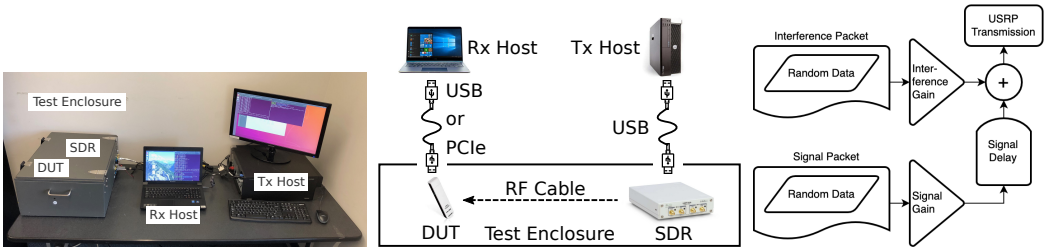
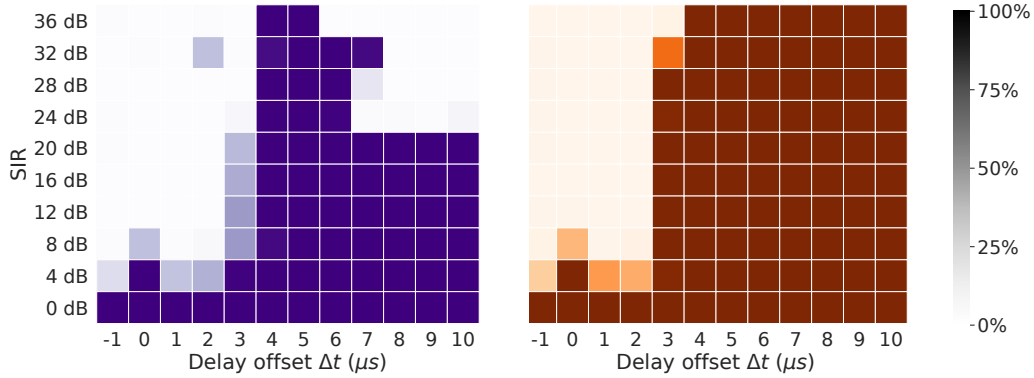
Graphical Abstract

Testing and Fingerprinting the Physical Layer of Wireless Cards with Software-Defined Radios

Johannes K Becker, Stefan Gvozdenovic, Liangxiao Xin, David Starobinski





(a) Atheros AR5B22

(b) TP-Link TL-WN722N

(c) Panda Wireless PAU06

(d) AmazonBasics Wi-Fi 11N

Highlights

Testing and Fingerprinting the Physical Layer of Wireless Cards with Software-Defined Radios

Johannes K Becker, Stefan Gvozdenovic, Liangxiao Xin, David Starobinski

- Experimental testbed architecture based on commodity software-defined radio for generating precisely timed traffic on the physical layer.
- The testbed can generate one or multiple packets at different power levels, and emulate wireless interference and signal collisions on Software-Defined Radio (SDR) hardware to test real network devices in reproducible conditions.
- Key features of the experimental testbed include measuring device receiver sensitivity, packet loss rate under different signal gains, and measuring device behavior when subjected to precisely-timed emulated packet collisions.
- Fingerprinting of Wi-Fi device types based on their distinct response to the capture effect and specially crafted "truncated packets".

Testing and Fingerprinting the Physical Layer of Wireless Cards with Software-Defined Radios

Johannes K Becker[a,1,*], Stefan Gvozdenovic[a,1], Liangxiao Xin[b,1], David Starobinski[a]

[a]*Boston University, Boston, MA, USA*
[b]*Sony Electronics, CA, USA*

Abstract

Many performance characteristics of wireless devices are fundamentally influenced by their vendor-specific physical layer implementation. Yet, characterizing the physical layer behavior of wireless devices usually requires complex testbeds with expensive equipment, making such behavior inaccessible and opaque to the end user, and complex to perform for wireless researchers. In this work, we propose and implement a new testbed architecture for software-defined radio-based wireless device performance benchmarking. The testbed allows tight control of timing events, at a microsecond time granularity, and is capable of accessing and measuring physical layer protocol features of real wireless devices, which allows to fingerprint the device type with high accuracy. Using the testbed, we measure the receiver sensitivity and signal capture behavior of Wi-Fi devices from different vendors. We identify marked differences in their performance, including a variation of as much as 20 dB in their receiver sensitivity. We further assess the response of the devices to truncated packets and show that this procedure can be employed to fingerprint device types with high consistency in both wired and wireless lab setups using only commodity SDR equipment.

Keywords: Testbed, Wi-Fi, device fingerprinting, signal synthesis, interference, capture effect, interframe spacing, RX-to-RX turnaround time

*Corresponding author

Email addresses: jkbecker@bu.edu (Johannes K Becker), tesla@bu.edu (Stefan Gvozdenovic), xlx@bu.edu (Liangxiao Xin), staro@bu.edu (David Starobinski)

[1]These authors contributed equally.

1. Introduction

With the explosion of wireless device adoption, the problems of Wi-Fi channel congestion and resilience to interference are becoming more acute than ever, especially in densely populated areas. New Wi-Fi specifications such as 802.11ax (Wi-Fi 6) aim to mitigate this problem by supporting existing as well as anticipated additional unlicensed spectra (such as the new 3.5 GHz spectrum [1] and the expanded 6 GHz spectrum [2]) to avoid congestion. However, the large and growing number of legacy Wi-Fi devices means that performance bottlenecks on the given spectrum cannot be avoided. Hence, ensuring high performance despite channel congestion and interference is essential.

Wi-Fi devices are commodity hardware on a product level. Yet, subtle manufacturer-specific physical layer implementations can result in substantial performance differences that are opaque to end users and complex to investigate for researchers. Benchmarking Wi-Fi performance and investigating behavior resulting from complex real-world situations, such as hidden nodes, currently require expensive physical setups in anechoic chambers under high time synchronization constraints. Specialized test equipment vendors offer wireless device testing equipment consisting of complex, specialized hardware and software modules [3–5], which have to be integrated by trained specialists to perform as intended.

To address this problem, we propose a novel testbed architecture for physical layer benchmarking that consists of a simple setup made from cost-effective, commodity components. The key novelty of this architecture resides in emulating parts of the channel environment (including interference from other users) within a SDR-based toolchain. The testbed reduces the complexity and expense required to conduct high-precision physical layer performance benchmarking, while leveraging the precise time synchronization and parameter control within the SDR to enable consistent and reproducible testing results.

We demonstrate the testbed capabilities by comparing the behavior and performance of Wi-Fi cards from four different manufacturers under precisely controlled physical layer testing conditions. First, we show that the cards exhibit noticeable differences in their receiver sensitivity (i.e., the lowest power level at which they can detect and demodulate RF signals). Next, we subject the devices to precisely time- and power-controlled collisions to assess their response to perturbed signals, thus demonstrating their different signal

capture behavior. Finally, we show how device types can be fingerprinted based on chipset-specific implementations. In particular, our results indicate distinct device responses to precisely crafted packet collision scenarios as well as so-called "truncated packets" that the testbed allows us to craft.

In summary, this paper makes the following contributions:

1. We propose an experimental testbed architecture for generating precisely timed traffic on the physical layer, subjecting real network devices to reproducible test conditions. The testbed generates one or multiple packets at different power levels, emulate wireless interference and signal collisions on SDR hardware. Then, it transmits the resulting composite signal to the device under test (DUT) wirelessly or using coaxial cable with configurable attenuation.
2. We demonstrate key features of the experimental testbed by measuring the devices' sensitivity and packet loss rate under different signal gains, and subjecting real Wi-Fi devices to packet collisions with high-fidelity control of timing and signal-to-interference ratio (SIR) parameters.
3. We show that it is possible to fingerprint different Wi-Fi device types reliably based on (a) their response to specifically crafted packet collisions triggering the capture effect, (b) "truncated packets", and (c) varying support for interframe spacings (IFS) that are smaller than the IEEE 802.11 specification defines.

The rest of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we describe our testbed architecture and our experimental setup. In Section 4, we discuss the experimental results. Finally, we conclude the paper and discuss future work in Section 5.

An earlier and shorter version of this paper appeared in the proceedings of the 22nd ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2019) [6], This journal version expands the work in multiple areas: First, it introduces both wired and wireless testbed configurations for devices under test. Second, it expands experimentation and analysis around receiver state machine behavior regarding shortest supported interframe spacings of tested devices. Finally, it expands on the consistency of fingerprinting characteristics within a certain device type.

2. Related Work

In this section, we provide an overview of previous work related to wireless testbeds and benchmarking, as well as theoretical and experimental analysis of the capture effect in Wi-Fi.

2.1. Benchmarking and Testbeds

Nychis et al. [7] propose an SDR-based platform that achieves precise packet timing by pre-loading a packet from the host to the FPGA and triggering its transmission based on the FPGA main clock on the USRP instead of the host clock (general purpose processor). Subsequent works aiming to satisfy the real-time requirements of wireless protocols follow this "split functionality" approach as well [8, 9], delegating the most real-time constrained functions within the protocol to customized FPGA modules. These workarounds are required to overcome processing, queuing, and bus transfer delays, which can add up to hundreds of microseconds [10]. Our testbed is significantly simpler, since it requires no FPGA modifications. Moreover, overlapping frames are added in software so that their offset is not affected by the host-radio hardware latency.

Park et al. [11] propose a wired testbed where signal of interest and interferers are generated on separate USRPs which are combined with a power combiner. While that testbed uses a sync cable to synchronize the two USRPs, our testbed generates both signals on the same device and therefore the same clock, precluding any frequency offset/drift errors.

Murphy [12] and Hunter [13] developed the WARP project – implementing the DCF MAC and OFDM PHY layers from IEEE 802.11-2012 on an FPGA platform. They benchmark interframe spacings (IFS) in order to evaluate the IFS of various 802.11 devices for standard compliance. In contrast, we characterize the minimal IFS that the radio device type is capable of, which is obtained with active radio fingerprinting. Similarly, passive device-characteristic IFS times have been used for passive 802.11 device fingerprinting [14]. In contrast to that work, we do not use naturally occurring IFS times for fingerprinting, but measure and characterize the smallest possible IFS time supported by a device's receiver.

Khorov et al. [15] present a Wi-Fi testbed for investigating the capture effect. The testbed generates two data streams on the application layer, and processes them in parallel USRP transmission chains before sending them

out over two antennas, generating a packet collision over-the-air. The offset between the frames is set by assigning each frame a different number and duration of backoff slots. The two transmitters are synchronized with a common local oscillator. However, our testbed only requires a single transmission chain for multiple colliding packets, and does not require additional synchronization mechanisms.

Our work differs from the related works in the following aspects. First, we provide a cost effective (single USRP) experimental testbed that allows fine control of transmission frame parameters such as power, delay offset between frames, modulation, and frequency channel. As such, we are able to intentionally generate precise collision scenarios of interest instead of relying on a large volume of collision-producing traffic and subsequent filtering of suitable collisions in post-processing [11]. Furthermore, our testbed can easily compare multiple Wi-Fi devices directly, and without requiring calibration. This allows us to reveal differences in manufacturer implementation of the physical layer. Although we showcase the testbed with Wi-Fi devices, this methodology can be applied to devices implementing other protocols. This opens the door for device co-existence testing with multiple protocol stacks easily implemented in GNU Radio, similar to Liu et al. [16].

2.2. Capture Effect

The *capture effect* describes a scenario in which a Wi-Fi receiver receives multiple transmissions at once, and can properly decode the stronger frame despite the signals overlapping. This effect is highly time dependent. The physical layer (PHY) state machine in 802.11 starts by detecting a signal preamble, and – after successfully receiving metadata on the demodulation type and decoding rate of the signal – subsequently decoding the contained symbols into received data. If a stronger signal arrives at just the right time, it may supersede the existing signal on the receiver. Note that overlapping signals can occur in several practical situations, for instance if two nodes transmit (or re-transmit) packets at the same back-off slot time [17] or in a hidden node scenario [18–20] when two transmitting nodes cannot sense each other.

Traditional analytical models for IEEE 802.11 performance analysis do not take the capture effect into consideration. For instance, Bianchi's Markov chain model [17] and its refined models [21–24] simply regard a packet collision as a packet loss. The work in [25] analyzes the performance of multi-hop 802.11 networks, under a full capture model (i.e., the stronger signal always

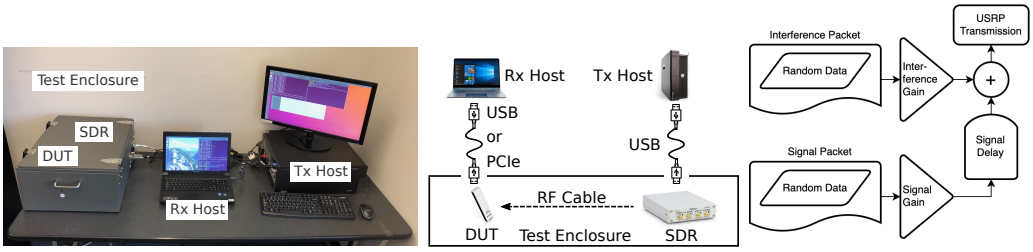
captures the channel) and a limited capture model (i.e., the stronger signal captures the channel only if it comes first). In our paper, we show that none of the tested Wi-Fi devices behaves in full accordance with either one of these models.

Other work, such as Chatzimisios et al. [26] and Daneshgaran et al. [27], propose analytical models to calculate packet loss based on the bit error rate (BER). However, those analytical results are only verified in simulation environments and do not consider the additional complexities arising from physical layer implementation in real hardware.

Experimental studies on IEEE 802.11 networks consider the physical layer behavior of Wi-Fi devices. Ware et al. [28] demonstrate that the channel is always captured by the packet having the strongest SIR in hidden node scenarios. This capture behavior can cause unfairness issues within Wi-Fi networks, despite the use of request to send/ clear to send (RTS/CTS). However, the SIR is the only parameter studied in that work. In this paper, we consider additional parameter such as packet arrival time and different chipsets.

The work by Ganu et al. [29] evaluates the capture effect using the ORBIT indoor wireless testbed [30] in a scenario with no hidden nodes. Their experimental results show that the capture effect significantly reduces throughput fairness: When two stations transmit packets to the same receiver, the transmitter with weaker received signal strength indication (RSSI) has higher packet loss probability and longer backoff delays, resulting in negative impact on its throughput. However, they do not test the capture effect in a hidden node scenario. In this paper, we evaluate the capture effect in situations when the transmitters could be hidden nodes with respect to each other (i.e., there is a significant delay between the starts of overlapping frames). Furthermore, we do not require an expensive and complex setup to generate precisely timed signal collisions.

Lee et al. [31] design a testbed based on Atheros Wi-Fi cards and carry out a measurement study on the capture effect with hidden node scenario in IEEE 802.11a networks. They reveal the conditions under which the capture effect takes place, such as packet arrival timing, signal-to-interference ratio (SIR), and bit rate. Furthermore, they show that the the packet preamble is more vulnerable to interference than the payload. However, this testbed consists of several independent Wi-Fi nodes, acting as sender, interferer, receiver, and sniffers. As a result, time synchronization between the nodes drifts over time, and other parameters like SIR cannot be precisely controlled.



(a) Testbed lab setup.

(b) Testbed hardware diagram.

(c) Collisions are emulated within a transmission flowgraph.

Figure 2: Testbed architecture. The SDR and the device under test (DUT) are placed in a shielded test enclosure and controlled from dedicated hosts on the outside.

Our testbed allows for full control over all relevant parameters while requiring fewer devices and no complex topology and device manipulation in order to obtain precise results.

Finally, all aforementioned papers except Khorov et al. [15] focus on evaluating the behavior of a single type of Wi-Fi card (chipset). In contrast, we compare the behaviors of multiple cards and show that they vary significantly while producing consistent results for each device type.

3. Testbed and Experimental Set-up

This section lays out the overall testbed hardware and software, and introduces our experimental setup and the necessary theoretical background required for the experiments we conduct.

3.1. Testbed

The proposed testbed emulates one or multiple transmission signals on a single host and sends the resulting signal with a USRP to real wireless devices, where reception statistics are collected. Thus, the testbed allows us to emulate physical layer signal collisions and allows fine-grained control of the parameters of the transmitting frames and of the channel, such as gain (attenuation), offset between frames, modulation, and channel frequency.

3.1.1. Hardware

The hardware setup of the testbed involves a transmitting host and a receiving host, and can be set up on a simple lab desk (see Figure 2(a)),

whereas other wireless testbeds such as the ORBIT require extremely complex hardware configurations [32].

As shown in Figure 2(b), the transmitter consists of an Ettus USRP B200 SDR board connected to a host PC[2] via USB, and the receiver consists of a separate host PC configured with the appropriate USB- or PCIe-based network card (i.e., the device under test (DUT)). The SDR and the DUT are placed in a shielded enclosure, i.e., a Ramsey Shielded Test Enclosure STE3500, providing more than 90 dB of isolation in the 2 GHz spectrum [33]. There are two ways to connect DUTs to the testbed:

1. Using a RF cable to connect the USRP to the DUT where the cable has configurable attenuation to emulate signal loss on the transmission path. The advantage of this setup is precise control of the exact signal strength the DUT is subjected to, without having to worry about the dimensions and orientation of the transmitter and DUT.
2. Using a 2.4GHz antenna on the USRP and over-the-air transmission to the DUT. This setup supports testing devices that do not expose their antennae with standardized connectors (such as printed circuit board (PCB) patch antennae). Furthermore, it reduces setup complexity in cases when precise control of the signal strength is not required (note that the relative signal strengths of interferer and signal packet are still precisely controlled, as they are part of the transmission chain itself – see Figure 2(c)).

3.1.2. Software

The software stack of our testbed consists of GNU Radio for signal generation, and the packet analyzer `tcpdump` [34] for collecting receiver data. On the transmitter side, we periodically generate Wi-Fi packets, using the gr-ieee802-11 library [35]. We emulate channel environment characteristics, such as relative signal strength, packet collision, and interference, directly on the transmitting host.

As shown in Figure 2(c), complex samples of signal and interference packets are summed up before transmission. Their transmission power gain as well as their delay relative to each other can be precisely controlled since they are both generated and added together on a symbol-level in software on the

[2]Dell Precision Tower 5810 XCTO Base (CPU: Intel Xeon Processor E5-1607 v3 3.10 GHz × 4, RAM: 15.6 GB).

Table 1: Tested Wi-Fi cards.

Make	Model	Interface	Protocols	Chipset
Atheros	AR5B22	Mini PCIe	a/b/g/n	Atheros AR9462
TP-Link	TL-WN722N N150	USB	b/g/n	Atheros AR9271
Panda Wireless	PAU06 300Mbps N	USB	b/g/n	Ralink RT5372
AmazonBasics	Wi-Fi 11N USB Adapter - 300 Mbps	USB	b/g/n	Realtek RTL8192EU

host (i.e., in GNU Radio) and transmitted with a single USRP. This setup ensures time synchronization in a much more straightforward way compared to setups with multiple physical transmitters. The two competing packets (signal of interest and interferer) are sent out with different MAC addresses to allow for easy packet statistics collection on the receiver side.

On the receiver side, a Wi-Fi card under test is connected to a separate host PC to receive Wi-Fi packets from the USRP. The card is set to monitor mode and data is collected via `tcpdump`. We then count the number of received signal packets and compare it to the number of packets transmitted to obtain the packet loss statistics under each configuration.

3.2. Experimental Setup

We next describe the experiments performed using the testbed, including experimental setup, parameters, and performance metrics.

3.2.1. Devices under Test (DUTs)

Our objective is to benchmark Wi-Fi cards with USB and PCIe-based interfaces, as shown in Table 1. All tested devices are popular, commodity devices using different Wi-Fi chipsets.

3.2.2. Parameters

The experiments take advantage of the high degree of parameter control that the testbed offers. In particular, we control the following parameters:

Delay offset (Δt), defined as the difference between the start time of the signal packet and the start time of the interference packet. Note that if the signal packet starts before the interference packet, the delay offset is negative. In the experiments, the delay offset is varied in steps of 1 µs.

Signal and interference gains, which can be controlled directly within the transmission flowgraph.

Signal-to-interference ratio (SIR), which is the ratio of the strength of the signal packet to the strength of the interference packet in dB. Precise control of the SIR allows for reproducibility in experiments related to packet collisions.

3.2.3. Signal Gain and SIR

In order to achieve desired signal and interference gains and SIR, we adjust the amplitudes of the signal and interferer samples before they are summed up in GNU Radio.

Specifically, a wireless signal s can be represented as a sequence of discrete complex samples, with the n^{th} sample denoted by $s[n]$. We denote the transmission power gain of signal s by G_s. The (normalized) power of signal s is

$$P_s(G_s) = \frac{1}{N} \sum_{n=0}^{N-1} |G_s s[n]|^2. \tag{1}$$

The parameter G_s allows us to control the gain of the signal. Therefore, converting to dB units, we have

$$P_s(G_s)\ (\text{dB}) \quad = \quad 20 \log_{10}(G_s) + P_{\text{USRP}}, \tag{2}$$

with the first term in the right hand side representing the *signal gain* (in dB), and the second term representing the transmission power offset of the USRP. We stress that the signal gain G_s is a relative quantity that is not calibrated to a specific output transmission power (i.e., one needs to estimate P_{USRP} if one wishes to know the actual transmission power P_s).

Note that Equation (2) is only applicable in the linear region of the transmitter's RF power amplifier. A too large value for G_s will eventually saturate the output power P_s to its maximum rated output power. Conversely, a too low value for G_s will flatten the output power at the noise floor.

Next, if we consider a desired signal s and interference signal i, we can express the signal-to-interference ratio (SIR) as

$$\text{SIR} \quad = \quad P_s - P_i \quad = \quad 20 \log_{10}(\frac{G_s}{G_i}), \tag{3}$$

where P_i is the interference power and G_i is the interference gain. In this paper, we use Equation (3) to calculate the SIR (e.g., setting $G_s = 1.0$ and $G_i = 0.1$ results in a SIR of -20 dB). G_s and G_i are chosen within the linear region of the transmitter's RF power amplifier where Equation (2) holds.

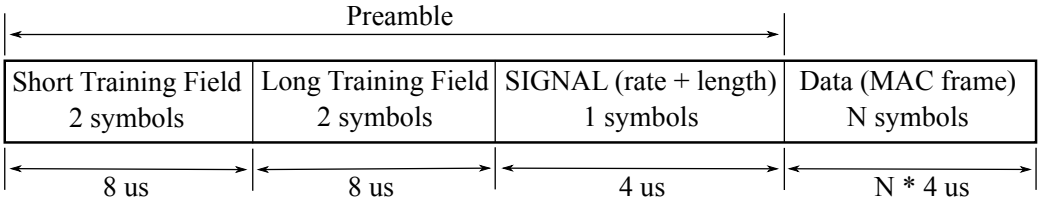


Figure 3: IEEE 802.11a/g packet format.

3.2.4. Experiments

In the experiments conducted in this paper, the signal packets consist of 200 byte-long IEEE 802.11g packets transmitted at 6 Mbit/s. The generated packets have payload containing random contents. The results are averaged over a larger number of packets (e.g., 100 or 1000).

Each packet contains both a preamble and a data payload (see Figure 3). Therefore, the duration of each packet is 328 µs, whereby the duration of the preamble is always 20 µs and the duration of the data is 308 µs. The preamble consists of a 2-symbol (or 8 µs) short training field. The following long training field (of the same length) is used for channel estimation, fine frequency offset estimation, and fine symbol timing offset estimation [36]. Finally, the third part of the packet preamble (the SIGNAL field) lasts 4 µs and encodes the packet length and bit rate.

Using this configuration, we conduct the following experiments and measure the corresponding packet loss statistics:

1. **Receiver sensitivity** experiments measure and compare how devices react to different transmission power levels. We increase the signal gain G_s from -80 dB to 0 dB in steps of 4 dB. At each step, we transmit 1000 packets and record packet loss statistics. The RF cable has a 60 dB attenuation to protect the DUT. In this experiment, no interference packet is added.
2. **Capture effect** experiments investigate packet loss during packet transmissions, as illustrated in Figure 4. Each experiment generates two packets: one packet defined as the *signal packet* and another packet defined as the *interference packet*. We craft precisely-timed packet collisions and measure whether the DUT experiences the capture effect, i.e., captures the signal packet despite the presence of an interference packet. We subject the DUTs to a range of colliding transmissions, varying Δt in increments of 1 µs. We transmit 1000 packets for each

setting and record packet loss statistics. We further distinguish between the following three cases:

- **Preamble capture effect:** The signal packet starts before or during the preamble of the interference packet.
- **Body capture effect:** The signal packet starts during the frame (body) of the interference packet.
- **Trailer capture effect:** The signal packet starts near the end of the interference packet.

Note that all the packet reception statistics reported in this paper pertain to signal packets. Interference packets are only used for emulating collisions.

To investigate the capture effect in the tested devices, we perform two experiments:

(a) **SIR dependency:** We vary the SIR from 0 dB to 36 dB by fixing $G_s = 0$ dB and varying the interfering signal gain from 0 dB to -36 dB in steps of 4 dB. We also vary the delay offset from -1 µs $\leq \Delta t \leq 10$ µs. For each configuration, we generate 100 packets and measure the packet loss of signal packets (i.e., a total of $12 \cdot 10^3$ probes per tested device).

(b) **Delay offset dependency:** We fix the signal gain $G_s = 1.0$ and $G_i = 0.1$, such that SIR = 20 dB. At these settings, both packets would be reliably received if they were sent without overlap. We vary the delay offset Δt from -5 µs to 335 µs and transmit 1000 packets for each configuration (i.e., a total of $340 \cdot 10^3$ probes per tested device), collecting packet loss statistics at the receiver, in order to find out whether signal capture behavior occurs at any delay offset across the whole length of a packet.

3. **Shortest Interframe Spacing (IFS) support:** The Interframe spacing (IFS) experiment measures the minimal delay between two packets (i.e., the time between the end of the first and the beginning of the second packet) such that both are successfully received by the DUT (i.e., the RX-to-RX turnaround time).

   This setup is similar to the trailer capture effect experiment except that the two packets arrive with equal signal strength, and the focus is on the time right after the first packet is already received. The packet reception rate of the second packet is conditioned on the first packet's

successful reception, i.e., both packets must be received and there is no "capture" of one competing signal over another.

4. **Truncated Packet Fingerprinting** experiments aim to characterize different devices based on their behavior in the presence of a specially crafted collision. We create an interference packet that contains a preamble, but no data afterwards. This truncated packet collides with a regular signal packet. We investigate the following variants of this scenario:
   (a) **Receiver state machine test**. We investigate how long it takes for a device to recover from such a bogus packet, i.e., at what time after the end of bogus packet can a valid packet be received again. We vary the delay offset from -5 µs to 335 µs to capture packet loss statistics across the full length of a signal packet.
   (b) **Consistency within device types**. We investigate, to which degree individual devices of the same type show measurable differences, or whether devices of the same type share the same characteristics.
   (c) **Wireless test validation**. To validate the previously described wired setup as a realistic alternative to over-the-air testing, we conduct the same experiment as for the intra-device type consistency test on a subset of devices, running the experiment in two configurations:
      i. Wired setup (as previously described)
      ii. Over-the air setup: We place the transmitter and receiver 40 cm apart and conduct the exact same experiment.

4. Experimental Results

In this section, we detail the results of our of experiments for each of the four DUTs listed in Table 1.

4.1. Receiver Sensitivity

In our first experiment, we evaluate DUT performance in terms of their receiver sensitivity. Specifically, we measure the packet loss ratio as a function of the transmission power gain G_s.

Subjecting all DUTs to test packets with varying signal gain G_s, we obtain the results shown in Figure 5. We can clearly identify and distinguish

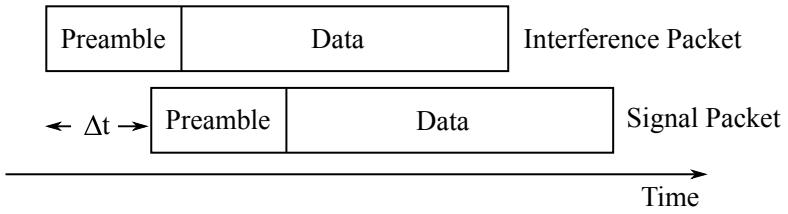


Figure 4: Packet transmissions for the capture effect.

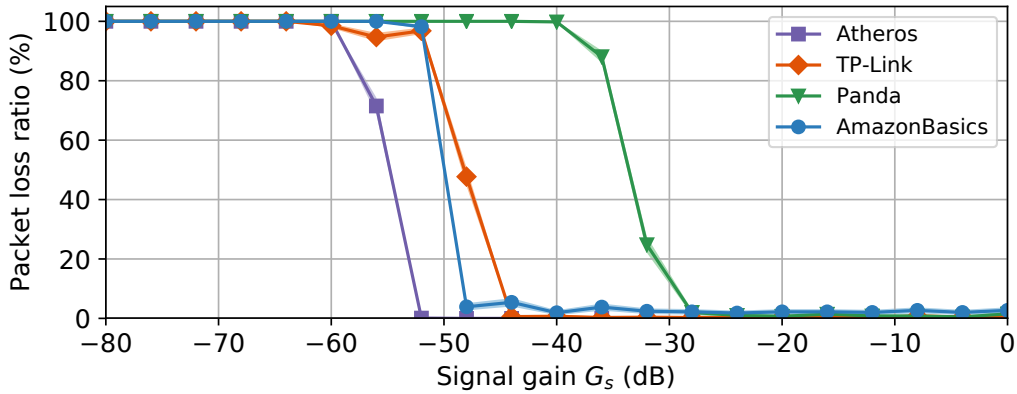
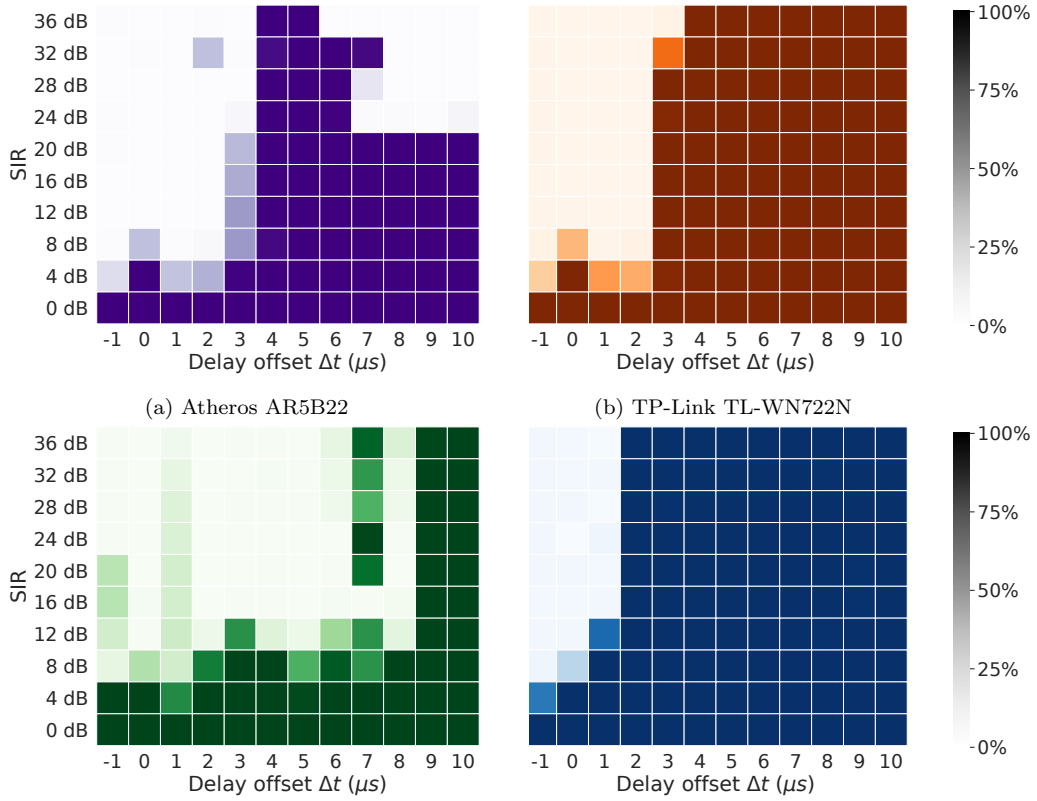


Figure 5: Receiver sensitivity of different Wi-Fi cards depending on the transmission power gain G_s.

the receiver sensitivity of different devices with great precision (the 95% confidence interval based on 1000 samples is tight (±0.47% around the mean), as indicated by the barely visible colored bands around the chart lines.

Interestingly, the devices exhibit markedly different sensitivity. In particular, the Atheros and TP-Link cards first start picking up packets at −60 dB and −56 dB, respectively, whereas the Panda card only starts picking up packets at −36 dB. Being able to distinguish these differences in receiver sensitivity allows us to compare devices regarding their performance in weak signal scenarios, such as strong attenuation occurring in densely developed areas.

We also note that in the range between −28 dB and 0 dB, packets are reliably picked up by all of the devices. In subsequent experiments involving packet collision, we use signal gains in this range, as we need to ensure that packets would have been received correctly if they were transmitted without overlap.



(a) Atheros AR5B22

(b) TP-Link TL-WN722N

(c) Panda Wireless PAU06

(d) AmazonBasics Wi-Fi 11N

Figure 6: Impact of SIR and packet delay on the capture effect in different Wi-Fi cards. Darker shade means higher packet loss.

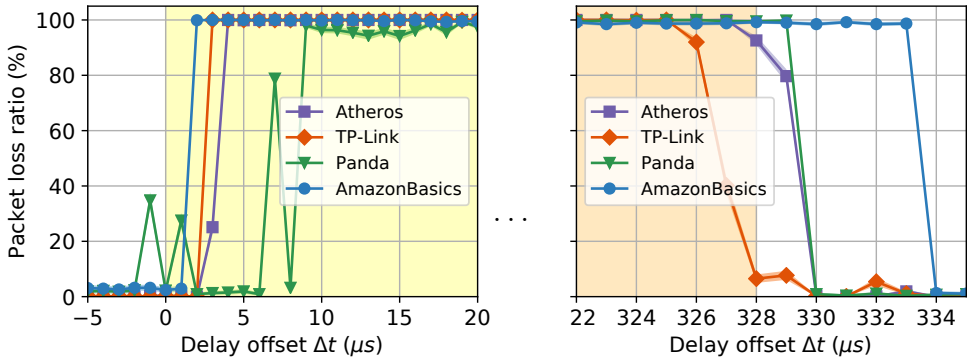
4.2. Capture Effect

We then apply our testbed to investigate the capture effect occurrence in different Wi-Fi devices. Successful capture in the presence of interference depends on different parameters, such as the SIR, and the delay offset Δt.

4.2.1. SIR

We first determine the power and delay conditions under which the capture effect occurs. Despite the lower amount of probes (100 per SIR and offset combination compared to 1000 in the subsequent experiment), we observe an average 95%-confidence interval of $\pm 1.3\%$ around the mean across all measurements.

Figure 6 shows the packet loss of signal packets at different SIRs and Δt.



(a) Packet loss ratio at the beginning of an interference packet.

(b) Packet loss ratio at the end of an interference packet.

Figure 7: Packet loss depending on the signal delay offset Δt relative to the beginning of an interference packet at 20 dB SIR. Figure (a) shows packet loss at low Δt, and Figure (b) around the end of the interference packet. Yellow and orange background indicates collision with the preamble and payload of the interference packet, respectively.

This graph shows bright spots for all parameter configurations with reliable reception (low packet loss) of the signal packet and darker spots wherever the packet loss is high.

In Figure 6, we observe that the devices behave quite differently, i.e., they experience the capture effect within different boundary conditions. For example, the TP-Link manages to receive the signal packet only if the SIR is above 4 dB, but, independently of the SIR, only up to a delay of 3 µs. In contrast, the Panda Wireless device requires a higher SIR for successful reception, but is capable of receiving the signal up to 8 µs after the interference packet, while showing a greater variance in its behavior overall.

In general, the data shows that the capture effect requires a certain minimum SIR and gives reason to assume that after a certain Δt, the capture effect does not occur any more – independent of the SIR. This may be due to the receiver already locking on to a signal during the preamble, based in individual vendor implementation.

The Atheros AR5B22 card is an exception to this observation. In Figure 6(a), we observe that the Atheros card stops capturing new packets – independent of the chosen SIR – at 4 µs, but then resumes capture above a certain SIR threshold. To confirm this finding, we conduct further related experiments in Section 4.2.2.

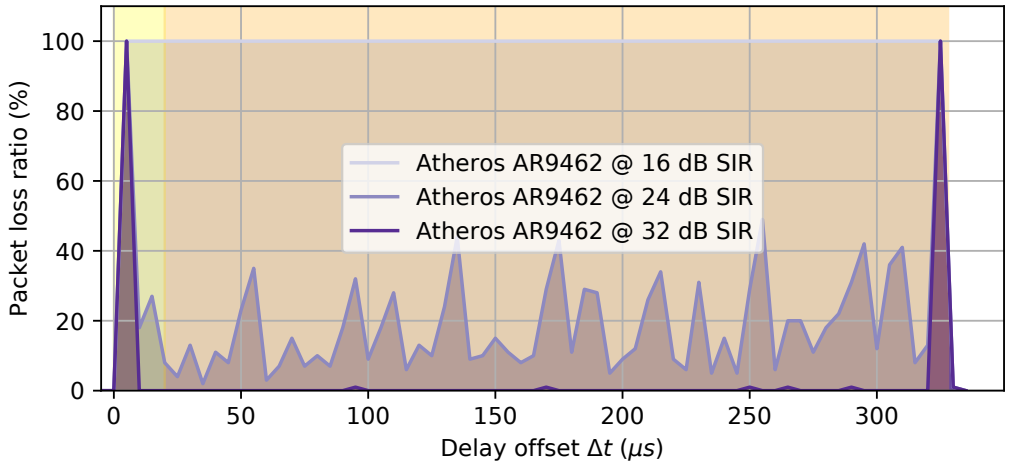


Figure 8: The Atheros AR9462 chipset captures new packets even while it is already receiving a packet body, if the SIR is sufficiently high. The graph shows packet loss for different packet delay offsets and SIR.

4.2.2. Delay Offset

The previous experiments showed that after a certain delay offset, the capture effect does no longer occur in several of the devices. We investigate whether this result remains consistent throughout the whole range of possible delay offsets, i.e., for all possible overlaps between interference and signal packets.

Figure 7(a) shows the capture effect of different cards for low Δt. We observe that each tested device has a characteristic capture behavior, and transitions to 100% packet loss after a certain delay offset. This result indicates that the capture effect occurs only if the delay offset is small, and implies that the receiver locks on to the packet after it receives the first few bits of a packet. Then, receivers typically cannot detect another packet until the packet transmission ends. This result shows that the delay offset plays a critical role in the packet loss of the signal packets.

Indeed, this behavior remains consistent until the end of the interference packet. However, as shown in Figure 7(b), we can observe that devices again behave differently after receiving an incoming packet. Some devices exhibit the capture effect shortly before the interference packet ends (at 328 µs), while others cannot immediately switch to receive the signal packet after the end of the interference packet. We believe this is again due to different

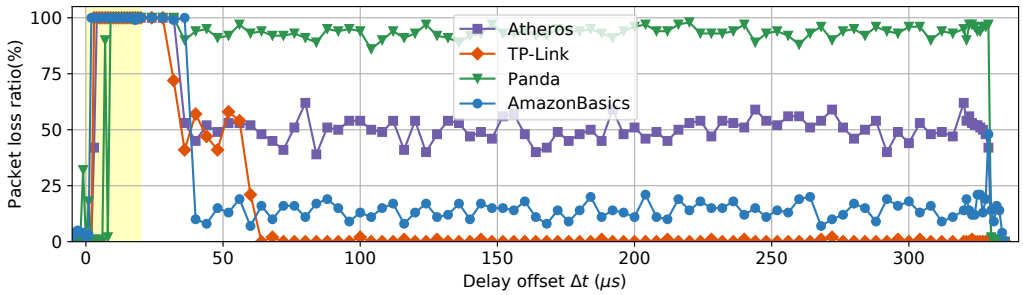


Figure 9: To fingerprint Wi-Fi chipsets, we generate collisions between signal packets and specially crafted truncated packets containing only a preamble, and measure the DUT's packet loss. Note that no actual signal collision occurs after the end of the preamble, i.e., packet loss at $\Delta t \geq 20$ µs is only a result of the receiver's physical layer state machine implementation.

physical layer implementations of the standard in the various chipsets.

Coming back to the Atheros AR9642 chipset, we run additional tests on the Atheros AR5B22 card only, varying the SIR from between 16, 24, and 32 dB SIR, and testing the whole range of Δt from the beginning of the interferer preamble at $\Delta t = 0$ µs until the end of the packet (at $\Delta t = 328$ µs) in steps of 5 µs. Indeed, as shown in Figure 8, capture is possible not only during the whole length of the preamble, but along the full length of the interference packet, if the SIR is strong enough. In other words, the Atheros AR9462 chipset seems to implement body capture above a certain SIR[3]. We note that this behavior can be found in the Atheros AR9462 chipset, but not in the AR9271 chipset of the TP-Link device that we tested.

4.3. Shortest Interframe Spacing (IFS) support

The Interframe spacing (IFS) experiment measures minimal delay between two incoming packets (end of the first and beginning of the second packet) such that both are successfully received by the DUT (a.k.a. RX-to-RX turnaround time). We send two 200 byte test packets spaced from each other by a range from -2.5 µs (i.e., a slight overlap) to 12.5 µs apart with 0.5 µs steps. We craft these pairs of packets close to each other to actively detect the minimal RX-to-RX turnaround time of a device. In this exper-

[3]This confirms a recent finding by Khorov et al. [15], who identified the body capture effect in the similar, but not identical, AR9485.

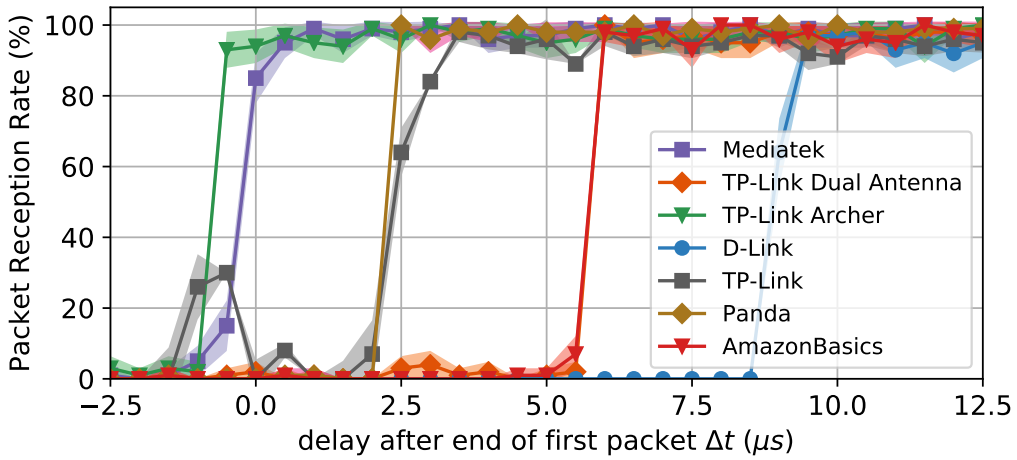


Figure 10: The reception rate of the second packet is measured by only counting it as a success if the first packet was also successfully received. Hence, the transitions from low to high packet reception rates represent the time after which the device is ready to receive another packet. Each point is measured 100 times and 95% confidence intervals are tight.

iment, we tested additional cards, namely the Mediatek MT7612UN, TP-Link TL-WN822N (dual antenna), Archer T2UH, and D-Link DWL-G122 (see Figure 10).

The IFS experiments for a specific device are reproducible well within sub-microsecond precision. The non-zero reception rate (Mediatek and TP-Link Archer) at negative delays (indicating a packet collision) from -1 µs to 0 µs could be attributed to 802.11's error correction codes. Therefore, Mediatek and TP-Link Archer can literally receive packets back-to-back with zero IFS. On the other hand, worst performing D-Link has RX-to-RX turnaround time just below 10 µs.

For reference, 10 µs is the short interframe spacing (SIFS) required for the IEEE 802.11g standard which uses the ERP-OFDM physical layer [37, p. 2328]. In other words, while all cards are clearly standards-compliant based on these results, we can measure that some devices have a RX-to-RX turnaround time below the SIFS, which can be used to fingerprint these devices. For example, one could send a pair of test packets with only 1 µs spacing between them, and if both are succesfully received, they are likely to be of the Mediatek or TP-Link Archer device type (cf. Figure 10) rather than any of the other tested devices, which only start receiving after larger

interframe spacings.

4.4. Truncated Packet Fingerprinting

4.4.1. Receiver State Machine Test

Wi-Fi devices implement the physical layer as a state machine, i.e., the receiver has one state to detect the packet preamble and another state to receive the packet frame [38]. We next investigate whether different devices implement such state machines in different ways by examining their response to certain crafted signals.

The experiment setup is the same as in Section 4.2.2, except that the interference is not a valid Wi-Fi packet this time. Instead, we only transmit a preamble and truncate the packet data (MAC frame). Generally, if the signal packet arrives after the end of the interference preamble (without a frame) and experiences loss, such packet loss is not caused by a collision (as there is no data to collide with). Instead, the reason for the packet loss is that the receiver is in a state that does not allow it to capture a new packet.

Figure 9 depicts the results of this experiment. When $\Delta t \leq 20$ µs, the interference packet collides with the preamble of the truncated packet. The packet loss ratio jumps to 100% after a few microseconds delay offset, in the same chipset-specific way that we observed in the capture effect experiment. This shows that the truncated packet colliding with the preamble of the signal packet results in the same capture behavior as described in the previous section.

Once Δt exceeds 20 µs, the delay offset is such that the signal preamble would collide with the data field of the interference packet. However, since the truncated interference packets have no data field, there is no data to collide with. Interestingly, the behavior of the DUTs in this scenario varies considerably: Whereas after 30 to 50 microseconds the TP-Link and AmazonBasics cards recover to a state in which they can capture new packets, the Atheros card experiences about 50% packet loss for the whole duration of the non-existent interference packet's data, and the Panda card experiences near total packet loss until the nominal end of the expected packet duration. This demonstrates that the tested cards have widely different state machine implementations, especially regarding the transition from the state of packet preamble detection to the state of packet reception and back.

Probing devices with such specially crafted signals allows for physical layer fingerprinting of the devices based on their chipset implementation. Such way of fingerprinting could be used, for instance, as an additional factor

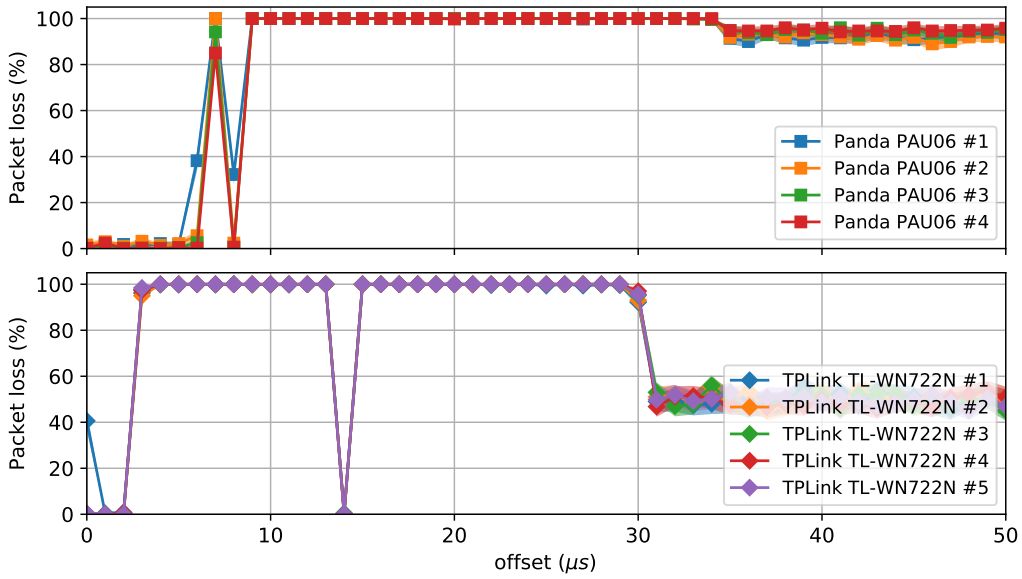


Figure 11: Measuring multiple devices of the same type yields very consistent results.

in authentication scenarios in which the physical device identity is critical, confirming or rejecting that communication is coming from the desired device without alerting the application layer. While in-depth design of such an authenticity challenge i not the focus of this work, one could devise a decision tree of tests at specific discriminating offset times, which allow categorization without having to run a time-consuming lab test. For example, referring to Figure 11, one could probe a device with a truncated packet test timed at exactly 5 µs offset. If the packet loss is high, it is likely to be a TPLink device, whereas if it is low it is likely to be a Panda device. This test can be repeated over multiple times, and at different offset locations, until the desired certainty is obtained. Thus, testing for the presence of absence of known devices can be performed with much less probing effort than the experiments tracing previously unknown device responses in this work.

4.4.2. Consistency within Device Types

As shown in the previous section, device types of all the tested devices can clearly be distinguished by a characteristic response curve shown in Figure 9. To investigate the degree to which a single device is representative for a device type, i.e., how consistently a device type behaves the same, additional tests

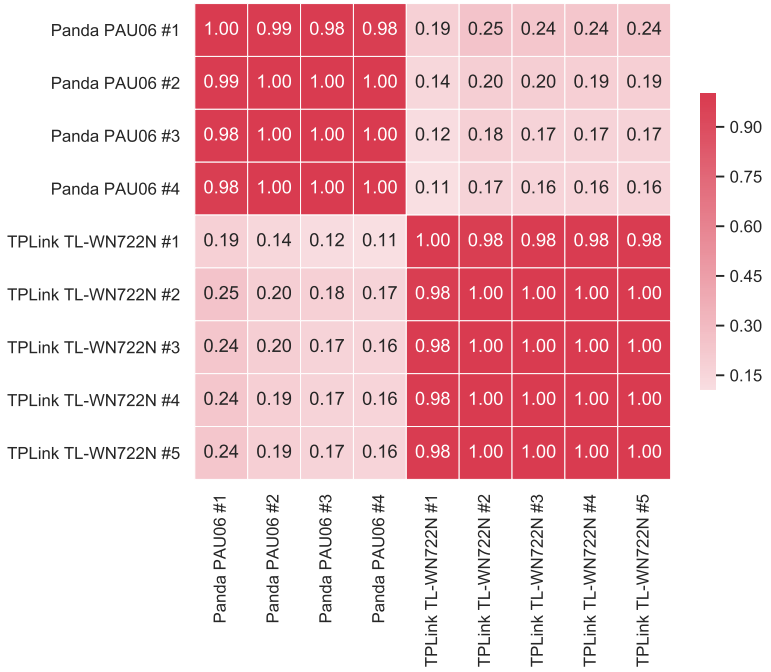


Figure 12: The correlation between devices of the same type is consistently high while it is lower than 0.25 for devices of different type.

were run on a subset of the previously tested device, namely the TP-Link TL-WN722N and the Panda Wireless PAU06. From these two device types, multiple devices (4 and 5, respectively) were subjected to an additional test to determine device type consistency. As a lot of the feature differentiation visible in Figure 9 occurs within the first 50 µs, we measured each device using 500 packets for each delay offset from 0 to 50 µs in steps of 1 µs. The result – visualized in Figure 11 – shows a high degree of consistency between devices.

This means that the fingerprints identified in the previous section can indeed be considered reliable indicators of a certain device type. However, identification of individual devices of the same type cannot confidently be deducted from these results, as all measured devices of the same type behave the same with mostly overlapping 95% confidence intervals (which are shown as translucent areas around the graph in Figure 11, but are hard to distinguish because the data overlaps tightly).

Further analysis shows high consistency within a device type as evidenced

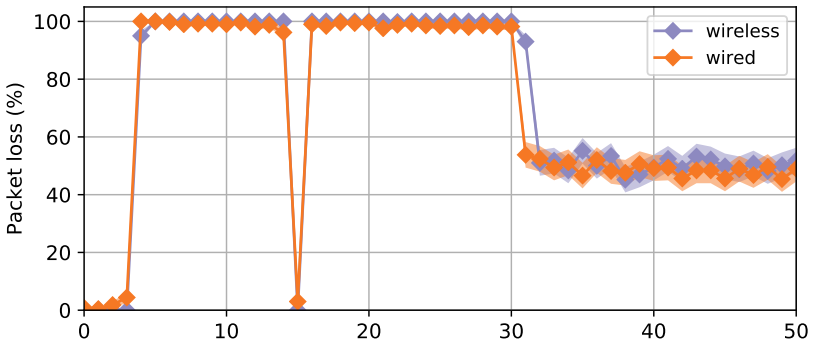


Figure 13: Wired and wireless test setup produce very consistent results for experiments that test the receiver state machine. In this figure, the experiment from Figure 11 is repeated with both a wired and a wireless TPLink device, with very consistent results (98% covariance).

by the correlation matrix in Figure 12: Whereas each device type shows correlation coefficients greater than 0.98, whereas the coefficients of devices of differing type never exceed 0.25.

4.4.3. Wireless Test Validation

Running the previous test in both the wireless as well as the wired configuration shows that card behavior is quasi identical (around 98% covariance between the wired and wireless test, see Figure 13) which is in the same range as devices of the same type are between each other, as shown previously. We conclude that assuming sufficient shielding of the overall experimental setup, both configurations can be used interchangeably for experiments that test for logical behavior of a DUT. However, it should be noted that experiments that are highly sensitive to the absolute received signal strength, such as the receiver sensitivity test results of Section 4.1, will produce different results based on the wire attenuation chosen in the wired case, and the distance and orientation of transmitter and DUT to each other in the wireless case, making them harder to compare easily.

5. Conclusion

In this paper, we present an SDR-based testbed that achieves precise parameter control suitable for wireless device testing. We use the testbed to evaluate a range of Wi-Fi cards regarding different performance aspects,

using both a wired configuration which allows for precise power control, as well as a wireless mode which is easier to set up and capable of measuring devices without a coaxial cable-compatible antenna port. In particular, the receiver sensitivity of the cards varies by as much as 20 dB. We also investigate the capture effect on IEEE 802.11 networks by designing experiments that allow us to capture differences emerging on the scale of microseconds. Thanks to the precise parameter control made possible by the testbed, we provide quantitative analysis on the impact of packet arrival time, SIR and manufacturer-specific implementation on the occurrence of the capture effect. Notably, among the four different Wi-Fi cards tested, capture of the preamble varies by as much as 7 µs in terms of the delay offset. The experiments further show that some Wi-Fi cards exhibit body capture effects while others do not, thus cross-validating findings from [15].

Our work shows that it is valuable to compare multiple cards at high temporal resolution, because manufacturers differently implement physical layer features that are not precisely defined in the standard. Thus, one should not assume that implementation characteristics of a specific Wi-Fi card are generally applicable to all Wi-Fi cards. This finding is especially important when developing analytical and simulation models of Wi-Fi networks.

Another interesting finding is that two of the tested Wi-Fi chipsets appear to return to the preamble detection state earlier than the standard defines. This specific feature may potentially have performance benefits in congested networks, allowing them to detect PHY preambles more aggressively.

The experimental results of this paper can further serve to fingerprint the tested devices, especially based on their physical layer responses to truncated interference packets. Since these responses are hardwired into the chipset, the fingerprints may be of interest both as an additional authentication factor, as well as for covert device tracking devices (circumventing higher layer anonymization).

Indeed, open issues in previous works [6] were addressed in this work by measuring multiple devices of the same type: Testing multiple devices of the same type revealed a high behavioral consistency of the fingerprint resulting from this experiment. Correlation coefficients of 98% and more between individual tested devices of the same type suggest that these fingerprints can indeed be used to identify a device type based on its receiver response to a precisely timed truncated packet collision at specific delay offset times. The fingerprint achieved by truncated packet fingerprinting should thus be considered a *device type* fingerprint, rather than an individual *device* fingerprint.

The security implications of being able to distinguish devices by their chipset (or other inherent, physical layer features) is of great utility when device authenticity is important and the risk of device impersonation needs to be addressed and mitigated. In such a scenario, the fingerprints of legitimate hardware deployed in an organization could be recorded, and devices in the field could be challenged to respond to a number of crafted signals generated by the testbed. Our assumption is that typical 802.11 NICs cannot easily conceal or manipulate this low-layer state machine-based response, and thus legitimate devices would respond in a pattern that matches the pre-recorded fingerprint, whereas rogue devices of a different manufacturer would fail this challenge.

Much additional work can be performed based on the testbed proposed in this paper, as it provides a flexible platform for any kind of wireless experimentation and is not limited to a specific communication protocol. For instance, one could investigate the impact of the bit rate and the modulation on the device behavior, and whether these variables can also help deriving device-specific characteristics. Aside from further expansion on performance characterization of Wi-Fi cards (including different 802.11 variants), one could expand the scope of this work to investigate low-layer performance, privacy, and security characteristics of other popular wireless protocols, such as Bluetooth. Future work could also involve expanding the testbed to bidirectional communication testing, which opens up a new range of methods, e.g., fingerprinting based on response delays.

Acknowledgements

This work is funded in part by NSF under grant CNS-1409053.

References

[1] C. Sulhoff, FCC Takes Action To Encourage Increased Investment And Deployment In The 3.5 GHz Band (2018).
URL `https://www.fcc.gov/document/fcc-acts-increase-investment-and-deployment-`

[2] N. Grace, FCC Proposes More Spectrum For Unlicensed Use (2018).
URL `https://www.fcc.gov/document/fcc-proposes-more-spectrum-unlicensed-use`

[3] octoScope Inc., Wireless Personal Testbeds (2019).
URL `http://octoscope.com/English/Products/Ordering/index.html`

[4] Rohde & Schwarz GmbH, Test Systems & Accessories (2019).
URL https://www.rohde-schwarz.com/us/products/test-and-measurement/wireless-c

[5] National Instruments, Simple Solutions to Complex Problems (2019).
URL http://www.ni.com/en-us/shop.html

[6] L. Xin, J. K. Becker, S. Gvozdenovic, D. Starobinski, Benchmarking the physical layer of wireless cards using software-defined radios, in: Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2019, pp. 271–278.

[7] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, P. Steenkiste, Enabling MAC Protocol Implementations on Software-de ned Radios, NSDI'09 Proceedings of the 6th USENIX symposium on Networked systems design and implementation (2009) 91–105.

[8] P. Di Francesco, S. McGettrick, U. K. Anyanwu, J. C. O'Sullivan, A. B. MacKenzie, L. A. DaSilva, A Split MAC Approach for SDR Platforms, IEEE Transactions on Computers 64 (4) (2015) 912–924. doi:10.1109/TC.2014.2308197.
URL http://ieeexplore.ieee.org/document/6747972/

[9] B. Bloessl, A. Puschmann, C. Sommer, F. Dressler, Timings matter, in: Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH '14, ACM Press, New York, New York, USA, 2014, pp. 57–64. doi:10.1145/2643230.2643240.
URL http://dl.acm.org/citation.cfm?doid=2643230.2643240

[10] T. Schmid, O. Sekkat, M. B. Srivastava, An experimental study of network performance impact of increased latency in software defined radios (2007) 59doi:10.1145/1287767.1287779.

[11] J. S. Park, H. Yoon, B. J. Jang, SDR-based frequency interference analysis test-bed considering time domain characteristics of interferer, International Conference on Advanced Communication Technology, ICACT 2016-March (2016) 517–521. doi:10.1109/ICACT.2016.7423454.

[12] P. Murphy, Design, implementation and characterization of a cooperative communications system, Ph.D. thesis, Rice University (2010).
URL http://warp.rice.edu/w/MurphyPhDThesis

[13] C. Hunter, Distributed protocols for signal-scale cooperation, Ph.D. thesis, Rice University (2012).
URL http://warp.rice.edu/trac/wiki/HunterPhDThesis

[14] J. Cache, Fingerprinting 802.11 implementations via statistical analysis of the duration field, Uninformed.org 5 (2006).

[15] E. Khorov, A. Kureev, I. Levitsky, A. Lyakhov, Testbed to Study the Capture Effect: Can We Rely on this Effect in Modern Wi-Fi Networks, in: 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), IEEE, 2018, pp. 1–5. doi:10.1109/BlackSeaCom.2018.8433688.
URL https://ieeexplore.ieee.org/document/8433688/

[16] W. Liu, E. De Poorter, J. Hoebeke, E. Tanghe, W. Joseph, P. Willemen, M. Mehari, X. Jiao, I. Moerman, Assessing the Coexistence of Heterogeneous Wireless Technologies With an SDR-Based Signal Emulator: A Case Study of Wi-Fi and Bluetooth, IEEE Transactions on Wireless Communications 16 (3) (2017) 1755–1766. doi:10.1109/TWC.2017.2654256.

[17] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18 (3) (2000) 535–547. doi:10.1109/49.840210.
URL http://ieeexplore.ieee.org/document/840210/

[18] S. Ray, D. Starobinski, J. B. Carruthers, Performance of wireless networks with hidden nodes: a queuing-theoretic analysis, Computer Communications 28 (10) (2005) 1179–1192. doi:10.1016/j.comcom.2004.07.024.
URL https://linkinghub.elsevier.com/retrieve/pii/S0140366404002750

[19] L. Xin, D. Starobinski, G. Noubir, Cascading denial of service attacks on Wi-Fi networks, in: 2016 IEEE Conference on Communications and Network Security (CNS), IEEE, 2016, pp. 91–99.

doi:10.1109/CNS.2016.7860474.
URL http://ieeexplore.ieee.org/document/7860474/

[20] L. Xin, D. Starobinski, Mitigation of Cascading Denial of Service Attacks on Wi-Fi Networks, in: 2018 IEEE Conference on Communications and Network Security (CNS), IEEE, Beijing, China, 2018, pp. 1–9. doi:10.1109/CNS.2018.8433124.
URL https://ieeexplore.ieee.org/document/8433124/

[21] D. Malone, K. Duffy, D. Leith, Modeling the 802.11 Distributed Coordination Function in Nonsaturated Heterogeneous Conditions, IEEE/ACM Transactions on Networking 15 (1) (2007) 159–172. doi:10.1109/TNET.2006.890136.
URL http://ieeexplore.ieee.org/document/4100720/

[22] Z. Hadzi-Velkov, B. Spasenovski, Saturation throughput - delay analysis of IEEE 802.11 DCF in fading channel, in: IEEE International Conference on Communications, 2003. ICC '03., Vol. 1, IEEE, IEEE, 2003, pp. 121–126. doi:10.1109/ICC.2003.1204154.
URL http://ieeexplore.ieee.org/document/1204154/

[23] I. Tinnirello, G. Bianchi, Yang Xiao, Refinements on IEEE 802.11 Distributed Coordination Function Modeling Approaches, IEEE Transactions on Vehicular Technology 59 (3) (2010) 1055–1067. doi:10.1109/TVT.2009.2029118.
URL http://ieeexplore.ieee.org/document/5191039/

[24] J. Robinson, T. Randhawa, Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function, IEEE Journal on Selected Areas in Communications 22 (5) (2004) 917–928. doi:10.1109/JSAC.2004.826929.
URL http://ieeexplore.ieee.org/document/1303760/

[25] M. Durvy, O. Dousse, P. Thiran, Modeling the 802.11 Protocol Under Different Capture and Sensing Capabilities, in: IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, IEEE, 2007, pp. 2356–2360. doi:10.1109/INFCOM.2007.280.
URL http://ieeexplore.ieee.org/document/4215862/

[26] P. Chatzimisios, A. Boucouvalas, V. Vitsas, Performance analysis of IEEE 802.11 DCF in presence of transmission errors, in: 2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577), Vol. 7, IEEE, IEEE, 2004, pp. 3854–3858. doi:10.1109/ICC.2004.1313274.
URL http://ieeexplore.ieee.org/document/1313274/

[27] F. Daneshgaran, M. Laddomada, F. Mesiti, M. Mondin, M. Zanolo, Saturation throughput analysis of IEEE 802.11 in the presence of non ideal transmission channel and capture effects, IEEE Transactions on Communications 56 (7) (2008) 1178–1188. doi:10.1109/TCOMM.2008.060397.
URL http://ieeexplore.ieee.org/document/4568459/

[28] C. Ware, J. Judge, J. Chicharo, E. Dutkiewicz, Unfairness and capture behaviour in 802.11 adhoc networks, in: 2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record, Vol. 1, IEEE, IEEE, 2000, pp. 159–163. doi:10.1109/ICC.2000.853084.
URL http://ieeexplore.ieee.org/document/853084/

[29] S. Ganu, K. Ramachandran, M. Gruteser, I. Seskar, J. Deng, Methods for restoring MAC layer fairness in IEEE 802.11 networks with physical layer capture, in: Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality - REALMAN '06, ACM, ACM Press, New York, New York, USA, 2006, p. 7. doi:10.1145/1132983.1132986.
URL http://portal.acm.org/citation.cfm?doid=1132983.1132986

[30] ORBIT Lab, Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) (2019).
URL https://www.orbit-lab.org/

[31] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, Y. Choi, An experimental study on the capture effect in 802.11a networks, in: Proceedings of the the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WinTECH '07, ACM, ACM Press, New York, New York, USA, 2007, p. 19. doi:10.1145/1287767.1287772.
URL http://portal.acm.org/citation.cfm?doid=1287767.1287772

[32] ORBIT Lab, Hardware (2016).
URL https://www.orbit-lab.org/wiki/Hardware

[33] Ramsey Electronics, STE3500 (2019).
URL http://www.ramseyelectronics.com/product.php?pid=14

[34] TCPDUMP & LIBPCAP, TCPDUMP & LIBPCAP (2019).
URL https://www.tcpdump.org/

[35] B. Bloessl, M. Segata, C. Sommer, F. Dressler, An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio, in: Proceedings of the second workshop on Software radio implementation forum, ACM, 2013, pp. 9–16.

[36] The Mathworks Inc., WLAN Packet Structure (2019).
URL https://www.mathworks.com/help/wlan/ug/wlan-packet-structure.html#buytqq7

[37] IEEE Standards Association, 802.11-2016 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Sp, IEEE, 2016. doi:10.1109/IEEESTD.2016.7786995.
URL https://ieeexplore.ieee.org/servlet/opac?punumber=7786993

[38] L. Xin, D. Starobinski, Cascading Attacks on Wi-Fi Networks with Weak Interferers, in: Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWIM '18, ACM Press, New York, New York, USA, 2018, pp. 255–258. doi:10.1145/3242102.3242142.
URL http://dl.acm.org/citation.cfm?doid=3242102.3242142