

**DESIGNING WIRELESS NETWORKS FOR  
DELAY-SENSITIVE INTERNET OF THINGS**

by

**Haoyang Lu**

B.E., Hangzhou Dianzi University, 2010

M.E., University of Chinese Academy of Sciences, 2013

Submitted to the Graduate Faculty of  
the Swanson School of Engineering in partial fulfillment  
of the requirements for the degree of

**Doctor of Philosophy**

University of Pittsburgh

2018

UNIVERSITY OF PITTSBURGH  
SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Haoyang Lu

It was defended on

September 24, 2018

and approved by

Wei Gao, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Amro El-Jaroudi, Ph.D., Associate Professor, Department of Electrical and Computer  
Engineering

Zhi-Hong Mao, Ph.D., Professor, Department of Electrical and Computer Engineering

Jingtong Hu, Ph.D., Assistant Professor, Department of Electrical and Computer  
Engineering

David Tipper, Ph.D., Professor, School of Computing and Information Sciences

Dissertation Director: Wei Gao, Ph.D., Associate Professor, Department of Electrical and  
Computer Engineering

# DESIGNING WIRELESS NETWORKS FOR DELAY-SENSITIVE INTERNET OF THINGS

Haoyang Lu, PhD

University of Pittsburgh, 2018

Internet of Things (IoT) applications have stringent requirements on the wireless network delay, but have to share and compete for the limited bandwidth with other wireless traffic. Traditional schemes adopt various QoS-aware traffic scheduling techniques, but fail when the amount of network traffic further increases. In addition, IoT technology using 802.11 with CSMA with collision avoidance (CSMA/CA) mechanism enables the coexistence of multiple wireless links but avoids concurrent transmissions, yielding severe channel access delay on the delay-sensitive traffic when the channel is busy.

To address the aforementioned limitations, we present two novel designs of wireless side channel, which operate concurrently with the existing wireless network channel without occupying extra spectrum, but is dedicated to real-time traffic. Our key insight to realizing such a side channel is to exploit the excessive SNR margin in the wireless network by encoding data as patterned interference. First, we design such patterned interference in the form of energy erasure over specific subcarriers in OFDM systems. Delay-sensitive messages can be delivered simultaneously along with other traffic from the same transmitter, which reduces the network queuing delay. Furthermore, we propose *EasyPass*, another side channel design that encodes data in the same OFDM scheme as being used by the main channel, but using weaker power and narrower frequency bands. By adapting the side channel's transmit power under the main channel's SNR margin, the simultaneous main channel transmission would suffer little degradation. *EasyPass* reduces the channel access delay by providing extra transmission opportunities when the channel is occupied by other links. Last, we present a

novel modulation design that transforms the choices of link rate adaptation from discrete to continuous. With minimum extra overhead, it improves the network throughput and therefore reduces the network delay.

## TABLE OF CONTENTS

<b>1.0 INTRODUCTION</b> . . . . .	1
1.1 Background . . . . .	1
1.2 Motivation . . . . .	3
1.2.1 SNR margin . . . . .	3
1.2.2 SNR Gap: Inefficient Utilization of Wireless Channel Capacity . . . . .	4
1.3 WiFi Primers . . . . .	6
1.3.1 OFDM . . . . .	6
1.3.2 Channel coding and Viterbi decoder . . . . .	7
1.3.3 Channel Equalization . . . . .	8
1.4 Contributions . . . . .	8
1.4.1 A High-throughput Side Channel Design to combat queuing delay . . . . .	8
1.4.2 A Side Channel Design to Combat Channel Access Delay . . . . .	9
1.4.3 Continuous Wireless Link Rates to improve the network throughput . . . . .	9
1.5 Organization . . . . .	10
<b>2.0 SUPPORTING REAL-TIME WIRELESS TRAFFIC THROUGH A HIGH-THROUGHPUT SIDE CHANNEL</b> . . . . .	11
2.1 Introduction . . . . .	11
2.2 Related Work . . . . .	14
2.3 Overview . . . . .	16
2.3.1 Motivation . . . . .	16
2.3.2 Big Picture . . . . .	18
2.4 Side Channel Design . . . . .	19

2.4.1	Overall System Design . . . . .	19
2.4.2	Improvement of Data Throughput . . . . .	20
2.4.3	Probabilistic Detection of Energy Erasure . . . . .	21
2.5	Connection Establishment and Transmission Control . . . . .	23
2.5.1	Connection Establishment . . . . .	24
2.5.2	Side Channel Addressing . . . . .	24
2.6	Performance Evaluation over Software-Defined Radios . . . . .	26
2.6.1	System Implementation and Experiment Setup . . . . .	26
2.6.2	Data Transmission Delay . . . . .	27
2.6.3	Detection Rate of Patterned Interference . . . . .	29
2.6.4	Data Throughput of the Side Channel . . . . .	31
2.6.5	Impacts on the Main Channel . . . . .	32
2.7	Hardware Implementation . . . . .	32
2.7.1	Implementation Overview . . . . .	33
2.7.2	Delay Reduction . . . . .	35
2.7.3	Channel Symbol Synchronization . . . . .	36
2.8	Performance Evaluation over Wireless Hardware . . . . .	37
2.8.1	Experiment Setup . . . . .	37
2.8.2	Side Channel Performance . . . . .	37
2.8.3	Computation Delay and Overhead . . . . .	38
2.8.4	Impact of Channel Interference . . . . .	39
2.9	Chapter Summary . . . . .	41
<b>3.0</b>	<b>EASYPASS: COMBATING IOT DELAY WITH MULTIPLE ACCESS</b>	
	<b>WIRELESS SIDE CHANNELS . . . . .</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	EasyPass Design . . . . .	46
3.2.1	Two-step Equalization . . . . .	47
3.2.2	Multiple Accessibility of Side Channel . . . . .	48
3.2.3	Minimizing Symbol Errors . . . . .	49
3.3	Transmit Power Control . . . . .	51

3.3.1	Determine $P_{\max}^S$ . . . . .	52
3.3.1.1	Estimating $PL_{S \rightarrow M}$ . . . . .	52
3.3.1.2	Estimating $N_M$ . . . . .	53
3.3.2	Determine $P_{\min}^S$ . . . . .	54
3.4	Distributed Subcarrier Allocation . . . . .	54
3.4.1	Subcarrier Occupancy Sensing . . . . .	55
3.4.2	Subcarrier Allocation . . . . .	56
3.4.3	Addressing Conflicts . . . . .	57
3.5	Implementation . . . . .	58
3.5.1	PHY Implementation . . . . .	58
3.5.2	MAC-Layer Implementation . . . . .	59
3.6	Performance Evaluation . . . . .	60
3.6.1	Experiment Setup . . . . .	60
3.6.2	Network Delay . . . . .	62
3.6.3	Side channel throughput . . . . .	63
3.6.4	Computational Overhead . . . . .	65
3.7	Real-World Experimentation . . . . .	66
3.7.1	Subcarrier Allocation . . . . .	66
3.7.2	Highly Volatile Conditions . . . . .	68
3.8	Related Work . . . . .	69
3.9	Chapter Summary . . . . .	71
<b>4.0</b>	<b>CONTINUOUS WIRELESS LINK RATES FOR IOT</b> . . . . .	<b>72</b>
4.1	Introduction . . . . .	72
4.2	Limitation of the Existing Approaches . . . . .	75
4.2.1	Variable-Length Code . . . . .	77
4.3	Overview . . . . .	78
4.4	VLC-based Modulation . . . . .	80
4.4.1	Constitution of Continuous Link Rates . . . . .	80
4.4.2	Constellation Diagram Design . . . . .	83
4.4.3	Codeword Construction and Mapping . . . . .	84

4.5	VLC-based Demodulation . . . . .	86
4.5.1	Recognition of the Received Signal . . . . .	88
4.5.2	Resistance against Channel Noise . . . . .	89
4.6	Implementation . . . . .	91
4.6.1	PHY-Layer Implementation . . . . .	91
4.6.2	MAC-Layer Implementation . . . . .	92
4.7	Performance Evaluation . . . . .	93
4.7.1	Experiment Setup . . . . .	94
4.7.2	Packet Delivery Ratio . . . . .	95
4.7.3	Overhead . . . . .	96
4.8	Real-World Experimentation . . . . .	97
4.8.1	Experiment Setup . . . . .	97
4.8.2	Throughput Improvement . . . . .	99
4.8.3	Stability of Channel Throughput . . . . .	100
4.8.4	Impact of Channel Interference and Fading . . . . .	101
4.9	Related Work . . . . .	102
4.10	Discussion . . . . .	104
4.11	Chapter Summary . . . . .	105
<b>5.0</b>	<b>CONCLUSION . . . . .</b>	<b>106</b>
	<b>BIBLIOGRAPHY . . . . .</b>	<b>108</b>

## LIST OF TABLES

2.1	Side channel detection rate with a 9 Mbps data rate . . . . .	30
2.2	Side channel throughput (Mbps) with one subcarrier being erased in each OFDM symbol . . . . .	31
2.3	SNR requirement with two subcarriers being erased in each OFDM symbol . . . . .	32
2.4	Channel Throughput in Mbps . . . . .	40
3.1	Path loss and $N_M$ estimation . . . . .	62
3.2	Path loss of the side channel links . . . . .	62
3.3	Side channel throughput with different amounts of main channel SNR margin being exploited . . . . .	64
3.4	Tx computation overhead . . . . .	65
3.5	Rx computation overhead ( $\mu s$ ) . . . . .	66
3.6	Using different frequencies of PC requests . . . . .	70
4.1	Continuous link rates with two VLC codeword lengths in the Huffman code . . . . .	81
4.2	Tx communication overhead ( $us/bit/MHz$ ) . . . . .	96
4.3	Rx computation overhead ( $us$ ) . . . . .	97

## LIST OF FIGURES

1.1	Distribution of the channel SNR under different data rates . . . . .	3
1.2	Statistical distribution of SNR margin . . . . .	4
1.3	Inefficient utilization of WiFi channel capacity . . . . .	5
1.4	Throughput in practical wireless networks . . . . .	6
1.5	OFDM scheme . . . . .	7
2.1	Channel BER under different SNR conditions . . . . .	16
2.2	Encoding data as patterned interference into an OFDM symbol . . . . .	17
2.3	Side channel design over WiFi networks . . . . .	19
2.4	Constellation diagram of 16-QAM (3/4) under different channel SNR conditions	21
2.5	Design and detection of side channel preamble . . . . .	25
2.6	Data transmission delay . . . . .	27
2.7	Evaluation with VoIP traffic . . . . .	28
2.8	Detection rate of patterned interference . . . . .	30
2.9	Constellation diagrams under different SNR (QPSK 3/4) . . . . .	31
2.10	Main channel BER increase due to the existence of the side channel . . . . .	33
2.11	WARP3 Platform . . . . .	34
2.12	Side Channel Implementation over WARP . . . . .	34
2.13	Details of Tx design . . . . .	35
2.14	Side Channel BER in the 2.4GHz channel . . . . .	38
2.15	Main Channel Throughput . . . . .	39
2.16	Nodes Placement . . . . .	40
2.17	Side Channel BER with wireless interference . . . . .	41

3.1	Asynchronous signal extraction in EasyPass . . . . .	47
3.2	Multiple accessibility for side channel . . . . .	49
3.3	Side channel signals after the 2nd equalization . . . . .	50
3.4	Error of symbol alignment under 2.5 dB main channel SNR . . . . .	50
3.5	Power bounds of side channel . . . . .	51
3.6	Adaptive transmit power control . . . . .	52
3.7	FDR vs. SNR . . . . .	53
3.8	Subcarrier occupancy sensing . . . . .	55
3.9	Subcarrier reallocation when User 4 exits . . . . .	56
3.10	Flotations of subcarrier allocation . . . . .	57
3.11	PHY implementation of <i>EasyPass</i> . . . . .	58
3.12	Experiment scenario . . . . .	60
3.13	Wireless transmission delay . . . . .	60
3.14	Side channel throughput . . . . .	61
3.15	Main channel throughput loss . . . . .	63
3.16	Number of subcarriers for side channel transmission using different timeout ( $T_o$ ) values . . . . .	67
3.17	Parameters of subcarrier allocation . . . . .	68
3.18	Varying channel conditions and link rates of both the main and side channel (Node 2) . . . . .	69
4.1	Overhead of rateless codes . . . . .	77
4.2	VLC-based modulation . . . . .	78
4.3	An example of VLC-based modulation . . . . .	82
4.4	Design choices for 6-QAM . . . . .	85
4.5	Conveying information of VLC codeword lengths . . . . .	87
4.6	Recognition of the received signal . . . . .	89
4.7	PHY-layer vMod implementation . . . . .	90
4.8	Modified frame format . . . . .	92
4.9	Packet delivery ratio (PDR) under different channel conditions . . . . .	94
4.10	Computational overhead . . . . .	95

4.11 vMod's throughput improvement over existing WiFi with a single narrowband link . . . . .	98
4.12 Comparison of achieved wireless throughput with existing WiFi and rateless codes over a single narrowband link . . . . .	98
4.13 Stability of channel throughput over dynamic channel conditions . . . . .	100
4.14 Throughput in an interfered channel . . . . .	101
4.15 Throughput over fading channel . . . . .	103

## 1.0 INTRODUCTION

### 1.1 BACKGROUND

The soaring popularity of IoT solutions has been profoundly transforming every aspect of modern industries such as healthcare, smart home, transportation and industrial automation. Many IoT applications desire minimum delay of data transmission, so that they can promptly react to environmental events. For example, smart vehicles need to timely transmit images about their surroundings to the cloud for real-time processing, so as to recognize and avoid potential risks [54, 65].

In practice, such strict requirement may not be met due to the following reasons:

- **Excessive queuing delay.** The real-time wireless traffic of these applications may be seriously delayed when competing with other data traffic generated from the *same* transmitter. Traditional designs of wireless networks eliminate such network congestion via Quality of Service (QoS)-aware traffic scheduling [18] or flow control [19], but fail when the amount of network traffic further increases.
- **Channel access delay.** In CSMA-based WiFi network, after sensing a busy channel, the node defers its transmission until the end of the current transmission, and re-attempts to transmit after a random backoff delay, resulting in long channel access delay. In other words, the channel access delay is incurred due to the competition for the channel among *different* transmitters. Traditional wisdom reduces such delay by optimizing the backoff algorithms [15, 11, 29]. However, since CSMA/CA only allows transmission from a single user at each time, the channel access delay is lower-bounded by the duration of the ongoing transmission and can not be further reduced.

- **Throughput degradation.** Sufficient wireless throughput is another critical factor to timely transfer bulky data (e.g., high-resolution photos and video clips) in IoT. In order to maximize the throughput over dynamic fluctuations of wireless channel quality, current wireless systems adapt the link rate to the instantaneous channel condition, but fail to fully utilize the channel capacity due to the discrete choices of available link rates and the gap between these rates. Rateless codes improve throughput at the cost of extra communication and computation overhead, which prevents them from being practically applied in resource-constrained IoT applications.

One straightforward solution to alleviate the network delay is to allocate additional wireless spectrum that is exclusively used for real-time wireless traffic. For example, a dedicated spectrum is designated as the control plane in cellular networks. However, such exploitation of additional spectrum is costly due to the severe scarcity of wireless spectrum resources nowadays.

Instead, recent studies have demonstrated that wireless side channel designs could operate concurrently with the existing in-band wireless channel <sup>1</sup> without occupying any additional spectrum. By creating extra transmission opportunities under busy channel, side channel designs could become a viable solution to removing the fundamental limitation on supporting low latency wireless traffic. When WiFi transmitter transmits other traffic, or the wireless channel is sensed to be occupied, instead of deferring from transmission, wireless node would transmit delay-sensitive traffic through the side channel, which eliminates the excessive queuing delay and channel access delay, respectively. The key insight of a side channel design is that rate adaptation algorithms operate in a conservative fashion, which select lower data rates than the one that is supported by the instantaneous channel SNR so as to avoid costly decoding failure and retransmissions. Therefore, by controlling the side channel interference within the scope of the main channel's SNR margin, the existence of side channel brings little degradation on the main channel and makes the most of the channel capacity.

---

<sup>1</sup>The existing channel being used in current wireless networks is denoted as the "main channel" throughout the rest of this paper.

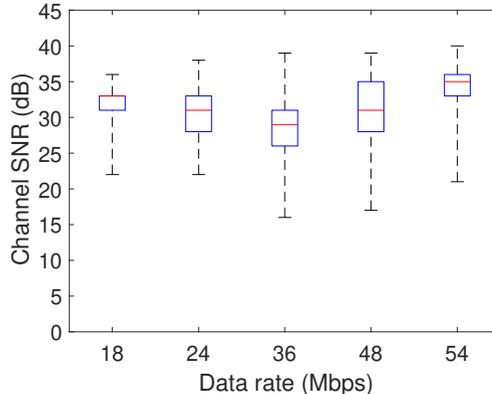


Figure 1.1: Distribution of the channel SNR under different data rates. Lowest three data rates 6, 9 and 12 Mbps are not observed. (—: Median; —: Min/Max; □: 25 to 75 Percentage)

## 1.2 MOTIVATION

### 1.2.1 SNR margin

Our major motivation for the side channel design is the existence of the wireless channel SNR margin in the wireless channel, i.e., the difference between the instant channel SNR and the minimum channel SNR that is required to support the data rate being used. SNR margin is intrinsic to the rate adaptation approaches due to their effort in avoiding costly retransmissions.

To evaluate the amount of SNR margin in practical WiFi networks, we conduct the following experiment using COTS devices. In specific, we connect a laptop (MacBook Air 2014) to the router (Netgear N300 WiFi C3000 Router) and stream video from HBO GO. We configure the router to work only at IEEE802.11g standard in Channel 11 (2462 MHz), which supports 8 data rates<sup>2</sup> ranging from 6 Mbps to 54 Mbps. The WiFi frames received by laptop are captured by network sniffer WireShark. We extract the received power (RSS)

<sup>2</sup>Different data rates are adopted by using different modulation methods and code rates. For example, 36 Mbps is realized by 16-QAM with the code rate 3/4.

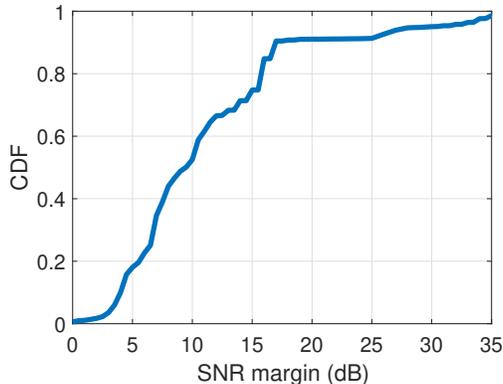


Figure 1.2: Statistical distribution of SNR margin

and the noise of the frames that are successfully decoded and calculate the SNR of each frame using their difference in dB. Figure 1.1 shows the SNR distribution with respect to each data rate. Based on [80]<sup>3</sup>, 12 dB is the minimum required SNR for supporting 24 Mbps, while the minimum SNR value observed is 22 dB, yielding 10 dB SNR margin. The CDF of the SNR margin is shown in Figure 1.2, which shows that more than 95% of the frames operate with  $> 5$ dB SNR margin, and the SNR margin can be up to 25 dB.

The existence of such SNR margin, therefore, validates our proposed side channel design. The key challenge, however, is how to appropriately limit the effect of the side channel under main channel’s SNR margin.

### 1.2.2 SNR Gap: Inefficient Utilization of Wireless Channel Capacity

Discrete choices of wireless link rates and the gap between these rates result in low efficiency of existing wireless networks in utilizing the channel capacity. To measure such gap, we conduct experiments for both narrowband and wideband WiFi channels over USRP N210 SDR boards. In these experiments, we emulate different levels of wireless channel SNR, and measure the receiver’s channel status in realtime to compute the corresponding Bit Error Rate (BER). The results in Figure 1.3 show the relationship between channel SNR and BER

---

<sup>3</sup>Similar results are reported in [98, 21].

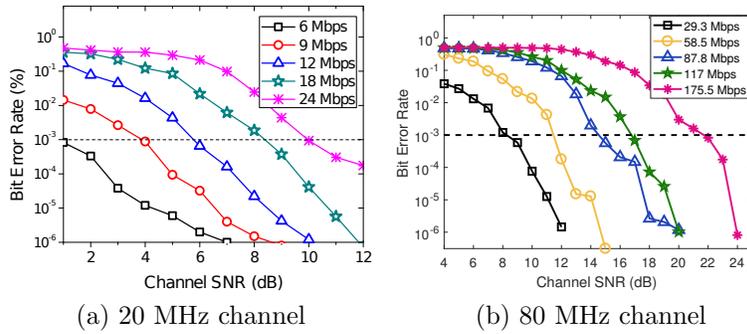


Figure 1.3: Inefficient utilization of WiFi channel capacity

when different link rates are used. When the narrowband 20 MHz channel is used, Figure 1.3(a) shows that 4 dB and 6 dB are the minimum SNRs to support 9 Mbps and 12 Mbps, respectively, in order to reach a BER less than 0.1%. The gap between these two link rates is hence 2 dB, corresponding to 3 Mbps of channel capacity being under-utilized. Similarly, when the link rate increases to 18 Mbps, a gap larger than 3 dB could be observed.

Recent technical advances in wireless networks are incapable of eliminating such gap. For example in Figure 1.3(b), the latest 802.11ac WiFi standard expands the channel bandwidth to 80 MHz through channel bonding, but also increases the SNR gap between consecutive link rates to be larger than 5 dB. These results highlight that the channel capacity could be better utilized by providing continuous choices of link rates to the gap between existing rates.

Furthermore, new generations of wireless networks, while improving the highest link rate under good channel conditions, make limited contribution to the achievable throughput in practical wireless networks with dynamic channel conditions. To verify this limitation, our experiment measures the WiFi signal strength and the corresponding link rates that are perceived by a Samsung Galaxy S4 smartphone at different locations. For a fair comparison, the channel bandwidth is limited to 20MHz and the signal strength over the 5 GHz band is converted to the equivalent number in the 2.4GHz band. The results in Figure 1.4 show that the link rates provided by different WiFi standards only exhibit noticeable differences when

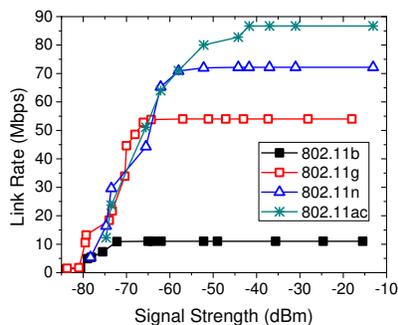


Figure 1.4: Throughput in practical wireless networks

the signal strength is higher than -60 dBm, which may not be achieved in many large-scale scenarios of public wireless networks. Improving the wireless throughput in these scenarios, instead, depends on continuous link rates that completely eliminate the gap during link rate adaptation.

## 1.3 WIFI PRIMERS

### 1.3.1 OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation technique and is widely used in existing WiFi and cellular networks because of its high efficiency of spectrum utilization and resistance to multipath channel fading and inter-carrier interference. As shown in Figure 1.5, data in OFDM is transmitted concurrently in closely spaced subcarriers, each of which corresponds to a frequency band that is orthogonal to others. Specifically, each 20 MHz WiFi channel comprises of 64 OFDM subcarriers that are spaced 312.5 kHz apart. In an OFDM symbol in the IEEE 802.11a standard, 12 subcarriers (including DC) are null as the guard band, 4 subcarriers are occupied by pre-known pilots for frequency offset compensation, and 48 subcarriers are used for data transmission. The duration of each OFDM symbol is  $4 \mu s$ .

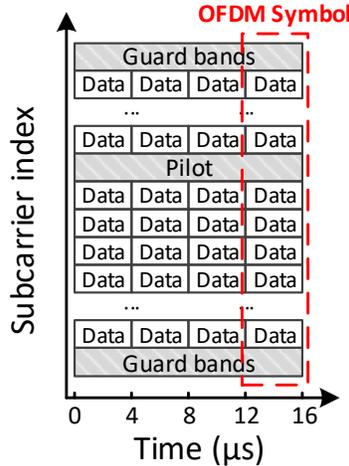


Figure 1.5: OFDM scheme

As a result, one OFDM packet could be viewed as a two-dimensional time-frequency grid, as shown in Figure 1.5. All data subcarriers within one packet use the same modulation symbol mapper (e.g., BPSK, QPSK, 16-QAM or 64-QAM) and the same code rate (e.g., 1/2, 2/3 or 3/4), according to the data rate being adopted. Different settings of the modulation symbol mapper approach and code rate yield different data rates and anti-interference capability. For example, BPSK with 1/2 code rate yields the lowest 6 Mbps data rate but is the most resistant to noise, and is hence adopted in poor SNR scenarios. On the contrary, 64-QAM with 3/4 code rate provides the maximum 54 Mbps throughput by using a denser constellation diagram, and is applied in good SNR scenarios.

### 1.3.2 Channel coding and Viterbi decoder

WiFi standards use Forward Error Correction (FEC) or LDPC code to enhance the immunity of systematic throughput against channel noise and environmental uncertainties. Lower code rate brings more redundancy and hence is more robust against data distortion. WiFi receivers

use Viterbi decoder to detect the errors introduced by noises, and further recovers the data from deteriorated signals. As a maximum likelihood decoding, Viterbi decoder has been widely applied in modern communication systems.

### 1.3.3 Channel Equalization

Channel equalization algorithms aid signal recovery through channel estimation and compensation. WiFi uses Long Training Sequence (LTS) in the frame preamble as the reference signal, which consists of two OFDM symbols spanning the entire subcarriers. The channel estimation is estimated as the distortion of the LTS in the frequency domain. Pilot-based approach is also applied in WiFi to track and compensate the fast fading channel during the span of transmission. Four subcarriers ( $\pm 21, \pm 7$ ) are reserved specifically for the insertion of pilots.

## 1.4 CONTRIBUTIONS

### 1.4.1 A High-throughput Side Channel Design to combat queuing delay

We present a novel design of high-throughput wireless side channel, which efficiently supports real-time wireless traffic without consuming any additional wireless spectrum resource. The experiment results verify the effectiveness of the side channel in reducing the data transmission latency and providing a data throughput higher than 1 Mbps, with minimum impact on the performance of the main wireless channel. I make the following contributions.

- We designed a side channel based on the OFDM technique in WiFi PHY layer, by employing subcarrier erasure to encode side channel information. The side channel throughput can reach 1.25 Mbps and 2.5 Mbps when one or two subcarriers in each symbol are erased.

- We proposed a probabilistic approach to achieve a high side channel detection rate. Compared with the energy-based approach which decodes the weakest subcarrier as the erased subcarrier, the proposed detection algorithm is resistant to multipath fading and ambient noise, and could increase the detection accuracy by 10%.

#### 1.4.2 A Side Channel Design to Combat Channel Access Delay

We present *EasyPass*, a practical side channel design that efficiently supports delay-sensitive traffic from multiple IoT links. *EasyPass* encodes data in the same OFDM scheme as being used by the main channel, but using weaker power and narrower frequency bands. By adapting the side channels transmit power under the main channel’s SNR margin, the simultaneous main channel transmission would suffer little degradation even without any MAC/PHY modifications. Furthermore, without the aid of frame synchronization, *EasyPass* enables multi-access by designating non-overlapping subbands to multiple simultaneous side channel links.

#### 1.4.3 Continuous Wireless Link Rates to improve the network throughput

We present a vMod, a lightweight solution towards maximum wireless throughput for IoT, by transforming the choices for link rate adaptation from discrete to continuous. We design constellation diagrams that map constellation points with the variable-length code (VLC), hence conveying a fractional number of data bits in each symbol. We implemented and evaluated our design over practical SDR platforms, and show that vMod greatly improves the wireless throughput with negligible overhead. The experiment results show that vMod scales well with the dynamic channel conditions and improves the WiFi throughput by 30% over a single narrowband link, but consumes up to 95% less computation and communication overhead compared to existing rateless codes.

## 1.5 ORGANIZATION

The rest of this dissertation is organized as follows. Chapter 2 introduces a high-throughput design that reduces the queuing delay. Chapter 3 presents *EasyPass*, a practical design to reduce the channel access delay. Chapter 4 presents *vMod*, a rateless modulation design to support continuous wireless link rate. Finally, we conclude this dissertation in Chapter 5.

## 2.0 SUPPORTING REAL-TIME WIRELESS TRAFFIC THROUGH A HIGH-THROUGHPUT SIDE CHANNEL

### 2.1 INTRODUCTION

Mobile computing has been an indispensable part of every aspect of modern life, by enabling highly cognitive and interactive mobile applications over heterogeneous types of mobile devices. Representative examples of these mobile applications include mobile cloud computing [61], resource sharing among mobile systems [3], wearable computing [39], and multi-party mobile gaming [60], which fundamentally transform the way people access information and interact with each other. Performance of these applications depends on the transmission latency of wireless link connecting mobile devices with each other and the remote cloud, so as to support real-time wireless traffic that is vital to prompt application response.

In practice, the real-time wireless traffic of these applications may be seriously delayed when competing with other data traffic being transmitted concurrently over the same wireless channel. Traditional designs of wireless networks eliminate such network congestion via Quality of Service (QoS)-aware traffic scheduling [18] or flow control [19], but fail when the amount of network traffic further increases. Another option to alleviate such network congestion is to allocate additional wireless spectrum that is exclusively used for real-time wireless traffic. For example, a dedicated spectrum is designated as the control plane in cellular networks [68]. However, such exploitation of additional spectrum is infeasible due to the severe scarcity of wireless spectrum resources nowadays.

Instead, another viable solution to removing this fundamental limitation on supporting real-time wireless traffic is to explore *a wireless communication side channel*, which operates concurrently with the existing wireless channel<sup>1</sup> over the same spectrum but dedicates to transmitting real-time traffic. When the main channel is congested, real-time traffic is transmitted through the side channel. Hence, communication of delay-sensitive applications between mobile devices will never be delayed by concurrent wireless traffic, and its latency only depends on the link propagation delay. The key insight of such a side channel is that the Signal-to-Noise Ratio (SNR) of a wireless channel is usually higher than the SNR required to support the data rate being used, due to inaccurate SNR estimation and conservative rate adaptation in wireless networks. This in-band *SNR margin* can be exploited to encode data as *patterned interference* over the main channel. The impact of such a side channel on packet decoding over the main channel, on the other hand, could be efficiently eliminated by limiting the amount of additional patterned interference within the scope of the main channel’s SNR margin.

In order to support real-time wireless traffic, the wireless side channel has to provide sufficient data throughput that is required by modern interactive mobile applications. Existing schemes have designed the side channel over commodity wireless networks such as WiFi [21] and ZigBee [108], but are limited to achieving a data throughput of several hundreds of kbps. This side channel, hence, can only be used to transmit small network control messages (e.g., RTS/CTS frames) but is far from sufficient for mobile application traffic. Further improving the side channel throughput, in theory, needs to encode more bits into the patterned interference. Such throughput improvement, however, is difficult because the amount of interference that can be exploited for data encoding is limited by the channel SNR margin.

In this chapter, we present a practical high-throughput design of the wireless side channel that efficiently supports real-time wireless traffic, by exploiting the unique properties of modern digital modulation methods, particularly OFDM which will be the technical foundation of next-generation high-speed wireless networks (e.g., LTE-A and 5G) [104]. Our basic idea of the side channel design is to encode data as patterned interference by erasing the en-

---

<sup>1</sup>The existing channel being used in current wireless networks is denoted as the “main channel” throughout the rest of this thesis.

ergy of specific subcarriers in the main channel’s OFDM symbols. Since such energy erasure does not increase the RF transmit power, it can be used to encode data into every OFDM symbol in the main channel, hence dramatically increasing the side channel throughput. On the other hand, since OFDM modulates data into separate subcarriers in both time and frequency domains, the amount of patterned interference could be efficiently controlled by interfering only a small portion of OFDM subcarriers, without affecting the packet decoding in the main channel and its resistance to channel contention.

The major challenges of such high-throughput side channel design, however, are two-fold. First, it is hard to ensure precise detection of patterned interference in the form of energy erasure over OFDM subcarriers, which can be easily corrupted by channel noise. A straightforward solution is to detect energy erasure as the subcarrier with the minimum level of energy. However, when the channel noise is sufficiently high, it may increase the energy level of the erased subcarrier and hence results in detection errors. Second, our proposed design also raises unique challenge to channel operation and control. It is difficult to distinguish data transmitted in the side channel from background noise, without knowing the existence of such a side channel in advance. Lightweight techniques for connection establishment over the side channel are hence needed without continuous detection of patterned interference.

To overcome the aforementioned challenges, we first propose a probabilistic approach to detecting patterned interference, which takes the channel noise model into account and achieves a high detection rate over the side channel. This approach is also resistant to the multipath channel fading and ambient noise. We then indicate the existence and the destination of a wireless side channel by exploiting the idle time periods between data frames in the main channel and inserting a specialized preamble between these frames. This approach, therefore, does not require active cooperation of the main channel and will not be affected by the traffic status in the main channel. We have implemented the proposed system design over practical Software-Defined Radio (SDR) and WARP hardware platforms, and evaluated the effectiveness of our design over realistic VoIP applications. The experimental results show that our system can provide a side channel throughput of up to 2.5 Mbps, which is 10 times

higher than existing work with minimal impairment to the main channel performance. It also reduces the data transmission delay in commodity 802.11 networks by up to 90%, and significantly eliminates the chance of delay jitters in such networks.

The rest of this chapter is organized as follows. Section 2.2 reviews the existing work. Section 2.3 gives a brief overview on the principle of our design. Sections 2.4 and 2.5 present the technical details of the designs of the side channel and connection establishment. Section 2.6 evaluates the performance of our system over the SDR platform via network emulation. Section 2.7 and Section 2.8 further present details of our implementation and performance evaluation over the WARP hardware platform. Finally, Section 2.9 concludes the chapter.

## 2.2 RELATED WORK

To efficiently eliminate the network congestion, various QoS control schemes have been proposed to prioritize delay-sensitive network traffic [86], and TCP flow control also helps reduce the channel congestion [58]. However, these designs cannot scale to the increase of network traffic.

Instead, recent studies aim to reduce the transmission latency by exploiting extra communication opportunities beyond the existing wireless channel. Existing designs of the in-band side channel are mainly applied for delivering coordination and control information, so as to improve the network performance without introducing extra costs compared to the out-of-band approaches. The side channel conveying the control messages is operated concurrently with data transmissions in the main channel, without jeopardizing the latter's performance. Wu et al. [108] observes the under-utilization of interference-tolerance provided by most physical layer implementations, and encodes a small amount of control messages by intentionally interfering several chips in each ZigBee symbol. Flashback [21] builds a control plane with a 400 kbps data rate in OFDM-based WiFi packets, by interpolating high-energy spikes into OFDM subcarriers. These spikes transmit at a power level that is 64 times greater than the power used for regular data symbols, hence allowing easier detection. However, all these designs are limited to conveying small-size network control messages over low-throughput

side channels. In contrast, the major focus of this chapter is to develop a side channel with a throughput that is sufficiently high to support real-time wireless traffic of mobile applications when the main channel is congested. We build our side channel upon commodity WiFi standard and achieve a high throughput by interfering one or more subcarriers in *every* OFDM symbol.

Orthogonal Frequency-Division Multiple Access (OFDMA) [27], a multi-user version of the OFDM design, can be potentially used as an alternative of the side channel, by dedicating a certain amount of subcarriers exclusively for the delay-sensitive traffic. However, the channel throughput for other traffic will inevitably suffer from degradation. For example, if two subcarriers are dedicated to delay-sensitive traffic, the throughput degradation for other traffic will be degraded 4% (2/48). In addition, since our design is implemented at the PHY layer and requires no modification of MAC, it can also work in parallel with other advanced MAC-layer network standards such as 802.11e [52], which prioritizes the delay-sensitive traffic at the MAC. Applications which require immediate responses such as workload offloading in the mobile cloud [30, 31, 99], furthermore, can also benefit from our work.

Our design for connection establishment and control in the side channel is partly inspired by another type of side channel design, which is achieved through appending preambles to the wireless channel. E-Mili [112] constructs the M-preamble to facilitate the sampling-rate invariant packet detection, so that the WiFi receivers could recover from the downclocked idle listening mode once a valid preamble is received. The M-preamble is comprised of duplicated versions of complex Gold sequences (CGS), whose length implicitly conveys the address information. Gap Sense [113] allows interaction between mobile devices with heterogeneous types of wireless networks, by prepending legacy packets with a preamble consisting of multiple energy pulses with quiet periods in between. This quiet period conveys control messages, and can be detected by neighboring nodes with incompatible physical layers. Being different from existing schemes which develop new preambles and hence require specialized hardware or software to operate these preambles, our approach uses a simple duplication of regular WiFi preambles. Hence, our proposed schemes could be easily deployed over commodity WiFi transceivers by being integrated into their firmware, without any modification to the hardware.

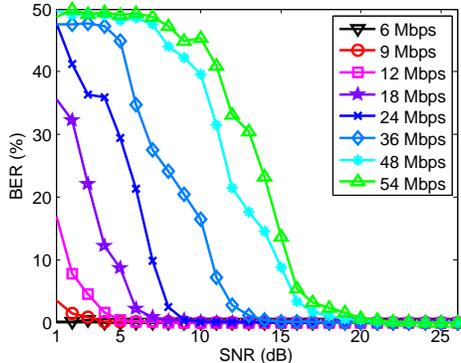


Figure 2.1: Channel BER under different SNR conditions

## 2.3 OVERVIEW

In this section, we first elaborate our motivation of developing a side channel for supporting real-time wireless traffic, by demonstrating the existence of SNR margin in practical wireless channels. Next, we present the high-level picture of our proposed side channel design.

### 2.3.1 Motivation

As described above, our major motivation of designing the wireless side channel is the existence of SNR margin in the wireless channel, i.e., the difference between the actual channel SNR and the minimum channel SNR that is required to support the channel data rate being used. In practical WiFi networks, such minimum channel SNR is specified in order to maintain a sufficiently low Bit Error Rate (BER). For example, 10.7 dB is the minimum channel SNR required by the 802.11a standard to ensure a channel BER lower than 1%, when a 9 Mbps data rate (BPSK with 3/4 code rate) is being used [98].

To further evaluate the amount of SNR margin in practical WiFi networks, we conduct a preliminary experiment over USRP N210 SDR boards with GNURadio toolkit<sup>2</sup>. More specifically, we emulate different levels of channel SNR at the sender side by adjusting the transmit gain of the sending USRP board. At the receiver side, we estimate the channel

<sup>2</sup><http://gnuradio.org/redmine/projects/gnuradio>

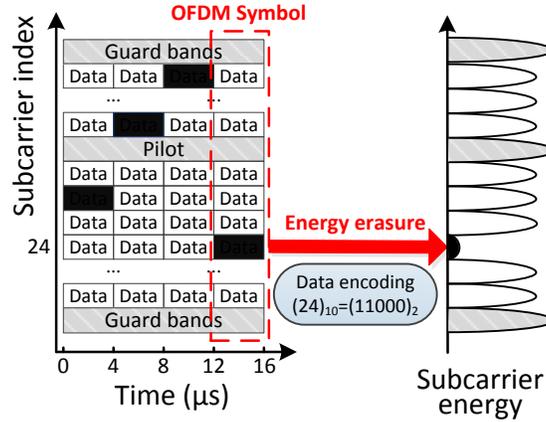


Figure 2.2: Encoding data as patterned interference into an OFDM symbol

status at real-time and compute the BER by comparing the received data frames with expected ones. The experiment results in Figure 2.1 show the relationship between channel SNR and BER when different data rates are being used<sup>3</sup>. For example, it shows that 5.8 dB is the minimum required SNR for the 9 Mbps data rate, and the SNR margin is hence 4.9 dB (calculated as the difference between 5.8 dB and the required 10.7 dB by the 802.11a standard). Such SNR margin will further increase when the wireless channel condition improves and a higher data rate is adopted accordingly.

The existence of such SNR margin, therefore, validates our proposed side channel design. Particularly, note that traditional designs of wireless side channels only exploit a small portion of the SNR margin to deliver small-size control messages. In contrast, we envision that the SNR margin can be further exploited to increase the throughput of the side channel without impairing the performance of main channel. The key challenge, however, is how to appropriately limit the amount of patterned interference being applied.

<sup>3</sup>Different data rates are adopted by using different modulation methods and code rates. For example, 6 Mbps is realized by BPSK with the code rate 1/2.

### 2.3.2 Big Picture

As shown in Figure 2.2, our primary design of the side channel is to erase the energy of one specific subcarrier in each OFDM symbol. At the sender’s side, the patterned interference is only applied to one OFDM subcarrier in each symbol, and hence controlled within the scope of channel SNR margin. At the receiver’s side, an erased subcarrier can be detected as the subcarrier with minimum level of energy. Since we do not require any additional hardware or increase the RF transmit power, little energy consumption is incurred by the side channel. As a result, when we erase the energy of one of the 48 OFDM subcarriers<sup>4</sup>,  $\lfloor \log_2 48 \rfloor = 5$  data bits can be encoded into each symbol and the maximum data rate of the side channel is  $5b/4\mu s = 1.25$  Mbps. For example in Figure 2.2, erasing subcarrier 24 transmits data bits  $(11000)_2$  over the side channel. In practice, this data rate depends on the accuracy of detecting patterned interference, and we propose a probabilistic approach to ensure precise detection of such interference in Section 2.4.

In our side channel design, the main channel information modulated in the subcarriers being erased is unrecoverable, which could potentially impact the performance of the main channel. However, such impact could be efficiently controlled and mitigated if prior knowledge of the corrupted (erased) subcarriers is available. More specifically, the wireless receiver first detects the OFDM subcarriers being corrupted by the side channel before feeding the OFDM symbols to the Viterbi decoder in the main channel. Then, the Viterbi decoder in the main channel could utilize such information to reduce the weight of those bits contained in the corrupted subcarriers when doing its best-path search, known as the erasure operation, so as to avoid decoding error and data packet corruption.

---

<sup>4</sup>Four pilot subcarriers are not used.

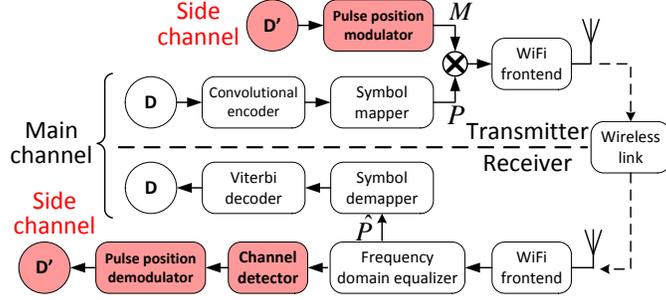


Figure 2.3: Side channel design over WiFi networks

## 2.4 SIDE CHANNEL DESIGN

In this section, we first describe the overall system design of the wireless side channel over commodity WiFi networks. Then, we present the probabilistic approach to detecting patterned interference, which ensures the detection rate of patterned interference in a wireless channel with additive white Gaussian noise (AWGN).

### 2.4.1 Overall System Design

The system architecture of the side channel over WiFi networks is shown in Figure 2.3. Specifically, the real-time traffic is denoted as  $D'$  and conveyed in the side channel, and other network traffic is denoted as  $D$  and transmitted in the main channel. Energy erasure of OFDM subcarriers is implemented through a simple masking and multiplication approach. More specifically, in the transmitter side, a Pulse Position Modulator (PPM) [94] is applied onto  $D'$ . Correspondingly, a sequence of binary numbers, denoted as the mask  $M$ , is generated with ‘1’ and ‘0’ referring to either reserve or erase the energy in a specific subcarrier. For example, in Figure 2.2, erasing the energy of the 24th OFDM subcarrier, which corresponds to encode data  $(11000)_2$  in the side channel, maps to the following mask:

$$\underbrace{1}_{c_{31}} \underbrace{1\dots1}_{c_{24}} \underbrace{0}_{c_{24}} \underbrace{1\dots1}_{c_0} \underbrace{1}_{c_0}$$

This mask contains a single ‘0’ in the position of  $c_{24}$ , and is then applied on the main channel subcarriers, which is equivalent to erasing subcarrier  $c_{24}$  while maintaining all other ones unchanged.

Based on such scheme, the wireless network system consisting of both a main channel and a side channel works as follows. First, the modulation process in the main channel remains unchanged. As shown in Figure 2.3, the convolutional encoder, which uses one type of the Forward Error Correction (FEC) coding scheme, introduces redundancy into the modulation process for error correction. The symbol mapper transforms the binary sequence to a sequence of constellation points  $\mathbf{P}$ , according to the modulation methods being used. After the masking procedure, the carrier mapper would assemble an OFDM packet by filling the data sequence into the time-frequency grids. The FFT module then transforms the data from frequency domain to time domain for transmission. Second, at the receiver side, the signal distortion introduced by multipath fading is compensated by the frequency domain equalizer, by measuring the misshaping of the pre-known long training sequences. The packet  $\hat{\mathbf{P}}$  after equalization is then fed to the symbol demapper and Viterbi decoder for demodulation, and to side channel detector for detection of the erased subcarriers. The side channel information  $\mathbf{D}'$  is then recovered through pulse position demodulator. Since our design retains the OFDM-based PHY layer unchanged, it could be easily deployed onto commodity WiFi systems through firmware integration without any modification to existing hardware or requiring any extra hardware.

#### 2.4.2 Improvement of Data Throughput

Based on this basic design of the side channel, the major approach to further improve the data throughput of the side channel is to erase multiple subcarriers in an OFDM symbol. Let  $K$  be the number of subcarriers being erased in each OFDM symbol, the number of data bits in the side channel that can be transmitted per symbol is hence  $\lfloor \log_2(C_{48}^K) \rfloor$ , which will quickly increase along with the value of  $K$ . For example, a throughput of 2.5 Mbps can be achieved by erasing two subcarriers per OFDM symbol. Essentially, this scheme improves

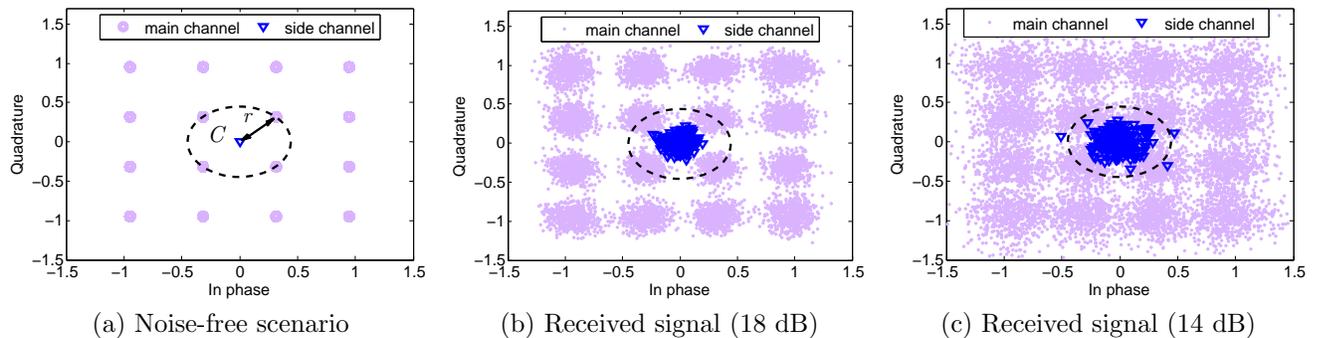


Figure 2.4: Constellation diagram of 16-QAM (3/4) under different channel SNR conditions

the data throughput of the side channel by increasing the amount of patterned interference applied to the main channel. Since such interference is applied as energy erasure, its amount can be arbitrarily increased, in theory, without breaching the radiation power limit.

However, increasing the amount of patterned interference improves the side channel throughput at the cost of increasing the decoding errors and BER in the main channel, as more information being transmitted is removed from the main channel. Therefore, to efficiently control the amount of patterned interference within the scope of channel SNR margin, we limit  $K \leq 2$  (4.1% subcarriers) in our design. In practice, the value of  $K$  is declared in the SIGNAL symbol in a regular WiFi frame. According to the 802.11 standard [1], the SIGNAL symbol is the first symbol after preambles, followed by DATA symbols, determining the data rate and the length of the frame. To ensure reliable delivery, the SIGNAL symbol is always transferred in the lowest data rate, e.g. 6 Mbps using BPSK. Similarly, we apply the lowest data rate ( $K = 1$ ) in the SIGNAL symbol, to encode the information about the side channel data rate being used over the subsequent DATA symbols.

### 2.4.3 Probabilistic Detection of Energy Erasure

Precise detection of patterned interference in form of energy erasure is the vital factor that determines the performance of both the main channel and the side channel. Detection errors not only reduce the throughput of the side channel, but also increase the main channel BER

by misinforming the Viterbi decoder to erase the correct main channel bits while keeping the corrupted bits. The impact of such detection errors is even more severe when multiple subcarriers are being erased.

The major challenge hindering precise detection of patterned interference is channel noise. When sufficient noise exists in the erased OFDM subcarrier, this subcarrier may not be the one with the minimum level of energy and results in detection error, when the straightforward detection approach presented in Section 2.4.1 is being used. According to [91], the received signal  $\mathbf{Y}$  in the frequency domain (after FFT) can be written in time-frequency matrix notation as

$$\mathbf{Y} = \text{diag}(\mathbf{X})\mathbf{H} + \mathbf{N} \quad (2.1)$$

where  $\mathbf{N}$  is AWGN noise and  $\mathbf{H}$  is the Fourier Transform of the system function of the multipath channel  $h(\tau)$ . Since OFDM can efficiently reduce the impact of multipath fading by adopting low-rate symbols and pilot-based frequency-domain equalization [111, 66], we consider AWGN as the major source of the distortion.

Based on such modeling of channel noise, our basic idea of improving the detection accuracy is to probabilistically evaluate the chance for each OFDM subcarrier to contain the patterned interference, and determine such probabilities based on subcarriers' energy levels and up-to-date characteristics of channel noise. We denote  $\mathcal{S}$  as the set of points on the constellation diagram corresponding to the digital modulation method used in the main channel, and  $X$  and  $Y$  as the constellation symbols of the transmitted and received signals in a subcarrier, respectively. From the viewpoint of constellation diagram, the procedure of subcarrier erasure is equivalent to designate an extra constellation point, i.e., the origin  $O = (0, 0)$ , to the side channel signals. Therefore,  $X \in \mathcal{S} \cup O$ . Then, the posterior probability for this subcarrier to contain the patterned interference can be calculated by the receiver as

$$\mathbb{P}\{X = O|Y\} = \frac{\mathbb{P}\{Y|X = O\} \cdot \mathbb{P}\{X = O\}}{\sum_{s \in \mathcal{S} \cup O} \mathbb{P}\{Y|X = s\} \cdot \mathbb{P}\{X = s\}}. \quad (2.2)$$

When there is no channel noise,  $Y$  always equals to  $X$  and the subcarrier with the minimum energy always contains the patterned interference. Otherwise, the smaller  $Y$  is, the higher probability is computed in Eq. (2.2). The prior probability of  $\mathbb{P}\{X = O\} = \frac{K}{N}$  where  $N$  is the number of OFDM subcarriers that are used to transmit data in the main channel.

$\mathbb{P}(X = s) = 1/|\mathcal{S}| \cdot (1 - K/N)$  for all  $s \in \mathcal{S}$ . With AWGN, the conditional probability  $\mathbb{P}\{Y|X = s\}$  can be computed from a two-dimensional Gaussian distribution  $\mathbb{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with  $\boldsymbol{\mu} = O$ , based on the distance between  $X$  and  $Y$  in the complex plane. The noise level is reflected by the co-variance matrix  $\boldsymbol{\Sigma}$  of Gaussian distribution, which correlates to channel SNR and is continuously estimated at real-time [74].

Since Eq. (2.2) computes  $\mathbb{P}\{Y|X = s\}$  for every  $s \in \mathcal{S} \cup O$ , such probabilistic detection of patterned interference with a higher order modulation incurs additional computational overhead. For example, the above probability needs to be computed 65 times if 64-QAM modulation is applied. To reduce such overhead, we investigate the spacial distribution of the erased subcarriers in the constellation diagram. As shown in Figure 2.4(b) and Figure 2.4(c), with a moderate channel condition where the main channel information could be correctly decoded, almost all the received side channel signals in the constellation diagram are within the circle  $C$  that centers on the origin, whose radius  $r$  is the distance between origin and its closest constellation point in the main channel. In other words, the side channel signals are very likely to be falsely classified as the main channel signals with the lowest power in poor SNR scenarios.

In this case, we only need to concern about the received signals with a magnitude less than  $r$ , and classify others as the main channel data. We hence replace the set  $\mathcal{S}$  with  $\mathcal{S}'$  in the denominator of Eq. (2.2), where  $\mathcal{S}'$  only consists of the weakest main channel signals in the constellation diagram.  $\mathcal{S}'$  contains at most 4 elements (2 elements for BPSK), which significantly reduces the computational workload of the probabilistic detection approach.

## 2.5 CONNECTION ESTABLISHMENT AND TRANSMISSION CONTROL

In our design, since data is transmitted over the side channel as patterned interference, it is difficult to distinguish the side channel from background noise, without knowing the existence of such a side channel in advance. Lightweight techniques for connection establishment over the side channel are hence needed without continuous detection of patterned interference.

In this section, we indicate the existence and destination of a side channel by appending specialized preambles to the regular data frames in the main channel, so as to facilitate connection establishment and transmission control in the side channel.

### 2.5.1 Connection Establishment

We enable lightweight connection establishment over the side channel by exploiting the idle time periods between data frames in the main channel. For example, the 802.11 standard requires a DCF Interframe Space (DIFS) of  $34\mu s$  between every two WiFi frames [70, 112]. More specifically, each OFDM-based 802.11 data frame starts with Short Training Sequences (STSs), which consists of 10 duplicate short symbols spanning 16 samples, i.e.,  $0.8\ \mu s$  in a 20 MHz bandwidth system each. The frame detection, together with the Automatic Gain Control (AGC) and timing synchronization, is achieved by computing the auto-correlation of the STS. Hence, as shown in Figure 2.5(a), our approach indicates the existence of the side channel by placing a special preamble, as a set of STSs, to the idle period between two frames in the main channel.

Since a STS is the beginning indicator of a WiFi frame, it can be efficiently detected by the embedded WiFi AGC via auto-correlation without any additional hardware. Such auto-correlation exploits the cyclic characteristic of STSs and takes the power level of the received signal into consideration [84]. In WiFi systems, a new data frame is considered to be detected once the auto-correlation coefficients  $c[n]$  of  $N$  consecutive samples exceeds the threshold  $C_{th}$ . The special preamble indicating the side channel existence, hence, can be detected in the same way. From Figure 2.5(b) which demonstrates such auto-correlation process with 10 dB of channel SNR, we can see that this special preamble can be efficiently distinguished from the data frame in the main channel, even in low-SNR scenarios.

### 2.5.2 Side Channel Addressing

The destination of a side channel may be different from that of the main channel. Hence, we encode the information about the traffic destination of side channel as the relative time location ( $L$ ) of the special preamble in the idle period between data frames. The address

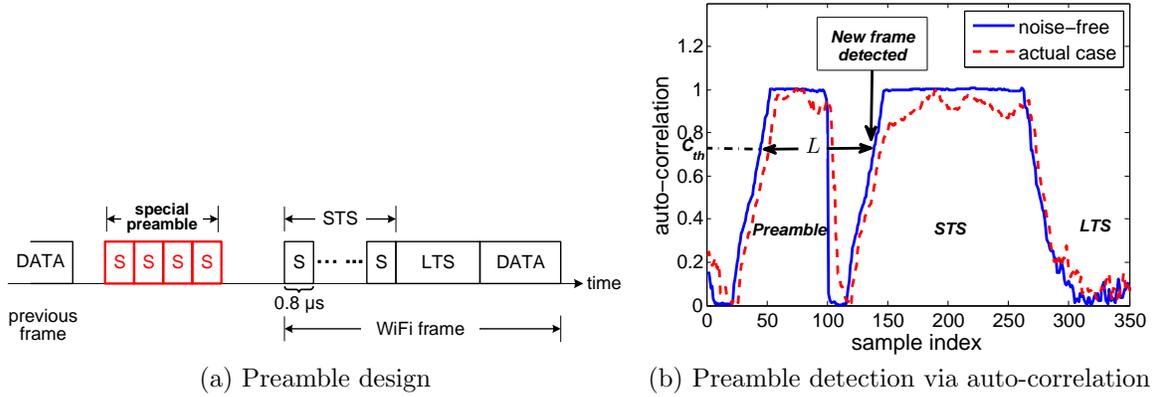


Figure 2.5: Design and detection of side channel preamble

information is represented as

$$Address = (L - L_{guard} - L_{Preamble})/D, \quad (2.3)$$

where  $L_{guard}$  is the guard interval between the special preamble and the regular WiFi frame,  $L_{Preamble}$  is the duration of the special preamble, and  $D$  is the resolution of preamble detection. When  $L_{guard} = L_{Preamble} = 0$  and  $D$  equals to 10 WiFi samples ( $0.5 \mu s$ ), the WiFi receiver can distinguish  $34\mu s/0.5\mu s = 68$  different time locations in the idle period, allowing  $\lceil \log_2 68 \rceil = 6$  data bits being encoded and a maximum number of  $2^6 = 64$  devices to be interconnected at the same time. These data bits could also be used to transmit data acknowledgements that are necessary for reliable data transfer.

The actual amount of data bits being encoded also depends on the values of  $L_{guard}$  and  $L_{Preamble}$ . A small  $L_{guard}$  or a large  $L_{Preamble}$  would mistakenly indicate an immediate start of regular frame, leading to frame detection error in the main channel. On the contrary, a large  $L_{guard}$  reduces the number of data bits being encoded, and a small  $L_{Preamble}$  makes the preamble detection prone to channel noise. By experimentally investigating with the off-the-shelf WiFi receivers, we found the preamble consisting of 2 STSs with a guard time of 10 samples is the most suitable design choice for the wireless side channel.

## 2.6 PERFORMANCE EVALUATION OVER SOFTWARE-DEFINED RADIOS

In this section, we evaluate the performance of our proposed side channel design over USRP/GNU Radio platforms. The metrics being used in our evaluations include the data transmission delay, channel throughput, and the detection rate of patterned interference.

### 2.6.1 System Implementation and Experiment Setup

We implemented our design over USRP SDR boards with the GNURadio toolkit, which realize an 802.11a WiFi transceiver over the 5.85 GHz frequency band. USRP N210 motherboards and UBX 40 RF daughterboards are used, which operates in the range between 10 MHz and 6 GHz. In the main channel, the frame generation follows the IEEE 802.11a standard [1]: the first OFDM symbol of each frame stores 3-byte Physical Layer Convergence Protocol (PLCP) header indicating the length and data rate of the frame, followed by the MAC Protocol data unit (MPDU) where a random sequence of strings is located onto its frame body. For each OFDM symbol, 32 subcarriers with index -24 to 11 are exploited for the side channel, excluding the DC (index 0) and pilot (index  $\pm 21$  and  $\pm 7$ ) subcarriers.

We use this implementation to conduct experiments in a  $5m \times 5m$  office which contains rich multipath fading effect. The sender and receiver are placed out of line of sight and 4 meters away from each other. We tune the channel SNR by adjusting the USRP parameter of Tx Gain, which represents the RF transmit gain within a range from 2 dB to 30 dB. The maximum Tx power then corresponds to 13 dBm. In each experiment, 250 WiFi packets of 1 kB are synthesized and sent with an interval of 2 seconds, ensuring the timely executions of BER computation and other statistical operations on the PC host connected to USRP boards. In addition, the noise invariance is estimated every 5 frames.

Besides the synthesized packets being sent at a consistent rate, we also evaluate our system with realistic real-time WiFi traffic, more specifically, peer-to-peer Voice over IP (VoIP) traffic. To ensure smooth conversations between endpoints, the VoIP traffic has strict requirements on the network transmission delay: any delay exceeding 400 ms will seriously

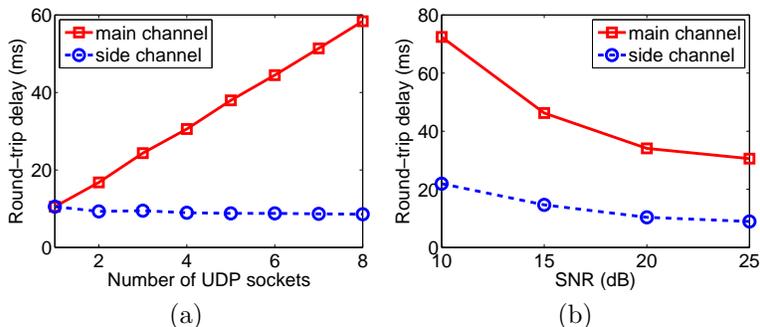


Figure 2.6: Data transmission delay

impair the quality of conversation [55]. We choose to use the VoIP traffic generated by Skype, which transfers an uplink UDP packet every 20 ms. Due to the difficulty of directly interacting between the GNURadio platform and the Skype application, we use the network sniffer Wireshark to capture the uplink Skype packets from a laptop, during which we upload large files to Google Drive to simulate congested wireless network scenarios.

## 2.6.2 Data Transmission Delay

We first evaluate the data transmission delay over the wireless side channel. In particular, we measure the round-trip delay for the sender to “ping” the receiver. To emulate the network congestion in the main channel, we generate multiple UDP sockets in the main channel between the sender and receiver. In each UDP socket, we continuously transmit packets of 1 kB every 100 ms. At the PC host operating the USRP board, each UDP socket is operated in a separate application thread and the time for each transmission is individually recorded. Meanwhile, the same type of packets are sent through the side channel with the same interval between consecutive packets.

The round-trip data transmission delay through the main channel and the side channel, with different number of UDP sockets in the main channel, is shown in Figure 2.6(a). When the number of UDP sockets increases, the main channel is getting congested due to their contention for channel access, and the data transmission delay increases linearly. However,

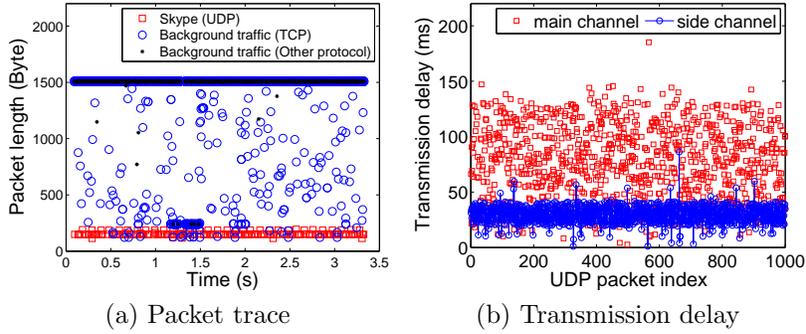


Figure 2.7: Evaluation with VoIP traffic

the transmission delay over the side channel remains constantly at a low level in all cases, because data delivery will never be delayed in the side channel and only experience the link propagation delay. It's noted that there is a slightly descending trend of the side channel delay with a larger number of UDP sockets in the main channel. The major reason of this descending trend is the multi-threaded decoding operations at the PC host, which affects the packet processing delay. The OS at the PC host would provide more than enough resources when multiple threads are used.

Besides, we evaluate the data transmission delay under different channel SNR scenarios. We use 4 UDP sockets in our experiment to emulate the main channel traffic, and tune the RF Tx gain to simulate different SNR conditions. The main channel data rate is configured to be 24 Mbps. As shown in Figure 2.6(b), when the channel SNR drops, the data transmission delay in the main channel quickly increases due to the higher chance of packet corruption and retransmission. Comparatively, data traffic in the side channel is only slightly affected and experiences a delay increase less than 10 ms. Such resistance to SNR degradation and fluctuation is another major advantage of our side channel design, and is important to support real-time data traffic.

The realistic VoIP traffic has also been exploited to evaluate the effectiveness of our design in supporting real-time applications in practice. In our experiment, the data rate in the main channel is set to be 18 Mbps. The characteristics of the UDP VoIP traffic and concurrent TCP traffic, which are recorded as described in Section 2.6.1, are shown in Figure

2.7(a). After packet extraction and reconstruction, we set one USRP board as the sender to deliver the recorded packets concurrently to another two USRP boards, which are set to be the recipients of the VoIP packets and other traffic packets, respectively.

The VoIP packets are carried by the side channel and the main channel in two separate experiments for fair comparison, and the transmission delays under these two cases are shown in Figure 2.7(b). The results in Figure 2.7(b) shows that the side channel can effectively reduce the transmission delay by 60% (reducing the average transmission delay from 75.1 ms to 29.7 ms). In addition, since the side channel is dedicated to the real-time traffic, packets being transmitted in the side channel will not experience any channel contention from other ongoing traffic, and hence experience a much small delay jitter.

### 2.6.3 Detection Rate of Patterned Interference

We first evaluate the detection rate of patterned interference, when the subcarrier containing patterned interference is detected as the one with the minimum level of energy. The detection rates of such patterned interference, when one and two subcarriers are erased in an OFDM symbol, are shown in Figures 2.8(a) and 2.8(b), respectively. When the channel SNR is higher than 10 dB, the detection rate is nearly 100% with BPSK and QPSK. When higher-order modulation schemes such as 16-QAM and 64-QAM are being used, the detection rate would improve as the SNR increases, and 90% detection rate could be achieved under when the SNR is higher than 20 dB.

Figure 2.8 shows that the detection rate of patterned interference is tightly correlated with the SNR condition. Such correlation can be better illustrated by Figure 2.9 which plots the constellation diagrams of the received symbols in different SNR conditions. When the SNR is good as shown in Figure 2.9(b), all the side channel symbols are near the origin on the constellation diagram and can be accurately separated from the adjacent four constellation points used by the main channel. However, when the SNR drops, we notice from Figure 2.9(a) that the symbols of the main channel and the side channel are mixed together and become harder to be separated. In addition, the detection rate is also affected

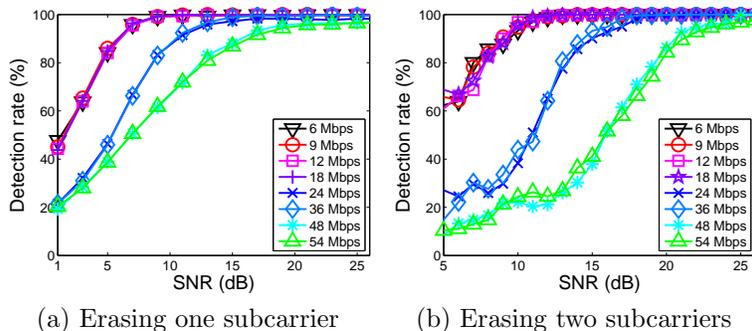


Figure 2.8: Detection rate of patterned interference

by the modulation scheme being used. When a higher-level modulation such as 64-QAM is applied, the distance between constellation points becomes smaller, making it more difficult to differentiate between the side channel and the main channel signals.

Table 2.1: Side channel detection rate with a 9 Mbps data rate

SNR (dB)	4	6	8	10	12	14	16
Basic approach (1)	0.22	0.45	0.72	0.90	0.98	0.99	1.00
Prob. approach (1)	0.34	0.61	0.84	0.93	0.98	0.99	1.00
Basic approach (2)	0.07	0.24	0.56	0.86	0.96	0.98	1.00
Prob. approach (2)	0.18	0.42	0.74	0.90	0.97	0.98	1.00

Based on these results, we further evaluate the performance of our proposed approach in Section 2.4.3 to probabilistic detection of patterned interference. The evaluation results are summarized in Table 2.1, where (1) and (2) denote the cases in which one and two subcarriers are erased in each OFDM symbol, respectively. Generally speaking, the probabilistic approach of patterned interference detection is able to improve the detection rate by 20%, especially in low SNR scenarios.

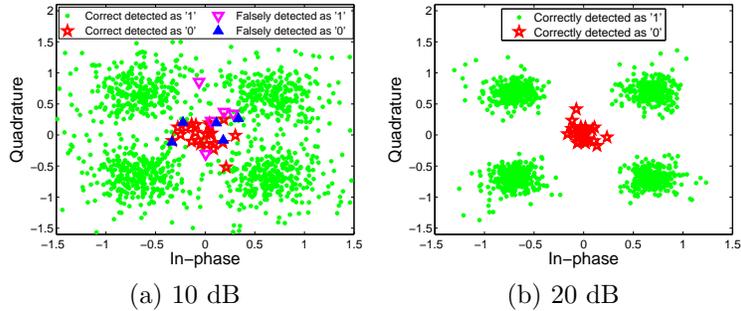


Figure 2.9: Constellation diagrams under different SNR (QPSK 3/4)

### 2.6.4 Data Throughput of the Side Channel

Built on precise detection of patterned interference, we evaluate the data throughput of the side channel in practical network scenarios with different channel SNR conditions. Intuitively, the higher the channel SNR is, the more accurate the patterned interference could be detected, and hence the higher data throughput could be realized in the side channel. Table 2.2 describes the throughput of the side channel with one subcarrier being erased in each OFDM symbol. The side channel could achieve a throughput higher than 1 Mbps with BPSK when the SNR is higher than 8 dB. The maximum data rate of 1.25 Mbps can be approached when the SNR is higher than 16 dB.

Table 2.2: Side channel throughput (Mbps) with one subcarrier being erased in each OFDM symbol

SNR (dB)	4	8	12	16	24	30
BPSK	0.781	1.170	1.213	1.233	1.235	1.235
QPSK	0.784	1.176	1.237	1.226	1.235	1.225
16-QAM	0.397	0.807	1.116	1.186	1.188	1.210
64-QAM	0.350	0.620	0.888	1.101	1.157	1.200

Furthermore, when two subcarriers are erased in each OFDM symbol, the minimum SNR required to achieve 95% of the maximum throughput of 2.5 Mbps is shown in Table 2.3. Compared to Table 2.2, at least another 3 dB is required in the channel SNR to maximize the data throughput.

Table 2.3: SNR requirement with two subcarriers being erased in each OFDM symbol

Modulation type	BPSK	QPSK	16-QAM	64-QAM
Minimum SNR (dB)	13	15	20	27

### 2.6.5 Impacts on the Main Channel

The encoded information conveyed by the subcarriers being erased is entirely discarded, which will inevitably have impacts on the capability of noise resistance of the main channel. When the side channel is enforced to the main channel with a BER of 1%, we evaluate the increase of the main channel BER due to the existence of the side channel, and the results are shown in Figure 2.10. Since our proposed design is able to appropriately control the amount of patterned interference being applied to the main channel, we are able to limit the increase of the main channel BER within 0.6%, and hence retain the performance of the main channel in various practical scenarios.

## 2.7 HARDWARE IMPLEMENTATION

In order to evaluate our proposed side channel design over more realistic wireless scenarios, we also implemented our side channel design over WARP v3 boards, which is a FPGA-based wireless hardware platform and allows hardware-based implementation of a fully functional wireless transceiver over its FPGA core [37] as show in Figure 2.11. It hence allows real-time wireless networking and full compatibility with commodity wireless devices: since all the Tx and Rx operations are being conducted over the FPGA hardware instead of via GNU software

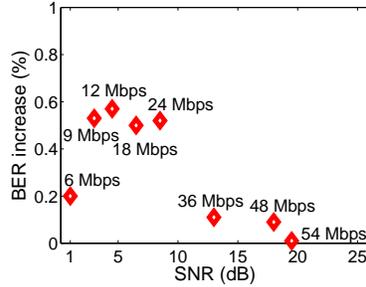


Figure 2.10: Main channel BER increase due to the existence of the side channel

emulation, it incurs near-zero computation delay to our proposed side channel operation. Our implementation requires minimal modification to the existing wireless system by adding an extra processing subsystem to the original 802.11 PHY. In this section, we describe in detail how the side channel design is implemented into the WARP platform.

### 2.7.1 Implementation Overview

The architecture of our hardware implementation of the side channel design is shown in Figure 2.12. Based on the original 802.11 PHY structure, the added side channel subsystem is highlighted in color. It does not affect any existing modules in the existing 802.11 PHY, but instead adds new sub-blocks after channel modulation in the Tx side and channel equalization in the Rx side.

At the Tx side, the side channel block is added when the main channel’s modulation starts after its OFDM subcarriers are generated. The side channel block is operating in parallel with the modulation block in the main channel. In the side channel, the input data stream will build a data codeword with every 5 data bits, and this side channel codeword represents the location of subcarrier to be erased in the main channel. Afterwards, the I/Q data for this subcarrier in the main channel will be changed to zero, indicating erasure of subcarrier energy. Details will be described in Section 2.7.2.

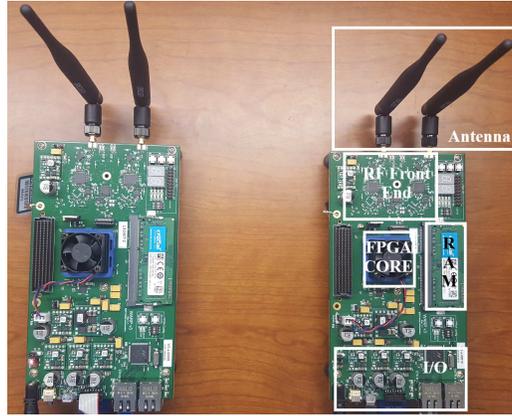


Figure 2.11: WARP3 Platform

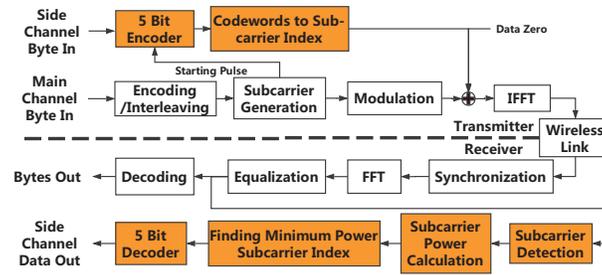


Figure 2.12: Side Channel Implementation over WARP

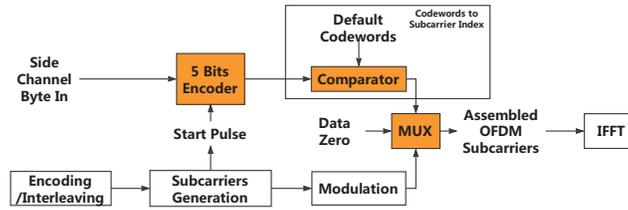


Figure 2.13: Details of Tx design

At the Rx side, after the received RF signal passes through channel equalization, a subsystem for detecting the index of subcarrier with the minimum power in the received OFDM symbols is added to the original 802.11 receiver, and the side channel codeword is then represented by the subcarrier index with the lowest power. In our implementation, the main channel subcarrier with the minimum power is decided by comparing the I/Q signal of the OFDM symbol in different main channel subcarriers. Finally, a 5-bit hardware decoder is used to convert the side channel codeword to side channel data.

### 2.7.2 Delay Reduction

Adding the side channel operation to erase the subcarrier energy in the existing 802.11 system may incur additional computational delay on the Tx side. Such delay in FPGA design is represented by clock cycles, i.e., the amount of main clock cycles that are needed to process each side channel codeword.

To minimize such delay, as shown in Figure 2.13, we strive to implement the side channel Tx by solely using comparators and multiplexers that introduces no extra delay in hardware, and avoid using blocks of multipliers or adders, each of which adds three extra clock cycles of delay into the design over the Virtex-6 FPGA core on WARP v3. More specifically, as shown in Figure 2.13, we pre-load the entire set of 32 possible 5-bit codewords (i.e., 00000 to 11111) in the on-board RAM of WARP. A comparator that compares the side channel codeword with the pre-load codewords is then used to find out which codeword is actually

being transmitted in the side channel. Afterwards, instead of using a multiplier, we use a multiplexer (MUX) to apply signal ‘0’ to the corresponding main channel subcarrier by replacing the original signal in that subcarrier. In this way, we can effectively reduce the delay caused by the side channel computation to zero in theory on Tx.

However, multipliers will be required at the Rx side to calculate the power in each main channel subcarrier from its I/Q signal. Therefore, the additional computation delay at the Rx side is inevitable, and we will experimentally evaluate such delay in Section 2.8.3.

### 2.7.3 Channel Symbol Synchronization

Another crucial issue in our implementation is how to precisely synchronize between the side channel and the main channel. If they are not synchronized, the side channel codeword will be mapped to wrong subcarriers in the main channel and result in serious bit error. However, the accuracy of such synchronization is generally difficult to be ensured at FPGA, because the generation of main channel subcarriers and PHY-layer mapping of the side channel codewords could take different amounts of computational time at the FPGA core of WARP.

To solve this problem, as shown in Figure 2.13, our system asserts a starting pulse after the first symbol is generated in the main channel. This starting pulse is sent to the side channel block for synchronization, and the generation of side channel codewords will then be triggered by this pulse. Since the main channel modulation takes negligible computational delay after the rising edge of the starting pulse, and the generation of side channel codewords also produces zero computation delay according to Section 2.7.2, they can be precisely synchronized with each other.

## 2.8 PERFORMANCE EVALUATION OVER WIRELESS HARDWARE

Built on our hardware implementation in Section 2.7, we further evaluate the performance of our side channel design over the 2.4 GHz frequency band in practical wireless scenarios, in which the wireless network may be unexpectedly interfered by concurrent wireless transceivers or disrupted due to various environmental uncertainty.

### 2.8.1 Experiment Setup

Our experiments are solely operated by the WARP v3 boards without the help of host PC, and are completely compatible with commodity WiFi devices in the same frequency band. In our experiments, packets with random payload of 1400 bytes are constantly transmitted in the main channel, and one subcarrier among the 52 OFDM data subcarriers in each symbol of the main channel is exploited as the side channel. To test the performance of our design under different channel conditions, we tuned the Tx gain on WARP transceivers from -9dB to 11dB, to achieve a maximum main channel SNR at 23dB. To evaluate the communication and computation overhead of the side channel operations, each WARP board is connected to a laptop PC with Intel 4720HQ 2.6Ghz and 16Gb of memory through an ethernet cable at 1Gbps. All the experiment results are averaged over multiple experiment runs under different indoor setup for statistical convergence.

### 2.8.2 Side Channel Performance

In this section, we evaluate the performance of our side channel design in the more commonly used 2.4GHz frequency band. Our experiment is conducted between two WARP boards, one of which serves as the wireless access point (AP) and provides wireless access in the 2414MHz channel with different modulation schemes.

As shown in Figure 2.14, the side channel BER steadily drops when the main channel SNR improves. When the main channel SNR is greater than 12dB, the side channel BER is lower than 5% with BPSK and QPSK modulation. When the main channel SNR is higher than 19dB, the side channel BER is lower than 5% for all modulation schemes. Furthermore,

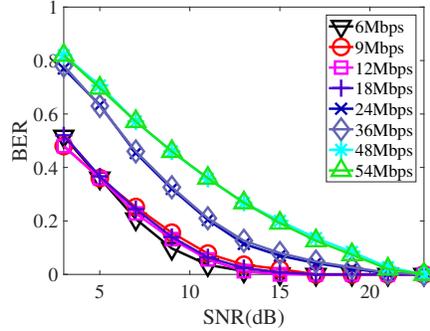


Figure 2.14: Side Channel BER in the 2.4GHz channel

Figure 2.15 shows that the main channel throughput suffers little performance loss from the existence of the side channel, and retains more than 98.5% of its original throughput. In particular, when different modulation schemes are being applied in the main channel, the loss in the throughput is negligible.

In our experiments, we observed a 5% difference in the side channel BER, when comparing our results with the results under the same main channel SNR obtained in Section 6.3 over the 5GHz band. The primary reason for such difference is that the 2.4GHz band is more commonly used and hence has more interference from surrounding wireless devices. Such difference may also be caused by the RF heterogeneity of different wireless transceivers being used in experiments. Generally speaking, we conclude that our side channel design has a consistent performance over both 2.4Ghz and 5Ghz wireless channels.

### 2.8.3 Computation Delay and Overhead

In this section, we evaluate extra computation delay caused by the side channel operation at the wireless PHY. As we stated in Section 2.7.2, the side channel Rx will introduce 8 additional clock cycles to finish due to the involvement of multipliers. With a 160Mhz main clock on the WARP v3 board, such delay brought by the side channel operation is 50ns at the receiver.

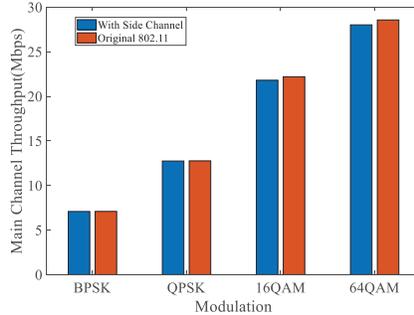


Figure 2.15: Main Channel Throughput

Based on such result, we can further evaluate the extra processing power required by the side channel operations. Since we implemented the side channel over WARP’s FPGA core as slices with basic digital logic blocks such as LUTs, flip-flops and carry logic elements, we can measure such computation overhead as the amount of slice registers being used in the FPGA implementation. More specifically, the Virtex-6 FPGA on the WARP v3 board has 37680 slices of configurable logic blocks, and our side channel implementation only uses less than 1% of these computational resources.

#### 2.8.4 Impact of Channel Interference

To further investigate the performance of our side channel design in practical wireless scenarios with strong interference, we introduce a third WARP node running as an original WiFi AP into experiments, and place it between the two other wireless transceivers that operate the side channel. This node is configured to keep broadcasting dumb data over the same frequency band where the main channel is operated. As shown in Figure 2.16, the amount of channel interference could be then controlled by varying the distances  $d1$  and  $d2$  between the three wireless nodes.

In the first experiment, we set  $d1 = 3m$  and  $d2 = 0.15m$ , and the Tx power for node C is set to 15dBm. We then evaluate the main channel and side channel throughputs at the same time. Table 2.4 lists the maximal throughputs in the side channel and the main channel,

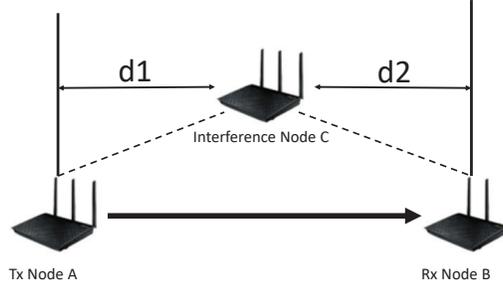


Figure 2.16: Nodes Placement

respectively, and shows that the main channel throughput can suffer up to 55% degradation with 64-QAM modulation, compared to that without interference. On the contrary, the side channel throughput remains at the same level, and we only witness a maximum of 1% performance loss with 64-QAM. This result shows that when the main channel throughput is greatly impacted by the interference, the impact of throughput on the side channel is negligible.

Table 2.4: Channel Throughput (Mbps). Numbers in brackets are the throughput without interference.

Modulation	BPSK	QPSK	16-QAM	64-QAM
Main channel	5.4 (6.85)	8.19 (12.54)	11.68 (17.97)	13.99 (25.46)
Side channel	1.22 (1.22)	1.22 (1.22)	1.21 (1.21)	1.20 (1.19)

Furthermore, we increase  $d1$  to 7m, so that the wireless interference from C is relatively even stronger at the side channel receiver B. We fixed the A's transmission power and B's receiver gain at the same level of -5dB, which gives the channel SNR around 9dB at B. According to the results shown in Figure 2.17, when 64-QAM modulation is being used in the main channel, the side channel BER could increase by 7%. For all other modulation schemes

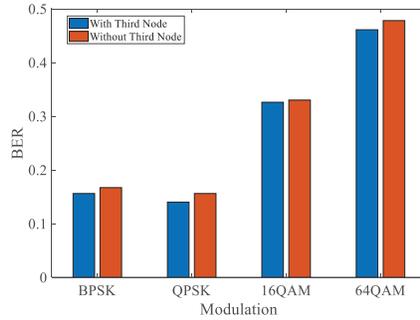


Figure 2.17: Side Channel BER with wireless interference. Tx power is set as -5dB.

being used in the main channel, our design could effectively restrain the BER increase within 5%. Considering the low levels of the side channel BER, the impact of channel interference on the side channel operation is nearly negligible. These experiment results show that, in most of the practical environments, the extra wireless transceiver will have negligible impact on our side channel operation, even when the main channel performance is heavily impacted by the interference.

## 2.9 CHAPTER SUMMARY

In this chapter, we present a novel design of high-throughput wireless side channel, which efficiently supports real-time wireless traffic without consuming any additional wireless spectrum resource. The basic idea of our side channel design is to exploit the SNR margin in the main channel to encode data as patterned interference, and realize such patterned interference as energy erasure over OFDM subcarriers. We have implemented and evaluated the side channel design over practical SDR platforms and FPGA based hardware platform. The experiment results verify the effectiveness of the side channel in reducing the data transmis-

sion latency and providing a data throughput higher than 1 Mbps, with minimum impact on the performance of the main wireless channel. Our future work will incorporate the proposed side channel design into latest wireless network standards, such as 802.11n and 802.11ac.

### 3.0 EASYPASS: COMBATING IOT DELAY WITH MULTIPLE ACCESS WIRELESS SIDE CHANNELS

#### 3.1 INTRODUCTION

The Internet of Things (IoT) consists of billions of heterogeneous sensing devices that interact with and collect data from the physical world. Delay of transmitting such sensory data in IoT, in many cases, should be minimum for a prompt response against the environmental context. For example, smart vehicles need to timely transmit sensory data about their surroundings to each other or the remote cloud for real-time processing, so as to recognize and avoid potential risks [54, 4, 53]. Real-time data transfer is also required for remotely monitoring and controlling smart home appliances [71], smart grid systems [28] and intelligent robots [7].

Unfortunately, the real-time IoT data traffic may be seriously delayed in practice, when competing with other concurrent data transmissions for wireless channel access. A straightforward solution to such channel contention is to exploit additional wireless spectrum, but is usually infeasible due to the scarcity of available spectrum resources or poor propagation characteristics of the new spectrum (e.g., short wavelength and high directionality of the mm-wave band [110, 93, 96]). Recent technical advances such as OFDMA [67] and 802.11ax [5] allow splitting an existing wireless channel to allocate dedicated spectrum for IoT traffic, but require centralized coordination that is difficult to be realized over distributed IoT devices with limited local resources. Instead, researchers suggest to adapt the current CSMA-based wireless MAC protocols to the instantaneous channel condition [11, 15] or traffic characteristics [43, 29] for better channel accessibility, but cannot scale up to the large population of IoT devices.

Our experiments over the off-the-shelf wireless devices demonstrate the potential of combating such IoT delay by exploring new data transmission opportunities, as a *wireless side channel*, over the existing wireless channel that is occupied but under-utilized<sup>1</sup>, but also pose its limitations. Such a side channel builds on conservative rate adaptation in current wireless networks, which results in channel SNR that is usually higher than the SNR required to support the link data rate being used. This in-band *SNR margin* could be as high as 7 dB in commodity wireless networks. It can hence be exploited to transmit delay-sensitive IoT data without waiting for idleness of the main channel, and results in the minimum transmission delay over highly congested wireless links. Recent research builds the side channel over WiFi [21, 69] and ZigBee [109] by alternating the characteristics of the main channel transmitter, such as signal duration [17], signal strength [48, 77], and power spectral density [42, 34, 102]. but only operates the side channel at the sole wireless device that accesses the main channel. The side channel throughput, as a result, is lower than 100 Kbps due to the FCC limit of the main channel’s transmit power, which prevents the SNR margin from being fully exploited and only allows transmitting small control messages (e.g., RTS/CTS frames) or secret keys.

Supporting multiple access to the wireless side channel from different IoT devices, on the other hand, is the key to remove this limitation, but is a challenging problem for three key reasons. First, when the side channel and the main channel are operated by different wireless devices, the RF signal of the side channel will be appended to the existing main channel signal as *extra interference* over the same spectrum, and it is hence difficult to demodulate the side channel data being transmitted from the mixed signal. Second, the side channel signal has to be transmitted with much weaker power to minimize its impact on the main channel. This reduction in Tx power lowers the side channel’s effective SNR and increases the error rate of data demodulation. Third, the environmental dynamics and uncertainty in IoT applications results in sporadic and intermittent access to the side channel, leading to additional difficulty in efficiently utilizing the side channel spectrum from distributed IoT devices.

---

<sup>1</sup>Such existing wireless channel is denoted as the “main channel” throughout the rest of this chapter.

In this chapter, we present *EasyPass*, a new wireless PHY technique that ensures ultra-low transmission latency over multiple IoT devices, by allowing them to concurrently operate the wireless side channel in the same occupied spectrum. EasyPass achieves its objective by introducing three key design components: 1) *Asynchronous signal extraction*: A real-time signal processing module that utilizes the asynchrony between the main channel and the side channel for precise side channel demodulation; 2) *Adaptive transmit power control*: A collection of power control mechanisms that adapt the amount of side channel’s extra interference to the main channel’s SNR margin, hence maximizing the side channel throughput with negligible impact to the main channel; 3) *Fine-grained spectrum allocation*: A distributed algorithm that allocates the available side channel spectrum among IoT devices with respect to their real-time traffic demands, with minimum overhead and conflict. Built on these components, EasyPass is fully compatible with commodity wireless radios. Instead of completely redesigning the wireless PHY [107], side channels in EasyPass use the same modulation and coding techniques as those in the main channel but require minimum coordination with the main channel. Hence, it only modifies the PHY hardware at the side channel receivers, and retains all the main channel transceivers as intact.

How can a side channel receiver precisely extract the side channel data being transmitted from the mixed RF signal on the air? Conventional wisdom transmits side channel data as extra interference with pre-known patterns [69], but will experience great accuracy degradation when the side channel’s transmit power drops. Instead, EasyPass holds the transmission of each side channel frame until the end of a main channel frame’s preamble, from which the main channel’s impulse response can be estimated. This estimation, then, allows the side channel receiver to completely remove the main channel signal from the mixed RF signal being received, without requiring any time synchronization between the main channel and the side channel. Even if the main channel is always occupied when a side channel is being used, the transmission latency in the side channel is limited to the duration of one data frame in the main channel, and is much shorter than that caused by channel contention.

The side channel, however, has to transmit data with much weaker power. To ensure correct data reception, EasyPass exploits the unique properties of modern digital modulation methods, particularly OFDM as the technical foundation of next-generation high-speed

wireless (e.g., gigabit WiFi, LTE-A and 5G) [104], to apply the side channel’s extra interference only to a small portion of OFDM subcarriers in the main channel’s spectrum. In this way, stronger power can be applied to each subcarrier being used by the side channel for higher side channel SNR. EasyPass flexibly controls such number of subcarriers being used and the power level of interference being applied to each subcarrier, to minimize their impact on the main channel. It also allows multiple access to the side channel by allocating these subcarriers among IoT devices in a fully distributed manner.

We implemented EasyPass over the USRP Software-Defined Radio (SDR) platform with the GNURadio toolkit over a 5.8 GHz WiFi frequency band. We adopt the well-established power control scheme in commodity wireless standard to control the side channel RF power, and use commodity rate adaptation algorithms to emulate the practical MAC behaviors. We tested the performance of EasyPass over a small-scale SDR testbed that is deployed with highly dynamic variations of wireless link quality. Experiment results show that EasyPass can efficiently restrain the IoT data transmission delay within 10 ms over highly congested wireless spectrum, and also greatly suppress the delay jitter with dynamically varying IoT population and traffic patterns. Meanwhile, it provides a cumulative side channel throughput up to 2.5 Mbps that can be flexibly allocated to up to 13 IoT devices in a single narrowband channel, and hence becomes a key enabler of many delay-sensitive IoT applications.

### 3.2 EASYPASS DESIGN

EasyPass provides multiple accessibility to the wireless side channel by allowing each IoT device to apply extra interference to an exclusive group of OFDM subcarriers. The core component of EasyPass is the *Asynchronous Signal Extraction* module. Unlike traditional signal unmixing algorithms that can only apply to signals with fixed patterns from a pre-known dictionary [41, 59], EasyPass utilizes the asynchrony between the main channel and the side channel to extract the side channel signal from the mixed RF signal on the air, with minimum errors and overhead. For a typical mixed channel transmission model shown in Figure 3.1(a), the side channel Tx uses the same frame format as the main channel Tx does,

but overhears the main channel and holds its transmission of each frame until the end of LTS in a main channel frame. Then, as shown in Figure 3.1(b), the side channel Rx estimates the main channel response before the side channel frame starts, and uses this knowledge to remove the main channel signal from the received signal on the air.

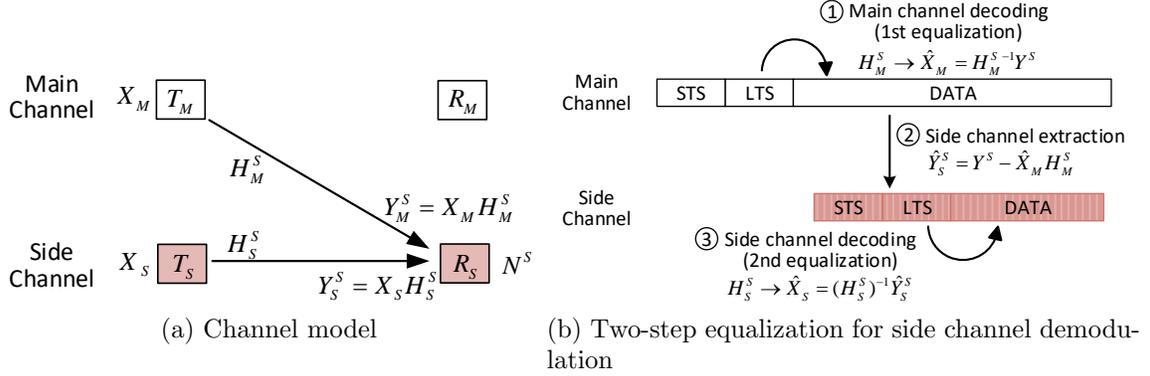


Figure 3.1: Asynchronous signal extraction in EasyPass

### 3.2.1 Two-step Equalization

Such asynchronous signal extraction is achieved by two consecutive steps of channel equalization at the side channel Rx. For each subcarrier where the side channel operates, as shown in Figure 3.1(a), the mixed signal  $Y$  received by the side channel Rx is

$$Y = Y_M + Y_S + N_S = X_M H_M + X_S H_S + N_S, \quad (3.1)$$

where  $H_M$  and  $H_S$  indicate channel responses and  $N_S$  indicates channel noise. Since the side channel Tx only starts transmitting its frame after the LTS of the main channel frame ends, the side channel Rx estimates the main channel response  $H_M$  in advance, and further use  $H_M$  to demodulate the main channel signal as  $\hat{X}_M = H_M^{-1} Y$  by considering  $Y_S$  as background noise after the side channel frame starts. We subtract  $\hat{X}_M(H_M)$  from  $Y$ , and detect the preamble of the side channel frame from the residual signal  $\hat{Y}_S$  to estimate  $H_S$ . Then, the 2nd channel equalization over the residual signal is able to demodulate the side channel data as  $\hat{X}_S = H_S^{-1} \hat{Y}_S$ .

Note that, since the side channel frame does not need to align with the LTS end of the main channel frame, EasyPass does not require any strict time synchronization between the main channel and the side channel, and is hence applicable to any IoT devices with local resource constraints. The local processing delay of the extra channel equalization at the side channel Rx, on the other hand, is proportional to the total number of OFDM subcarriers used by the side channel. Since the side channel usually operates only a small portion of the available OFDM subcarriers, this delay will be much lower than that of an ordinary wireless receiver and have negligible impact on the side channel transmission delay. We will experimentally investigate this delay in Section 3.6.4.

### 3.2.2 Multiple Accessibility of Side Channel

Built on the two-step equalization, multiple accessibility of the side channel hence depends on the number of subcarriers that can be used by the side channel without impacting the main channel. To investigate such multiple accessibility, we operate the main channel and the side channel between two separate pairs of USRPs, which are placed 3 meters away and both transmit at 6 Mbps. The side channel is only applied to some ( $N$ ) of subcarriers with 3dB lower power.

Figure 3.2 plots the correlation between the main channel SNR and side channel SNR with different numbers ( $N$ ) of subcarriers being used by the side channel, and demonstrates that the maximum value of  $N$  closely relates to the SNR margin in the main channel. As the main channel SNR improves, stronger power can be used in the side channel as extra interference, which can then be distributed to more subcarriers to transmit data. For example, when the main channel SNR improves from 3.5 dB (the minimum  $SNR_{req}$  required for 6 Mbps) to 6.5 dB and results in a 3 dB SNR margin, it allows  $N$  to increase from 4 to 16<sup>2</sup>. Such number could be even higher when higher-order data rates are being adopted in the main channel, allowing more IoT devices to access the side channel at the same time. On the other hand, when the link quality in the main channel degrades, the side channel can also

---

<sup>2</sup>The value of  $N$  must ensure that the side channel SNR also exceeds 3.5 dB.

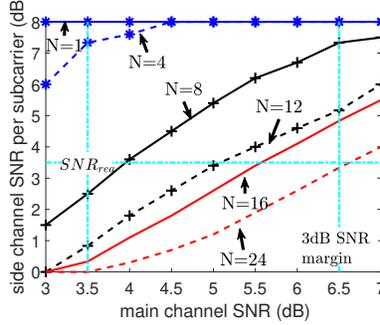


Figure 3.2: Multiple accessibility for side channel

ensure correct data transmission by reducing the number of  $N$  and increasing the transmit power in each subcarrier being used. We will further elaborate on balancing between these two aspects via *Adaptive Transmit Power Control* in Section 3.3.

### 3.2.3 Minimizing Symbol Errors

Figure 3.3(a) plots the side channel signals extracted by the 2nd equalization, and demonstrates non-negligible demodulation errors when the main channel operates at 6 Mbps under a dispersive fading channel with 3 dB SNR. The side channel in this experiment, on the other hand, only occupies 3 subcarriers with a power of 2 dB higher than the noise floor. The major reason for these errors is that the channel response could vary over time in practice, especially in fast-fading channels. The estimated channel response during two-step equalization, hence, could be obsolete and inaccurate when demodulating some symbols in a frame. Commodity wireless solves this problem by continuously tracking the channel through coherent signal detection in specialized pilot subcarriers, which are however, infeasible for a side channel user that only operates few subcarriers.

Instead, EasyPass uses block-type pilots [87, 89], which reserve one OFDM symbol every  $K$  symbols to deliver the pilot tones in the time domain.  $H_S$ , in this case, will be updated as the distortion of pilot tones that continuously tracks the fading channel. We choose

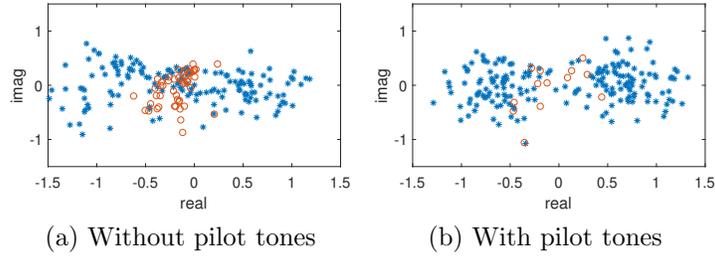


Figure 3.3: Side channel signals after the 2nd equalization. (‘o’ indicates demodulation error)

$K = 10$  to balance the tradeoff between accuracy and overhead (data throughput loss) of such tracking, and Figure 3.3(b) demonstrates that the majority of demodulation error can be successfully eliminated.

Another possible source of demodulation error comes from the possible mis-detection of the symbol boundary in a side channel frame. Current techniques detect such boundary via frequency-domain correlation over LTS, but fail when being applied to the side channel accessed by multiple users: LTSs from different users will collide if each of them spans all OFDM subcarriers, but provide insufficient frequency diversity of precise detection if being transmitted over only few subcarriers. Our experiment results in Figure 3.4 show that a LTS needs to be transmitted over at least  $N_{LTS} = 3$  subcarriers to restrain the error of symbol boundary under 8 samples, which is half of the cyclic prefix that provides extra robustness

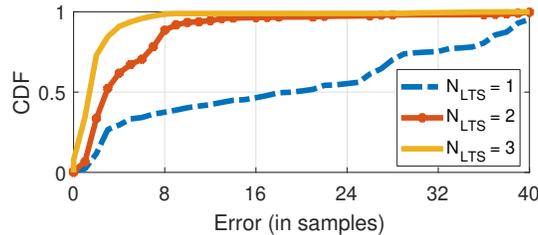


Figure 3.4: Error of symbol alignment under 2.5 dB main channel SNR

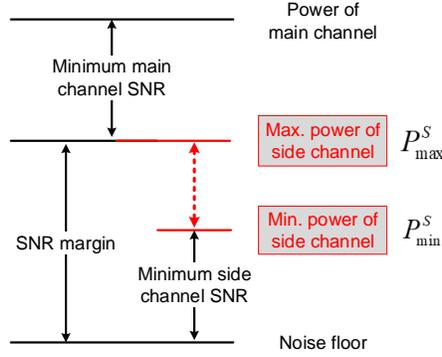


Figure 3.5: Power bounds of side channel

against symbol misalignment. As a result, EasyPass allocates a minimum of 3 subcarriers to each side channel user, and supports a maximum of 13 users by applying the side channel to the available 39 data subcarriers in the main channel<sup>3</sup>. *Fine-grained Spectrum Allocation* of these subcarriers to different side channel users in a distributed manner will be described in Section 3.4.

### 3.3 TRANSMIT POWER CONTROL

The key to maximize the side channel performance is to appropriately control its transmit power that decides the amount of extra interference being applied to the main channel. As shown in Figure 3.5, the maximum transmit power of the side channel ( $P_{\max}^S$ ) has to be controlled within the main channel SNR margin ( $\overline{SNR}^M$ ) to minimize its impact on the main channel. On the other hand, the minimum transmit power *per subcarrier* in the side channel ( $P_{\min}^S$ ) should be sufficient for the side channel receiver to correctly demodulate data. The number of subcarriers that the side channel can operate, hence, is decided as  $\lfloor P_{\max}^S / P_{\min}^S \rfloor$ .

<sup>3</sup>Among the 48 data subcarriers, 8 data subcarriers next to pilot subcarriers are reserved for correct main channel equalization, and one subcarrier is used as the control plane in the side channel.

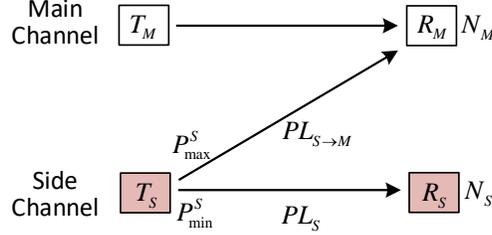


Figure 3.6: Adaptive transmit power control

### 3.3.1 Determine $P_{\max}^S$

As shown in Figure 3.6, the side channel’s received power ( $P_{\max}^S - PL_{S \rightarrow M}$ ) should equal to the minimum SNR ( $\overline{SNR}^M + N_M$ ) required by the main channel Rx ( $R_M$ ), where  $PL_{S \rightarrow M}$  indicates the path loss from  $T_S$  to  $R_M$ , and  $N_M$  indicates the noise floor at  $R_M$ . Therefore,  $P_{\max}^S$  could be calculated as

$$P_{\max}^S = PL_{S \rightarrow M} + \overline{SNR}^M + N_M \quad (3.2)$$

One straightforward solution to obtain  $P_{\max}^S$  is that  $R_M$  actively reports  $\overline{SNR}^M$  and  $N_M$  to the side channel, but requires hardware modification on the commodity wireless devices that operates the main channel. Instead, we indirectly estimate the quantities required in Eq. (3.2) at the side channel users, which actively probe the main channel characteristics using commodity protocols.

#### 3.3.1.1 Estimating $PL_{S \rightarrow M}$

EasyPass leverages the power control mechanisms in existing wireless networks (e.g., Transmit Power Control in 802.11 [78, 79] and Loop Power Control in LTE [92, 35]), which allow a wireless sender to select the most appropriate level of transmit power based on the link path loss reported from the receiver. In specific, the side channel Tx sends a power control (PC) request frame to the main channel Rx, which

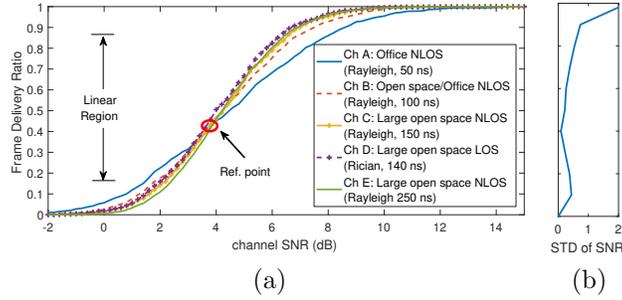


Figure 3.7: FDR vs. SNR. Timing (in nanosecond) indicates the r.m.s delay spread of the tapped delay.

responds with a PC response frame containing its transmit power for sending this frame.  $PL_{S \rightarrow M}$  hence can be calculated as the difference between the transmit and receive power of PC response frame.

**3.3.1.2 Estimating  $N_M$**  Estimating the noise floor ( $N_M$ ) at the main channel Rx from the side channel is much harder, because the side channel has limited knowledge about the main channel's propagation. In EasyPass, the side channel Tx probes the main channel Rx by sending multiple PC requests with increasing levels of transmit power  $P^S$ , and calculate the Frame Delivery Ratio (FDR) of the corresponding PC responses to estimate the receiving SNR at the main channel Rx as  $f(FDR)$ .  $N_M$  can then be decided as

$$N_M = P^S - PL_{S \rightarrow M} - f(FDR) \quad (3.3)$$

In theory, FDR always increases when channel SNR improves. To precisely correlate between FDR and the corresponding channel SNR, we tested the FDR under different channel SNR with HIPERLAN/2 fading channel models [23, 25], which provide models for typical indoor IoT environments such as office, home, and large buildings. The multipath fading is modeled with tapped delay line models that emulate multiple echoes from the transmitted signals.

According to the results in Figure 3.7(a), each channel model requires a different value of SNR in order to reach a certain FDR, with a variance up to 2 dB when FDR is higher than 85%. The standard deviation of such variance is shown in Figure 3.7(b), where the minimum (0.09 dB) is reached when FDR is 0.4. In other words, given  $P^S$  that yields an FDR of 0.4, the corresponding channel SNR (3.8 dB) is nearly constant for all the channel models. Hence, we use  $(SNR_{ref}, FDR_{ref}) = (3.8, 0.4)$  as the reference mapping point:  $P^S$  is adjusted until the FDR reaches  $FDR_{ref}$ .  $N_M$  can then be calculated as

$$N_M = P_{FDR_{ref}}^S - PL_{S \rightarrow M} - SNR_{ref}, \quad (3.4)$$

where,  $P_{FDR_{ref}}^S$  is the side channel transmit power to reach  $FDR_{ref}$  at the main channel Rx. Note that  $N_M$  is usually constant and hence only needs to be estimated once.

### 3.3.2 Determine $P_{min}^S$

$P_{min}^S$  can be determined in a similar way as described in Section 3.3.1, but requires less operations by piggybacking the required information about link characteristics in the side channel Rx's ACK frames back to the side channel Tx. Then,  $P_{min}^S$  can be calculated as

$$P_{min}^S = PL_S + SNR_{min}^S + N_S, \quad (3.5)$$

where  $SNR_{min}^S$  indicates the minimum side channel SNR required that can be decided by the side channel Rx according to the error rate of data demodulation.

## 3.4 DISTRIBUTED SUBCARRIER ALLOCATION

Among the 48 data subcarriers, EasyPass reserves the 8 subcarriers next to pilot subcarriers to ensure correct main channel equalization, and use one subcarrier as the control plane. It hence uses 39 subcarriers to support multiple access from a maximum of 13 side channel users, each of which is assigned to at least one group of 3 consecutive subcarriers. In this way, EasyPass allows the main channel to efficiently eliminate concentrated bit errors from these interfered subcarriers via interleaver shuffling across the frequency band [1].

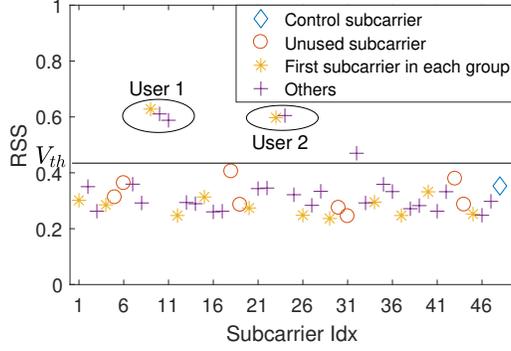


Figure 3.8: Subcarrier occupancy sensing.  $V_{th} = 0.45$  when  $\bar{V} = 0.32$  and  $\alpha = 1.4$ .

Being different from centralized resource allocation in cellular networks (e.g., OFDMA in LTE downlink) [105, 88], EasyPass decides the subcarriers to be used by different side channel users in a fully distributed manner, and aims to allocate all the available spectrum resources to users with maximum fairness, i.e., each user is assigned to  $\lfloor 13/N \rfloor$  subcarrier groups where  $N$  is the current population in the side channel.

### 3.4.1 Subcarrier Occupancy Sensing

EasyPass decides whether a subcarrier is being used by a side channel user by measuring its level of energy after two-step equalization. Figure 3.8 plots the energy level of 48 data subcarriers, and demonstrates that the subcarriers being used by side channel users have significantly higher levels of energy. In practice, we use a RSS threshold  $V_{th} = \alpha \bar{V}$  to decide whether a subcarrier is occupied, where  $\bar{V}$  denotes the average power of the 8 unused data subcarriers (the ones next to pilot subcarriers) and  $\alpha$  is the scaling factor that varies with the channel condition. For example, we set  $\alpha = 1.4$  to ensure  $> 95\%$  detection ratio of subcarrier occupancy, when the side channel SNR is 3 dB.

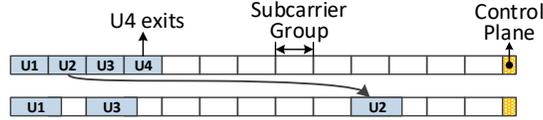


Figure 3.9: Subcarrier reallocation when User 4 exits. User 2 is reallocated since both of its neighboring groups are occupied.

### 3.4.2 Subcarrier Allocation

Built on accurate subcarrier occupancy sensing, side channel users independently decide their subcarrier occupancy at run-time. When a newcomer accesses the side channel, it announces itself by transmitting in the control subcarrier, so that all users are aware about the update of  $N$  and autonomously reduce their occupancy of subcarrier groups by  $\lfloor 13/N \rfloor - \lfloor 13/(N+1) \rfloor$ .

In addition, side channel users keep monitoring the occupancy of each subcarrier group, in order to detect completions of data transmission. In these cases, the spare subcarrier groups will be reallocated to other uses for maximum spectrum utilization. For example in Figure 3.9, when U4 exits, both U1 and U2 cannot expand due to the subcarrier collision. To address this problem, each user calculates the number of its neighboring groups that are occupied, and adjusts their utilization to the vacant subcarriers if both its neighboring groups are occupied.

To evaluate the effectiveness of such allocation scheme, we conducted preliminary experiments using 3 side channel users with dynamic traffic patterns, and then collect the real-time subcarrier occupancy from the side channel Rx. Results in Figure 3.10 show that users initiate the link by accessing the control plane, such as User 3 does at frame 8. All users can successively detect and corresponds by reducing their subcarrier utilization (Frame 9). When the User 1 remains inactive for a certain period (5 seconds), User 2 and 3 would enlarge their resources (Frame 11) to fully utilize the channel capacity. User 1 needs to re-establish the link afterward (Frame 13) in order to access the side channel. These results demonstrate

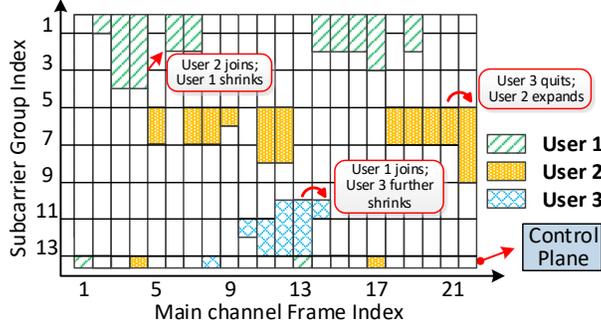


Figure 3.10: Flotations of subcarrier allocation

that EasyPass can promptly adapt to the dynamic traffic patterns in IoT applications with high utilization to the available side channel spectrum: the overall throughput of the side channel could reach 2.25 Mbps (18 subcarriers) at Frame 12.

### 3.4.3 Addressing Conflicts

When multiple users access the side channel simultaneously, it is likely that a subcarrier is accessed by more than one user due to incorrect subcarrier occupancy sensing or the hidden terminal problem, resulting in continuous frame conflict and transmission error. To address such conflict, the side channel users keep track of the occupation of all subcarrier groups, and maintain a list of consecutive vacant groups. A weight  $w_n = 2^n$  is then assigned to each of the  $n$  vacant groups. Upon detecting conflicts from successive data transmission failures, the user would randomly reselect its occupancy from the list according to their weights, hence preventing conflicting users from jumping to the same group again.

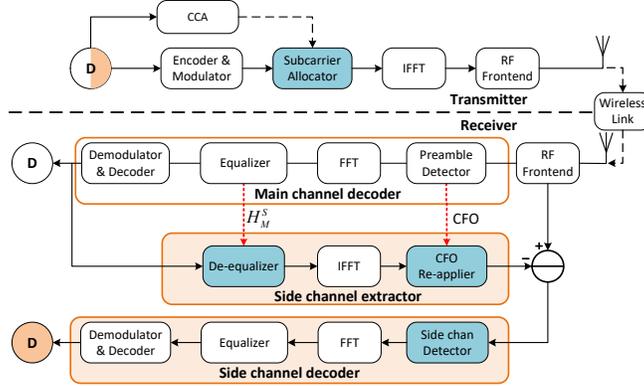


Figure 3.11: PHY implementation of *EasyPass*. The newly added components are in blue.

### 3.5 IMPLEMENTATION

We implemented EasyPass over USRP SDR with the GNURadio toolkit, which realizes a WiFi transceiver over the 5GHz frequency band. EasyPass only modifies the side channel transceivers but retains the main channel transceivers intact.

#### 3.5.1 PHY Implementation

The architecture of the PHY system implementation is shown in Figure 3.11, which is based on the WiFi PHY architecture [13] with the newly added or modified modules highlighted in color, which are mainly related to the two-step equalization procedure to be performed in the receiver.

At the transmitter, we modify the subcarrier allocator module, which assembles OFDM symbols by filling the data into corresponding subcarriers. WiFi utilizes all 48 subcarriers for data transmission and additional 4 for transmitting pilots. *EasyPass*, however, only transmits at a subset of data subcarriers and will not transmit pilots. Therefore, modified subcarrier allocator module allocates subcarriers based on the channel selected, which is decided by the channel idleness upon Clear Channel Assessment (CCA).

At the receiver, the key challenge is to estimate and remove the arrival signals of the main channel from the total received signals, which are decided by 1) the estimation of main channel Tx signals, 2) the channel distortion, and 3) Carrier Frequency Offset (CFO) due to the mismatch of crystal oscillator between side channel Rx and main channel Tx. In order to obtain these, we pass the received signals into regular WiFi decoding procedure, and store the estimated channel response and CFO for later use. Afterward, we recover the main channel Tx signals by passing the decoded bits into regular WiFi encoding procedure. At meanwhile, the channel distortion and CFO are re-applied through De-equalizer and CFO re-applier modules, respectively. The residual signals of the side channel can then be subtracted from received signals. Finally, we use side channel detector to detect the arrival of the side channel signals, and then decode them upon the presence of the side channel.

### 3.5.2 MAC-Layer Implementation

We implemented a complete WiFi MAC layer with functionalities such as carrier sensing and retransmission [38]. However, since the RF signals harvested by USRP have to be sent to and processed at the host PC, a long delay is incurred at signal transmission and processing. It is hence difficult to meet the timing requirements of the Distributed Coordination Function (DCF) in WiFi. To address this challenge, we follow the same method in [38] and use a scaling factor  $\beta (= 1000)$  to emulate the real WiFi MAC.

Adaptive Auto Rate Fallback (AARF) [56] rate adaptation algorithm is integrated into our implementation to emulate the realistic behaviors of commodity wireless devices in practical scenarios, which is required for the main channel operations. Specifically, AARF increases the link rate after a given number ( $T$ ) of consecutive successful frame transmissions, and switches to a low data rate if  $T/2$  consecutive transmissions fail. If the first transmission at a higher rate fails, it returns to the prior data rate and updates  $T$  [56, 12].

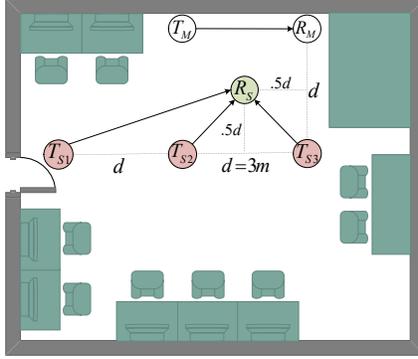


Figure 3.12: Experiment scenario

### 3.6 PERFORMANCE EVALUATION

Built on our implementation in Section 3.5, we evaluate the performance of *EasyPass* by comparing it with existing WiFi.

#### 3.6.1 Experiment Setup

We conduct our experiments in a  $9\text{m} \times 9\text{m}$  open space office using six USRP N210 motherboards with UBX-40 RF daughterboards. As shown in Figure 3.12, the main channel Tx continuously transmits frames with known payloads of 1000 bytes to the main channel Rx,

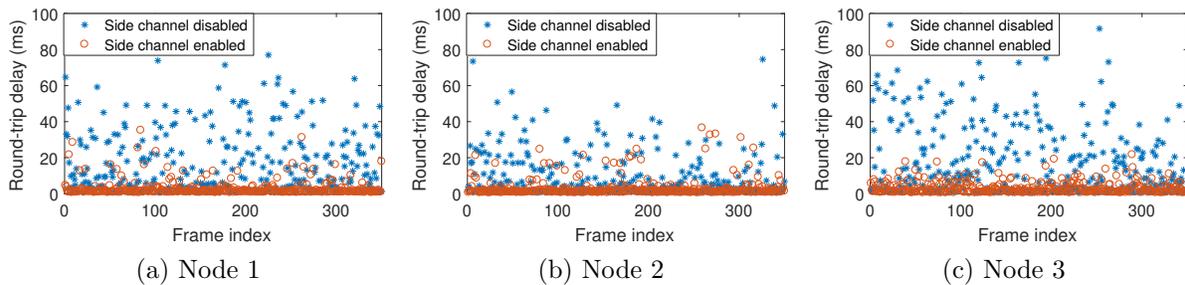


Figure 3.13: Wireless transmission delay

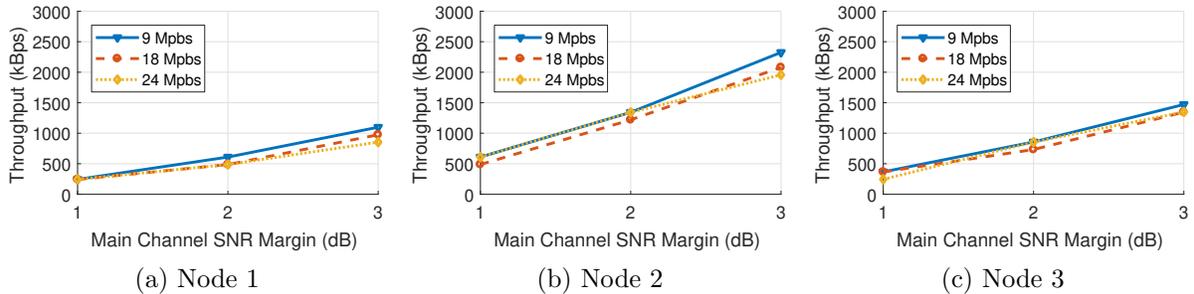


Figure 3.14: Side channel throughput

which decodes and measures the BER of each frame. In order to emulate the state of a busy main channel, we make a modification on the MAC behavior on the main channel Tx, such that it does not go through the backoff procedure and hence has greedy access to the channel. We tune the main channel SNR by adjusting the USRP Tx Gain, which ranges between 0 dB and 31.5 dB at the step of 0.5 dB, and the maximum Tx power is equivalent to 13 dBm.

We use three USRPs as side channel transmitters, which all transmit to one side channel Rx. Noted that although the side channel Rx is the shared receiver of the side channel nodes, it emulates three separate receivers with respect to each side channel transmitter by separating the information, such as noise estimation and subcarrier occupancy from each other. Thereby, the resource of the side channel is allocated in a distributed manner, as illustrated in Section 3.4. Every three USRPs are connected to a Dell desktop PC, which is equipped with an 8-core Intel Xeon CPU 3.5 GHz and 16 GB memory and runs OS of Ubuntu 16.04 LTS. The communication and computation overheads of transmission are then measured at the host PCs of both sender and receiver. In this section, we disable the rate adaptation and instead use constant data rate during each experiment.

We tested this experiment setup with the power control schemes being proposed in Section 4. As shown in Table 3.1, the three side channel transmitters exhibit different levels of path loss that result in dynamic levels of transmit power. An error of  $\leq 1$  dB is ensured by all three transmitters when estimating  $N_M$ . Note that although User 3 is closer to the

Table 3.1: Path loss and  $N_M$  estimation

Node	$PL_{S \rightarrow M}$ (dB)	$P_{\max}^S$ (dBm)	$FDR_1$	$P_{\max}^S$ (dBm)	$FDR_2$	$N_M$ (dBm)
1	34.5	-21.5	0.72	-25	0.30	-62.5
2	28.5	-27.5	0.66	-30.5	0.46	-63.7
3	33.0	-25.5	0.58	-27.5	0.30	-63.6

Table 3.2: Path loss of the side channel links

Link	$PL_{S_1}$	$PL_{S_2}$	$PL_{S_3}$
Path loss (dB)	31.5	22.5	29.0

main channel Rx than User 2, it suffers more path loss due to the imperfection of the omnidirectional property of antennas. The path loss of the side channel links are listed in Table 3.2.

### 3.6.2 Network Delay

In this experiment, we evaluate the performance of *EasyPass* in reducing the data transmission delay. The main channel Tx continuously transmits at 12 Mbps with a 3 dB SNR margin. To emulate a congested network, we made a minor revision on the main channel's contention window to  $[0,1]$  so that it can greedily occupy the channel and defer the transmission of other devices. Each side channel device transmits a total of 700 frames: the first 350 frames are transmitted without side channel, i.e., when the channel is occupied, the device defers its transmission, sets up a random timer and re-senses the channel after the timer elapses. The backoff timer will increase if the channel is still busy upon re-sensing, which further deteriorate the network latency. The functionality of the side channel would be enabled after transmitting these 350 frames, which bypasses the backoff procedure and transmits under the busy channel. Figure 3.13 plots the round-trip delay of the side channel

devices which compares the first and second half of the frames, and demonstrates that Easy-Pass can achieve >90% delay reduction over a single congested wireless link. Meanwhile, it avoids overlong network delay (> 0.5s) due to the failure of multiple carrier sensing attempts, and hence significantly alleviates delay jitter.

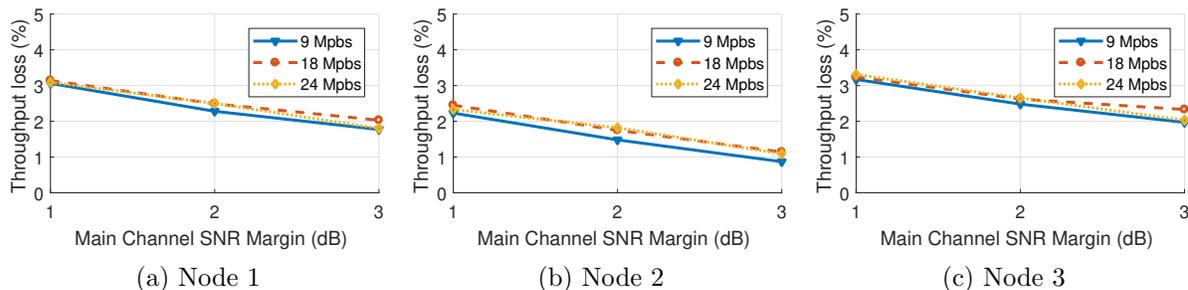


Figure 3.15: Main channel throughput loss

### 3.6.3 Side channel throughput

Based on the noise floor and path loss estimation, we evaluate the main channel degradation and the side channel detection rate under different SNR scenarios. In specific, the main channel transmits in 9, 18 and 24 Mbps. We tune the tx gain of the main channel Tx such that 1% main channel BER is measured at the main channel Rx and add extra SNR margin to the main channel. The side channel Tx is configured to exploit the whole SNR margin. The side channel throughput, as shown in Figure 3.14 will increase as SNR margin increases and could achieve  $\geq 2.5$  Mbps for Node 2 upon 3 dB SNR margin. The main channel data rate also affects both channels' performance. For example, side channel incurs more BER when the main channel operates at 18 Mbps than 9 Mbps since 18 Mbps uses the same code rate but denser constellation diagram which is prone to the interference than the 9 Mbps. The performance of using 24 Mbps is close or even surpass the 18 Mbps case since it uses a lower code rate which could correct more symbol errors due to the interference. Table 3.3 lists the bounds of the side channel power and the number of subcarriers the side channel can utilize when the main channel operates at 9 Mbps. As SNR margin increases, the upper

Table 3.3: Side channel throughput with different amounts of main channel SNR margin being exploited

User	$\overline{SNR}^M$ (dB)	$P_{\max}^S$ (dBm)	$P_{\min}^S$ (dBm)	$N$	Data rate (Mbps)
User 1	1	-40	-44	2	0.25
	2	-36		5	0.625
	3	-34		9	1.125
User 2	1	-47	-54	5	0.625
	2	-43		11	1.375
	3	-41		19	2.375
User 3	1	-42	-47	3	0.375
	2	-39		7	0.875
	3	-36		12	1.5

bound of the side channel power  $P_{\max}^S$  increases, yielding a maximum data rate of 2.375 Mbps by interfering 19 subcarriers. In practice, the wireless link generally operates in  $> 3$  dB SNR margin, as shown in Figure 1.2. Therefore, a throughput of  $\geq 1$  Mbps is often guaranteed.

The main channel throughput loss is shown in Figure 3.15, showing that the existence of the side channel can cause at most 3% throughput loss when the main channel transmits at 24 Mbps with 1 dB SNR margin. As SNR margin improves, such degradation decreases and will achieve nearly  $\leq 2\%$  kbps upon 3 dB SNR margin. In addition, higher main channel data rate yields larger degradation as a result of denser constellation diagram which is susceptible to interference. It is expected that exploiting partial rather than the complete SNR margin will reduce the main channel throughput loss at the cost of reduced side channel throughput.

Table 3.4: Tx computation overhead. Side channel using the entire (48) subcarriers is equivalent to the 6 Mbps case in the main channel.

No. of subcarriers	3	6	9	12	48
Tx time ( $\mu s$ )	34.1	31.15	30.01	29.35	28.42

### 3.6.4 Computational Overhead

We evaluate the computational overhead of EasyPass by measuring the runtime of the signal processing blocks over the PC. Table 3.4 lists the Tx computational time for generating a single 100-byte frame, showing that the Tx processing time increases when fewer subcarriers are being used. Without modification, the modulation and coding scheme of the existing WiFi (6 Mbps) is reused to the payload, whose computational overhead is only decided by the payload size. The successive subcarrier allocator fills the modulated signals onto the specific subcarriers of each symbol. However, different from the main channel which utilizes the entire subcarriers, side channel Tx only uses a small portion of subcarriers and hence requires more OFDM symbols in order to carry the same amount of the data, which requires more IFFT operations to convert each symbol into the time domain and hence incurs more overhead.

Table 3.5 lists the Rx computation overhead assuming the main channel utilizes the same number of OFDM symbols as the side channel. Decoding and removing the main channel at the side channel Rx is only decided by the frame length and the data rate of the main channel, since the traditional equalization and decoding schemes will be applied to the mixed signals. After removing the side channel, the residual signals will be demodulated and decoded using the built-in blocks. The processing time is proportional to the size of the side channel payload that is decided by the number of subcarriers it exploits. The experimental results prove such linearity. In general, the computational cost of operating a side channel is lower than operating the main channel in the same situation.

Table 3.5: Rx computation overhead ( $\mu s$ )

Main channel data rate	No. of subcarriers			Main channel overhead
	3	6	9	
6 Mbps	653	852	1100	464
12 Mbps	839	1010	1259	632

### 3.7 REAL-WORLD EXPERIMENTATION

In this section, we further apply EasyPass to real-world network scenarios with dynamic channel conditions and commodity rate adaptation approaches. A different number of wireless devices is being used as the side channel users to evaluate the multiple accessibility provided by EasyPass.

#### 3.7.1 Subcarrier Allocation

We evaluate the effectiveness of subcarrier allocation algorithm in Section 3.4 by simulating a network comprised of three side channel users, whose capacity  $N$ , i.e., the maximum number of subcarriers to interfere, are 13, 12, and 19, respectively. Side channel users decide their channel occupancy in a distributed manner by estimating the number of neighboring nodes that are sharing the spectrum. To achieve this, they continuously monitor the neighboring users including their active/inactive status, subcarrier occupancy and last transmission time. A user without transmitting for timeout ( $To$ ) seconds will be considered as inactive, and its subcarriers will be returned to the remaining users. Figure 3.16 plots the total number of subcarriers for side channel transmission under different timeout  $To$  parameters. When  $To = 1s$ , the total number of side channel subcarriers varies frequently since all users continuously adapt their occupancy due to the others' constant quitting and rejoining. As  $To$  increases, users only return its allocated resources after a longer period of inactivity, resulting in less frequent subcarrier adaptations. In this case, a single user is unlikely to preoccupy the

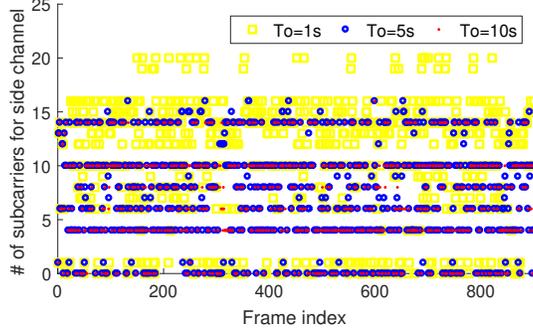


Figure 3.16: Number of subcarriers for side channel transmission using different timeout ( $To$ ) values

entire resources, since the resources are often reserved for the remaining inactive users. As a contrary, when  $To = 1s$ ,  $\geq 19$  subcarriers could be utilized at the same time when user 3 preoccupies the entire resources.

We further use three metrics to evaluate the performance, they are 1) utilization rate; 2) conflict rate and 3) actual throughput, which are shown in Figure 3.17(a). The utilization rate is modeled as  $\sum_i n_i/N_i$ , where  $n_i$  and  $N_i$  are the numbers of subcarriers for instantaneous transmission and the user's capacity. Higher utilization rate is desired ensuring the full exploitation of the side channel spectrum. Figure 3.17(a) shows that although smaller timeout yields higher utilization rate. The actual throughput, however, does not benefit due to its high conflict rate, which is a result of the frequency subcarrier adaptation. In real scenarios,  $To = 5s$  is used in order to achieve the best throughput, which can be further improved by taking advantage of the traffic pattern. This will be part of our future work. Furthermore, we evaluate the scalability w.s.t to the network size of the side channel. The system characteristics under different number of side channel users are listed in Figure 3.17(b). It shows that as network size increases, more conflicts are expected resulting in degraded throughput. The utilization rate, however, would remain around 40% or higher in all cases.

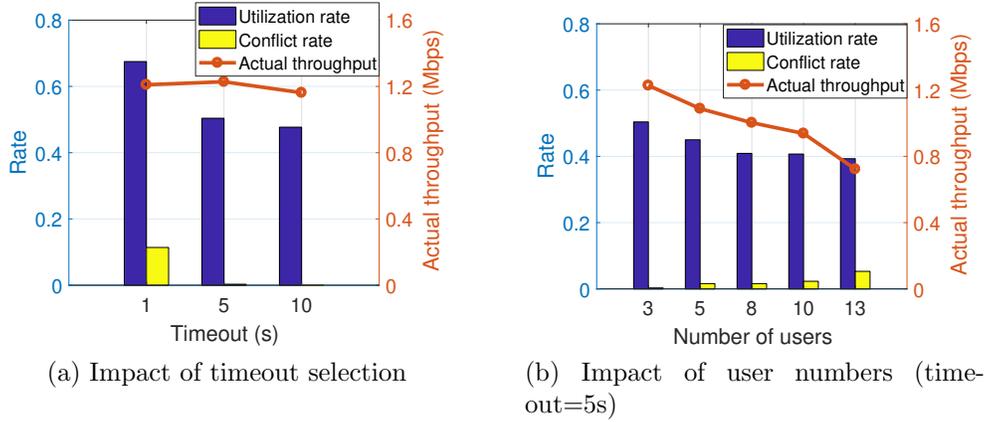
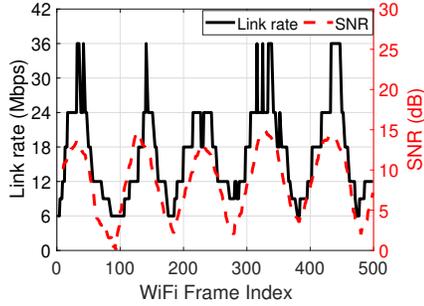


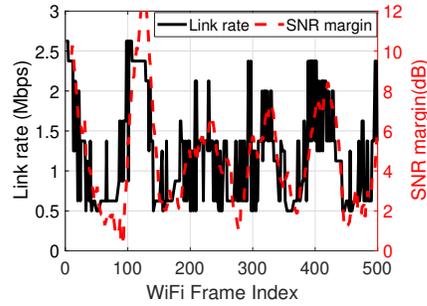
Figure 3.17: Parameters of subcarrier allocation

### 3.7.2 Highly Volatile Conditions

In this section, we configure the Tx gain of the main channel for sending the  $N$ -th data frame as  $12(1 + 6\sin(2\pi\omega N))$ , where  $\omega = 0.01$  (i.e. the period of the Tx gain change is 100 frames) and 12 is the average level of the Tx gain being configured between 6dB and 18dB. As shown in Figure 3.18(a), when the Tx gain of the USRP sender is periodically fluctuating, the channel SNR varies accordingly and leads to different link rates achieved in the wireless network over time. The highest throughput of the side channel is achieved when the main channel operates in low data rates, especially between 6 ~ 12 Mbps. When the main channel data rate reaches 36 Mbps, the side channel can only provide nearly 0.5 Mbps data rate. The reasons are two-fold: first, as mentioned in Section 3.6.3, higher data rates in the main channel use denser constellation, which is susceptible to the ambient interference; second, the USRP can only provide limited SNR margin at such high data rate due to the hardware limitation [14]. The variation of link rates of Node 2 is shown in Figure 3.18(b), showing its strong correlation with the main channel SNR margin.



(a) Varying channel conditions of existing WiFi



(b) Side channel link rate vs. SNR margin

Figure 3.18: Varying channel conditions and link rates of both the main and side channel (Node 2)

Based on this setup, we test the main channel’s performance and the side channel delay using different frequency of PC frames. Table 3.6 shows that main channel BER would decrease as the frequency of PC request frames increases, which keeps track of the channel variations. On the other hand, however, more PC request frames would increase the burden of the side channel link, yielding more delay.

### 3.8 RELATED WORK

**Side Channel Design.** Prior designs of the in-band side channel are mainly applied for delivering coordination and control information, so as to improve the network performance without introducing extra costs compared to the out-of-band approaches. Most schemes hide information into the side channel for private communication, by alternating the physical characteristics of the main channel such as signal duration [17], the position of interfered chips [108], timing of periodic beacon frames [50], intentionally corrupted checksums [72] and the position of the blank subcarriers [69]. These side channel designs can only provide limited throughputs and do not support multi-access operations due to their dependency with the main channel. hJam [107] exploits the preamble redundancy in OFDM, and detects side

Table 3.6: Using different frequencies of PC requests

# of PC requests per 10s		0	1	5	10
Main channel BER		1.1	2.0	1.8	1.3
Delay	N1	19.6	8.9	5.3	6.6
	N2	14.0	6.8	8.0	9.4
	N3	21.8	5.2	4.7	5.9
Delay jitter	N1	16.1	5.7	6.0	7.0
	N2	11.9	6.0	6.0	7.0
	N3	19.9	5.9	7.0	7.1

channel signals using those over blank subcarriers in frame preamble as the reference. The main channel data is recovered through interference cancellation, which however, requires firmware modifications of main channel receivers.

**Interference Cancellation.** Existing researches on full duplex communication and interference cancellation techniques give insights to our work but are different from our proposed approach. Full duplex radio [20, 81, 10, 26, 47] allows simultaneous transmit and receive by estimating the phase and amplitude of its transmitted signals at the receiver antenna, and subtracting it from the receiving antenna receives. The full-duplex design, however, is not suitable for our proposed work for two reasons. First, in our work, the receiver of the extra channel aims at extracting the useful information from the main channel signals, rather than the locally transmitted signals which are known to the receiver itself. Second, the implementation of full-duplex usually requires modifications on the RF front end, and hence doesn't meet our objective. Zigzag decoding [33] utilizes interference cancellation technique to recover information from collided frames. It exploits the asynchrony of successive collisions caused by the hidden terminal. These successive transmissions of both channels, though, is unavailable in our work.

**Subcarrier Allocation.** Subcarrier allocation has been widely studied to optimize system metrics, such as power efficiency [105] and channel throughput [88], among which almost all require centralized control. [51] suggests using average SNR for each user to decide the total number of subcarriers to be assigned. SNOW [82, 83] allows single-hop communication between the base station and sensors in sensor networks, by utilizing the untapped TV whitespace spectrum. Sensors are assigned non-overlapping narrowband orthogonal subcarriers to avoid collisions, which is centrally controlled by BS. BS also controls the timing of data transmission in a TDMA fashion. Different from our work, each subcarrier of the TV whitespace spectrum is occupied only by SNOW nodes. Interference control is hence not required.

### 3.9 CHAPTER SUMMARY

In this chapter, we present EasyPass that minimizes wireless latency from multiple IoT devices over a congested link. These users concurrently access a wireless side channel and adapt channel occupancy in a distributed manner without impacting the main channel. We implemented and evaluated our design over practical SDR platforms, and demonstrate more than 90% delay reduction over up to 13 IoT devices.

## 4.0 CONTINUOUS WIRELESS LINK RATES FOR IOT

### 4.1 INTRODUCTION

Internet of Things (IoT) computerizes physical objects and integrates them into a networked ecosystem [62], producing an unprecedented amount of data about humans and their surrounding environment. It hence has stringent requirements on the wireless throughput to timely transmit such big data. To meet such demanding requirements, an increasing amount of wireless spectrum has been exploited. For example, higher frequency bands at 5 GHz are exploited to expand the bandwidth of a WiFi channel from 20 MHz to 160 MHz [2] and increase the highest link rate to  $> 500$  Mbps. They also allow multiple data links to co-exist through MIMO [95].

Unfortunately, current wireless networks have limited efficiency in utilizing these spectrum resources under dynamic wireless conditions in IoT. Particularly in future smart cities, IoTs being deployed at large public facilities, such as massive transportation systems, shopping malls and museums, may experience serious degradation of wireless signal quality due to the long communication distance or complicated indoor layout. Wireless networks in these scenarios, hence, cannot operate at the highest link rate and instead have to adapt to lower link rates according to the instantaneous channel condition [56, 106, 101, 85]. Such link rate adaptation, however, fails to fully utilize the wireless spectrum due to the *discrete* choices of link rates and the *gap* between these rates: a wireless link operates at the same data rate before the channel quality reaches the requirement of the next higher rate. The network throughput during the meantime, then, remains constant and may be much lower than the momentary channel capacity. For example, 802.11a provides only 8 link rates and suggests to stay at 24 Mbps (16-QAM 1/2) as long as the channel SNR is between 17.7 dB

and 21.7 dB [98]. New generations of wireless standards such as 802.11ac, while enabling more link rates with higher orders of channel modulation schemes (e.g., 256-QAM) and code rate options (e.g., 5/6 and 7/8), further enlarge the gap between these link rates to >5 dB [75].

To address this challenge, existing research designs wireless networks to be rateless, i.e., the link rate continuously varies based on the channel condition without requiring channel state feedback. For example, rateless codes allow the sender to keep transmitting encoded data chunks at the highest rate, and the actual link rate is determined by the number of data chunks being transmitted until successful decoding of the entire data message [90, 36, 76]. However, rateless codes improve throughput at the cost of extra communication and computation overhead, which prevent them from being practically applied in resource-constrained IoT. First, a rateless sender consumes more energy to send data, because its wireless radio always stays at the high-power Tx mode [46]. Especially when the wireless link quality degrades, more data chunks are transmitted to deliver the same amount of data, even if many data chunks being transmitted are corrupt and wasted<sup>1</sup>. Second, rateless codes are computationally expensive, because the receiver has to continuously decode incoming data chunks. Even when prototyped over high-end processors with abundant computational capacity, they can only support a throughput up to 12.5 Mbps [45].

In this chapter, we envision that a better solution towards maximum wireless throughput in IoT, instead of regardlessly sending and decoding rateless data chunks, is to transform the available choices for link rate adaptation from *discrete* to *continuous*, so that the link rate being chosen is always optimal for any channel condition. We present **vMod** (VLC-based Modulation), a novel modulation technique that realizes such continuous wireless link rates for IoT. Its design has the following important characteristics:

- vMod’s modulation is *continuous*. It can modulate an arbitrarily fractional number of data bits into a symbol, while preserving every individual symbol to be independent from others. It can provide any link rate that is supported by the channel condition, and hence maximizes the network throughput under dynamic wireless channel conditions.

---

<sup>1</sup>Ordinary wireless senders switch their radios to low-power Rx mode after each transmission and wait for channel feedback. They hence consume less energy to send data.

- vMod is *lightweight*. vMod does not produce any redundant data transmission, and hence retains the power efficiency of current wireless networks. Besides, vMod's demodulation over each symbol is independent and only involves demapping from a constellation point to data bits. It hence incurs 10x less overhead compared to rateless codes.
- vMod is *applicable* to any commodity wireless system with minimum hardware modification. It builds on the existing QAM modulation and works with link rate adaptation in current wireless networks, and hence incurs minimum modification on the PHY hardware. It does not require modification of any other PHY/MAC hardware or software, such as RF frontend, channel encoder/decoder and link ARQ.

How can a fractional number of data bits be modulated into a symbol that contains an integer number of constellation points? Our key insight is to design a Variable-Length Code (VLC) and split the data bitstream into variable-length codewords, which are then mapped to constellation points in symbols. Hence, each symbol randomly carries a variable amount of data bits, and any link rate can be statistically achieved by adjusting the range and constitution of codeword lengths. For example, by splitting the bitstream with a VLC of 4-bit and 5-bit codewords, we can achieve any link rate between 24 Mbps and 30 Mbps (with a  $1/2$  code rate), by adjusting the ratio between 4-bit and 5-bit codewords in the VLC. Such granularity of continuous link rates could be further improved by combining vMod with the existing variety of code rate options.

We ensure the generality and power efficiency of vMod by exploiting the modules in existing wireless systems whenever possible. First, we build our VLC by extending the fixed-length code being used in existing systems, and construct new constellation diagrams for modulation by mapping the new variable-length codewords to the unused signal positions in existing QAM constellations. In this way, we ensure that any link rate could be achieved by the same generic approach with the minimum wireless transmit power. Second, information about the lengths of VLC codewords carried by symbols is the key to correct demodulation. To convey such information without impairing throughput, we transmit both regularly modulated and VLC-modulated symbols in the same wireless link, so that such information can be indicated by the permutation of how these two types of symbols alternatively appear.

We implemented vMod over USRP with GNURadio toolkit, based on a 5 GHz WiFi transceiver. Minimum modification is conducted over WiFi PHY and MAC. Based on this implementation, we evaluated vMod under dynamic channel conditions, and also compared vMod with existing rateless codes (Strider [36] and Spinal codes [76]). The experiment results show that vMod scales well with the dynamic channel conditions and improves the WiFi throughput by 30% over a single narrowband link, but consumes up to 95% less computation and communication overhead compared to existing rateless codes. vMod, hence, could be easily integrated with other advanced wireless techniques, such as channel bonding and MIMO, to achieve much higher improvement of wireless throughput.

The rest of this chapter is organized as follows. Section 4.2 discusses the limitation of the existing approaches. Section 4.3 gives an overview on vMod. Sections 4.4 and 4.5 present the details of vMod’s modulation and demodulation. Section 4.6 presents our implementation of vMod. Section 4.7 and 4.8 evaluate the performance of vMod over realistic wireless network scenarios. Section 4.9 reviews the related work. Section 4.10 discusses and Section 4.11 concludes the chapter.

## 4.2 LIMITATION OF THE EXISTING APPROACHES

The key to continuous link rates is to carry a fractional number of data bits in each symbol. The most intuitive approach to this objective is to break the independency between symbols and convey  $N$  data bits with  $M$  symbols as a group [73], so that each symbol carries  $N/M$  data bits. However, the key limitation of this approach is that the  $N$  data bits are demodulated at the same time over the  $M$  symbols, and demodulation error of any symbol makes all the  $N$  bits to be wrong. Letting  $p$  denote the probability of symbol demodulation error determined by channel SNR, the probability for  $N$  bits to be correctly demodulated will be  $(1 - p)^M$ , which diminishes when  $p$  or  $M$  increase. This approach, hence, is infeasible in practice due to its susceptibility to demodulation errors.

Another alternative is to apply multiple existing modulation schemes over the time or frequency domains in the same wireless frame. For example, multiple modulation schemes could be applied to different symbols of a frame. By applying QPSK to 25% of symbols in a frame and 16-QAM to other symbols in the same frame, we are able to convey  $2 \times 0.25 + 4 \times 0.75 = 3.5$  bits per symbol on average. However, this approach builds on existing link rates and does not eliminate the gap between these link rates. Hence, it leads to much higher chances of demodulation error. When both QPSK and 16-QAM are used in the same frame, the channel SNR will not be high enough to support 16-QAM and demodulation of the 16-QAM symbols will be certainly prone to channel noise. On the other hand, FARA [80] adopts multiple constellation diagrams over different OFDM subcarriers in a frame by exploiting the frequency diversity of wideband channel. However, the performance of FARA could be easily impaired by inter-carrier interference or strong fading in frequency-selective channels. The large number of OFDM subcarriers also increases the overhead of operating FARA.

Rateless codes are capable of approximating link throughput to the channel capacity, but consume much more power in both communication and computation that is unaffordable by the existing wireless system. In this chapter, we evaluate such energy consumption of rateless codes using the Tx communication overhead and Rx computation overhead as the metrics. First, the Tx communication overhead is measured as the average duration of RF waveforms on the air for sending one frame, which is proportional to the energy consumption of the Tx's RF frontend under constant transmit power. Second, the Rx computational overhead is measured by the average amount of time needed for decoding one frame on the receiver side and reflects the power consumption of the Rx's digital logic. To evaluate such overhead, we implemented two rateless code designs, namely Strider [36] and Spinal codes [76], over the GNURadio/USRP platform based on the open-source library of wireless system simulations<sup>2</sup>. For the Spinal code, we use  $2^{20}$ -QAM as the constellation diagram and 256 bits and the data chunk size. For the Strider code, we use the recommended setup in the chapter, i.e.,  $K = 33$ , and the data chunk size is set to be 9500 bits. At the Tx side, we use the per frame duration on RF as the metric to evaluate the communication overhead, and Figure 4.1(a) shows that

---

<sup>2</sup><http://www.yonch.com/wireless>.

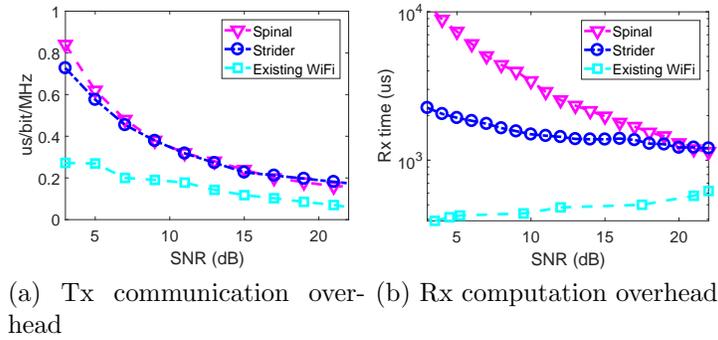


Figure 4.1: Overhead of rateless codes

the transmission overhead at a rateless sender is at least 100% higher than that of existing WiFi. Especially when the channel SNR is lower than 10 dB and corrupts many rateless data chunks being transmitted, such difference of overhead could increase to 400%. At the Rx side, we use the per packet decoding time to evaluate the computation overhead, and Figure 4.1(b) shows that the decoding complexity of rateless code is up to 10x higher than existing WiFi, which requires more powerful chips for real-time decoding. Such high overhead, hence, seriously limits the applicability of rateless codes in practice.

#### 4.2.1 Variable-Length Code

VLC is widely used in source coding, which maps source symbols to a variable number of bits. By assigning shorter codewords to symbols that appear more frequently, a VLC code requires less bits to represent a certain number of symbols, making it a good solution for lossless data compression. In this chapter, we use Huffman code [44] in our vMod design because of the following reasons. First, Huffman code is a classic prefix code with zero redundancy, i.e., no codeword in a VLC is the prefix of another. It hence provides a simple yet reliable solution to VLC codeword construction, by extending the existing fixed-length codewords with negligible computation overhead. Second, by using the Huffman code, vMod requires minimum modification to the existing wireless systems except for channel modulation. Other

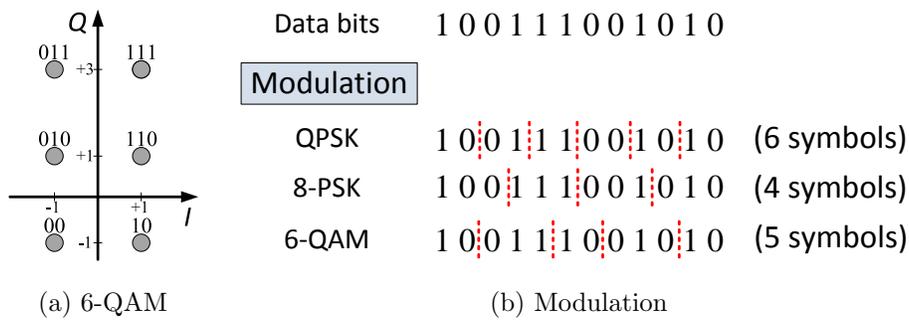


Figure 4.2: VLC-based modulation

advanced coding techniques, such as Variable-Length Error-Correcting (VLEC) code [8, 9] and Reversible Variable-Length Code (RVLC) [97], have to be either used for joint source-channel coding or require PHY redesign.

### 4.3 OVERVIEW

In order to convey a fractional number of bits in each symbol while preserving independence between these symbols, vMod modulates a data codeword with a variable bit length into each symbol and statistically controls the link rate by adjusting the bit lengths of codewords. Figure 4.2 shows an example of how such VLC-based modulation is operated over a 6-QAM constellation diagram, to which a VLC of two 2-bit and four 3-bit codewords is mapped as shown in Figure 4.2(a). Then, as depicted in Figure 4.2(b), the VLC-based modulation uses five 6-QAM symbols to transmit 12 data bits, and the bit lengths of codewords are either 2 or 3. It hence provides a link rate between the ones provided by QPSK and 8-PSK, which require 6 symbols and 4 symbols to transmit the same amount of data bits, respectively. To realize this high-level design, the following technical challenges need to be addressed.

- First, for efficient modulation, the VLC codewords need to be appropriately constructed and mapped to the constellation diagram. Since the number of codewords in the VLC is arbitrary, the number of constellation points on the constellation diagram may not be an integer power of 2 and the constellation diagram hence needs to be redesigned. How to develop a generic approach that constructs and maps the VLC codewords towards any arbitrary link rate, while preserving the code completeness and retaining the wireless transmit power at the same time, is a challenging problem.
- Second, to minimize the amount of demodulation errors, information about the lengths of VLC codewords in each symbol has to be known by the receiver. Otherwise, if the received signal is distorted by channel noise and erroneously demodulated to another codeword with a different length, it will displace all the succeeding data bits and result in cascading failures. For example in Figure 4.2(b), if the first codeword “10” is demodulated as “110”, the next codeword “011” will be placed to the 4th position of bitstream and result in additional errors. The major challenge, then, is how to convey such information without impairing throughput.

vMod addresses the first challenge by expanding the QAM-based constellation diagrams and fixed-length codewords being used in existing wireless systems for VLC-based modulation. For a given VLC with  $m$  codewords, we construct its constellation diagram by adding additional constellation points to an existing QAM constellation diagram with the size smaller than but closet to  $m$ . For example, the 6-QAM constellation diagram in Figure 4.2(a) is constructed from QPSK by adding two additional points. Then, we build our Huffman code by appending bits to some of the existing fixed-length codewords, and map the generated VLC codewords to the additional constellation points being added to the constellation diagram. For example in Figure 4.2(a), the four 3-bit codewords are expanded from QPSK codewords “01” and “11”. In this way, vMod designs new constellation diagrams for different choices of continuous link rates in a generic manner by exploiting the symmetry of QAM, and continuously varies the link rate by adjusting the amount of fixed-length codewords being expanded. More details about such modulation are provided in Section 4.4.

To address the second challenge, vMod does not transmit the information of codeword lengths through frame preambles or data symbols, to avoid any non-negligible throughput loss. Instead, we embed such information into the pattern of how different modulation

methods are used in a symbol. For each symbol modulated by OFDM, we only apply VLC-based modulation to half of its subcarriers and modulate the other half of subcarriers using the regular method in existing systems. Then, the information of VLC codeword lengths being used in this symbol is indicated as the permutation of how these two types of subcarriers alternatively appear in the symbol. During demodulation, this information can be retrieved based on the measurable difference between VLC-based and existing QAM-based constellation diagrams being used. More details about conveying such information are provided in Section 4.5.

## 4.4 VLC-BASED MODULATION

In this section, we present technical details of our proposed VLC-based modulation. We first present the constitution of continuous link rates enabled by the VLC, particularly, the Huffman code. Then, we describe how to realize such continuous link rates by constructing the Huffman code and the corresponding constellation diagram, with an end-to-end example shown in Figure 4.3.

### 4.4.1 Constitution of Continuous Link Rates

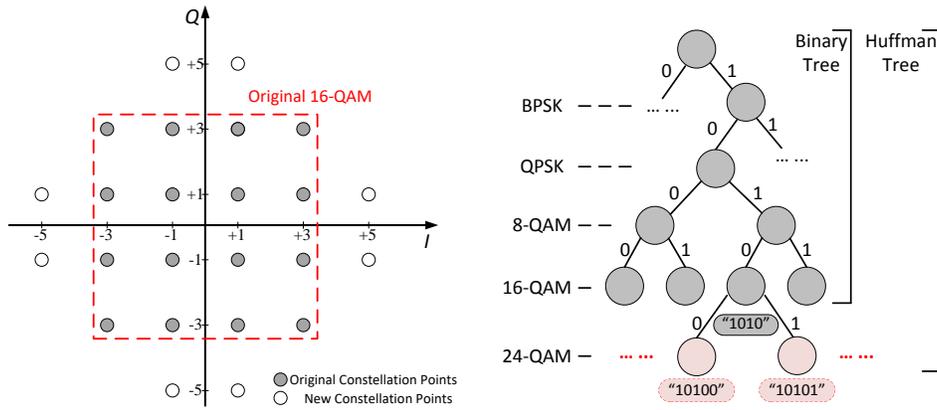
In vMod, we construct a Huffman code and map its codewords to the points in a constellation diagram. A link rate is then constituted by the codewords with different lengths that are actually used in modulation. Assuming that each bit in the input data has the equal probability to be “0” or “1”, the probability for an  $l$ -bit Huffman codeword to be used is  $2^{-l}$ . Then, the average number of data bits being modulated by a constellation symbol is

$$l_{avg} = \sum_{i=1}^M p_i l_i = \sum_{i=1}^M 2^{-l_i} l_i, \quad (4.1)$$

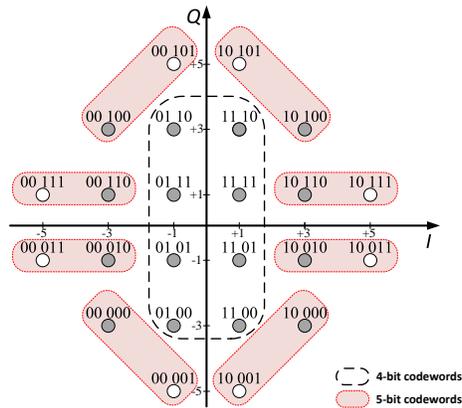
where  $M$  is the number of codewords in the Huffman code, and  $l_i$  and  $p_i$  are the length and probability of the  $i$ -th codeword, respectively. For any Huffman code, we always have  $\sum_{i=1}^M p_i = 1$  to ensure the completeness of the code.

Table 4.1: Continuous link rates with two VLC codeword lengths in the Huffman code

$l_{avg}$	Number of $k$ -bit codewords ( $n_k$ )					Link rate (Mbps)
	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	
2	4	-	-	-	-	12.0
2.25	3	2	-	-	-	13.5
2.5	2	4	-	-	-	15.0
2.75	1	6	-	-	-	16.5
3	-	8	-	-	-	18.0
3.25	-	6	4	-	-	19.5
3.5	-	4	8	-	-	21.0
3.75	-	2	12	-	-	22.5
4	-	-	16	-	-	24.0
4.25	-	-	12	8	-	25.5
4.5	-	-	8	16	-	27.0
4.75	-	-	4	24	-	28.5
5	-	-	-	32	-	30.0
5.25	-	-	-	24	16	31.5
5.5	-	-	-	16	32	33.0
5.75	-	-	-	8	48	34.5
6	-	-	-	-	64	36.0



(a) Constellation diagram design of 24-QAM based on 16-QAM (b) Construction of Huffman codewords



(c) Mapping Huffman codewords to 24-QAM

Figure 4.3: An example of VLC-based modulation: realizing the link rate of 27 Mbps with 24-QAM and  $l_{avg} = 4.5$

As a result, we are able to continuously realize any arbitrary link rate by varying the number of codewords and the lengths of these codewords in the Huffman code. For example, when the codewords in the Huffman code have only two different lengths, i.e.,  $k$  bits and  $k + 1$  bits, the code is always complete when

$$n_k 2^{-k} + n_{k+1} 2^{-(k+1)} = 1, \quad 0 \leq n_k \leq 2^k \quad (4.2)$$

where  $n_k$  and  $n_{k+1}$  are the numbers of  $k$ -bit and  $(k + 1)$ -bit codewords, respectively. Furthermore, we have

$$\begin{aligned} l_{avg} &= k n_k 2^{-k} + (k + 1) n_{k+1} 2^{-(k+1)} \\ &= k + 1 - n_k 2^{-k}, \quad 0 \leq n_k \leq 2^k \end{aligned} \quad (4.3)$$

By controlling the values of  $n_k$  and  $n_{k+1}$ , we can achieve different values of  $l_{avg}$  with an interval of  $2^{-k}$ . Table 4.1 lists 17 intermediate link rates that can be realized between the 5 existing 802.11 a/g link rates<sup>3</sup>, with a fractional value of  $l_{avg}$  (the 5 existing link rates correspond to integer values of  $l_{avg}$ ). To further reduce the gap between link rates, more options of channel code rates and a larger variety of VLC codeword lengths could be incorporated.

#### 4.4.2 Constellation Diagram Design

Based on the choice of link rate, the key problem of realizing this link rate is then how to construct the Huffman code and the corresponding constellation diagram, to which the codewords will be mapped. The number of constellation points in the constellation diagram has to be equal to the number of codewords in the Huffman code. When  $l_{avg}$  is fractional, the number of constellation points is not an integer number of the power of 2. Hence, a new constellation diagram needs to be designed for every different value of  $l_{avg}$ . For example when  $l_{avg} = 4.5$ , a constellation diagram with 24 constellation points is needed according to Table 4.1.

---

<sup>3</sup>A fixed channel code rate of 1/2 is used.

Our primary goal is to design new constellation diagrams for different values of  $l_{avg}$  in a generic manner, so that different choices of continuous link rates could be supported by the same wireless hardware. To reach this objective, our basic idea is to design the required constellation diagram by expanding an existing QAM constellation diagram with the size smaller than but closet to the size of Huffman code. For example, for the Huffman code with a size of 24, we construct 24-QAM from 16-QAM used in existing wireless systems. Generality of designing new constellation diagrams, in this way, is then ensured by the symmetry of QAM constellations<sup>4</sup>.

Note that, for each value of  $l_{avg}$ , there may be multiple ways to design the new constellation diagram from an existing QAM constellation diagram. To minimize the average transmit power, as illustrated in Figure 4.3(a), we place the additional constellation points to the unused signal positions with the smallest amplitude in the complex plane. If there are multiple choices with the same level of average transmit power, the one that maximizes the symmetry of constellation diagram will be chosen. For example, among the two choices of 6-QAM shown in Figure 4.4, the one with better symmetry in Figure 4.4(b) will be chosen.

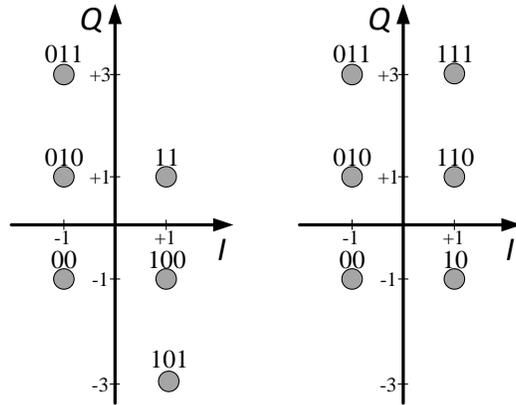
### 4.4.3 Codeword Construction and Mapping

Being similar to our constellation diagram design, we construct the Huffman codewords based on the fixed-length codewords in existing wireless systems. Any  $2^K$ -QAM constellation diagram corresponds to a set of  $2^K$  fixed-length codewords, which can be represented by the leaf nodes of a full binary tree with a height of  $K$ . Our approach, then, constructs the Huffman tree by appending new leaf nodes to the corresponding binary tree of fixed-length codewords.

More specifically, two new leaves are appended to a leaf node in the binary tree, hence replacing one  $K$ -bit codeword with two  $(K+1)$ -bit codewords. The amount of leaf nodes in the binary tree being appended depends on the value of  $l_{avg}$  and the constitution of VLC codewords. For example, when  $l_{avg} = 4.5$ , according to Eq. (4.3) and Table 4.1, 16 5-bit

---

<sup>4</sup>Other constellation shapes such as triangular [24] constellations, while providing better power efficiency, are generally asymmetric. Hence, new constellation diagram for each value of  $l_{avg}$  has to be designed in a distinct way implemented with specialized hardware.



(a) Asymmetric design    (b) Symmetric design

Figure 4.4: Design choices for 6-QAM

codewords are needed in the VLC and hence 8 leaf nodes in the binary tree of 16-QAM need to be appended. As shown in Figure 4.3(b), by appending two new leaves to the leaf “1010” of the binary tree, two 5-bit codewords (“10100” and “10101”) are generated. The remaining 4-bit codewords in the binary tree, on the other hand, are also used as Huffman codewords. It is straightforward to prove that the set of Huffman codewords generated by this approach is complete.

The remaining problem, then, is how to determine the leaves in the binary tree to be appended. Principally, we choose to append the leaf nodes that correspond to the constellation points with the largest amplitude in the QAM constellation diagram. For example in Figure 4.3(c), the 4-bit codewords corresponding to the 8 constellation points at the two outer sides of the 16-QAM constellation diagram are appended to generate 5-bit codewords. These 5-bit codewords are then mapped to the same set of constellation points of 16-QAM, as well as the new constellation points added for 24-QAM.

Note that the constellation diagrams, Huffman codewords and their mappings between each other are completely constructed offline. During the online modulation and demodulation, the wireless system converts the data codewords and constellation symbols between each other based on the pre-defined mapping, and hence incurs negligible computational overhead on both Tx and Rx.

## 4.5 VLC-BASED DEMODULATION

In principle, vMod demodulates data by demapping the received symbols from the constellation diagrams developed in Section 4.4.2. However, due to the variable lengths of VLC codewords, the information about the lengths of VLC codewords being modulated in a symbol is crucial to the correctness of VLC-based demodulation. As described in Section 4.3, with such information, erroneous demodulation of any signal will not affect demodulation of others and the accuracy of demodulation is then linearly proportional to the level of channel noise. In this section, we describe how to convey such information without consuming any additional wireless spectrum or impairing any data throughput.

First, to minimize the amount of data sent, we do not send the numbers of VLC codeword lengths directly in data symbols. Instead, we partition the subcarriers within a symbol into equally sized groups and convey the required information with respect to each group, i.e., the numbers of VLC codewords with different lengths being used in each group. These numbers may be different over multiple groups according to the input bitstream.

Then, to send such information to the receiver without impairing data throughput, we only apply VLC-based modulation to some of the subcarriers in each group and modulate other subcarriers in the group using the regular modulation scheme in existing systems. For example in Figure 4.5, when the codewords in the Huffman code only have two different lengths, the number of longer VLC codewords ( $N_L$ ) in each group is represented by how these two types of subcarriers in the group are permuted in the frequency domain. In particular, for a group of  $N$  subcarriers<sup>5</sup>, the value of  $N_L$  is ranged between 0 and  $N/2$ , and each

---

<sup>5</sup>For simplicity, the value of  $N$  is always an even integer.

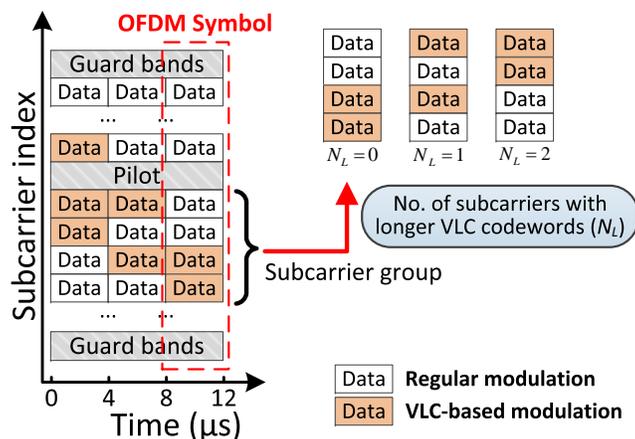


Figure 4.5: Conveying information of VLC codeword lengths. Every four subcarriers in a symbol forms a group. VLC codewords only have two different lengths.

possible value of  $N_L$  corresponds to a specific permutation of the two types of subcarriers. It is easy to see that the number of possible permutations always exceeds the number of possible values of  $N_L$ . When more types of VLC codewords are involved in the Huffman code, larger groups of subcarriers and more complicated permutation of subcarriers will be needed to convey the extra information about codeword lengths.

To convey such information, a naive approach is to embed the value of  $N_L$  into the packet preamble or header, which can be represented by at least  $\lceil \log_2(N/2 + 1) \rceil$  data bits. However, this would significantly hurt the data throughput. For example, 4 data bits are required to represent  $N_L$  when  $N = 24$ , which translates to 1/6 and 1/12 throughput loss with BPSK and QPSK modulation, respectively<sup>6</sup>. Instead, our approach seeks to embed this information without throughput loss, and minimize the size of control information being embedded into the frame header.

<sup>6</sup>Assuming that the same code rate is applied over preamble and data bits.

Based on this design, we further solve two technical problems: 1) How could the demodulator decode the conveyed information by recognizing whether the received signal in a subcarrier is VLC-modulated? 2) How could the demodulator ensure the correctness of such information, which may be distorted by channel noise?

#### 4.5.1 Recognition of the Received Signal

The demodulator determines whether a subcarrier is VLC-modulated based on the measurable difference between the constellation diagrams used for regular modulation and VLC-based modulation.

In practical WiFi systems, in order to maintain the transmit power to be constant, the power of every constellation diagram needs to be normalized before being used for modulation. Therefore, although the constellation diagram being used for VLC-based modulation is an expansion of an existing QAM-based constellation diagram (e.g., the 24-QAM constellation diagram is an expansion of 16-QAM as shown in Figure 4.3(a)), the positions of its constellation points are changed due to power normalization. For example in Figure 4.6 which shows parts of the 16-QAM and 24-QAM constellation diagrams in the first quadrant, although the constellation points with smaller amplitudes in 24-QAM are directly inherited from the ones in 16-QAM, their positions in 24-QAM are different from those in 16-QAM. Our measurement studies show that the power ratio of such difference could be as large as 0.165 under the unity power constraint, corresponding to more than 10 dB in transmit power that can be reliably measured by commodity transceivers.

Based on such difference, when we overlap the regular and the VLC-based constellation diagrams together, we map the received signal to the nearest constellation point on this overlapped diagram. Then, we can tell whether this constellation point belongs to the VLC-based constellation diagram. Further, according to the mapping between VLC codewords and constellation points in Section 4.4.3, we can decide whether this constellation point refers to a longer VLC codeword. As shown in Figure 4.6, we measure  $d^L$  and  $d^S$  as the distance from the received signal to the nearest VLC-based constellation point with shorter and longer codewords, respectively, and measure  $d^R$  as its distance to the nearest regular constellation

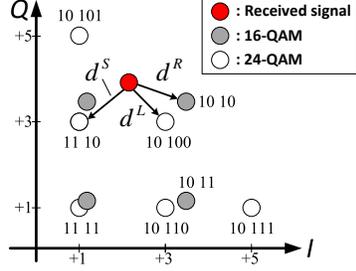


Figure 4.6: Recognition of the received signal

point. If  $\min\{d^L, d^S\} < d_R$ , the demodulator considers the corresponding subcarrier as VLC-modulated, and in this case if  $d^L < d^S$ , the received signal is considered to carry a longer VLC codeword.

#### 4.5.2 Resistance against Channel Noise

With the VLC-modulated subcarriers being recognized, the numbers of VLC codewords with different lengths can be retrieved according to the permutations of subcarriers in the symbol. However, in practical wireless networks where the channel is always noisy, if the type of any one subcarrier in the group is incorrectly recognized, the retrieved information about VLC codeword lengths will be wrong and prevent correct demodulation of all subcarriers.

To minimize such risk and resist against channel noise, we encode permutations of subcarriers using the Hamming code, so that the error on recognizing the received signal in any single subcarrier could be corrected. For each group with a size  $N$ , we indicate the permutation of subcarriers in the group as a  $N$ -bit binary string: “0” indicates that the corresponding subcarrier is regularly modulated and “1” indicates a VLC-modulated subcarrier. For example, when the VLC codewords only have two different lengths, every possible value of  $N_L$  corresponds to a Hamming codeword with an equal number of 0s and 1s. All the codewords in the (8,4,4) Hamming code, except the two codewords of all 0s and all 1s, can be used

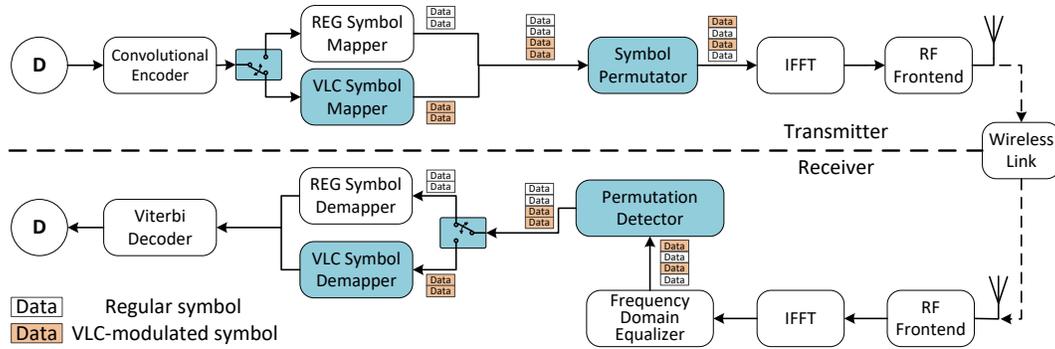


Figure 4.7: PHY-layer vMod implementation

for permutations of subcarriers. At the receiver, if the numbers of recognized regular and VLC-modulated subcarriers are not equal, the error could be easily corrected by a Hamming decoder.

As discussed in Section 4.4.1, using more ( $N > 2$ ) codeword lengths could further increase the granularity of wireless link rates. For example, for the VLC of 24-QAM shown in Figure 4.3(c), we could further increase the average code length from 4.5 bit to 4.53125 bit by splitting one 5-bit codeword into two 6-bit codewords, which provides extra granularity to the codes that consist of only 4-bit and 5-bit codewords. In this case, the total number of bits within one group is not only decided by the number of the longest codeword that occurs, but also by that of the other  $N - 2$  codeword lengths. For example, a group of 3 subcarriers carrying 15 bits can be represented as three subcarriers carrying 4-bit, 5-bit and 6-bit codewords, respectively, or three 5-bit subcarriers. In this case, our proposed technique could be easily extended to convey the additional information about more VLC codeword lengths, by adopting more complicated permutation patterns of subcarriers.

## 4.6 IMPLEMENTATION

We implemented vMod over USRP SDR boards with the GNURadio toolkit, which realizes a WiFi transceiver over the 5GHz frequency band. vMod requires the minimum amount of modifications over existing WiFi PHY by revising only the modulation and demodulation modules. Same as WiFi, the link rate of vMod frames is specified by the transmitter according to the instantaneous channel estimation, which is conducted by the rate adaptation algorithm. A specific OFDM symbol is appended to the frame header to indicate the vMod link rate, as well as other parameters required for decoding vMod frames.

### 4.6.1 PHY-Layer Implementation

The architecture of the PHY-layer system implementation is shown in Figure 4.7, which is based on the existing WiFi PHY with the newly added modules highlighted in color. Our design requires the minimum amount of modification on the WiFi modulation and demodulation, retaining other modules of WiFi PHY intact.

At the transmitter, we add a new VLC Symbol Mapper, which uses VLC-based constellation diagrams developed in Section 4.4.2, to be operated in parallel with the Regular (REG) Symbol Mapper in the existing WiFi systems. The input datastream, after channel encoding, is distributed to the two mappers, which both produce signal symbols to convey the information of VLC codeword lengths as described in Section 4.5. More specifically, the bitstream divider first passes the required number of data bits being used in the regular constellation diagram to the REG Symbol Mapper. Afterwards, it keeps passing the following data bits to the VLC Symbol Mapper until the required amount of signal symbols are generated. The outputs of these two mappers are merged to the Symbol Permutator, which embeds the information of VLC codeword lengths as described in Section 4.5.2.

At the receiver, the received RF signal will be first dissembled to OFDM symbols by FFT and recovered from channel distortion by the Frequency Domain Equalizer. Afterwards, the Permutation Detector detects the permutation pattern over symbols to retrieve the

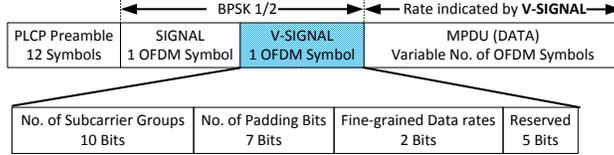


Figure 4.8: Modified frame format

information of VLC codeword lengths in the VLC-modulated symbols. It also reorganizes the signal symbols and passes the symbols to the REG and VLC Symbol Demapper, respectively. Finally, the Viterbi decoder decodes the demodulated bitstream.

This implementation over USRP also has limitations. At the physical layer, we estimate the channel response and equalize the signals in the frequency domain. However, compared to commodity WiFi devices, the performance of the RF receiver at the USRP board is restricted due to the absence of automatic gain control (AGC) which cannot be emulated or implemented in the GNURadio software. This AGC, instead, shall be further implemented by customizing the FPGA core in the USRP hardware.

#### 4.6.2 MAC-Layer Implementation

We implemented a complete WiFi MAC layer with functionalities such as carrier sensing and acknowledgements [38]. However, since the RF signals harvested by USRP have to be sent to and processed at the host PC, a long delay is incurred at signal transmission and processing. It is hence difficult to meet the timing requirements of the Distributed Coordination Function (DCF) in WiFi. To address this challenge, we follow the same method in [38] and use a scaling factor  $\beta (= 5000)$  to emulate the real WiFi MAC.

vMod can work with any existing rate adaptation algorithms, by providing them continuous choices of link rates to select from. We incorporated two mainstream rate adaptation algorithms, namely Minstrel [64] and Adaptive Auto Rate Fallback (AARF) [56], into our implementation to emulate the realistic behaviors of commodity wireless devices in practical scenarios. Minstrel is widely used in commodity systems based on past statistics of packet

reception, and has been integrated as part of Linux OS kernel. AARF adaptively adjusts the link rate based on recent consecutive successes or failures of frame transmissions. This implementation will also be used in our real-world experiments in Section 4.8.

Our implementation also contains a modified format of WiFi frame. In frame assembly of WiFi, the number of OFDM symbols and padding bits used for a payload is only determined by the link rate and payload length, and is independent of payload content. However, this independency is invalid in vMod because of the usage of VLC. VLC-based modulation brings uncertainty to the codeword lengths and furthermore the number of OFDM symbols being used, and this uncertainty is related to payload content. To address this uncertainty, we insert one additional OFDM symbol, namely V-SIGNAL, between the SIGNAL symbol and MPDU symbols in a WiFi frame. As shown in Figure 4.8, this symbol carries information about the number of subcarrier groups and padding bits. It also uses two additional bits, along with the 4-bit RATE field in the existing SIGNAL symbol, to indicate the link rate being used. Being similar to SIGNAL symbol, the V-SIGNAL symbol is transmitted in the lowest link rate. Afterwards, MPDU is transmitted using the actual link rate declared in V-SIGNAL.

## 4.7 PERFORMANCE EVALUATION

Built on our implementation in Section 4.6, we evaluate the performance of vMod by comparing it with existing WiFi systems and rateless code designs. Our experiment results show that vMod is effective to fill the gap between existing discrete link rates. We also show that the communication and computation overhead of vMod is comparable to that of the existing WiFi, and is up to 10x lower than that of rateless codes. These results, then, ensure that vMod could be practically utilized to improve the network throughput in various severe wireless network scenarios.

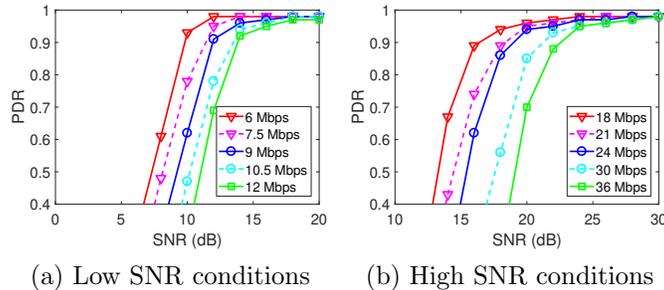


Figure 4.9: Packet delivery ratio (PDR) under different channel conditions. The new and existing link rates are displayed in dotted and solid lines, respectively.

#### 4.7.1 Experiment Setup

We conduct our experiments in a  $10\text{m} \times 10\text{m}$  open space office using two USRP N210 motherboards with UBX-40 RF daughterboards: one USRP continuously transmits 250 frames with random payloads of 300 bits, and another USRP decodes the frames and counts the ones that are received properly. Due to the MAC scaling factor  $\beta$ , the interval between two consecutive transmissions is around 0.4 seconds. We tune the channel SNR by adjusting the USRP Tx Gain between 2 dB and 30 dB, and the maximum Tx power is equivalent to 13 dBm. Each USRP is connected to a Dell desktop PC, which is equipped with an Intel i5-3475S CPU 2.9GHz and 8GB memory. The communication and computation overheads of transmission are then measured at the host PCs of both sender and receiver.

We use the same implementations of rateless codes as in Section 4.2. Ideally, the rateless sender continues to send new data chunks without channel estimation or feedback, until the receiver successfully decodes the packet. However, when being implemented over GNURadio, if the receiver misses a data chunk because of failing to detect the packet rather than failing the packet CRC test, it will prolong the decoding process or even fail. Hence, we set the receiver to acknowledge reception of chunks in order. The receiver would pause further reception of chunks till a chunk with correct sequence number is received. Once the receiver successfully decodes a packet, it sends a *ACK\_FOR\_PACKET* frame to the sender.

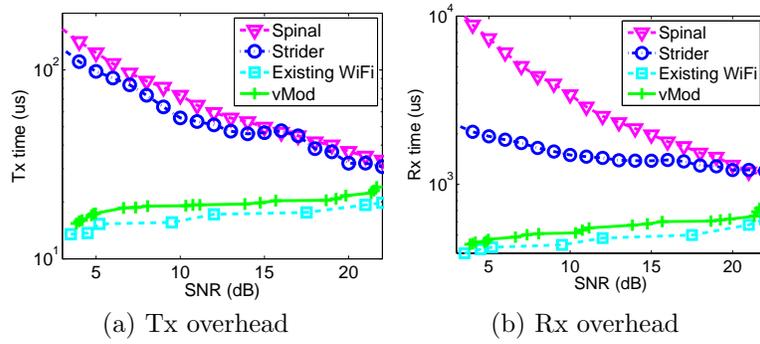


Figure 4.10: Computational overhead

#### 4.7.2 Packet Delivery Ratio

We first evaluate the Packet Delivery Ratio (PDR) of the continuous link rates under different channel SNR conditions. In order to achieve the same level of PDR, a better channel condition with higher SNR is required for a higher link rate. As shown in Figure 4.9 which contains cases of both low and high channel SNR conditions, for any given SNR condition, the PDR of the intermediate continuous link rates always lies between that of their adjacent existing link rates. The PDRs of both categories of link rates eventually converge to 1 when the channel SNR condition is sufficiently good.

These results show that vMod is very effective to fill the gap between existing link rates and hence provides better choices for dynamic channel conditions. For example, when the channel SNR is 17 dB and the PDR is required to reach 90%, the link rate of 21 Mbps could be used to replace the existing link rate of 18 Mbps, yielding 16.7% improvement of throughput. Such improvement of data throughput in practical wireless scenarios with dynamic channel conditions will be further investigated in Section 4.8.

### 4.7.3 Overhead

We evaluate the Tx communication overhead and Rx computation overhead caused by vMod operations, which also measure the power consumption of vMod as explained in Section 4.2. The communication overhead at the wireless sender for sending a unit amount of data is shown in Table 4.2. The results indicate vMod’s overhead is similar to that of WiFi, and is at least 50% lower than that of rateless codes. The major reason for such overhead reduction is that vMod constantly puts the RF radio at the wireless sender to idle after each transmission, instead of keeping the radio alive for sending rateless data chunks. Such reduction of wireless channel occupancy, in addition, also provide extra channel access to other concurrent wireless traffic in the same channel.

Table 4.2: Tx communication overhead (*us/bit/MHz*)

SNR	vMod	WiFi	Strider	Spinal
6	0.2043	0.2347	0.5152	0.5500
12	0.1463	0.1603	0.2955	0.3000
24	0.0499	0.0505	0.1591	0.1500

The computation overheads for processing one data frame are shown in Figure 4.10. For rateless codes, such time is measured until the entire data packet is successfully decoded. At both sides, the computational overhead of vMod is similar to that of existing WiFi and remains nearly constant. Comparatively, rateless codes incur up to 10x higher computational overhead, especially with low SNR, because a rateless receiver needs to continuously decode incoming data chunks even if many of them are actually corrupt. Due to such processing of extra data chunks, even when the channel SNR improves to 24 dB, Table 4.3 shows that the computational overhead of rateless codes is still more than 100% higher than that of vMod. Less computational complexity of vMod also reduces its requirement on computing power of digital processing chips.

Table 4.3: Rx computation overhead (*us*)

SNR	vMod	WiFi	Strider	Spinal
6	486	430.1	1842.9	6074.6
12	560.1	480.2	1436.7	2543.5
24	753.4	635.0	1282.5	1042.4

## 4.8 REAL-WORLD EXPERIMENTATION

In this section, we further apply our designs into real-world network scenarios with dynamic channel conditions and commodity rate adaptation approaches. In this way, we evaluate the improvement of data throughput by vMod. Our experiment results show that vMod outperforms rateless codes and improves the WiFi throughput by up to 30% over a narrowband link, but incurs 95% less overhead than rateless codes. It also retains such throughput in severe channel conditions with strong interference and channel fading.

### 4.8.1 Experiment Setup

Without loss of generality, in this section we only evaluate the wireless throughput over a single narrowband link. Further improvement of throughput in practical scenarios, can be achieved by either expanding the link bandwidth or employing multiple links at the same transceiver, and vMod is fully compatible with these techniques. We adopt both Minstrel and AARF algorithms for link rate adaptation, and conduct our experiments in a  $5m \times 5m$  office with strong multipath fading. The sender and receiver are placed out of line of sight and 4 meters away from each other.

Similar to Section 4.7, we tune the Tx gain of the USRP sender to emulate different SNR conditions. To emulate both fast and slow channel SNR variations, we configure the Tx gain for sending the  $N$ -th data frame as  $A(1 + \rho \sin(2\pi\omega N))$ , where  $\omega = 0.1$  (i.e. the

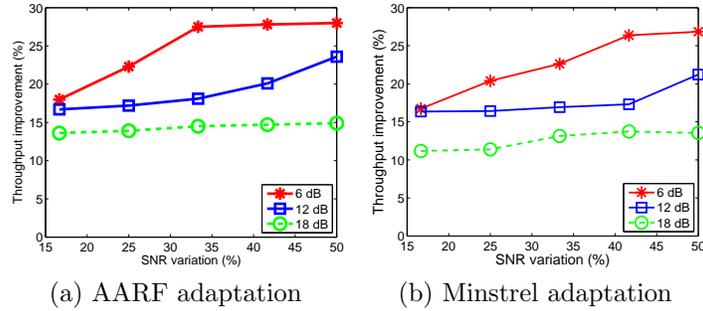


Figure 4.11: vMod's throughput improvement over existing WiFi with a single narrowband link

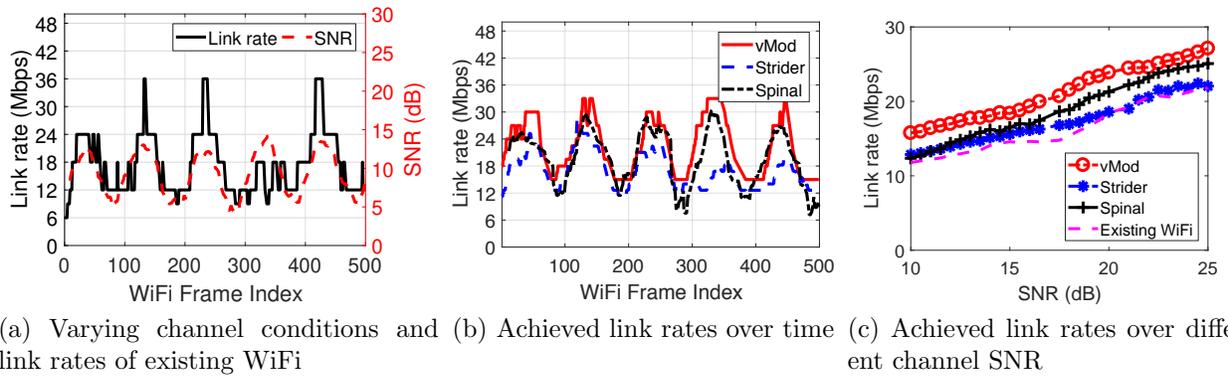


Figure 4.12: Comparison of achieved wireless throughput with existing WiFi and rateless codes over a single narrowband link. Tx gain is  $6 \pm 3$  dB. vMod outperforms Spinal and Strider by up to 25%, especially when the channel SNR is low. It also incurs much lower computational overhead compared to Spinal and Strider, as shown in Figure 4.10.

period of the Tx gain change is 10 frames) and  $A$  is the average level of the Tx gain being configured between 6dB and 18dB. The parameter  $\rho \in \{\frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{5}{12}, \frac{1}{2}\}$  is an indicator of the channel variation, where a larger  $\rho$  indicates a fast fluctuating channel.

## 4.8.2 Throughput Improvement

As shown in Figure 4.11(a) and 4.11(b), vMod improves WiFi throughput with both AARF and Minstrel rate adaptation approaches. In AARF, the throughput is improved by up to 30%. As the channel condition becomes better, the throughput improvement decreases and remains nearly constant 14% when  $A$  reaches 18 dB. The figure shows that in lower channel condition, intermediate link rates can provide more throughput gain. For example, when in low channel condition, major link rates being used are 6 Mbps and 9 Mbps. When the channel condition cannot support 9 Mbps but is surplus to support 6 Mbps, an intermediate link rate of 7.5 Mbps can provide an extra 1.5 Mbps link rate, which is a 25% improvement compared to 6 Mbps. In higher SNR condition, however, 12 Mbps, 18 Mbps and 24 Mbps are mainly used and an extra 1.5 Mbps throughput only accounts for at most 12.5% throughput gain. Similarly, as shown in Figure 4.11(b), such throughput improvement could be further up to 30% with the Minstrel approach, although the overall throughput is slightly lower than that of AARF.

Furthermore, we have also compared the throughput achieved by vMod with that of rateless codes. As shown in Figure 4.12(a), when the Tx gain of the USRP sender is periodically fluctuating between 3 dB and 9 dB, the channel SNR varies accordingly and leads to different link rates. Nevertheless, under such dynamic channel conditions, Figure 4.12(b) shows that vMod can always provide a network throughput that exceeds the throughput achieved by both Strider and Spinal codes. Such advantage over different channel conditions is further demonstrated by Figure 4.12(c): although rateless codes are not based on link rate adaptation and also continuously vary the actual link rate with the channel SNR condition, vMod outperforms their performance by up to 25% under poor channel conditions. The

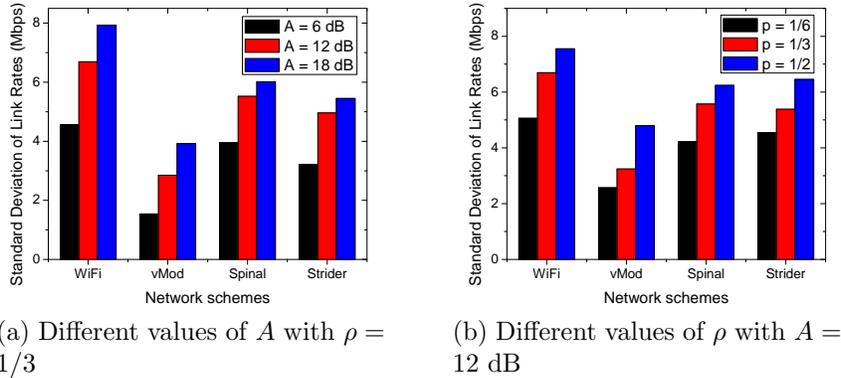


Figure 4.13: Stability of channel throughput over dynamic channel conditions

major reason of this advantage is that under such severe scenarios, rateless codes require transmitting a large amount of data trunks in order to successfully deliver one data packet, and hence incur extra transmission latency.

On the other hand, when the channel condition improves, although less data trunks are transmitted in rateless codes, the gap between discrete link rates also increases. Therefore, the throughput advantage of vMod over rateless codes remains at the same level. In particular, vMod improves the network throughput with a much lower amount of communication and computation overhead as shown in Section 4.7.3, and is hence applicable for practical deployment over the off-the-shelf wireless systems.

### 4.8.3 Stability of Channel Throughput

In practical wireless network scenarios where the channel SNR fluctuates, Figure 4.12(b) also shows that corresponding link rate changes over time, and demonstrates that the curve of link rate fluctuation using vMod is smoother than both the existing WiFi and rateless code designs. This experimental result, hence, verifies our expectation that continuous link rates can better accommodate to the momentary channel conditions. Furthermore, we also calculated the standard deviation of the link rates over the experiment period with different values of  $A$  and  $\rho$  that decides the fluctuation of channel condition. The results in Figure

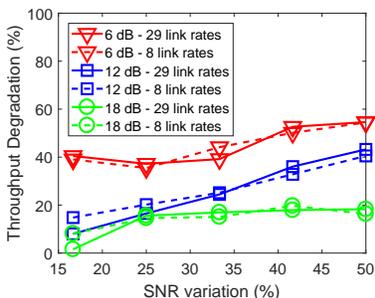


Figure 4.14: Throughput in an interfered channel

4.13, again, verify the vMod’s advantage in providing more stable throughput over dynamic channel conditions. In particular, when the channel condition fluctuates faster with a smaller value of  $\rho$ , vMod provides better stability to the WiFi network throughput.

#### 4.8.4 Impact of Channel Interference and Fading

We investigate the anti-interference capability of vMod by adding an interfering source node, which transmits a 1,500-byte WiFi frame every 0.1 seconds in the same frequency band as USRP. The throughput degradation caused by such interference is shown in Figure 4.14, where 8 link rates are employed by the existing WiFi standard and 29 continuous link rates are employed by vMod as specified in Table 1. When the channel SNR is lower or fluctuates more quickly, the throughput degradation becomes larger. However, such degradation with continuous link rates provided by vMod is very close to that with existing coarse-grained link rates in all channel SNR conditions. This result demonstrates that the adoption of continuous link rates is suitable in practice and does not impair the capability of WiFi networks in resisting channel interference.

We also tested the performance of vMod over a frequency selective fading channel, by passing the generated Tx waveform to a software-defined selective fading channel before emitting it to the air. The experiment results are shown in Figure 4.15, and demonstrate

40% and 30% throughput improvement compared to existing WiFi under poor (SNR=6 dB) and medium (SNR=12 dB) channel condition, respectively. vMod also outperforms rateless codes under different conditions in such a frequency selective fading channel.

## 4.9 RELATED WORK

**Adaptive Modulation.** Our proposed design of vMod is related to existing work on adaptive modulation, which improves throughput over fading channels by adjusting the modulation scheme to the instantaneous channel condition [32]. As a representative approach to adaptive modulation, Trellis Coded Modulation (TCM) [103] introduces intermediate data rates by combining the modulation procedure with the channel coding and applying different Trellis codes over the QAM constellation points. However, TCM requires large constellation diagrams and is rarely used in wireless networks due to the high computational complexity. Furthermore, TCM is known to be prone to phase ambiguity and hence not suitable to be used in dynamic wireless scenarios. Other approaches provide more choices of link rates through constellation diagrams that contain  $3 \times 2^p$  signal points [73, 57] and have been extended to the plastic optical fiber communication [100], but can only reduce the gap between consecutive link rates down to half bit per data symbol. Adjusting the RF transmit power over fading channel is proposed in [49, 32] so as to maximize the channel throughput. However, the performance gain is restricted due to the limited levels of RF transmit power in commodity WiFi devices [22].

**Rateless Codes.** Rateless codes aim to fully utilize the channel capacity without channel feedback [90]. Existing designs differ in the way they create continuous streams of encoded data: Strider [36] builds on a conventional channel code and transmits linear combinations of conventionally encoded symbols, and Spinal code [76] utilizes a pseudo-random hash function to produce the sequence of coded symbols. However, as described in Section 4.2, these designs incur a large amount of extra communication and computation overhead, and are hence inapplicable to be deployed over resource-constrained wireless devices, particularly embedded sensors in IoT applications. In addition, new MAC protocols are also needed for

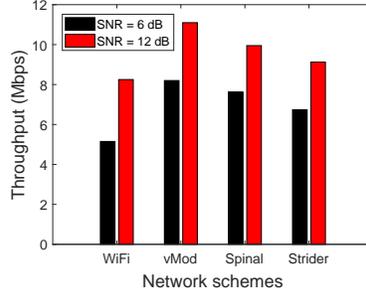


Figure 4.15: Throughput over fading channel

rateless codes [46], because the transmitter and receiver have to be appropriately coordinated so that the sender can know when the packets are successfully decoded and adjust the interval of packet transmissions accordingly. Such MAC protocol is, however, incompatible with any existing wireless network and requires fundamental network redesign. Instead, our approach is completely compatible with any existing MAC protocol, and does not require any modification of the network protocol stack.

**Wireless Networking.** Many techniques have been developed to tackle with wireless channel fluctuations, improve the throughput and reduce the delay. Such techniques include multiuser MIMO (MU-MIMO) [95], full-duplex communication [20] and side channels [69], and have been integrated to recent wireless standards such as LTE, 802.11ac and 802.11ax [6]. Since vMod provides continuous choices of link rates by modifying the channel modulation, it is independent from most PHY and MAC modules of existing wireless systems. It is fully compatible with and can be applied to the aforementioned advanced techniques, to further improve their performance in highly severe application scenarios.

## 4.10 DISCUSSION

**Compatibility with link rate adaptation.** Most rate adaptation algorithms reactively adapt the link rate according to frame delivery in the past [106]. Others adjust the link rate based on different channel state indicators, including the signal strength [16], channel state information (CSI) [63] or effective SNR [40]. Details of packet reception, such as channel BER [101] and symbol error vectors [85], have also been used. vMod, as a channel modulation scheme, provides continuous choices to link rate adaptation and is hence compatible with any link rate adaptation algorithm. It does not require any modification to existing rate adaptation algorithms, either.

**Variable-length codewords based on Huffman code.** Huffman code in this chapter is exploited to construct the variable-length codewords from existing fixed-length codewords in QAM modulations. As described in Section 4.4, the completeness and optimality of Huffman code in vMod are ensured by the constitution of such variable-length codewords and the corresponding continuous link rates. In particular, multiple codeword lengths provide more choices of continuous link rates, but also increase the complexity of demodulation, because the number of each type of codeword being used must be explicitly notified to the receiver for correct demodulation.

**Accuracy in VLC-based demodulation.** The accuracy of VLC-based demodulation is decided by two major factors: *i*) the accuracy of symbol recognition, and *ii*) the capability of error correction over the Hamming code being used. Symbol recognition requires detectable distance between constellation points from the regular QAM diagram and the new vMod diagram. As channel SNR improves, denser constellation diagrams are used and hence reduce such distance. However, the higher SNR also suppresses channel noise and achieves higher accuracy of symbol detection. Hence, the dependency of Hamming code on successful decoding varies across different levels of channel SNR. We found that a revised (8,4,4) Hamming code meets the requirements of any constellation diagram smaller than 64-QAM. We refer exploration of such accuracy in higher-order constellation diagrams as future work.

**I/Q imbalance in vMod.** The power efficiency and demodulation of vMod could also be affected by the heterogeneous I/Q imbalance caused by the asymmetry of the proposed constellation diagrams. However, in practice, such I/Q imbalance is usually calibrated by manufacturers according to the IEEE standard [1], before the wireless system has been released to market. The impact of such I/Q imbalance at run-time, hence, can be efficiently eliminated by the RF frontend of wireless, before the received wireless symbols are being demodulated in the digital logic.

#### 4.11 CHAPTER SUMMARY

In this chapter, we present a lightweight solution towards maximum wireless throughput for IoT, by transforming the choices for link rate adaptation from discrete to continuous. Our design builds on the variable-length code (VLC) that conveys a fractional number of data bits in each symbol. We implemented and evaluated our design over practical SDR platforms, and show that vMod greatly improves the wireless throughput with negligible overhead.

## 5.0 CONCLUSION

This dissertation focuses on reducing the wireless network delay for real-time traffic, by exploring the wireless channel SNR margin in the wireless channel, i.e., the difference between the instant channel SNR and the minimum channel SNR that is required to support the data rate being used. SNR margin is intrinsic to the rate adaptation approaches due to their effort in avoiding costly retransmissions. By only exploring a small portion of the SNR margin, i.e., injecting certain pattern of interference to the existing transmission (main channel), we build extra side channels that operate concurrently with the main channel hence greatly reducing the network latency.

In Chapter 2, we conduct experiments to measure the actual amount of SNR margin that exists in the IEEE802.11a standard. Based on the measured value of the SNR margin and the multi-carrier property of the OFDM modulation, a side channel is built up which encodes information as the patterns of the energy erasure over OFDM subcarriers. By actively interfering with the generated main channel signals from the same device, the side channel reduces the network queuing delay by allowing low-latency traffic to transmit concurrently with the outgoing traffic from the same device. Experiments over both the SDR platforms and FPGA based hardware platform demonstrate the effectiveness of the side channel in reducing the data transmission latency and providing a data throughput higher than 1 Mbps, with minimum impact on the performance and cost of the main wireless channel.

In Chapter 3, we extended our research to enable multiple IoT devices to share the same occupied spectrum concurrently, so as to reduce their wireless channel access delay due to CSMA mechanism. In order to achieve this, we dynamically allocate the subcarriers of the side channel to different devices with respect to their real-time traffic demands, which would not interfere others due to the orthogonality of the OFDM subcarriers. A real-time

signal processing module is proposed to demodulate the side channel that utilizes the asynchrony between the main channel and the side channel for precise side channel demodulation. Furthermore, the transmit power of the side channel is adaptively controlled to limit extra interference incurred. Experiment results demonstrate that EasyPass’s capability in restraining the wireless latency within 10ms regardless of traffic patterns in the main channel, and its scalability that can support up to 13 IoT devices.

Different from Chapter 2 and Chapter 3, Chapter 4 indirectly reduces the network delay for IoT devices by improving the network throughput. A lightweight solution towards maximum wireless throughput for IoT is proposed which transforms the choices for link rate adaptation from discrete to continuous. The key idea of vMod is to modulate a fractional number of data bits into each symbol by employing the Variable-Length Code (VLC), which is able to statistically yield any link rate. Our design builds on the variable-length code (VLC) that conveys a fractional number of data bits in each symbol. We implemented and evaluated our design over practical SDR platforms, and show that vMod greatly improves the wireless throughput with negligible overhead.

## BIBLIOGRAPHY

- [1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard 802.11*, 2012.
- [2] IEEE approved draft standard for IT - telecommunications and information exchange between systems. *IEEE P802.11ac/D7.0*, pages 1–456, 2013.
- [3] A. Amiri Sani, K. Boos, M. H. Yun, and L. Zhong. Rio: A system solution for sharing I/O between mobile systems. In *Proceedings of the 12th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 259–272. ACM, 2014.
- [4] D. Becker, B. Schäufele, J. Einsiedler, O. Sawade, and I. Radusch. Vehicle and pedestrian collision prevention system based on smart video surveillance and c2i communication. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 3088–3093. IEEE, 2014.
- [5] B. Bellalta. Ieee 802.11 ax: High-efficiency wlans. *IEEE Wireless Communications*, 23(1):38–46, 2016.
- [6] B. Bellalta, L. Bononi, R. Bruno, and A. Kessler. Next generation ieee 802.11 wireless local area networks: Current status, future directions and open challenges. *Computer Communications*, 2015.
- [7] M. Benjamin. *Drone warfare: Killing by remote control*. Verso Books, 2013.
- [8] M. Bernard and B. D. Sharma. Some combinatorial results on variable length error correcting codes. *Ars Combinatoria*, 25:181–194, 1988.
- [9] M. A. Bernard and B. D. Sharma. A lower bound on average codeword length of variable length error-correcting codes. *IEEE transactions on information theory*, 36(6):1474–1475, 1990.
- [10] D. Bharadia, E. McMillin, and S. Katti. Full duplex radios. *ACM SIGCOMM Computer Communication Review*, 43(4):375–386, 2013.

- [11] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans. In *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, volume 2, pages 392–396. IEEE, 1996.
- [12] S. Biaz and S. Wu. Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison. In *IEEE ISCC*, pages 130–136, 2008.
- [13] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. An IEEE 802.11a/g/p OFDM Receiver for GNU Radio. In *ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF)*, pages 9–16, 2013.
- [14] B. Bloessl, C. Sommer, and F. Dressler. Power matters: Automatic gain control for a software defined radio ieee 802.11 a/g/p receiver. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 25–26. IEEE, 2015.
- [15] F. Cali, M. Conti, and E. Gregori. Ieee 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *IEEE journal on selected areas in communications*, 18(9):1774–1786, 2000.
- [16] J. Camp and E. Knightly. Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. In *ACM MobiCom*, 2008.
- [17] K. Chebrolu and A. Dhekne. ESense: communication through energy sensing. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–96. ACM, 2009.
- [18] L. Chen and W. B. Heinzelman. QoS-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):561–572, 2005.
- [19] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 17(1):1–14, 1989.
- [20] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *ACM MobiCom*, 2010.
- [21] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath. Flashback: Decoupled lightweight wireless control. *ACM SIGCOMM Computer Communication Review*, 42(4):223–234, 2012.
- [22] Cisco. Channels and maximum power settings for cisco aironet lightweight access points. [https://www.cisco.com/c/en/us/td/docs/wireless/access\\_point/channels/lwapp/reference/guide/lw\\_chp2/1100\\_chp.html#wp1042809](https://www.cisco.com/c/en/us/td/docs/wireless/access_point/channels/lwapp/reference/guide/lw_chp2/1100_chp.html#wp1042809), 2017.

- [23] E. N. Committee et al. Channel models for hiper-lan/2 in different indoor scenarios. *ETSI EP BRAN 3ERI085B*, 1998.
- [24] X. Dong, N. C. Beaulieu, and P. H. Wittke. Signaling constellations for fading channels. *IEEE Transactions on Communications*, 47(5), 1999.
- [25] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson. A comparison of the hiperlan/2 and ieee 802.11 a wireless lan standards. *IEEE Communications magazine*, 40(5):172–180, 2002.
- [26] M. Duarte and A. Sabharwal. Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 1558–1562. IEEE, 2010.
- [27] M. Ergen, S. Coleri, and P. Varaiya. QoS aware adaptive resource allocation techniques for fair scheduling in ofdma based broadband wireless access systems. *IEEE Trans. on Broadcasting*, 2003.
- [28] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid - the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2012.
- [29] E. Felemban and E. Ekici. Single hop ieee 802.11 dcf analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases. *IEEE Transactions on Wireless Communications*, 10(10):3256–3266, 2011.
- [30] W. Gao, Y. Li, H. Lu, T. Wang, and C. Liu. On exploiting dynamic execution patterns for workload offloading in mobile cloud applications. In *Proceedings of the IEEE 22nd International Conference on Network Protocols (ICNP)*. IEEE, 2014.
- [31] Y. Geng, W. Hu, Y. Yang, W. Gao, and G. Cao. Energy-efficient computation offloading in cellular networks. In *IEEE ICNP*, 2015.
- [32] A. J. Goldsmith and S.-G. Chua. Adaptive coded modulation for fading channels. *IEEE Trans. on Communications*, 1998.
- [33] S. Gollakota and D. Katabi. *Zigzag decoding: combating hidden terminals in wireless networks*. ACM, 2008.
- [34] S. Gollakota and D. Katabi. Physical layer wireless security made fast and channel independent. In *Proceedings of IEEE INFOCOM*, pages 1125–1133. IEEE, 2011.
- [35] J. Gora, K. I. Pedersen, A. Szufarska, and S. Strzyz. Cell-specific uplink power control for heterogeneous networks in lte. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1–5. IEEE, 2010.
- [36] A. Gudipati and S. Katti. Strider: Automatic rate adaptation and collision handling. In *ACM SIGCOMM*, 2011.

- [37] S. Gupta, C. Hunter, P. Murphy, and A. Sabharwal. Warpnet: Clean slate research on deployed wireless networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 331–332. ACM, 2009.
- [38] J. R. Gutierrez-Agullo, B. Coll-Perales, and J. Gozalvez. An IEEE 802.11 MAC software defined radio implementation for experimental wireless communications and networking research. In *Wireless Days (WD), 2010 IFIP*, 2010.
- [39] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan. Towards Wearable Cognitive Assistance. In *Proceedings of ACM MobiSys*, 2014.
- [40] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM*, 2010.
- [41] D. C. Heinz et al. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 39(3):529–545, 2001.
- [42] N. Homma, S. Nagashima, Y. Imai, T. Aoki, and A. Satoh. High-resolution side-channel attack using phase-based waveform matching. In *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems*, pages 187–200. 2006.
- [43] C.-L. Huang and W. Liao. Throughput and delay performance of iee 802.11 e enhanced distributed channel access (edca) under saturation condition. *IEEE Transactions on wireless communications*, 6(1), 2007.
- [44] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, Sept 1952.
- [45] P. Iannucci, K. E. Fleming, J. Perry, H. Balakrishnan, and D. Shah. A hardware spinal decoder. In *ACM/IEEE ANCS*, 2012.
- [46] P. A. Iannucci, J. Perry, H. Balakrishnan, and D. Shah. No symbol left behind: a link-layer protocol for rateless codes. In *ACM MobiCom*, 2012.
- [47] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 301–312. ACM, 2011.
- [48] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 321–332. ACM, 2009.
- [49] J. Jang and K. B. Lee. Transmit power adaptation for multiuser ofdm systems. *IEEE Journal on selected areas in communications*, 21(2), 2003.

- [50] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 317–330. ACM, 2015.
- [51] D. Kivanc, G. Li, and H. Liu. Computationally efficient bandwidth allocation and power control for ofdma. *IEEE transactions on wireless communications*, 2(6):1150–1158, 2003.
- [52] Z.-n. Kong, D. H. Tsang, B. Bensaou, and D. Gao. Performance analysis of IEEE 802.11e contention-based channel access. *Selected Areas in Communications, IEEE Journal on*, 2004.
- [53] J. Kotte, C. Schmeichel, A. Zlocki, H. Gathmann, and L. Eckstein. Concept of an enhanced v2x pedestrian collision avoidance system with a cost function–based pedestrian model. *Traffic injury prevention*, 18(sup1):S37–S43, 2017.
- [54] S. Kumar, S. Gollakota, and D. Katabi. A cloud-assisted design for autonomous driving. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 41–46. ACM, 2012.
- [55] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6th edition, 2012.
- [56] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 rate adaptation: A practical approach. In *ACM MSWiM*, 2004.
- [57] A. T. Le and K. Araki. A group of modulation schemes for adaptive modulation. In *Proceedings of the 11th IEEE Singapore International Conference on Communication Systems*, pages 864–869, 2008.
- [58] L. B. Le, E. Modiano, and N. B. Shroff. Optimal control of wireless networks with finite buffers. *IEEE/ACM Transactions on Networking*, 20(4):1316–1329, 2012.
- [59] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.
- [60] K. Lee, D. Chu, E. Cuervo, A. Wolman, and J. Flinn. DeLorean: using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services (MobiSys)*, 2014.
- [61] Y. Li and W. Gao. Code offload with least context migration in the mobile cloud. In *Proceedings of IEEE INFOCOM*, 2015.
- [62] Y. Li and W. Gao. Interconnecting heterogeneous devices in the personal mobile cloud. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

- [63] Z. Li, Y. Xie, M. Li, and K. Jamieson. Recitation: Rehearsing wireless packet reception in software. In *ACM MobiCom*, 2015.
- [64] Linux Wireless. Minstrel rate control algorithm for 802.11 networks. <https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel>, 2018.
- [65] T. Litman. *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute, 2017.
- [66] Z. Liu, Y. Xin, and G. B. Giannakis. Linear constellation precoding for OFDM with maximum multipath diversity and coding gains. *IEEE Transactions on Communications*, 51(3):416–427, 2003.
- [67] D. López-Pérez, A. Valcarce, G. De La Roche, and J. Zhang. Ofdma femtocells: A roadmap on interference avoidance. *IEEE Communications Magazine*, 47(9), 2009.
- [68] R. Love, R. Kuchibhotla, A. Ghosh, R. Ratasuk, B. Classon, and Y. Blankenship. Downlink control channel design for 3GPP LTE. In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pages 813–818. IEEE, 2008.
- [69] H. Lu and W. Gao. Supporting real-time wireless traffic through a High-Throughput side channel. In *In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2016.
- [70] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee. WiFi-Nano: Reclaiming WiFi efficiency through 800ns slots. In *Proceedings of ACM MobiCom*, pages 37–48. ACM, 2011.
- [71] S. W. Moon, Y. J. Kim, H. J. Myeong, C. S. Kim, N. J. Cha, and D. H. Kim. Implementation of smartphone environment remote control and monitoring system for android operating system-based robot platform. In *Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 211–214, 2011.
- [72] A. Najafizadeh, R. Liscano, M. V. Martin, P. Mason, and M. Salmanian. Challenges in the implementation and simulation for wireless side-channel based on intentionally corrupted fcs. *Procedia Computer Science*, 5:165–172, 2011.
- [73] S. Noda, Y. Saito, and T. Yoshida. Configuration and error ratio performance of M-QAM whose number of signal points is not a power of 2. *Electronics and Communications in Japan*, 90(2):46–57, 2007.
- [74] D. R. Pauluzzi and N. C. Beaulieu. A comparison of SNR estimation techniques for the AWGN channel. *IEEE Transactions on Communications*, 48(10):1681–1691, 2000.
- [75] E. Perahia and M. X. Gong. Gigabit wireless LANs: an overview of IEEE 802.11ac and 802.11ad. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2011.

- [76] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah. Spinal codes. In *ACM SIGCOMM*, 2012.
- [77] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. Secret key extraction from wireless signal strength in real environments. *IEEE Transactions on Mobile Computing*, 12(5):917–930, 2013.
- [78] D. Qiao, S. Choi, A. Jain, and K. G. Shin. Miser: an optimal low-energy transmission strategy for ieee 802.11 a/h. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 161–175. ACM, 2003.
- [79] D. Qiao, S. Choi, and K. G. Shin. Interference analysis and transmit power control in ieee 802.11 a/h wireless lans. *IEEE/ACM Transactions on Networking*, 15(5):1007–1020, 2007.
- [80] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini. Frequency-aware rate adaptation and mac protocols. In *ACM MobiCom*, 2009.
- [81] A. Sahai, G. Patel, and A. Sabharwal. Pushing the limits of full-duplex: Design and real-time implementation. *arXiv preprint arXiv:1107.0607*, 2011.
- [82] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu. Snow: Sensor network over white spaces. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 272–285. ACM, 2016.
- [83] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra. Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces. In *SenSys*. ACM, 2017.
- [84] T. M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for OFDM. *Communications, IEEE Transactions on*, 45(12):1613–1621, 1997.
- [85] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. AccuRate: Constellation based rate estimation in wireless networks. In *Proceedings of USENIX NSDI*, 2010.
- [86] W.-L. Shen, K. C.-J. Lin, S. Gollakota, and M.-S. Chen. Rate adaptation for 802.11 multiuser MIMO networks. *IEEE Transactions on Mobile Computing*, 13(1):35–47, 2014.
- [87] Y. Shen and E. Martinez. Channel estimation in ofdm systems. *Freescale Semiconductor*, 2(6), 2006.
- [88] Z. Shen, J. G. Andrews, and B. L. Evans. Adaptive resource allocation in multiuser ofdm systems with proportional rate constraints. *IEEE transactions on wireless communications*, 4(6):2726–2737, 2005.

- [89] L. Shi, B. Guo, and L. Zhao. Block-type pilot channel estimation for OFDM systems under frequency selective fading channels. In *Proceedings of the International Communication Conference on Wireless Mobile and Computing*, pages 21–24, 2009.
- [90] A. Shokrollahi. Raptor codes. *IEEE transactions on information theory*, 52(6):2551–2567, 2006.
- [91] O. Simeone, Y. Bar-Ness, and U. Spagnolini. Pilot-based channel estimation for OFDM systems by tracking the delay-subspace. *IEEE Transactions on Wireless Communications*, 3:315–325, 2004.
- [92] A. Simonsson and A. Furuskar. Uplink power control in lte-overview and performance, subtitle: principles and benefits of utilizing rather than compensating for sinr variations. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE, 2008.
- [93] P. F. Smulders. Statistical characterization of 60-ghz indoor radio channels. *IEEE Transactions on Antennas and Propagation*, 57(10):2820–2829, 2009.
- [94] H. Sugiyama and K. Nosu. MPPM: a method for improving the band-utilization efficiency in optical PPM. *Journal of Lightwave Technology*, 7(3):465–472, 1989.
- [95] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim. Practical MU-MIMO user selection on 802.11ac commodity networks. In *ACM MobiCom*, 2016.
- [96] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan. 60 ghz indoor networking through flexible beams: A link-level profiling. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, pages 71–84. ACM, 2015.
- [97] Y. Takishima, M. Wada, and H. Murakami. Reversible variable length codes. *IEEE Transactions on Communications*, 43(2/3/4), 1995.
- [98] J. Thomson et al. An integrated 802.11a baseband and MAC processor. In *IEEE International Solid-State Circuits Conference*, pages 126–451, 2002.
- [99] L. Tong and W. Gao. Application-aware traffic scheduling for workload offloading in mobile clouds. In *IEEE INFOCOM*, 2016.
- [100] D. Visani et al. QAM constellation formats for DMT over 1-mm core diameter plastic optical fiber. *IEEE Photonics Technology Letters*, 23(12):768–770, 2011.
- [101] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *ACM SIGCOMM Computer Communication Review*, 39(4):3–14, 2009.
- [102] Q. Wang, H. Su, K. Ren, and K. Kim. Fast and scalable secret key generation exploiting channel phase randomness in wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1422–1430. IEEE, 2011.

- [103] L.-F. Wei. Trellis-coded modulation with multidimensional constellations. *IEEE Trans. on Information Theory*, 1987.
- [104] T. Wild, F. Schaich, and Y. Chen. 5G air interface design based on universal filtered (UF-)OFDM. In *Proceedings of the International Conference on Digital Signal Processing (DSP)*, pages 699–704, 2014.
- [105] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch. Multiuser ofdm with adaptive subcarrier, bit, and power allocation. *IEEE Journal on selected areas in communications*, 17(10):1747–1758, 1999.
- [106] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, 2006.
- [107] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, Q. Zhang, and L. Ni. HJam: Attachment transmission in WLANs. In *Proceedings of IEEE INFOCOM*, pages 1449–1457, 2012.
- [108] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni. Side channel: Bits over interference. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, page 13, 2010.
- [109] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni. Side channel: bits over interference. *IEEE Transactions on Mobile Computing*, 11(8):1317–1330, 2012.
- [110] H. Xu, V. Kukshya, and T. S. Rappaport. Spatial and temporal characteristics of 60-ghz indoor channels. *IEEE Journal on selected areas in communications*, 20(3):620–630, 2002.
- [111] B. Yang, K. Letaief, R. S. Cheng, and Z. Cao. Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modeling. *IEEE Transactions on Communications*, 49(3):467–479, 2001.
- [112] X. Zhang and K. G. Shin. E-MiLi: energy-minimizing idle listening in wireless networks. In *Proceedings of ACM MobiCom*, pages 205–216, 2011.
- [113] X. Zhang and K. G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *Proceedings of IEEE INFOCOM*, pages 3094–3101, 2013.