from the Bluetooth throughput measurements in section 3.1. A power management IC (PMIC) regulates voltage, charges and manages the LiPo battery connected through the standard JST connector. Sitara contains a JLink interface to allow programming, terminal logging and debugging. The CC1200's RF chain interfaces with an RF-tuned circuit terminating on a μFL connector. For our experiments and the results presented here, we use a Yageo Penta-Band WWAN antenna, but other antennas could also be used. The board, battery, and antenna are designed to fit within a standard 70 by 50 by 20 mm plastic case, which provides mechanical protection while being carried by a volunteer. The battery is recharged by the volunteer using a standard micro USB cable, likely to be familiar to an Android phone user.

At the time of writing, the total cost for the bill of materials (BOM) in quantities of 1000 was estimated to be $38.00 per device. Please refer to our github repository to view the current BOM, source code and design documents [36]. Additional information about our system implementation can be found in arXiv:1905.13172 [cs.NI].

2.2 Sitara Firmware

We develop the SoC firmware using the nRF5 SDK v13.0.0 from Nordic Semiconductor, compiled using the GNU ARM toolchain v7.2.1. The firmware executable code resides in flash on-board the nRF52840 SoC. The SoftDevice, a precompiled protocol stack, is also stored in flash and loaded into RAM at run-time. An event-driven API allows the firmware to interface with the SoftDevice to access Bluetooth functions.

Once powered on, the SoC initializes and configures the external CC1200 transceiver, on-chip Bluetooth radio and other peripherals then enters a sleep state while awaiting commands. As we mentioned, minimizing power consumption is a key design driver, so minimizing the time that components are powered on and active is a recurring theme. This allows us to achieve an 80% power reduction for most applications.

Most commands perform a single function then return the CPU to a sleep state. The continuous SAMPLE CAPTURE command enters a loop in which data is acquired and sent via the Bluetooth interface. Because continuous sample capture is an important aspect of our design we will discuss its operation in more detail.

Sample capture utilizes the Programmable Peripheral Interconnect (PPI), which permits on-chip peripherals to interact through task-event relationships, independent of the CPU. We configure an interrupt event associated with the magnitude-valid output signal from the CC1200 to trigger a burst-read SPI transaction task which reads the registers containing the sample data. The magnitude-valid signal asserts when a new IQ sample is ready on the CC1200 and occurs at a set rate dependent on the configured receiver filter bandwidth. While most of these actions are automated using hardware peripherals, the nRF52840 SoC does require an interrupt and CPU intervention to rotate the pointer for the receive buffer in the SPI-DMA transaction; the time required to service the interrupt is approximately 22 μs. Therefore, sampling rates with a period equal to or greater than 22 μs (45 kS/s) may be delayed by at most 1 sample period. An additional factor that complicates this process and limits the maximum sampling rate over SPI is the absence of a hardware-enabled control function which allows consecutive burst reads from multiple registers (although this is available for repeated reads from a single register). This negatively impacts performance in two ways:

- Each sample read must include two command bytes for SPI burst read which are added to the total length of the transaction, thus increasing each transfer duration.
- During a SPI transfer, a byte is received during each clock cycle, thus the receive buffer will always contain the two status bytes received during the clock cycles which the two command bytes are sent, in addition to the bytes containing the actual data.

This results in inefficient use of memory and a receive buffer containing status bytes interleaved with sample data requiring extra processing steps to extract samples. Nevertheless, this does not impact performance because the Bluetooth connection ultimately limits throughput as we note in Section 3.1.

Bluetooth transfer of packets is not real-time, and due to packet collisions and errors, MAC delays and retransmissions, any finite buffer can experience an overflow. We chose to handle this by pausing capture acquisition and discarding samples while the Bluetooth interface catches up. This is problematic for RF sample capture because it can break continuity and introduce timing offsets. To compensate for these errors, we maintain a 16 MHz counter during acquisition which is activated while sample capture is paused and then reports the elapsed pause time once sample capture resumes. This information is then also sent over Bluetooth so the end-user is able to accurately reconstruct and preserve timing of sample captures and interpolate missing values. During testing, sample dropouts occurred infrequently so we adopted this approach rather than a more complex adaptive sampling scheme based on error rate or incurring the additional overhead of retransmissions.

2.3 Crowdsourcer and Server Software

The choices of server architecture and software frameworks were driven primarily by convenience, ease of use and adaptability rather than resource optimization as we see for the Sitara. The server application uses common tools and frameworks including Python Flask and an Apache front-end paired with Gunicorn, a Python WSGI server. The control functions are accessed from an interactive Javascript-based GUI in a web browser. Socket.IO libraries for Python, Javascript and Java provide low-latency, standardized event-based communication between different the server, Web GUI and the device gateways, respectively.

A remote operator issues commands from a web client which are processed by the server and relayed to the appropriate client device gateways. The web client, server, and gateway each activate a set of event listeners which filter relevant messages. Messages containing measurement results in response to commands, such as RSSI and IQ data, are stored in the server's database. The web interface provides a Google Maps overlay which can display real time client location and associated measurement data as shown in Fig 3. In addition to real-time monitoring of sensors, the web client also provides convenient tools for filtering and displaying

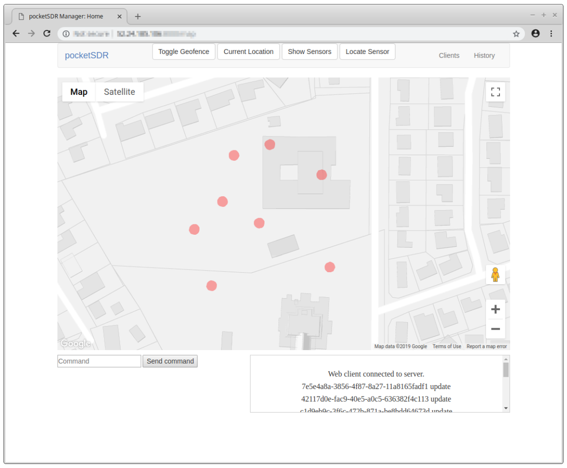

Fig. 3: Server homepage showing locations of active sensors

subsets of measurements according to parameters such as time, frequency, RSSI threshold and location. Some of these capabilities will be demonstrated in section 4.

The Server application provides a high-level abstraction for commands, leaving implementation details and low-level commands to client device gateways. For example, a server RSSI command emits a single message containing parameters such as frequency, bandwidth, interval and report type. The device gateway receives this message and issues multiple commands as appropriate, such as frequency tune and RSSI capture, over Bluetooth to individual Sitara devices. A Sitara in turn will interpret each of these commands and perform the appropriate functions to accomplish this task. Combining and abstracting commands at the top level in this way reduces latency, preserves bandwidth and spares server resources by minimizing the number of message exchanges for a given task. This abstraction also provides a modular approach making the sensor implementation-agnostic to the server.

During design, we considered implementing an existing standard such as IEEE 802.22.3 Spectrum Characterization and Occupancy Sensing (SCOS) Sensor [29], an extension of the SigMF specification [16]. We determined that such a framework provides a level of complexity and larger message set beyond our immediate needs. Specifically, SCOS implements a RESTful API typically requiring each sensor to host a web server; we instead use an open Socket.IO protocol built on WebSockets to maintain full duplex persistent connections with lower overhead for message exchange between server and mobile devices. Notwithstanding these differences, our implementation could be adapted to comply with the SCOS standard. Rather than implementing a RESTful API on each sensor, a gateway could be installed on the server which is compatible with the SCOS standard which emulates the API for each device. As requests are received at this server, the SCOS gateway would forward the appropriate commands, after converting to the Socket.IO protocol, to the mobile devices connected to the sensors. This would provide all the benefits of compatibility with the SCOS standard with minimal system redesign. Alternatively, a more complex approach could implement the SCOS standard as an application running on each mobile device with the server then acting as a passthrough or portal to connect using the SCOS protocol.

3 Evaluating Our Solution

We now present measurements characterizing the performance of our system under varying conditions. A small sample size is used to obtain reasonable expectation of performance.

3.1 Data Throughput

Three data paths potentially limit the real-time throughput of the Sitara system. The Bluetooth data rate, the SPI interface from the CC1200 radio to the SoC, and the SoC's onboard CPU itself. Depending on the use case, any of these could become a bottleneck. In our application, the maximum data transfer across the SPI bus exceeds the data rate of the Bluetooth link.

The CC1200 specifications limit the minimum SPI clock rate to 7.7 MHz for extended register reads, which include the registers of interest. In practice we have successfully achieved an 8 MHz clock rate. Two common modes of operation read from the CC1200 either three magnitude and two angle registers (8-bits each) or only the two angle registers. The maximum achievable sampling rates for these two modes were 64 kS/s and 104 kS/s, respectively.

The Bluetooth 5 standard defines a maximum transfer rate of 2 Mbps [8]. By utilizing this LE 2M PHY option, packet length extension and configuring the Bluetooth Maximum Transmission Unit (MTU) to match the packet length, we maximized throughput. Initial testing with special firmware achieved an effective data rate of approximately 1.3 Mbps between a Sitara device and a suitable phone. In typical usage scenarios we observe an average throughput greater than 1 Mbps. We obtained these results from a series of tests under varying environmental conditions using two different mobile phones, denoted *device 1* and *device 2* in Fig. 4. For each of these measurements, 244KB of data was transmitted over the Bluetooth link while the Sitara recorded the transmit time on its system clock. The 244KB transmission was then repeated at least 10 times for each test. For the environmental test (Fig. 4, top) we used two controls, the first places a mobile phone within 15 cm of the Sitara in an environment without in-band WiFi or Bluetooth activity as observed on a spectrum analyzer. The second control repeats this test, but separates the paired devices by 12 meters. We then perform additional measurements as follows: with the participant keeping the Sitara in a pocket while holding the phone in hand (Test 1), with the Sitara inside a backpack while the participant holds the phone in hand (Test 2), and finally, in a variety of different indoor and outdoor environments with the Sitara in the participant's pocket (Test 3). In addition to the environmental tests, we also conduct interference tests (Fig. 4, bottom). For these experiments, up to four interfering devices transmit over Bluetooth at their maximum data rate while the unit under test performs measurements. For these tests, the paired devices are separated by one meter. The controls consist of the same test, with no interferers (Control 1) and four interferers (Control 2); though in the case of the latter, the separation is reduced to 15 cm while other conditions remain the same. Test 3 of the interference measurements notably exhibits higher deviation than others. We speculate that this is due to characteristics of the Bluetooth protocol at marginal capacity or differences in hardware among devices. Further analysis of the Bluetooth performance and characterization is beyond this scope; these tests are intended only to approximate worst-case conditions expected in real-world scenarios.

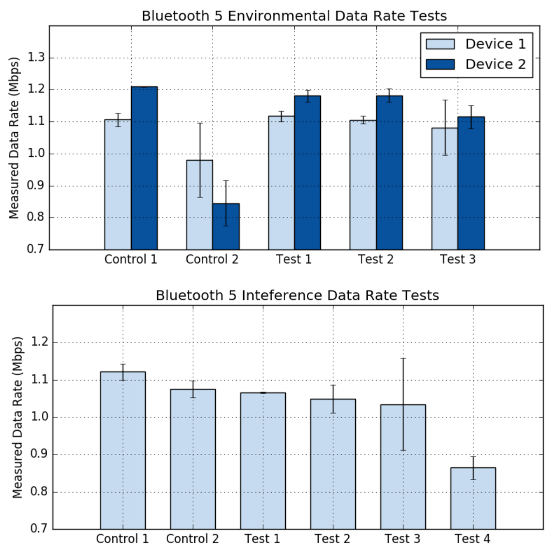

Fig. 4: Measured Bluetooth 5 data rate between Sitara and two mobile devices under varying test scenarios (top). Measured Bluetooth 5 data rate in the presence of varying numbers of interferers, where test number refers to the number of interferers (bottom). All error bars represent standard deviations.

In our application, minimal processing was required by the CPU during sample acquisition and was not found to impact throughput, therefore no attempt was made to evaluate load or processor utilization on the Sitara. If additional signal processing were to be carried out on the Sitara, then this may require further investigation.

3.2 Power Characteristics

The Sitara's power consumption ranges from approximately 18 mW to 180 mW depending on operational mode. Power usage in typical scenarios is shown in Fig. 5 below. In the Sitara idle state, the CC1200 radio is set to a low powered state, maintaining power only to the crystal oscillator and digital core; when no other commands are present the CPU in the nrRF52840 chip receives the wait-for-event (WFE) command which likewise powers down nonessential modules. The CPU will periodically wake up to handle events necessary to maintain a Bluetooth connection. During RX and TX states, the CC1200 remains powered on along with necessary RF, clock and interface peripherals; the CPU is more heavily utilized in these states, but no quantitative analysis was performed to determine the duty cycle of the active versus inactive/WFE state.

We initially tested battery life with a 3.7 V 180 mAh LIPO battery, allowing the Sitara to operate for several hours in

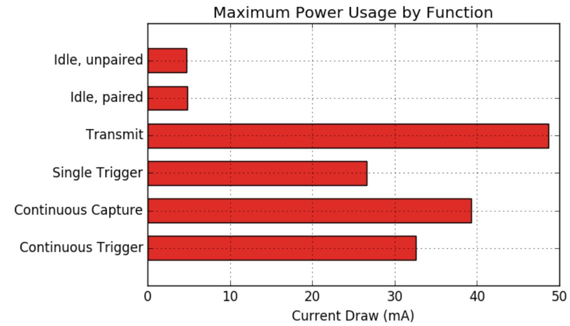

Fig. 5: Comparison of maximum Sitara energy consumption across a range of operational states.

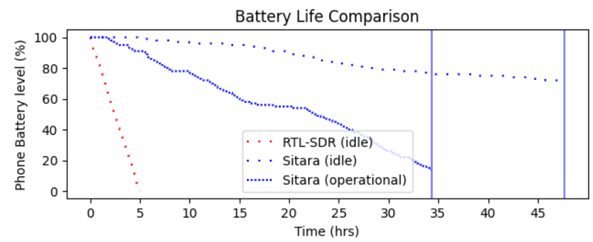

Fig. 6: Battery life comparison of phone connected to Sitaras in idle and RSSI capture operational modes against an idle RTL-SDR connected to a phone. Vertical bars depict time of battery exhaustion of the Sitaras.

any state. Repeated tests demonstrate that at full charge, the Sitara maintains a consistent Bluetooth connection until loss of power after 37 hours; this is consistent with the estimated battery life based on the measured idle power consumption. We later opted for a larger 850 mAh battery to extend battery life up to a week with a commensurate improvement in high duty-cycle operation. For comparison, the USRP E312 SDR uses a 3200 mAh battery to achieve 5.5 hours at idle [13]. We also tested the battery life of a phone tethered to an RTL-SDR in a connected but idle state (Fig. 6). The phone's 2600 mAH battery was depleted after about 5 hours. The same phone connected to a Sitara with a 180 mAh battery lasts longer than the Sitara at 35 hrs. Against TinySDR, in TX or RX states, Sitara has more than a 100mW power consumption advantage [21].

3.3 Timing and Synchronization

The CC1200 contains a phase-locked loop (PLL) which allows timing recovery in hardware for available modulation schemes. In other applications, our firmware performs a phase-lock function to effectively tune to a signal of interest. Regular synchronization may be necessary to maintain clock accuracy in use cases such as Doppler, time-of-arrival, or synchronous RF network architectures.

Sitara maintains a system clock for timing and measurement timestamps. The Sitara can synchronize its clock by receiving a clock command containing current GPS time from the mobile gateway over Bluetooth. Multiple such clock commands are sent until the mobile gateway observes a minimum round-trip time between the sent time and command acknowledgement — a technique not very different from many network time protocols. This synchronizes devices with GPS time with an error on the order of 10 milliseconds; this approaches the lower bound on latency or, more precisely, round-trip time between a mobile device and Sitara. More precise measurements can be obtained using a triggered RSSI measurement which will upload a timestamp associated with the RF trigger event to the server. Relative clock offsets between different nodes can be computed after triggered, timestamped RSSI measurements of a common event are uploaded to the server. If an application requires a more accurate node synchronization, then a more sophisticated approach would be appropriate [18], [20].

3.4 Server Performance

Although some performance compromises are necessary to meet our design goals for the Sitara hardware, cloud resources present few practical constraints as they are easily re-configured and scaled to meet demand. As a base configuration we reserve a host with 1 x86 vCPU and 0.5 GB of RAM. This adequately displays and serves 250 simultaneous clients, each reporting one RSSI measurement per second. Additional clients reporting RSSI, or multiple clients uploading IQ samples may require more resources if the live reporting mode is desired. As our system is expected to handle dropouts and latency associated with mobile data links, we place no hard timing requirements on server resources and assume best-effort. Notwithstanding, test results in this configuration exhibit a minimum round-trip time of 300 ms for an RSSI command issued from the server and processed on the Sitara with the result returned.

4 Experimental Results

In this section, we present results demonstrating how our system is used in two different applications. We choose these applications to demonstrate the key features of our system, namely: the portability of the Sitara for crowdsourced measurement and its utility as an SDR. We also explain how these demonstrations can extend to much more complicated experiments.

Note that for experiments involving volunteers, we obtained an institutionally-approved IRB to ensure consent and proper handling of potentially sensitive user data.

4.1 Spectrum Sensing

One of our primary design goals for the Sitara is to offer a convenient platform for crowdsourced spectrum sensing. As such, we demonstrate the versatility of our system in this capacity in real-world settings.

4.1.1 Transmitter Localization: Single Sitara

In this scenario, we deploy one Sitara acting as a receiver which is controlled locally using an open-source third-party mobile application: *nRF UART*. The user issues commands from the mobile gateway while walking to capture RSSI measurements which are later used to estimate the location of a "rogue" transmitter.

Multiple RSSI measurements are used to estimate the location of a receiver. The measurement points are chosen arbitrarily along pedestrian-accessible paths. The green circles in Fig. 7 (Left) represent measurement points used for localization. The radius of the green circles are proportional to their measured RSSI values. The red circle denotes the true transmitter location. The test site covers a roughly 28,000 m^2 area, with the farthest discernible measurement captured at a distance approximately 120 m from the transmitter.

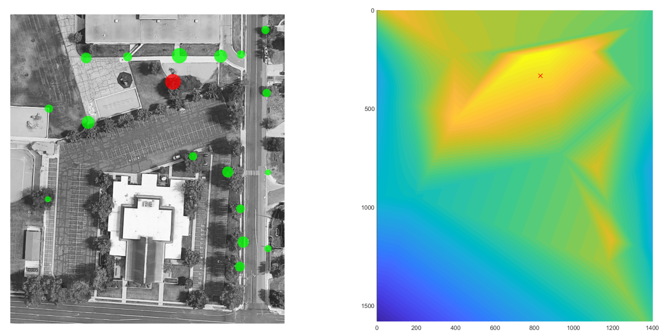

Fig. 7: (left) Outdoor test area with transmitter location (●) and measurement locations (●), with radius proportional to the measured RSSI value. (right) Transmitter location estimation map and true location (×).

Once the data is captured, the measurement coordinates are mapped to pixels from the test site image obtained from Google Maps. To test robustness, we discard the highest three RSSI measurements. The remaining points are overlaid to produce Fig. 7 (left), then as inputs to a Matlab interpolant object from which Fig. 7 (right) is generated. In this case, linear interpolation is used. The true location is approximately 9 m away from the estimate as indicated by the peak of the yellow region in the figure. Increasing the sampling rate or employing multiple simultaneous devices would improve this accuracy.

This scenario deploys a single Sitara with a local, active participant to capture RSSI measurements at different points. Local Sitara operation may be convenient for directed experiments which may not accommodate multiple passive users in a crowdsourced scenario. A more typical use case will involve multiple participants–active or passive–and rely on automated, server-initiated measurement commands. We demonstrate these capabilities next.

4.1.2 Crowdsourcing: Multiple Sitaras

We deploy twelve Sitaras among passive participants in a series of experiments. By passive we mean the participants are not directed but walk around freely while the nodes are operated remotely from the server, requiring no interaction from participants. By automating different test scenarios using server-side scripts, we are able to rapidly acquire large volumes of data. Fig. 8 depicts path loss in a suburban environment calculated from RSSI measurements and GPS coordinates of multiple devices obtained using a round-robin transmit scheme, in which individual nodes take turns operating as a transmitter while others measure RSSI. This is a fast and efficient method to generate data for inputs into complex propagation models that may otherwise be difficult to obtain [27]. Fig. 9 (left) presents a server-generated overlay of RSSI measurements obtained from twelve Sitaras scanning a range of frequencies over a user-specified duration. Here, each point radius scales relative to RSSI and each color, again, represents a unique device.

Fig. 9 (right) shows a server-generated overlay from prior measurements stored in the database using an automated command script. This image is generated by querying the server for measurements (blue dots) from a specific device between two time points. These measurements are not intended to locate a known transmitter, but instead demonstrate the passive crowdsourcing capability. Regardless, if our objective were to determine locations of possible sensors, we could request another overlay including multiple devices and an RSSI threshold. Along one path in the figure, we also see a drop in measurements, this likely indicates a loss of mobile data service in that particular area.

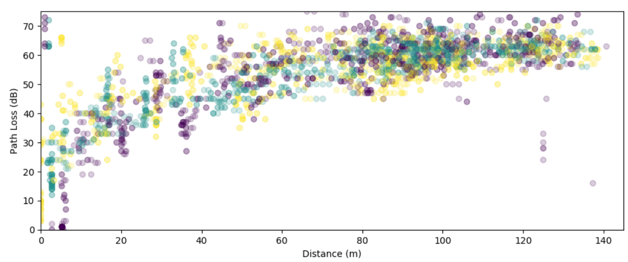

Fig. 8: Path loss using RSSI measurements from multiple Sitaras, distinguished by color, as a function of distance from a transmitter. Using Sitara, such data sets are easily generated and can be used for propagation models.

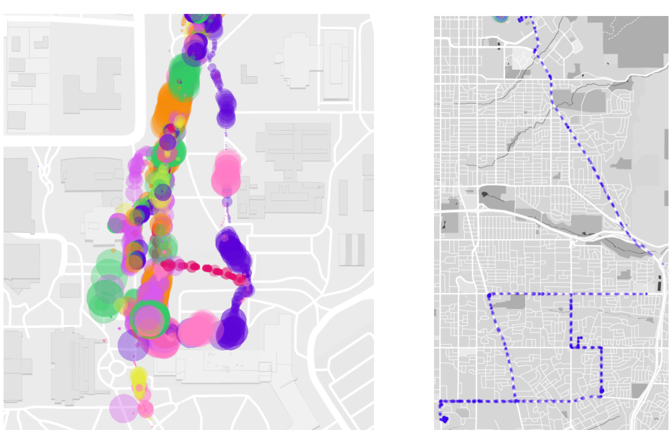

Fig. 9: Server-generated overlays of remote RSSI measurements from twelve devices carried by participants over a series of experiments, where each color represents a unique device and each point radius is scaled relative to RSSI value (left) and server generated overlay of RSSI measurements from one device over a wide-area (right).

We present these examples to demonstrate the utility of the Sitara for conducting distributed spectrum measurement experiments. In Section 6, we discuss further uses for our platform.

4.2 Server-side Processing with GNU Radio

In this application, we report the capability of the Sitara to capture IQ samples from an over-the-air frequency-shift keying (FSK) transmission, upload the raw data to a remote server and recover the original message by demodulating the signal in software. The experimental setup uses multiple Sitaras, one acting as the transmitter, while the others receive. The user gives the transmitter a transmit command which directs the Sitara's radio to send a short message consisting of a preamble, sync word and user-defined payload, using the CC1200's native 2-FSK modulation format. The receiving Sitaras execute a triggered capture command when a carrier signal is detected. The captured IQ samples are uploaded to the server and processed using a GNU Radio flow-graph to extract the payload message. We show phase plots captured from three different receivers in Fig. 10.

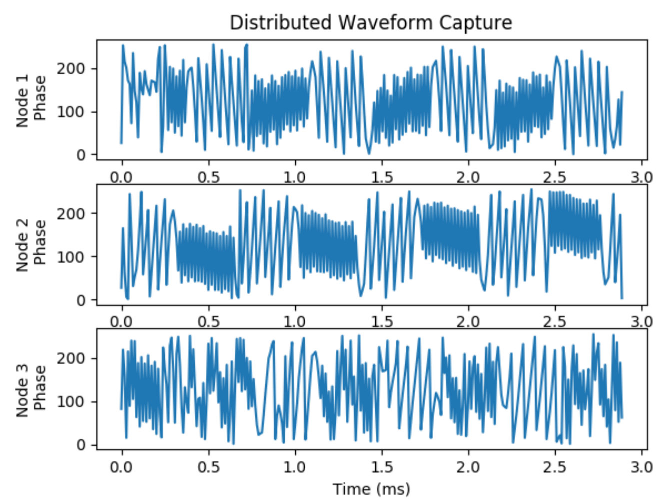

Fig. 10: Waveform phase measurements obtained from a triggered waveform capture using three Sitaras

Although only one captured signal was needed to demodulate the incoming signal in this case, processing captures from multiple Sitaras could further reduce the error rate. The waveforms in Fig. 10 can be synchronized using the capture timestamp and signal correlation. This example shows how Sitaras can be used to obtain a coordinated capture of an RF event and upload the data to the server for cloud-based reception. Similar processing has been effectively demonstrated in static receivers [12] but Sitara can enable such experiments with mobile endpoints.

5 Comparative Analysis

We compare Sitara with other crowdsensing systems and share a qualitative and quantitative analysis of user preferences and incentives that factor into the success of such applications. These factors, which we examine here, in part drove the development of Sitara.

5.1 Incentive Mechanisms

The fundamental question of incentive in crowdsourcing is whether the perceived value of participation meets or exceeds the cost or burden to the user. In the case of Sitara the cost would be the marginal effort of carrying an additional compact mobile device, charging it, and potentially giving up some privacy due to the transmitted location data. Depending on the usage scenario, there could also be a slight increase in power consumption and bandwidth on the connected mobile device. The perceived value would depend on the purpose for which the system is deployed. We begin this analysis by surveying some related systems, in no particular order, that employ differing crowdsource paradigms.

5.1.1 Other Crowdsourcing Applications

Opensignal is a mobile analytics company that measures mobile network performance to characterize real-world RF conditions [3]. This is accomplished through a software framework installed on partnering apps across more than 100 million devices. Under this model, no additional effort is required of the user, as this functionality is incorporated into apps already installed on a user's device. Opensignal compensates partner application developers based on the monthly active users from which these measurement metrics are obtained. Notably, the user receives no direct incentive — indeed, the user may not be aware that metrics are gathered without scrutinizing the Terms of Service or Software License Agreement.

The *OpencellID* Project is supported by more than 75k registered users who collaboratively localize cell towers [2]. This information contributes to a GPS-independent localization service and cellular coverage maps. This data is generally distributed according to a contribution-based mutual sharing agreement and is available for commercial use. End users who derive benefit from the service, either directly or indirectly, also contribute to the service.

Two other crowdsourced map-related services, *Mapillary* and *OpenStreetMap* aggregate and share map information, images and associated datasets [1], [4]. In the case of Mapillary, the company collects and openly shares data and offers machine-learning object location among its products. Openstreetmap, in contrast, is a nonprofit organization supported by a large community of contributors, some of which incorporate map data into their own products. In either case, contributors are not directly compensated.

Safecast is a nonprofit organization using volunteers to collect, primarily, radiation measurements [7]. Most of these measurements come from Bluetooth-capable Geiger counters developed and sold by the organization. An estimated 3000 participants used some 2000 sensors to contribute to the service's database more than 70 million data points over a 6 year period [7].

Crowdsourcing has also been adapted to applications in spectrum sensing [25], [28] for opportunistic dynamic spectrum access [22] and cognitive radio in vehicular networks [39], [41]. These latter systems generally include spectrally efficient ad-hoc networks to provide location based services in an otherwise congested RF environment.

5.1.2 Contact Tracing Adoption Rates

An application with a different but related goal to those mentioned is *contact tracing*. With recent viral outbreaks, including the SARS-CoV-2 coronavirus, interest in digital contact tracing has grown. A contact tracing system's effectiveness is directly dependent on its adoption rate [15], [30]. Existing solutions exhibit varying degrees of privacy and user discretion in their deployment. Some governments, such as those of Qatar and Turkey, mandate installation of a contact tracing application on personal mobile devices, while others, such as those in Iceland or Singapore, only encourage or support deployment of contact tracing applications [17], [30]. One study of American users' preferences with regard to contact tracing suggests that 44% would download a hypothetical app with personal data stored on the device, which drops to 39% when this would be stored on a central server [30]. This same study also examines how the developer of the application, such as local government, Federal government, industry or university, influences the likelihood of a user to download and install it. Yet another cross-country survey found that more, nearly 75% of respondents, would *probably* or *definitely* install a hypothetical contact tracing application [6]. Another report focuses on user engagement in contact tracing programs [26]. For comparison, the adoption rate of the Rakning C-19 application in Iceland saw an *adoption* rate of around 40% [24], while the NHS app for England and Wales achieved 28% [40].

5.1.3 Differing Incentives

From these many examples we can see a variety of successful crowdsourcing models with differing contribution and incentive models. For many participants, a collaborative community offers shared value and adequate incentive. In others, such as Mapillary, some services are offered freely to ordinary users while heavy or commercial users are charged. Safecast differs still from others in that participants are required to obtain a sensor which can cost more than $400, instead of merely installing a mobile application — yet a large community evolved to meet the goals of the project, an indication of the value and importance perceived by participants. Interestingly, in one publication, the Safecast team reports that consistent user participation is a challenge when sensors were provided free-of-charge to volunteers, tending to fade after some weeks of participation; but when volunteers are required to purchase their own sensors, greater participation rates and consistency were observed [7]. The authors suggest this was because the recruitment process would self-select from a more technologically-versed and dedicated pool of volunteers. Still others, while not widely deployed, propose frameworks that balance compensation with incentives to users, sometimes as a trade-off for surrender of some privacy while providing useful spectrum sensing information. These approaches are found among dynamic spectrum access systems [23], [38].

To better understand incentive mechanisms that could apply to Sitara, we examine these factors in a crowdsourced spectrum monitoring scenario. Volunteers would carry a Sitara sensor to measure and report RF spectrum measurements to a remote server in real-time. The burden on the participant would be carrying a compact sensor device and potentially consuming cellular data to report measurements. From the above examples, this would most closely resemble the Safecast system. In contrast, the Safecast system does not require real-time measurements or live communication to support its goals. This model is not directly applicable because Sitara would require a larger number of active users in a given area in order to be effective, whereas the utility of Safecast may only depend on one user to occasionally collect measurements over a large area. Recruiting enough users and achieving a reliable distribution sufficient for a potential Sitara application may prove challenging without some additional incentive. Transmitter localization, for example, would typically require multiple participants simultaneously within detectable signal range. Contrast this again with Safecast, where a single user may only need to drive through a geographical region once during the day to acquire the necessary data. Still, the user participant threshold remains be far less than what contact tracing requires.

A successful deployment of Sitara may depend on a reduced burden on participants or a strong incentive. Ideally, no effort beyond that of ordinary daily activities would be required. Opensignal, OpencellID and others benefited from this approach by incorporating measurements into existing applications on the users' mobile devices. For reasons addressed previously, this approach would not currently work for RF sensor measurements but could be developed in partnership with manufacturers to incorporate such sensors. Admittedly, this would cost much more than simply partnering with application developers. For this reason, we further examine user preferences in order to understand what approach would be most effective for Sitara. We present results from this analysis in the following section.

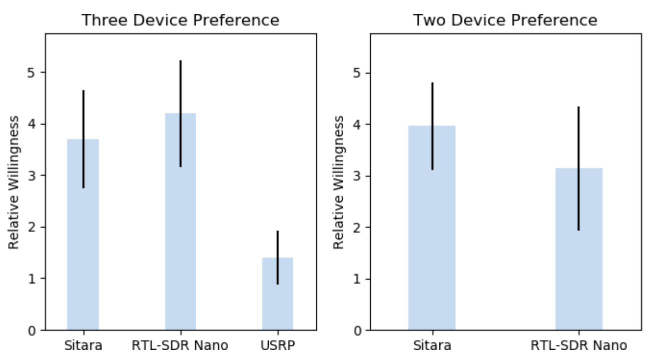

Fig. 11: Shows respondents' relative (average) willingness to carry selected sensor device (1-Unwilling, 5-Willing) with standard deviations.

5.2 User Experience

An effective crowdsource application depends not only on the capabilities of the system, but also the ability to recruit and employ users. The user component can depend on a number of factors such as incentives, as mentioned above, burden or convenience of the task, privacy controls and even the goals or purpose of the application. In order to discern the relative influence of these factors on an individual's willingness to volunteer for a crowdsourced task, we present selected results from a broader user study we conducted with more than 100 respondents.

5.2.1 Device Comparison

Our first comparison is based strictly on device specifications, examining current software-defined radio products available today. We borrow from the analysis put forth by [21] to present an objective comparison in Table 1. This is complemented by a subjective user evaluation presented in Figure 11. These results derive from an in-person component of our survey in which participants were handed, in a random order, three devices, the Sitara, RTL-SDR Nano and USRP E310 or only two devices, the Sitara and RTL-SDR Nano, and asked to rate their willingness to carry around each device as a volunteer in an experimental crowdsource setting on a scale of 1 (unwilling) to 5 (willing). The three device case data was collected in the first round of ($N = 10$) surveys and, as expected, showed a preference for the smaller devices, but no statistically significant difference in preference between the Sitara and RTL-SDR Nano ($P = 0.274$). For the two device case, ($N = 22$) respondents were handed both devices along with a demo phone to show how the RTL-SDR would connect. Results from these respondents demonstrate a statistically significant preference ($P = 0.0126$) for the Sitara over the RTL-SDR Nano.

5.2.2 Privacy

Respondents were asked to express some of their privacy preferences and habits in a series of questions. These show that users will sometimes allow online services to access location information but limit public access to other personal information such as names. Additionally, respondents tend to prefer privacy measures that increase anonymity, such as location obfuscation and geofencing, rather than those that only report when or what data is collected.

5.2.3 Organizational Associations

Another question addresses motivations relating to participation in a crowdsourced RF measurement activity by distinguishing between organizational sponsors and goals ranging from direct financial incentive to more altruistic endeavors. The latter would represent a crowdsourcing model resembling, for example, that used by Safecast participants. These results are presented in Figure 11.

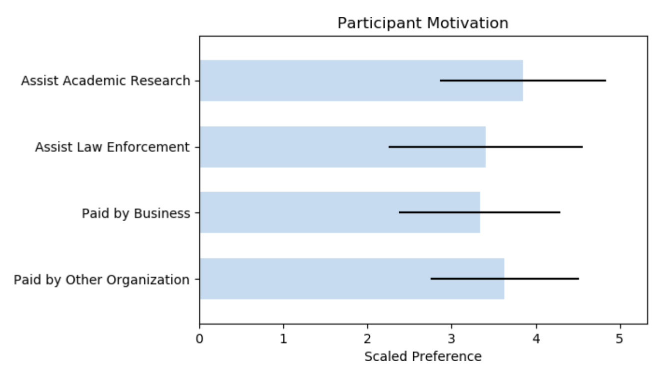

Fig. 12: Respondent willingness to participate based on organizational sponsor (1-Unwilling, 5-Likely to participate).

Here we observe minimal differentiation between user preferences, with a slight preference ($P = 0.056$) towards university or academic research. It is worth mentioning, however, that this could be an artifact of the sample population — primarily recruited from a university setting. The least favorable organizational sponsor is a business, which appears consistent with the preferences expressed towards

Platform Name	Size (mm)	Control Interface	Power Source	Battery Life TX mode	Cost
USRP E310	133 x 68 x 32	USB/Ethernet	battery	4 hours	$3200
PSDR	110 x 70 x 30	USB	battery	*	$500
HackRF	124 x 80 x 18	USB	USB	N/A	$300
LimeSDR	100 x 60	USB	USB	N/A	$300
PlutoSDR	117 x 79 x 24	USB	USB	N/A	$100
RTL-SDR	69 x 27 x 13	USB	USB	N/A	$20
Charm	89 x 57 x 25	Ethernet	PoE	N/A	N/A
TinySDR	30 x 50	LoRa Wireless	battery	*	$55
Sitara	70 x 50 x 20	Bluetooth 5	battery	17 hours	$38

TABLE 1: Price and portability feature comparison of SDR platforms

**We were unable to test or otherwise obtain these measurements*

app developers for government or academic organizations over business (in this case specifically Apple and Google) reported in a related study [43].

6 Discussion

6.1 Follow-on Efforts

We plan to expand our framework to further test localization algorithms and develop propagation models enabled by system. Hundreds of participants can accumulate millions of pair-wise propagation measurements between mobile nodes to generate and train models. By extending server-side applications, we can deploy Sitara in distributed, coordinated signal processing as proposed by others [11], [12], [18], [21]. Our system can be adapted to operate as a mesh network with other devices or a testbed for additional RF sensing applications as demonstrated by LLOCUS [31].

6.2 When should Sitara not be used?

Outside our use cases, there remain areas where Sitara is not particularly suited. The radio and RF front end limit the frequencies in which Sitara can operate. Additionally, the sample throughput constrains the SDR to narrowband operation. This is a direct consequence of the design choices made to minimize cost and maximize portability via a wireless back-channel. A more sophisticated receiver would require an FPGA or another ASIC with a much higher clock frequency and a delicate analog RF front-end, resulting in reduced battery life and increased device cost. Our platform is designed to operate in ISM bands where transmission is permissible and operating frequency is inherently restricted; it is not intended to be a wideband transceiver. For low duty-cycle or high bandwidth applications, TinySDR offers advantages [21]. For mobile spectrum sensing and other scenarios [19], Sitara may be preferred.

6.3 Challenges of Crowdsourcing

In this paper we present technical details of a system for RF crowdsourcing experiments and related user concerns. While we have covered these topics in other work [25], [34], solutions to these problems, such as privacy, are beyond the scope of this paper. As we have emphasized, the adoption rate and subsequent success of a crowdsourced system largely depends on a user's willingness to share potentially sensitive information. Future research could seek address these related questions: Does the collected information guarantee an appropriate level of anonymity? Are there techniques which can conceal the precise location of participants but still provide enough information to make sensing applications possible–the utility-privacy trade-off? If an effective technical solution to these problems does not exist, then the question may turn to user incentive to compensate for the risk or loss of privacy. In the case of spectrum monitoring, the viability of such a system may hinge on the economic balance of actual costs of RF spectrum abuse and financial incentive sufficient for recruitment. While efforts to address these problems can be found in literature [22], [25], [34], [38], [39], many of these challenges remain.

6.4 An Alternate Approach

From a user adoption perspective, an ideal system would operate transparently to the user, requiring no components external to a phone. This could be achieved through partnership and integration with the manufacturer. Allowing firmware adaptation could enable integration with a phone's existing radios but due to the heterogeneity of mobile hardware, logistics of development schedules and risk of performance impacts to the device, this may be too challenging. Instead, a separate chip and antenna could be integrated into the device with an open, modular driver and API. Just as other peripheral components such as cameras and sensors are incorporated into a product, so could a compact SDR. An internet service provider (ISP) or mobile device manufacturer could then market this as a ”radio-as-a-service” perhaps sold to a third party and used to subsidize the cost of the mobile device to the end user as an incentive for adoption and to defray the integration and development costs. This shared service model is in some ways similar to the shared software approach employed by other systems discussed in section 5. Another option would have the original equipment manufacturer (OEM) include this hardware and license or lease these capabilities to third parties.

7 Conclusion

The absence of viable options for large scale, coordinated, mobile, crowdsourced spectrum sensing motivated our development of Sitara, which we present here. We characterize the system and highlight its advantages for distributed spectrum measurement activities. We promote our design based on its merits as follows:

- Energy efficiency, with a battery life lasting up to one week — sufficient for a broad range of experiments.
- An inexpensive, compact form-factor including a wireless back-haul, offering an ideal solution for mobile, crowdsourced scenarios.
- Capability of local, manual or automated, and remote operation of sensors within a network distributed across a wide geographical area.
- SDR capabilities to measure complex temporal and spatial RF interactions.

We demonstrate the Sitara's capabilities in real-world scenarios and evaluate its performance. We examined the factors influencing the adoption of successful crowdsource system deployments and demonstrate the advantages of Sitara in this regard over existing solutions. The Sitara is a valuable open-source resource for research in distributed software-defined radio sensing.

Acknowledgment

This material is based upon work supported by the US National Science Foundation under Grant Nos. 1564287, 1622741, and 1827940.

References

[1] Mapillary.com.
[2] OpenCelliD - The world's largest Open Database of Cell Towers.
[3] Opensignal: Mobile Analytics Insights.
[4] OpenStreetMap.
[5] Radiocells.org.
[6] S. Altmann, L. Milsom, H. Zillessen, R. Blasone, F. Gerdon, R. Bach, F. Kreuter, D. Nosenzo, S. Toussaert, and J. Abeler. Acceptability of app-based contact tracing for COVID-19: Cross-country survey evidence (Preprint). 05 2020.
[7] Azby Brown and Pieter Franken and Joe Moross and Nick Dolezal and Sean Bonner. THE SAFECAST REPORT, 2017.
[8] Bluetooth SIG. *Bluetooth Core Specification*, December 2016. v5.
[9] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das. Specsense: Crowdsensing for efficient querying of spectrum occupancy. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.
[10] E. C. L. Chan. *Introduction to wireless localization : with iPhone SDK examples*. 2012.
[11] B. V. den Bergh, D. Giustiniano, H. Cordobés, M. Fuchs, R. C.-P. S. Pollin, S. Rajendran, and V. Lenders. Electrosense: Crowsourcing spectrum monitoring. *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN 2017)*, pages 1–2, March 2017.
[12] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe. Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '18, pages 60–71, Piscataway, NJ, USA, 2018. IEEE Press.
[13] M. Ettus. Universal Software Radio Peripheral. [Online].
[14] Ettus Research. *Getting Started with the Ettus Research USRP E312 SDR*, March 2016. Rev 1.
[15] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. Abeler-Dörner, M. Parker, D. Bonsall, and C. Fraser. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491), 2020.
[16] GNU Radio Foundation. *Signal Metadata Format Specification*, 7 2018.
[17] Government of Singapore. TraceTogether.
[18] E. Hamed, H. Rahul, and B. Partov. Chorus: Truly distributed distributed-mimo. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, pages 461–475, New York, NY, USA, 2018. ACM.
[19] G. Hattab and D. Cabric. Spectrum sharing protocols based on ultra-narrowband communications for unlicensed massive iot. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–10, Oct 2018.
[20] J. He, P. Cheng, L. Shi, and J. Chen. Time synchronization for random mobile sensor networks. volume 63, pages 2712–2717, 12 2012.
[21] M. Hessar, A. Najafi, V. Iyer, and S. Gollakota. TinySDR: Low-Power SDR Platform for Over-the-Air Programmable IoT Testbeds. In *NSDI*, 2020.
[22] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang. SpecGuard: Spectrum Misuse Detection in Dynamic Spectrum Access Systems. *IEEE Transactions on Mobile Computing*, 17(12):2925–2938, 2018.
[23] X. Jin and Y. Zhang. Privacy-Preserving Crowdsourced Spectrum Sensing. *IEEE/ACM Transactions on Networking*, 26(3):1236–1249, 2018.
[24] B. Johnson. Nearly 40% of Icelanders are using a covid appand it hasnt helped much.
[25] M. Khaledi, M. Khaledi, S. Sarkar, S. Kasera, N. Patwari, K. Derr, and S. Ramirez. Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, page 235247, New York, NY, USA, 2017. Association for Computing Machinery.
[26] C. McClain and L. Rainie. The Challenges of Contact Tracing as U.S. Battles COVID-19. *Pew Research*, 2020.
[27] Y. S. Meng, Y. H. Lee, and B. C. Ng. Empirical near ground path loss modeling in a forest at vhf and uhf bands. *IEEE Transactions on Antennas and Propagation*, 57(5):1461–1468, May 2009.
[28] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng. Empirical validation of commodity spectrum monitoring. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, page 96108, New York, NY, USA, 2016. Association for Computing Machinery.
[29] NTIA. *IEEE 802.22.3 Spectrum Characterization and Occupancy Sensing*, 1 2018. v0.0.2.
[30] P. H. ONeill. No, coronavirus apps dont need 60% adoption to be effective.
[31] S. Sarkar, A. Baset, H. Singh, P. Smith, N. Patwari, S. Kasera, K. Derr, and S. Ramirez. Llocus: Learning-based localization using crowdsourcing. *ACM Mobihoc: International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2020.
[32] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder. 5g: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE Journal on Selected Areas in Communications*, 35(6):1201–1221, June 2017.
[33] S. K. Sharma, T. E. Bogale, L. B. Le, S. Chatzinotas, X. Wang, and B. Ottersten. Dynamic spectrum sharing in 5G wireless networks with full-duplex technology: Recent advances and research challenges. *IEEE Communications Surveys Tutorials*, 20(1):674–707, Firstquarter 2018.
[34] H. Singh, S. Sarkar, A. Dimri, A. Bhaskara, N. Patwari, S. Kasera, S. Ramirez, and K. Derr. Privacy Enabled Crowdsourced Transmitter Localization Using Adjusted Measurements. In *2018 IEEE Symposium on Privacy-Aware Computing (PAC)*, pages 95–106, 2018.
[35] P. Smith, A. Luong, S. Sarkar, H. Singh, N. Patwari, S. Kasera, K. W. Derr, and S. Ramirez. Sitara: Spectrum Measurement Goes Mobile Through Crowd-Sourcing. *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 46–54, 2019.
[36] SPAN lab. Sitara. https://github.com/SPAN-WashU, 2019.
[37] C. Wang, F. Haider, X. Gao, X. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir. Cellular architecture and key technologies for 5G wireless communication networks. *IEEE Communications Magazine*, 52(2):122–130, February 2014.
[38] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang. Incentivizing Crowdsensing With Location-Privacy Preserving. *IEEE Transactions on Wireless Communications*, 16(10):6940–6952, 2017.
[39] Z. Wei, F. R. Yu, and A. Boukerche. Cooperative Spectrum Sensing with Trust Assistance for Cognitive Radio Vehicular Ad Hoc Networks. In *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '15, page 2733, New York, NY, USA, 2015. Association for Computing Machinery.

[40] C. Wymant, L. Ferretti, D. Tsallis, M. Charalambides, L. Abeler-Dörner, D. Bonsall, R. Hinch, M. Kendall, L. Milsom, M. Ayres, et al. The epidemiological impact of the NHS COVID-19 App. *Nature*, pages 1–8, 2021.

[41] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor. Mobile Crowdsensing Games in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 67(2):1535–1545, 2018.

[42] W. Xue, w. qiu, X. Hua, and K. Yu. Improved wi-fi rssi measurement for indoor localization. *IEEE Sensors Journal*, PP:1–1, 01 2017.

[43] B. Zhang, S. Kreps, and N. McMurry. Americans' perceptions of privacy and surveillance in the COVID-19 Pandemic. 05 2020.

[44] M. Zhou, Z. Tian, K. Xu, X. Yu, and H. Wu. Theoretical entropy assessment of fingerprint-based wi-fi localization accuracy. *Expert Systems with Applications*, 40(15):6136 – 6149, 2013.