HARNESSING LOW-PASS FILTER DEFECTS FOR IMPROVING WIRELESS LINK PERFORMANCE: MEASUREMENTS AND APPLICATIONS

By

Alireza Ameli Renani

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science - Doctor of Philosophy

2018

ABSTRACT

Harnessing Low-Pass Filter Defects for Improving Wireless Link Performance: Measurements and Applications

By

Alireza Ameli Renani

The design trade-offs of transceiver hardware are crucial to the performance of wireless systems. The effect of such trade-offs on individual analog and digital components are vigorously studied, but their systemic impacts beyond component-level remain largely unexplored. In this dissertation, we present an in-depth study to characterize the surprisingly notable systemic impacts of low-pass filter design, which is a small yet indispensable component used for shaping spectrum and rejecting interference.

Using a bottom-up approach, we examine how signal-level distortions caused by the trade-offs of low-pass filter design propagate to the upper-layers of wireless communication, reshaping bit error patterns and degrading link performance of today's 802.11 systems. Moreover, we propose a novel unequal error protection algorithm that harnesses low-pass filter defects for improving wireless LAN throughput, particularly to be used in forward error correction, channel coding, and applications such as video streaming. Lastly, we conduct experiments to evaluate the unequal error protection algorithm in video streaming, and we present substantial enhancements of video quality in mobile environments.

Dedicated to my parents, Roya and Ahmad;
and to my wife, Elaheh.

ACKNOWLEDGEMENTS

First of all, I want to thank my adviser, Dr. Guoliang Xing. You helped me become an independent researcher. Thank you for taking a chance on me, I truly appreciate all your support.

To Dr. Abdol-Hossein Esfahanian, I cannot thank you enough for encouraging me to apply for PhD here at MSU, and for all your help and support during this journey. It has been a pleasure to work on projects under your supervision. You have been there whenever I needed any help or guidance. I am truly grateful and honored.

I want to extend my heartfelt thanks to Dr. Richard Enbody. I could not ask for a better mentor. I learned how to be passionate in teaching by observing how you teach, support and lead. I have learned so much from you I cannot express in words. Your office door is always literally and metaphorically open to everyone, and I stopped by your office countless number of times. I am truly honored to have a chance to teach beside you and call you my colleague. Thank you so much for all your mentorship and support.

I want to thank my committee member, Dr. Matt Mutka. I appreciate all your help and support which made my PhD completion much more achievable. I further want to extend my thanks to my committee member, Dr. Hayder Radha. I appreciate your guidance and insights whenever I needed them. I also Want to thank Dr. Jun Huang for his collaborations in my projects.

To my parents, Roya and Ahmad, thank you for always encouraging me to follow my intellectual curiosity my entire life. I have achieved so many personal and professional goals thanks to your never ending encouragements, patience and support. Along with my brother,

Mohammd Javad and my sister, Zahra, I want to thank you all for always being on my side through all the ups and downs in my life. I am truly blessed to have a wonderful family like you.

To all my friends, thank you all so much for always standing by me and making my journey more tolerable. I wish I could name you all. I wish you the best of luck wherever you are.

To my wonderful wife, Elaheh, thank you so much for being with me this entire journey. Meeting you was the best thing that happened in my life. Thank you for being such an amazing partner and standing by me no matter what life threw at us. I am looking forward to the next chapter of our lives together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

1.1 Background and Motivation

With the rapid advance of physical layer (PHY) technology, the implementation of 802.11 transceiver hardware is becoming increasingly complex. From 802.11g-2003 [7] to 802.11n-2009 [8], the number of transistors required to implement the transceiver increased 12.5 times in just six years [20] [65], which is 1.56 times faster than the Moore's rate. This trend is expected to continue in 802.11ac and the upcoming 802.11ad [9] [10]. To cope with this trend, a key trade-off in transceiver design is to balance the performance and cost of hardware implementation. As an example of such trade-offs, implementing a flash analog-to-digital converter (ADC) of n-bit resolution requires $2^n - 1$ comparators. The better the precision of conversion, the larger the package size and the higher the power consumption. Similarly, the gain and stability of a low-noise amplifier (LNA) can be improved by increasing current density, which is however at the expense of higher current draw [29].

The design trade-offs in transceiver hardware are crucial to the performance of wireless systems. However, although the effects of such trade-offs on *individual* analog and digital components are well-studied, their *systemic* impacts beyond component-level remain largely unexplored. The problem is particularly complex in commercial wireless systems, where consumers and researchers are not privy to the proprietary design choices, resulting in

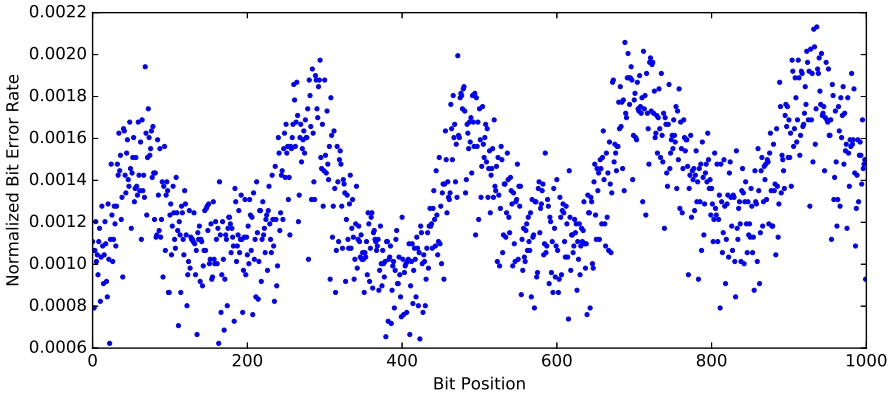

Figure 1.1 Normalized bit error frequency for transmission rate 54Mbps.

hardware-related communication and performance issues at upper-layers that are difficult to diagnose and control [34] [35] [50] [64]. For example, recent work [34] observed that off-the-shelf 802.11 devices suffer from performance degradation due to a twisted bit error pattern shown in Fig. 1.1, where bits transmitted at certain positions of a frame experience significantly higher error rates. This pattern persists even in over-cable communication, implying that the cause is rooted in transceiver hardware. Although a number of protocols have been developed to optimize communication performance based on this pattern [37] [57], the root cause of such error pattern remains unidentified.

The contents of this chapter were adapted from a published conference paper [1].

1.2 Problem Statement and Goals

In order to precisely understand the performance of today's wireless systems, we present an in-depth study to characterize the surprisingly notable systemic impacts of hardware defects caused by component-level design trade-offs. To this end, we take Low-Pass

[1] **"Harnessing hardware defects for improving wireless link performance: Measurements and applications"**. Alireza Ameli Renani, Jun Huang, Guoliang Xing, Abdol-Hossein Esfahanian. INFOCOM 2017 - IEEE Conference on Computer Communications.

Filter (LPF) – a small yet indispensable component used for shaping spectrum and rejecting interference – as an example, and study the impacts of its defects on the performance of off-the-shelf 802.11 systems. Our contribution is three-fold:

1. *Measurement of signal-level distortions caused by LPF defects* Signal-level distortions play a significant role in overall wireless system performance. It is thus crucial to measure and understand such distortions caused by hardware components like LPF. A basic trade-off in LPF design is to balance the steepness of frequency roll-off with implementation complexity. A slow roll-off may distort wanted signal components in the passband, while a steep roll-off requires higher implementation costs. To understand the effects of such trade-off in commercial wireless systems, we present microscopic measurements and models to characterize the signal-level distortions caused by LPF roll-off, and evaluate its significance in practical environments in the presence of multipath fading and mobility.

2. *Characterization of the impacts of LPF defects on link reliability* Using a bottom-up approach, we examine how signal-level defects caused by LPF roll-off propagate to the upper-layers of wireless communication, reshaping bit error patterns and degrading link performance. To this end, we build standard-compliant emulation tools on a software-defined radio to measure and diagnose the impacts of LPF, and quantify the degradation of link reliability caused by LPF defects.

3. *Harnessing LPF defects for mobile video streaming* In addition to measuring and modeling the effects of LPF, we propose algorithms that harness LPF defects for improving mobile video streaming, which entails significant challenge due to fast varying channel conditions. Our approach is driven by the insight that distortions caused by hard-

ware defects are invariant of wireless channel. As a result, although the instantaneous performance of video streaming is highly dependent on channel condition, hardware defects typically result in error patterns that are stable and predictable on mobile links, after the distortion introduced by wireless channel is averaged out after a short-period of movement. Based on this observation, we design an unequal error protection (UEP) algorithm that reorders video frame bytes based on the error pattern caused by LPF defects, which substantially enhances video streaming performance in mobile environments.

Chapter 2

Related Work

2.1 Cross-Layer Design

Numerous cross-layer protocols have been developed for various wireless networks. Channel information is used in [55] to provide unequal error protection for upper-layer traffic flows. The fact that decoded erroneous symbols are still a good "approximation" of the original transmitted symbol is the foundation in designing a UEP algorithm. [23] leverages CSI to recover partially corrupted FEC groups and facilitate FEC decoding. EffSNR [23], AccuRate [56] and SoftRate [59] exploit various PHY hints to facilitate rate adaptation. These cross-layer designs exploit PHY information at the link layer or above to improve wireless performance. TiM [60] or time-line modulation, adds time dimension into amplitude-phase scheme for a three-dimensional modulation scheme to fill the gap of not having a one-to-one mapping from channel conditions to modulation schemes.

2.2 Error Estimation and Recovery

Maranello [36] and PPR [40] reduce unnecessary retransmissions by only retransmitting bit blocks that contain errors. ZipTx [45] sends parity bits to correct bit errors. SOFT [62] allows multiple receivers to cooperatively repair corrupted packets. Bit error rate

estimation is central to various wireless protocols. Previous solutions either use error estimation codes (EEC) [28], or PHY hints for bit error rate (BER) estimation. However, EEC requires transmitter cooperation for code embedding, while soft-values are not available on many production wireless NICs. Han *et al.* [34][35] identified the bit error pattern using experimental results and hypothesize their cause for upper-layer applications.

2.3 Video Streaming

There is a large body of work on video coding and video streaming to increase the throughput, quality and performance of the video streaming. SoftCast [39] broadcasts video without specifying a data rate and each receiver obtains a video quality depending on its channel quality. Recitation [44] employs CSI to predict error-prone bit positions and expose them to upper layers. FlexCast [19] provides a video codec to increase video streaming performance. piStream [63] estimates the bandwidth by monitoring the physical layer resource allocation and enhances adaptive video streaming over LTE by utilizing an agile video rate adaptation. A-HDAVT [66] is an adaptive hybrid digital-analog video transmission scheme that is designed for video streaming in mobile networks. The study combats channel fading and improves the system performance unlike most previous studies that improve the system only in unrealistic Gaussian channels.

2.4 Vehicular Communication

VANETs are about to be deployed on a large scale, hence recent papers studied different aspects of it, from modeling and simulations to field studies and physical layer analysis. Physical layer effects are very hard to model and there is a need to understand the performance of IEEE802.11p receivers.

Bloessl et al. took the first steps toward an open source 802.11p stack by implementing a complete OFDM transceiver and extending it to a simulation and experimentation framework for 802.11p [25] [26]. They validated the implementation by testing it using multiple wifi devices and two 802.11p prototypes, Cohda MK2 and Unex DCMA-86P2. They further conducted a Field Operational Test (FOT) using a modified 802.11a chipset as the transmitter and an Software Defined Radio (SDR) as the receiver [24]. They collected traces in an hour long test over 56 km in city, rural, freeway and highway environments. They further evaluated different receive algorithms on the collected data. Another SDR implementation of IEEE 802.11p is presented in [47]. The authors focus on power efficiency of the implementation which is prohibitive in mobile environments.

Field trials for studying DSRC in VANETs can be very expensive specially when the number of vehicles involved is high. Therefore, many studies used network or physical layer simulations. The simulators often abstract details of the other layers or even do not consider them at all, which is problematic. Recent study proposed a solution by integrating IEEE 802.11p physical layer simulator with NS-3 which is a discrete-event network simulator[49].

Bernado et al. modeled and simulated the physical layer of 802.11p and studied the effect of many factors on the channel estimation performance such as strength of diffuse components, Line-Of-Sight (LOS) and SNR [22]. Sassi et al. modeled and simulated IEEE

802.11p PHY and evaluated the OFDM transmission performance by studying the affect of various parameters on Bit Error Rate (BER) versus SNR[54].

Qualcomm Research [12] has introduced Cellular V2X (C-V2X) technology which is designed to connect vehicles to vehicles, infrastructure, pedestrians, network and basically everything (V2X). It can coexist with 802.11p and its trials start in 2017 with commercial solutions.

IEEE 802.11p specifies four Spectrum Emission Masks (SEMs) which introduces a challenge for filtering signals in 802.11p to meet the standard. Recent work investigated multiple techniques and proposed a method to relax the constraints of radio frequency front-end such that implementation of 802.11p system would be possible by using 802.11a hardware [52].

Experimental measurements and studies of WAVE started long before IEEE 802.11p was published in 2010. Kukshya et al. conducted various experiments using modified 802.11a chipsets [43]. They considered line-of-sight scenarios with no vehicle traffic, moderate vehicle traffic and high vehicle density in parking lot, street and freeway environments. They studied the effect of SNR on BER and also the effect of distance on SNR of the received packets.

Based on small differences between IEEE 802.11p and other IEEE 802.11 standards, [58] presents an approximation of 802.11p standard using commercial off-the-shelf 802.11a hardware and some software adjustments. Using the proposed implementation makes it possible to use cheap 802.11a hardware compared to rather expensive 802.11p prototypes for VANET research. It can also turn available WLAN testbeds into valuable VANET research testbeds. The most important drawback of this work and many other similar implementations is that not all parameters of 802.11p can be modified on a 802.11a hardware. For example, they used 20 MHz channels instead of 10 MHz channels, which means parameters

such as symbol duration, guard time and subcarrier spacing do not represent a real 802.11p hardware.

Most of the prior work in this field focus on physical layer information to increase the communication performance while we study and utilize the hardware defects to create our state of the art unequal error protection algorithm which can improve communication performance in any scenario that data is divided into different levels of importance such as error estimation and recovery, video streaming and vehicular communication.

Chapter 3

Background

In this chapter we present a brief background on 802.11 architecture, OFDM basics and Low Pass Filter which are the subjects of our study.

3.1 802.11 Primer

Fig. 3.1 shows the high level architecture of 802.11. The transmitter first scrambles the original bit-stream and randomizes the data bits. Then the data bits are coded using convolutional encoding. Orthogonal Frequency Division Multiplexing (OFDM) divides the 20 MHz 802.11g channel bandwidth into 64 312.5 KHz subcarriers. 802.11g uses 4 pilot subcarriers, 12 null subcarriers and the remaining 48 subcarriers for data. The center or "DC" subcarrier is not used. Out of the remaining 11 subcarriers, 6 of them are in the beginning of the channel and 5 of them are at the end. The transmitter interleaves and maps the coded bits to subcarriers and also modulates them into symbols. The combined signal of all subcarriers is called OFDM symbol. Each received OFDM symbol undergoes the reverse operations to recover the original data.

According to Part 11 of IEEE Standard for Wireless LAN MAC and PHY Specification, the transmitted spectrum using 20 MHz channel spacing should fall within the spectral mask described below.

The following items and Fig. 3.2 present the spectral mask that the signal should fall

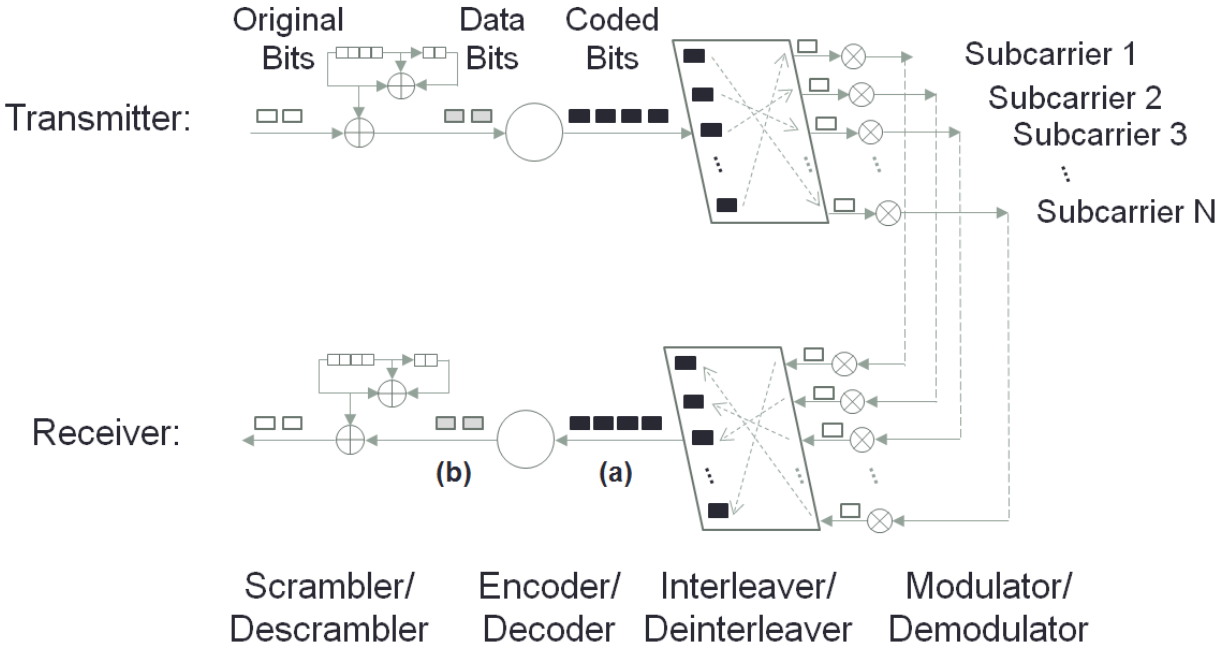

Figure 3.1 802.11 architecture at a high level. Figure illustrates scrambling, coding and interleaving in the architecture.

within. The units are in dBr which is dB relative to the maximum spectral density of the signal. The transmitted spectrum should have a 0 dBr bandwidth not exceeding 18 MHz centered at f_c , the channel center. -20 dBr at 11 MHz frequency offset. -28 dBr at 20 MHz frequency offset. The maximum of 40 dBr and 53 dBm/MHz at 30 MHz frequency offset and above.

3.2 Low-Pass Filter

Low-pass filter is widely used in wireless systems for reshaping spectrum and rejecting interference. An ideal low-pass filter passes signals with frequencies lower than a specified cutoff frequency, and eliminates other signals with higher frequencies. The frequency response of the ideal LPF is rectangular, and its impulse response is a Sinc function presented

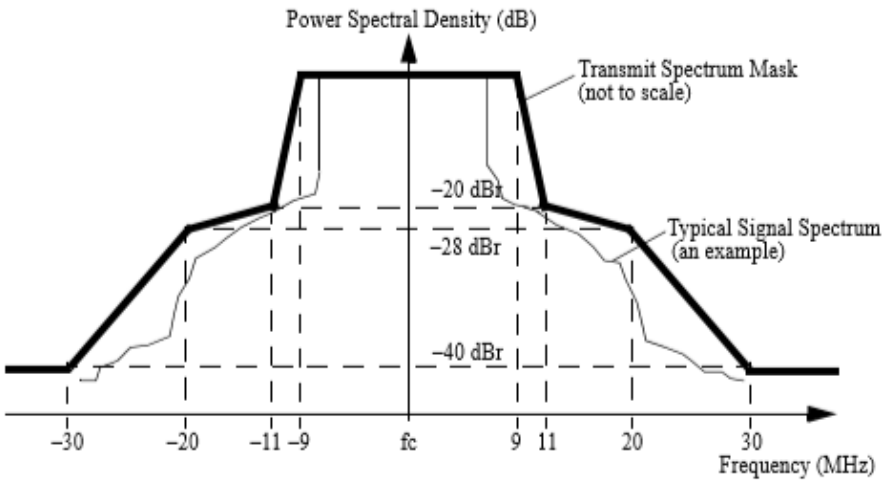

Figure 3.2 Transmit spectrum mask for 20MHz transmission [7].

in the Fig. 3.3, whose support region extents to all past and future time. As a result, the ideal LPF is impossible to realize, because it requires the knowledge of signals in the infinite future and past.

The practice of designing LPF is to approximate the ideal LPF by truncating and windowing the infinite impulse response, which results in a *transition band* that exhibits a *roll-off* between the passband and the stopband. A filter is characterized by its cutoff frequency and rate of frequency roll-off which is expressed in dB/decade or dB/8ve. Signal attenuation after cutoff frequency depends on the design of the filter, specifically the *order* of the filter. Fig. 3.4 illustrates the trade-off in designing LPF. As shown in the figure, high-order filters have steeper roll-offs, resulting in less distortion of wanted signal in the passband, which is however at the cost of higher implementation complexity.

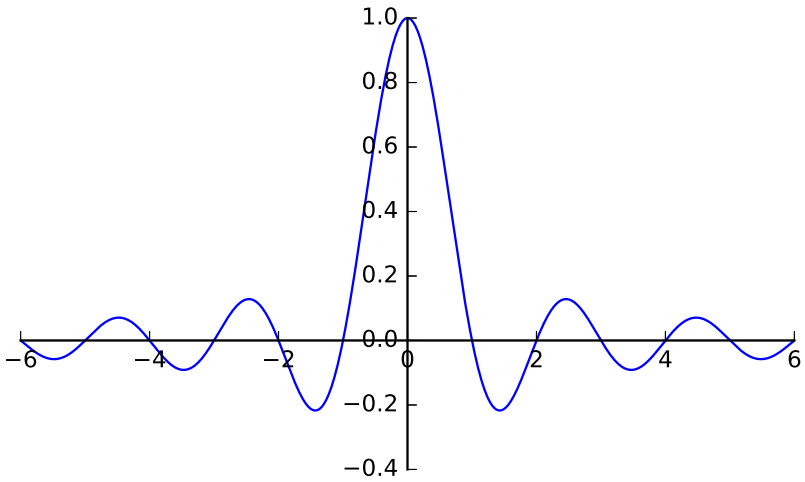

Figure 3.3 Impulse response of Rect function, Sinc(x).

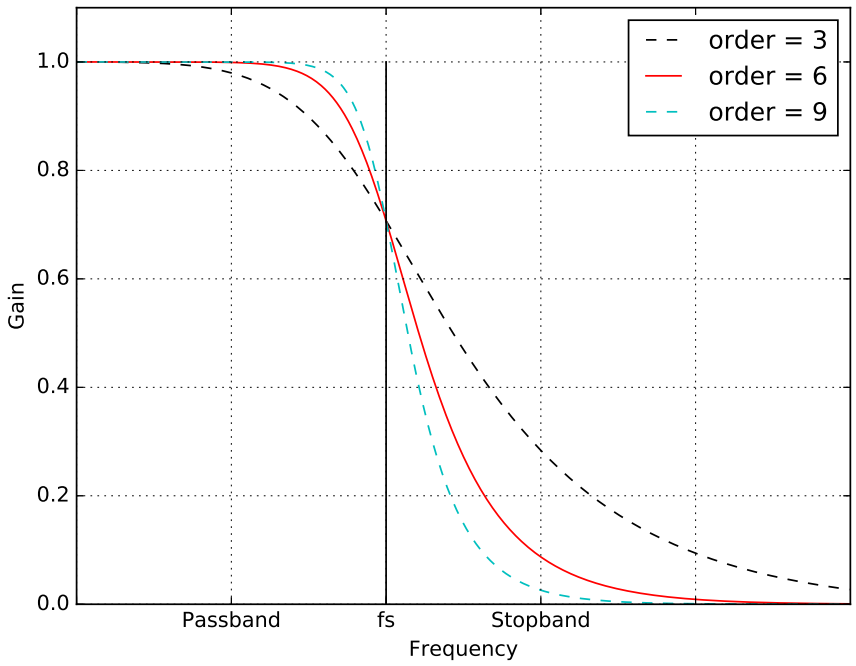

Figure 3.4 Effects of filter order on roll-off: Higher order filters have steeper transitions and smaller transition bands.

Chapter 4

The Impacts of LPF Defects Under the Microscope

In this chapter, we present microscopic measurements and models to characterize the surprisingly notable impacts of LPF design on the performance of today's 802.11 systems. Using a bottom-up approach, we demonstrate how signal-level defects induced by LPF roll-off propagate to the upper layers of wireless communication, reshaping bit error patterns and degrading link performance.

The contents of this chapter were adapted from a published conference paper [1].

4.1 Signal-Level Characterization

In this section, we characterize the signal-level defects caused by LPF design. Specifically, a basic trade-off in LPF design is to balance the steepness of frequency roll-off with implementation complexity. As discussed before, a slow roll-off may distort wanted signal components in pass-band, while a steep roll-off requires higher implementation cost. We demonstrate the effect of roll-off induced signal distortion in different off-the-shelf chips, and evaluate its significance in practice in the presence of multi-path fading and mobile

[1] **"Harnessing hardware defects for improving wireless link performance: Measurements and applications"**. Alireza Ameli Renani, Jun Huang, Guoliang Xing, Abdol-Hossein Esfahanian. INFOCOM 2017 - IEEE Conference on Computer Communications.

environments.

4.1.1 Methodology

We characterize LPF design by measuring the frequency roll-off in the output signal of off-the-shelf 802.11 chips. Existing measurement tools have several limitations for this purpose. General-purpose spectrum analyzers can measure the power spectral density of received signal, but cannot differentiate between signals of different transmitters. Therefore, in uncontrolled setups, their measurements can be polluted when the target signal mixes with noise or interference. The firmwares of a few 802.11 chips can provide channel state information (CSI) that characterizes signal-level distortions [33]. However, the reported CSI is subject to the manipulation of CSI calibration algorithms, thus cannot accurately measure frequency roll-off. The GNURadio/USRP platform [6] can be programmed to analyze raw 802.11 signals, but its results can be distorted by various processings along the receiving pipeline of the USRP hardware.

To address these limitations, we carefully examine signal distortions introduced by various analog and digital components of USRP, and then insert a compensation filter to correct distortions before feeding the signal to the 802.11-compliant receiver. Fig. 4.1 shows the architecture of the receiver pipeline used in our measurements, which uses a daughterbord of SBX. By checking the datasheet of each analog component of SBX, we find that it introduces ignorable distortion to the 802.11 signal on a 20 MHz channel. For example, the LPF of SBX operates on a passband of 40 MHz. Thus the effect of its roll-off on the 20 MHz channel of 802.11 is non-significant.

We find that the major source of distortion inside the receiver is caused by the digital down converter (DDC) of USRP, where the output of the ADC (100 million samples per

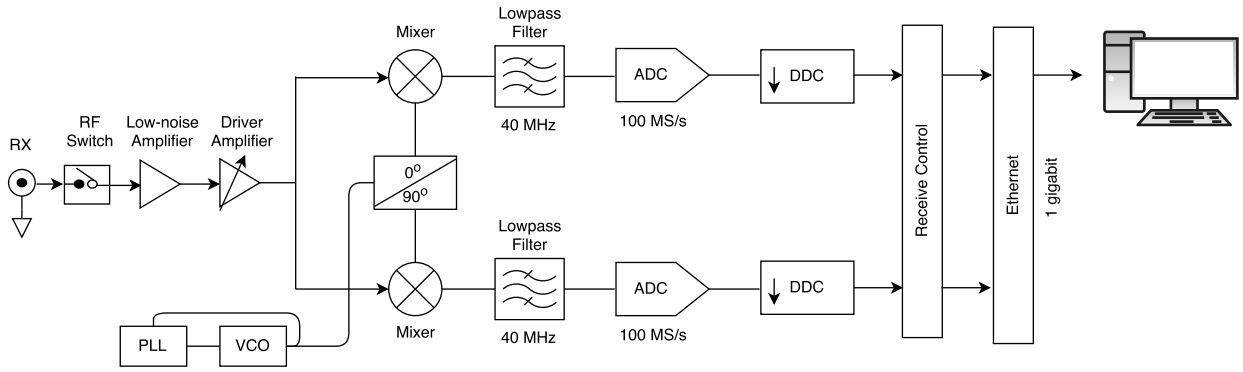

Figure 4.1 Architecture of the receiver front-end, consisting of a USRP2 and an SBX daughterboard.

second (MS/s)) is down-sampled by a factor of 5 in order to generate a 20 MHz signal. To suppress the aliasing caused by down-sampling, the DDC incorporates a 4-stage cascaded integrator-comb (CIC) filter. Its magnitude response is given by

$$|H(f)| = |\frac{\sin(\pi f)}{\sin(\frac{\pi f}{R})}|^4 \approx |\frac{R\sin(\pi f)}{\pi f}|^4 \tag{4.1}$$

where R is the rate change factor. The magnitude response is poor when R is odd, resulting in significant distortion to the received 802.11 signal. Fig. 4.2 illustrates the effects of the CIC filter on the received signals. As shown in the figure, the CIC filter causes a power degradation as high as 18 dB at the border of the 802.11 20 MHz channel, and results in a power loss of approximately 12 dB at the right-most and left-most data subcarriers. To correct this distortion, the magnitude response of the compensation filter must be the inverse of Eq. (4.1), which can be computed as

$$|G(f)| \approx \left|\frac{\pi f}{R\sin(\pi f)}\right|^4 \tag{4.2}$$

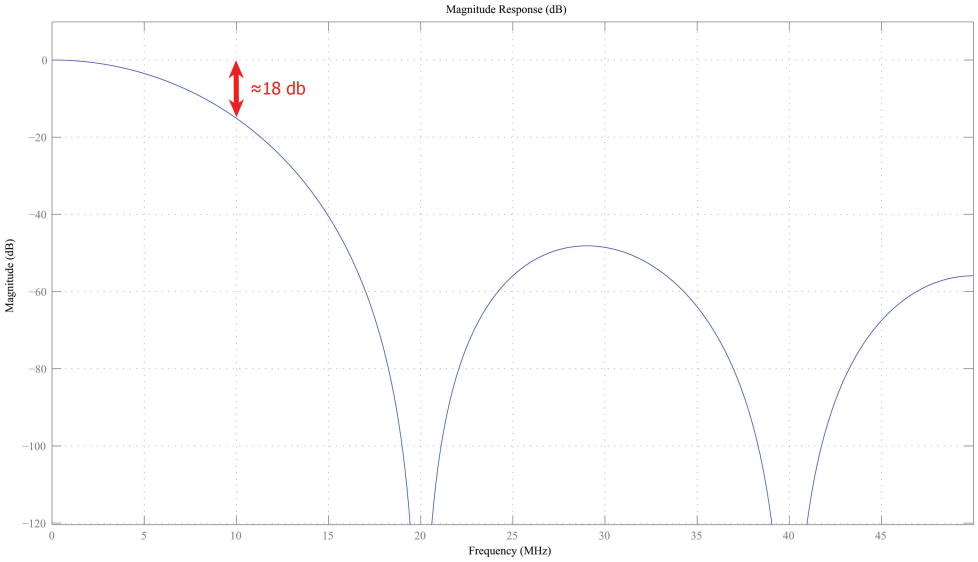

Figure 4.2 Magnitude response of CIC filter for the USRP2.

We then compute the filter coefficients by applying an inverse Fourier transform to Eq. (4.2), and apply the filter on the signal received from USRP to compensate signal loss.

After compensating for the distortion caused by CIC filter, we measure frequency roll-off using the long preamble of each received packet. Specifically, each long preamble of an 802.11 consists of known symbols of equal power transmitted on all subcarriers. As a result of frequency roll-off, symbols transmitted on subcarriers close to the channel border may have lower power. To quantify the power loss caused by roll-off, we first correct the frequency offset between the transmitter and the USRP, and then normalize power measurements for each subcarrier by computing the ratio with respect to the maximum power (measured in dB), which can be expressed as

$$\overline{p_i} = 10log_{10}\frac{p_i}{\frac{1}{n}\sum_{k=1}^{n} p_k} \tag{4.3}$$

where n is the number of subcarriers and p_i is the power of symbol transmitted on subcarrier

i.

4.1.2 Measuring Frequency Roll-Off

To obtain precise measurements of the frequency roll-off, we first conduct experiments in a controlled setup where transmitter and receiver are connected using a cable to exclude the interference and distortion induced by wireless channel. Specifically, the 802.11 chip is associated with an off-the-shelf access point (AP). The output signal of the 802.11 chip is split into two ways using an RCA signal splitter. One way of the signal is fed to an external antenna, which allows the 802.11 chip to communicate with the AP. The other way is connected with an USRP2 via cable, such that the effect of channel fading is excluded in our measurement, allowing us to focus on the effect of hardware-induced distortions. A step attenuator is used to reduce the transmitter power by 10 dB, thus minimizing the risk of any damage to the USRP2. We have checked the datasheets of the signal splitter and the step attenuator to make sure they introduce ignorable distortions to signal. Fig. 4.3 shows the setup of our measurement.

We measure the frequency roll-off induced by LPF in different laptops and access points. These devices use 802.11n chips manufactured by Broadcom, Qualcomm and Intel, respectively (three 802.11 chip vendors that share a majority of the global market). The BCM43xx series is widely used in mobile devices, including Apple MacBooks and iPhones, while AR92xx chips are widely used in 802.11n access points of major vendors such as D-Link, Linksys, TP-Link, and Netgear. The Intel 53xx series are popular in laptops, and are widely used in academic research thanks to its capability of reporting channel state information [33] [38] [46].

Fig. 4.4 shows the raw and compensated transmitter power spectral densities (PSDs)

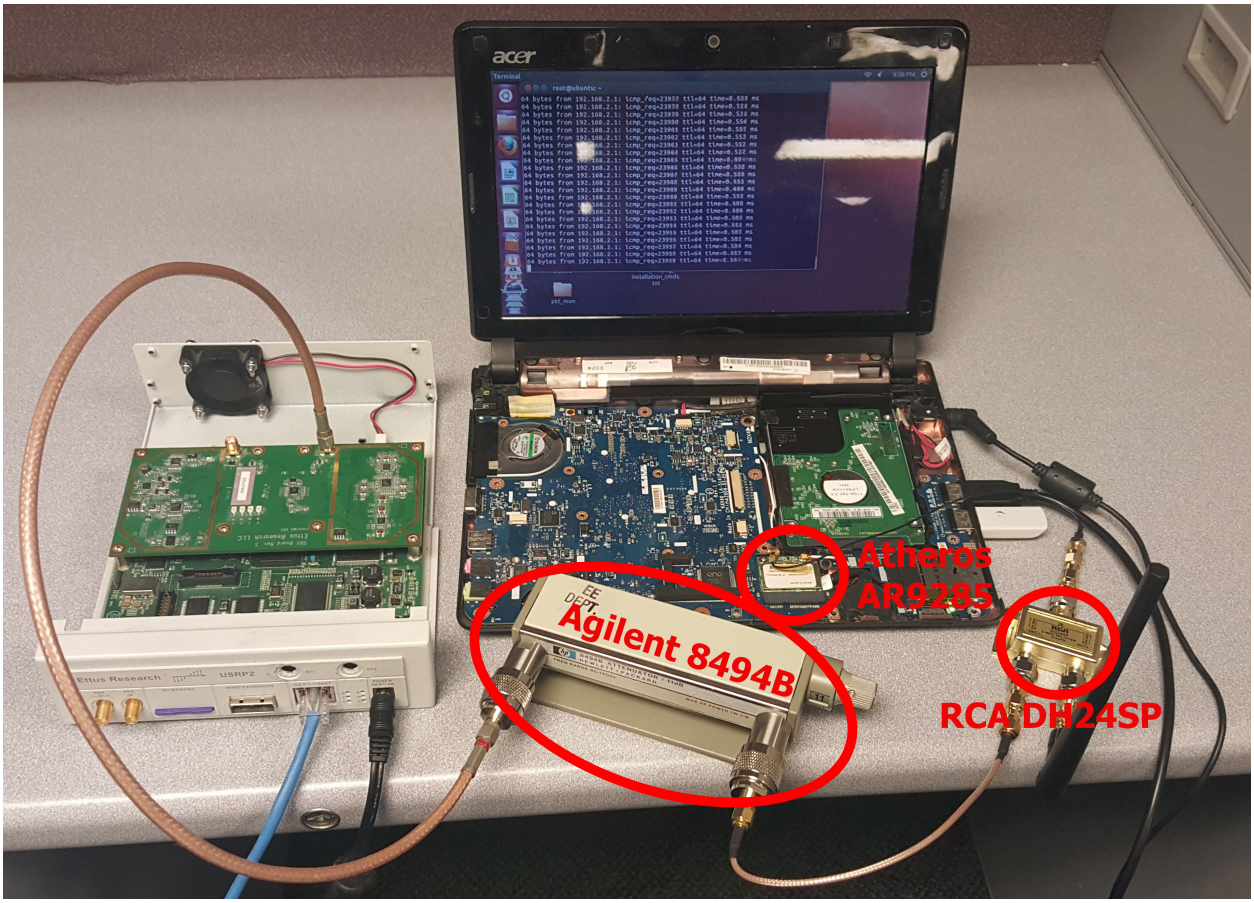

Figure 4.3 Setup of over-cable measurements using an USRP and a 2×2 MIMO AR9285 chip. The AR9285 is connected to an Agilent 8494B attenuator through an RCA DH24SP 2-way signal splitter, and the signal was measured from the main antenna port.

measured using five 802.11 chips, including Atheros' AR2425 and AR9285, Broadcom's BCM4312 and BCM4331, and Intel5300. Roll-off in power spectral density caused by LPF design is observed in all of the chipsets we studied. We also observe that signal power loss at the channel border is different across different chips. For example, the AR2425 has a loss of 2 dB on subcarrier -26, while BCM4312 shows a 4 dB loss on the same subcarrier. Such difference is probably caused by the different design of LPF across chips. As we expected, the effect of roll-off can be clearly observed on symbols received at border subcarriers. Moreover, the CIC filter of USRP's DDC introduces significant additional distortions, which demonstrates the necessity of careful compensation. In particular, the power degradation caused

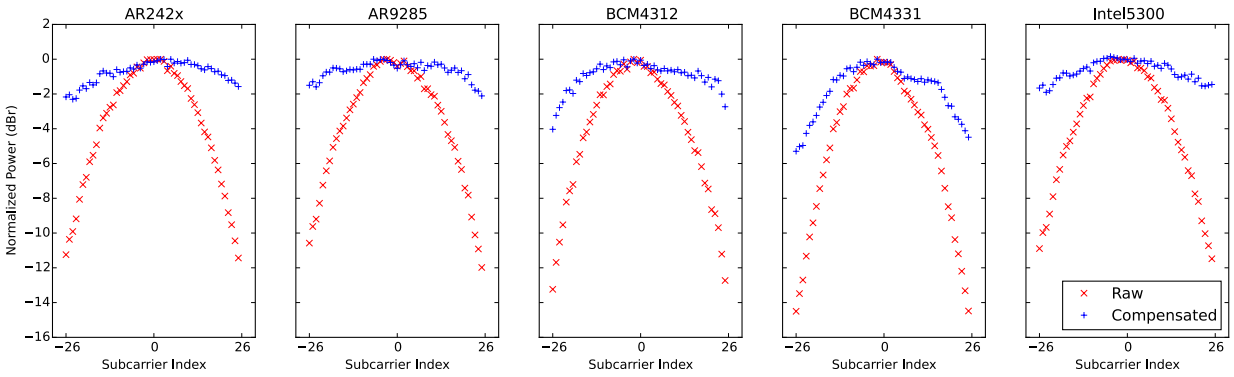

Figure 4.4 Transmitter power spectral densities of different 802.11 chipsets measured before and after compensation for CIC roll-off.

by the roll-off of the transmitter's LPF can be clearly observed even after compensating the distortion of CIC filter. As an example, Fig. 4.5 compares the power loss on subcarriers -26 and +26 (border subcarriers) with that on subcarriers -1 and +1 (center subcarriers). As shown in the figure, border subcarriers suffer up to 5 dB additional power loss because of the roll-off of LPF.

The measurements shown in Fig. 4.4 and Fig. 4.5 only account for the loss caused by the transmitter's LPF. In practice, LPF in the receiver may introduce additional degradations, which further distorts the PSD of received signal. Unfortunately, measuring the distortions caused by the receiver's LPF is not feasible, so we assume that the receiver's LPF roll-off is the same as that of the transmitter. Fig. 4.6 depicts the projected power loss on subcarrier 26 for different transmitter and receiver pairs. The power loss of border subcarriers can be as high as 10 dB.

4.1.3 Significance in Practical Environments

We have shown that the distortion caused by LPF roll-off is significant in a controlled setup where the effect of channel fading is excluded. In the following, we study the effect

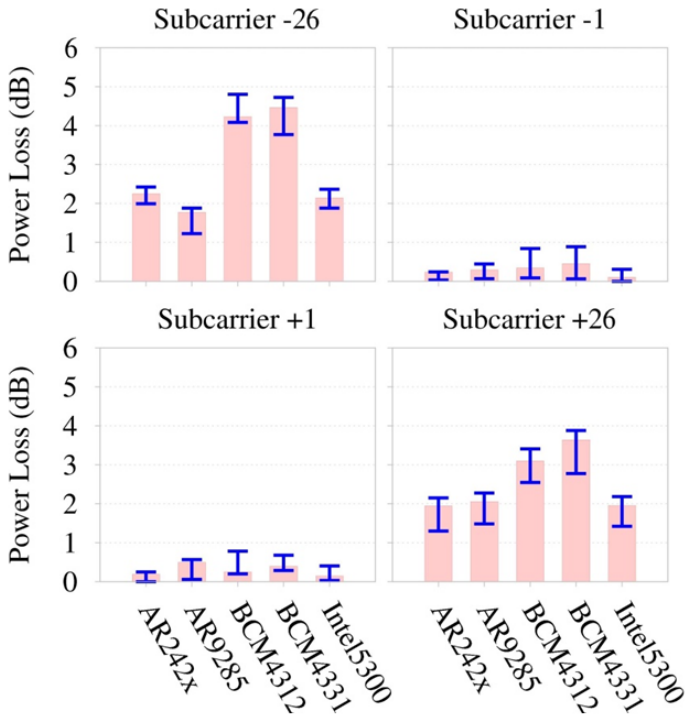

Figure 4.5 Power losses caused by transmitter LPF on border subcarriers (subcarriers -26 and +26) and center subcarriers (subcarriers -1 and +1). Error bars show the minimum and maximum measurement results.

of LPF roll-off in practical environments in the presence of multipath fading and device mobility which causes frequency-selective fading and fast channel variation, respectively. Our experimental setup consists of a client associated with an AP, which use AR2425 and BCM4331, respectively. Another netbook equipped with Intel5300 is employed to measure the PSD of signals.

To study the effects of LPF roll-off in the presence of multipath fading, we deploy the client in a position outside of the AP's line-of-sight, with a distance about 15 feet between them. The client and the AP are kept stationary during measurement. Fig. 4.7 shows the PSD of the AP's signal measured using the netbook equipped with Intel5300 at three randomly selected positions inside an office building. As shown in the figure, the PSDs of received signals differ significantly when measuring at different positions. The deepest fading

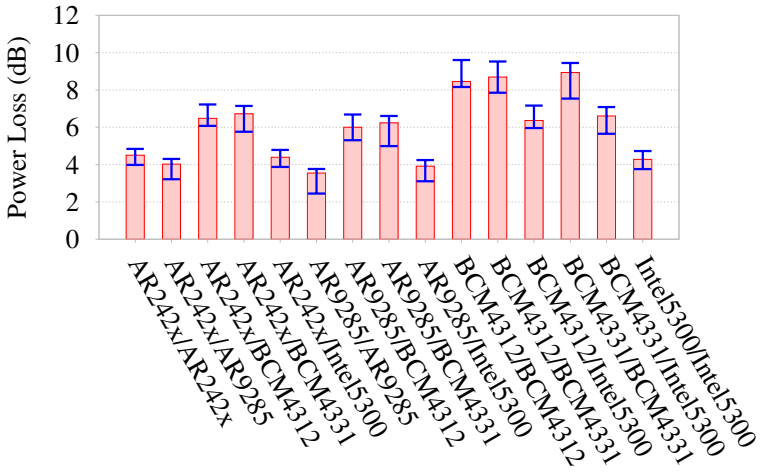

Figure 4.6 Projected power losses on subcarrier 26 for different transmitter and receiver pairs. The results were computed based on the measurement shown in Fig. 4.5.

ranges from 12 dB to 30 dB, which are observed at different subcarriers. In comparison, power loss at border subcarriers becomes less significant. Fig. 4.8 illustrates the CDFs of symbol power measured on channel border (i.e., subcarrier -26 and +26) and channel center (i.e., subcarrier -1 and +1) during the same experiment. In particular, we observe that the power losses are similar on border and center subcarriers. The results indicate that in most cases, the effect of LPF roll-off is overwhelmed by the distortion caused by multipath fading.

To study the effects of LPF roll-off in the presence of device mobility, we conduct experiments using the same devices, and have the client move at a walking speed of about 3 mph. Fig. 4.9 shows the average PSD of the AP's signal when the number of measurement packets increases. The figure shows that, upon receiving 500 packets, the average power losses on border subcarriers increase to 3 dB and 3.5 dB, respectively, while other subcarriers experience lower power losses. The result indicates that, the LPF-induced roll-off becomes significant in the average PSD as the number of received packets increases. This is because of the fast variation of wireless channel, which cancels the effects of multipath fading in the average of PSD after a short period of device movement. Fig. 4.10 shows how the average

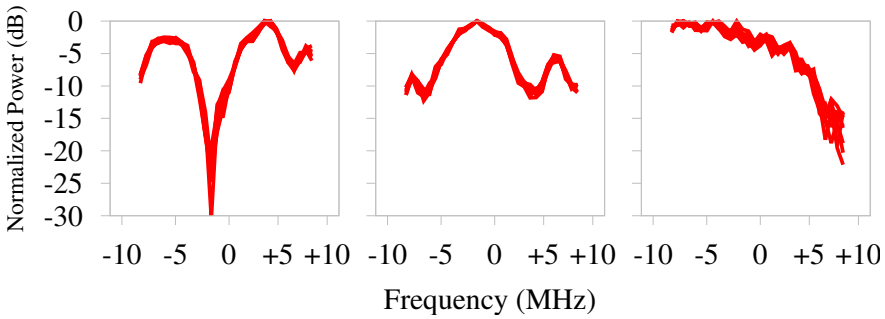

Figure 4.7 Transmitter spectral densities measured at three randomly selected static links within a single office building. The LPF-induced roll-off became insignificant as a result of the severe distortion caused by multipath fading.

symbol powers converge with time. At the beginning of the measurement, both border and center subcarriers exhibit similar power losses. After receiving 1500 packets, the effect of LPF roll-off can be clearly observed, as symbols received on border subcarriers suffer a higher power loss than that received on center subcarriers. After 3000 packets, the average power loss on center and border subcarriers converge to lower than 0.5 dB and above 2 dB, respectively.

4.1.4 Summary

In summary, our signal-level measurements show that:

- In off-the-shelf 802.11 devices, the design trade-off of LPF causes notable signal-level distortions, where subcarriers close to channel border suffer higher power loss due to LPF roll-off.

- In the presence of multipath propagation, the effects of frequency-selective fading overwhelms the impacts of LPF roll-off on static links.

- In mobile environments, the effects of frequency-selective fading can be averaged out after a short-period of movement, making LPF roll-off the dominant cause of signal

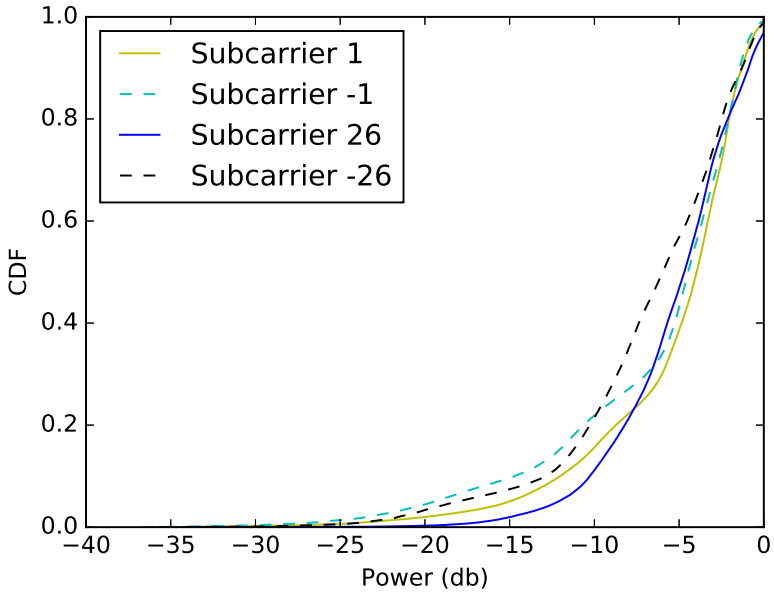

Figure 4.8 CDFs of power losses measured in the presence of multipath fading. The power losses at channel center (i.e., subcarrier +1 and -1) and channel border (i.e., subcarrier +26 and -26) are similar, as the effect of LPF roll-off is overwhelmed by multipath fading.

distortion.

4.2 Impacts on Upper Layers

In the following, we study how signal-level distortion caused by LPF roll-off propagates to the bit-level and link layer, reshaping bit error patterns and degrading link reliability.

4.2.1 From Signal-Level to Bit-Level

Modern 802.11 adapters adopt OFDM in PHY, which transmits data on different subcarriers. To understand how LPF induced signal-level defects propagate to the upper-layers of wireless communication, Fig. 4.11 gives a simple example and demonstrates how LPF roll-off affects the data transmitted on different subcarriers. Specifically, when the effect of LPF roll-off is significant, subcarriers at channel border will experience higher power losses,

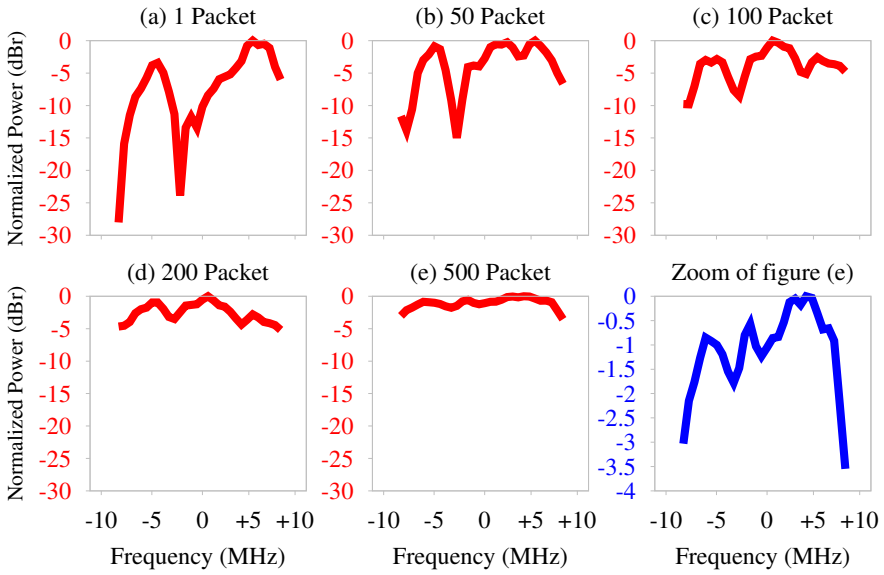

Figure 4.9 The average PSD measured on an 802.11 receiver moving at walking speed. LPF-induced roll-off becomes increasingly pronounced as the number of received packets increases. Roll-off is the most significant error source in the average PSD measured using 500 packets.

leading to lower SNRs and higher error rates, which will result in a periodic error pattern that repeats itself every OFDM symbol.

We note that although other components in the transceiver pipeline of 802.11 may also affect communication performance, the impact of roll-off induced signal distortion remains significant at bit-level. Specifically, in 802.11, data bits are first scrambled, channel coded, and interleaved before transmitting on different subcarriers using OFDM. As we discussed earlier, LPF roll-off will introduce a periodic error pattern to received coded bits. After demodulation, coded bits are first de-interleaved, where positions of coded bits are changed pseudo-randomly. According to 802.11 standard, the interleaving depth equals OFDM symbol size, and the interleaving map remains the same across all OFDM symbols. Therefore, the effect of interleaving is reshaping rather than eliminating the periodic error pattern of coded bits. During channel decoding, whether a data bit can be correctly decoded or not depends on the errors of coded bits. Therefore, errors in the output of channel decoder is

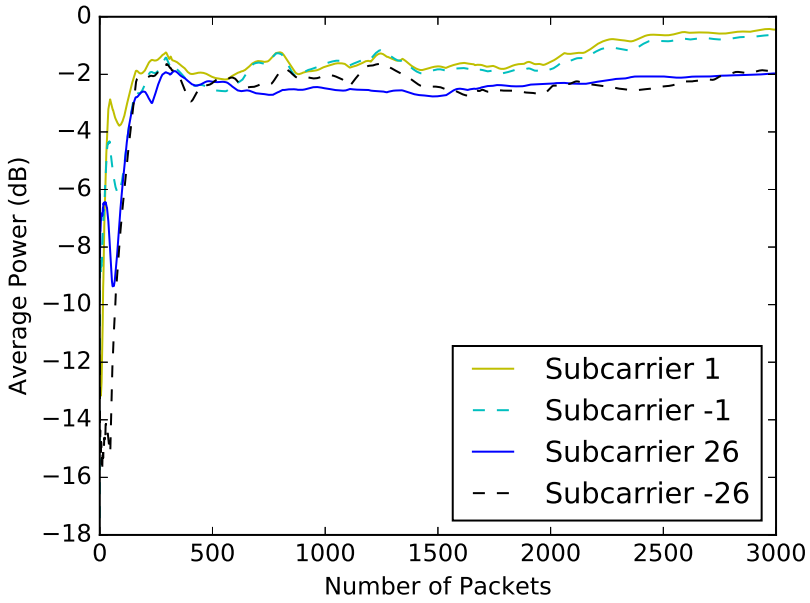

Figure 4.10 The average of power losses measured at channel border (i.e., subcarrier +26 and -26) and channel center (i.e., subcarrier +1 and -1) as the number of received packets increases.

determined by the error pattern of coded bits. Finally, the de-scrambler XORs decoded bits with a pre-defined pseudo-random sequence, which does not affect bit errors. As a result, the periodic error pattern caused by LPF roll-off will be transformed and observed at bit-level.

In the following, we first conduct measurements to confirm the existence of periodic bit error pattern, and then examine its relation with LPF design based on simulations with a standard-compliant software transceiving pipeline.

4.2.2 Measuring Bit Error Patterns

In this section, we conduct experiments to examine the bit error rate to study the effect of LPF roll-off in bit-level using Commercial Off-The-Shelf (COTS) 802.11 devices. We modified an ath9k which is a free and open source software (FOSS) wireless driver for Atheros IEEE 802.11n chipsets to report packets with CRC errors. That makes AR9285

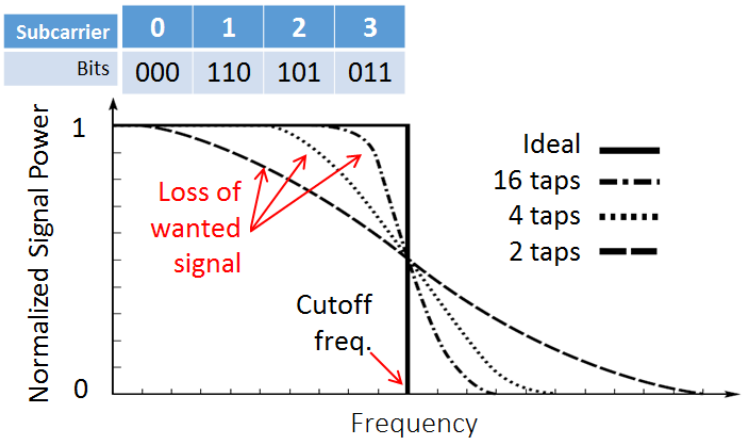

Figure 4.11 Bits close to the border suffer from in-band signal loss.

report corrupted packets which allows us to use the device as receiver for measuring bit error patterns. Other devices are employed as transmitters. We set up a Virtual Access Point (VAP) on the receiver so we can control the transmission power and transmission rate. We examine the effect of the following factors in the study of bit-level patterns:

- **Transmitter Chipset**. Our experiment setup consists of four devices equipped with AR9285, AR2425, AR9001, and Intel5100.
- **Transmission Power**. We conduct experiments using a wide range of transmission powers, from minimum to maximum allowed on a chipset.
- **Transmission Rate**. We repeated the experiments using all the 802.11g rates.
- **Distance**. We conduct the experiments over a wide range of transmitter to receiver distances.
- **Payload**. We use different payloads such as loading all the bytes with 0xff, 0x00, and also with random values.

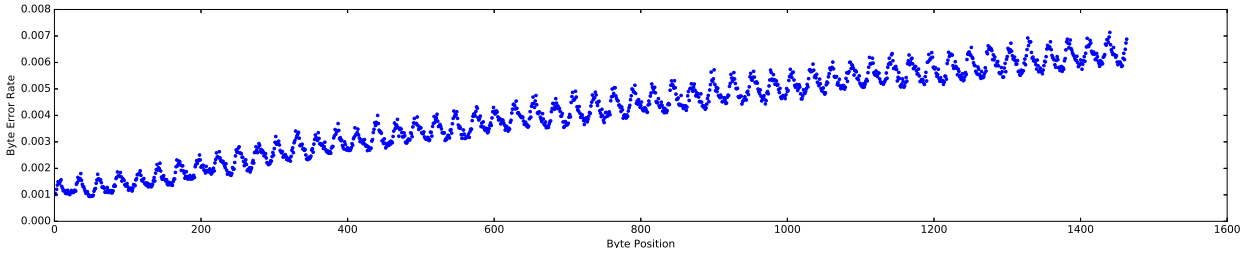

Figure 4.12 Byte error rates for transmission rate of 54Mbps. The vertical axis represents the normalized bit error rate and the horizontal axis represents the byte position within the payload.

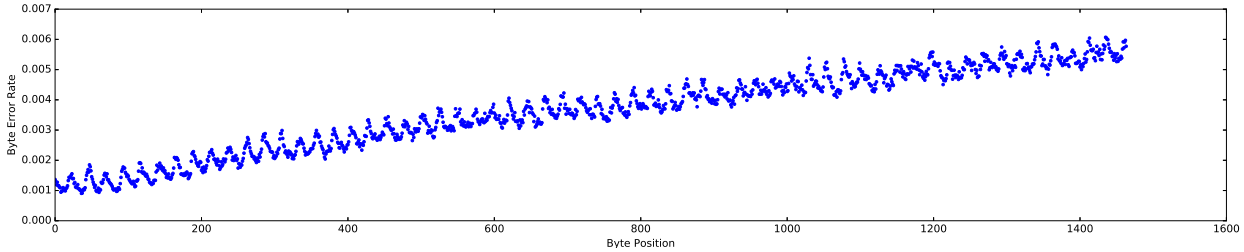

Figure 4.13 Byte error rates for transmission rate of 48Mbps. The vertical axis represents the normalized bit error rate and the horizontal axis represents the byte position within the payload.

- **Mobility**. We conduct the experiments using stationary and mobile transmitter and receiver.

We observe a periodic bit error pattern when the receiver receives enough errors in conducting experiments using all the above factors. Fig. 4.12 and Fig. 4.13 plot byte error rates of 100,000 packets transmitted by AR2425 device over a stationary link with transmission rates of 54Mbps and 48Mbps respectively. The error rates of 8 bits are averaged to a byte error rate to decrease the number of data points for plotting the entire payload. We observe that bits at certain positions experience significantly higher error rates. Similar bit error patterns are also observed on other devices in all configurations.

In most of the experiments, the rate of the bit errors also increased in proportion to the position of the bits in the payload. Han *et al.* referred to this pattern as slope pattern [34][35]. They concluded that clock drift and changes in channel condition are the main

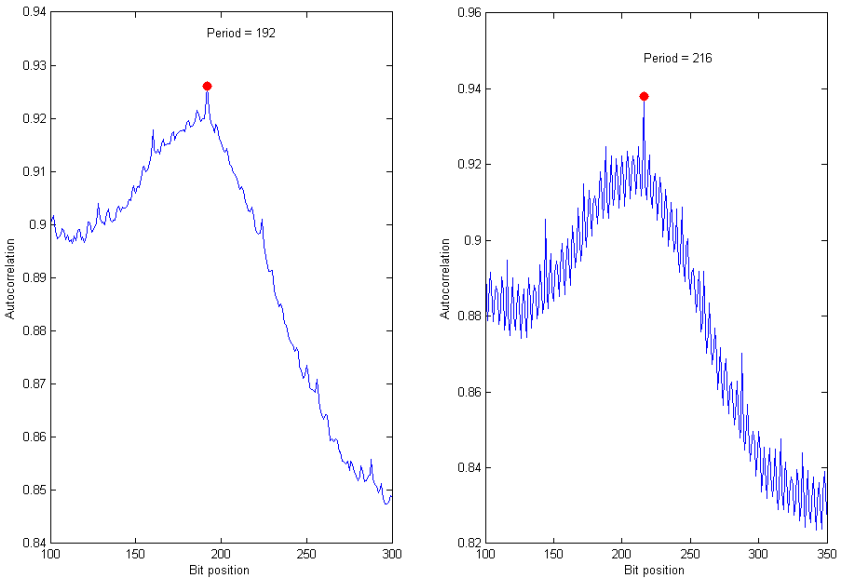

Figure 4.14 The autocorrelation of error rates over bit positions under different transmission rates. The period of the error pattern equals to the number of bits carried by one OFDM symbol. The transmission rates are 48Mbps and 54Mbps for left and right figures, respectively.

reasons of the slope pattern.

We further validate the periodicity of the bit error pattern by computing the autocorrelation of error rates over bit positions. Fig. 4.14 shows the results measured using transmission rates of 48Mbps and 54Mbps. The peaks are identified at bit positions 192 and 216 respectively, which are equal to the number of bits carried by one OFDM symbol at 48Mbps and54 Mbps transmission rates. Table 4.1 shows the modulation, coding rate and number of data bits per symbol for all the transmission rates in 802.11g. The results are consistent with previous measurement study [34] [35], and are accordance with our analysis discussed in Section 4.2.1, confirming that the cause of the periodic bit error pattern is not effected by environmental factors and it's rooted in the transceiver hardware.

Transmission Rate(mbps)	Modulation	Coded bits per carrier	Coded Bits per OFDM Symbol	Coding Rate	Data Bits per Symbol
6	BPSK	1	48	1/2	24
9	BPSK	1	48	3/4	36
12	QPSK	2	96	1/2	48
18	QPSK	2	96	3/4	72
24	16-QAM	4	192	1/2	96
36	16-QAM	4	192	3/4	144
48	64-QAM	6	288	2/3	192
54	64-QAM	6	288	3/4	216

Table 4.1 802.11g OFDM modulations, coding and data rates [31] [61].

4.2.3 Simulation-Based Diagnosis

We conduct simulation-based diagnosis using a pair of 802.11-compliant software transceivers. Both transmitter and receiver are interfaced with a digital LPF, and then connected via an AWGN channel of tunable SNR. To uncover the impacts of LPF roll-off, we measure bit error patterns and packet error rates using extensive simulations and compare the results with that measured using COTS 802.11 devices in testbed experiments. We then tune the order and cut-off frequency of LPFs to diagnose their impacts at bit-level and link layer.

Fig. 4.17 shows the bit error patterns measured in simulations and experiments at 54 Mbps. Similar results are observed under other settings, which are skipped here due to space limit. The experiment result shown in Fig. 4.17a is obtained on a WiFi link consisting of two AR9285 chips. Fig. 4.17b shows simulation result obtained under a SNR of 24 dB, using a 5-order LPF with a cut-off frequency of 5 subcarriers at each edge of the channel. The periodic bit error pattern can be clearly observed and is accordance with experiment result. In comparison, Fig. 4.17c shows simulation results obtained under the same SNR,

but using a 31-order LPF with a cut-off frequency of only 2 subcarriers. As shown in the figure, the periodic pattern is completely eliminated. This is because a LPF of higher order and lower cut-off frequency will result in a sharp transition between passband and stopband, and effectively reduce the power loss of wanted signal caused by roll-off.

Figs. 4.15 and 4.16 show the effects of cutoff frequency and filter order on the packet success rate when the SNR is 20dB. As shown in the figures, both cut-off frequency and filter order have significant impacts on packet loss rate. Low filter order makes the transition between passband and stopband slower, which results in more signal distortions and packet loss. In particular, changing the cutoff frequency from 5 subcarriers down to 2 subcarriers and increasing the filter order from 4 to 10 increases the packet success rate from less than 20% to more than 70%.

The simulation results also imply that LPF roll-off is the root cause of the periodic bit error pattern observed on COTS 802.11 devices. The result demonstrate the significant impact of LPF design on link reliability.

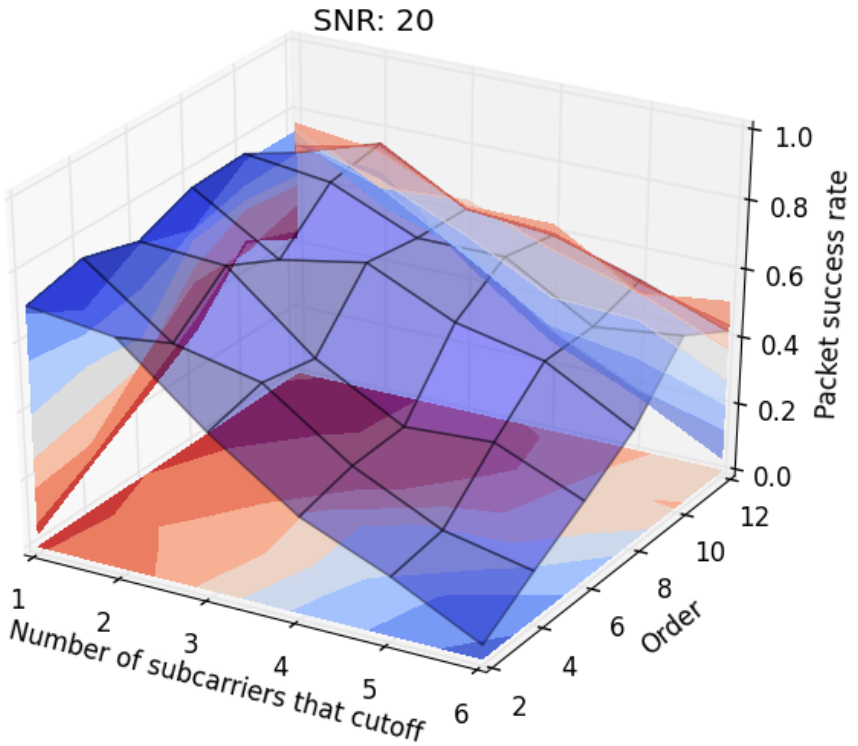

Figure 4.15 Filter simulation results illustrating the effect of cut-off frequency and filter order on packet success rate.

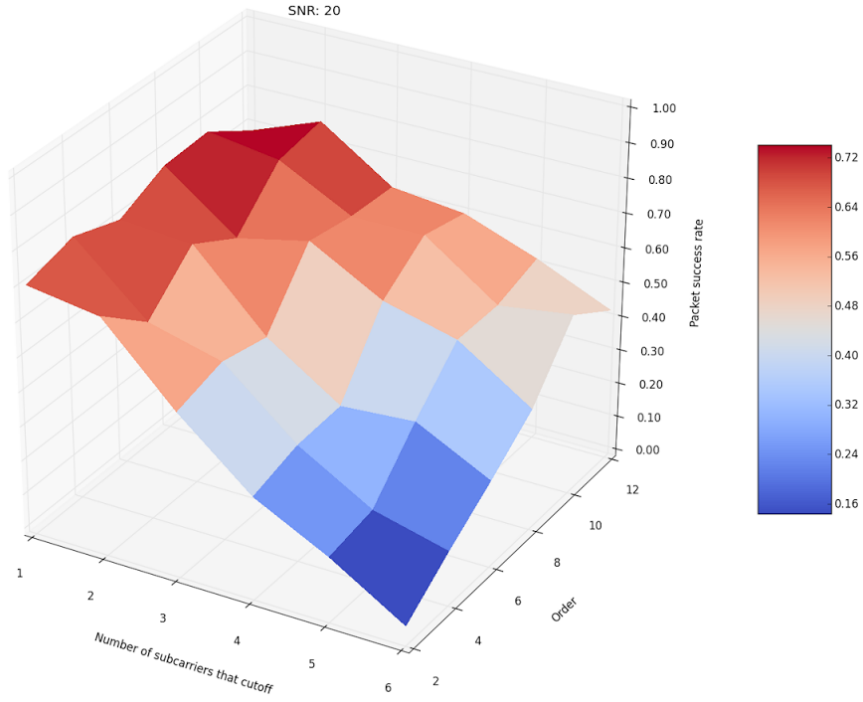

Figure 4.16 Heat map of filter simulation results illustrating the effect of cut-off frequency and filter order on packet success rate.

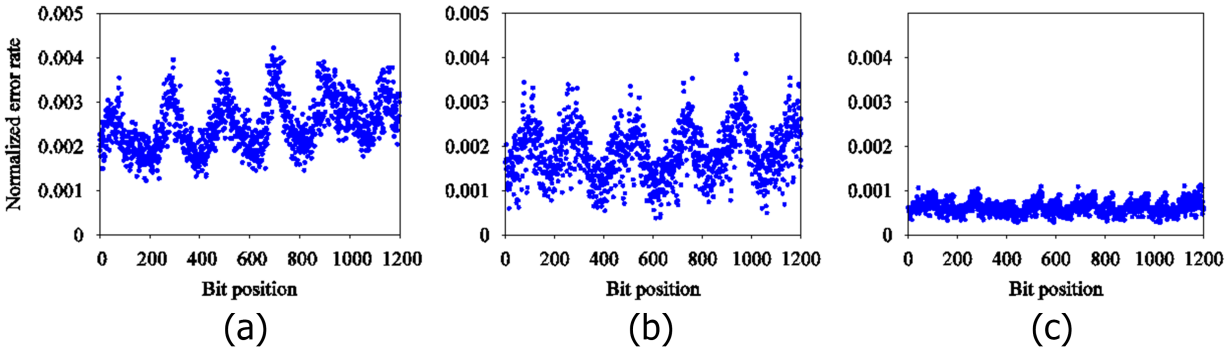

Figure 4.17 Bit error rates measured using a) WiFi link of two AR9285 chips; b) simulation of a 6-tap filter with cut-off frequency of 5 subcarriers; c) simulation of a 32-tap filter with cut-off frequency of 2 subcarriers.

Chapter 5

Application in Video Streaming

The periodic bit error rate pattern has a high potential to get utilized in applications which transmit data with different levels of priority. Video streaming and forward error correction (FEC) are examples of such applications. In this chapter, we evaluate the effect of utilizing the BER pattern on video streaming performance.

The contents of this chapter were adapted from a published conference paper [1].

5.1 The Unequal Error Protection Algorithm

According to the Cisco Visual Networking Index [1], video traffic will make up 80% of all consumer Internet traffic by 2019. Mobile video traffic accounted for 55% of total mobile data traffic in 2015. This traffic will increase by 11-fold by 2020, accounting for 75% of total mobile data traffic. Unfortunately, the fluctuations caused by the inherent unpredictability of the wireless medium being used for streaming also causes fluctuations in both the bit error rates and throughput, resulting in choppy video playback [19].

A large body of work on video coding and streaming has explored various means to improve the throughput, quality and performance of video streaming [39] [55] [19] [44]. Most of them are designed based on channel measurements and estimations which introduce

[1] **"Harnessing hardware defects for improving wireless link performance: Measurements and applications"**. Alireza Ameli Renani, Jun Huang, Guoliang Xing, Abdol-Hossein Esfahanian. INFOCOM 2017 - IEEE Conference on Computer Communications.

processing overhead to the system.

In this section we present our state of the art unequal error protection algorithm which increases the video streaming performance with minimal processing overhead and no data overhead. Based on our study the processing overhead of our UEP algorithm is less than the buffering time in the receiver, so it is not noticeable.

5.1.1 Background

Lightweight User Datagram Protocol or UDP-Lite is a protocol that delivers damaged payloads to upper layers rather than discarding them. UDP-Lite does not drop corrupted packets if the header does not contain errors. One of the major applications of UDP-Lite is video streaming for the same reason. Receiving corrupted packets in video streaming are useful and displaying them to the user is better than discarding them.

The H.264 codec [53] is the most widely used codec for video streaming [13]. We used H.264 implemented in x264 [14]. The x264 has been used in many video services on the web such as Youtube, Facebook, Vimeo and Hulu [14] [15]. H.264 includes three major type of frames: I-frame, P-frame and B-frame. One I-frame and multiple P- and B-frames form a single Group Of Pictures (GOP). The I-frame is the intra-coded picture and is the key frame as it contains static image from a given scene. The P-frame is the predicted picture and only holds changes made from one frame to another (i.e., only moving parts are encoded onto P-frames). The B-frame is the bi-predictive picture and retains the differences between the current frame and the two frames that precede and follow it.

Because the I-frame is required to decode all consecutive P- and B-frames within a given GOP, it is necessarily the most valuable frame, since losing it would mean the loss of the entire GOP. P-frames are second in line of priority, since B-frames need them for decoding.

This means B-frames have the lowest priority, as no other frame types are dependent on them. Based on this observation, we assign different levels of priority to video frame types. I-frames have the highest priority among the frames, then P-frames and finally B-frames.

5.1.2 Basic Idea

Our basic idea to improve the video streaming quality is to transfer the data with highest priority in payload byte positions with lowest error rate. Therefore I-frame data should be positioned into payload bytes that experience the lowest error rates. From the remaining byte positions, P-frames should take the bytes that experience lowest error rates. Finally B-frames should take the remaining positions in the payload. Such reordering moves the errors to less sensitive parts of the data which results in higher quality in the final result. Based on this idea, we plan to develop our UEP algorithm to increase the quality of the video streaming without increasing data overhead.

We explain the display order of the frames in MPEG compression by an example. Assuming GOP = 12 with two B frames between two P frames or between an I frame and a P frame, the display sequence of a GOP will be:

$$I, B_1, B_2, P_1, B_3, B_4, P_2, B_5, B_6, P_3, B_7, B_8 \tag{5.1}$$

The order of the frames change in standard video streaming and is different from their display order because of the dependencies between them. The idea is to stream the frames that are dependent to other frames last as they should get decode last. So the order

of the frames in the standard data stream will be:

$$I, P_1, B_1, B_2, P_2, B_3, B_4, P_3, B_5, B_6, I_2, B_7, B_8 \tag{5.2}$$

In the sequence above, I_2 is the I-frame for the next GOP which comes before B_7 and B_8 because both B-frames are dependent to that I frame.

Our algorithm reorders the above data stream based on priority of the frames. I frames have the highest priority because all P and B frames need them to decode. Next are the P frames as they only need the I frames to decode. Finally B frames are the least important in the GOP as no other frames depend on them. Here is the data stream after reordering the frames:

$$I, P_1, P_2, P_3, B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8 \tag{5.3}$$

The above step can be reversed in the receiver to reorder the data back to the original order so they can get decoded and displayed to the user. The order of the frames becomes 5.2 after reverting the changes made by the transmitter. The rest of the process will be the same as standard streaming.

5.1.3 Algorithm Design

The goal of our unequal error protection algorithm is to increase the quality of the video streaming without adding extra data overhead and minimizing the processing overhead. Given the BER pattern as the input of the algorithm, the output is the mapping of the video bytes into wireless frames.

We make two contributions in the design of UEP algorithm:

- *Reordering the payload based on byte error rate.* UEP algorithm takes all the required wireless packets into account and sorts the payload bytes based on their error rate which is known to both transmitter and receiver ahead of time, and moves the highest priority data bytes into positions that experience the lowest error.

- *Video frame based FEC.* In wireless communication, errors usually happen in bursts so instead of using FEC to wireless packets, our UEP algorithm uses FEC on video frames before reordering. This causes the burst of errors in a wireless packet to get distributed among different video frames which have different FEC data hence results in more error correction capability.

The UEP algorithm starts by sorting the payload bytes by their byte error rates which is known prior to streaming by both transmitter and receiver. Next it reads and decodes all the frames in a GOP, then it sorts them based on their priority. Each GOP will most likely get transmitted using more than one wireless packet, so the size ratio of I frame in GOP should be calculated in order to map the I frame data in that ratio of each packet payload. The same goes for P and B frames. Next, the UEP algorithm maps the highest priority bytes (I-frame bytes first) to the payload bytes with lowest error rates across all the wireless packets.

Fig. 5.1 presents the original video frame positions in wireless packets. In this example, the GOP consists of one I frame, one P frame and two B frames and the GOP requires five wireless packets to transmit all the video frames. Fig. 5.2 presents the how UEP algorithm orders the same GOP using the same number of wireless packets based on their priority to protect them against periodic errors. Fig. 5.3 illustrates the mapping of higher priority

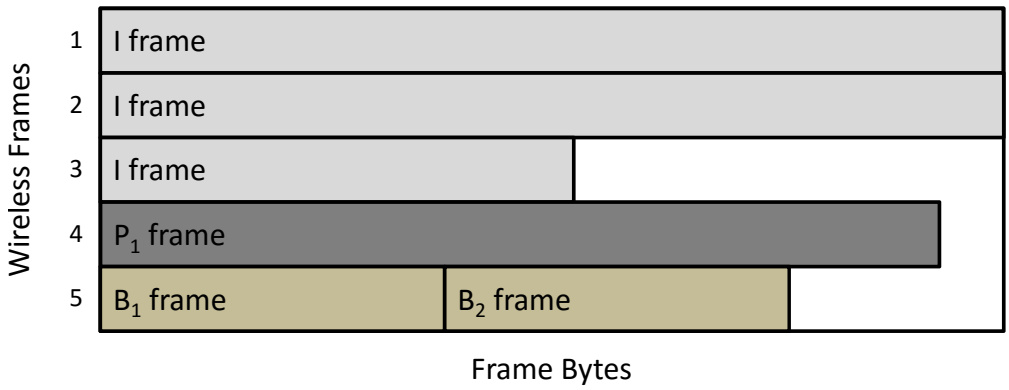

Figure 5.1 Standard video streaming is used to transmit a GOP with one I-frame, one P-frame and two B-frames using five wireless packets.

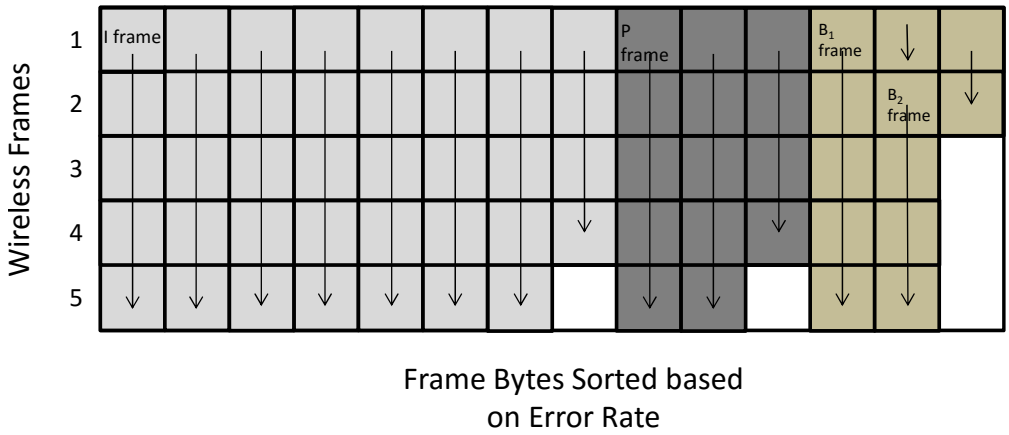

Figure 5.2 UEP algorithm is used to transmit a GOP with one I-frame, one P-frame and two B-frames using five wireless packets.

data to payload positions with lower error rates in just one wireless packet. The receiver goes through the steps in reverse and outputs the original video frame order illustrated in Equ.5.2. The pseudocode of the UEP algorithm is described in Algorithm 1.

The UEP algorithm distributes errors in a single wireless packet to multiple video frames. To maximize the effect of this scheme, we further incorporate a video-frame-based FEC. In wireless communication, errors usually happen in bursts, so using FEC on video frames before reordering causes errors to get distributed among different video frames in the same wireless frame which leads to more error correction. For the purpose of the video streaming application, we decided to use a popular Reed-Solomon code RS(255,223) for both

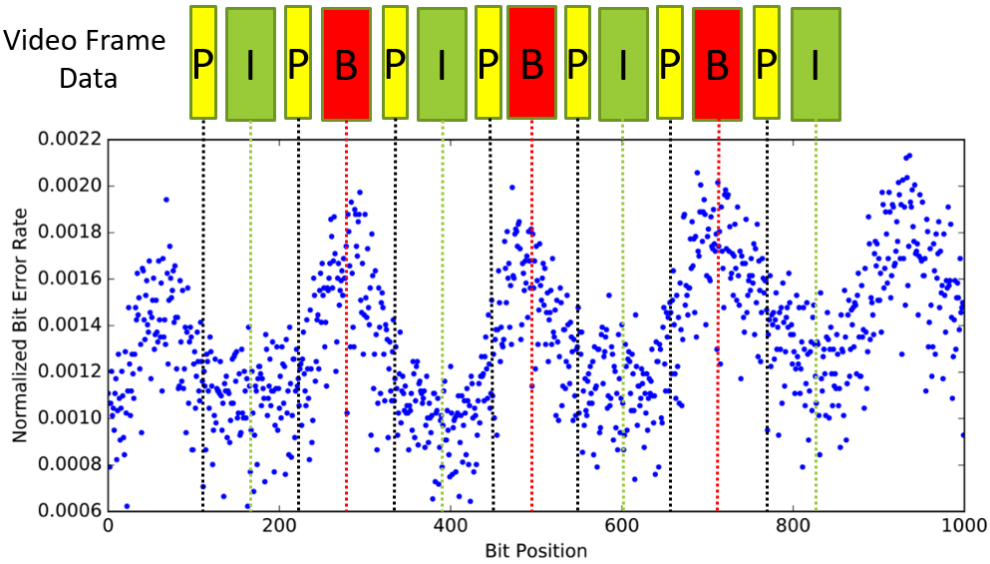

Figure 5.3 UEP algorithm maps higher priority data to bit positions that experience lowest error rates in a wireless packet.

standard and UEP streamings. RS(255,223) adds 32 parity bytes for each 223 bytes of data to a sum of 255 byte block and it can recover up to 16 corrupted bytes in each data block.

5.2 Implementation

In this section we present the details of the implementation of the UEP algorithm. We use trace-driven simulation to evaluate the improvements resulting from our UEP algorithm compared to standard video streaming. The experiment setup consists of two netbooks with Atheros AR2425 and AR9285 chipsets. There is no need to modify the firmware in order to report corrupted frames to the user space, compared to some chipsets like Broadcom which require a modification to an open source firmware like OpenFWWF [32], to pass the corrupted frames to user space. Although some drivers require minimal modification. Simply removing the condition to drop frames containing CRC errors in the ath9k drivers is sufficient to enable all Atheros chipsets with that driver to support our UEP algorithm. We

theorize that other drivers that drop frames containing CRC errors would also require similar modification to support our algorithm, but this would be proven in our future work. It should be noted that there are other drivers currently available that support our UEP algorithm, such as the driver used by in [34] [35]. For these experiments, we installed the modified ath9k wireless driver mentioned in 4.2.2 to have the driver report all packets including corrupted ones that would fail the CRC, to the user space. We must state that the simple modification we made in the driver does not affect the generality of our evaluation results, and our UEP algorithm can deliver the desired improvement as long as the error pattern exists. Although the UDP-lite protocol was specifically designed for use in cases where receiving corrupted frames was preferential to dropping them, it is still not fully supported, and lower layers still discard packets containing CRC errors. However, if firmwares and drivers support UDP-lite in the future, our UEP algorithm will then be viable for use on all chipsets.

We use ffmpeg multimedia framework [5] to encode, decode, transcode and play the video files. The ffmpeg command line tool is used for transcoding and the ffmpeg libraries are used to decode the input files. After executing our UEP algorithm and masking the data with our traces to simulate streaming, we use the ffmpeg libraries to encode the resulting video files into *yuv* formatted videos. We further process the outputs using the VQMT tool [11], which is a program for objective video quality assessment.

5.3 Evaluation

The primary focus of this section is to compare traditional video streaming to streaming using UEP algorithm. A major advantage of our UEP algorithm is that it can be combined with any physical or MAC layer schemes.

5.3.1 Setup and Configuration

We use trace-driven simulations to evaluate both traditional video streaming and UEP streaming. The experiment setup to collect traces consists of an Eee PC with Atheros AR9285 chipset as the receiver and an Acer aspire with Atheros AR2425 chipset as transmitter. The transmitter has a virtual access point (VAP) setup on it, thus enables us to control the transmission rate. As discussed in 4, the transmission rate affects the period of the BER pattern, making it crucial to keep it at a constant rate for our experiments. Using the same trace for standard and UEP streamings makes the comparison fair in all aspects. We leave studying the effect of dynamic transmission rate on the video streaming performance for our future work.

For trace collection, we use the ping command to generate wireless frames with maximum payload size of 1464 bytes. We set the protocol to udp-lite as video streaming and multimedia communication in general is one of the major applications of that protocol. We collect 10,000 packets in each trace. The payload is set to 0xff for all the bytes of the payload. According to [34][35] and our analysis in 4, the payload data does not have any effects on the bit error rate. The major advantage of using 0Xff as payload is that any 0 bit in the bit stream is an error bit, so a simple XNOR of the payload with video bit stream in the next step affects the appropriate bit in the video stream simulating the actual transmission. We apply the same trace to both traditional and UEP video streams and then evaluate the results.

In all experiments the receiver is mobile and traveling with walking speed within the 15 feet range of the transmitter in a 300 square feet room, representing a mobile device in an office environment.

PSNR (dB)	MOS
> 37	5 (Excellent)
31 - 37	4 (Good)
25 - 31	3 (Fair)
20 - 25	2 (Poor)
< 20	1 (Bad)

Table 5.1 PSNR to MOS conversion [41] [51].

We use the well-known *Akiyo* video file for these experiments. The file format is *mp4* and the video is 10 seconds long. The frame rate is 30fps so the video file has 300 frames in total. The GOP size of the file is 30.

5.3.2 Baseline and Metric

We utilize PSNR, the most widely used metric, to quantify video quality. PSNR is purely mathematically defined metric and some literature recommend using metrics that have similar characteristics of Human Visual System (HVS) [42]. For that reason, we also use the Structural SIMilarity (SSIM) index to compare the video quality. The last metric we use is the mean opinion score (MOS) which is the perception of video quality to human eyes. MOS expresses the video quality in five levels: excellent, good, fair, poor and bad. Table 5.1 shows the PSNR to MOS conversion.

5.3.3 Results

Fig. 5.4 shows one video frame recorded using traditional video streaming with PSNR of 12, SSIM of 0.57 and based on Table 5.1 falls into the **Bad** category in MOS. Fig. 5.5 shows the exact same frame recorded using our UEP streaming in the same experiment. The measured PSNR is 30, SSIM is 0.93 and it falls into **Fair** category in MOS. The improvement

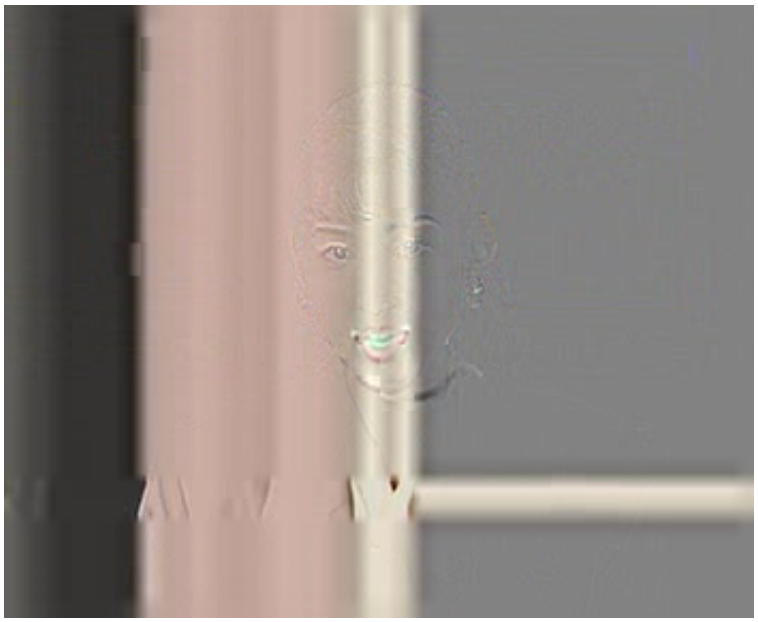

Figure 5.4 A video frame of akiyo.mp4 recorded from standard streaming. Many parts of the frame are completely corrupted. With PSNR of 12 it falls into the lowest MOS category.

of the quality in UEP streaming is obvious to human eye, but all the metrics validate it as well. In standard streaming most of the video frame parts are either completely lost or extremely corrupted, but the UEP algorithm, with minor corruptions, is able to deliver impressively higher quality video over the same unreliable wireless channel.

Fig. 5.6 plots the PSNR for every frame in the *akiyo.mp4* video streamed using both traditional and UEP streamings. We repeat the experiment 10 times using 10 different traces and average the PSNRs for each frame. As shown in the figure, the PSNRs range from 20 dB to 70 dB in traditional streaming while UEP keeps the PSNR constantly at or above 70 dB.

Fig. 5.7 presents the histogram of frame qualities for the same experiment. As shown in the figure, UEP streaming produces fewer frames with bad, poor, fair and good ratings, and 40% more frames with excellent rating compared to traditional streaming. The video streaming experiment results indicate that our UEP algorithm surpasses the standard

Figure 5.5 A video frame of akiyo.mp4 recorded from UEP streaming which shows major improvements in quality of the video. With PSNR of 30, it is in the high range of fair category in MOS.

streaming by a clear margin without adding extra data overhead.

The average SSIM index of all the video frames in the abovementioned experiment for standard and UEP video streamings are 0.93 and 0.96, respectively. The UEP algorithm shows massive improvements in video streaming according to all three metrics we use in this work.

We discussed the aggregated results which show UEP algorithm is successfully moving the bit errors to less sensitive parts of the video. We compared the results with standard streaming by using the same trace for both methods and UEP algorithm achieved extremely higher quality videos. Next we discuss the individual experiment results and also the effect of video frame based FEC and its impact on correcting the burst of errors distributed by the UEP algorithm to different video frames.

Table 5.2 shows the average PSNR and SSIM in all the individual experiments and also the overall average of the 10 experiments conducted in this section.

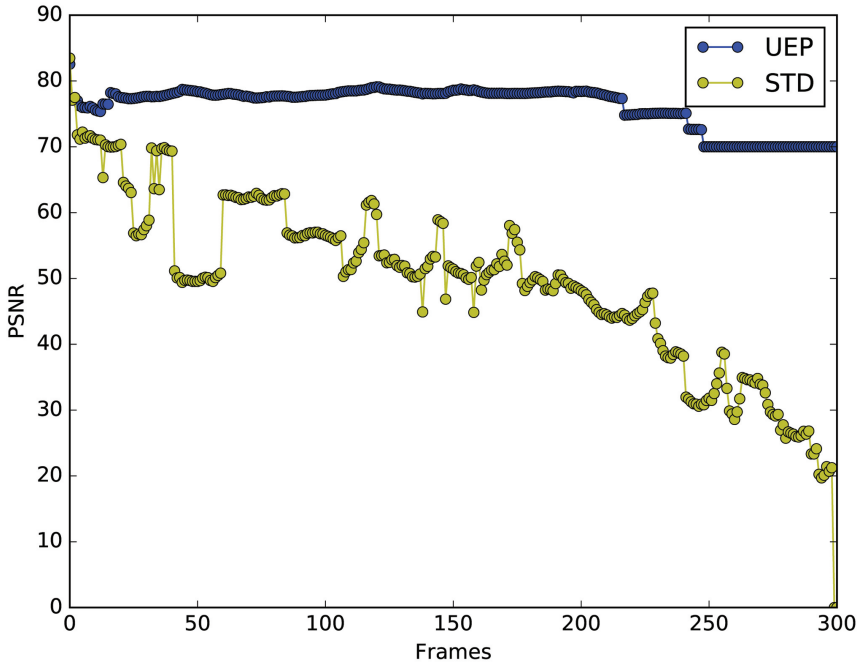

Figure 5.6 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings.

Figures 5.8 to 5.27 plot the PSNR and MOS quality levels for every frame in the *akiyo.mp4* video streamed using both traditional and UEP streamings in the rest of the experiments.

Fig. 5.8 plots the PSNR for every frame in the *akiyo.mp4* video streamed using both traditional and UEP streamings in experiment number 1. The figure shows that UEP algorithm is able to deliver all 300 video frames with PSNR of 100 and in contrast traditional streaming has only 154 video frames with PSNR of 100. As UEP algorithm only relocates the errors, we can conclude that it is the video frame based FEC that is able to correct the errors that appeared in almost 50% of the video frames and it corrected all video frames to achieve PSNR of 100. Fig. 5.9 presents the histogram of frame qualities for the same experiment. The figure indicates that the UEP is successfully distributing the burst of errors even in *poor* video frames and video frame based FEC is able to correct all those errors and deliver 100

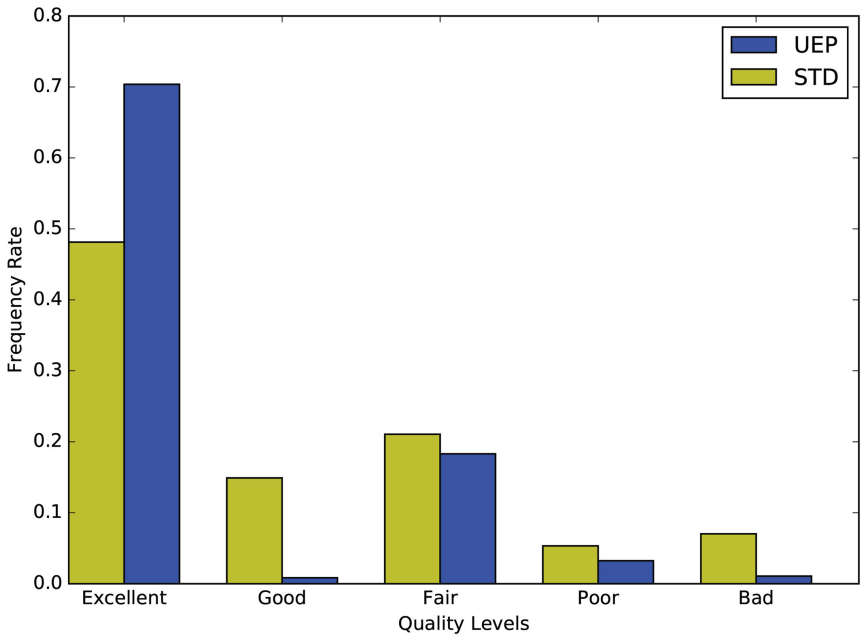

Figure 5.7 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings.

PSNR for all the video frames. Figures 5.10 and 5.11 present the result of the experiment number 2 which has a highly corrupted trace. Even in the extreme case scenario which most of the video frames streamed using standard streaming fall into the *bad* MOS, the UEP is able to deliver most of the video frames with *fair* MOS, which is two level improvement compared to standard streaming.

Figures 5.12 to 5.15 present the result of the experiment numbers 3 and 4. These two experiments are the only experiments that standard streaming is delivering slightly higher quality video frames. The average PSNR of all the frames in experiment 3 is 24 and 21 for standard and UEP streamings. The values are 23 and 21 for standard and UEP streamings in experiment 4. This is expected as the goal of the UEP algorithm is to increase the average performance and does not guarantee higher quality in every situation. In chapter 4 we prove that the periodic bit error pattern exists in every wireless communication, no matter what

	PSNR		SSIM	
Experiment	UEP	STD	UEP	STD
1	100	64.91	1	0.97
2	27.94	20.22	0.89	0.76
3	26.02	25.6	0.87	0.86
4	25.8	26.72	0.87	0.86
5	100	81.02	1	1
6	100	61.88	1	0.99
7	100	48.4	1	0.99
8	100	79.15	1	0.99
9	100	65.41	1	0.99
10	100	32.79	1	0.95
Average	77.976	50.61	0.963	0.936

Table 5.2 Average PSNR and SSIM of all the experiments.

environmental variables are. We expect anomalies and sometimes they dominate the periodic bit error pattern, and that explains the weak performance of UEP algorithm in these two experiments.

Figures 5.16 to 5.27 present the experiments 5-10. The UEP streaming is delivering all the video frames with PSNR of 100 while the average PSNR of standard streaming reaches below 32 in experiment 10. This shows a significant improvement of video quality achieved by combining the state of the art UEP algorithm to distribute the bit errors to different video frames, with the novel idea of applying FEC on video streaming which corrected all the bit errors in all the video frames.

We list the FEC code we use in streaming in Chapter 5. It is a popular Reed-Solomon code RS(255,223) for both standard and UEP streamings. RS(255,223) adds 32 parity bytes for each 223 bytes of data to a sum of 255 byte block and it can recover up to 16 corrupted bytes in each data block. We can use more parity bytes to increase the number of bytes the system can correct. In the future work, we can study different coding schemes and the effect of changing that threshold on system performance, and possibly use a dynamic FEC

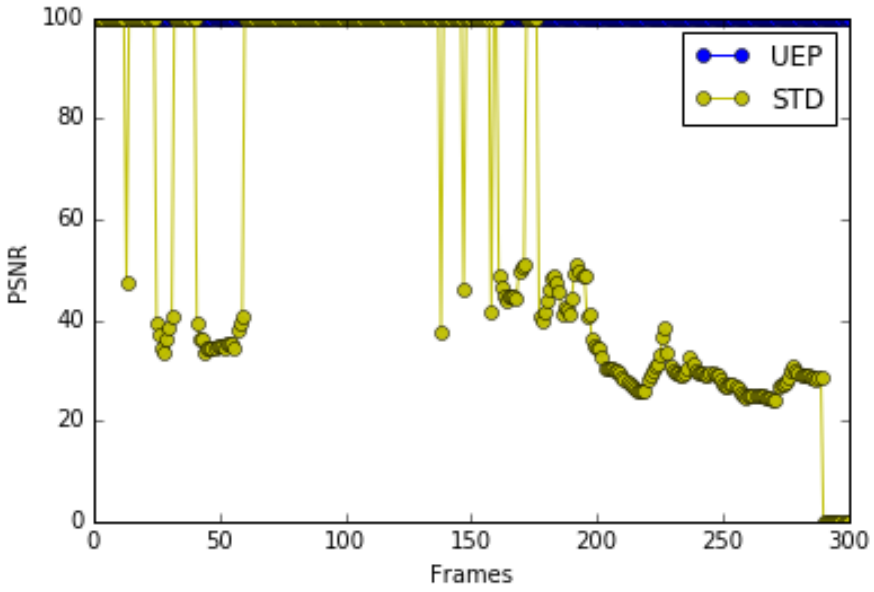

Figure 5.8 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 1.

to minimize the redundancy and maximize the error correction depending on the number of errors.

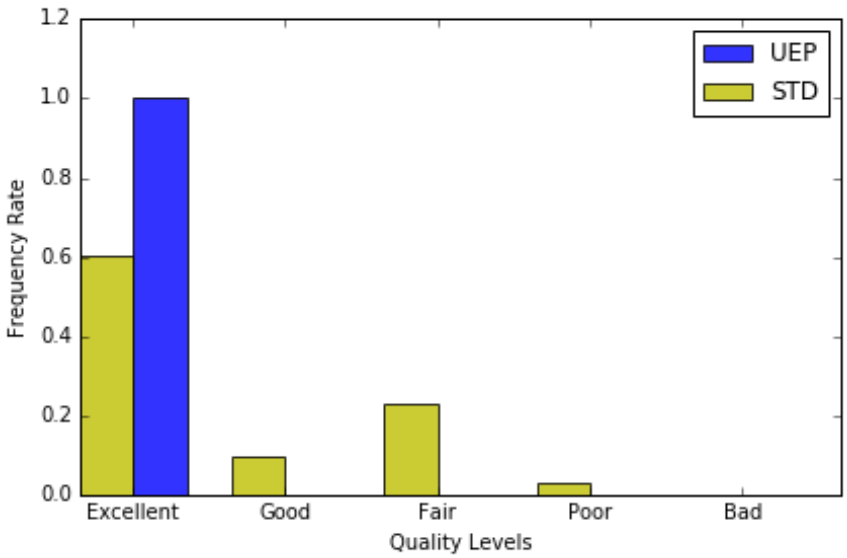

Figure 5.9 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 1.

Algorithm 1 Pseudocode of the UEP algorithm.

```
max_pkt_size = 1500
error_rates = sort(read_error_rates())// read and sort BERs
gop = read_video_frames()// read video frames
p, b = 0, 0
for i = 0 → len(gop) do
    if gop[i].type() == "I" then
        I_frame = gop[i]
    else if gop[i].type() == "P" then
        P_frames[p++] = gop[i]
    else if gop[i].type() == "B" then
        B_frames[b++] = gop[i]
wireless_pkts_count = len(gop)/max_pkt_size

                                                    ▷ reorder gop based on frame priority
i = 0
reordered_gop[i++] = I_frame
for j = 0 → len(P_frames) do
    reordered_gop[i++] = P_frames[j]
for j = 0 → len(B_frames) do
    reordered_gop[i++] = B_frames[j]

                                                    ▷ get bytestream of the data
data = get_byte_stream(reordered_gop)
k = 0
for i = 0 → max_pkt_size do
    for j = 0 → wireless_pkts_count do

▷ put data with highest priority to byte position with lowest error rate across all the wireless packets
        uep_data[error_rates[i]][j] = data[k++]
for i = 0 → max_pkt_size do
    transmit(uep_data[:][i])                                    ▷ transmit packet i
```

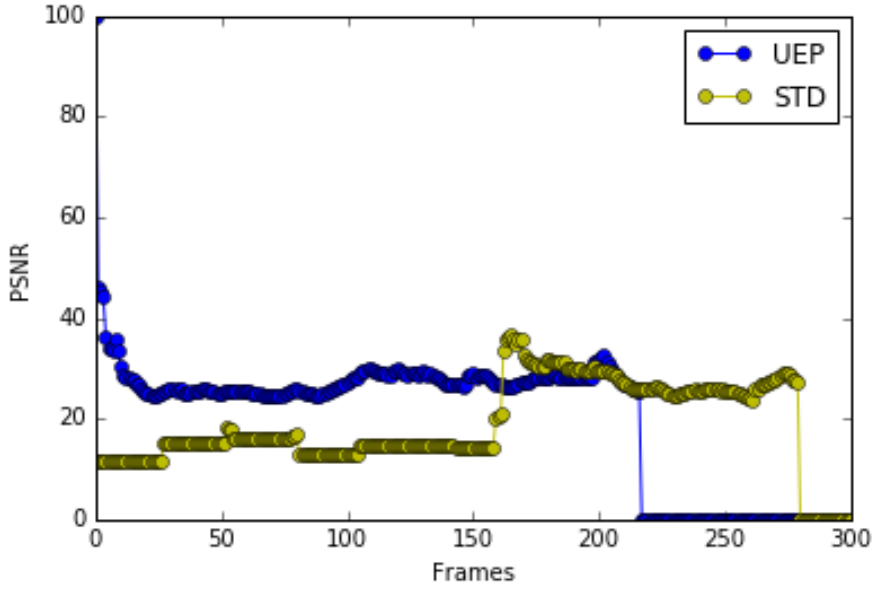

Figure 5.10 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 2.

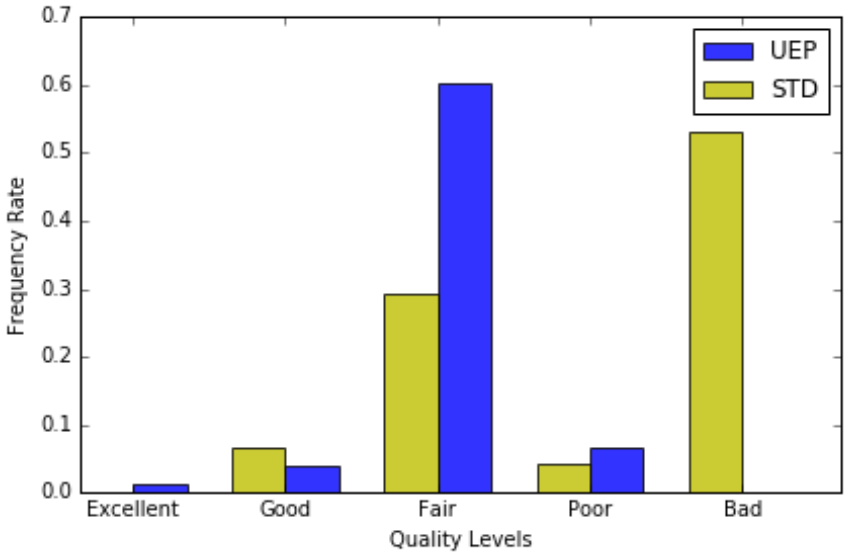

Figure 5.11 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 2.

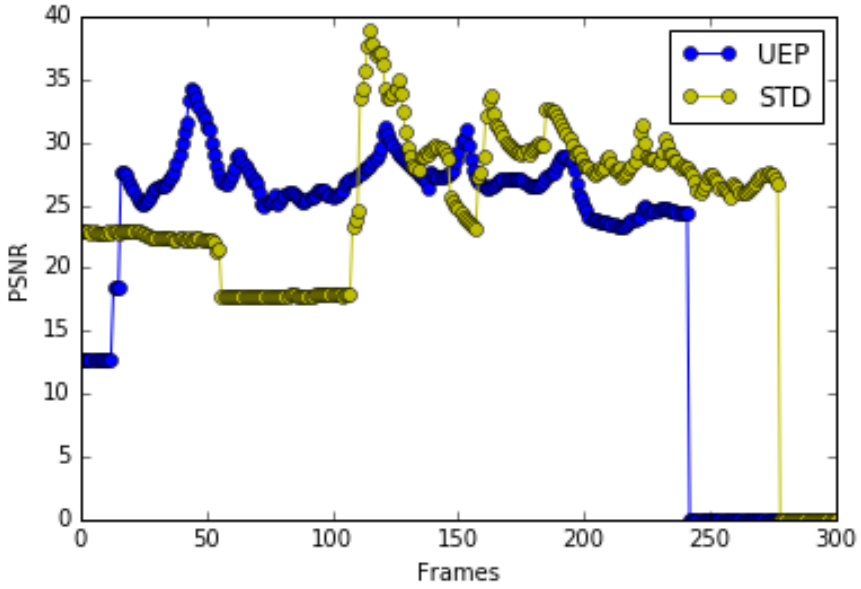

Figure 5.12 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 3.

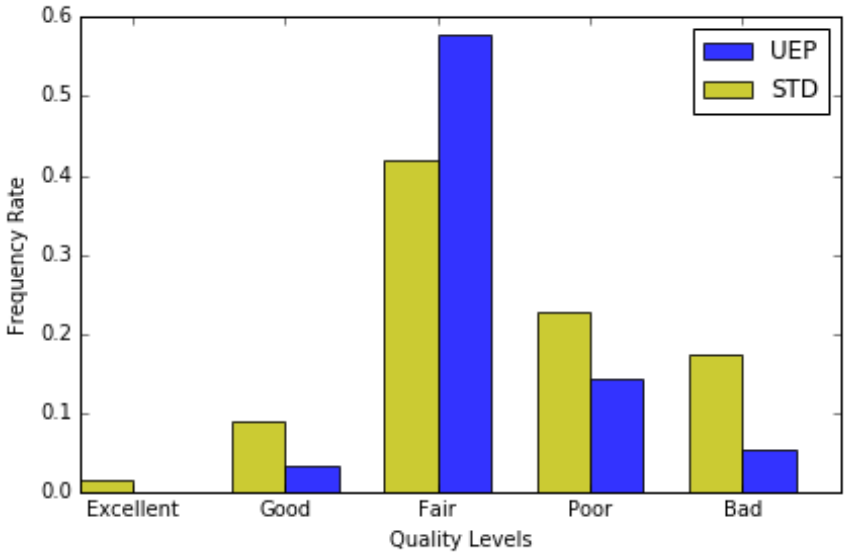

Figure 5.13 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 3.

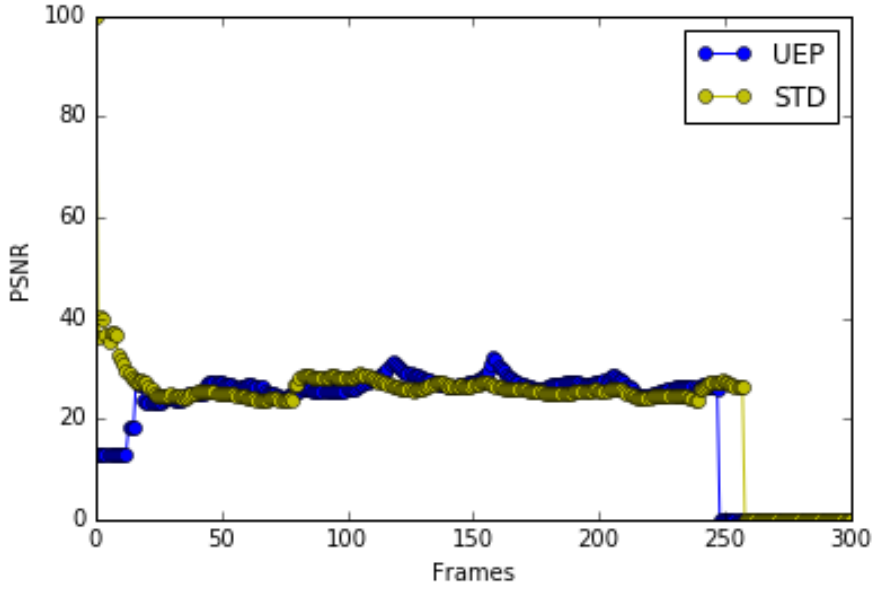

Figure 5.14 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 4.

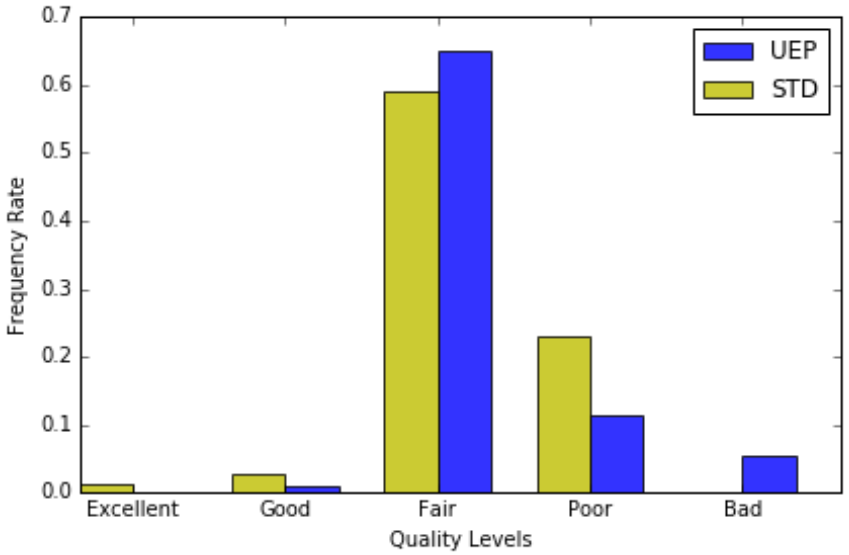

Figure 5.15 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 4.

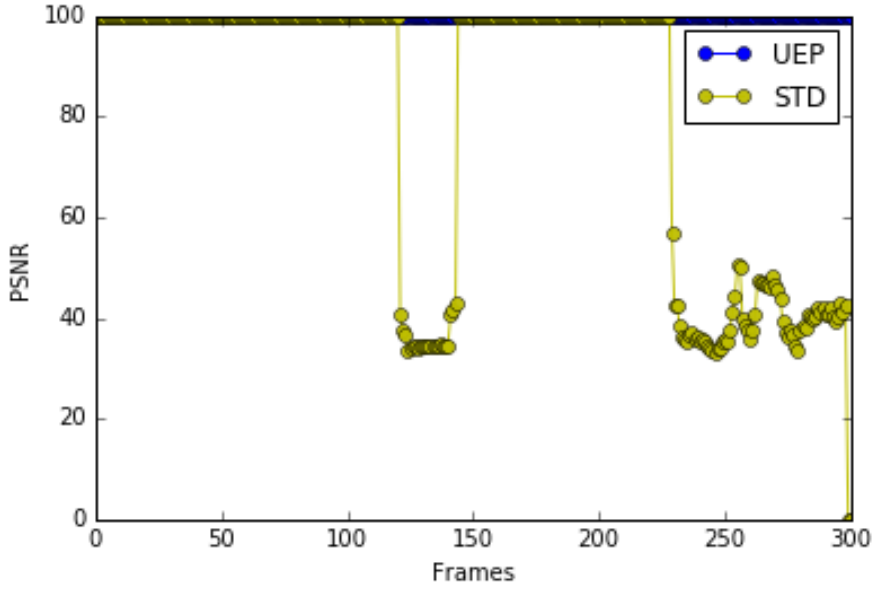

Figure 5.16 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 5.

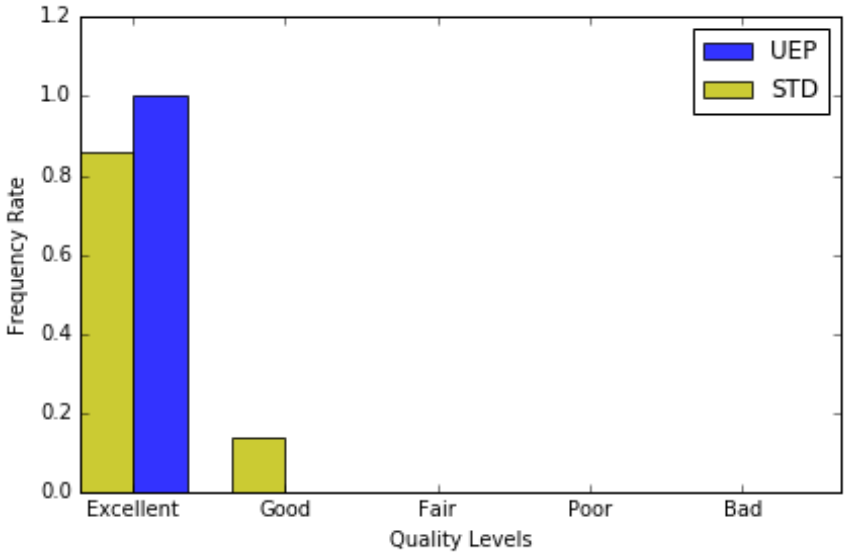

Figure 5.17 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 5.

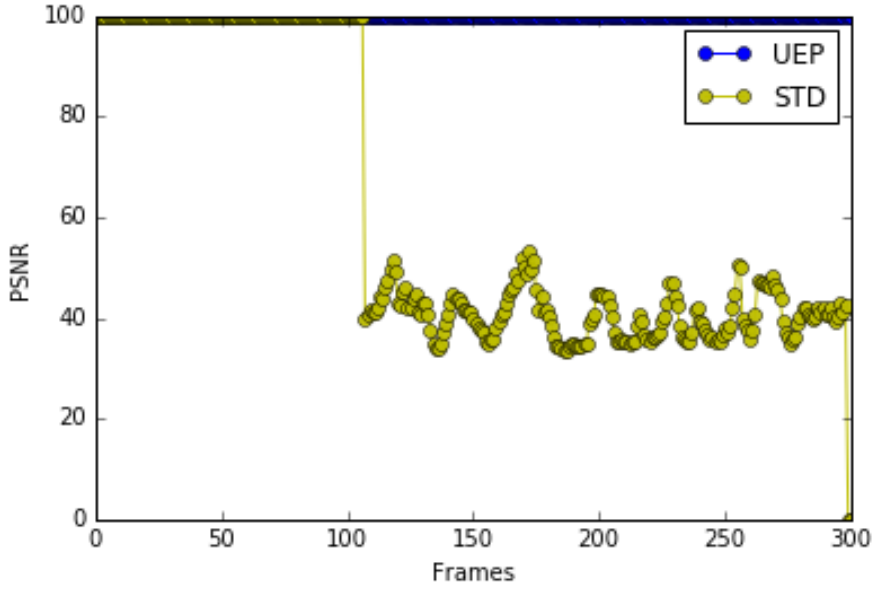

Figure 5.18 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 6.

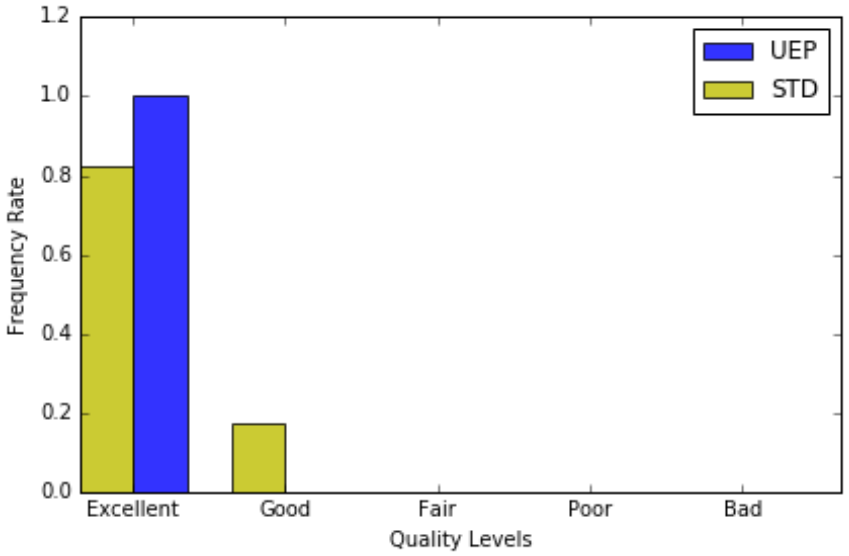

Figure 5.19 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 6.

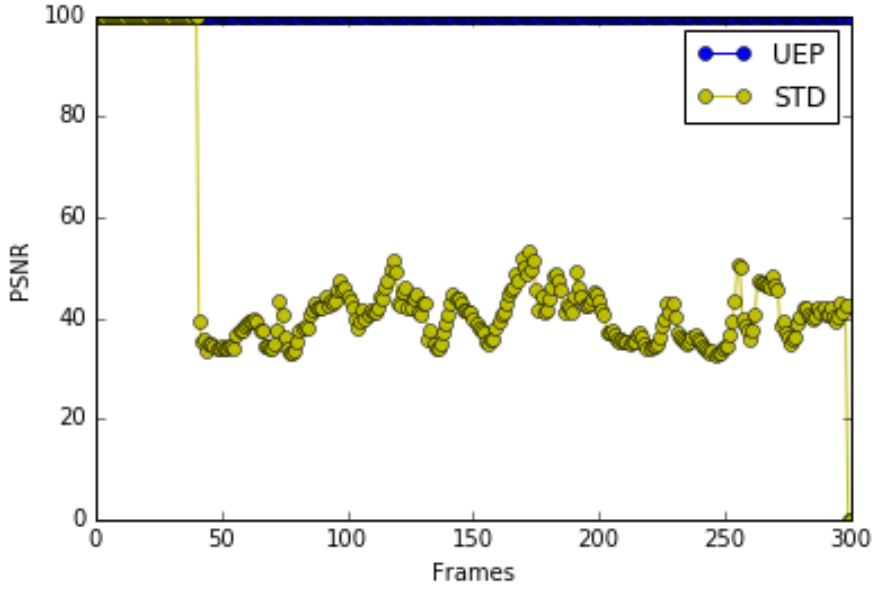

Figure 5.20 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 7.

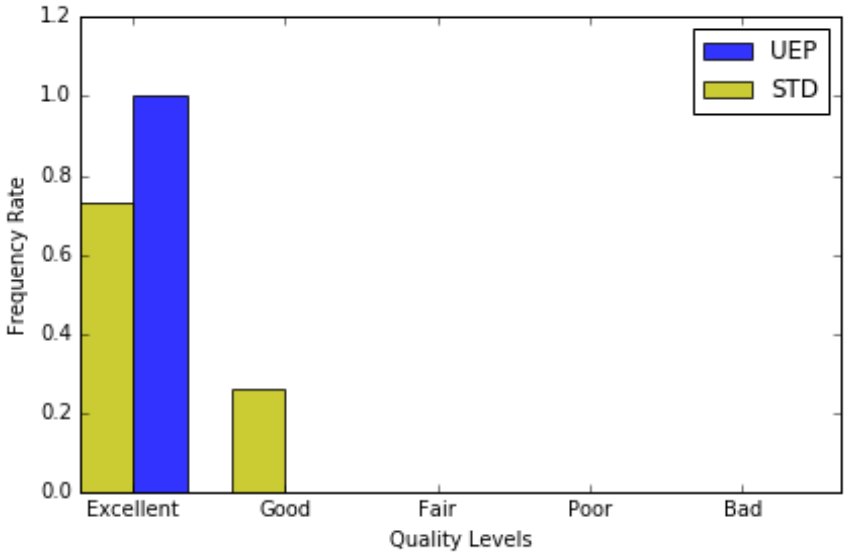

Figure 5.21 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 7.

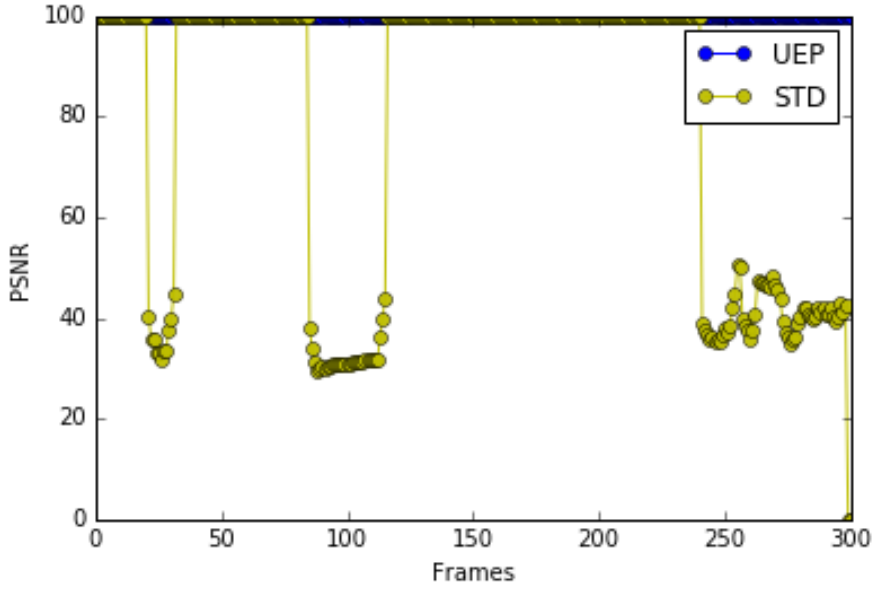

Figure 5.22 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 8.

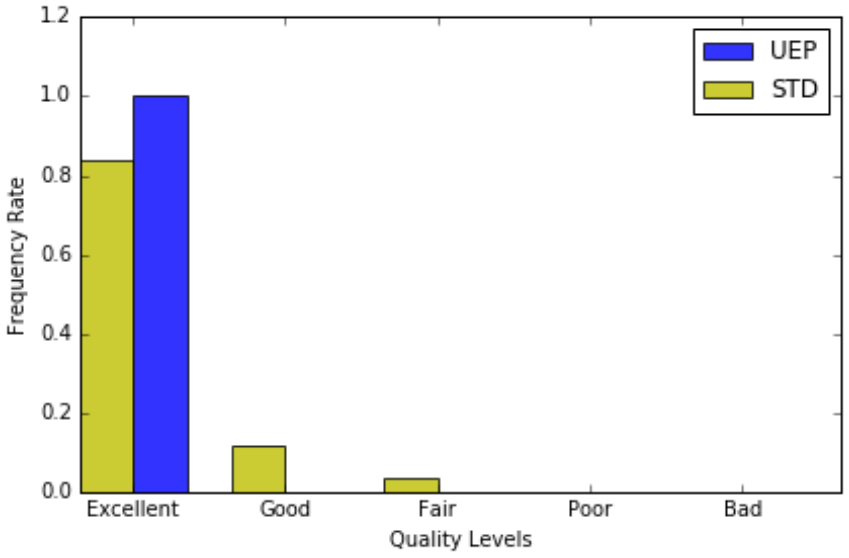

Figure 5.23 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 8.

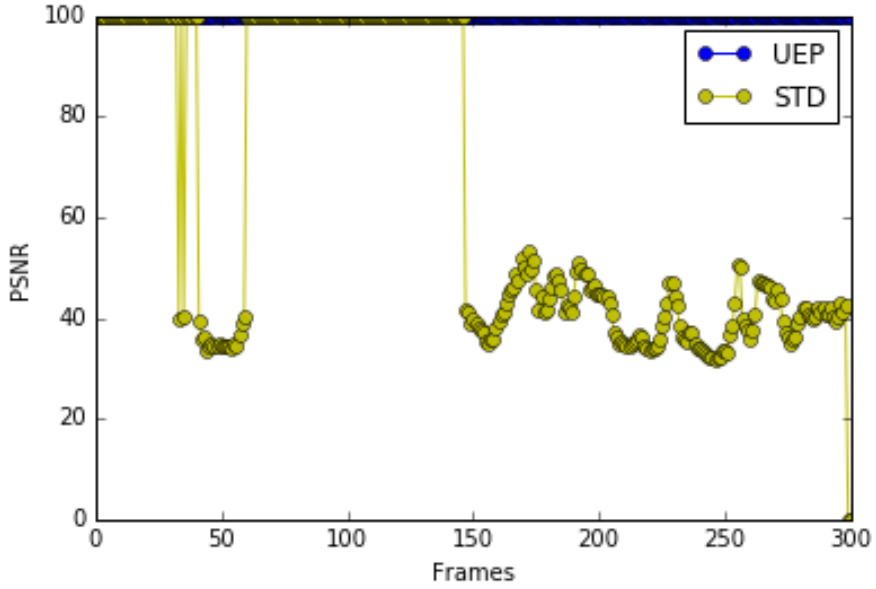

Figure 5.24 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 9.

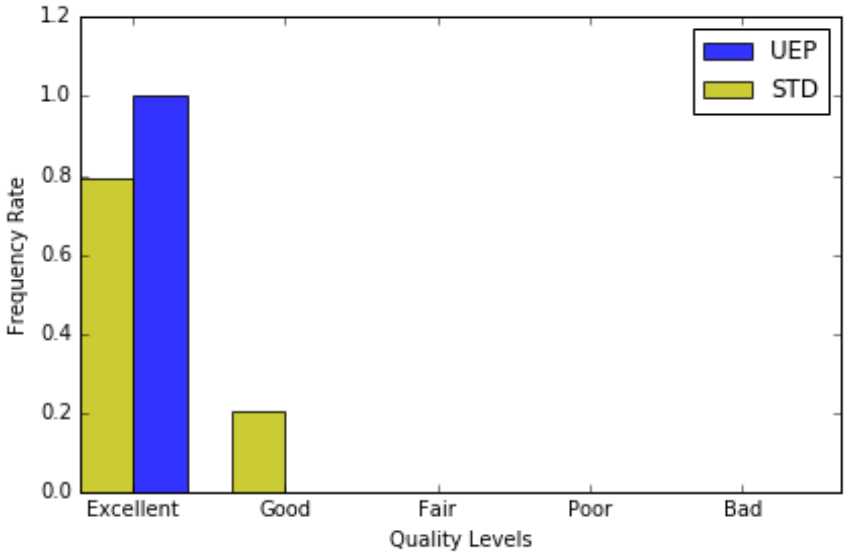

Figure 5.25 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 9.

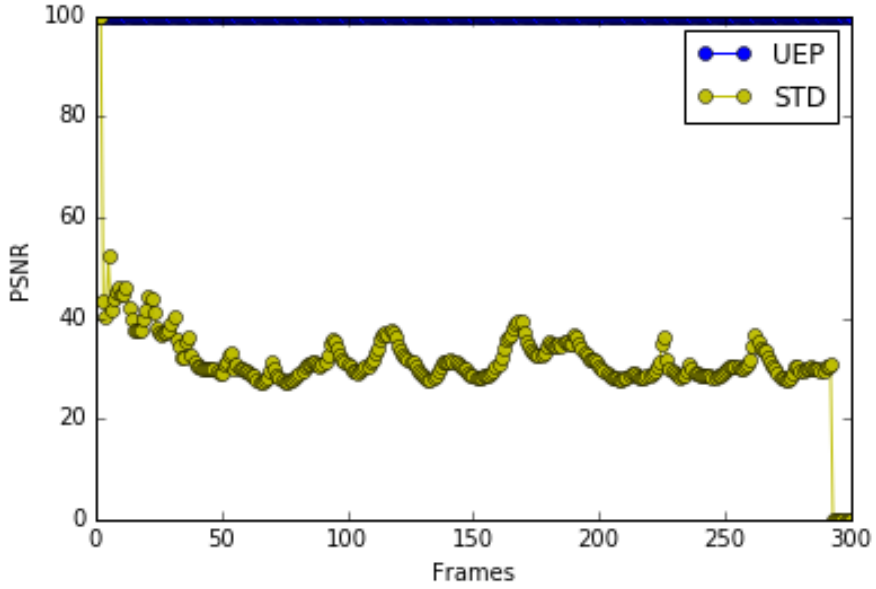

Figure 5.26 PSNR of all 300 frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 10.

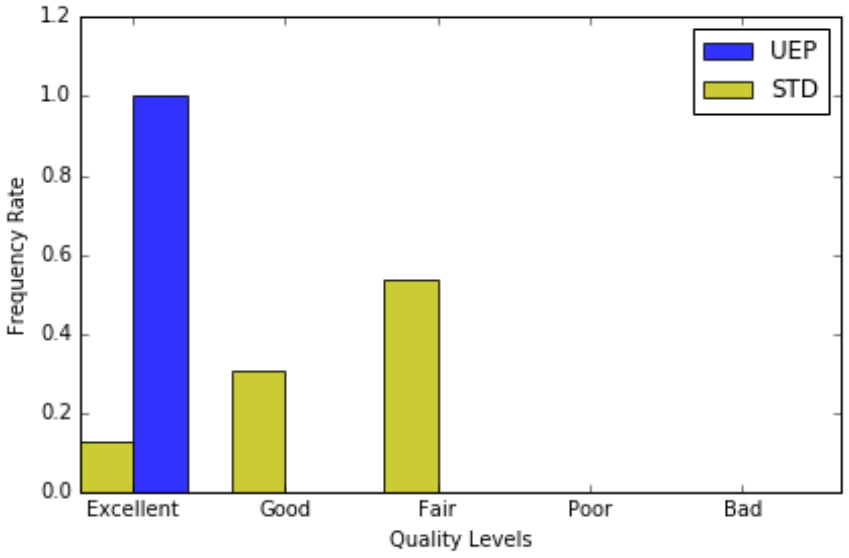

Figure 5.27 MOS of all the frames in akiyo.mp4 streamed using UEP and standard streamings in experiment number 10.

Effect of retransmission. Thus far, we showed the impressive effect of UEP algorithm in increasing the video streaming performance without considering retransmission. In this section, we study the effect of packet retransmission on the video quality in both UEP and traditional streamings. Retransmission makes it possible for both streamings to retransmit the wireless packet in case of failure. As the UEP algorithm has higher chance of correcting the errors, this gives the standard streaming an advantage to close the quality gap and deliver video frames with higher qualities, possibly very close to what UEP algorithm can achieve. To study the effect of retransmission in video quality, we conduct experiments with 1, 2 or 3 retransmission possibilities.

Table 5.3 presents the average byte errors in I, P and B frames in all 10 experiments without considering retransmission. It also presents the total errors observed in both standard and UEP streamings. The UEP streaming is experiencing approximately 50% fewer errors than standard streaming which is significantly lower considering all experiments are conducted over identical traces.

Table 5.4 presents the average byte errors in I, P and B frames in all 10 experiments by having 3 possible retransmissions for every wireless packet. Each algorithm retransmits the wireless packet if the packet get delivered corrupted. The table shows much lower errors in both standard and UEP streamings, and a closer gap in the total errors. UEP algorithm is experiencing 20% fewer errors than standard streaming which is still significant considering UEP algorithm shifts errors from sensitive parts of the video to less sensitive parts. As an example, the total errors in the I frames are 1700 and 502 in standard and UEP streamings, respectively. This means UEP is experiencing 70% fewer errors in the most sensitive parts of the video which shows significant improvement in video streaming quality.

Frame type	STD	UEP
I frames	14255.9	5979.5
P frames	18102.8	9346.1
B frames	17260.8	9343.4
Total errors	49619.5	24669

Table 5.3 Average byte errors in I, P and B frames in standard and UEP streamings without retransmission.

Frame type	STD	UEP
I frames	1700.8	502.5
P frames	6269.3	3214.3
B frames	4945.7	6505.9
Total errors	12915.8	10222.7

Table 5.4 Average byte errors in I, P and B frames in standard and UEP streamings with 3 retransmissions.

Chapter 6

Application in Vehicular Communication

Recent years have witnessed the emergence of vehicular networks deployed for smarter and safer transportation services, such as vehicle safety communication systems, context-aware drive tuning, remote diagnosis and network-assisted autonomous driving. These applications typically impose stringent performance requirements on vehicular networks, including low delivery delay, high packet reception ratio and spectrum efficiency. To support performance-demanding applications in vehicular networks, previous research mainly focuses on designing communication protocols and architectures that address the fast variation of the wireless channel [21] [30] [48]. However, the effects of hardware defects have been largely unexplored, which leads to communication behaviors and performance issues at upper-layers that are difficult to diagnose and control [38] [34].

6.1 Background

With recent emerging growth in autonomous driving, Wireless Access in Vehicular Environments (WAVE) is receiving a great interest. There are numerous benefits for WAVE, but we focus on safety aspect of it in Vehicular Ad hoc Networks (VANETs). WAVE is facing many challenges mostly due to extremely fast varying channels hence it is crucial that packets

get delivered successfully specially in emergency scenarios.

There has been a great body of work in simulation and SDR based study of WAVE which we discussed in Chapter 2. Recent developments of autonomous vehicles are bringing more attention to Dedicated Short Range Communications (DSRC) in VANETs, opening a huge market for wireless chipset vendors to make 802.11p chipsets. Recent studies have focused on using 802.11a chipsets instead of 802.11p chipsets for two main reasons:

1. IEEE 802.11p standard [17] is based on IEEE 802.11a standard [16] with differences in bandwidth, carrier spacing and symbol length. In 802.11p the channel *bandwidth* of 10MHz is usually used, with the optional 20MHz mode. The *carrier spacing* is reduced by half compared to 802.11a. The *symbol length* is doubled.

2. There are no 802.11p chipsets in production at the time of writing this dissertation. There are a few products available, but they are not mass produced and the vendors do not sell them directly to costumers.

Almost every study that used 802.11a chipsets modified them to make them similar to 802.11p standards. However, these modified chipsets suffer from at least omitting one of the specifications because of various hardware and software limitations, such as using 20MHz channel instead of 10MHz channel, or using 2.4GHz band instead of 5.9GHz band of 802.11p which ranges from 5.850 to 5.925 GHz.

There are a few prototypes for On Board Units (OBUs) and Road Side Units (RSUs) from vendors such as Cohda Wireless [3], Unex [2] and DENSO [4] which make it possible to conduct field tests and execute measurement experiments.

6.2 Harnessing Hardware Defects in Vehicular Communication

IEEE 802.11p standard is based on IEEE 802.11a, so our measurements discussed in Chapter 4 is applied in 802.11p. Building on our measurements and models we conclude that our UEP algorithm presented in Chapter 5 can be modified to harness hardware design tradeoffs and improve DSRC throughput by introducing minimal processing overhead, potentially saving lives in emergency scenarios. FEC can also be employed to reduce the errors caused by fast varying channels in vehicular environments.

We design an architecture to exploit the systemic impacts of transceiver hardware defects in vehicular networks. By modeling the systematic impacts of hardware defects, the system can be employed for directing the tuning of various analog and digital components, given the system performance requirements imposed by network applications. In addition, it harnesses hardware defects to improve communication performance, leveraging the fact that hardware induced error patterns are invariant in despite of the fast channel variation in vehicular networks.

We describe the architecture of the system next, and then explain the three main components of the system, *hardware defect modeling*, *harnessing hardware defect at PHY* and *exploiting BER pattern at link layer*. Finally based on our proposed methodology and architecture, we explain how one may implement the experiments and evaluate the results.

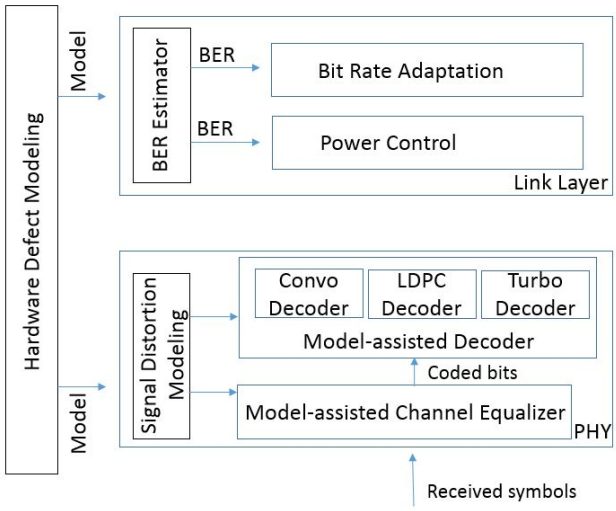

Figure 6.1 The architecture of the system.

6.3 Architecture

As shown in Fig. 6.1, our system consists of the following major components. The *hardware defect modeling* models the error patterns caused by hardware defect and exposes them to PHY and link layer components. At PHY, the system leverages the pattern to improve channel equalization and decoding performance. At the link layer, our system leverages error patterns for bit error estimation and uses the results to optimize various link layer protocols, such as data rate adaptation and transmission power control.

6.3.1 Hardware Defect Modeling

Hardware defect modeling component models the error pattern based on communication theory rather than mathematical modeling. Recent research shows that accurate per-packet BER estimation has the potential of greatly improving the performance of wireless networks [28]. As an example, in autonomous driving systems, sensory data streaming techniques can tolerate a certain level of errors, where a corrupted packet can be recovered

using application layer code if the BER of the packet is lower than a threshold. Exposing the error patterns to PHY and link layer protocols to enable fine-grained performance optimization. Conducting a systematic measurement study is required to gain insights into the characteristics of vehicular network traffic in different applications and scenarios in 802.11p DSRC based V2V communications. Based on the results, efficient algorithms can be developed to model the error pattern.

6.3.2 Harnessing Hardware Defects at PHY

If PHY knows the signal distortions prior to receiving the signal, it can optimize the channel equalization and decoding algorithms to improve their performance. We tackle two major challenges at the PHY of vehicular network stack. First, wireless communication in vehicular networks suffers from noise and interference significantly more than conventional wireless networks. Second, communication in vehicular networks is subject to the fast fading of wireless channel. Conventional receiver obtains channel estimation using packet preamble at the beginning of packet receiving. However, such estimation may be quickly outdated in vehicular environment due to the fast movements of vehicles, causing substantial bit errors. The proposed new system addresses the above challenges using novel system designs that integrate BER model from *hardware defect modeling* and *signal distortion modeling*.

6.3.3 Exploiting BER Pattern at Link Layer

The link layer can exploit the BER pattern models to significantly improve the performance of various link layer protocols such as bit rate adaptation, power control, selective retransmission and hybrid automatic repeat request (ARQ). It can improve quality of service

while reducing spectrum overhead. Since this component operates at the link layer, it can be directly implemented on off-the-shelf devices

6.4 Suggested Evaluation Methodology

Based on our proposed architecture, one may evaluate the system with a combined effort of theoretical analysis, simulation study and real testbed experiments. Publicly available datasets [27] [18] can be used to evaluate the performance of *hardware defect modeling*. Real channel traces can be fed into simulation tools for realistic and extensive simulation study of the proposed algorithms and protocols. The performance of the system in real wireless network testbeds can be also evaluated. The BER estimator and link layer protocols can be implemented and evaluated on commercial off-the-shelf devices. A testbed of laptops and smartphones may be required to evaluate the impact of the system on link layer throughput and delay under mobility. One may conduct the following real testbed experiments to study the systemic effect of hardware defects in vehicular communication.

To obtain precise measurements of the frequency roll-off caused by the OBU, experiments should be conducted in a controlled setup where transmitter and receiver are connected using a cable to exclude the interference and distortion induced by wireless channel. We presented our controlled experiments we conducted for 802.11g chipsets in Section 4.1.2.

The setup should consist of an 802.11p prototype and a USRP2. Fig. 6.2 shows en example of such setup which consists of a Cohda MK5 OBU and a USRP2 and an XCVR2450 daughterboard. The Cohda MK5 is connected to the USRP2 through a Mini-Circuits VAT-20+ SMA fixed attenuator to reduce the transmission power by 20 dB, minimizing the risk of any damage to the USRP2 or the daughterboard. We have checked the datasheet of such

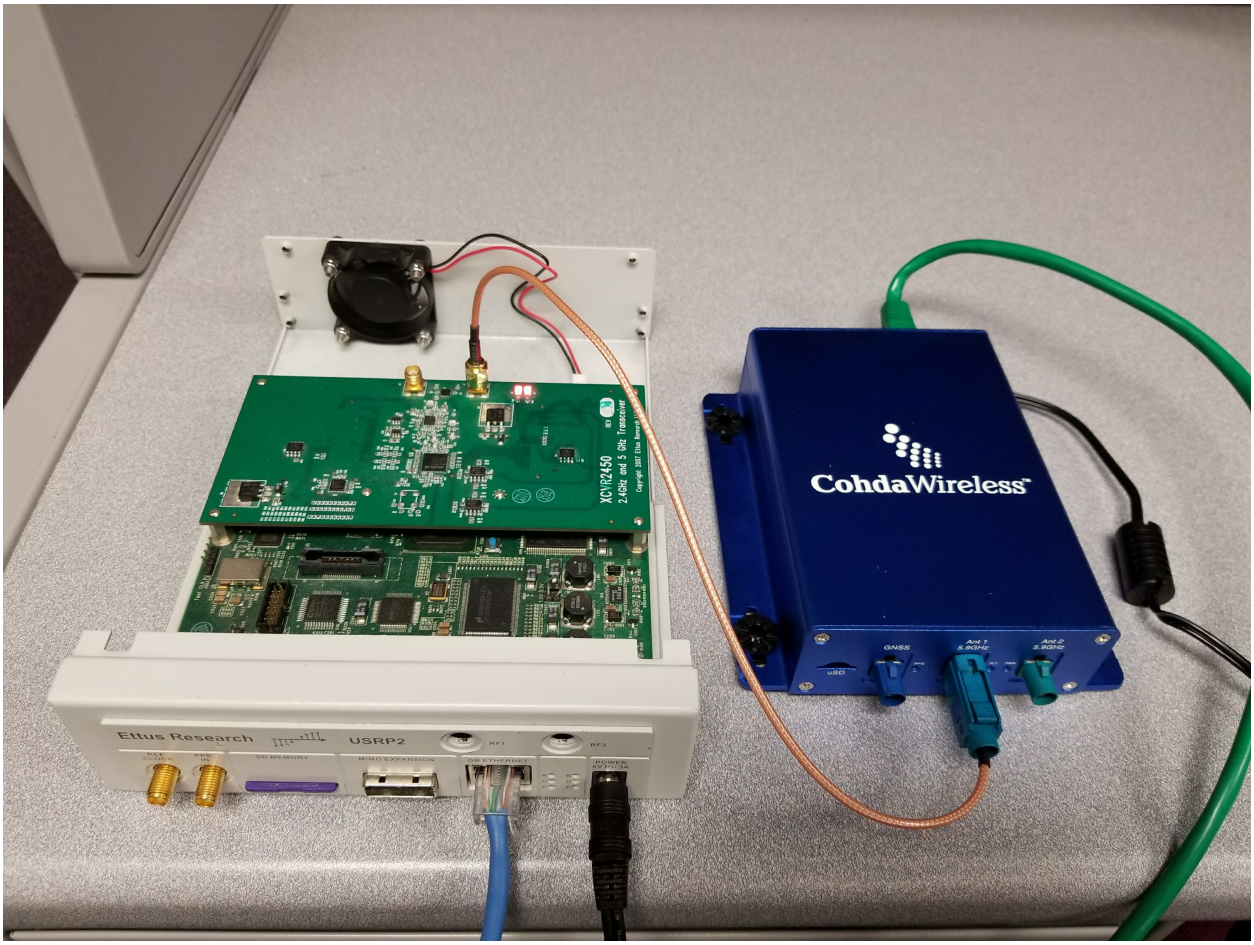

Figure 6.2 Setup of over-cable measurements using an USRP2 and Cohda MK5 OBU. The MK5 is connected to a Mini-Circuits VAT-20+ attenuator, and the signal was measured from the antenna port 1.

attenuator to make sure it introduces ignorable distortions to signal.

After conducting the experiments in a controlled setup to measure the effect of the transmitter's defects on the signal, one should conduct another set of controlled experiments using two 802.11p prototypes such as Cohda MK5 OBU, to determine the effect of the receiver's defects on the signal. The two prototypes should be connected using a cable to avoid any distortions caused by the wireless channel. An attenuator such as Mini-Circuits VAT-20+ should be used to reduce the risk of damage to the prototypes. In this set of experiments the signal passes through both transmitter and receiver, and it gets affected by

Figure 6.3 Setup of measurements using two MK5 units over a stationary link. The antenna is MGW-303 which includes two WiFi antennas and a GPS antenna.

the defects of both hardware, without any distortions caused by the wireless channel. By comparing the results with the other controlled experiment, one can determine the effect of the receiver's defects on the signal which is extremely hard to measure and requires specialized measuring tools and techniques.

Next experiment should be conducted over a static wireless link to measure the effect of wireless channel on the bit error rate. To connect the two 802.11p prototypes over a wireless link, one can use MGW-303 antenna which includes two WiFi antennas and a GPS antenna. This experiment simulates a communication between two vehicles when they are stationary. Fig. 6.3 shows an example of such setup.

The final experiments should represent the extreme case scenario which is when two vehicles are moving and communicating with each other. The same two devices as the last experiment can be used to study the effect of fast varying channels of vehicular networks on bit error rates. It is preferred to use an antenna with a magnetic mount such as MGW-303 antenna. It makes it possible to mount it on the roof of the car to minimize the antenna movement and vibration while driving, minimizing the noise caused by the movement. One can also collect traces on these experiment to evaluate the UEP algorithm and study its impact on improving the wireless communication in vehicular networks. We conducted the the last two experiments on 802.11g chipsets in Section 4.1.3.

Chapter 7

Conclusion

We present an in-depth study of the surprising impacts of LPF defects on the performance of today's 802.11 systems. Based on microscopic measurements and models, we show that LPF roll-off causes notable signal-level defects in off-the-shelf 802.11 chips, and illustrate how the defects propagate to the upper-layers of wireless communication, reshaping bit error patterns and degrading link performance.

Our results show that the impacts of LPF defects are particularly pronouncing on mobile links, where LPF roll-off becomes the dominant source of error when the effects of varying channel are averaged out after a short period of movement. Driven by this insight, we develop a novel unequal error protection algorithm that reorders video frame bytes based on the error pattern caused by LPF defects, which substantially enhances video streaming performance in mobile environments. A major advantage of our UEP algorithm is that it can be combined with any physical or MAC layer schemes. It also has a great potential to be used in forward error correction, channel coding, and many other applications to bring about improvements on mobile links.

REFERENCES

REFERENCES

[1] Cisco visual networking index, http://www.cisco.com/.

[2] Cohda wireless, https://unex.com.tw/.

[3] Cohda wireless, http://www.cohdawireless.com/.

[4] Denso, https://www.denso.com/.

[5] ffmpeg, http://www.ffmpeg.org/.

[6] The gnu software radio, http://gnuradio.org/trac.

[7] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11g-2003*.

[8] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11n-2009*.

[9] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11ac*.

[10] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11ad*.

[11] Moscow state university video quality measurement tool (msu vqmt), http://compression.ru/.

[12] Qualcomm research, https://www.qualcomm.com/invention/research.

[13] Streaming media, http://www.streamingmedia.com/.

[14] x264 - a free h.264/avc encoder, http://www.videolan.org/developers/x264.html.

[15] x264 licensing , https://licensing.x264.org/en/.

[16] Ieee standard for telecommunications and information exchange between systems - lan/man specific requirements - part 11: Wireless medium access control (mac) and physical layer (phy) specifications: High speed physical layer in the 5 ghz band. *IEEE Std 802.11a-1999*, pages 1–102, Dec 1999.

[17] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pages 1–51, July 2010.

[18] A. Schulman, D. Levin, and N. Spring. Crawdad data set umd/sigcomm2008 (v. 2009-03-02).

[19] Siripuram Aditya and Sachin Katti. Flexcast: Graceful wireless video streaming. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 277–288, New York, NY, USA, 2011. ACM.

[20] R. Ahola, A. Aktas, J. Wilson, K. R. Rao, F. Jonsson, I. Hyyrylainen, A. Brolin, T. Hakala, A. Friman, T. Makiniemi, J. Hanze, M. Sanden, D. Wallner, Yuxin Guo, T. Lagerstam, L. Noguer, T. Knuuttila, P. Olofsson, and M. Ismail. A single-chip cmos transceiver for 802.11a/b/g wireless lans. *IEEE Journal of Solid-State Circuits*, 39(12):2250–2258, Dec 2004.

[21] Fan Bai, Daniel D. Stancil, and Hariharan Krishnan. Toward understanding characteristics of dedicated short range communications (dsrc) from a perspective of vehicular network engineers. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 329–340, New York, NY, USA, 2010. ACM.

[22] Laura Bernadó, Nicolai Czink, Thomas Zemen, and Pavle Belanovic. Physical layer simulation results for ieee 802.11 p using vehicular non-stationary channel model. In *Communications Workshops (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[23] Apurv Bhartia, Yi-Chao Chen, Swati Rallapalli, and Lili Qiu. Harnessing frequency diversity in wi-fi networks. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 253–264, New York, NY, USA, 2011. ACM.

[24] Bastian Bloessl, Mario Gerla, and Falko Dressler. Ieee802. 11p in fast fading scenarios: from traces to comparative studies of receive algorithms. In *Proceedings of the*

First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services, pages 1–5. ACM, 2016.

[25] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. An ieee 802.11 a/g/p ofdm receiver for gnu radio. In *Proceedings of the second workshop on Software radio implementation forum*, pages 9–16. ACM, 2013.

[26] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. Towards an open source ieee 802.11 p stack: a full sdr-based transceiver in gnu radio. In *Vehicular Networking Conference (VNC), 2013 IEEE*, pages 143–149. IEEE, 2013.

[27] C. Phillips and S. Singh. Crawdad data set pdx/vwave (v. 2009-07-04).

[28] B. Chen, Z. Zhou, Y. Zhao, and H. Yu. Efficient error estimating coding: Feasibility and applications. *IEEE/ACM Transactions on Networking*, 20(1):29–44, Feb 2012.

[29] Tim Das. Practical considerations for low noise amplifier design. *Freescale Semiconductors, Inc*, 2013.

[30] Thierry Ernst. The information technology era of the vehicular industry. *SIGCOMM Comput. Commun. Rev.*, 36(2):49–52, April 2006.

[31] Matthew Gast. *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2005.

[32] F. Gringoli and L. Nava. Open firmware for WiFi networks. http://www.ing.unibs.it/openfwwf/.

[33] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53–53, January 2011.

[34] Bo Han, Lusheng Ji, Seungjoon Lee, B. Bhattacharjee, and R.R. Miller. All bits are not equal - a study of ieee 802.11 communication bit errors. In *INFOCOM 2009, IEEE*, pages 1602–1610, April 2009.

[35] Bo Han, Lusheng Ji, Seungjoon Lee, Bobby Bhattacharjee, and Robert R Miller. Are all bits equal?: experimental study of ieee 802.11 communication bit errors. *IEEE/ACM Transactions on Networking (TON)*, 20(6):1695–1706, 2012.

[36] Bo Han, Aaron Schulman, Francesco Gringoli, Neil Spring, Bobby Bhattacharjee, Lorenzo Nava, Lusheng Ji, Seungjoon Lee, and Robert Miller. Maranello: Practical partial packet recovery for 802.11. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 14–14, Berkeley, CA, USA, 2010. USENIX Association.

[37] Wei Hu, Jin Xie, and Zhenghao Zhang. pmore: Exploiting partial packets in opportunistic routing. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*.

[38] Jun Huang, Yu Wang, and Guoliang Xing. Lead: Leveraging protocol signatures for improving wireless link performance. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 333–346, New York, NY, USA, 2013. ACM.

[39] Szymon Jakubczak, Hariharan Rahul, and Dina Katabi. One-Size-Fits-All Wireless Video. In *Proc. Eighth ACM SIGCOMM HotNets Workshop*, New York City, NY, October 2009.

[40] Kyle Jamieson and Hari Balakrishnan. Ppr: Partial packet recovery for wireless networks. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, pages 409–420, New York, NY, USA, 2007. ACM.

[41] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid a framework for video transmission and quality evaluation. In Peter Kemper and WilliamH. Sanders, editors, *Computer Performance Evaluation. Modelling Techniques and Tools*, volume 2794 of *Lecture Notes in Computer Science*, pages 255–272. Springer Berlin Heidelberg, 2003.

[42] Zoran Kotevski and Pece Mitrevski. *ICT Innovations 2009*, chapter Experimental Comparison of PSNR and SSIM Metrics for Video Quality Estimation, pages 357–366. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[43] V. Kukshya and H. Krishnan. Experimental measurements and modeling for vehicle-to-vehicle dedicated short range communication (dsrc) wireless channels. In *IEEE Vehicular Technology Conference*, pages 1–5, Sept 2006.

[44] Zhenjiang Li, Yaxiong Xie, Mo Li, and Kyle Jamieson. Recitation: Rehearsing wireless packet reception in software. In Serge Fdida, Giovanni Pau, Sneha Kumar Kasera, and Heather Zheng, editors, *MobiCom*, pages 291–303. ACM, 2015.

[45] Kate Ching-Ju Lin, Nate Kushman, and Dina Katabi. Ziptx: Harnessing partial packets in 802.11 networks. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, pages 351–362, New York, NY, USA, 2008. ACM.

[46] Xiao Lin Liu, Wenjun Hu, Qifan Pu, Feng Wu, and Yongguang Zhang. Parcast: Soft video delivery in mimo-ofdm wlans. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 233–244, New York, NY, USA, 2012. ACM.

[47] Daniele Lo Iacono and Teo Cupaiuolo. Power efficient sdr implementation of ieee 802.11a/p physical layer. *J. Signal Process. Syst.*, 73(3):281–289, December 2013.

[48] Ratul Mahajan, Jitendra Padhye, Sharad Agarwal, and Brian Zill. High performance vehicular connectivity with opportunistic erasure coding. In *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, pages 237–248, Boston, MA, 2012. USENIX.

[49] J. Mittag, S. Papanastasiou, H. Hartenstein, and E. G. Strom. Enabling accurate cross-layer phy/mac/net simulation studies of vehicular communication networks. *Proceedings of the IEEE*, 99(7):1311–1326, July 2011.

[50] Allen Miu, Hari Balakrishnan, and Can Emre Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, MobiCom '05, pages 16–30, New York, NY, USA, 2005. ACM.

[51] Jens-Rainer Ohm. Bildsignalverarbeitung fuer multimedia-systeme. skript, 1999. *Skript*.

[52] T. H. Pham, I. V. McLoughlin, and S. A. Fahmy. Shaping spectral leakage for ieee 802.11p vehicular communications. In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2014.

[53] I.E. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. Wiley, 2004.

[54] A. Sassi, F. Charfi, L. KAMOUN, Y. Elhillali, and A. Rivenq. OFDM transmission performance evaluation in V2X communication. *International Journal of Computer Science Issues*, 9:141–148, 2012.

[55] Sayandeep Sen, Syed Gilani, Shreesha Srinath, Stephen Schmitt, and Suman Banerjee. Design and implementation of an "approximate" communication system for wireless media applications. *SIGCOMM Comput. Commun. Rev.*, 41(4):–, August 2010.

[56] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Accurate: Constellation based rate estimation in wireless networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 12–12, Berkeley, CA, USA, 2010. USENIX Association.

[57] S. Subik and C. Wietfeld. Integrated pmr-broadband-ip network for secure realtime multimedia information sharing. In *Technologies for Homeland Security (HST), 2011 IEEE International Conference on*, pages 20–25, Nov 2011.

[58] W. Vandenberghe, I. Moerman, and P. Demeester. Approximation of the ieee 802.11p standard using commercial off-the-shelf ieee 802.11a hardware. In *2011 11th International Conference on ITS Telecommunications*, pages 21–26, Aug 2011.

[59] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-layer wireless bit rate adaptation. *SIGCOMM Comput. Commun. Rev.*, 39(4):3–14, August 2009.

[60] G. Wang, S. Zhang, K. Wu, Q. Zhang, and L. M. Ni. Tim: Fine-grained rate adaptation in wlans. *IEEE Transactions on Mobile Computing*, 15(3):748–761, March 2016.

[61] K Daniel Wong. *Fundamentals of wireless communication engineering technologies*, volume 98. John Wiley & Sons, 2011.

[62] Grace R. Woo, Pouya Kheradpour, Dawei Shen, and Dina Katabi. Beyond the bits: Cooperative packet recovery using physical layer information. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, pages 147–158, New York, NY, USA, 2007. ACM.

[63] Xiufeng Xie, Xinyu Zhang, Swarun Kumar, and Li Erran Li. pistream: Physical layer informed adaptive video streaming over lte. *GetMobile: Mobile Comp. and Comm.*, 20(2):31–34, October 2016.

[64] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 53–64, New York, NY, USA, 2015. ACM.

[65] M. Zargari, L. Y. Nathawad, H. Samavati, S. S. Mehta, A. Kheirkhahi, P. Chen, K. Gong, B. Vakili-Amini, J. A. Hwang, S. W. M. Chen, M. Terrovitis, B. J. Kaczynski, S. Limotyrakis, M. P. Mack, H. Gan, M. Lee, R. T. Chang, H. Dogan, S. Abdollahi-Alibeik, B. Baytekin, K. Onodera, S. Mendis, A. Chang, Y. Rajavi, S. H. M. Jen, D. K. Su, and B. A. Wooley. A dual-band cmos mimo radio soc for ieee 802.11n wireless lan. *IEEE Journal of Solid-State Circuits*, 43(12):2882–2895, Dec 2008.

[66] X. Zhao, H. Lu, C. W. Chen, and J. Wu. Adaptive hybrid digital-analog video transmission in wireless fading channel. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(6):1117–1130, June 2016.