accepted to appear in Proceedings of Percom 2020, Austin, TX

Performing WiFi Sensing with Off-the-shelf Smartphones

Steven M. Hernandez and Eyuphan Bulut
Department of Computer Science, Virginia Commonwealth University
401 West Main St. Richmond, VA 23284, USA
{hernandezsm, ebulut}@vcu.edu

***Abstract*—WiFi sensing has recently attracted a lot of attention in providing a method for device-free sensing with standard WiFi devices through the use of Channel State Information (CSI). However, access to CSI from user-level applications is not provided by most WiFi devices, specifically ubiquitous smartphone devices. In this demonstration, we present our custom developed application for collecting, labeling and processing CSI on-device, in real-time, for standard off-the-shelf Android smartphones. We additionally demonstrate the use of our application on select device-free sensing tasks.**

I. Introduction

WiFi sensing uses the WiFi communication between connected devices to enable device-free sensing [1], [2]. Channel State Information (CSI) is the metric used to describe phase and amplitude information for the received signal across multiple subcarrier frequencies. Each subcarrier is uniquely affected based on obstructions within an environment thus providing insights into actions occurring within that environment. Given access to CSI by toolkits such as the *Linux 802.11n CSI Tool* [3] and *Atheros CSI Tool* [4], it is possible for researchers to evaluate WiFi Sensing with Network Interface Cards (NICs) such as the Intel 5300. However, very few other WiFi NICs or WiFi-enabled devices provide application-level access to this rich CSI data source. Recent additions to the Espressif IoT Development Framework (ESP-IDF) introduce the ability to collect CSI directly from user-programmed software on an ESP32 WiFi microcontroller. We leverage this new functionality to give CSI access to standard Android smartphones. In this work, we demonstrate how a smartphone with our custom developed app can collect, process and label CSI through a number of WiFi sensing experiments.

II. Implementation

Our implementation allows off-the-shelf Android smartphones with our custom developed app to collect, label and process Channel State Information directly on-device. This can allow for more ubiquitous deployment of CSI collecting devices beyond what is currently possible. Our system diagram in Fig. 1 shows the target

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1744624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

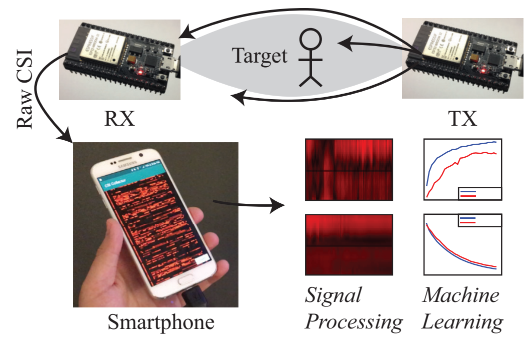

Fig. 1. Our system consists of a transmitter device (TX) sending packets to a receiver (RX). CSI values are affected by the presence of a target near TX and RX. Raw CSI is then sent to our custom built smartphone application for processing and labelling.

person located between an ESP32 WiFi transmitter (TX) and a receiver (RX). As TX sends packets to RX, the presence and actions performed by the target uniquely changes the amplitude and phase across different subcarrier frequencies for the received signal. These changes are then used by the RX to compute the raw CSI which is then sent directly to our custom designed smartphone application through a wired connection. From here, the application allows us to label samples which can then be stored for later analysis such as signal processing and machine learning classification tasks.

A. ESP32 based CSI Collection

Using the Espressif IoT Development Framework (ESP-IDF) that provides CSI access to user-level programs with recent additions, we developed a toolkit[1] with ESP32 microcontrollers to collect and record CSI. Our toolkit consists of code for WiFi access points and stations in both active transmitting and passively receiving modes. Our toolkit can be used both for WiFi sensing tasks as well as related tasks such as localization [2]. The goal of our toolkit is to leverage added benefits enabled by the ESP32 microcontroller; namely, the small size, low cost and thus mobile-capabilities of the device.

B. ESP32 with Smartphone

ESP32 microcontrollers are limited by both memory and CPU frequency; thus, computationally expensive

[1]https://stevenmhernandez.github.io/ESP32-CSI-Tool/

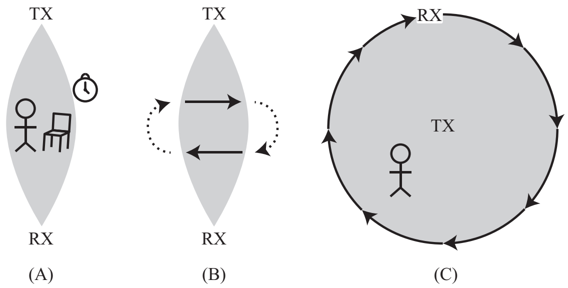

Fig. 2. Experiments performed: (A) Timed experiment with states *sit* and *stand*, (B) Experiment where the only recorded event is when the target is in the LOS of TX and RX, (C) Continuous experiment, where RX travels around a circular target area marking angular position.

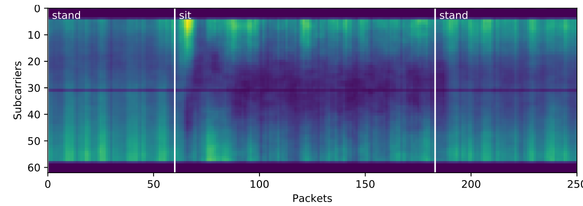

Fig. 3. Heatmap representing standing/sitting actions of a target showing that different actions produce unique effects on subcarriers.

tasks can instead be offloaded to a smartphone attached to the microcontroller. Standard Android smartphones allow UART serial communication through the use of an On-The-Go (OTG) USB cable. We use this cable to transmit raw CSI directly from the ESP32 to our application. By doing this, we enhance the abilities of the ESP32 further by also allowing access to power, real-time clock, sensors and networking capabilities (i.e., cellular) available from the smartphone. Currently, smartphone apps are not given access to CSI data. Instead, with our ESP32 to smartphone connection, we develop an Android app which can receive CSI from the ESP32 without requiring any special firmware manipulation. Further, by collecting CSI at the smartphone, we can label this experimental data in real-time. To facilitate these experiments, we develop three methods for recording and labeling CSI data with our application:

1) *Timed* - For experiments requiring a set of actions to be performed every t seconds.
2) *Press and Hold* - For experiments where actions occur only when a button is held.
3) *Toggle* - For experiments where actions switch immediately as a continuous stream.

Providing these three methods promotes a standardization in experimental recording techniques for many commonly performed WiFi sensing tasks. Existing CSI tools give programmatic access to raw CSI, but require users to then develop their own system for retrieving and labelling this data for their applications. By providing standardized methods through our application, this study allows fellow researchers to focus on experimental design and performing experiments rather than the design and development of a low level system for CSI extraction and labelling.

III. Experiments

To demonstrate the use of our system for WiFi sensing tasks, we detail three example use-cases. First, we consider a timed experiment, where during the labeling time, a target is instructed to stand-up for a few seconds, then told to sit down for a few seconds, illustrated in Fig. 2 (A). Next, we consider a target moving across the line of sight (LOS) of the transmitter and receiver, illustrated in Fig. 2 (B). Finally, we consider a third experiment, where the tracker walks a continuous path around a target area while collecting CSI as shown in Fig. 2 (C).

A. Sitting/Standing

Our first experiment makes use of the *Timed* labeling method provided by our application. For this, we place the ESP32 and smartphone 4 meters away from the transmitter. The target is positioned directly in the LOS of these two devices as shown in Fig. 2 (A) and is then instructed to sit, then stand at given intervals by the application. Fig. 3 shows the resulting CSI amplitude across 64 subcarriers as the target performs these actions. Visually, we can recognize when sitting, subcarriers become darker indicating that the amplitude increases. As the target changes position, their effect on the LOS signal also changes causing greater or lesser effect on the signal. In this case, sitting produces less LOS-blockage than the standing case. Recognizing the activity performed by a target (e.g. sitting or standing) in an indoor environment without requiring attached sensors has been a common task for CSI-based device-free activity-recognition research [5], [6].

B. Detecting Mobile Human Targets

For our next experiment, we demonstrate the use of the *Press and Hold* labeling method. This method is useful for experiments where the execution time of different human movement patterns has more variance than in the timed case. For this experiment, we place the transmitter and receiver 2.5 meters apart. The target then walks perpendicular to the LOS of the two devices as shown by the arrows in Fig. 2 (B). As the target reaches the LOS, the tracker presses the button indicating the target has reached the LOS. Then, the tracker releases the button as they exit the LOS area. Notice in the figure, the dotted lines indicate when the target exits the target LOS, in which case no label is specified. We can see that the CSI amplitude heatmap in Fig. 4 (Top) shows distinct signal variations while the target is travelling through the LOS. Considering the received signal strength indicator (RSSI) collected at the same time (Fig. 4 (Bottom)), we

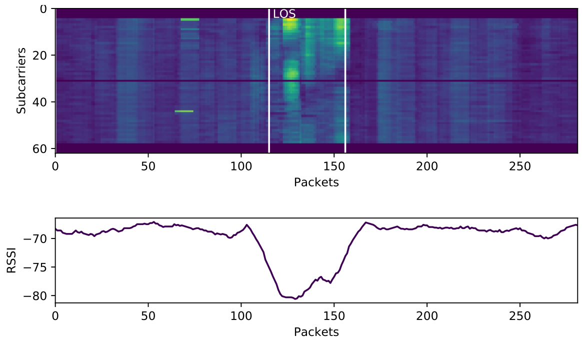

Fig. 4. (Top) Heatmap representing subcarriers effected by a passing target. (Bottom) Corresponding RSSI revealing a possible transition point.

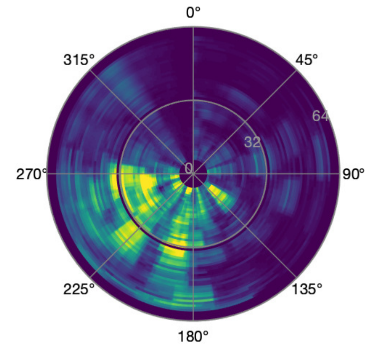

Fig. 5. Heatmap showing the amplitude of all 64 subcarriers as device travels a circular path around a target located at the 225° point.

can recognize the target most likely reached the central LOS around the 125 packets mark, where the RSSI reaches the lowest point. Identifying targets passing through a LOS area has been shown to allow for device-free tasks such as localization [7] and crowd-counting, even in through-wall situations [8].

C. Detecting Targets with Mobility

For our final experiment, we consider a case which requires continuous label changes, which we accomplish by using our *Toggle* labeling method. For this, we have two individuals, a static target and a mobile tracker holding our ESP32-smartphone system. Additionally, a static transmitter module is positioned in the middle of the target area. The target stands in a single position approximately 1 meter away from the transmitter. The tracker then walks around the central transmitter slowly while keeping a distance from the transmitter of approximately 2 meters. As the tracker moves, they press a button on the app each 45° segment to mark their position on the circumference of the target area. Completing one revolution around the target area provides the tracker with the circular CSI amplitude heatmap in Fig. 5. We can recognize the target position at 225° point visually by recognizing the brighter areas in the heatmap across most of the 64 subcarriers. We do notice however that the target affects the signal from 180° up to 270°, meaning that the presence of the target affects a 90° arc of movement. Identifying static targets can be a harder task to solve with device-free sensing because the lack of movement causes less obvious variance and noise in the received signal as shown in [9]. With our addition of movement with the receiver device, we see that static targets can be visible as they pass the LOS of the mobile receiver. Further work in this area will include identifying ways to limit the noise introduced by the mobile-receiver.

IV. Conclusion

In this work, we demonstrate our system for collecting, labeling and processing Channel State Information directly on-board a common off-the-shelf smartphone device connected to an ESP32 microcontroller programmed with our ESP32-CSI-Tool. We demonstrate the use of our system through three experiment collection methods using our system. We present the CSI collected in these experiments, demonstrating that targets can have noticeable effect on CSI. Future work includes exploring further the use of mobility for WiFi sensing tasks as well as using labeled CSI to perform on-device Machine Learning for an even more comprehensive WiFi sensing solution on standard smartphones.

References

[1] D. Wu, D. Zhang, C. Xu, H. Wang, and X. Li, "Device-free wifi human sensing: From pattern-based to model-based approaches," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 91–97, 2017.

[2] J. Wang, H. Jiang, J. Xiong, K. Jamieson, X. Chen, D. Fang, and B. Xie, "Lifs: low human-effort, device-free localization with fine-grained subcarrier information," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 243–256.

[3] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, no. 1, p. 53, Jan. 2011.

[4] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: ACM, 2015, p. 53–64. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790124

[5] M. Al-qaness, F. Li, X. Ma, Y. Zhang, and G. Liu, "Device-free indoor activity recognition system," *Applied Sciences*, vol. 6, no. 11, p. 329, 2016.

[6] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "Deepsense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[7] J. Wang, J. Xiong, H. Jiang, K. Jamieson, X. Chen, D. Fang, and C. Wang, "Low human-effort, device-free localization with fine-grained subcarrier information," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2550–2563, 2018.

[8] O. T. Ibrahim, W. Gomaa, and M. Youssef, "Crosscount: A deep learning system for device-free human counting using wifi," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9921–9928, 2019.

[9] Z. Yuan, S. Wu, X. Yang, and A. He, "Device-free stationary human detection with wifi in through-the-wall scenarios," in *International Conference on Wireless and Satellite Systems*. Springer, 2019, pp. 201–208.