AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms

Francesco Gringoli
University of Brescia/CNIT
Italy
francesco.gringoli@unibs.it

Marco Cominelli
University of Brescia
Italy
marco.cominelli@unibs.it

Alejandro Blanco
IMDEA Networks Institute
Universidad Carlos III de Madrid
Spain
alejandro.blanco@imdea.org

Joerg Widmer
IMDEA Networks Institute
Spain
joerg.widmer@imdea.org

ABSTRACT

Channel state information (CSI) is paramount to modern Wi-Fi communication systems, as it allows for proper equalization of frames at the receiver side and enables advanced signal processing techniques such as beamforming and MIMO. Given that the CSI can accurately mirror physical changes in the wireless channel, CSI analysis has become a valuable resource to many wireless sensing applications based on the opportunistic use of Wi-Fi signals. Since CSI can usually not be accessed by users directly, several CSI extraction tools have been published over the last few years for various Wi-Fi chipsets. In this paper, we present the first system ever capable of extracting CSI from 802.11ax consumer devices using the Broadcom 43684 Wi-Fi chipset. This platform can extract up to 160 MHz-wide CSI using 4x4 MIMO, and it is compatible with the latest HE PHY. We make our CSI extraction tool available to the research community to foster further work on this emerging topic.

CCS CONCEPTS

• **Hardware → Wireless devices**; • **Networks → Wireless local area networks**.

KEYWORDS

Wi-Fi, Channel State Information, 802.11ax

ACM Reference Format:
Francesco Gringoli, Marco Cominelli, Alejandro Blanco, and Joerg Widmer. 2022. AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms. In *The 15th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH'21), January 31-February 4 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3477086.3480833

WiNTECH'21, January 31-February 4 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-8703-3/22/01…$15.00
https://doi.org/10.1145/3477086.3480833

1 INTRODUCTION

Orthogonal frequency-division multiplexing (OFDM) is at the core of many wireless communication technologies, including new-generation Wi-Fi and 5G cellular networks. In general, OFDM systems need to accurately estimate the distortions introduced by the wireless channel on wide-band signals to work correctly. To this end, Wi-Fi devices use the headers of received frames to measure the channel state information (CSI), which is a discrete and quantized approximation of the channel's frequency response. However, CSI analysis has experienced a surge in popularity among researchers over the last decade as a practical tool for many sensing applications [8].

The propagation of signals through a wireless channel is affected by the configuration of the surrounding environment and by the presence of "obstacles" that alter the electromagnetic properties of the channel itself. Even minor variations in the environment are likely to be detected by observing the CSI since the overall physical properties of the channel have changed. This characteristic of wireless channels enables a wide range of applications: from gesture and activity recognition [2, 14] to device-free localization [15] and vital sign monitoring [12], to name a few. These applications use Wi-Fi signals opportunistically, analyzing the CSI to detect and learn specific patterns in the channel frequency response. Other recent works have also considered the possibility of "obfuscating" the information contained in the CSI to prevent unauthorized sensing [3, 7]. Even though arbitrary manipulation of the CSI today can be performed via software-defined radio (SDR) only, these works show that the "obfuscated" CSI can still be received successfully by off-the-shelf devices without hindering the communication. However, all these works were limited by the physical characteristics of the Very-High Throughput (VHT) PHY of IEEE 802.11ac—or even older PHYs—and do not exploit the new features from the new 802.11ax standard, like the new structure of High Efficiency (HE) frames.

While all the previous versions of the Wi-Fi standard have always focused on increasing the channel bandwidth, the most notable feature introduced in 802.11ax targets a more efficient usage of the available bandwidth. By reducing the subcarrier spacing, more subcarriers can fit into the same bandwidth than before, i.e., the CSI is much more "dense." First and foremost, this improvement enhances the estimation of the channel frequency response for

complex applications like beamforming and spatial multiplexing using multiple-input multiple-output (MIMO) transmissions. Higher resolution in the frequency domain implies that a receiver can discriminate more precisely a higher number of multipath components, which allows to better estimate channel parameters such as the angle of arrival or the time of flight, as well as the evaluation of Doppler shifts. We believe that also sensing applications will benefit significantly from higher spectral resolution, but this is yet to be proven.

Typically, the number of OFDM subcarriers depends on the actual channel bandwidth. If K subcarriers are used, the CSI corresponding to a single spatial stream can be expressed as a vector of K complex values $a_k \cdot \exp j\theta_k$, each corresponding to the magnitude a_k and the phase θ_k of the channel's frequency response measured on the k-th subcarrier. In a MIMO system, the CSI is usually computed for each pair of transmitting and receiving antennas. With the increased number of OFDM subcarriers available in 802.11ax, each channel state estimate contains much more valuable data.

This work presents the first publicly available system[1] that can extract CSI data from off-the-shelf devices supporting the HE PHY introduced in the latest 802.11ax Wi-Fi standard, with bandwidth up to 160 MHz and 4x4 MIMO. These new features will likely enable unprecedented improvements for all of the CSI-based applications described above.

The rest of the paper is organized as follows: we first present different solutions available today for CSI extraction in Section 2; then, we introduce our system in Section 3, and we compare its performance with its predecessor in Section 4; in Section 5, we show for the first time some impressive results achieved with the new HE PHY in 802.11ax; finally, we draw the conclusions in Section 6.

2 RELATED WORK

The CSI is a key element in Wi-Fi communications. However, most off-the-shelf devices just use the CSI internally in the PHY section of the Wi-Fi system, and only a few of them can report them to the user by default.

This work stems from Nexmon CSI, a popular CSI extraction platform for 802.11ac Broadcom chipsets [4]. As of today, this is one of the most comprehensive tools so far for CSI analysis, enabling the CSI extraction from 802.11ac frames (supporting both VHT PHY and 4x4 MIMO) on a wide range of Broadcom chipsets. However, the motley environment of CSI analysis includes several other CSI extraction tools for a wide range of Wi-Fi versions and chipsets.

The literature suggests that the most popular chipset used for CSI-based sensing research is the Intel Wi-Fi Link 5300. On this chipset, the Linux 802.11n CSI Tool [5], based on custom firmware and open-source Linux drivers, is used to access the CSI. However, the spectral resolution of the CSI is limited to 30 values corresponding to "subcarrier groups" rather than one value for each OFDM subcarrier. A complex value with signed 8-bit real and imaginary parts is reported for each group, which means that each group roughly corresponds to two subcarriers for 20 MHz channels and four subcarriers for 40 MHz channels. Recently, a newer tool was released for an 802.11ac Intel platform, specifically for the Intel 9260 card [16]. However, this card has only two antennas; thus, the best MIMO setting is 2x2, limiting research based on the angle of arrival and departure. By contrast, our CSI extractor tool enables 4x4 MIMO, which allows for discriminating more paths in the space domain.

CSI data of individual subcarriers can be extracted for Qualcomm Atheros chipsets using the Atheros CSI Tool [13]. This platform is entirely implemented in software and builds on top of the open-source Linux kernel driver `ath9k`, supporting many different chipsets like the AR9580, AR9590, AR9344, and QCA9558. Unlike the Intel Tool, the Atheros one offers finer quantization of the data extracted for each subcarrier, as both the real and the imaginary part take value in the range of integers $[-512, 512]$. However, also this platform is limited to 802.11n. To the best of our knowledge, newer Qualcomm Atheros QCA988x chipsets, supporting 802.11ac and powered by the `ath10k` wireless driver, still cannot be used for CSI collection as no open-source nor proprietary CSI extraction platforms were released.

Quantenna Communications, a chipset manufacturer for high-end access points (APs), offers another solution for CSI analysis. Some recent papers [1, 9] revealed that some Quantenna chipsets could report the CSI for 802.11ac frames. However, there is little knowledge about the performance of this platform since it is provided only to customers and system developers.

Lastly, it is worth mentioning alternative solutions based on SDRs. Well-known SDRs like the Wireless Open-Access Research Platform (WARP) [11] and the Universal Software Radio Peripheral (USRP) [10] could be used in principle for collecting CSI. Such platforms combine high flexibility with large bandwidths, as some radio front-ends can achieve 160 MHz and even more. Implementing a real-time Wi-Fi stack in software is, however, not straightforward. Recently, a project named *openwifi* developed on field-programmable gate arrays (FPGAs) an open-source implementation of the Wi-Fi stack [6] that allows CSI analysis and manipulation. However, *openwifi* supports only 802.11n at 20 MHz with no multiplexing (there is actually an ongoing effort to add support for 2x2 MIMO). Moreover, the major drawback of using SDR platforms is that they are usually orders of magnitude more expensive than consumer Wi-Fi devices; hence, their usage is often confined to specialized research labs.

3 THE AX-TENDED CSI EXTRACTOR

The implementation of this new CSI extraction tool has been inspired by our previous work on 802.11ac [4]. More specifically, we ported the tool we developed for the Broadcom 4365 chipset and adapted it for the new Broadcom 43684 802.11ax chipset. The reference platform is the AP RT-AX86U, developed by Asus, which incidentally is the successor of the RT-AC86U, the previous reference for our 802.11ac CSI extractor. However, adapting the tool to 802.11ax required significant modifications.

Figure 1 shows the classic FullMAC architecture adopted by Broadcom. All the 802.11-specific functions are managed internally by the chipset; on the Host, the main Linux operating system only configures the radio and exchanges data through the top DMA interface in the form of Ethernet-like traffic. Inside the wireless card, operations are split between the ARM CPU and the D11 microcontroller. The ARM CPU runs the "Wi-Fi OS" that controls all the

[1] We release our CSI tool at https://ans.unibs.it/projects/ax-csi

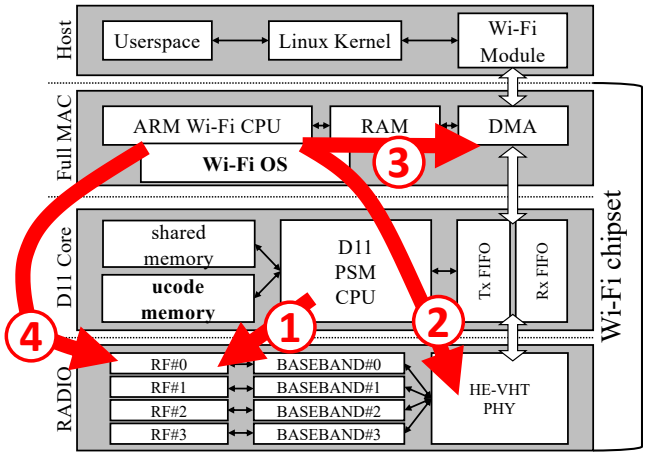


Figure 1: Overview of the CSI extraction process in the adopted architecture. Operations are split between the D11 core and the ARM CPU and performed in the order indicated by the arrows. When a target frame is received, the D11 core switches off the radio (1); then, the ARM CPU reads the CSI data (2), sends them to the userspace (3), and finally restarts the radio (4).

functions that are not time-critical while the D11 microcontroller manages time-sensitive operations, like channel access and generation of reply frames. In our previous 802.11ac tool, CSI extraction is entirely managed by the D11 core: when a new target frame is decoded by the underlying hardware, i) it switches off the receiving circuitry to freeze the CSI data; ii) it extracts the CSI data from the PHY and pushes it to the ARM CPU in the form of additional frames following the original payload; and iii) it finally restarts the radio receiver. No delays occur with this approach, and CSI data flow from the bottom to the top through the vertical double arrows in Fig. 1 on the right. Unfortunately, we could not directly port this approach from the old 4365 to the new 43864 chipset. In the latter, in fact, the ucode memory is already almost full because of the complexity of 802.11ax operations. There is room only for a few instructions inside the main loop, and the ARM CPU must now perform most of the CSI-related operations.

We will now describe in detail how the original 802.11ax architecture has been modified to extract the CSI, with reference to Fig. 1. First, we patched the D11 core to react to frames with specific content only, e.g., to a specific MAC address or frame control type. When a new target frame is received, the D11 core stops the radio (1) and no other frames can be received from now on. Then, the frame is pushed through the Rx FIFO towards the memory of the ARM CPU as it would usually happen in the standard architecture. We modified the main function in the Wi-Fi OS that processes incoming frames. When the frame coming from the Rx FIFO is detected, a new function is called to extract the CSI data from the PHY (2), embed it into one or more crafted UDP datagrams, and deliver it to the Host through the DMA interface (3). An application running on the main CPU of the Host receives the UDP datagrams containing the CSI and stores them into a packet trace. Finally, once all the CSI data are sent to the Host, the modified function in the Wi-Fi OS re-enables the radio to receive new Wi-Fi frames (4). As we will see later, performing all these operations in the ARM CPU rather than in the D11 core is slower, and in general it reduces the CSI throughput of the platform.

Table 1: Mapping of OFDM subcarriers to memory indices for the first PHY table for a generic receiving radio core. For spatial stream k, the corresponding PHY table starts at location k · 2048.

PHY	Bandwidth	Subcarriers	Memory indices
VHT, HT and Legacy	20 MHz	64	[0, 64)
	40 MHz	128	[0, 128)
	80 MHz	256	[0, 256)
	160 MHz	512	[0, 256) and [1024,1280)
HE	20 MHz	256	[0, 256)
	40 MHz	512	[0, 512)
	80 MHz	1024	[0, 1024)
	160 MHz	2048	[0, 2048)

3.1 CSI data layout

Like in the previous 802.11ac chipset, CSI data are organized into four PHY tables, one for each of the four radio cores. However, the structure of the tables is different, as it accounts for the higher number of OFDM subcarriers available in the HE PHY introduced with IEEE 802.11ax.

First of all, our preliminary analysis revealed that each PHY table now holds up to 8192 complex values: trying to read more leads to crashing the system. This number corresponds exactly to the maximum number of subcarriers expected in a 160 MHz HE frame with 4x4 MIMO, since every single stream can have up to 2048 subcarriers. Further investigation allowed us to map the data in the PHY tables to specific OFDM subcarriers, as shown in Tab. 1, where we report the offsets for a generic spatial stream; in general, CSI data for the k-th stream start at offset $k\cdot$ 2048. We have determined the CSI data layout by dumping the content of the PHY table under different conditions: i) by tuning the radio on different bandwidths; ii) by receiving frames with different bandwidths; iii) with different types of encodings (i.e., VHT or HE); iv) and MIMO configurations. Since UDP datagrams are limited to 1500 B, we can embed no more than 256 CSI values in each datagram reporting the CSI to the user space. While one single UDP datagram is sufficient to report one spatial stream of a 80 MHz VHT frame, two are required for each spatial stream of a 160 MHz VHT frame. This implies that 32 datagrams are generated when extracting 160 MHz VHT CSI with 4x4 MIMO, which increases to 128 for HE frames of identical configuration.

Interestingly, the layout of the CSI data in the PHY tables is rather peculiar, especially for 160 MHz-wide VHT transmissions, as shown in Fig. 2. The 160 MHz spectrum is not reported in contiguous memory locations, but it is split into two halves corresponding to the lower and higher 80 MHz parts of the spectrum. This gives us a useful insight into the implementation of the radio subsystem on the chipset: rather than a single radio working at 160 MHz, the system likely employs two radios at 80 MHz, each one providing

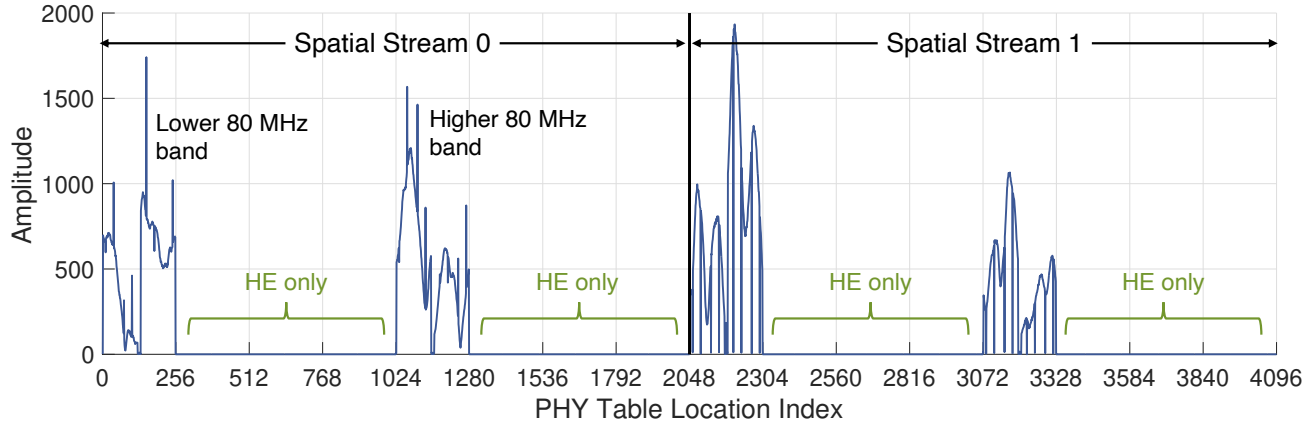


Figure 2: Layout in the PHY table of core #0 of the data associated to every OFDM subcarrier for a 160 MHz VHT frame using 2x2 MIMO. Only two spatial streams are reported here; four of them will require up to 8192 values.

on half of the entire band. A similar mechanism has been already observed on the old 802.11ac 4365 chipset. Even though this chipset officially supports only 80 MHz bandwidth with 4x4 MIMO, we experimentally extracted 160 MHz CSI with 2x2 MIMO by assigning different radios to different parts of the spectrum. The new chipset apparently uses the same principle but doubled the number of radios in order to achieve 4x4 MIMO at 160 MHz. We believe that the rationale behind this implementation is that the same circuitry should be able to manage not only full 160 MHz transmissions, but also 80P80 configurations in which two 80 MHz signals can be located everywhere in the 5 GHz band. In addition, using two radios with smaller bandwidth might be cheaper than implementing a single radio with very large bandwidth.

3.2 Testing the CSI extraction platform beyond 80 MHz with SDRs

Our experiments require crafting Wi-Fi frames with precise features in order to showcase the performance of the novel CSI extraction tool. For this reason, we resort to SDR platforms for some of the tests presented in the next sections. While we also added functions to the CSI extraction tool to inject HE-encoded frames through the Asus AP, we cannot finely control their timings. SDRs radios, instead, can transmit frames with very accurate timings: still, transmitting 160 MHz frames requires quite expensive hardware. For instance, the USRP N300 SDRs manufactured by Ettus Research, which is already an expensive solution, can manage the transmission of Wi-Fi frames up to 80 MHz. We present here a workaround that allows us to use a couple of these devices to transmit up to 2x2 160 MHz HE-encoded frames, or alternatively four much cheaper USRP N210 SDRs to transmit similarly encoded frames but limited to a 1x1 spatial configuration.

In Fig. 3 we focus on the first solution and we show how we can jointly use two N300 SDRs radios to transmit 160 MHz Wi-Fi frames. Each board is responsible for the transmission of half of the signal, i.e., either the lower or higher 80 MHz portion of the complete frame spectrum. When properly synchronized, the two halves sum up and generate a valid 160 MHz OFDM frame.

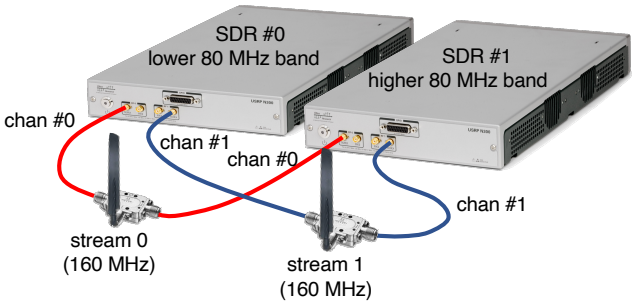


Figure 3: SDR setup for transmitting 160 MHz frames with 2x2 MIMO using two SDRs with smaller bandwidth. The SDRs need to be externally synchronized.

In Algorithm 1, we summarize the steps needed to generate the two 80 MHz portions of the frames in MATLAB using the WLAN Toolbox. By default, MATLAB creates a wide-band Wi-Fi signal by generating the corresponding I/Q samples that should be transmitted at 160 MS/s. Our script parses the MATLAB vector with the I/Q samples and recovers the complete sequence of OFDM symbols. Knowing the specific format of each OFDM symbol, the script applies to it an FFT and obtains the original OFDM constellation spectrum that is composed of 512 subcarriers in a VHT frame, or 2048 in an HE frame. Splitting each OFDM symbol into two *semi-symbols* in the frequency domain is straightforward as we only have to consider half of the total subcarriers, either in the lower or in the higher part of the spectrum. At this point, our script simply applies an IFFT to each semi-symbol and, after adding back the guard interval as dictated by the standard, it ends up with two *semi-signals*, each one having half of the spectral content of the original 160 MHz signal.

In order to transmit a decodable Wi-Fi frame, the two semi-signals must be transmitted by two separate USRP N300, one tuned to the central frequency of the lower 80 MHz portion of the spectrum and the other tuned to the center of the upper 80 MHz one. We cannot, in fact, use the two TX chains of a single USRP N300 as they

Algorithm 1 Split a 160 MHz Wi-Fi frame generated with the MATLAB WLAN Toolbox into two 80 MHz semi-signals. The spectrum of each semi-signal corresponds to the lower/higher part of the spectrum of the original signal.

Require: 160 MHz Wi-Fi frame at 160 MS/s
Ensure: Two 80 MHz signals at 80 MS/s whose joint spectrum is equivalent to the one of the input signal
1: Separate OFDM symbols & remove guard interval
2: Apply FFT to each OFDM symbol
3: Split left/right (low/high) band
4: **for** each semi-signal **do**
5: Apply IFFT on each half-symbol
6: Add back guard interval
7: **end for**

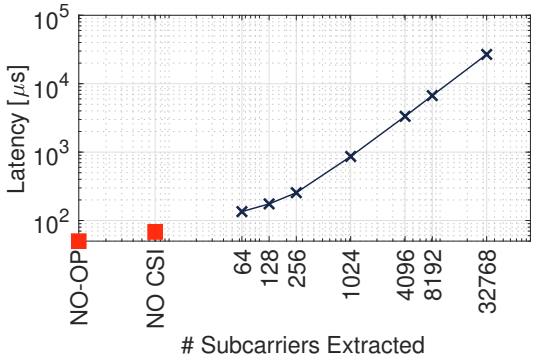


Figure 4: The latency introduced by the CSI processing chain depends on the number of extracted subcarriers.

cannot be tuned to different frequencies by design. While Fig. 3 also shows that two separate SDRs—each with two TX chains—allow to transmit 2 spatial streams, we only use a single stream configuration in this work. Needless to say, the two semi-signals must be synchronized: in our experiments, we achieve this by using the clock distribution module named Octoclock, also manufactured by Ettus Research.

4 CSI EXTRACTION PERFORMANCE

To cope with the limited amount of space available in the ucode memory of the new D11 core, we had to move a large part of the CSI processing code to the memory of the ARM processor. Even though the ARM CPU is quite fast, the concurrence with the D11 core and the increased latency have a detrimental effect on the system's performance.

In the old CSI extraction system [4], the D11 core had complete control over the data transfer operations. Once a target frame was detected, the D11 core itself configured the *deaf mode* on the radio hardware and immediately pushed the CSI data to the upper layers of the processing chain. However, due to the space constraints in this implementation, now the D11 core can only be configured to set the *deaf mode* on the radio, and we have to wait for a trigger that indicates that the data of a frame are available in the ARM CPU's memory before proceeding with the CSI extraction. While waiting for this trigger, the Wi-Fi chipset remains idle.

In order to estimate the impact of waiting for the complete transfer to the ARM memory on the system's performance, we set up an experiment to measure this latency. In this experiment, the ARM CPU does not extract any CSI data nor sends UDP datagrams to the user space, but just restarts the receiver. We use the SDR to transmit identical frames multiple times with a fixed time delay between successive frames. The fixed delay is reduced every time we repeat the experiment until we find that not all the transmitted frames have been received. This implies that the delay between two frames is too small and the receiving radio has not yet been restarted. The minimum delay for which all the frames are received is 50 μs, i.e., this is the transfer latency we are trying to determine. This is a considerable amount of time, given that some frames with a single data symbol (encoded with short guard interval at 80 MHz) can last as little as 43.6 μs. In Fig. 4, we indicate this result with the label `NO-OP`.

Then we repeat the same experiment to measure the latency introduced by the process of pushing the CSI to the user space. We do not extract CSI data yet, but we just craft a single UDP datagram that it is sent to upper layers through the DMA interface as seen in Fig. 1. We repeated the experiment for different sizes of the UDP datagram corresponding to the size it would have when receiving 20 MHz, 40 MHz or 80 MHz VHT frames respectively. In all cases, we measured a latency of 68 μs that is independent of the datagram size. This is reported in Fig. 4 with the label `NO CSI`.

Finally, we run the experiment with the CSI extraction mechanism in place. We measure the latency for different number of extracted subcarriers, from a minimum of 64 (VHT frame at 20 MHz, 1x1) up to a maximum of 32768 (HE frame at 160 MHz, 4x4). We notice that when the CSI data fit into a single UDP datagram (256 subcarriers or less) the delay is equal to 95 μs plus a quantity that is proportional to the size of the CSI (625 ns per subcarrier). The additional delay of 27 μs with respect to the one measured in the `NO CSI` case is due to the necessity of stopping/restarting the D11 core before/after reading the PHY tables. When more UDP datagrams are generated, we can see in Fig. 4 that the latency increases almost linearly up to the maximum 32768 subcarriers.

We then run another experiment to have a more convenient comparison of the performance of AX-CSI with respect to Nexmon CSI, which is the previous tool developed for 802.11ac. In Tab. 2 we report the number of CSI extracted each second for different configurations using the VHT PHY. Since the old tool cannot extract CSI from the latest HE PHY (and from 160 MHz VHT frames neither, by default), for this configuration we only report results for the new tool. Overall, the new tool performs slightly worse in terms of CSI captured per second than the previous one for 802.11ac. For instance, the old tool can extract almost three times more CSI than the new one from VHT frames transmitted at 80 MHz with different configurations. However, the situation is different when more data have to be processed, like in the 80 MHz 4x4 case where AX-CSI extracts almost twice the data extracted by Nexmon CSI in the same time span. This is due to a hardware bottleneck in the previous implementation in which D11 core was directly pushing CSI data to the DMA interface. Although the new system appears to be slower

Table 2: Performance comparison between the previous 802.11ac tool (Nexmon CSI) and the 802.11ax extractor (AX-CSI) in terms of CSI extracted per second.

encoding	VHT	VHT	VHT	VHT	VHT	HE
BW [MHz]	80	80	80	80	160	160
MIMO	1x1	4x1	1x4	4x4	4x4	4x4
# subcarriers	256	1024	1024	4096	8192	32768
NexmonCSI	8223	3034	2927	168	–	–
AX-CSI	3348	1101	1087	295	148	37

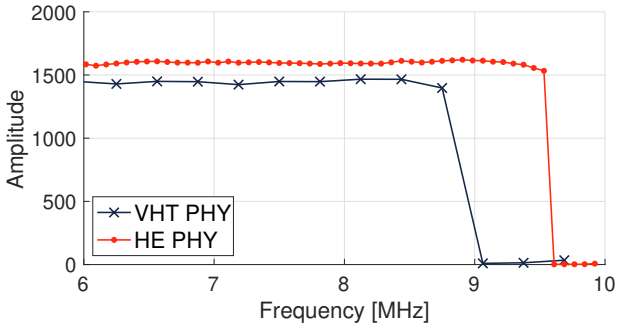


Figure 5: Detail of the CSI of 20 MHz frames. The spectral resolution achieved with HE PHY is four times larger than with VHT PHY. Amplitude is measured in arbitrary units as reported by the tool.

than the old one for several configurations, we believe that it can extract CSI measurements at a sufficiently high rate to enable many interesting wireless applications.

5 RESULTS WITH HE PHY

In this section, we investigate some interesting features of the CSI extracted using our tool from 802.11ax frames based on the new HE PHY. We start by showing in Fig. 5 a comparison between the CSI extracted from 20 MHz frames using the VHT PHY and HE PHY, respectively. Both frames are transmitted from the same SDR connected by cable to one receiving radio of the Asus AP. Here, we zoom into a small portion of the band to better appreciate the increased spectral resolution obtained with HE encoding (red dots) with respect to the VHT encoding (blue crosses). The former, in fact, adopts a four-times smaller subcarrier spacing, i.e., 78.125 kHz instead of the usual 312.5 kHz. Higher spectral resolution is key to accurately model the frequency response of channels with steep variations between adjacent subcarriers. We also notice that the HE spectrum has a much smaller guard band (see Fig. 5, on the right) which actually increases the amount of "useful" bandwidth. With respect to sensing applications, this slightly wider spectrum available in HE PHY may provide better time-of-flight or angle-of-arrival estimates even in the 20 MHz band.

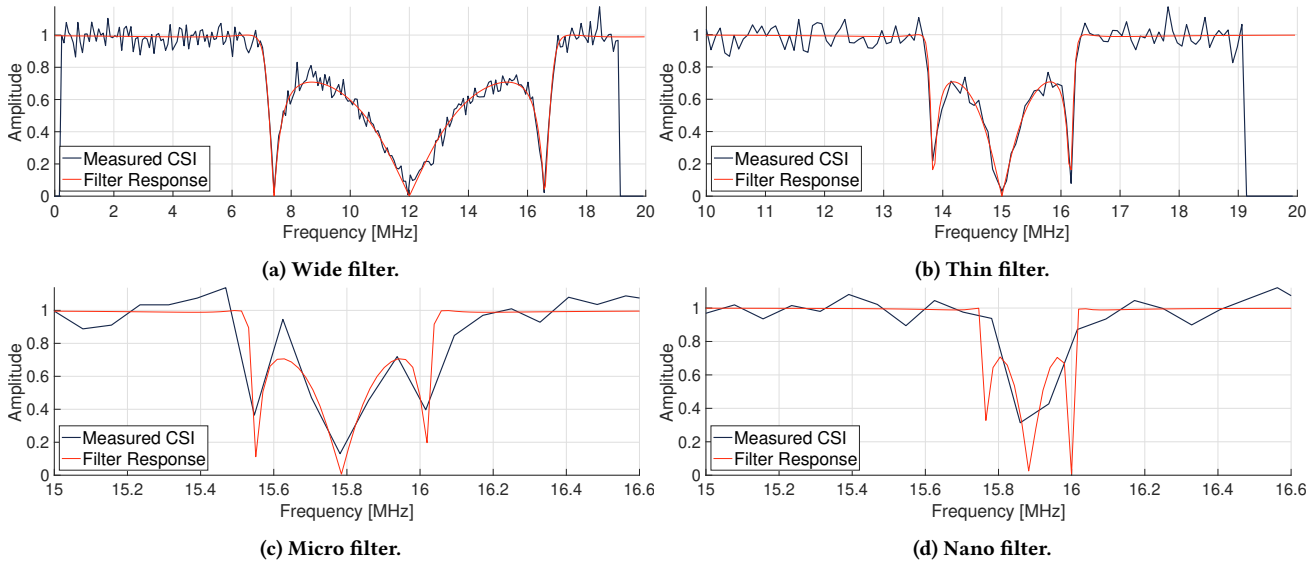
We then ran some experiments to showcase the advantages mentioned above introduced by HE encoding. To this end, (i) we generated with MATLAB a 40 MHz HE frame; (ii) we filtered the corresponding sequence of I/Q samples with a stop-band complex filter; and (iii) we transmitted the signal using cables from the SDR to the target Asus AP from which we collected the CSI as usual. We repeated this experiment decreasing the filter width each time as we show in Fig. 6a to 6d. For Fig. 6a and 6b we chose quite large stop-bands, respectively 8 MHz and 2 MHz. The specific frequency response of the filter can be determined with high fidelity in the collected CSIs: this means that even high frequency components in the CSI profile can still be discriminated at the receiver. In the last two figures at the bottom, instead, we chose very thin stop-bands, respectively 400 kHz and 200 kHz, which are respectively slightly larger and smaller than the subcarrier spacing in the case of VHT-encoded frames, to make the experiment more challenging. In the first case the shape of the CSI is still good enough to roughly represent the filter response. In the second case, instead, the width of the filter response becomes too small to be captured accurately. However, there are still two subcarriers that provide some hints about the central frequency of the filter.

In Fig. 7 we show the CSI from a 160 MHz HE frame transmitted from the SDR over the air. We immediately notice that the wireless channel between transmitter and receiver appears to be extremely selective in frequency. The level of details that is available with such a large bandwidth makes the extraction tool extremely interesting for sensing experiments where minor variations in the environment can be observed only in some portions of the spectrum; with our tool, we can capture all of them at once. This is highlighted in the figure where we also overlay the CSIs of the 8 individual 20 MHz HE subchannels, properly normalized, each with its 256 subcarriers. Apart from the obviously different behavior at each channel boundary (caused by the guard bands), the 20 MHz subchannels closely match the eight times larger 160 MHz spectrum.

Finally, we report in Fig. 8 a visualization of the full CSIs extracted for one 160 MHz HE frame with 4x4 MIMO, transmitted by another Asus AP. With our tool we can capture all the sixteen resulting CSI profiles for an astonishing number of 32768 subcarriers in total. To the best of our knowledge, this is the first CSI extraction tool that is capable of extracting such a huge amount of data from a single Wi-Fi frame.

6 CONCLUSIONS & FUTURE WORK

The capability to collect and analyze CSI data from real wireless network deployments is currently driving many research activities. In particular, the interest of many research groups is focused on opportunistic Wi-Fi sensing where it is important to collect CSI data at very high rate, that cover as large a portion of the spectrum and as many subcarriers as possible. In this paper, we presented a first tool to collect the most accurate CSI ever, thanks to its compatibility with the latest generation Wi-Fi standard IEEE 802.11ax. Our tool supports CSI collection from transmissions with up to four spatial streams and up to 160 MHz of spectral bandwidth per stream, extracting up to 32768 subcarriers per incoming frame. We believe that our system is an important contribution to the research community and has the potential to become a widely adopted tool. At the moment, the tool is only compatible with the Asus RT-AX86U AP which uses the Broadcom 43684 chipset. In the future, we plan to port our tool to other common devices using the same chipset.



(a) Wide filter.

(b) Thin filter.

(c) Micro filter.

(d) Nano filter.

Figure 6: CSI extracted from 40 MHz HE frames when the transmitted frames are arbitrarily pre-distorted using filters with a particular frequency response. CSI amplitude is normalized to 1 outside the stop-band region.

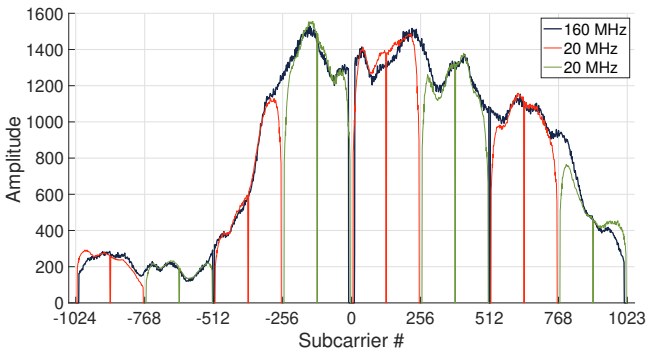


Figure 7: Spectrum of a 160 MHz frame overlaid with that of the eight constituting 20 MHz channels.

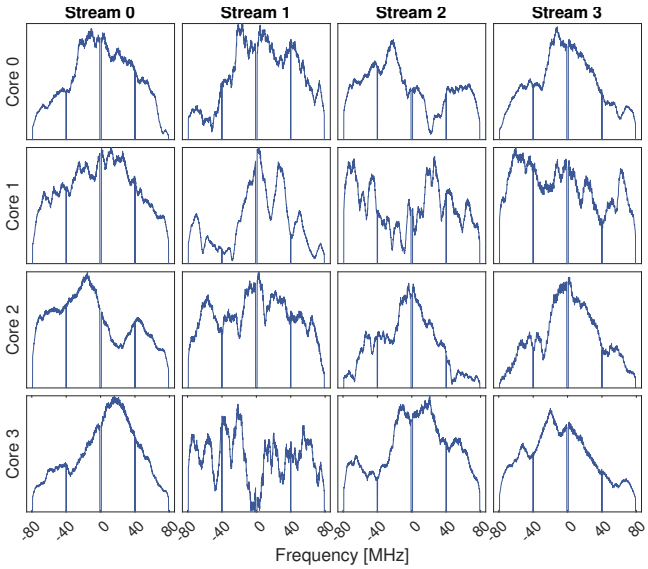


Figure 8: Amplitude of all the 16 CSI profiles extracted from a single 4x4 160 MHz HE frame. Each row contains the CSI collected at one RX core; columns represent the four spatial streams.

Future plans also include support for OFDMA, which would enable the simultaneous capture of CSI data vectors from multiple transmitters connected to the same network.

ACKNOWLEDGMENTS

This work has been co-funded/supported by the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 MAKI; and was sponsored in part by the Spanish Ministry of Science and Innovation (MICIU) grant RTI2018-094313-B-I00 (PinPoint5G+) and the Region of Madrid through TAPIR-CM (S2018/TCS-4496).

REFERENCES

[1] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Shrivatsan Rajagopalan, Shreya Ganesaraman, Aravind Seetharaman, Ish Kumar Jain, and Dinesh Bharadia. 2020. LocAP: Autonomous Millimeter Accurate Mapping of WiFi Infrastructure. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (Santa Clara, CA). 1115–1129. https://www.usenix.org/conference/nsdi20/presentation/ayyalasomayajula

[2] Zhenghua Chen, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. 2019. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. *IEEE Transactions on Mobile Computing* 18, 11 (2019), 2714–2724. https://doi.org/10.1109/TMC.2018.2878233

[3] Marco Cominelli, Felix Kosterhon, Francesco Gringoli, Renato Lo Cigno, and Arash Asadi. 2021. IEEE 802.11 CSI randomization to preserve location privacy: An empirical evaluation in different scenarios. *Elsevier Computer Networks* 191 (2021), 1–12. https://doi.org/10.1016/j.comnet.2021.107970

[4] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. 2019. Free Your CSI: A Channel State Information Extraction Platform For Modern

Wi-Fi Chipsets. In *13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization* (Los Cabos, Mexico) *(WiNTECH '19)*. 21–28. https://doi.org/10.1145/3349623.3355477

[5] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *SIGCOMM Computer Communication Review* 41, 1 (Jan. 2011), 53. https://doi.org/10.1145/1925861.1925870

[6] Xianjun Jiao, Wei Liu, Michael Mehari, Muhammad Aslam, and Ingrid Moerman. 2020. openwifi: a free and open-source IEEE802.11 SDR implementation on SoC. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 1–2. https://doi.org/10.1109/VTC2020-Spring48590.2020.9128614

[7] Xianjun Jiao, Michael Mehari, Wei Liu, Muhammad Aslam, and Ingrid Moerman. 2021. Openwifi CSI Fuzzer for Authorized Sensing and Covert Channels. In *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (Abu Dhabi, United Arab Emirates) *(WiSec '21)*. 377–379. https://doi.org/10.1145/3448300.3468255

[8] Yongsen Ma, Gang Zhou, and Shuangquan Wang. 2019. WiFi Sensing with Channel State Information: A Survey. *ACM Computing Surveys* 52, 3, Article 46 (June 2019), 36 pages. https://doi.org/10.1145/3310194

[9] Mohamed Naoufal Mahfoudi. 2019. *Unlocking wireless sensing potential in Wi-Fi and IoT networks*. Theses. Université Côte d'Azur. https://hal.archives-ouvertes.fr/tel-02431424

[10] Ettus Research. 2021. The universal software radio peripheral USRP Software Defined Radio Device. https://www.ettus.com/

[11] Rice University. 2020. WARP: Wireless Open Access Research Platform. https://warpproject.org/

[12] Xuyu Wang, Chao Yang, and Shiwen Mao. 2020. On CSI-Based Vital Sign Monitoring Using Commodity WiFi. *ACM Transactions on Computing for Healthcare* 1, 3 (2020), 27. https://doi.org/10.1145/3377165

[13] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise Power Delay Profiling with Commodity WiFi. In *21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. 53–64. https://doi.org/10.1145/2789168.2790124

[14] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. 313–325. https://doi.org/10.1145/3307334.3326081

[15] Rui Zhou, Meng Hao, Xiang Lu, Mingjie Tang, and Yang Fu. 2018. Device-Free Localization Based on CSI Fingerprints and Deep Neural Networks. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2018.8397121

[16] Anatolij Zubow, Piotr Gawłowicz, and Falko Dressler. 2021. On Phase Offsets of 802.11 ac Commodity WiFi. In *2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS '21)*. 1–4. https://doi.org/10.23919/WONS51326.2021.9415548