Enhancing the performance of Wi-Fi system by exploiting physical layer information

Xie, Yaxiong

2016

Xie, Y. (2016). Enhancing the performance of Wi-Fi system by exploiting physical layer information. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/69395>

<https://doi.org/10.32657/10356/69395>

NANYANG TECHNOLOGICAL UNIVERSITY

Enhancing the Performance of Wi-Fi System by Exploiting Physical Layer Information

By

Yaxiong Xie

Supervisor: Dr. Mo Li

School of Computer Science and Engineering

This thesis is submitted to Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Abstract

Wi-Fi network has been ubiquitous nowadays and has changed our lifestyle. As a communication system, Wi-Fi delivers more than 50% of IP traffic. Consequently, the demand for higher transmission capacity has been increasing continuously and rapidly. The wireless spectrum for Wi-Fi communication is, however, finite. Therefore, one important research direction is to fully utilize the limited resources and at the same time improve the transmission throughput for Wi-Fi network. On the other hand, a lot of emerging applications, e.g., indoor navigation, human tracking, device free gesture control, are built on top of existing commercial Wi-Fi infrastructures to provide a variety of functionalities except communication. All those applications rely on Wi-Fi's capability of sensing the physical world: strong sensing capability will significantly improve the performance of existing applications and extend the scope of potential applications. Therefore, another important research direction is to enhance the sensing capability of existing Wi-Fi infrastructure. This thesis focuses on these two directions and exploits rich information in the PHY layer to build a better Wi-Fi system that has higher communication speed and stronger sensing capability.

We observe that modern wideband Wi-Fi communication has unevenly distributed bit BERs in a packet because of the frequency selective fading. Based on such an observation, we propose UnPKT, a system that can unequally protect Wi-Fi packet bits according to their BERs. By doing so, we can best match the effective transmission rate of each bit to channel condition, and improve throughput. We derive an accurate relationship between the frequency selective channel condition and the decoded packet bit BERs, all the way through the complex 802.11 PHY layer. A cluster-based protection scheme is proposed to protect packet bits using different MAC-layer FEC redundancies based on bit-wise BER estimation to augment wide band 802.11 transmissions. UnPKT is software-implementable and compatible with the existing 802.11 architecture. Extensive evaluations based on Atheros 9580 NICs and GNU-Radio platforms show

the effectiveness of our design. UnPKT can achieve a significant goodput improvement over state-of-the-art approaches.

When sensing the physical world using Wi-Fi, power delay profile is widely used in motion- or localization-based applications as it characterizes multipath channel features. Recent studies show that the power delay profile may be derived from the CSI traces collected from commodity WiFi devices, but the performance is limited by two dominating factors. The resolution of the derived power delay profile is determined by the channel bandwidth, which is however limited on commodity WiFi. The collected CSI reflects the signal distortions due to both the channel attenuation and the hardware imperfection. A direct derivation of power delay profiles using raw CSI measures, as has been done in the literature, results in significant inaccuracy. Therefore, we build Splicer, a software-based system that derives high resolution power delay profiles by splicing the CSI measurements from multiple WiFi frequency bands. A set of key techniques has also been proposed to separate the mixed hardware errors from the collected CSI measurements. Splicer substantially improves the accuracy in profiling multipath characteristics, reducing the errors of multipath distance estimation to be less than 2m. Splicer can immediately benefit upper-layer applications. Our case study with recent single-AP localization achieves a median localization error of 0.95m.

Acknowledgments

First and foremost, I would like to express my sincere appreciation to my PHD supervisor, Dr. Mo Li, for his support, patience and guidance. When join WANDS as fresh PHD student, I got no experience on doing research and lack the knowledge base to conduct cutting-edge research. I really want to thank Dr. Li for the career advice he gives me which helps me find my real research interests, the research skills he teaches me that facilitate my capability to carry out impactful research and the patience and freedom he put on me that allow me to establish my knowledge base on wireless communication before I can come up with my own idea to update the stat-of-the-art of the specific research domain. Dr. Li is also a nice guy who really cares for his student. I feel lucky to have such a mentor with me during the long journey pursuing my PHD degree.

I am also grateful to my lab-mates in WANDS: Yuanqing Zheng, Zhenjiang Li, Wan Du, Zhidan Liu, Pengfei Zhou, Jiajue Ou, Cheng Li, Wenwei Chen, Yuxiao Hou, Shiqi Jiang, Jansen Liando, Chu Cao, Huanle Zhang, Panrong Tong. I will always remember the days we fight together for catching the deadlines, the insightful discussions we have had about diverse research topics. In particular, I want to thank Dr. Zhenjiang Li for his generous help. I do learnt a lot from our cooperation.

I want to thank my collaborators, Professor Kyle Jamieson and Professor Jie Xiong, for their help in my research projects. Their insightful opinions would always lead me to the correct direction when I met problem in my research projects. Their valuable comments help to greatly enhance the quality and readability of the writing. It is really my honor to work with those respectful Professors.

Finally but most importantly, I would like to thank my family for the support they give me and in particular, I must acknowledge my fiancée, Xiaoxiao Tang. It is because of your love and support that I can finally get here.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	vi
1 Introduction	1
1.1 Exploiting CSI to improve link throughput of Wi-Fi	4
1.2 Exploiting CSI to improve sensing resolution of Wi-Fi	6
1.3 Organization of This Thesis	9
2 Related Work	10
2.1 IEEE 802.11 standards	10
2.2 Towards robust Wi-Fi transmissions	12
2.3 Towards precise Wi-Fi sensing	15
2.4 CSI tool	17
3 Augmenting Wide-band 802.11 Transmissions	19
3.1 Design	19
3.1.1 Motivation and Observations	19
3.1.2 Design Overview	23
3.1.3 Data rate selection	24
3.1.4 BER estimator	25
3.1.5 Packet protector	30
3.2 Evaluation	33
3.2.1 BER estimation evaluation	33
3.2.2 Trace-driven evaluation	35

3.2.3	Experimental Results	37
3.3	Discussion and limitation	39
4	Improving the Sensing Resolution of COTS Wi-Fi Devices	40
4.1	Design	40
4.1.1	Principle of CSI Splicing	40
4.1.2	CSI splicing in practice	43
4.1.3	Sources of CSI measurement errors	46
4.1.4	Phase error correction	48
4.1.5	CSI phase splicing	53
4.1.6	CSI amplitude splicing	55
4.1.7	Battle the coherence time constraint	55
4.2	Evaluation	59
4.2.1	Experimental setup	59
4.2.2	Results	60
4.2.3	Case study: indoor localization	65
4.3	Discussion and limitation	69
5	Conclusion and Future Work	70
5.1	Conclusion	70
5.2	Future Work	71
	Appendix	74
A	Author's Publications	74

List of Figures

1.1	Data from 20 MHz 802.11n wireless channels. (a) Signal-to-noise ratio (SNR) across different subcarriers, measured over eight received packets; (b) Bit error rates (BERs) across different subcarriers; (c) The BER of decoded data bits, by position in the packet.	2
1.2	Transformation from a channel frequency response to the power delay profile. (a) A channel frequency response, where f_0 , Δf , and B represent starting frequency, frequency sampling resolution, and bandwidth, respectively; (b) Derived power delay profile, where τ_0 and $\Delta\tau$ represent the propagation delay of LoS path and the power delay profile resolution, respectively.	6
3.1	(a) PRRs of all 166 links in three test-beds; (b) CDF of the PRR.	20
3.2	Measured BER of each bit position within a 1000-byte packet. We depict the first 3500 packet bits for the sake of a clear presentation.	21
3.3	(a) Average BER of each bit position in one period; (b) BER of the codeword bits in one OFDM symbol.	22
3.4	All the UnPKT operations are abstracted in the PTC layer.	23
3.5	Illustration of the decoding bit errors and the codeword bit errors.	26
3.6	$P_{group}(k, \eta)$ changes with η for the different convolutional coding rates (a) 1/2; (b) 2/3; (c) 3/4; (d) 5/6.	28
3.7	Puncturing technique to achieve different coding rates in 802.11.	29
3.8	Burst length (bits) under different data rates.	31
3.9	Clusters from multiple OFDM symbols are grouped together for BER estimation and protected using RS code.	32
3.10	Cluster-level BER ratio between estimations and measurements.	34
3.11	Computation delay of the BER estimation in UnPKT.	35

3.12	CDF of the normalized per-link goodput.	38
3.13	Goodput gain breakdown of UnPKT.	38
3.14	CDF of transmissions to delivery a packet.	38
3.15	(a) Fractions of data bit-rates used in different approaches; (b) Communication overhead of each approach.	39
4.1	Power delay profiles derived using two CSI with an amplitude offset of 7dB. . .	41
4.2	Power delay profiles derived using two CSIs with raw phases and average amplitude of the CSI traces in Figure 4.1.	42
4.3	CSI amplitude measurements. Raw CSI amplitudes from (a) one 40MHz 802.11n band and (b) three 20MHz bands; Amplitudes after the offset removal from (c) the 40MHz band and (d) three 20MHz bands.	44
4.4	CSI phase measurements. Raw CSI phases from (a) the 40MHz band and (b) three 20MHz bands; Phases after the offset removal from (c) the 40MHz band and (d) three 20MHz bands. The CSI traces in this Figure are the same with Figure 4.3.	45
4.5	Illustration of signal processing in 802.11.	46
4.6	Phase differences cross sub-carriers and the distribution of $\Delta\lambda_b$. (a) The phase differences $k \cdot \Delta\lambda_b$ of 8 randomly CSI pairs; (b) Distribution of slope $\Delta\lambda_b$ calculated from 180 CSIs.	49
4.7	The power delay profile similarities for two pairs of WiFi bands when ϵ varies from -0.1 to 0.1.	51
4.8	CSI phases after the removal of λ_s and λ_o	52
4.9	Spliced CSI phase refinement.	53
4.10	Empirical investigation of window size l	54
4.11	CSI splicing result. (a) The spliced CSI amplitude; (b) The spliced CSI phase. .	56
4.12	Ranging results at 500 different locations by different versions of Splicer in a single WiFi band.	61
4.13	CDF of ranging errors from a single band.	62
4.14	CDF of ranging errors with CSI splicing.	63
4.15	Measured power level variance for NLoS path components in the derived power delay profiles. (a) In 20MHz WiFi band; (b) In 200MHz WiFi band.	64

4.16	Computation delay of Splicer.	65
4.17	CDF of localization errors using the original CUPID and the enhanced CUPID by Splicer.	66
4.18	Bandwidth used for localization under different moving speeds. (Left) Spliced bandwidth; (right) Percentages of early terminations.	67
4.19	Localization errors in different environments using the original CUPID and the enhanced CUPID by Splicer.	68

List of Abbreviations

AGC	Automatic Gain Controller
ARQ	Automatic Repeat reQuest
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BICM	Bit-Interleaved-Coded Modulation
CBER	Codeword Bit Error Rate
CCK	Complementary Code Keying
CFO	Carrier Frequency Offset
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
CSI	Channel State Information
DSSS	Direct-Sequence Spread Spectrum
EVP	Error Event Probability
ECC	Error Correction Capability
FEC	Forward Error Correction
FMCW	Frequency-Modulated Carrier Wave
IEEE	Institute of Electrical and Electronics Engineers
LoS	Line of Sight
MAC	Media Access Control
MIMO	Media Access Control
NIC	Network Interface Card
NLoS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
PBER	Packet Bits Bit Error Rate
PBD	Packet Boundary Detection
PLCP	Physical Layer Convergence Protocol
PHY	PHYsical layer of the OSI model
PRR	Packet Reception Ratio
PTC	Protection Layer
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RS	Reed-Solomon
SDR	Software Defined Radio
SER	Symbol Error Rate
SFO	Sampling Frequency Offset
SNR	Signal Noise Ratio
UnPKT	Unequal Packet Protection
USRP	Universal Software Radio Peripheral
WLAN	Wireless Local Area Network

Chapter 1

Introduction

Mobile devices, such as the smartphone, tablets, and wearable-devices, are becoming increasingly popular all around the world. Mobile voice, data, and video services are already considered a necessity in our daily life. Because of that, the demand for mobile data is increasing fast. According to Cisco VNI report [6], mobile traffic has grown 4,000-fold over the past ten years and almost 400-million-fold over the past 15 years. To achieve faster speeds in the face of increasing user demand, wireless local and wide area networks now turn to wideband transmission [12, 30, 66]. For example, IEEE 802.11n [4] enables 40 MHz channelization, which doubles the bandwidth of the traditional 20 MHz channel. Its follower—IEEE 802.11ac [5], can now transmit over a full 160MHz of bandwidth.

Due to the frequency selective fading of indoor wireless channel, different OFDM (Orthogonal Frequency Division Multiplexing) subcarriers of a wideband channel will experience different channel qualities [10, 58, 81]. Commercial Wi-Fi cards can measure the channel quality of subcarriers and report it in CSI (Channel State Information) format. We plot the CSI of one channel in Figure 1.1 (a). From this figure, we could observe apparent variations in the channel qualities. Some deep fading subcarriers exhibit much lower SNR (Signal-to-Noise Ratio) compared with others, which results in significantly larger BER (Bit Error Rate).

Existing Wi-Fi system can not deal with such a highly uneven BER distribution: note the visible difference in BER for data bits in different positions of the packets, ranging from 10^{-6} to

Figure 1.1: Data from 20 MHz 802.11n wireless channels. **(a)** Signal-to-noise ratio (SNR) across different subcarriers, measured over eight received packets; **(b)** Bit error rates (BERs) across different subcarriers; **(c)** The BER of decoded data bits, by position in the packet.

almost 10^{-4} , as illustrated in Figure 1.1 (c). But, if we can construct a picture of a future packet like the one in Figure 1.1 (c) before we actually send the packet out, we are able to protect the packet bits “unequally” according to their BER level at MAC layer, without modifying the IEEE 802.11 WLAN framework. In order to do that, the CSI of the wireless channel like the one in Figure 1.1(a) must be accurately measured.

CSI is the information source of many Wi-Fi based localization or tracking systems. Essentially, CSI is channel measurement result in the frequency domain. Therefore, some prior works [68, 92, 94] use CSI as fingerprint to localize mobile devices actively or human being passively. The dynamics of CSI is also used in [36, 85, 87, 91, 93, 105] to classify different types of human activities.

The frequency domain CSI can also be transformed losslessly to the time domain power delay profile through IFFT (Inverse Fast Fourier Transform). The power delay profile gives the power strength of a signal received through a multipath channel as a function of propagation delay, that profiles the multipath arrivals of the signal. A power delay profile fully charac-

terizes a multipath channel, and has been recently used in various motion- or location-based applications [36, 68, 70, 87, 92, 93, 105] — multipath channel dynamics can be unveiled from consecutive measures of the power delay profile, *e.g.*, tracking the power delay profile changes in a multipath channel can detect an object’s movement [36, 87, 93], like a person’s walking, falling, talking, or gestures, etc. In addition, the exact power level measured from each signal path can also be used to estimate the path length, *i.e.*, ranging between a pair of transmitters [70, 92].

Therefore, extracting accurate, high precision and complete CSI information from commodity Wi-Fi devices is critical. Linux-CSI-tool [31] is a toolkit built on the Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware. Such a tool, however, cannot extract CSI of all the subcarriers in one Wi-Fi channel. Instead, it can only report 30 out of 56/114 data subcarriers of a 20/40 MHz channel. What’s more, the precision of reported CSI is not enough to support sensitive motion detection applications. The CSI is represented using 8-bit binary number, which can only differentiate 258 states. Most importantly, this tool is not open-source. People are not allowed to change or add their own functionality. It is also impossible to port this tool to another platform, such as Wi-Fi AP, mobile devices or other embedded devices.

Motivated by that, we build an open-source tool based on Atheros Wi-Fi NICs. We believe it is the first tool that can extract CSI of every subcarrier in a channel. The precision is also much higher since the CSI is represented using a 10-bit number. At last, this tool is portable to other platforms.

Based on our CSI tool, we build two systems: UnPKT and Splicer. UnPKT uses CSI to predict the BER of packet bits and provides accurate protection to them “unequally” according to their BER level and BER distribution. In such a way, the effective transmission rate of each packet bit can best match its experienced channel condition. UnPKT makes no amendments to current IEEE 802.11 Wi-Fi framework so that it can be implemented on commercial mobile devices all around the world with just a software update. Splicer can derive high resolution power

delay profiles by splicing the CSI measurements from multiple Wi-Fi frequency bands. With high resolution power delay profiles, the performance of a plethora of upper-layer applications, *e.g.*, localization, object tracking, gesture recognition, etc., can be significantly improved.

1.1 Exploiting CSI to improve link throughput of Wi-Fi

The BERs (Bit Error Rate) of decoded packet bits are hence highly *uneven*—decoding errors are more likely to occur in certain bit positions of a packet, just as shown in Figure 1.1 (c)—which is also observed in the prior measurements [34, 51]. Error-prone bits inherently dominate transmission failures and impair throughput. The default 802.11 rate selection mechanism can help to alleviate the error-prone bits [48, 75]. With a conservatively-selected rate, all packet bits will experience low BERs, which however under-utilizes the channel bandwidth and reduces the throughput. Simply relying on rate selection cannot fundamentally address the problem.

To ensure full utilization of the channel capacity and successful packet delivery, accurate protection should be given to packet bits “unequally” according to their BER level and BER distribution. In such a way, the effective transmission rate of each packet bit can best match its experienced channel condition, and the overall packet bit BERs become even and low. Transmission failures thus can be prevented, and the throughput improves. PHY layer “unequal” bit protection has been explored in the previous literature [15, 62]. The basic idea is to break the existing 802.11 uniform bit protection, and provide codeword bits with the most appropriate redundancies in the PHY layer according to the sub-carrier quality or importance of packet bits. However, those approaches need to redesign the PHY layer, which causes excessive development overhead and is not compatible to the existing 802.11 devices.

To comply with the 802.11 framework, we propose to supplement the MAC-layer FEC (Forwarding Error Correction) redundancies to packet bits before they are sent to the 802.11 PHY layer. Different redundancies are calculated and provided according to the BERs of different segments of packet bits. By doing so, we can provide effective and accurate packet protection

without altering 802.11. The major design challenge lies in deriving the accurate relationship between the heterogeneous subcarrier quality and the decoded packet BERs, all the way through the entire 802.11 PHY layer. In particular, codeword bits traverse different sub-carriers and undergo complicated PHY-layer operations at the receiver before decoding, *e.g.*, demodulation, deinterleaving, etc. However, the estimation of the final packet bit BERs from the codeword errors could be very difficult due to the arbitrarily distributed codeword bit errors as a result of frequency selective fading. Existing 802.11 packet bit BER analysis considers a homogenous codeword error distribution [9], which can be fully described in mathematics and greatly simplifies the analysis, but only works over the flat-fading channel. The decoded packet bit BERs with an arbitrary codeword bit error distribution is still unknown. Recitation [51] considers frequency selective fading channels and proposes the EVP metric to indicate the error-prone positions within a packet. Nevertheless, the EVP metric represents the likelihood of decoding error events [51], which cannot directly map to packet bit BERs.

A straightforward solution to sidestep such a challenge is to estimate an averaged BER of packet bits, according to which, a uniform protection (with a long FEC block length) is provided to the entire packet. Using an averaged BER, however, leads to an inaccurate estimation of protection needed since not only the amount of bit errors but their distribution will affect the number of block errors and thus the amount of protection. On the other hand, the encoding/decoding overhead of block codes (*e.g.*, RS codes, LDPC codes, etc.) significantly grows with the block size, making it computationally infeasible to treat the whole packet as one block.

In this work, we study the 802.11 decoding process and observe that the decoding error of a packet bit corresponds to a series of dense errors in a group of underlying codeword bits. We find that the probability of dense codeword bit errors, together with the error density, can well approximate the packet bit BERs, based on which, we propose a BER estimator that can estimate the bit-wise packet bit BERs. Our estimator is computationally efficient and takes CSI as the sole input. According to the estimated packet bit BERs, we protect the packet bit unequally. Transmission failures are largely prevented while throughput improves.

We use commodity Atheros 9580 Wi-Fi NICs to validate the effectiveness of the UnPKT design. We further compare UnPKT with the state-of-the-art approaches using the trace-driven evaluation on the GNU Radio platform. The results show that the goodput gain achieved by UnPKT is significant, ranging from 12.2% to 200%.

1.2 Exploiting CSI to improve sensing resolution of Wi-Fi

As we discussed above, the frequency domain CSI can be transformed lossless to the time domain power delay profile through IFFT (Inverse Fast Fourier Transform). Figure 1.2 illustrates the process (which will be detailed in §4.1.1).

Figure 1.2: Transformation from a channel frequency response to the power delay profile. (a) A channel frequency response, where f_0 , Δf , and B represent starting frequency, frequency sampling resolution, and bandwidth, respectively; (b) Derived power delay profile, where τ_0 and $\Delta\tau$ represent the propagation delay of LoS path and the power delay profile resolution, respectively.

The time resolution of the derived power delay profile from CSI, *e.g.*, $\Delta\tau$ in Figure 1.2(b), is limited by the bandwidth of the transmitted signal [29, 65], *e.g.*, B in Figure 1.2(a), and $\Delta\tau = 1/B$. A high resolution power delay profile can differentiate subtle multipath channel changes, and consequently detect tiny activities. For the widely used 20MHz bandwidth in 802.11n [38], the power delay profile resolution is up to $50ns$, which leads to a $15m$ resolution

in measuring the multipath lengths. Such a resolution imposes inevitable uncertainty in mobility detection [11, 36, 86, 87, 93, 105], gesture recognition [84], or localization [49, 70, 92]. Theoretically, at least 200MHz bandwidth is needed for a finer grained motion detection, *e.g.*, less than $1.5m$ uncertainty to differentiate slight human body movements, which is impossible for current commodity WiFi NICs. Some recent works directly use CSI in replacement of power delay profile to learn the channel dynamics. The CSI description of the channel is, however, essentially limited by the bandwidth. In addition, CSI description is indirect and dependent of hardware uncertainty.

We observe that although the width of each individual WiFi band is limited, *e.g.*, 20MHz/40MHz, the total bandwidth allocated to 802.11 WiFi is wide, *e.g.*, more than 200MHz at 5GHz frequency band in 802.11n, which covers 10/5 different 20/40MHz channels. Furthermore, the CSIs measured from these individual WiFi channels can be spliced to derive a finer power delay profile with much higher time resolution.

Splicing CSI from multiple bands together, however, is challenging. The CSI calculated by the commodity WiFi NICs contains the signal distortions due to both channel propagation and imperfect signal processing on the hardware, *e.g.*, imprecise sampling frequencies at the sender and receiver, shift of the central frequencies, and power control uncertainties. WiFi communication systems do not have to explicitly separate the two sources of signal distortions, because only end-to-end distortion needs to be captured and compensated as a whole in the equalization stage. To derive a precise power delay profile for the channel, however, it requires to precisely separate the channel attenuation part from the mixed signal distortions due to hardware imperfection, which is non-trivial. The sampling clock frequency uncertainty causes frequency-relevant CSI phase measurement errors in each individual channel. The central clock frequency shift and the power control uncertainty further introduce notable phase and amplitude offsets cross different channels, respectively. Based on the raw CSI measures, it is unknown how to compensate those errors for CSI splicing without the knowledge of ground-truth CSI. In addition to above challenges, wireless channels are time-varying, especially in the

mobile environment. Few CSI measurements are allowed for scanning the whole WiFi band during a short coherence time. To deal with such a practical limit, we have to devise an effective method to correct and splice CSI measurements with insufficient samples and affordable computation cost.

We propose a set of key techniques to address above challenges. At the high level, we exercise the observation that the CSIs collected from different frequency bands should lead to the same power delay profile that characterizes the communication channel itself. We propose an efficient method that searches for a CSI manipulation that maximizes the matching between the power delay profiles derived from CSIs obtained at different frequency bands, based on which we can perform a preliminary CSI splicing. However, the power delay profiles used for matching are derived from narrow WiFi bands with limited bandwidth, so the spliced CSI is still of low quality. We devise a wider frequency window and perform a rolling-based calibration on the spliced CSI, based on which we refine the error correction to achieve a precisely spliced CSI. To accommodate the computations in the limited coherence time, we further develop a lightweight scheduler that is able to determine the optimal number of CSIs to measure from each individual WiFi band to strike a trade-off between the error compensation and the total bandwidth that can be afforded for the CSI splicing.

We develop a system, called Splicer, to incorporate above techniques on commodity Atheros 9580 NICs. Our benchmark experiments show that Splicer can derive high resolution power delay profiles from spliced CSI. We evaluate the derived power delay profile by estimating the distance between the sender and the receiver. According to our experiments, Splicer can reduce the median ranging error from $7.1m$ to $1.63m$ compared with using raw CSI traces from NICs. In light of such a high resolution, Splicer can immediately enhance the performance of a plethora of upper-layer applications, *e.g.*, object tracking, gesture recognition, localization, etc, without additional modification to the original application design. We demonstrate this benefit with a case study. We build the recent single-AP localization CUPID [70] on top of

Splicer. Our evaluations show that the localization accuracy can be substantially improved. In particular, Splicer improves the CUPID localization accuracy by 71%, with median localization errors about $0.95m$.

1.3 Organization of This Thesis

This thesis consists of 5 chapters. The literature review is presented in Chapter 2. In Chapter 3, we introduce a system called “UnPKT” that protect packet bits unequally according to their BERs so that the effective transmission rate of each bit can best match channel condition. UnPKT makes minimum amendments to current IEEE 802.11 WLAN framework. In Chapter 4, we introduce a system called “Splicer” that can derive high resolution power delay profiles by splicing the CSI measurements from multiple Wi-Fi frequency bands. Finally, we conclude the thesis and describe our future work in Chapter 5.

Chapter 2

Related Work

We describe related work of our proposed schemes respectively. We first introduce IEEE 802.11 standard. We next describe related work on improving Wi-Fi throughput from different aspects. We then introduce related work of Wi-Fi based sensing and localization. We then briefly introduce all existing CSI tools.

2.1 IEEE 802.11 standards

IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in 2.4 GHz, 5 GHz and 60 GHz frequency bands. The base version of the standard was released in 1997, and has had subsequent amendments. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand. In the 802.11 family, 802.11b [2] was the first widely accepted one, followed by 802.11a [1], 802.11g [3], 802.11n [4] and 802.11ac [5]. The transmission rate increases with the update of the 802.11 standard. I will introduce all those standards one by one.

IEEE 802.11-1997 and IEEE 802.11b. IEEE 802.11 (legacy mode) or more correctly IEEE 802.11-1997 is the original version of the IEEE 802.11 wireless networking standard family. It can only support two raw data rates of 1 and 2 Mbps to be transmitted via DSSS (Direct-Sequence Spread Spectrum) modulation at 2.4GHz. The original standard defines the

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) as the medium access method to ensure the reliability of transmissions.

The first widely accepted standard is actually IEEE 802.11b [2], which is a direct extension of the original 802.11-1997. IEEE 802.11b uses CCK (Complementary Code Keying) as its modulation technique. Therefore, it can support a maximum transmission rate of 11 Mbps. The bandwidth of IEEE 802.11b is 20 MHz and there are 11 channels in total that can be used in 2.4GHz frequency band.

IEEE 802.11a/g. IEEE 802.11a was proposed in 1999. Comparing with 802.11b, it has advanced the Wi-Fi standard in two aspects. First of all, 802.11a is the first standard that adopts OFDM as its modulation technique. OFDM modulation has much higher spectral efficiency as compared to other double sideband modulation schemes like DSSS. Therefore, 802.11a is able to offer a maximum transmission rate of 54 Mbps. OFDM is also selected to fight against the mutlipath effect which is common in indoor communication. Second, 802.11a works on 5 GHz frequency band which is considered as much clean than the crowed 2.4 GHz band. Therefore we will observe much less interference from other wireless technology.

IEEE 802.11g was introduced in 2003. It works on the same 2.4 GHz as IEEE 802.11b, but adopts OFDM as modulation scheme. Therefore IEEE 802.11g can also provide a maximum transmission rate of 54 Mbps. Even though 5 GHz is less occupied by other techniques, 2.4 GHz can provide a much larger communication rage since its signal attenuation is smaller than 5 GHz signal. We need 2.4 GHz Wi-Fi to penetrate walls and provide whole-home Wi-Fi coverage.

IEEE 802.11n/ac. The IEEE 802.11n standard was initiated in January 2004 and finalized in 2008 with its formal publication in 2009. It improves network throughput over the two previous standards 802.11a and 802.11g with a significant increase in the maximum net data rate from 54 Mbit/s to 600 Mbit/s. With the improved performance offered by 802.11n, the standard soon became widespread and popular. Quite a few new techniques are incorporated into the 802.11n standard to achieve superior performance:

- **MIMO.** IEEE 802.11n devices are equipped with multiple antennas and exploit MIMO technique to harness the spatial diversity to improve channel capacity. One packet can be split into several parts and delivered through different spatial streams using multiple antennas. The throughput increases linearly with the number of streams.
- **40 MHz channelization.** IEEE 802.11n supports 40 MHz channel. Previous Wi-Fi standards use 20 MHz bandwidth for each channel. The trade-off is that there will be fewer non-overlapped channels and hence less devices that are transmitting concurrently.
- **Frame aggregation.** Frame aggregation is a process of packing multiple MSDUs or MPDUs together to reduce the overheads and average them over multiple frames, thereby increasing the user level data rate.

IEEE 802.11ac was developed from 2011 through 2013 and approved in January 2014. The specification has multi-station throughput of at least 1 gigabit per second and single-link throughput of at least 500 megabits per second (500 Mbit/s). This is accomplished by extending the air-interface concepts embraced by 802.11n: wider RF bandwidth (up to 160 MHz), more MIMO spatial streams (up to eight), downlink multi-user MIMO (up to four clients), and high-density modulation (up to 256-QAM).

2.2 Towards robust Wi-Fi transmissions

Our work aims to improve the Wi-Fi by accurately protecting the Wi-Fi packet bits, reducing the retransmissions and finally enhancing the throughput. Considering the wide deployment of IEEE 802.11 devices, we would like to make minimum extensions to the WLAN framework so as to allow easy integration into the IEEE 802.11 protocol stack. A large body of research effort has been made in the area. In the following, we briefly introduce related works and motivate our design choices.

Hybrid ARQ. With hybrid ARQ, packet bits are encoded, *e.g.*, by the convolutional code, before transmission, and the whole packet or additional coded bits are retransmitted if the original transmission fails [80]. 802.11 essentially follows the hybrid ARQ principle. However, retransmission causes non-negligible overhead, *e.g.*, transmission delay and MAC-layer overhead. A variety of partial packet retransmission schemes [35, 52, 95] have been proposed to improve the retransmission efficiency, which reduce the transmission delay of each retransmission, but still suffer from the MAC-layer overhead. Different from those approaches, our work predictively protects the packet bits of each transmission and completely eliminates most retransmission overhead. Thus, we can better utilize the channel bandwidth and achieve higher throughput.

Frequency Diversity. Due to frequency selective fading, the subcarriers of OFDM could exhibit highly different channel quality [15, 62], which has fundamental impact on the performance of 802.11 networks. To mitigate the impacts of frequency selective fading, FARA [62] introduces separated modulations and channel coding for each sub-carrier. [15] harnesses this frequency diversity by sending bits with higher priority to subcarriers with better channel quality. Apex [69] leverages the different reliabilities of codeword bits of constellation symbols to achieve unequal bit protection for video or voice communications. These designs require a complete redesign of 802.11 PHY layer, including the channel coding, interleaving and modulation modules. Our work, on the contrary, completely complies with the 802.11 framework and the existing 802.11 hardware design. We explicitly addresses the challenge of predicting diverse packet bit BERs within 802.11 framework.

Cross-Layer FEC. Cross-layer FEC provides additional packet bit protection above the PHY layer. It is mainly applied to the APP layer for the video/audio streaming by protecting certain key information in the streams, *e.g.*, key video frames, meta header information, etc., with additional redundancies [20, 46]. The redundancies can be equally or unequally applied to the protection objects. Those approaches are however content-aware and limited to specific

applications. On the contrary, UnPKT is content-oblivious and serve in a general purpose for all 802.11 wide-band transmissions. On the MAC-layer, [22] proposes to add equal MAC-layer FEC to battle packet bit errors in narrow-band channels. The redundancy is empirically added to each packet. Compared with [22], Our work is designed for modern 802.11 WLANs working in wide-band channels and solves the unique issue of the uneven packet bit BER estimation. Our experiments show that the design [22] works poorly in wide-band channels. In contrast, our design achieves much better performance.

Partial packet recovery. Partial packet recovery approaches such as PPR [39], SOFT [89], Maranello [35], ZipTx [52] and Unite [95] minimize retransmission overhead. They have the advantage when packet collisions are the dominant factor limiting throughput or the channel is extremely lossy even the lowest data rate cannot reliably transmit a packet. Given the data rate selected bit-rate adaptation selects with a low PER, the throughput may still increase when the bit-rate is further augmented, if the throughput gain from the higher data rate outweighs the retransmission overheads. The accurate BER our work predicts enables such an assessment to fully utilize the channel.

Bit-rate adaptation. A variety of SNR-based bit-rate adaptation approaches exist, *e.g.*, PBAR [37], OAR [67], CHARM [43], BlockRate [79], and Medusa [71]. Due to frequency selective fading, Zhang et al. [103] and Camp et al. [17] observe that the SNR measurement needs careful calibration. To overcome the SNR measurement issue, many solutions utilize packet reception statistics, *e.g.*, SampleRate [16], RRAA [88], and MIMO-Rate [59]. Shen et al. [76] further studies the rate adaptation in multi-user MIMO networks. [13, 18, 50] focus on the rate adaptation analysis, [47, 64] concentrate on the energy/power control, and [54, 104] explore to correlate the link behaviors to rate adaptation in a target network. Recent studies further propose to adapt data rates by directly measuring coded BER [82] or symbol-level dispersion in AccuRate [73], but require specialized hardware. EEC [19] can adapt rates based on the estimated data bit BER without the hardware modification. Although most of above

works are CSI-agnostic, they mainly adapt, instead of selecting, bit rates. The adaptation is less effective when channels change faster and the configuration space is more complex (*e.g.*, MIMO). The most recent approach [31] uses ESNR for the bit-rate selection. Benefiting from the aggressive rate selection based on the estimated packet bit BER, our work outperforms the best reported performance for both 802.11 SISO rates.

2.3 Towards precise Wi-Fi sensing

Channel sounding. Measuring the wideband channel frequency response requires high-end hardware with high sampling frequencies [26, 55]. The authors in [56] develop systems to measure channel frequency responses from a group of narrow bands to approximate a wideband channel. In [44], the receiver only listens to a few harmonics of a wideband signal each time and then can reconstruct the wideband frequency response. Such a design does not require the modification at the sender. In addition, CSI-SF proposed in [23] can estimate the channel state information for multi-streams using the single stream measurement result. Some ToA-based localization approaches [98, 100] also propose to increase the resolution using the channel combination, however, only with Software-Defined-Radio. Although Splicer shares a similar principle with those existing works, most of them require tight synchronization between the sender and receiver, *e.g.* devices are connected by the same clock or use GPS and atomic clocks. In this work, however, we meet and address particular challenges due to the hardware imperfection and stringent channel coherence delay constraint, which do not exist in any of existing works. In addition, Splicer can be integrated into commodity NICs without any hardware modification.

CSI phase calibration. Prior works also notice that the CSI traces reported by WiFi NICs contain phase errors introduced by hardware [97, 101, 102]. ArrayPhaser [28] enables the phased array signal processing on commodity WiFi devices. However, ArrayPhaser does not correct any of those phase errors, instead they just treat the phase values measured from one

NIC as the reference to calibrate the phase values of other NICs. Hence, they cannot truthfully remove the phase values to derive precise power delay profiles. Prior works [70, 90] try to synchronize the phases from two consecutive received CSIs via a linear transform. After the transformation, if the two measurements are from the same multipath channel, even the collected CSIs are different due to hardware noises, the transformation on these two CSIs leads to the same result, which could be used as fingerprint for localization. Some recent works aim to explicitly correct CSI phase errors, *e.g.*, MegaMIMO [63]. However, MegaMIMO requires both nanosecond-level synchronization and the access to the raw signal at PHY layer, which are not available on commodity NICs. In summary, existing works cannot directly remove measurement errors from CSIs reported by commodity WiFi NICs, and hence cannot address the challenges we met in Splicer.

Power delay profile based applications. At different locations, the received power delay profiles will be different, which can make them a good choice for the fingerprint-based localization design [72, 94]. On the other hand, the Line-of-Sight information can be directly inferred from the power delay profile [90]. The power level of the LoS path can also be used to ranging between a pair of transmitters [70, 92]. Indoor localization based on the ranging results requires no dense AP deployments, no manual fingerprinting site survey, and no sophisticated AP hardware [70, 92]. For activity recognition, although the detailed relation between the multipath channel variance and the different activities is unknown, recent works propose to learn the inner relation. For example, to detect the existence of human beings [105], to count the number of people moving around [93], to detect human falling down in [36], and recognize different types of human activities [8, 74, 87]. Splicer can benefit all above applications since we can obtain a wider CSI containing more frequency band information to derive a higher-resolution power delay profile, which more precisely describes the multipath channel.

2.4 CSI tool

Extracting accurate, high precision and complete CSI information from commodity Wi-Fi devices is critical. Building a tool that can help us extract CSI is very challenging because of the engineering effort it takes to modify the device driver of the Wi-Fi NIC. Linux-CSI-tool is the first tool that has been released. Our Atheros-CSI-tool is the second one. We will introduce them one by one.

Intel 5300 CSI tool. Linux-CSI-tool [31] is a toolkit built on the Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware. Such a tool, however, cannot extract CSI of all the subcarriers in one Wi-Fi channel. Instead, it can only report 30 out of 56/114 data subcarriers of a 20/40 MHz channel. What's more, the precision of reported CSI is not enough to support sensitive motion detection applications. The CSI is represented using 8-bit binary number, which can only differentiate 258 states. Most importantly, this tool is not open-source. People are not allowed to change or add their own functionality. It is also impossible to port this tool to another platform, such as Wi-Fi AP, mobile devices or other embedded devices.

Atheros CSI tool. Atheros-CSI-Tool [7] is an open source 802.11n measurement and experimentation tool. It enables extraction of detailed PHY wireless communication information from the Atheros WiFi NICs, including the Channel State Information (CSI), the received packet payload, and other additional information (the time stamp, the RSSI of each antenna, the data rate, etc.). Atheros-CSI-Tool is built on top of ath9k, which is an open source Linux kernel driver supporting Atheros 802.11n PCI/PCI-E chips, so theoretically this tool is supposed to be able to support all types of Atheros 802.11n WiFi chipsets. Atheros-CSI-Tool is open source and all functionalities are implemented in software without any modification to the firmware. Therefore, we are able to extend the functionalities of Atheros-CSI-Tool with our own codes. Atheros-CSI-Tool has following key functionality:

- **Non-grouping, non-compressed CSI reporting.** By non-grouping we mean that Atheros-CSI-Tool reports the CSI value for each subcarrier, i.e., 56 subcarriers for 20MHz channel and 114 subcarriers for 40MHz channel. By non-compressed we mean that Atheros-CSI-Tool reports high precision CSI, i.e., 10 bit resolution for both imaginary and real part of the CSI.
- **Detailed payload records.** Atheros-CSI-Tool records the payload of every received packet. Atheros-CSI-Tool gives the detailed information about error bits within the packet. A packet is received incorrect, due to CRC check failure or the PHY failures in the NIC. The error type of the packet failure is also recorded.
- **Rich status records.** Atheros-CSI-Tool retrieves rich status information about the received packet. The current version retrieves the following information: the channel frequency, the time stamp the packet is received, the transmitted payload length, the payload error type, the data rate, the channel bandwidth, the subcarrier number, the number of transmitting and receiving antenna, the RSSI of the combination of all active receiving chains and the detailed RSSI of each chain.

Atheros-CSI-Tool is portable to other platforms including the UAVs (Unmanned Aerial Vehicle), laptops, Wi-Fi routers and mobile devices like our phone.

Chapter 3

Augmenting Wide-band 802.11 Transmissions

3.1 Design

In this section, we describe our unequal packet bit protection design, *i.e.*, UnPKT.

3.1.1 Motivation and Observations

We investigate wireless links in all three test-beds described in Section 3.2 and measure the packet bit BERs over lossy links. In theory, the BER distribution should be measured when the link quality is fixed. In practice, the link quality varies all the time. To minimize measurement error, we conduct experiments at night with minimum dynamics in the environment. In addition, we scan the frequency band before each experiment to make sure there is no noticeable interference from other wireless sources. We disable packet retransmission for the measurements. For each link, the sender transmits packets to the receiver indexed by the sequence number. The receiver records both correct and corrupted packets. The payload of each packet is random content of 1000 bytes. For each corrupted packet that fails in the CRC check, we can identify the decoded bit errors referring to the ground truth at the sender side.

Measurements. Figure 3.1(a) plots the PRR (Packet Reception Ratio) of all 166 links in the three test-beds. Due to the data rate uniformity for all the packet bits and the limited

Figure 3.1: (a) PRRs of all 166 links in three test-beds; (b) CDF of the PRR.

rate choices in the 802.11 standards, it is unlikely that one selected data rate can perfectly match the frequency selective channel quality [24]. To maximize the throughput, transmissions are usually over marginal links, *e.g.*, $70\% < \text{PRR} < 98\%$, and transmission failures are inevitable [83]. The performance is thus limited by the retransmission overhead of corrupted packets. Figure 3.1(b) depicts the CDF of all the PRR values. From the figure, we observe that about 50% works on the marginal links with the PRRs. If the PRR of a link is sufficiently low, *e.g.*, smaller than 70%, a lower data rate needs to be selected to match the channel quality.

Periodic packet bit BER distribution. For each marginal link, we examine the decoded packet bit BER distribution. Figure 3.2 plots the packet bit BERs measured from a randomly selected link with convolutional coding rate 5/6 and 64-QAM modulation. Although not shown here, these results generalize for other links as well. The x -axis represents the bit position, and for each bit position, the y -axis indicates the measured BER over this link. Figure 3.2 shows that the BER distribution is highly uneven due to frequency selective fading across subcarriers. In addition, the BER distribution of the decoded packet bits have a strong periodic property where the period equals the number of packet bits within one OFDM symbol. The result is consistent with the observations in the literature [34, 51], which is fundamentally different

from the BER distribution of flat-fading narrow band channels. Figure 3.2 implies that given a wide-band channel, the location of decoding errors biases to certain bit positions. We can explicitly protect those error-prone bits with better redundancy to prevent transmission failures and improve the throughput.

Figure 3.2: Measured BER of each bit position within a 1000-byte packet. We depict the first 3500 packet bits for the sake of a clear presentation.

In Figure 3.2, we further zoom in two different periods and observe that the packet bit BERs are distributed similarly within the two windows. As a matter of fact, the BER distribution is similar in any two different periods. To show that, we plot the average BER distribution of decoded packet bits in one OFDM period in Figure 3.3(a), where the x -axis represents each bit position in the period, the y -axis is the average of the BERs over bits of the same position in OFDM symbols. The reason of the periodical property is as follows. The packet bits are coded into codeword bits and transmitted with OFDM symbols. The codeword bits in the same position of their own OFDM symbols are interleaved to the same sub-carrier, experiencing the same channel quality. The BER distribution of the codeword bits is thus periodical, leading to the BER of the decoded packet bits of a periodical property.

The above observation reveals that the packet-level BER diversity attributes to the uneven BER distribution in each individual period. As the BER distributes similarly among different periods, we can focus on identifying error-prone bit positions in one period. All error-prone bit positions from other periods are then equivalently obtained. The overhead to analyze the BER

distribution in one period is minor compared with the overhead of analyzing the entire packet. In addition, such an overhead is merely determined by the period length, which is oblivious to the packet length.

Figure 3.3: (a) Average BER of each bit position in one period; (b) BER of the codeword bits in one OFDM symbol.

Predictable BER distribution within a period. We find that the BER distribution in each period is predictable, which strongly relates to the density of errors that could occur in the underlying codeword bits.

In Figure 3.3(b), we plot the BER of each codeword bit in one OFDM symbol. The codeword bit BER values can be calculated using the channel state information (CSI) (see Section 3.1). As the channel coding rate is $\frac{1}{2}$, one packet bit is encoded into two codeword bits. We align each packet bit with its codeword bits in Figure 3.3, and observe that the packet bit positions of higher BERs usually correspond to codeword bits of high BERs. For example, within the codeword bit region $[1, 53]$ in Figure 3.3(b), for every 13 codeword bits, the BERs of three of them are greater than 4.04×10^{-2} and the maximal BER equals to 2.56×10^{-1} . In convolutional codes, ECC (Error Correcting Capability) stands for the maximum consecutive errors that a code can tolerate. For example, ECC equals 4 when the coding rate is $\frac{1}{2}$ [4]. In Figure 3.3(b), it is easier to have more than 4 (ECC=4) concurrent codeword bit errors in the region $[1, 53]$. The packet bits, within this region, are thus more likely to be erroneous. Essentially, the codeword

Figure 3.4: All the UnPKT operations are abstracted in the PTC layer.

bit BER distribution is determined by both the frequency selective channel condition (CSI) and the PHY-layer operations, whose diversity leads to the uneven packet bit BERs after decoding.

Summary. The 802.11 packet bit BER distribution is not equal, which relates to the occurrence of underlying codeword bit errors. In the next section, we show that the channel CSI determines how likely the codeword bit errors could occur, which can be used to approximately detect the error-prone bit positions in a packet and estimate their BERs. After that, packet bits of different BERs can be protected using the most appropriate redundancies.

3.1.2 Design Overview

In UnPKT, packet bits are protected based on their estimated BERs to prevent transmission failure and improve the throughput. The UnPKT design is encapsulated into a Protection (PTC) layer with a clean abstraction integrated in the existing 802.11 Wi-Fi network stack. The PTC layer is built atop the PHY layer and interacts with the MAC layer. Prior to a packet transmission, PTC layer intercepts the packet from the upper layer and returns the protected packet to MAC layer for transmission. PTC layer is purely software-implementable without any extra hardware support. We note here that, the protection process cause only negligible delay to the packet processing since we use a very efficient Reed-Solomon encoding implementa-

tion in software, which has been proven that can catch up with 802.11 packet transmission flow [52, 95].

Figure 3.4 depicts the architecture of UnPKT. At the sender side, the PHY layer measures the channel CSI for the MAC-layer to select a data rate for the next transmission (Section 3.1.3). The PTC layer takes as input the selected data rate, the estimated PRR, and the measured CSI. If the estimated PRR indicates the next transmission to be over a marginal link, *e.g.*, $PRR < 98\%$, PTC layer provides additional protection for the packet; Otherwise, the packet is transmitted directly. To enable the PTC-layer protection, the codeword BER (CBER) estimator module calculates the BER of each codeword bit given the selected data rate and measured CSI (Section 3.1.4). The packet bit BER (PBER) estimator then estimates the decoded packet bit BERs (Section 3.1.5), which are further used by the packet protector to add appropriate redundancies. Afterwards, the protected packet is returned to the MAC layer for transmission. In UnPKT, we also provide the option to augment the data rate if the estimated PRR (without protection) for the current data rate is high, *e.g.*, $PRR > 98\%$, and the estimated PRR (without protection) for the next higher data rate is sufficiently high as well, *e.g.*, $PRR > 70\%$ (PRR is estimated using the method described in Recitation [51]). PTC-layer protection can be performed for the augmented data rate. How the redundancies are added to the packet bits is recorded by a protection field in the packet header with 4-byte fixed overhead (Section 3.1.5).

Upon receiving a packet, if the packet is correctly decoded, the receiver delivers it to the upper layer; Otherwise, the receiver extracts the protection field information from the header and tries to recover the original packet. If the decoding still fails, the receiver explicitly requests for the retransmission.

3.1.3 Data rate selection

In UnPKT, we apply a state-of-art data rate selection protocol, Recitation [51], on the MAC layer. We adopt its implicit CSI feedback scheme to measure the channel using the reverse

ACK packets from the receiver to approximate the forward channel quality. We can then select the data rate and estimate the PRR for the next transmission.

3.1.4 BER estimator

To transmit a packet, all packet bits $b_n, n = 1 \cdots N$, are scrambled¹ and then encoded into a longer codeword bit sequence $c_k, k = 1 \cdots N'$, where N and N' are the lengths of the packet bits and codeword bits, respectively. To facilitate the discussion, we first focus on a $\frac{1}{2}$ coding rate to introduce the estimator design (i.e., $N' = 2N$), and postpone how the estimator can be applied to all other coding rates until the end of this subsection.

Codeword bits c_k are interleaved and modulated at the sender before transmission. The received codeword bits are de-modulated and de-interleaved by the receiver to reassemble the original codeword bits before decoding, whereas errors could occur in the reassembled codeword bits, which are denoted as $\tilde{c}_k$. The BER estimator is composed of two parts, the codeword bit BER estimator (CBER) and the packet bit BER estimator (PBER), where the CBER estimator first estimates the BERs of the codeword bits $\tilde{c}_k$, denoted as $e_k, k = 1 \cdots N'$. The PBER estimator further estimates the BERs of the decoded packet bits, denoted as $p_n, n = 1 \cdots N$, taking each e_k as input.

3.1.4.1 CBER estimator

The 802.11 standard employs a block interleaver of the size equal to the number of codeword bits in one OFDM symbol [4]. The goal of interleaving is to randomize the codeword bit order during the transmission such that long runs of low reliable codeword bits can be avoided. For any codeword bit c_k , the interleaver will interleave it to the position $k_2 = s \times \lfloor \frac{k_1}{B} \rfloor + (k_1 + B - \lfloor 13 \times \frac{k_1}{B} \rfloor) \bmod s$, where $s = \max\{1, Q/2\}$, Q is the size of one constellation point, B is the number of codeword bits contained in one OFDM symbol, and $k_1 = \frac{B}{13} \times (k \bmod 13) + \lfloor \frac{k}{13} \rfloor$.

¹As the scrambler performs bit-wise XOR between the original packet bits and a scrambling sequence, which is specified by 802.11 standards, knowing the scrambled packet bits is equivalent to knowing the original packet bits. In this report, we refer to the scrambled packet bits as packet bits for short.

Figure 3.5: Illustration of the decoding bit errors and the codeword bit errors.

After this permutation, the reordered codeword bits are further grouped into four clusters in sequence for the modulation. All the codeword bits in cluster one will be transmitted over sub-carrier $j \bmod 52$ in order, where $j = 1, 5, 9, \dots$, and 52 is the total number of sub-carriers. In general, the codeword bits in cluster i are transmitted over sub-carrier $j \bmod 52$, where $j = i, i + 4, i + 8, \dots$. Figure 3.3 illustrates how the codeword bits fall into the four clusters in one OFDM symbol.

The subcarrier over which codeword bits are transmitted is fixed, similar to [51], hence we know the channel quality that each codeword bit experiences during the transmission from the CSI, *e.g.*, the subcarrier SNR. In addition, subcarriers are narrowband in 802.11, *e.g.*, 312.5 kHz in 802.11n. The CBER estimator can employ a classical narrowband SNR-BER relation to estimate codeword bit BERs, $e_k, k = 1 \dots N'$, which serve as the input of the PBER estimator. In Figure 3.3(b), we have plotted the codeword bit BERs in one OFDM symbol. The four highlighted regions correspond to the four clusters formed during the modulation. In each region, the four repeated codeword bit BER patterns are because the transmission of the codeword bits in each cluster circulates among 13 ($=52/4$) fixed sub-carriers.

3.1.4.2 PBER estimator

As mentioned in Section 3.1.1, if the errors occur in a group of nearby codeword bits, the error density may exceed the protection capability of the convolutional code and lead to decoding bit errors. Figure 3.5 depicts such an instance, where “0” and “1” bits indicate the bit’s correctness,

e.g., they are for correct and erroneous bits, respectively. In general, the BER of packet bit n is composed of two parts:

$$p_n = \sum_{\eta} P_{group}(k, \eta) \times P_{fail}(\eta), \quad (\text{Eq. 3.1})$$

where $P_{group}(k, \eta)$ denotes the probability to form a codeword bit group with errors starting from codeword bit c_k of error density η , and $P_{fail}(\eta)$ denotes the probability that such a group could cause decoding bit errors. We now detail the calculation of Eq. 3.1.

$P_{group}(k, \eta)$ calculation. We consider the example in Figure 3.5 to introduce the calculation of $P_{group}(k, \eta)$. To facilitate the calculation, we define a *codeword bit group* with errors both starts from and ends at codeword bit errors, e.g., codeword bits $[k, k + 5]$ and $[k, k + 9]$ in Figure 3.5. We now focus on the latter group $[k, k + 9]$. In this group, $\eta = \frac{n_e}{G} = \frac{5}{10}$, where n_e is the number of errors and G is the group size. Given n_e and G , in principle, there are C_8^3 different group instances. Except the two errors stay at the beginning and the end of the group, the three ($= n_e - 2$) remaining errors could occur in the middle eight ($= G - 2$) codeword bits. Figure 3.5 depicts one such instance.

From the CBER estimator, we have obtained the codeword bit BER e_k . $P_{group}(k, \eta)$ in principle equals to the summation of the probabilities that each of the C_8^3 group instances will occur. However, it is computational intensive to enumerate all those combinations. To address this issue, we propose to use the average codeword bit BER ($\bar{e}$) of the eight middle codeword bits, e_2 to e_9 , to simplify the calculation as follows.

$$P_{group}(k, 5/10) = e_k \cdot e_{k+9} \cdot C_8^3 \cdot \bar{e}^3,$$

where $\bar{e} = (\sum_{x=k+1}^{k+8} e_x)/8$ and e_k is the BER of codeword bit $\tilde{c}_k$. For η equals to $\frac{n_e}{G}$ in general, $P_{group}(k, \eta)$ is calculated by the following equation:

$$P_{group}(k, n_e/G) = e_k \cdot e_{k+G-1} \cdot C_{G-2}^{n_e-2} \cdot \bar{e}^{n_e-2}, \quad (\text{Eq. 3.2})$$

Figure 3.6: $P_{group}(k, \eta)$ changes with η for the different convolutional coding rates (a) 1/2; (b) 2/3; (c) 3/4; (d) 5/6.

where $\bar{e} = (\sum_{x=k+1}^{k+G-2} e_x) / (L - 2)$. As we demonstrated in § 3.2.1, our algorithm can achieve accurate BER estimation with minimum computation overhead.

$P_{fail}(\eta)$ calculation. Given an arbitrary codeword bit group of size G with n_e errors, the decoding bit errors occur only when η is sufficiently high. Prior studies [57] have found that decoding bit errors occur mainly when n_e of a group equals to $EEC + 1$, where EEC is the protection capability of the convolutional code. $EEC = 4, 3, 2, 1$ for the convolutional coding rates $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}$, and $\frac{4}{5}$, respectively. Thus, we focus on $\eta = \frac{n_e}{G} = \frac{EEC+1}{G}$ in the $P_{fail}(\eta)$ calculation. Given the group error density η , there are $C_{G-2}^{n_e-2}$ different group instances. Although not all the instances lead to decoding bit errors, the probability that a group instance could cause decoding bit errors can be off-line tested. We manually create all possible instances of error groups with the same error density, pass them to the convolutional decoder and record how likely the decoding failure happens. We repeat such a process for all possible error densities. Results

Figure 3.7: Puncturing technique to achieve different coding rates in 802.11.

of the off-line testing is depicted in Figure 3.6. Even though we only plot the probability for $\eta = \frac{EEC+1}{G}$, all other error densities can be obtained in the same way. By doing so, $P_{fail}(\eta)$ can be directly obtained from Figure 3.6.

So far, we have obtained $P_{group}(k, \eta)$ and $P_{fail}(\eta)$. Then p_n is simply the summation of $P_{group}(k, \eta) \times P_{fail}(\eta)$ for all possible η s. According to Figure 3.6, we find that when η is sufficiently small, $P_{fail}(\eta)$ is close to zero. Hence only a small number of η s are involved in the p_n calculation.

On the other hand, as the packet bit n is encoded into two codeword bits, as shown in Figure 3.5, c_k and c_{k+1} , the decoding bit error at n can also be caused by the codeword bit groups starting from c_{k+1} . As a result, we have

$$p_n = \sum_{\eta} P_{group}(k, \eta) \times P_{fail}(\eta) + \sum_{\eta} P_{group}(k+1, \eta) \times P_{fail}(\eta), \quad (\text{Eq. 3.3})$$

p_n for other coding rates. We now extend the p_n calculation to all other coding rates. According to the 802.11 standard, all other coding rates, $\frac{2}{3}$, $\frac{3}{4}$, and $\frac{5}{6}$, are implemented based on the $\frac{1}{2}$ rate, using a puncturing technique as shown in Figure 3.7. For example, to achieve the $\frac{2}{3}$ coding rate, i.e., two packet bits are encoded into three codeword bits, two packet bits (e.g., b_n and b_{n+1}) are first encoded into four codeword bits, but the last codeword bit will not be transmitted, which we referred to as a *stolen bit*. The positions of the stolen bits are periodic, and specified in the 802.11 standard. To calculate p_n , if one of its encoded codeword bit is a

stolen bit, we simply ignore it in the calculation. For instance, $p_{n+1} = \sum_{\eta} P_{group}(k+2, \eta) \times P_{fail}(\eta)$.

Decoding error burst. We can now calculate the BER of any packet bit n , but we assume that only one erroneous packet bit will be caused by the decoding failure. In practice, as introduced in [27], a decoding failure of the convolutional code usually results in a set of packet bit errors, also known as error bursts. For instance, the burst in Figure 3.5 contains the errors in positions n and $n+3$. So far, p_n only calculates the probability when the error burst starts from packet bit n . p_n should also contain the probability that bit n is an error, but the error burst starts from prior packet bits.

The statistical error burst size has been well studied. Existing works [27] found that the error burst length follows the exponential distribution and the distribution parameters are determined by the average BER of the codeword bits, which can be easily calculated using the codeword bits BER predicted using CSI measured. Considering the error burst effect, we compute the expected error burst size l . After calculating p_n in Eq. 3.1, we add p_n to the BERs of the following l packet bits. Thus, for each packet bit, the output of the PBER estimator is the summation of the BER calculated from Eq. 3.1 and the BERs of the previous l packet bits.

3.1.5 Packet protector

The packet protector module provides unequal packet bit protection according to the BERs. A lightweight *cluster-based* method is used to provide unequal but appropriate redundancies to different packet segments.

Cluster-based protection. The interleaving operation of 802.11n standard divides coded bits in OFDM symbol into four consecutive clusters. Coded bits in different clusters are mapped to different sets of non-overlapping subcarriers. Such a process repeats κ times when 2^κ -QAM is used. As a result, coded bits in each cluster tend to have similar BERs. Convolutional coding induces error bursts and thus correlates BERs of nearby packet bits, so we can group

Figure 3.8: Burst length (bits) under different data rates.

sequential packet bits into four or $4 \times \kappa$ clusters and provide unequal protection based on their BERs. UnPKT sets the cluster number to four for all modulation choices and our experimental results in Section 3.2 demonstrate that it achieves good balance between the computational overhead and accuracy in estimation of redundancy requirement.

RS (Reed-Solomon) code. UnPKT employs RS code to provide protection for packet bits. RS code is efficient at correcting burst errors [60], which are common for the decoded packet bits after convolutional code. In RS code, one codeword consists of u RS symbols and each symbol consists of v bits. The u data symbols are composed of $u - w$ data symbols and w symbols as redundancy. As a convention, the RS code can be denoted as $RS(u, w)$. Any $RS(u, w)$ can correct up to $w/2$ symbol errors, which occurs when any number of bits in this symbol get wrong. UnPKT adopts the $RS(255, w)$ where each symbol consists of eight bits. This type of RS code has efficient software implementation and is widely used as 802.11 MAC-layer FEC [25, 35, 52, 95]. Figure 3.8 depicts bit error burst statistics from the corrupted packets measured in Section 3.1.1. The results show that only three to four $RS(255, w)$ symbols of redundancy are required to correct one error burst as the length of such bursts only last for 15 to 24 bits on average.

Encoding and decoding. In UnPKT, the packet protector groups clusters from the same portion across different OFDM symbols together and protects them using an RS code as shown

Figure 3.9: Clusters from multiple OFDM symbols are grouped together for BER estimation and protected using RS code.

in Figure 3.9. As a result, the clusters in one group manifest similar BERs. UnPKT uses the average BER of each group to calculate the redundancies required according to theory [60]. Note that the packet bits are not physically moved to form the groups of clusters. After determining the optimal w , the sender appends the RS parity symbols sequentially for each group and at the end of the packet. The value of w is encoded into the MAC header (a four byte overhead).

A simpler alternative of protecting packet bits is to encode the entire packet into a single RS codeword and estimate the redundancy based on the average BER of a packet. Using merely a packet level BER and ignoring the bit distribution, however, leads to inaccurate protection, *e.g.*, the distribution of eight bit errors in 1 RS symbol or in eight RS symbols requires very different level of protection, albeit they have the same average BERs. On the other hand, using longer codeword significantly increases the encoding/decoding complexity of RS code. If we encode the entire packet into one RS codeword, the computation it takes will be 64 times what of using $RS(255, w)$ [21].

Upon receiving the packet, the receiver first separates the redundancies from the payload (using the payload length and value of redundancy number w in the header) and then performs

error recovery with the redundancy if the packet fails the CRC check. When the error recovery succeeds, the decoded packet is passed to the upper layer. Otherwise, the receiver explicitly requests a retransmission.

3.2 Evaluation

In this section, we first use our test-beds to experimentally evaluate our BER estimator, which is the prerequisite that UnPKT can perform well. We then compare UnPKT with the state-of-the-art approaches using extensive trace-driven evaluations.

Test-beds. We use Atheros 9580 NICs that support 802.11n 20/40MHz channelization and operate in both 2.4GHz and 5GHz frequency band. The Wi-Fi NIC is configured to report CSI value for every non-empty subcarrier, *i.e.*, 56/114 subcarrier using 20/40MHz channels. Other information associated with the received packet, including the payload, RSSI, data rate and time-stamp, is recorded together with reported CSI [96]². We have developed and released an open-source toolkit that works with Ubuntu system [7].

We deploy Atheros 9580 nodes in three different test-beds in campus—an indoor office with 16 experimental locations, a parking lot surrounded by the cars and stores with nine experimental locations, and an open lecture hall with five experimental locations. The three test-beds are typical indoor 802.11 network environments with different degrees of frequency selective fading.

3.2.1 BER estimation evaluation

In Section 3.1.1, we maintained a stable experimental environment to measure the packet bit BERs over each marginal link in the three test-beds. Corrupted packets are collected from 76 marginal links and the payload of each packet is 1000-bytes. We hence have 608,000

²We develop the 802.11n CSI tool, instead of using the existing one based on Intel 5300 NICs [32], as corrupted packets are not accessible with Intel 5300 NICs, which are however useful for the evaluation in this report.

Figure 3.10: Cluster-level BER ratio between estimations and measurements.

(= $76 \times 1000 \times 8$) bit-wise BER estimations. As the packet protection of UnPKT is performed in the granularity of clusters, for each corrupted packet, we calculate the BER of each cluster, i.e., the average BER of all the bits in the same cluster. We then calculate the ratio between the estimated BER and the measured BER. Figure 3.10 plots the CDF of the BER ratio. The optimal estimation result yields to the ratio always being one, and so we see that the estimation of UnPKT in general is accurate. According to the statistics, we observe that about 80% and 50% of BER estimations are within 0.5 and 0.25 of one order of the magnitude compared with the BER measurements.

Computation overhead. UnPKT is accurate in the BER estimation. UnPKT can be practically useful only when its computation is efficient as well. We evaluate UnPKT's computation overhead on the Qualcomm System-On-Chip QCA9558 platform with a MIPS74Kc 720MHz CPU, which is a middle-end platform for 802.11 devices. We use the absolute computation delay as the performance metric and evaluate UnPKT for all the 802.11n SISO data rates in Figure 3.11. Overall, the computation delay prolongs as the data rate increases. It is because more codeword bits are contained in one OFDM symbol with a higher data rate for UnPKT to process. As an exception, the computation delay for the data rate 65Mbps is relatively low.

Figure 3.11: Computation delay of the BER estimation in UnPKT.

The reason is that only this rate uses the $\frac{5}{6}$ convolutional coding rate, which tolerates two dense codeword bit errors merely. Thus, for this rate, we only calculate $e_k \cdot e_{k+L-1}$ in Eq. (Eq. 3.2), which significantly accelerates the computation.

From Figure 3.11, we see that the BER estimation delay in UnPKT is tens of micro-seconds. The maximum and average delays are $36.0\mu s$ and $14.6\mu s$, respectively. Moreover, our implementation shows that the RS code encoding introduces an additional delay within $100\mu s$ and $200\mu s$. To put them into context, in 802.11, the DIFS duration before transmitting a packet is $34\mu s$. Mainly limited by the RS code encoding efficiency, the UnPKT protection cannot be finished before this deadline. However, the round-trip delay of one 802.11 transmission and the acknowledgement takes $500\mu s$ usually. UnPKT instead can protect the the packet after the next, using the current CSI measurement, whose channel states are still in the coherence time for most of the time. We envision that this limitation will vanish in the future as the hardware becomes more capable.

3.2.2 Trace-driven evaluation

In this subsection, we evaluate the overall performance of UnPKT using trace-driven simulations and compare it with state-of-the-art approaches.

3.2.2.1 Setup

The simulator is built on the GNU Radio platform based on the 802.11n PHY-layer specification, including the convolutional code, block interleaver, and OFDM modulation. We implement the convolutional coding rates from $\frac{1}{2}$ to $\frac{5}{6}$, and modulations from BPSK to QAM-64. For the data rate selection, we implement the most recent scheme ESNR [32]. We also implement the RS code encoder and decoder. The link quality between each pair of transceiver is directly from the CSI measurement of all the 166 links in our three test-beds collected in Section 3.1.1. Each CSI contains 52 sub-carriers, serves as the ground truth for the link quality, and is feeded to the simulator. We include the non-marginal links in the evaluation because we implement the full version of UnPKT which can augment the data rate (Section 3.1.2). In the evaluation, the sender transmits packets to the receiver and we measure the goodput achieved over each link.

3.2.2.2 Approaches for comparison

In addition to UnPKT, we also implement the following approaches for comparison.

802.11. The default 802.11 transmissions, which retransmits at most 7 times after the original transmission fails.

EqFEC. EqFEC [22] empirically adds a MAC-layer FEC to protect packets in narrow-band channels. For a fair comparison, we provide the packet-level BER to EqFEC. We also allow EqFEC to augment data rate the same as UnPKT.

MaNell. MaNell is short for Maranello in [35], which is a partial packet recovery approach with the best reported performance. Therefore, we do not compare UnPKT with other partial packet recovery approaches, *e.g.*, ZipTx [52], explicitly. MaNell divides a packet into blocks and only retransmits erroneous blocks after a transmission fails.

OPT. OPT adds the most appropriate MAC-layer FEC to each packet and completely avoids transmission failure.

3.2.3 Experimental Results

Goodput gain. Figure 3.12 examines the per-link goodput gains of EqFEC, MaNell, UnPKT, and OPT, normalized to the goodput achieved by 802.11 in the three test-beds. As the packet-level BER cannot fully represent the diverse packet bit BER distribution in the wide-band transmissions, the goodput gain of EqFEC is only 9.5% on average. In particular, it may perform slightly worse than 802.11, if the redundancies cannot recover the corrupted packets, especially when the data rate is augmented, while it introduces communication overhead to each packet. MaNell achieves 11.2% median and 40% maximal goodput gains over 802.11. The goodput of MaNell is limited mainly because the number of retransmissions in MaNell is still high. Benefiting from the appropriate unequal packet bit protection, the performance of UnPKT is within 4% of OPT. According to statistics, UnPKT outperforms 802.11, EqFEC, MaNell by 23.4%, 13.9%, and 12.2% on average, and 200%, 60%, and 49% at most.

Goodput gain analysis. We first analyze the goodput gain achieved by UnPKT, which is from the transmission failure avoidance over both the original and the augmented data rates. Figure 3.13 depicts the breakdown of the goodput gain. Transmission failure incurs the channel contention delay, packet retransmission delay, ACK feedback delay, etc. In addition, retransmissions usually adopt lower data rates. From Figure 3.13, we see that the transmission failure avoidance over the original data rate accounts for 58% of the goodput gain. As not all the data rates with high PRRs are augmented (Section 3.1.2), the goodput gain from the data rate augmentation is 42%. After the data rate is augmented, packets will be transmitted over the marginal links again. UnPKT can still prevent transmission failure and harness the extra goodput from the higher data rate.

To understand the goodput gains achieved by different approaches, we plot the average number of transmissions needed to delivery one packet from the sender to the receiver in Figure 3.14. From the result, we see that about 10% of packets need retransmissions in 802.11 and MaNell. However, the distribution in 802.11 suffers from a long tail, which leads to significant

Figure 3.12: CDF of the normalized per-link goodput.

Figure 3.13: Goodput gain breakdown of UnPKT.

Figure 3.14: CDF of transmissions to delivery a packet.

retransmission overhead to decrease the goodput. As the packet-level BER cannot precisely guide the amount of added FEC, the reduction in retransmissions is only slight. EqFEC cannot well prevent transmission failures, especially when the data rate is augmented. In UnPKT, the unequal packet protection bits can be properly supplemented. As a result, only about 1% of packets needs retransmissions.

In Figure 3.15(a), we further plot the percentages of the data rates used by different approaches in the evaluation. The x -axis represents all the single-stream data rates in 802.11n. As 802.11 and MaNell do not augment data rates and we adopt the same rate selection scheme for the four approaches, their rate selection choices are identical. Similarly, EqFEC and UnPKT have the same rate selection. From the figure, we see that more rate selections are given to the five highest data rates in EqFEC and UnPKT, which lead to potential higher goodput. As UnPKT precisely protects error-prone bits and prevents transmission failures, it can fully harness this goodput improvement opportunity.

Communication overhead. In Figure 3.15(b), we further analyze the communication overhead of each approach, which is measured by the extra error correction bits needed to deliver one 1000-byte packet. In 802.11, when a transmission fails, the sender retransmits the entire packet. As a result, the communication overhead is as high as 152 bytes in our evaluation. As the MAC-layer protection of EqFEC can prevent some transmission failures, the communication overhead of EqFEC is smaller than 802.11. However, EqFEC still suffers from non-negligible retransmissions, leading to high communication overhead. MaNell has the smallest

Figure 3.15: (a) Fractions of data bit-rates used in different approaches; (b) Communication overhead of each approach.

communication overhead. This is because when a transmission fails, MaNell only retransmits the blocks containing error bits. However, MaNell does not reduce the number of retransmissions needed. Hence, its goodput is still limited. OPT adds the most appropriate protection to each packet and completely prevents the transmission failure. OPT thus also introduces communication overhead. Due to the accurate BER estimation, UnPKT has only slight communication overhead, which is close to OPT.

3.3 Discussion and limitation

UnPKT provides unequal protection for Wi-Fi transmission in SISO configuration and demonstrates significant improvement over existing framework. However, in our current system design, we didn't cover the MIMO configuration. MIMO is widely adopted by nowadays Wi-Fi systems for improving throughput. It will be of great importance for us to extend our system to work in MIMO configuration. Another limitation of UnPKT is that, we only consider the convolutional code as the coding scheme, while some Wi-Fi standards such as 802.11 ac employs other coding schemes, such as the LDPC codes. In order to cope with diverse Wi-Fi standards, UnPKT should incorporate LDPC in the system design.

Chapter 4

Improving the Sensing Resolution of COTS Wi-Fi Devices

4.1 Design

We detail the design of our system, *i.e.* Splicer. We begin with the theoretical foundation of CSI splicing. Then, we discuss the problem we met when implementing Splicer atop of a practical Wi-Fi system and the algorithms we propose to solve those challenges.

4.1.1 Principle of CSI Splicing

In this section, we give the theoretical foundation for CSI splicing. According to [29, 65], the channel frequency response $h(f)$ can be expressed by Eq. (Eq. 4.1):

$$h(f) = \sum_{l=0}^L \alpha_l \cdot e^{-j \cdot 2\pi \cdot f \cdot \tau_l}, \quad (\text{Eq. 4.1})$$

where L is the total number of multipaths, α_l and τ_l stand for the attenuation and the propagation delay of the signal through path l , respectively. Figure 1.2 (a) depicts a channel frequency response when the channel bandwidth is B , *e.g.*, $f_0 \leq f \leq f_0 + B$. Channel frequency response is reported in the form of CSI in 802.11 WiFi, which is a set of discrete channel frequency response samples. With the sampling rate $F_f = \frac{1}{\Delta f}$, where Δf is the sampling resolution in the frequency domain, a receiver can obtain $M = \frac{B}{\Delta f}$ channel frequency response samples, and each sample contains amplitude and phase information.

Figure 4.1: Power delay profiles derived using two CSI with an amplitude offset of 7dB.

To obtain the power delay profile, the CSI can be transformed to the channel impulse response $f(t)$ by IFFT:

$$f(t) = \sum_{l=0}^L \alpha_l \cdot \delta(t - \tau_l), \quad (\text{Eq. 4.2})$$

where $\delta(\cdot)$ is the delta function, and L , α_l , and τ_l have the same definitions as they are in Eq. 4.1. Figure 1.2 (b) illustrates the channel impulse response transformed from Figure 1.2 (a). The norm of $f(t)$, $\|f(t)\|_2$, then gives the power delay profile, which describes the power levels of each multipath with different propagation delays.

Feasibility of CSI splicing. According to Eq. 4.1, given one multipath channel, *i.e.*, given each α_n , τ_n , and N in Eq. 4.1, and channel bandwidth B , the channel state information is *deterministic* at each frequency f . We can thus obtain all M CSI samples from either a single measurement covering the entire bandwidth or multiple measurements where each measurement covers a subset of M samples. With the M samples, we can derive a unique power delay profile using Eq. 4.2 and the norm operation.

Resolution of power delay profile. After the IFFT transformation, we obtain a series of signal samples in the time domain with various delays τ_l in Eq. 4.2. The norm of each multipath

Figure 4.2: Power delay profiles derived using two CSIs with raw phases and average amplitude of the CSI traces in Figure 4.1.

component, $\|\alpha_l \cdot \delta(t - \tau_l)\|_2$, indicates its power level as shown in Figure 1.2 (b), where the first impulse corresponds to the Line-of-Sight (LoS) path. According to the IFFT theory, the time resolution $\Delta\tau$ of power delay profile is connected to the sampling resolution Δf of the channel impulse response, *i.e.*, $\Delta\tau = 1/(N \cdot \Delta f)$, where N is the IFFT length. As $N \cdot \Delta f = B$, we have $\Delta\tau = 1/B$, where B is the bandwidth. Such a connection indicates that a wider bandwidth CSI leads to a higher resolution of power delay profile.

Given channel bandwidth B , two multipaths of propagation delays τ_1 and τ_2 are not distinguishable if $|\tau_1 - \tau_2| < 1/B$. Hence, all multipaths whose propagation delays differences are less than $1/B$ are viewed as one multipath component in the power delay profile, and the corresponding power level indicates the aggregated power level of those multipaths. As a result, the time resolution $\Delta\tau$ leads to $\frac{c}{B}$ uncertainty in terms of the length difference between non-distinguishable paths, where c is the speed of the signal propagation. In 802.11 WiFi in a 20MHz or 40MHz channel, the path length uncertainty is 15m or 7.5m, respectively, which can merely support coarse mobility tracking and activity recognition.

4.1.2 CSI splicing in practice

Theoretically, CSI can be spliced together directly. In practice, the CSI measurement results reported by the Wi-Fi NIC contains severe errors. If we directly derive the power delay profile from raw CSI measurements, the error will be passed to the power delay profile we obtained. Therefore, we investigate the impact of CSI amplitude and phase errors to the derived power delay profile.

CSI measurement errors. We perform preliminary CSI measurements to investigate how CSI measurement errors will affect the derived power delay profiles. We use Atheros 9580 NICs that support 802.11n with 20MHz/40MHz channels at the 2.4G/5G frequency band, and modify the driver to extract CSI from the physical layer. We configure Atheros nodes to transmit packets with minimum payload to ensure a short transmission delay, *i.e.*, about $0.2ms$ in our experiment. We collect the CSI traces from one 802.11n 40MHz band as well as three 20MHz bands. The measurement results are reported in Figures 4.3 and 4.4.

CSI amplitude. Figure 4.3 reports the results of our initial measurement study in 802.11n (detailed experiment settings reported in §4.1.2). The CSI measurements are complex values and hence contain two parts, the amplitude and the phase. Figure 4.3 (a) presents the CSI amplitude measurements (multiple times) from one 802.11n 40MHz band and Figure 4.3 (b) presents the CSI amplitude measurements from three 802.11n 20MHz bands that together cover the same 40MHz band in Figure 4.3 (a). From Figure 4.3 (a) and (b), we can observe obvious offsets between the measured CSI amplitudes at different times. After removing the amplitude offsets among all measurements, we see that the spliced CSI from 20MHz bands can be very similar in its shape to the one measured from the 40MHz band (Figure 4.3 (c) and (d)).

How would the offsets between CSI amplitudes affect the derived power delay profile? To answer that question, we select two arbitrary CSI traces from the same WiFi band (20MHz-2 in Figure 4.3) with an amplitude offset of 7dB, and derive two power delay profiles¹. Figure 4.1

¹To isolate the impact of the CSI phases, we use the average phases of the two CSI traces such that the derived power delay profiles only differ in the amplitude.

Figure 4.3: CSI amplitude measurements. Raw CSI amplitudes from (a) one 40MHz 802.11n band and (b) three 20MHz bands; Amplitudes after the offset removal from (c) the 40MHz band and (d) three 20MHz bands.

shows that although two derived power delay profiles have different power levels, *e.g.* the average difference is 7.05 dB, they follow similar shapes. We compute the variance of the power difference for each path to quantify the similarity of two power delay profiles, which is less than 1.0 dB. We observe similar results from other CSI combinations. All of these results indicate that the derived power delay profiles approximately characterize the same multipath channel environment except that their power levels are scaled due to amplitude offsets of the CSI measurements.

CSI phases. Compared with the amplitude splicing, the phase splicing may result much severer errors. Figure 4.4 depicts the phases of the same CSI traces in Figure 4.3. The raw CSI phases² measured at different times have offsets as well (Figure 4.4 (a) and (b)). However, even we remove their mutual offsets when splice the traces, the residual phases do not have a

²A raw CSI phase is in the range of $[-\pi, \pi)$. For a clear representation, we expand the measured CSI phases θ , using $\theta = \theta \pm 2k\pi$, to the range of $[-\infty, +\infty]$ across different channels.

Figure 4.4: CSI phase measurements. Raw CSI phases from (a) the 40MHz band and (b) three 20MHz bands; Phases after the offset removal from (c) the 40MHz band and (d) three 20MHz bands. The CSI traces in this Figure are the same with Figure 4.3.

common shape, demonstrating diverse phase shifts in different sub-carriers. Consequently, the multiple instances of the 40MHz CSI phase measurements as depicted in Figure 4.4 (c) do not match each other. The CSI traces from 20MHz bands cannot match the 40MHz measurement neither. To derive an accurate power delay profile, such phase shifts must be precisely compensated because the phase value falls in a small range of $[-\pi, \pi)$, and a slight phase error will result in significant inaccuracy in the power delay profile (as we will demonstrate in §4.1.2).

Similarly, we use the raw phases of two CSI traces in Figure 4.1 to derive two power delay profiles³. The two profiles in Figure 4.2 demonstrate opposite results. One power delay profile indicates the existence of the LoS between the transmission pair (*i.e.*, the first multipath component has the strongest power level), however another one indicates that there is no LoS (NLoS) path (*i.e.*, the first arrived signal is much weaker in strength than later arrived signals).

³Similar to the experiment in Figure 4.1, we use the average amplitude of the two CSI traces to derive power delay profiles to avoid the impact from the amplitude.

Figure 4.5: Illustration of signal processing in 802.11.

In addition, the power levels of each multipath component in these two profiles are very different, *e.g.*, the power difference of the LoS path is more than 10 dB and the variance of the power level differences is up to 4.7 dB. Figure 4.2 indicates that the CSI phase errors will significantly impact the derived power delay profiles, which completely change both the power loss and the multipath channel features.

4.1.3 Sources of CSI measurement errors

As our initial experiment results suggest, the CSIs collected from WiFi NICs are mixed with rich hardware distortions. The raw CSIs cannot derive accurate power delay profiles. In this section, we identify the measurement error sources. To do that, we start with a brief introduction to the signal process components in 802.11 physical layer before the CSI calculation. We then analysis how each of them impact the received signal and become the error source of the calculated CSI. At last, we introduce the error type that introduced by different error sources, *e.g.*, linear or constant.

Figure 4.5 illustrates the wireless signal processing in the 802.11 NICs. An incoming signal from the antenna is down converted to the base band signal $s(t)$ and sampled by Analog-to-Digital (ADC) to derive the digital $s[n]$. The packet boundary detector (PBD) performs correlation between $s[n]$ and a pre-defined 802.11 preamble pattern to confirm an incoming

packet. Once the preamble of a packet is detected, the signal central frequency is calibrated by the central frequency offset (CFO) corrector. The OFDM receiver estimates the CSI based on the calibrated $s[n]$ and the CSI is passed to the subsequent equalization module (not shown) to compensate errors prior to the packet decoding. Due to the hardware imperfection, the CSIs measured by NICs introduce the following errors.

Power control uncertainty. Limited by the hardware resolution, Automatic Gain Controller (AGC) cannot perfectly compensate the signal amplitude attenuation to the transmitted power level. The measured CSI amplitude equals to the compensated power level, which is mixed with the power control uncertainty error. According to [41], the CSI amplitude offsets in individual bands can be removed by averaging. However, if the number of CSI measurements on each channel is not sufficient, which may be the usual case due to the stringent delay constraint (§4.1.7), the averaging cannot perfectly eliminate the power uncertainty. The residual offset between different WiFi bands disallows a direct CSI amplitude splicing.

Sampling frequency offset (SFO). The sampling frequencies of a transmission pair exhibit an offset due to non-synchronized clocks, which can cause $s[n]$ after ADC a time shift τ_o with respect to the transmitted signal. Because clock offsets are relatively stable within a short time, τ_o will introduce near constant errors λ_o to the CSI phases measured from different sub-carriers.

Packet boundary detection (PBD) error. Due to correlator sensitivity of packet detector, the packet detection introduces another time shift τ_b , with respect to the transmitted signal [28, 77]. The timing shift τ_b causes random errors λ_b to the measured CSI phases.

Central frequency offset (CFO). The central frequencies of the transceiver cannot be perfectly synchronized. The central frequency offset is compensated by the CFO corrector, but due to the hardware imperfection, the compensation is incomplete. Signal $s[n]$ still carries residual errors, which can cause the CSI phase offsets β .

The last three error sources cause CSI phase measurement errors. Due to the diverse phase shifts from SFO and PBD, the phases of the overlapped sub-carriers measured from two consecutive bands are inconsistent (different), which impairs the CSI phase splicing. The CSI

phases measured from different bands also suffer from notable offsets. In the next section, we introduce our solutions to compensate above CSI errors.

4.1.4 Phase error correction

We denote S as the number of sub-carriers in one WiFi band. Based on [77, 78], the reported CSI phase value ϕ_k from any sub-carrier k by WiFi NICs can be expressed:

$$\phi_k = \theta_k + k \cdot (\lambda_b + \lambda_o) + \beta, \quad (\text{Eq. 4.3})$$

where θ_k is the phase rotation of subcarrier k which is caused by the channel propagation, λ_b and λ_o are phase errors introduced by the packet boundary detection uncertainty and the sampling frequency offset, respectively, β is the phase error caused by the central frequency offset, and $k = 1, 2, \dots, S$. As $\lambda_b + \lambda_o$ is multiplied by the sub-carrier index k in Eq. 4.3, the phase errors cross different sub-carriers are diverse among different CSI measures as shown in Figure 4.4. Our target is to obtain the phase value θ_k by eliminating the impact of other parameters, *i.e.*, the λ_b , the λ_o and the β . We focus on the removal of λ_b and λ_o from ϕ_k in the rest of this subsection, and introduce the removal of β when we splice the CSI phases in §4.1.5.

PBD phase error λ_b removal. Phase error λ_b is caused by the time shift τ_b from the packet boundary detection uncertainty. To investigate the effect of τ_b , we examine the discrete Fourier transform of the channel frequency response in Eq. 4.4:

$$h[k] = \sum_{n=0}^{N-1} f[n] \cdot e^{-j \cdot 2\pi \cdot k \cdot n / N}, \quad (\text{Eq. 4.4})$$

where $h[k]$ and $f[n]$ are the discrete versions of $h(f)$ in Eq. 4.1 and $f(t)$ in Eq. 4.2, respectively, and N is the IFFT length. With a time shift τ_b in $f[n]$, Eq. 4.4 can be rephrased as:

$$h[k] \cdot e^{-j \cdot 2\pi \cdot k \cdot \tau_b / N} = \sum_{n=0}^{N-1} f[(n - \tau_b)_N] \cdot e^{-j \cdot 2\pi \cdot k \cdot n / N},$$

where the term $e^{-j \cdot 2\pi \cdot k \cdot \tau_b / N}$ indicates that the time shift τ_b can introduce a phase error, $2\pi \cdot k \cdot \tau_b / N$, in each sub-carrier k . Therefore, $\lambda_b = 2\pi \cdot \tau_b / N$.

To remove λ_b from each ϕ_k , we leverage an observation that the time shift τ_b varies in each packet reception but follows a Gaussian distribution with the zero mean [77]. The error λ_b thus changes accordingly in different CSI measurements and $\lambda_b \sim N(0, \sigma^2)$, where σ is the standard deviation. According to the weak law of large numbers, λ_b can be removed by averaging over the measured CSI phases ϕ_k .

Figure 4.6: Phase differences cross sub-carriers and the distribution of $\Delta\lambda_b$. (a) The phase differences $k \cdot \Delta\lambda_b$ of 8 randomly CSI pairs; (b) Distribution of slope $\Delta\lambda_b$ calculated from 180 CSIs.

To validate this observation, we perform a trial of experiments in Figure 4.6. In Eq. 4.3, λ_b is mixed with λ_o , β , and θ_k in the CSI phase ϕ_k , and we cannot directly investigate its distribution. Therefore, for the collected CSIs from the same WiFi band, we calculate the mutual phase differences of those CSIs and obtain a set of $\Delta\theta_k + k \cdot (\Delta\lambda_b + \Delta\lambda_o) + \Delta\beta = k \cdot \Delta\lambda_b + a$ for each sub-carrier k^4 , where a is a constant. We thus examine the distribution of $\Delta\lambda_b$, because if $\lambda_b \sim N(0, \sigma^2)$, $\Delta\lambda_b$ should be a Gaussian with the zero mean as well. We collect 180 CSIs from a 20MHz channel within a short time interval when the environment is

⁴ λ_o is a constant that can be removed by the deduction, which is detailed in the SFO phase error removal.

stable. Figure 4.6 (a) plots the $k \cdot \Delta\lambda_b$ value versus the sub-carrier index k for 8 randomly selected CSI pairs (we omit the constant a for the presentation clarity). Figure 4.6 (a) shows each line is a straight line and the slopes of those lines are different. The result indicates that λ_b is a constant to each sub-carrier in each individual measure, but varies cross different measures. To further examine its distribution, in Figure 4.6 (b), we divide the range $[-0.075, 0.075]$ into 100 bins on the x -axis and plot the frequency of $\Delta\lambda_b$ falling into each bin on the y -axis. After the curve fitting, we find that $\Delta\lambda_b$ indeed follows a Gaussian distribution with the zero mean.

According to the λ_b distribution, we can remove it by averaging over the measured CSI phases ϕ_k s. In principle, more measurements lead to a better error removal, but it will prolong the latency to scan each single band. In §4.1.7, we will determine an optimal number of CSIs collected from each band to balance this trade-off subjected to the stringent channel coherence time. Given the optimal amount $\dot{n}_i$ for any band i , we calculate:

$$\overline{\phi}_k^i = \sum_{j=1}^{\dot{n}_i} \phi_k^i(j) / \dot{n}_i, \quad (\text{Eq. 4.5})$$

where $\phi_k^i(j)$ stands for the j -th CSI measure from band i . After λ_b is removed, $\overline{\phi}_k^i = \theta_k^i + k \cdot \lambda_o + \beta$. In the next subsection, we introduce how to remove λ_o from $\overline{\phi}_k^i$.

SFO phase error λ_o removal. Phase error λ_o is caused by the offset of the sampling frequencies of the sender and the receiver, f_s and f_r . We denote $\zeta = \frac{f_s}{f_r} - 1$ as the fractional difference in sampling frequency, and the effect of the sampling frequency offset is to introduce a term, $e^{j \cdot \zeta' \cdot k}$, to the channel frequency response: $h'[k] = h[k] \cdot e^{j \cdot \zeta' \cdot k}$, where ζ' stands for ζ multiplied with a constant and $h[k]$ is the channel frequency response without the sampling frequency offset (Eq. 4.4). Hence, $\lambda_o = \zeta'$. As the fractional frequency difference keeps stable in the order of minutes [40], λ_o is a constant during the process of the CSI splicing.

To remove λ_o from $\overline{\phi}_k^i = \theta_k^i + k \cdot \lambda_o + \beta$, obtained from Eq. 4.5, we leverage an observation that the power delay profiles derived from different WiFi bands should be the same after λ_o is removed (we will show in §4.1.5 that the phase offset β has no impact on the derived power

delay profile), since they characterize the same multipath channel. For any WiFi band i , the CSI phases $\bar{\phi}_k^i$ s from all S sub-carriers form a vector $\bar{\Phi}^i = [\bar{\phi}_1^i, \bar{\phi}_2^i, \dots, \bar{\phi}_S^i]^T$. Therefore, we propose to gradually “rotate” two distinct $\bar{\Phi}^i$ and $\bar{\Phi}^j$ in the frequency domain⁵ and stop when the two derived power delay profiles best match each other. We repeat this process for different pairs of WiFi bands to improve the accuracy. To quantify the likelihood of two power delay profiles, *e.g.*, P_1 and P_2 , we define their similarity as:

$$\rho(P_1, P_2) = \frac{1}{\|P_1 - P_2\|_2}, \quad (\text{Eq. 4.6})$$

where the dominator essentially measures the power level differences of each multipath component in the two power delay profiles. A large $\rho(P_1, P_2)$ value indicates that P_1 and P_2 are more similar.

Figure 4.7: The power delay profile similarities for two pairs of WiFi bands when ϵ varies from -0.1 to 0.1.

To illustrate this solution, we measure four different 20MHz WiFi bands, and compute the similarities for two channel pairs in Figure 4.7. We compensate λ_o by $\lambda_o - \epsilon$ and search for the optimal ϵ for both positive and negative directions. For the first pair, we observe four local

⁵We multiply $\bar{\Phi}^i$ and $\bar{\Phi}^j$ with $e^{j \cdot \epsilon \cdot k}$, and gradually vary ϵ .

maximum points, and for the second pair, we observe three local maximum points when ϵ varies in $[-0.1, 0.1]$. According to Algorithm 1, we can determine the final ϵ as the average of ϵ_1 and ϵ_2 in Figure 4.7. With the optimal ϵ obtained from Algorithm 1, for each $\bar{\Phi}^i$ after the PBD phase error λ_b removal, we can further remove λ_o by $\tilde{\Phi}^i = \bar{\Phi}^i - [\epsilon, 2\epsilon, \dots, S\epsilon]^T$, where S is the number of subcarriers.

Algorithm 1: SFO Phase Error Compensation

- 1 **for** each WiFi band pair c **do**
 - 2 Record the top-two local maximal similarity values and the corresponding ϵ :
 $\langle \epsilon_1^c, \rho_1^c \rangle$ and $\langle \epsilon_2^c, \rho_2^c \rangle$.
 - 3 Clustering on ϵ and find the cluster with the maximal similarity sum.
 - 4 Return the cluster center as the final ϵ value.
-

Although the ϵ searching introduces extra computational delays, Algorithm 1 does not need to be executed parallel to the CSI sampling in real-time. As a matter of fact, once sufficient CSIs can be obtained subjected to the stringent channel coherence time (§4.1.7), the latency of Algorithm 1 only impacts the frequency to generate power delay profiles to the upper-layer applications. We evaluate the computational efficiency in §4.2.

Figure 4.8: CSI phases after the removal of λ_s and λ_o .

4.1.5 CSI phase splicing

Figure 4.8 plots the corrected CSI phases at this stage for the three 20MHz channels in Figure 4.4 (b). From the result, we observe that after the phase error removals of λ_b and λ_o , the shapes of the overlapped sub-carrier phases from different WiFi bands now become similar and consistent. The only barrier that remains is the offsets. In this subsection, we target to removing offsets to finally enable the phase splicing.

CSI phase offset β removal. Phase offset β is caused by the central frequency offset of the transmission pair. Through our study, we find that for individual WiFi bands, phase offset β has no impact on the derived power delay profile, *i.e.*, given a pair of CSI amplitude (ω) and phase (θ), the power delay profile derived by ω and θ is identical to the one derived by ω and $\theta + \beta$. The reason is that offsets are frequency independent. After IFFT, the error will result in a constant phase rotation term in each α_l of Eq. 4.2, which will not change the norm operation result. Hence the power level of each multipath component keeps unchanged. According to this observation, we can use the phase measured from any band as a reference and compensate β by calibrating the phases measured from other bands with respect to the reference.

Figure 4.9: Spliced CSI phase refinement.

Spliced CSI phase refinement. So far, the CSI phases measured from different channels can be spliced already. However, we find that the CSI phase accuracy can be further improved by leveraging the primarily spliced result. In the SFO phase error λ_o removal, we rely on the similarity of the derived power delay profile to compensate λ_o , but the derived power delay

profiles are based on the CSI from single WiFi bands with limited bandwidth. The phase error λ_o thus cannot get fully corrected using low-resolution power delay profiles. To refine the phase information, in the phase splicer, we further manually divide the spliced channel bandwidth into multiple windows illustrated in Figure 4.9. Each window has a much wider bandwidth, denoted as l , than each single WiFi band. We slide the window and obtain a set of l -wide phase pieces. For each phase pair, we call Algorithm 1 to estimate the SFO phase error compensator ϵ , and use their average value to further compensate the spliced CSI phase.

Figure 4.10: Empirical investigation of window size l .

In Figure 4.9, the window size l balances a trade-off. A larger l will lead to a higher-resolution power delay profile, which potentially can better compensate the phase error λ_o . However, with a large l , the channel information carried by each divided CSI piece will be highly redundant (more overlapped sub-carriers). Two such power delay profiles will produce a large body of local maximal points in Figure 4.7, which are dominated by the channel information redundancy, instead of the removal of the phase error λ_o . It thus prevents to find the optimal ϵ to compensate λ_o . Therefore, the length l needs to be carefully selected. In Figure 4.10, we investigate this trade-off. The experimental setting is detailed in §4.2, where we use ranging accuracy as the metric for the evaluation. We set the window size l as a fraction of the total bandwidth that

the spliced CSI covers. Initially, when we increase the window size, the ranging accuracy improves because λ_o is better compensated by higher-resolution power delay profiles. However, when l is excessive large, *e.g.*, close to 0.95, the accuracy decreases due to the unreliability in the optimal ϵ search. According to Figure 4.10, we set l to be $\frac{3}{4}$ of the total bandwidth that the spliced CSI covers as default in Splicer.

4.1.6 CSI amplitude splicing

In Figure 4.3, we have shown that the amplitudes of raw CSIs also exhibit significant offsets. The reason is that the power control uncertainty [33, 41], which also follow a Gaussian distribution. However, different from the CSI phases, the power uncertainty is frequency band independent, *i.e.*, the amplitudes measured from different bands follow the same distribution. Thus, for CSI amplitudes, there is no need to average them for individual WiFi bands. Instead, we can average the amplitudes after we collect all CSIs for the splicing. The total number of CSIs to collect for splicing is determined by the channel coherence time, which will be discussed and given in the next subsection.

So far, both the CSI amplitudes and phases both can be spliced. Figure 4.11 depicts the splicing result covering the entire 802.11n WiFi band, 200MHz, obtained from our experiments in §4.2.

4.1.7 Battle the coherence time constraint

Wireless channels are time-varying, which enforce a stringent time budget for each round of CSI splicing, since the spliced CSI is valid only when the channel condition is relatively stable. In this subsection, we first estimate the minimum number of CSIs to collect from each individual band that can fully compensate phase errors, and then propose an efficient CSI sampling scheduler to balance the trade-off between the error compensation quality in each individual band and the total bandwidth that can be afforded for the CSI splicing within the time budget.

Figure 4.11: CSI splicing result. (a) The spliced CSI amplitude; (b) The spliced CSI phase.

Stringent time budget. The channel coherence time T_c can be expressed as $T_c \approx \frac{1}{2f_d}$, where f_d stands for the Doppler shift. Previous works [29] have studied the Doppler shifts caused in typical mobile environments. For instance, when people are walking, the Doppler shifts are usually less than 12Hz, which can be translated into $T_c = 40ms$ with 2.4GHz WiFi. In Splicer, the transmission delay of each packet (with minimum payload) is around $0.2ms$, *e.g.*, approximately 200 CSIs can be collected within the time budget. In our current design, we rely on the empirical results from the literature to set T_c . We leave the cooperation with advanced channel coherence time measurement schemes [45] in Splicer as the future work of this study.

CSI measurements for each band. As aforementioned in §4.1.4, for each WiFi band, we need to collect sufficient CSI measures to fully compensate phase error λ_b , which is caused by the signal boundary detection uncertainty and follows a Gaussian distribution. According to the weak law of large numbers, more CSIs lead to a better compensation. However, we have a stringent time budget T_c to scan the entire WiFi band, which on the other hand limits the number of CSI collected from each WiFi band. To deal with this trade-off, we first investigate the minimum number of CSIs for each band that can achieve a given confidence level.

When we collect n CSIs from one band, for any sub-carrier k , we can calculate the average phase value $\bar{\phi}_k$ by Eq. 4.5. According to [14], we can define a confidence range $(\bar{\phi}_k - \frac{\sigma}{\sqrt{n}}z_{\alpha/2}, \bar{\phi}_k + \frac{\sigma}{\sqrt{n}}z_{\alpha/2})$, where σ is the standard deviation of λ_b , α is an error rate, and z_j is a normal distribution related parameter that can be obtained from a table [14]. When n increases, the range shrinks, *i.e.*, the confidence increases. The theory in [14] proves that the probability that $E[\phi_k]$ falls into this range is greater than $1 - \alpha$. Therefore, given α and the confidence range length r , the minimal number of CSIs to collect, $\hat{n}$, can be determined when $\frac{\sigma}{\sqrt{\hat{n}}}z_{\alpha/2} \leq r/2$. We want to note here that, initially we use the standard deviation σ of last band as the value for current band, since we observe that the distribution of neighboring band is similar with each other. The standard deviation σ is updated after we collected multiple CSIs in current band.

We have an assumption in above algorithm that the CSI phase value ϕ_k follows a Gaussian distribution. However, according to Eq. 4.3, phase value ϕ_k consists of several random variables, namely λ_b , λ_o and β , and only λ_b follows a Gaussian distribution. To deal with this problem, we need to preprocess the original phase value ϕ_k before we can apply the algorithm discussed above. First of all, as we mentioned in §4.1.4, λ_o is a constant value across multiple CSI measurements for a lone time period, therefore, phase value $\lambda_o + \lambda_b$ follows an Gaussian distribution. On the other hand, β is neither a constant value across different CSIs or follow any distribution since it is caused by the residual CFO, but is a constant for one CSI measurement across different subcarriers. If we look at the phase difference of subcarriers with index $k = 0$ of two CSI phase ϕ_1 and ϕ_2 , we have

$$\begin{aligned} \phi_{dif} &= \phi_1 - \phi_2 \\ &= \beta_1 - \beta_2 \end{aligned} \tag{Eq. 4.7}$$

where β_1 and β_2 are the phase error β contained in two CSIs. From Eq. 4.8, we obtain the difference of two β contained in two different CSIs. Based on the difference, we can process the two CSIs so that they contained exactly the same β . Furthermore, if we use on CSI measurement as anchor, then we could make all the measured CSI contain the same amount of

phase error, we denote as β_a . After the processing, the reported phase value ϕ actually follows a Gaussian distribution and we could finally apply our algorithm atop of it.

To cover the entire WiFi band, we need to scan multiple (*e.g.*, C) individual WiFi bands, *e.g.*, $C = 5$ for 802.11n at 2.4GHz. However, due to the stringent time budget, it could be infeasible to collect all $C \times \dot{n}$ CSIs within T_c . Therefore, we propose to balance the trade-off between the error compensation quality in each individual band and the total bandwidth that can be afforded for the CSI splicing. Through our study, we observe that for any three consecutive WiFi bands, if the CSI phases in the first and the third channels are well compensated, they can serve as two anchors to further calibrate the CSI phase from the second band if its phase error is not fully corrected. In particular, we can rotate the CSI phase from the second band and stop when its phase differences for the overlapped sub-carriers with other two bands are minimized. With this observation, the scheduling of the CSI measurement for each band is as follows. We have three constraints:

- Collecting $\dot{n}$ CSIs from a set of WiFi bands whose indices are odd. They form anchors.
- For each even-indexed band in between, we set a lower bound that the number of CSIs collected from those even-indexed bands is greater than this bound.
- All CSIs are collected within the time budget T_c .

The scheduling objective is to maximize the total number of bands, both odd and even, that can be used for the CSI splicing. After solving this optimization problem, we can determine the optimal CSI collection assignment. When enough CSIs are collected from each band, the time delay handler informs the physical layer to send an ACK such that the sender and the receiver switch to the next band synchronously. As a contingency plan, the sender and receiver will switch back to the first channel if no packets received for a given time-out duration.

Early termination. Due to the wireless channel dynamics, the time budget we adopt from the literature may not always precisely capture the channel coherence time. It is possible that

the wireless channel changes dramatically in the middle of CSI collection so that the CSIs from rest WiFi bands will become useless for splicing. To address this issue, we propose an early termination strategy to detect such a case in real time, which leverages the following observation. If the channel is stable, θ_k for any sub-carrier k of a WiFi band in Eq. 4.3 keeps unchanged. When we collect multiple CSIs from this band and compute their phase differences, we can obtain a set of straight lines as shown in Figure 4.6. Later, when the channel condition is changed, if we collect another CSI trace from the new channel and compare the phase difference with the CSI from the previous channel, the result demonstrates not a straight line. Therefore, we can apply the linear regression on the phase difference and use the regression error to indicate the level of the channel condition change.

In light of this, after receiving all CSIs for each round of CSI splicing, the receiver will not directly use them. Instead, for each individual WiFi band involved in the splicing, the receiver will compute the phase difference between the first and the last CSIs collected from this band, and apply linear regression to check the regression error. If the error is greater than a threshold, the receiver will discard all CSIs from this band, as well as all subsequent bands (we will evaluate its effectiveness in §4.2). CSI splicing only utilizes non-discarded bands.

4.2 Evaluation

In this section, we conduct testbed-based experiments to evaluate the performance of Splicer. We introduce our experimental setting in §4.2.1, evaluate the efficacy of Splicer in §4.2.2, and report the end-to-end system performance of Splicer-enhanced CUPID in §4.2.3, which is one of the state-of-the-art indoor localization designs [70].

4.2.1 Experimental setup

We conduct experiments in a laboratory, which is a typical indoor office environment. We install five APs at five known locations. Each AP is a laptop connected to an Atheors 9580

NIC. The five APs are configured in the monitoring mode as receivers. We select 500 different locations in the laboratory and place another AP, which is set to the RootAP mode as sender, at each of these locations sequentially. At each location, the AP sender conducts multiple rounds of CSI splicing to each of the five AP receivers. To examine the accuracy of the power delay profiles derived by Splicer, we first evaluate the accuracy of the power level measured from the Line-of-Sight (LoS) path. To this end, we measure the distance between the sender and each of the receivers at all 500 different locations as the ground truth. At each location, we compare the derived distance from the power delay profile with the ground truth for evaluation. In addition to the LoS path, we also evaluate the quality of the derived power delay profiles using the power level stability for Non-Line-of-Sight (NLoS) paths.

We first provides a detailed performance analysis of Splicer in 4.2.2, we disable the sampling scheduler (described in 4.1.7), and manually control the number of scanned channels. From each channel, we collect 30 CSI traces. After that, in 4.2.3, we enable the sampling scheduler and evaluate the end-to-end performance with the full-version Splicer in a localization application.

4.2.2 Results

Phase error correction. To investigate the effectiveness of the CSI error correction designs in Splicer, we first evaluate the ranging performance to estimate the LoS path length d in each single 20MHz WiFi band without using CSI splicing. We evaluate Splicer for three different versions to investigate where the performance gains Splicer achieves come from: the full version with both λ_b and λ_o compensations, as well as two degraded versions — without λ_b compensation and without any phase error compensation. We compare the three versions against the measured ground truth to one randomly selected receiver in Figure 4.12. In the figure, the x -axis presents different locations and the y -axis illustrates the ranging results of each Splicer version (as well as the ground truth). Each reported accuracy value is the average

Figure 4.12: Ranging results at 500 different locations by different versions of Splicer in a single WiFi band.

from 10 measurements. From the result, we observe that without any phase error correction, the ranging performance is highly unreliable. After the phase error λ_b removal, the ranging accuracy Splicer achieves has been dramatically improved, but it is still far away from the ground truth. After the compensations of λ_b and λ_o , we find that the accuracy is very close to the true distance value between the sender and the receiver at different locations.

In Figure 4.13, we provide detailed statistical results for the performance achieved by different Splicer versions in Figure 4.12. From the result, we see that when the raw CSI phases are used, the ranging error is $10.7m$ for 80% of the measurements. The median and the maximum errors are $6.1m$ and $24m$, respectively, compared with the ground truth. After the compensation of phase error λ_b , the ranging error is reduced to $6.3m$ for 80% measurements. After the removal of both λ_b and λ_o , the ranging error is less than $4.3m$ for 80% of cases. The improvement is as high as $4.8m$ on average compared with the traditional ranging performance using the raw phase information.

CSI splicing. In Figure 4.14, we evaluate the ranging performance of Splicer after we enable CSI splicing from different WiFi bands. As the power delay profile resolution is determined by the channel bandwidth, we select two representative bandwidths after splicing, 200MHz that is

Figure 4.13: CDF of ranging errors from a single band.

the total bandwidth allocated to 802.11n and 120MHz that is in between the entire WiFi band and each single WiFi band. We select all spliced CSIs with these two selected bandwidths to report the performance. As a benchmark, we also include the performance of the full version Splicer in 20MHz for comparison.

Figure 4.14 plots CDF of the ranging errors from those three approaches. From the result, we find that the performance of Splicer using merely a single WiFi band is still limited, even with the CSI error correction. In general, the wider the bandwidth Splicer uses, the smaller error the ranging can achieve. The performance gain stems from more accurate power level measurement of the LoS path. According to the statistics, on average Splicer-120MHz and Splicer-200MHz can outperform Splicer-20MHz by 22.7% and 55.5%, respectively. The performance gain of Splicer stems from both the CSI splicing as well as the CSI phase refinement (§4.1.4). In Figure 4.14, we also show the performance gain the phase refinement provides. From the result, we see that the CSI phase refinement can reduce the ranging error by 0.42m approximately.

Non-line-of-sight paths. In Figures 4.12 to 4.14, we have evaluated the LoS power accuracy

Figure 4.14: CDF of ranging errors with CSI splicing.

in the derived power delay profiles. In this experiment, we investigate the quality of the derived power delay profiles for all other NLoS paths. In practice, the absolute power level of a NLoS path is not directly useful in applications. Instead, the relative change of the power level of each NLoS path indicates the multipath channel dynamics, which has been used for activity or gesture recognitions [61, 84, 87]. To this end, we randomly select 10 locations in the laboratory. At each location, the transmission pair performs multiple rounds of CSI splicing. We evaluate the stability of the measured power levels for NLoS paths.

Figure 4.15 depicts the results. In Figure 4.15 (a), we examine the NLoS path power level stability using a single 20MHz WiFi channel. However, we observe that without the phase error compensation, the power variance is high, *e.g.*, around 16 dB, even the pair of transmitters are static at their locations. Our CSI error compensation designs can reduce the variance to be less than 10 dB on average for single WiFi bands. In Figure 4.15 (b), due to the CSI splicing, we can derive higher-resolution power delay profiles. Hence, the measured power level for each multipath component is aggregated from fewer non-distinguishable multipaths (§4.1.1), which should suffer from even less uncertainty. The result in Figure 4.15 (b) is consistent to

Figure 4.15: Measured power level variance for NLoS path components in the derived power delay profiles. (a) In 20MHz WiFi band; (b) In 200MHz WiFi band.

our analysis, which shows that the power level variance for NLoS paths in the derived power delay profiles using 200MHz spliced bandwidth is less than 5.2dB in our experiment.

Computational delay. We evaluate the computational delay of Splicer on commodity WiFi APs to investigate its frequency to generate new power delay profiles to upper-layer applications. We examine the absolute computation delays for Splicer to complete one round of CSI splicing with different aggregated bandwidths in Figure 4.16. Overall, the computation delay increases when more bands are spliced, since we need to compare more pairs of single bands for the phase error λ_o removal. From Figure 4.16, we find that the delays increase from 4.1ms to 7.9ms when the total bandwidth varies from 20MHz to 200MHz. We suppose the channel coherence time budget is 50ms. With the maximum bandwidth, one round of CSI splicing can be finished within 60ms. Therefore, Splicer can generate 16 new power delay profiles in one second, which fits the needs of most mobile applications [53, 70].

Figure 4.16: Computation delay of Splicer.

4.2.3 Case study: indoor localization

In 4.2.2, we have evaluated the accuracy of the derived power delay profiles by Splicer. With high resolution power delay profiles, the performance of a plethora of upper-layer applications, *e.g.*, localization, object tracking, gesture recognition, etc., can be significantly improved. In this subsection, we take localization as a vehicle to demonstrate this capability of Splicer.

4.2.3.1 Case study overview

We integrate Splicer into the recent single-AP location approach CUPID [70]. In CUPID, a mobile user can locate its location as follows. The mobile device of the user transmits a packet to an AP with known location. The AP extracts the CSI upon receiving this packet and derives the power delay profile. According to the power level of the LoS path, the AP can estimate this path length. On the other hand, the AP further apply the MUSIC algorithm on the CSI to compute its pseudo spectrum, which can approximate the signal arrival direction from the LoS path. According to the path length and the signal arrival direction, the AP can locate the mobile user and send the localization result back to the user.

To enhance CUPID, we use the spliced CSI as the system input and keep the rest of the CUPID design unchanged. To evaluate the performance, we install one AP in the laboratory with known location and deploy one AP on a robot. The robot is programmed to move along a predefined trajectory with a known speed. According to the time stamp contained in each packet, we can calculate the instant location of the robot when the mobile AP transmits this packet. To evaluate the end-to-end system performance, we enable the sampling scheduler in Splicer.

Figure 4.17: CDF of localization errors using the original CUPID and the enhanced CUPID by Splicer.

4.2.3.2 Results

Localization accuracy. In Figure 4.17 (a), we compare the performance of the original CUPID and the enhanced CUPID by Splicer, denoted as Splicer-CUPID. We vary the moving speeds of the robot in the experiment. Since the localization in the original CUPID depends on a single CSI measurement, the localization performance of CUPID is not impacted by the moving speed of the robot. For Splicer-CUPID, a higher moving speed leads to a shorter channel coherence time. As a consequence, fewer WiFi bands can be included in each round of CSI splicing

and the localization accuracy will decrease. From Figure 4.17, we find that in general, the localization accuracy of CUPID is not quite accurate, *e.g.*, around $8m$ for 80% of localizations, which is consistent to the performance reported in [70]. For Splicer-CUPID, with a normal moving speed of a person ($< 2m/s$), the accuracy can be dramatically improved, *e.g.*, the median localization error is $2.3m$ and $2.5m$ when the speed is $1m/s$ and $2m/s$, respectively. The localization error is less than $6.4m$ throughout the experiment. With an even higher moving speed, *e.g.*, $7m/s$, Splicer-CUPID still outperforms CUPID.

Figure 4.18: Bandwidth used for localization under different moving speeds. (Left) Spliced bandwidth; (right) Percentages of early terminations.

In [70], the authors propose an AP selection scheme to further improve the localization accuracy by harnessing a dense AP deployment. According to the experiment results in [70], the localization accuracy of Splicer-CUPID using a single AP achieves comparable performance to CUPID with 5 APs, which can significantly improve the usefulness of indoor localizations. In Figure 4.17, we also leverage such an improvement opportunity. The result shows that the gain from Splicer-CUPID is significant. Localization errors are reduced to $1.75m$ for 80% cases and the median error is $0.95m$ when 5 APs are used. Although more APs may improve the localization performance, if the ranging accuracy is not high at the first place, improvement with more APs is limited.

Impact of moving speeds. In Figure 4.18 (left y -axis), we investigate the total bandwidth that can be spliced using Splicer with respect to different moving speeds of transmitters. In general, a higher speed leads to a shorter channel coherence time. As a result, CSI traces are spliced from fewer WiFi channels and the power delay profile resolution is lower. From Figure 4.18, we see that Splicer can make full use of the 200MHz available WiFi frequency band, when the speed is smaller than $2m/s$. The small localization errors observed in this case in Figure 4.17 is compatible with such an observation. When we accelerate the moving speed from $3m/s$ to $6m/s$, the utilized bandwidth drops from 130MHz to 60MHz, which, however, is still wider than a single WiFi channel, *i.e.* 20MHz or 40MHz. Further more, the bandwidth will drop to 45.5MHz when the speed increases to $7m/s$, which is comparable to one single 40MHz channel. Nevertheless, Splicer-CUPID still improve the localization performance since Splicer compensates the CSI measurement errors.

Figure 4.19: Localization errors in different environments using the original CUPID and the enhanced CUPID by Splicer.

We further examine the percentages of early terminations (§4.1.7) occurred in our experiment, in Figure 4.18 (right y -axis). From the result, we see that the early termination strategy can discover 4.8% to 18.7% rapid channel varying within the channel coherence time budget.

Performance in different environments. In addition to the evaluation in the laboratory environment, we further conduct experiments in other representative environments for localization, including a corridor, a car park, and a lecture hall. Figure 4.19 plots the median localization error for Splicer-CUPID in comparison with the original CUPID with normal walking speed ($1m/s$). From the results, we see that the localization error of the original CUPID design can be largely reduced by Splicer-CUPID, *e.g.*, 70.9% in laboratory, 74.0% in corridor, 74.2% in car park, and 76% in the lecture hall. In summary, Splicer achieves general performance gains with different environments.

4.3 Discussion and limitation

Splicer needs to hop from one frequency to another frequency so that it will interrupt the normal communication between the Wi-Fi transceiver, which is a limitation of the current design. To avoid such an limitation, we can schedule the system to hopping when there is no traffic between the transceivers, or we can design a dedicated Wi-Fi device for localization. The total bandwidth can Splicer can combine decreases as the speed of the mobile device increases and it will decrease to single channel if the moving speed is larger than $7m/s$, which is another limitation of the design. However, for most of the cases, the speed of human won't pass $7m/s$, which makes Splicer still a efficient system for scenarios where the speed of the motion is low. If the human is in car, the motion will hinders Splicer from splicing the Wi-Fi channels.

Splicer use iFFT to derive the power delay profile. However, it has been shown in Tone-Track [99] that applying super-resolution algorithms like MUSIC can further increasing the resolution of the system, while iFFT cannot achieve that. Therefore, the performance of Splicer could be further enhanced if we apply super-resolution algorithms instead of using iFFT.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

CSI information is critical for both Wi-Fi communication and Wi-Fi based motion tracking and localization systems. In this report, we build a CSI extraction tool based on Atheros Wi-Fi NIC. We believe it is the first tool that can extract CSI of every subcarrier in a channel. The precision is also much higher since the CSI is represented using a 10-bit number. At last, this tool is portable to other platforms.

Based on our tool, we build UnPKT that can provide unequal packet bit protection for commercial Wi-Fi devices. We address the fine-grained packet bit BER estimation issue. We observe that packet bit BERs strongly relate to the probability that dense errors occur in the codeword bits and the density of the codeword bit errors. With this observation, we propose a BER estimator, which can estimate the bit-wise packet bit BERs. Our estimator is computationally efficient and takes CSI as the sole input. UnPKT protects the packet bits using different MAC-layer FEC redundancies based on the bit-wise BER estimation to augment wide-band 802.11 transmissions. By doing so, UnPKT prevent transmission failures and improves the throughput. We extensively evaluate the performance of UnPKT using both Atheros 9580 NICs and the GNU Radio platform and obtain a significant experimental performance improvement over state-of-the-art approaches.

We build another system Splicer to derive precise power delay profiles on commodity WiFi devices. The Splicer design leverages the CSIs measured from individual WiFi bands to obtain the CSI of an equivalent wider WiFi band after the CSI splicing, based on which Splicer can derive high-resolution power delay profiles. The major design challenge is that the CSIs collected from commodity NICs do not merely contain the channel information. They are mixed with rich hardware errors. We propose a series of techniques to address the challenge and also battle the stringent channel coherence time constraint. Our experiments on commodity WiFi NICs report high accuracy of the power delay profiles derived by Splicer. The CSI measurement errors stem from several signal-processing modules in the physical layer of standard 802.11 NICs, including the AGC, signal sampler, and packet detector, so Splicer is general for different WiFi chipsets and independent to the hardware. Due to the channel hopping by Splicer, the normal communication of other users to an AP may be interrupted, which is a limitation of the current design. We believe that such a limitation can be mitigated by employing less-active or dedicated APs for Splicer. We use the indoor localization application to showcase that Splicer can immediately benefit existing motion- or location-based systems.

5.2 Future Work

Based on our previous studies, we believe our future work may include the following areas.

Unequal protection in MIMO communication. Currently, UnPKT focuses on providing unequal protection in Wi-Fi system communicate using SISO configuration. Wi-Fi technology has been evolving in past decade. In 2009, 802.11n [4] was proposed. It introduces MIMO technology to harness the antenna and spatial diversity in Wi-Fi system. IEEE 802.11ac [5], which was released in 2012, adopts Multi-User MIMO to support multiple concurrent transmissions. MIMO and Multi-User MIMO significantly improve the theoretical transmission rate of Wi-Fi system. Nevertheless, the real throughput is still affected by the frequency se-

lective fading in an indoor environment. Unequal protection is still required in those advanced systems.

In our future work, we consider extending UnPKT to Wi-Fi systems that are configured to use MIMO even Multi-User MIMO techniques. To realize such an envision, numerous challenges must be addressed. In SISO configuration, one packet is transmitted over a single link whose channel quality can be measured using CSI. However, in MIMO communication, one packet may be delivered through multiple channels and the signal received from different channels are combined to recover the original packets. In such a way, the SNR of the received packet can be improved. The prediction of the packet bits, however, becomes challenging. In Multi-User MIMO system, the AP may deliver multiple packets to multiple users simultaneously, which makes the BER prediction even more complicated.

Passive object tracking using high resolution power delay profile. As we have discussed, power delay profile is useful for a lot of motion tracking or localization systems. In our current design, we only build a case study of single access point indoor localization using power delay profile. Passive localization is an interesting and promising research area. Power delay profile gives us the time of arrival information of all resolvable multipaths. If we can identify and isolate the multipath that reflects off our moving target from power delay profile, we can localize the target. Wi-Track [8] has developed a system with similar idea, which, however, requires a customized, dedicated hardware to transmit their ultra-wideband FMCW (Frequency-Modulated Carrier Wave) sounding signal. Implementing such an approach using commodity Wi-Fi devices is much more challenging.

Localization using multiple-dimension information. In our current work, we only make use of time of flight information to localize the target. One multipath has multiple other dimensions of information that can help localization, *e.g.*, the angle of arrival, the angle of departure and Doppler shift. We can calculate the distance using time of arrival information. We can also get the direction of the target through angle of arrival and angle of departure. The Doppler shifts

give us the motion-related information about the target. For example, we can distinguish one multipath reflecting off a mobile object from the one reflecting off a static object by checking their Doppler shifts. The signal with non-zero Doppler shift value must arise from the mobile object. Wi-Deo [42] and SpotFi [49] have explored ideas of jointly using time of arrival and angle of arrival information to localize the target. It is still unclear how we can jointly exploit all those dimensions together. The resolution is also a big challenge. As we know the time resolution and angle resolution determined by the bandwidth and antenna number respectively are the major factors that hinder the performance of Wi-Fi sensing applications.

Appendix A

Author's Publications

- (i) **Yaxiong Xie**, Jie Xiong, Mo Li, Kyle Jamieson, “xD-Track: Leveraging Information from Multiple Dimensions for Passive Wi-Fi Tracking ”, in *ACM HotWireless*, 2016.
- (ii) **Yaxiong Xie**, Zhengjiang Li, Mo Li, Kyle Jamieson, “Augmenting Wide-band 802.11 Transmissions via Unequal Packet Bit Protection ”, in *IEEE INFOCOM*, 2016.
- (iii) **Yaxiong Xie**, Zhenjiang Li, Mo Li, “Precise Power Delay Profiling with Commodity WiFi ”, in *ACM MobiCom*, 2015.
- (iv) Zhenjiang Li, **Yaxiong Xie**, Mo Li, Kyle Jamieson, “Recitation: Rehearsing Wireless Packet Reception in Software ”, in *ACM MobiCom*, 2015.

References

- [1] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band. *IEEE Std 802.11a-1999*, 1999.
- [2] IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band. *IEEE Std 802.11b-1999*, 2000.
- [3] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, 2003.
- [4] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended*

REFERENCES

- by *IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009*), 2009.
- [5] 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) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)*, 2013.
- [6] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015 2020 White Paper. <http://blog.privatewifi.com/will-wifi-replace-cable-and-fiber-networks/>.
- [7] Atheros CSI tool. <http://pdcc.ntu.edu.sg/wands/Atheros/>.
- [8] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3d tracking via body radio reflections. In *USENIX NSDI*, 2014.
- [9] A.J.Viterbi and J.K.Omura. *Principles of digital communication and coding*. 1979.
- [10] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *ACM MobiCom*, 2005.
- [11] K. Ali, A. X. Liu, W. Wang, and M. Shahzad. Keystroke recognition using wifi signals. In *ACM MobiCom*, 2015.
- [12] M. Arslan, K. Pelechrinis, I. Broustis, S. Singh, S. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Acorn: An auto-configuration framework for 802.11n wlans. *IEEE/ACM Transactions on Networking*, 21(3):896–909, 2013.
- [13] M. Y. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Auto-configuration of 802.11n wlans. In *ACM CoNEXT*, 2010.

REFERENCES

- [14] A. Azzalini. A class of distributions which includes the normal ones. *Scandinavian journal of statistics*, 1985.
- [15] A. Bhartia, Y.-C. Chen, S. Rallapalli, and L. Qiu. Harnessing frequency diversity in Wi-Fi networks. In *ACM MobiCom*, 2011.
- [16] J. Bicket. Bit-rate selection in wireless networks. In *Masters thesis, Massachusetts Inst. of Technology*, 2005.
- [17] J. Camp and E. Knightly. Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. In *MobiCom*, 2008.
- [18] P. Chaporkar, A. Proutiere, and B. Radunoviae. Rate adaptation games in wireless lans: Nash equilibrium and price of anarchy. In *IEEE INFOCOM*, 2010.
- [19] B. Chen, Z. Zhou, Y. Zhao, and H. Yu. Efficient error estimating coding: Feasibility and applications. In *ACM SIGCOMM*, 2010.
- [20] C.-M. Chen, C.-W. Lin, and Y.-C. Chen. Unequal error protection for video streaming over wireless LANs using content-aware packet retry limit. In *IEEE ICME*, 2006.
- [21] N. Chen and Z. Yan. Complexity analysis of reed-solomon decoding over $gf(2^m)$ without using syndromes. *EURASIP Journal on Wireless Communications and Networking*, 2008:16:1–16:11, 2008.
- [22] S. Choi, Y. Choi, and I. Lee. Ieee 802.11 mac-level fec scheme with retransmission combining. *IEEE Transactions on Wireless Communications*, 5(1):203–211, 2006.
- [23] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets. CSI-SF: Estimating wireless channel state using CSI sampling and fusion. In *IEEE INFOCOM*, 2012.
- [24] W. Dong, Y. Liu, Y. He, and T. Zhu. Measurement and analysis on the packet delivery performance in a large scale sensor network. In *IEEE INFOCOM*, 2013.

REFERENCES

- [25] W. Dong, J. Yu, and X. Liu. CARE: Corruption-aware retransmission with adaptive coding for the low-power wireless. In *IEEE ICNP*, 2015.
- [26] T. Felhauer, P. Baier, W. Konig, and W. Mohr. Optimum spread spectrum signals for wideband channel sounding. *Electronics Letters*, 1993.
- [27] A. Franchi and R. A. Harris. On the error burst properties of viterbi decoding. In *IEEE ICC*, 1993.
- [28] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *ACM MobiCom*, 2014.
- [29] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [30] D. Gong and Y. Yang. AP association in 802.11n WLANs with heterogeneous clients. In *IEEE INFOCOM*, 2012.
- [31] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SigComm*, 2010.
- [32] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM*, 2010.
- [33] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM*, 2010.
- [34] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. R. Miller. All bits are not equal-a study of IEEE 802.11 communication bit errors. In *IEEE INFOCOM*, 2009.
- [35] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. Miller. Maranello: Practical partial packet recovery for 802.11. In *USENIX NSDI*, 2010.
- [36] C. Han, K. Wu, Y. Wang, and L. Ni. WiFall: Device-free fall detection by wireless networks. In *IEEE INFOCOM*, 2014.

REFERENCES

- [37] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *ACM MobiCom*, 2001.
- [38] IEEE Standard for Information Technology—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications, Mar. 2012.
- [39] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, 2007.
- [40] S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. In *ACM MobiCom*, 2008.
- [41] V. Jimenez, M. Fernandez-Getino Garcia, F. Serrano, and A. Armada. Design and implementation of synchronization and agc for ofdm-based wlan receivers. *IEEE Transactions on Consumer Electronics*, 2004.
- [42] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti. Wideo: Fine-grained device-free motion tracing using rf backscatter. In *NSDI*, 2015.
- [43] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *ACM MobiSys*, 2008.
- [44] B. Kempke, P. Pannuto, and P. Dutta. Harmonia: Wideband spreading for accurate indoor rf localization. In *ACM HotWireless*, 2014.
- [45] M. Khalid, Y. Wang, I. Butun, H.-j. Kim, I.-h. Ra, and R. Sankar. Coherence time-based cooperative mac protocol 1 for wireless ad hoc networks. In *EURASIP Journal on Wireless Communications and Networking*, 2011.
- [46] M. Khan, A. Moinuddin, E. Khan, and M. Ghanbari. Optimized cross-layer unequal error protection for wireless video communication. In *IEEE TENCON*, 2013.
- [47] M. O. Khan, V. Dave, Y.-C. Chen, O. Jensen, L. Qiu, A. Bhartia, and S. Rallapalli. Model-driven energy-aware rate adaptation. In *ACM MobiHoc*, 2013.

REFERENCES

- [48] M. O. Khan, L. Qiu, A. Bhartia, and K. C.-J. Lin. Smart retransmission and rate adaptation in wifi. In *IEEE ICNP*, 2015.
- [49] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM*, 2015.
- [50] R. Kumar, S. Tati, F. de Mello, S. V. Krishnamurthy, and T. L. Porta. Network coding aware rate selection in multi-rate ieee 802.11. In *ACM ICNP*, 2010.
- [51] Z. Li, Y. Xie, M. Li, and K. Jamieson. Recitation: Rehearsing wireless packet reception in software. In *ACM MobiCom*, 2015.
- [52] K. C.-J. Lin, N. Kushman, and D. Katabi. ZipTx: Harnessing partial packets in 802.11 networks. In *ACM MobiCom*, 2008.
- [53] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. SAIL: Single access point-based indoor localization. In *ACM MobiSys*, 2014.
- [54] M. Mirza, P. Barford, X. Zhu, S. Banerjee, and M. Blodgett. Fingerprinting 802.11 rate adaption algorithms. In *IEEE INFOCOM*, 2011.
- [55] A. Molina, P. Fannin, and J. Timoney. Generation of optimum excitation waveforms for mobile radio channel sounding. *IEEE Transactions on Vehicular Technology*, 1995.
- [56] D. Molkdar and P. Matthews. Measurements and characterization of the UHF mobile radio channel. part 1: Measurements over the band 853-885 MHz. *Electronic and Radio Engineers, Journal of the Institution of*, 1988.
- [57] S. Nanda and K. REge. Frame error rates for convolutional codes on fading channels and the concept of effective eb/n0. *IEEE Transactions on Vehicular Technology*, 47(4):1245–1250, 1998.
- [58] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan. Wifi-reports: Improving wireless network selection with collaboration. In *ACM MobiSys*, 2009.

REFERENCES

- [59] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu. Mimo rate adaptation in 802.11n wireless networks. In *ACM MobiCom*, 2010.
- [60] J. G. Proakis. *Digital Communications*. 2007.
- [61] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *ACM MobiCom*, 2013.
- [62] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini. Frequency-aware rate adaptation and MAC protocols. In *ACM MobiCom*, 2009.
- [63] H. S. Rahul, S. Kumar, and D. Katabi. Jmb: Scaling wireless capacity with user demands. In *ACM SIGCOMM*, 2014.
- [64] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser. Symphony: Synchronous two-phase rate and power control in 802.11 wlans. In *ACM MobiSys*, 2008.
- [65] T. S. Rappaport et al. *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996.
- [66] N. K. M. S. Sen and S. Banerjee. Scalable WiFi media delivery through adaptive broadcasts. In *NSDI*, 2010.
- [67] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *ACM MobiCom*, 2002.
- [68] S. Sen, R. R. Choudhury, and S. Nelakuditi. Spinloc: Spin once to know your location. In *ACM HotMobile*, 2012.
- [69] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. Design and implementation of an "approximate" communication system for wireless media applications. In *ACM SIGCOMM*, 2010.
- [70] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *ACM MobiSys*, 2013.

REFERENCES

- [71] S. Sen, N. K. Madabhushi, and S. Banerjee. Scalable wifi media delivery through adaptive broadcasts. In *NSDI*, 2010.
- [72] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: Spot localization using PHY layer information. In *ACM MobiSys*, 2012.
- [73] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Accurate: Constellation based rate estimation in wireless networks. In *NSDI*, 2010.
- [74] L. Shanguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu. Relative localization of rfid tags using spatial-temporal phase profiling. In *USENIX NSDI*, 2015.
- [75] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen. Rate adaptation for 802.11 multiuser mimo networks. In *ACM MobiCom*, 2012.
- [76] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen. Rate adaptation for 802.11 multiuser mimo networks. In *ACM MobiCom*, 2012.
- [77] M. Speth, S. Fechtel, G. Fock, and H. Meyr. Optimum receiver design for wireless broad-band systems using OFDM. i. *IEEE Transactions on Communications*, 1999.
- [78] J. K. Tan. An adaptive orthogonal frequency division multiplexing baseband modem for wideband wireless channels. Master’s thesis, Massachusetts Institute of Technology, 2006.
- [79] X. Tie, A. Seetharam, A. Venkataramani, D. Ganesan, and D. L. Goeckel. Anticipatory wireless bitrate control for blocks. In *ACM CoNEXT*, 2011.
- [80] T.K.Moon. *Error Correction Coding*. 2005.
- [81] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM*, 2009.
- [82] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SigComm*, 2009.

REFERENCES

- [83] P.-J. Wan, B. Xu, O. Frieder, S. Ji, B. Wang, and X. Xu. Capacity maximization in wireless MIMO networks with receiver-side interference suppression. In *ACM MobiHoc*, 2014.
- [84] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni. We can hear you with Wi-Fi! In *ACM MobiCom*, 2014.
- [85] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie. Human respiration detection with commodity wifi devices: Do user location and body orientation matter? In *ACM UbiComp*, 2016.
- [86] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu. Understanding and modeling of wifi signal based human activity recognition. In *ACM MobiCom*, 2015.
- [87] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu. E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures. In *ACM MobiCom*, 2014.
- [88] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, 2006.
- [89] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi. Beyond the bits: Cooperative packet recovery using physical layer information. In *ACM MobiCom*, 2007.
- [90] C. Wu, Z. Yang, Z. Zhou, K. Qian, Y. Liu, and M. Liu. Phaseu: Real-time LOS identification with wifi. In *IEEE INFOCOM*, 2014.
- [91] D. Wu, D. Zhang, C. Xu, Y. Wang, and H. Wang. Widir: Walking direction estimation using wireless signals. In *ACM UbiComp*, 2016.
- [92] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. Ni. FILA: Fine-grained indoor localization. In *IEEE INFOCOM*, 2012.

REFERENCES

- [93] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang. Electronic frog eye: Counting crowd using WiFi. In *IEEE INFOCOM*, 2014.
- [94] J. Xiao, K. Wu, Y. Yi, and L. Ni. FIFS: Fine-grained indoor fingerprinting system. In *IEEE ICCCN*, 2012.
- [95] J. Xie, W. Hu, and Z. Zhang. Revisiting partial packet recovery in 802.11 wireless LANs. In *ACM MobiSys*, 2011.
- [96] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wifi. In *ACM MobiCom*, 2015.
- [97] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *USENIX NSDI*, 2013.
- [98] J. Xiong, K. Jamieson, and K. Sundaresan. Synchronicity: Pushing the envelope of fine-grained localization with distributed mimo. In *ACM HotWireless*, 2014.
- [99] J. Xiong, K. Sundaresan, and K. Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *ACM MobiCom*, 2015.
- [100] J. Xiong, K. Sundaresan, and K. Jamieson. Tonetrack: Overcoming bandwidth constraints for indoor wireless localization. In *ACM MobiCom*, 2015.
- [101] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices. In *ACM MobiCom*, 2014.
- [102] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu. Season: Shelving interference and joint identification in large-scale rfid systems. In *IEEE INFOCOM*, 2011.
- [103] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical snr-guided rate adaptation. In *INFOCOM*, 2008.
- [104] X. Zhou, J. Zhao, and G. Yang. Correlation based rate adaptation via insights from incomplete observations in 802.11 networks. In *IEEE ICC*, 2007.

REFERENCES

- [105] Z. Zhou, Z. Yang, C. Wu, L. Shangguan, and Y. Liu. Omnidirectional coverage for device-free passive human detection. *IEEE Transactions on Parallel and Distributed Systems*, 2014.