Evaluating and Characterizing the Performance of 802.11 Networks

by

Ali Abedi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Carey Williamson Professor, Department of Computer Science, University of Calgary
Supervisor(s):	Tim Brecht Associate Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member:	Srinivasan Keshav Professor, Cheriton School of Computer Science, University of Waterloo
	Bernard Wong Associate Professor, Cheriton School of Computer Science, University of Waterloo
Internal-External Member:	Patrick Mitran Associate Professor, Department of Electrical and Computer Engineering, University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The 802.11 standard has become the dominant protocol for Wireless Local Area Networks (WLANs). As an indication of its current and growing popularity, it is estimated that over 20 billion WiFi chipsets will be shipped between 2016 and 2021. In a span of less than 20 years, the speed of these networks has increased from 11 Mbps to several Gbps. The ever-increasing demand for more bandwidth required by applications such as large downloads, 4K video streaming, and virtual reality applications, along with the problems caused by interfering WiFi and non-WiFi devices operating on a shared spectrum has made the evaluation, understanding, and optimization of the performance of 802.11 networks an important research topic.

In 802.11 networks, highly variable channel conditions make conducting valid, repeatable, and realistic experiments extremely challenging. Highly variable channel conditions, although representative of what devices actually experience, are often avoided in order to conduct repeatable experiments. In this thesis, we study existing methodologies for the empirical evaluation of 802.11 networks. We show that commonly used methodologies, such as running experiments multiple times and reporting the average along with the confidence interval, can produce misleading results in some environments.

We propose and evaluate a new empirical evaluation methodology that expands the environments in which repeatable evaluations can be conducted for the purpose of comparing competing alternatives. Even with our new methodology, in environments with highly variable channel conditions, distinguishing statistically significant differences can be very difficult because variations in channel conditions lead to large confidence intervals. Moreover, running many experiments is usually very time consuming. Therefore, we propose and evaluate a trace-based approach that combines the realism of experiments with the repeatability of simulators. A key to our approach is that we capture data related to properties of the channel that impact throughput. These traces can be collected under conditions representative of those in which devices are likely to be used and then used to evaluate different algorithms or systems, resulting in fair comparisons because the alternatives are exposed to identical channel conditions.

Finally, we characterize the relationships between the numerous transmission rates in 802.11n networks with the purpose of reducing the complexities caused by the large number of transmission rates when finding the optimal combination of physical-layer features. We find that there are strong relationships between most of the transmission rates over extended periods of time even in environments that involve mobility and experience interference. This work demonstrates that there are significant opportunities for utilizing relationships between rate configurations in designing algorithms that must choose the best combination of physical-layer features to use from a very large space of possibilities.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Tim Brecht, for his continuous support throughout the course of my PhD. I truly appreciate all the time and effort he put into helping me become an independent researcher. Thank you for your enthusiasm, inspiration, and keeping a sense of humor even when I had lost mine.

I thank the members of my PhD committee for their valuable and constructive feedback: Carey Williamson, Patrick Mitran, Srinivasan Keshav, and Bernard Wong.

I would like to thank my colleague, Andrew Heard, for his friendship and hard work on our joint projects. Andrew is a great software engineer and I learned a lot from him. I like to thank my friends Milad Ghaznavi, Ben Cassell, Jim Summers, Tyler Szepesi, Omid Ardakanian, and Saman Barghi for their support and friendship over the past few years.

Lastly, and most importantly, I thank my wife for her tremendous support during the many years of my studies, while being far from her family. I dedicate this thesis to her.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless Local Area Networks (WLANs) are widely used in homes, schools, and many other public places. The growth of the WiFi market is driven by many factors, such as the widespread use of smartphones and the demand for free public WiFi hotspots. It is anticipated that over 20 billion WiFi chipsets will be shipped between 2016 and 2021 [1]. The number of public WiFi hotspots in the world is projected to increase six-fold from 94 million in 2016 to 542 million by 2021 [34]. Despite introduction of 4G and 5G cellular data technologies, WiFi remains the dominant technology for wireless communications on mobile devices. In 2016, 60% of the cellular network traffic was off-loaded[1] to WiFi networks and this number is expected to increase to 63% by 2021. It is estimated that in 2021, of all fixed and mobile IP traffic (i.e., last hop), 50% will be WiFi, 30% will be wired, and 20% will be mobile. The WiFi share of the IP traffic has increased since WiFi contributed to 42% of IP traffic in 2015 [34]. As we rely more on WLANs and applications demand more bandwidth, the performance of these networks in terms of the provided throughput becomes more important.

IEEE 802.11 standards have been the most successful standards for wireless local area networks in the last two decades. It has been estimated that 802.11 will remain the dominant standard for WLANs at least in the near future [2, 11]. A recent study [103] reports that local wireless network bottlenecks are more common than ISP link bottlenecks. More specifically, for home networks with down-stream links faster than 20 Mbps, WiFi networks are usually the root cause of the performance issues, and ISP link bottlenecks are "relatively rare". In the next section, we argue why the performance evaluation and characterization of 802.11 is important. Then, in Section 1.2 we describe the existing

[1] The use of WiFi for delivering data originally targeted for cellular networks.

challenges and shortcomings in the performance evaluation of 802.11 networks. The thesis goals and contributions are presented in Sections 1.3 and 1.4, respectively.

1.1 Motivation

In this section, we describe why the performance evaluation and characterization of 802.11 networks are important.

1.1.1 More Devices, More Interference

The rapid growth of WiFi-equipped devices is a potential threat to the performance of these devices in terms of their obtained throughput. 802.11 networks operate in the 2.4, 5 and 60 GHz ISM[2] bands. These frequency bands are shared by many WiFi and non-WiFi devices. If two or more signals are received simultaneously at the receiver's antenna, it may not be possible to decode any of them.

The reception of unwanted signals at the receiver's antenna is called interference. The source of unwanted signals may be a WiFi or non-WiFi device. Consequently, WiFi devices have to take turns using the spectrum, which results in a significant throughput loss. This problem becomes even more serious as the density of 802.11 devices increases. A recent study [45] reports that, on average, there are 7 devices in every home and the median number of access points which can be seen per household is 20 in developed countries. As a result, it is important to study the performance of 802.11 networks in environments with interference and when used in the presence of many other devices. In this regard, handling different sources of interference is carefully considered in the methodologies we design and the studies conduct in this thesis.

1.1.2 Unspecified Components in 802.11 Standards

802.11 standards precisely define the MAC and physical layer protocols. However, they do not specify how some procedures should be done. For example, 802.11 standards support different physical-layer transmission rates, however, they do not specify when and how the transmission rate should be changed. Another example is MAC-layer frame aggregation which was introduced in the 802.11n standard. Although the standard precisely specifies

[2] Industrial, Scientific and Medical (ISM) radio bands.

how aggregation should be done, it does not provide an algorithm for choosing the number of subframes in an aggregated frame.

In this thesis, we refer to algorithms that select the transmission configurations of 802.11 standards as *optimization algorithms* (e.g., rate adaptation and frame aggregation algorithms). 802.11 standards leave the design of these optimization algorithms to chipset manufacturers and device drivers. As a result, they are active areas of research. Since the performance of optimization algorithms significantly impacts the performance of 802.11 networks, the proper evaluation of such algorithms is crucial. As a result, we carefully consider the support for various optimization algorithms when designing performance evaluation methodologies. Our survey of the performance evaluation methodologies utilized in the literature reveals that a poor evaluation of optimization algorithms is unfortunately common (e.g., associating performance changes that might be due to environmental changes to the performance of an algorithm).

1.1.3 Future 802.11 Standards and Optimization Algorithms

The ever-increasing demand for more wireless bandwidth has motivated the IEEE 802.11 working groups to release new faster 802.11 standards every few years. The research conducted in a wide variety of areas from faster modulation techniques to more efficient MAC-layer protocols have made faster 802.11 standards feasible. For instance, Multiple Input Multiple Output (MIMO) systems were being researched more than a decade before this concept was introduced in the 802.11n standard. Researchers require proper tools to better understand and characterize current standards in order to design more efficient algorithms (e.g., frame aggregation and channel bonding algorithms) and systems (e.g., MIMO communication systems) required for future 802.11 standards. Current methodologies for the performance evaluation of 802.11 networks lack realism or repeatability, and utilizing them to conduct experiments is difficult. With this idea in mind, we design a trace-based evaluation methodology that can be a great tool for designing and evaluating future 802.11 algorithms and protocols. In addition, using our trace collection methodologies, we characterize the relationship between 802.11n transmission rates. We believe that our findings can be very valuable towards designing new more efficient optimization algorithms. For instance, as described in Section 6.6, by utilizing these relationships in a rate adaptation algorithm, we could achieve up to 28% improvement in throughput compared to the default version.

1.2 Performance Evaluation Challenges

In this section, we describe some of the challenges in evaluating the performance of 802.11 networks. In Section 2.2, different solutions that have targeted these challenges are presented along with their shortcomings.

1.2.1 Repeatability

Wireless channels are time variant and the impulse response of the channel at two different times can be different. Repeatability requires that the impulse response of the wireless channel remains identical across two experiments. This issue is more significant when the rate of change in the wireless channel is higher, for example in mobile environments when a WiFi device travels through a multipath field or when WiFi and non-WiFi interference are present. Repeatability is particularly important when comparing multiple competing alternatives, since the fair and valid comparison of these alternatives requires identical conditions across all experiments. In this thesis, we refer to *alternatives* as solutions, systems, or algorithms that depend on 802.11 networks. Some examples might include comparing different 802.11 rate adaptation and frame aggregation algorithms, different versions of TCP, or different video streaming techniques. For example, when comparing the performance of two algorithms A and B, it is valid to conclude that algorithm B outperforms algorithm A *only if* the two algorithms are evaluated under identical conditions. If the experiments can be repeated, then any performance difference can be attributed to differences in algorithms. Otherwise, any performance differences can be *wrongly* attributed to the better or worse performance of the alternatives. To achieve repeatability, every aspect of the experiment that affects the channel needs to be identical, such as the speed, the path taken, and the position and speed of obstacles. Therefore, in reality it can be difficult or impossible to precisely repeat an experiment.

1.2.2 Realism

Sometimes, wireless channel models are used in the evaluation of the performance of 802.11 networks. These models are used in the analysis, simulation, and emulation of wireless channels. In the process of designing a channel model, many details are ignored to make modeling feasible. The level of details involved in channel models vary, from very simple models that assume a perfect circle as the transmission range [93] to more complicated models trained by empirical data [49, 84]. Although such models are useful in studying particular aspects of wireless channels, they lack realism when a high level of fidelity is required.

Unfortunately, it is extremely difficult to accurately model the behavior of wireless channels. It requires mathematically modeling the physical properties of signals traveling through space, bouncing off, being absorbed by and passing through walls, ceilings, people and other objects. Signal propagation can also be affected by the materials found in these objects, which means that different models may be required for different materials. A model of a home may be quite different from a model used for an office and in fact, in order to obtain realistic models, one may need different models for every home and every office. Such models also need to include representations of interference from other WiFi and non-WiFi devices and their patterns of use. The addition of mobility into these models makes the problem substantially more difficult if not intractable. As a result, evaluation techniques that use such models can suffer from a lack of realism (more details are provided in Section 2.2.2).

1.2.3 Representativeness

The performance evaluation of 802.11 networks should be conducted in the environments that are representative of the environments where these networks may be used. Unfortunately, to address other challenges such as repeatability, researchers are often forced to conduct 802.11 experiments under conditions that are not representative. For example, performance evaluations may be performed in interference-free environments [89, 114], while actual environments are prone to different sorts of interference [79, 95].

1.2.4 Comprehensiveness

Finally, a comprehensive evaluation requires testing in *many* representative environments. Unfortunately, it is difficult or very time consuming to conduct many experiments in a variety of environments. Therefore, evaluations are usually confined to a very small number of (representative or non-representative) environments [70, 75, 89, 114].

Ideally, it is desirable to achieve all of these four criteria when evaluating the performance of an 802.11 network. Unfortunately, in practice, it is extremely challenging to achieve all of them and there are trade-offs between these criteria. Usually, evaluations sacrifice one or more of these goals in order to achieve others.

1.3 Goals

An important aspect of any type of research is being able to fairly and validly evaluate, compare, understand, and draw conclusions regarding the relative merits of multiple competing systems or techniques. In 802.11 networks, performing fair comparisons of different alternatives can be extremely challenging because channel conditions can vary significantly over time. As a result, analytic models, simulation and channel emulation are enticing because they can be used to ensure that each alternative being compared is exposed to precisely the same channel conditions. Unfortunately, wireless channel models generally suffer from a lack of realism (as described in Section 2.2.2).

Therefore, empirical evaluations are highly desirable because they can provide a level of realism and accuracy that is difficult to achieve otherwise. However, empirical evaluations are not a panacea. Because channel conditions can vary over time, conducting repeatable experiments can be very challenging. One of the goals of this work is to understand the possibilities and limitations of conducting empirical measurements when using increasingly popular 802.11 networks and to examine other alternative methodologies. One of these methodologies is the trace-based evaluation of 802.11 networks. At a high-level, the goal of this approach is to enable traces to be collected with relative ease, under a variety of channel conditions. Each trace is processed using the alternatives being studied, to evaluate their performance. Because each alternative uses exactly the same trace, this enables the fair, realistic, and repeatable comparison of multiple alternatives. Another goal of this thesis is to better understand 802.11 networks by characterizing these networks for the purpose of designing new performance evaluation methodologies and improving the performance of optimization algorithms.

The focus of our research is on the performance evaluation of 802.11 (WiFi) networks using widely available commodity WiFi devices. This strategy enables all researchers to participate in and utilize the findings of our projects with reasonable effort. Specifically, we avoid using specialized hardware such as software-defined radios (SDR) for the following reasons. Although software-defined radios are more flexible than commodity devices and provide in-depth information, such as the bit error rate, they may not be able to support high transmission rates supported by some 802.11 standards due to hardware limitations [19, 110, 112]. In addition, implementing the entire MAC and physical layers in a software radio is a difficult task, therefore, an abstraction of the protocol stack is usually implemented, which does not match the behavior and performance of an actual 802.11 device [110]. As a result, we do not utilize software-defined radios in our research.

1.4 Contributions

The contributions of this thesis are in three areas. For the first time in 802.11 networks, we evaluate the efficacy of different existing empirical performance evaluation techniques. We propose the use of a new methodology that expands the scenarios in which repeatable 802.11 experiments can be conducted. However, there are scenarios with highly variable channel conditions where we cannot draw conclusions about the differences between the alternatives due to large confidence intervals. As a result, we design, develop and evaluate a novel trace-based evaluation methodology for 802.11g and 802.11n networks. This approach benefits from the realism of traces captured during actual experiments and repeatability achieved by using the same trace to evaluate different alternatives. Finally, in order to better understand 802.11n networks, we characterize the relationships between numerous transmission rates in these networks. We now explain these contributions in detail.

Conducting Repeatable Experiments Using 802.11n Networks

- We examine different existing methodologies for conducting experiments to compare the performance of systems that use 802.11n MIMO networks.
- We show that some commonly used techniques for comparing the performance of different alternatives are flawed, even in highly-controlled environments that are free from interference from other WiFi and non-WiFi devices. Using these techniques could result in misleading conclusions.
- We show that our proposed multiple interleaved trials methodology provides repeatable results and can be used to distinguish differences in performance, even with highly variable channel conditions.

This work is described in Chapter 3 and has been published in ACM SIGOPS Operating Systems Review (OSR), Special Issue on Repeatability and Sharing of Experimental Artifacts (2015) [16]. We have also studied the efficacy of our proposed methodology in achieving repeatability in highly variable cloud computing environments. The results of this study have been published in the proceedings of the 8th ACM/SPEC International Conference on Performance Engineering (ICPE 2017) [14]. This paper demonstrates that our proposed randomized multiple interleaved trials methodology can be used in other settings where environmental changes can impact performance and make it difficult to conduct fair and valid comparisons of competing alternatives. The details of this study are not presented in this thesis.

T-RATE: A Framework for the Trace-Driven Evaluation of 802.11g RAAs

- We collect, in traces, the effect of environmental factors on the throughput of Rate Adaptation Algorithms (RAAs). This includes information pertaining to Channel Access (carrier sensing and virtual carrier sensing) and Channel Error Rate (signal propagation, WiFi and non-WiFi interference).
- We have implemented an 802.11g trace-processing engine that relies on relatively simple and yet highly-accurate channel models. This enables the fair comparison and evaluation of multiple rate adaptation algorithms using identical channel conditions.
- We evaluate the efficacy of a trace-driven approach for comparing RAAs in 802.11 networks and show that it can produce highly-accurate results.

This work is described in Chapter 4 and has been published in the proceedings of the IEEE 22nd International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2014) [12].

T-SIMn: A Trace Collection and Simulation Framework for 802.11n Networks

- We design and evaluate a trace collection methodology (called *indirect measurement*), which is suitable for devices with many transmission rates (e.g., desktop systems with three antennas supporting 96 rates). We demonstrate that indirect measurement can be combined with our trace-based framework for 802.11n networks (T-SIMn) to obtain highly-accurate simulations of a device with 96 rates.
- While designing the indirect measurement methodology, we developed a new technique for estimating the frame error rate of subframes within an aggregated frame that takes into account channel dynamics. We first show that changes in the Received Signal Strength Indicator (RSSI) of ACKs can be used to infer the channel dynamics which in conjunction with the delivery ratio of the first MAC Protocol DATA Unit (MPDU) can be used to accurately estimate the fate of other subframes in an aggregated MPDU (A-MPDU). We then develop a model that can be used with the T-SIMn simulator and demonstrate that the model accurately estimates the fate of subframes.

The design of our 802.11n trace-based framework, joint work with Andrew Heard [50], appears in the proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2016) [17]. As one of the top papers in the conference, this paper was invited for submission for possible fast track publication in a Special Issue of the Elsevier Computer Communications Journal after further extensions. The new materials that appear in the journal paper [15] (i.e., the indirect measurement trace collection methodology) are presented in Chapter 5 along with an overview of the T-SIMn framework (from the original MSWiM paper).

Examining Relationships Between 802.11n Transmission Rates

- We design a methodology for evaluating relationships among rate configurations that can be used in mobile environments with WiFi and non-WiFi interference.
- We characterize relationships under a variety of channel conditions and study changes in relationships over time. Interestingly, we find that many relationships exist, over surprisingly long periods of time, even in the presence of mobility and interference.
- Using our methodology, we find that the Long Guard Interval (LGI) provides higher throughput than the Short Guard Interval (SGI) in several scenarios. This is contrary to the notion that the LGI may not be required in indoor environments [42,78,91,109].
- By using relationships found between configurations, we demonstrate that it is feasible to improve throughput obtained using the Minstrel HT rate adaptation algorithm by up to 28%.

This work is described in Chapter 6 and has been published in the proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2016) [13].

1.5 Chapter Summary

Half of the worldwide IP traffic (to/from end users) will be carried over WiFi networks by 2021, according to a recent CISCO report [34]. Although WiFi networks continue to grow, the need for more bandwidth and the interference, caused by high deployment densities, have created interesting challenges for researchers. A survey of the methodologies used for the performance evaluation of 802.11 networks reveals that existing techniques can suffer from serious flaws. In this thesis, we design an experimental methodology that achieves repeatability in a wider variety of scenarios than previously possible. We also design a trace-based framework for the performance evaluation of 802.11 networks that achieves repeatability and realism at the same time. In addition, conducting performance evaluations using our framework is much easier than conducting experimental measurements, making more representative and comprehensive evaluations easier to achieve. We also characterize the relationship between 802.11n transmission rates and find that a large proportion of rates are highly related. Our findings can be utilized to design more efficient optimization algorithms, such as rate adaptation and frame aggregation algorithms.

Chapter 2

Background and Related Work

2.1 Background

In this section, we provide an overview of the 802.11 MAC and physical layers, the environmental factors, and different types of interference that impact the throughput of 802.11 networks. We focus on aspects of these concepts that help us to characterize and evaluate the performance of 802.11 networks.

2.1.1 802.11 Physical Layer

We now review the 802.11 physical-layer features that affect the throughput of 802.11 networks. In our trace-based simulator (described in Chapters 4 and 5), it is crucial to precisely consider those features that determine the transmission time of a frame in order to achieve realistic results. In addition, understanding these features helps us explain the characterization results presented in Chapter 6.

Modulation and Coding Rate

Modulation is the process of converting a digital bit stream to an analog signal that can be transmitted physically. Depending on the complexity of the modulation technique, different amount of data can be carried by the analog signal. Since wireless networks are prone to error, forward error correction is implemented in 802.11 standards. The coding rate specifies the non-redundant portion of the transmitted data. For example, if the coding rate is k/n, to transmit k bits of useful information, n bits are transmitted ($n - k$ redundant bits are added for robustness).

Channel Width

The Shannon theorem states that if the channel width is increased, the capacity of the channel increases as a result. To increase the physical transmission rate, the 802.11n standard introduced the 40 MHz channel width option, which combines two 20 MHz channels. Similarly, 60 and 80 MHz channels have been added to the 802.11ac standard.

Multiple-Input/Multiple-Output (MIMO)

To explain MIMO, we need to understand the basics of the modulation techniques used in the 802.11 physical layer. Two modulation techniques have been used in the design of 802.11 networks, namely, Direct-Sequence Spread Spectrum (DSSS) and Orthogonal Frequency Division Multiplexing (OFDM). The 802.11b standard adopts DSSS, while 802.11g supports DSSS for backward compatibility with 802.11b devices. Other standards (802.11 a/g/n/ac) use OFDM modulation techniques in their physical layer. Since OFDM has become the popular modulation technique in the 802.11 standard, we now describe OFDM in more detail.

In Orthogonal Frequency Division Multiplexing, the channel bandwidth is divided into smaller 312.5 KHz subcarriers. These subcarriers are used to transmit data in parallel. In 802.11a/g, the 20 MHz channel is divided into 64 subcarriers. 48 of them are used to transmit data and 4 are utilized as pilot carriers[1]. Pilot carriers transmit a predefined bit sequence to track channel quality. 802.11n utilizes 52 subcarriers and 4 pilot carriers for use with 20 MHz and 108 subcarriers and 6 pilot carriers for 40 MHz channels.

Multiple-Input/Multiple-Output (MIMO) is the most important feature of the physical-layer of the 802.11n standard when compared with that of older 802.11 standards, since the physical-layer transmission rate is increased by up to four times using MIMO. In contrast to 802.11a/b/g, 802.11n utilizes multiple antennas for transmission and reception of data streams. Note that while some 802.11a/b/g devices have two antennas, these antennas are only used for *antenna diversity* (in which only the best signal from one antenna is used at any time). Therefore, only one stream of data is transmitted. In contrast, MIMO utilizes a technique called spatial multiplexing to transmit multiple independent streams of data concurrently.

[1] The rest of the subcarriers are unused.

Spatial multiplexing takes advantage of the propagation properties of signals. The random nature of signal propagation in multi-path scattering environments has enabled MIMO technology to be used to transmit multiple streams of data simultaneously, using multiple antennas, over the same wireless channel and successfully decode and merge the streams of data at the receiver [41]. Figure 2.1 illustrates a 2x2 MIMO configuration (i.e., the sender and receiver both use two antennas for communication). At the transmitter side, two independent signals, t_1 and t_2, are transmitted simultaneously. h_{ij} denotes the channel gain (i.e., attenuation and phase shift) for each antenna pair i and j (transmitter and receiver antenna, respectively). The received signal at each antenna is calculated using Equation 2.1:

$$\begin{aligned} r_1 &= h_{11}t_1 + h_{21}t_2 \\ r_2 &= h_{12}t_1 + h_{22}t_2 \end{aligned} \tag{2.1}$$

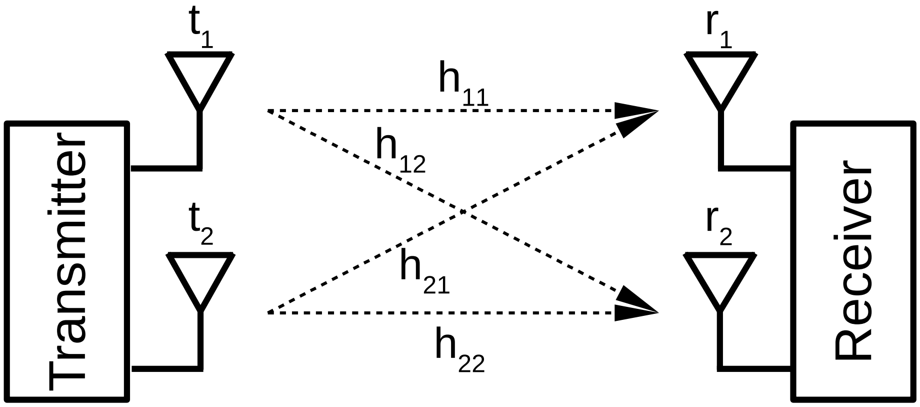


Figure 2.1: A MIMO system in a 2x2 configuration

In 802.11n, the receiver measures the channel gain, h_{ij}, for each antenna using a training field in the frame preamble. If the transmission path between the antenna pairs are independent, Equation 2.1 can be solved and t_1 and t_2 can be calculated.

802.11 chipsets measure h_{ijs} for each OFDM subcarrier, where i and j are the transmitting and receiving antennas, respectively, and s is the OFDM subcarrier. The matrix H_s specifies the channel for subcarrier s:

$$H_s = \begin{bmatrix} h_{11s} & h_{12s} \\ h_{21s} & h_{22s} \end{bmatrix} \tag{2.2}$$

In general, H_s is an $N \times M$ matrix, where N and M are the number of transmitting and receiving antennas, respectively. The set of all H_s matrices (for all subcarriers), defined as H, is the Channel State Information (CSI).

Guard Interval

In the OFDM scheme, symbols (carrying different numbers of bits depending on the data rate) are transmitted over multiple sub-carriers in parallel. When the transmission of the current symbol is done, the next round of transmissions is started if more data needs to be transmitted. Due to multipath fading, the symbols in the previous round could interfere with the current symbols; this problem is called Inter-Symbol Interference. To prevent this problem, a gap is inserted between the consecutive symbols so that all the shadow signals of the previous symbol are gone by the time the next symbol is started. This gap is called the guard interval. 802.11n/ac standards support 400 ns and 800 ns guard intervals.

2.1.2 802.11 MAC Fundamentals

The performance of 802.11 networks also depends heavily on the efficiency of the MAC layer. Our trace-based simulator precisely implements MAC-layer features, such as the Distributed Coordination Function (DCF), explicit acknowledgment, and frame aggregation to achieve realistic results. We now describe these features and the MAC-layer low-efficiency problem [42], along with the frame aggregation mechanism that has been designed to solve this problem.

Explicit Acknowledgment

Unlike wired networks, such as IEEE 802.3, where the bit error rate is very low, wireless networks are prone to noise, interference, and multipath fading. Therefore, the bit error rate can be very high compared to wired networks. As a result, network protocol designers decided to add positive acknowledgments to 802.11 networks to handle frame loss at the physical layer and avoid longer overheads involved in handling retransmission at higher layers. When an 802.11 station receives a data frame correctly, it sends back an ACK frame to the sender to acknowledge the correct reception of the frame. The transmission is considered successful only when the acknowledgment frame is received by the sender. If no ACK is received, the data frame is considered lost and a retransmission procedure is initiated. A frame is retransmitted until an ACK is received or the retransmission limit is reached and the frame is dropped. Note that the actual value of the retransmission limit depends on the hardware, drivers, and rate adaptation algorithms.

DCF Operation

The Distributed Coordination Function (DCF) is a widely-implemented strategy for channel access in the 802.11 standard. DCF is a set of rules that defines contention-based access to the wireless medium without any control messages.

Every station that wants to access the channel should sense the channel before starting the transmission. The channel must be idle for the duration of DIFS (DCF Interframe Space) before a station can access the medium. If the channel is detected to be idle, a random backoff time is chosen. The channel has to remain idle for the extra wait time specified by the random backoff time. This mechanism is used in DCF to obtain channel access in a distributed fashion without any need for control messages. Among the competing stations, the one with the shortest backoff time will start its transmission first. Upon detecting a transmission, other stations stop their backoff timer and resume this procedure when the channel is detected to be idle again. If two or more stations choose the same random backoff time, frame collision happens due to concurrent transmissions. Therefore no ACK is received by any station, frame loss is detected, and a retransmission procedure is initiated. To efficiently avoid frame collision when many stations compete for the medium, 802.11 standards increase the backoff time exponentially in case of frame loss.

Frame Aggregation

The 802.11n standard is able to achieve 450 Mbps at the physical-layer thanks to new features such as MIMO and channel bonding. However, the MAC-layer throughput is just a fraction of the physical-layer speed due to the overhead imposed by explicit acknowledgments and the use of DCF. The efficiency of 802.11 networks is known to be about $50-60\%$ [41] due to the overhead in the MAC and physical layers. For example, if a WiFi device achieves 17 Mbps of physical layer throughput, the application layer throughput (i.e., goodput) will be slightly over 10 Mbps. In order to reduce this overhead, a MAC-layer frame aggregation mechanism has been introduced in 802.11n that aggregates multiple MAC-layer frames into one larger physical frame. As a result, rather than sending multiple small frames, which require their own backoff, DCF inter-frame spacing and acknowledgments, one aggregated frame containing multiple smaller frames is transmitted instead. Note that the frame transmission is still unicast and all subframes belong the same destination. The 802.11n standard defines two types of frame aggregation: *Aggregated MAC Protocol DATA Unit (A-MPDU)* and *Aggregated MAC Service DATA Unit (A-MSDU)*. These two types of frame aggregation differ by where in the protocol stack aggregation is done. In this thesis, we use A-MPDU frame aggregation, since it is more widely supported by WiFi devices, including the Atheros devices used in this thesis. During A-MPDU frame aggregation, a block-acknowledgment is sent back to the transmitter by the receiver to acknowledge the individual frames inside the aggregated frame. This mechanism increases the efficiency of 802.11n significantly.

To demonstrate the importance of frame aggregation (i.e., A-AMPDU), Figure 2.2 shows the maximum theoretical throughput obtained using the highest 802.11n physical-layer transmission rate for one, two and three spatial streams, respectively. Without frame aggregation throughput is limited to about 50 Mbps. However, when aggregating 32 frames, throughput increases to 350 Mbps.

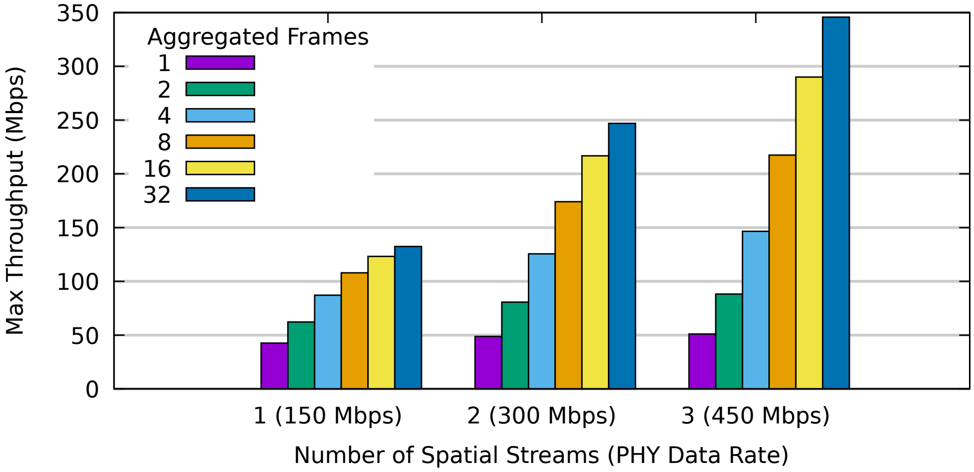


Figure 2.2: Frame Aggregation: Maximum Theoretical Throughput [17]

2.1.3 Throughput of 802.11 Networks

Although WiFi devices are sold with promises of high transmission rates, the actual achieved throughput is only a fraction of the advertised rates. Unfortunately, many environmental factors and the 802.11 MAC and physical layer efficiency significantly reduce the achievable throughput (Figure 2.3).

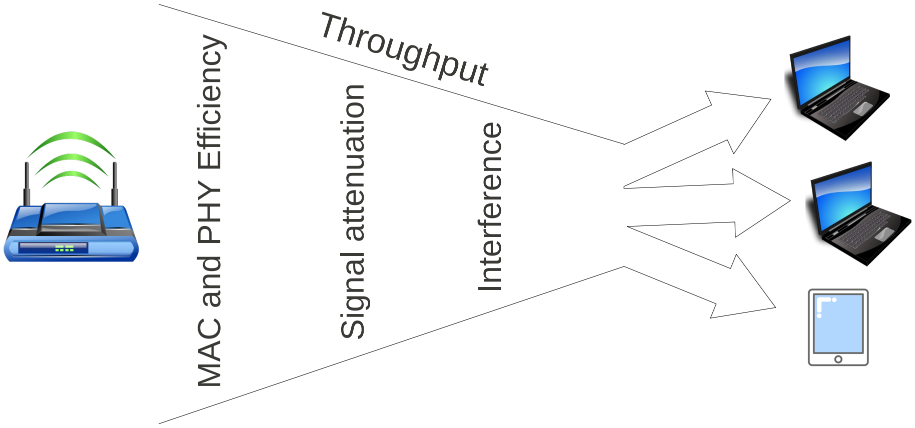


Figure 2.3: The effect of 802.11 efficiency and environmental factors on the performance of an 802.11 network

Environmental factors

The throughput of an 802.11 network is related to the quality of the signal observed at the receiver's antenna. Signal attenuation and interference can significantly affect the received signal quality. If the sender and receiver are far apart or there is no line of sight between them, a lower transmission rate has to be used to cope with the high bit error rate resulting from a weak signal. In a typical environment where 802.11 networks are used, such as home and office environments, the highest transmission rate is limited by the distance and the obstacles between the sender and receiver.

Another problem in 802.11 networks that further limits the maximum achievable throughput is interference. 802.11 networks operate in the 2.4 or 5 GHz ISM bands[2]. These frequency bands are shared by many WiFi and non-WiFi devices. If two or more signals are received simultaneously at the receiver's antenna, none of them can be decoded, since the received signals are added and create a new signal that is not representative of any of the signals. The reception of unwanted signals at the receiver's antenna is called interference.

[2] Industrial, Scientific and Medical (ISM) radio bands.

The source of the unwanted signal can be a WiFi or non-WiFi device. It is crucial to accurately handle different sources of interference in our trace-based framework. In addition, interference affects performance evaluation methodologies and the characterization of 802.11 networks. Therefore, we now briefly explain WiFi and non-WiFi interference:

WiFi interference

If two 802.11 frames are received simultaneously at the receiver's antenna, none of them can be decoded due to interference. The interfering devices can be associated with the same or a different access point. 802.11 devices in two neighboring networks, as well as devices in the same network, can interfere with each other if they share the same wireless channel or operate on *overlapping channels*. Two channels are overlapping if they partially share the same spectrum, as illustrated in Figure 2.4. The figure shows overlapping and non-overlapping channels in the 802.11b standard [4]. Channels 1, 6 , and 11 are non-overlapping (solid lines) while other channels (dotted lines) overlap with at least one of these channels. 802.11 devices in two networks with overlapping channels can also interfere with each other.

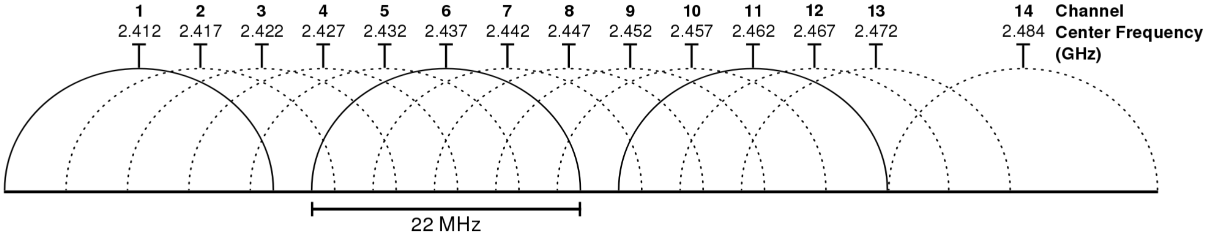


Figure 2.4: The 802.11b channels in the 2.4 GHz ISM band [4]

Two nearby 802.11 networks with overlapping channels interfere with each other if they transmit simultaneously. Therefore, they need to time share the transmission medium and as a result the maximum achievable throughput is decreased significantly. In environments where WiFi networks are heavily deployed, such as office buildings, apartment buildings, and dense residential areas, the problem of WiFi interference can be severe and the throughput of 802.11 networks in these environments is reduced significantly as a result [24, 40].

Non-WiFi interference

In addition to interfering 802.11 networks, there are many non-WiFi devices that operate in the 2.4 or 5 GHz band. Non-WiFi devices, such as microwave ovens, cordless phones, and bluetooth-based devices, are often present in an environment where 802.11 networks are used. The interference coming from these devices can be destructive. As a result, the effective throughput is further decreased when non-WiFi interferers are active on the wireless channel. Studies show that non-WiFi devices are highly active across many locations including public places, enterprise and residential environments [79,95]. Surprisingly, a relatively high level of activity from non-WiFi devices is reported even at midnight [95]. This time is often considered to be interference-free for conducting experiments [37, 89, 114], therefore, the experiments might be affected by one or more unknown interferers and a wrong conclusion might be drawn from the results of the experiments.

2.2 Related Work

In this section, we discuss the efficacy of performance evaluation methodologies for 802.11 networks to understand the shortcomings of these techniques in terms of four criteria, namely realism, repeatability, representativeness and comprehensiveness. Our goal is to design a performance evaluation methodology that achieves all four criteria. We present an overview of state-of-the-art performance evaluation methodologies for 802.11 networks, which generally fall into four categories, namely experiments, simulation, emulation, and trace-based simulation. Finally, we review studies that characterize 802.11 networks for the purpose of improving the performance of 802.11 optimization algorithms, such as rate adaptation and frame aggregation algorithms.

2.2.1 Experimental Evaluation of 802.11 Networks

The most intuitive methodology for evaluating the performance of 802.11 networks is to conduct experiments using real systems. With this approach, the performance of all alternatives is measured using an actual 802.11 network. The main advantage of this methodology is realism, since it uses actual devices and wireless channels. Unfortunately, channel conditions vary with time and changes can be difficult to monitor without relatively expensive equipment. As a result, existing research often compares alternatives by attempting to run experiments using *known*, or *repeatable* conditions without confirming that experiments can be repeated.

Several studies [31,35,40,56,59,96] have considered the issue of repeatability when performing empirical performance evaluations of 802.11 networks. Ganu et al. [40] show that the calibration of testbed hardware is essential for reproducibility of experiments. They report on variations across five runs in a semi-controlled environment using the 802.11b standard. Despite using calibrated equipment, a variation of up to 30% is observed across different runs. However, they do not provide any analysis about whether or not the results could be repeated. In Chapter 3, we present results of 802.11n MIMO experiments along with a statistical analysis of these results and report on the ability to repeat experiments as well as fairly compare multiple alternatives. We show that the results obtained using existing methodologies could be flawed and can lead to invalid conclusions. We propose a new empirical performance evaluation methodology and show that repeatable results can be obtained using this methodology when evaluating 802.11n networks.

Burchfield et al. [31] study different factors that affect the repeatability of 802.11 experiments. They show that the outcome of an experiment can change significantly due to simple changes in the environment, such as interference or switching the location of the sender and receiver. Therefore, they suggested using wireless channel emulation to solve the repeatability problem. Wireless channel emulators can repeat identical channel conditions across trials [31,59]. However, they suffer from a lack of realism, since they rely on simplified models of wireless channels (see Section 2.2.2 for more details).

In an effort to make experiments involving mobility repeatable, robots have been used to carry a wireless node [35,56,96]. These robots follow the same path for each experiment to attempt to minimize the variation that could be caused by following slightly different paths in every run. Rensfelt et al. [96] evaluate the potential effect of using robots, instead of a person carrying a mobile device, on the repeatability of the results. They show that robots can lower the variability of RSSI in 802.15.4 sensor networks. In Chapter 3, we study the effect of using a toy train to repeat mobile experiments (in terms of RSSI and throughput) in more widely used 802.11n networks. We find that the use of a toy train can potentially reduce the variability across multiple runs. However, it may not be representative of when a person carries a handheld device, due to lower fluctuations of the signal.

To address problems with repeating experiments, researchers suggest avoiding situations where the channel conditions are likely to change rapidly, by conducting experiments "in the middle of the night", "when no one else is around" [89,114], or using the 5 GHz spectrum to avoid interference [70,75,89,114]. Two main pitfalls occur in many of these evaluations, namely *unrepresentativeness* and *the assumption of repeatability*.

Running experiments at midnight may be unrepresentative because in an attempt to ensure that channel conditions do not change across experiments, WiFi and non-WiFi interference (present in most environments) are avoided. For the same reason, running experiments using the 5 GHz spectrum, where less interference is expected, is not representative of the interference-prone 2.4 GHz spectrum. Moreover, signal propagation is different in the 2.4 and 5 GHz spectrum. For instance, the absorption of radio signals passing through obstacles depends on the signal frequency. Therefore, results obtained in the 5 GHz spectrum may not be representative of the 2.4 GHz spectrum.

Another similar approach for avoiding interference is to conduct experiments in anechoic chambers [23,60] where signal reflection and multi-path fading are eliminated and the chamber is protected from external interference by special materials used in the construction of the chambers. While this approach yields repeatable experiments, it also suffers from a lack of representativeness for the same reasons.

In addition, the dynamics of a wireless channel such as the movements of objects is different during the day when people are around and at midnight. Judd et. al. [60] reported that even when the sender and receiver are stationary but the environment is dynamic due to the movement of people, the channel quality varies significantly with time in the same way as when the sender or receiver are mobile. Therefore, the stationary and stable channel conditions achieved when running an experiment at midnight may not be representative of a dynamic channel during working hours.

The second potential pitfall of running experiments at midnight or in the 5 GHz spectrum is the assumption of repeatability. Sometimes experiments are conducted in environments that are expected to be free of interference for example during the night without validating this assumption. Although the background WiFi and non-WiFi traffic is reported to be lower during the night [79, 95], it does not obviate the need for monitoring the channel as the background traffic and interference may not be eliminated. Similarly, despite lower utilization of the 5 GHz spectrum compared to the 2.4 GHz band, it is still necessary to monitor the channel for background traffic and interference all the times. Some work has partially avoided this pitfall by monitoring for WiFi traffic using a sniffer [89,114] during midnight experiments. However, the channel was not monitored for non-WiFi interference. Therefore, the obtained results may be affected by non-WiFi interference and a wrong conclusion might be drawn from the results.

Another approach used when conducting empirical performance evaluations is to run experiments multiple times and report the average performance and some notion of the variability of the obtained results. Unfortunately, running experiments multiple times is usually avoided in practice [75, 89] because it is very difficult and time consuming to conduct a wireless experiment several times. On the other hand, despite running multiple experiments, some papers do not report the standard deviation or confidence intervals [37, 70, 114].

Other papers [47, 92, 107] do report the standard deviation or confidence intervals. However, in our evaluation, we observe that confidence intervals, while useful, may provide a false sense of rigor and validity. In Section 3.4.2, we demonstrate how statistically significant differences were obtained (i.e., non-overlapping confidence intervals) for two sets of identical experiments, even in controlled environments with stationary devices and no WiFi or non-WiFi interference. We show that if an experiment using alternative A is repeated for multiple iterations, followed by multiple iterations using alternative B, changes in channel conditions can result in incorrect conclusions.

2.2.2 Simulation and Emulation of 802.11 Networks

The operation of the Internet protocol suite and wireless channel conditions can be simulated. This requires the use of a variety of models. Usually an event-driven simulator simulates packet transmission between two or more stations and access points and performance metrics such as the achieved throughput and average network delay are reported. NS-2 [5], NS-3 [6], OPNET (became part of Riverbed [9]), Qualnet [10], and OMNET++ [8] are some of the most prominent 802.11 simulators. The advantage of simulation over experiments is that once the simulator is implemented, it is relatively easy to measure performance, unlike experiments that require actually building and configuring networks in different environments, which can be time consuming and difficult. Another advantage is that a simulation can be repeated under identical channel conditions. If model parameters are unchanged, the same channel quality is observed across different iterations of the simulation. Therefore, repeatability can be guaranteed in simulation. On the other hand, all aspects of the systems, protocols, physical environments and channel conditions must be modeled accurately. Because this is extremely difficult, simulators suffer from a lack of realism as explained in the rest of this section. Simulators also suffer from long execution times when the details of the physical layer are simulated to increase realism [81].

Physical-layer simulation is the most challenging part of the simulation, since it involves modeling the complex propagation and reception of radio signals, while well-defined MAC-layer specifications are much easier to simulate. Figure 2.5 shows the steps involved in simulating the 802.11 physical layer. In the first step, signal propagation is simulated, which requires models for path loss, multipath propagation, and large and small-scale fading. The output of this step is the estimated SNR at the receiver. To accurately model signal propagation, all objects in the vicinity of the wireless devices need to be modeled by considering the exact shape and materials used in the construction of these objects. The precise modeling of objects is required as radio signals are reflected, diffracted, and scattered when interacting with the objects in the environment. The interaction of radio signals with objects changes the amplitude and shape of the signal affecting the reception and correct decoding at the receiver. In addition, multipath propagation has a significant effect on the correct reception of the signal. The correct simulation of multipath propagation also requires precise modeling of the objects in the vicinity of the sender and receiver. Moreover, when the sender, receiver, or objects (including people) move, the signal reception is affected by the Doppler effect. Precise modeling of all mentioned signal propagation characteristics is difficult, therefore, abstract versions are implemented or they are ignored when modeling signal propagation. For instance, the widely used NS-3 simulator currently considers only path loss when simulating radio signal propagation and ignores the Doppler effect, multipath, signal reflection, diffraction or, in general, any signal propagation characteristics related to the interaction of radio signals with objects [102]. In addition, NS-3 currently does not support MIMO [7] mostly due to utilizing propagation models that do not support multipath (which is essential for MIMO). The result of these abstractions are models that can be far from reality in practice [83]. The path loss models used in NS-3 have been shown to perform poorly in terms of matching the actual error rates obtained from experiments [102]. The lack of models for other signal propagation effects further reduces the simulation realism.

Signal propagation modeling estimates the SNR of the received signal at the receiver. As depicted in Figure 2.5, the same process is repeated for all senders that might interfere with each other. A model is required to approximate the effect of the interference by estimating the SINR (signal to interference and noise ratio).

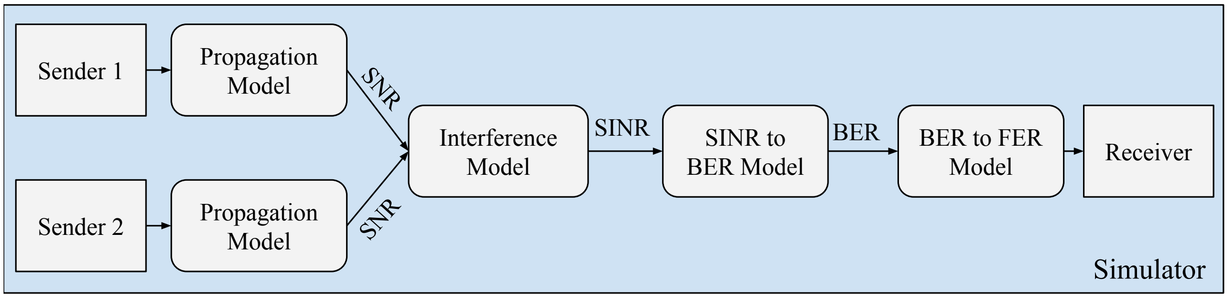


Figure 2.5: Physical layer simulation

In the next step, the SINR is used to estimate the bit error rate (BER) for a given physical-layer transmission rate. The SINR to BER mapping is performed by mathematical modeling or empirical measurements. Several studies [20, 37, 48] have shown that SINR is a very coarse predictor of the bit error rate and can lead to significant errors in estimating the bit error rate or frame error rate from SINR. The main reason behind this inaccuracy is that the SINR is the average SINR over all OFDM sub-carriers. However, due to frequency-selective fading, different sub-carriers undergo independent fading, resulting in a different SINR for each sub-carrier [48]. Therefore, the average SINR is not representative of channel

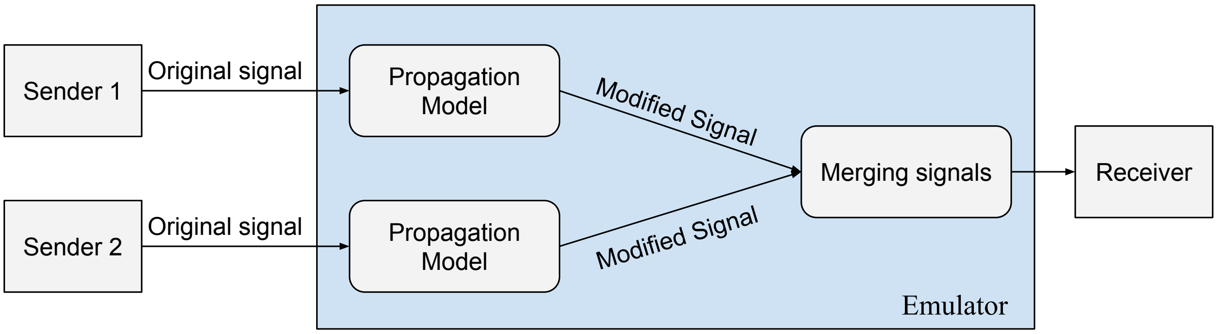


Figure 2.6: Physical layer emulation

quality for all sub-carriers. As a result, using SINR for bit error rate prediction can lead to significant estimation errors.

Finally, the frame error rate (FER) is estimated from the bit error rate by simulating the forward error correction mechanism. In order to precisely calculate the frame error rate, the exact position of bits in error is required to be known. However, such information is not available (i.e., the models do not provide this information) and it is assumed that the position of bits in error follow some distributions (e.g., uniform distribution) and the frame error rate is estimated based on this assumption. Since the recoverability of a frame depends heavily on the position of these bits, a simplified modeling of bits in error may add further inaccuracies in the estimation of the frame error rate [51].

The process of modeling packet reception involves many simplified models and assumptions that provide results that are far from those obtained from actual experiments. Several studies [57,87,105] report that the simulator results may deviate significantly from the experimental results. As a result, the simulation approach is insufficient when high fidelity results are required, as may be the case when trying to compare different rate adaptation and frame aggregation algorithms.

In an emulator [20,30,57,59,87], the analog output signal from the sender's WiFi card is converted to a digital signal, as illustrated in Figure 2.6. The effects of signal propagation such as large-scale path-loss and small-scale fading are then estimated, using a channel model, and the digital signal is modified using a Digital Signal Processing (DSP) engine accordingly. As with the simulation, signal propagation is still modeled inside the DSP engine and the signal is modified based on the channel models in use. Therefore, emulators can also suffer from a lack of realism. Since signals need to be modified physically in the emulation approach, further restrictions are involved when compared to the simulation methodology. For example, when simulating multipath transmission, the speed of the processors used in the hardware of an emulator determines the maximum number of signal copies it can process. The maximum channel bandwidth and the maximum number of clients connected to an access point are also limited by the capabilities and speed of the processors.

Finally, a digital to analog conversion is then performed and the combination of signals from all WiFi and non-WiFi sources is sent to the receiver's WiFi card through an antenna port. The signal is received by a WiFi card and the fate of the packet is determined by the actual hardware rather than SNR-to-FER models used in simulation. The strength of an emulator is that actual devices and protocols can be used and no models are utilized for calculating the bit error rate and frame error rate. The weakness is that it is extremely difficult to emulate complex channel behavior, signal propagation, and interference accurately. In addition, the realism of an emulator is further limited by the hardware used to build the emulator. The inventors of this approach acknowledge the limitations: "real world experiments are essential when high radio channel fidelity is required" [59].

2.2.3 Trace-Based Evaluation of 802.11 Networks

To our knowledge, the first proposal for a trace-based evaluation of a wireless network was in 1997 [86]. This evaluation uses traces from wireless transmissions collected at the end hosts or at a sniffer. Delay and loss statistics are computed from the trace to train a wireless channel model. The trained model is then used to emulate wireless networks using wired networks. Since the wireless channel model is trained using a trace from an actual experiment, it provides more realistic results than mathematical models. This approach was applied to a WaveLAN network with only a single transmission rate. Unfortunately, modern networks such as IEEE 802.11 support multiple transmission rates which makes collecting traces challenging. The problem is that at any time t one would like to know if a packet sent at rate r would be received or not for each available rate.

More recently, traces have been used to evaluate rate adaptation algorithms by combining results obtained from experiments with a simulator. In some cases [58, 116], some portions of the physical models that are commonly used in simulators or emulators are replaced with the traces collected in real experiments. For example, a trace of SNR (Signal-to-Noise Ratio) [116] or CSI (Channel State Information) [48] values obtained from an actual experiment are used to improve the realism of a simulator (we refer to this approach as SNR-to-SIM) [116] and an emulator (we refer to this approach as SNR-to-EMU) [58]. Unfortunately, the SNR often inaccurately predicts the fate of a packet [48] [37]. Although the CSI provides a more accurate measure of channel quality than the SNR [48], both are oblivious to WiFi and non-WiFi interference and do not reflect the level of interference experienced at the receiver. Therefore, accurate frame loss estimation using only SNR or CSI can be impractical, since they have no notion of interference.

In addition, other factors such as multi-path are not captured by the SNR. Although emulators try to model multi-path fading, the number of paths that can be modeled is restricted to a few transmission paths due to hardware limitations and therefore they are unable to model complex environments accurately [59]. Moreover, complex building structures, other obstacles and the materials they are constructed from should be considered which makes accurate multi-path emulation and simulation very difficult.

Another approach that utilizes traces completely replaces the channel propagation model in the NS-3 simulator with an actual trace that specifies the fate of a frame at any given time (we refer to this approach as FL-to-SIM because the Frame Loss (FL) information is directly measured and stored in a trace) [94, 111]. In this case, traces are collected in *ideal interference-free environments* and path loss and fading information are stored in the traces. This approach collects frame loss information for all transmission rates by sending back-to-back frames at different rates in a round-robin fashion. Interference-free environments are required with this approach because the trace collection mechanism assumes that any frame loss is due to path loss or fading, since this approach only replaces the channel model with traces and it simulates interference. Therefore, this approach cannot be used to collect traces in uncontrolled environments with different sources of interference. Consequently, wireless channel models, which may not be realistic, should be used to simulate WiFi and non-WiFi interference. In contrast, our proposed frameworks, namely, T-RATE (in Chapter 4) and T-SIMn (in Chapter 5), do not require an interference-free environment for trace collection, and we directly capture the effect of interference on the channel access and channel error rate during trace collection.

As shown in Table 2.1, previous work on trace-based evaluation of 802.11 networks focuses only on capturing channel propagation properties in a trace (the shaded area in Figure 2.7). Ignoring environmental factors during trace collection introduces inaccuracies in the results obtained using these traces. Therefore, the environmental factors that affect throughput must be captured to accurately replicate real environments. None of the work shown in Table 2.1 is able to collect traces in uncontrolled environments, instead they simulate the effect of environment factors. In contrast, our frameworks capture these factors intrinsically in traces (CT[3]-to-SIM) collected during actual experiments, thus simplifying and minimizing the use of wireless channel models.

Previous work ignores, simulates, or emulates carrier sensing and virtual carrier sens-

[3] Compete Trace.

Methodology	Frame-loss (propagation)	Frame-loss (interference)	SNR	(V) CS* (WiFi)	CS* (non-WiFi)
SNR-to-SIM [116] [48]	✗ Simulated	✗ N/A	✓ Data frames	✗ Simulated	✗ N/A
SNR-to-EMU [58]	✗ Emulated	✗ Emulated	✓ Data frames	✗ Emulated	✗ Emulated
FL-to-SIM [111] [107] [100]	✓ Receiver-based	✗ Sim. WiFi interference	✗ N/A	✗ Simulated	✗ N/A
CT-to-SIM	✓ Sender-based	✓ WiFi & non-WiFi	✓ Data and ACK	✓	✓

Table 2.1: Comparison of trace-based performance evaluation methodologies
* (V) CS: (Virtual) Carrier Sensing

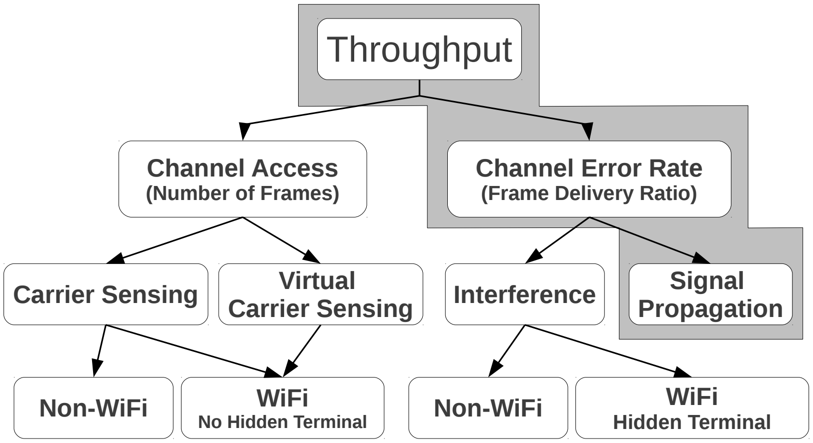


Figure 2.7: Factors affecting throughput

ing. Unfortunately, modeling carrier sensing is as complex as modeling frame loss because it involves modeling signal propagation for interfering sources. It is extremely difficult to estimate the effect of a WiFi or non-WiFi source on the channel observed at the sender, since signal propagation from all other sources should be modeled and requires complex wireless channel models. As a result, simple channel models that suffer from a lack of realism are used. Similarly, previous work ignores or simulates frame loss due to interference. WiFi interference (the hidden terminal problem) is simulated using simple and unrealistic models. For instance, it is assumed that any overlap of two frames results in the loss of both frames [111]. However, in reality, one of these frames may be received correctly if the signal strength of these frames differ considerably.

To our knowledge, none of these methods (e.g., simulation, emulation, and the reviewed trace-based methodologies) are able to handle non-WiFi interference[4]. This is another reason why the environments in which these techniques can be used is restricted. Some studies show that non-WiFi devices are widely used, including public places, and in enterprise and residential environments [79] [95]. A surprisingly high level of activity for non-WiFi devices is reported even at midnight [95]. Hence, a trace collection methodology that can be used in such environments needs to capture the effect of non-WiFi devices. By intrinsically capturing environmental factors that affect throughput, shown in Figure 2.7, we achieve two important goals: 1) We minimize the use of wireless channel models that reduce realism. and, 2) Our trace-based framework can be used in a wider variety of environments than previously possible.

2.2.4 Comparison of the Evaluation Methodologies

There is a trade-off between realism and repeatability in the discussed methodologies. Although experiments are the gold standard for realism, they are usually not repeatable. On the other hand, simulation and emulation provide repeatability at the cost of lower realism. Table 2.2 shows the trade-off between realism and repeatability. Ideally, an evaluation methodology can achieve realism and repeatability at the same time. Trace-driven evaluations attempt to solve this problem by combining the realism of experiments with the repeatability of simulation.

In terms of representativeness, the experimental approach may or may not be representative depending on the scenario under which an evaluation is performed. As mentioned previously more representative environments are usually avoided to achieve repeatability. The wireless channel models used in simulation and emulation are mostly not representative of complex scenarios that are affected by various environmental factors such as different sorts of fading and interference. On the other hand, we have devised trace collection methodologies that allow traces to be collected in any environment (even highly variable channel conditions), resulting in representative outputs if traces are collected in representative environments.

[4] Emulators can incorporate non-WiFi interference for some devices such as cordless phones that can be physically attached to the emulator but they cannot handle many other types of devices, such as, microwaves

Methodology	Realism	Repeatability	Representativeness	Comprehensiveness
Experiment	High	Low	Varies	Difficult
Simulation	Low	High	Low	Almost impossible
Emulation	Low	High	Low	Almost impossible
Trace-based	Relatively high	High	Varies	Relatively easy

Table 2.2: Comparison of evaluation methodologies

Achieving a comprehensive evaluation using the experimental approach is relatively difficult and time consuming, since it requires conducting multiple experiments for each competing alternative in many environments. On the other hand, only one experiment (i.e., trace collection) is required to be conducted in each environment making a comprehensive evaluation much easier. Our vision is to create a public repository that contains a variety of traces collected in different environments. Researchers can contribute to this repository by publishing their traces collected in new environments. This repository can significantly help researchers to increase comprehensiveness when evaluating 802.11 networks. The simulation and emulation techniques are usually incapable of achieving such comprehensive evaluations as complex wireless channel models would have to be developed and validated for each new environment.

As a result, in this thesis, we study the experimental and trace-based approaches to 802.11 performance evaluation to evaluate their potential and limitations.

2.2.5 Characterizing 802.11n Networks

In addition to studying 802.11 performance evaluation methodologies, we have characterized 802.11n networks (in Chapter 6) in an attempt to find relationships between numerous transmission rates. As explained in Chapter 6, optimization algorithms such as rate adaptation algorithms can utilize such relationships to improve their performance by limiting their search space (i.e., when probing for finding the best transmission rate to use). We now provide an overview of the studies that characterize 802.11n networks. To better understand 802.11n networks, researchers have characterized these networks from different perspectives, such as the relationship between physical layer features and throughput [66], jitter and energy efficiency [115]. We now review some of these 802.11n characterization studies. Although some of these studies are not directly related to our work, they help to understand different viewpoints related to the characterization of 802.11n networks.

LaCurts et al. [69] use over 1,400 access points to empirically study the correlation between sender-side SNR and throughput. The results show that SNR is not sufficient to determine the best transmission rate for a network although with sufficient training, SNR can be a good predictor of the throughput for a specific link. Halperin et al. [48] have conducted a similar study and found that due to the frequency-selective fading problem,[5] the SNR is an inaccurate metric of the quality of the communication channel, since it does not represent the quality of the signal received by each OFDM sub-carrier. They propose and evluate a packet delivery prediction model that uses the effective SNR, which is derived from the Channel State Information (CSI). The shortcoming of this technique is that in practice many 802.11n chipsets do not report the channel state information [28,67] and a mechanism is required to transfer CSI data (only available at the receiver) back to the sender.

To our knowledge, the closest work to our characterization study (in Chapter 6) is the study conducted by Kriara et al. [66]. They characterize the positive and negative impact of each physical transmission feature on throughput, FER, and jitter. The authors argue that to maximize throughput, all transmission features should be optimized jointly. In other words, if the transmission features are optimized one by one, a sub-optimal throughput might be achieved. This finding emphasizes the need to limit the search space when selecting the best rate configuration, since a divide and conquer (i.e., optimizing each transmission feature separately) approach does not work.

To limit the search space when optimizing rate configurations, Kriara et al. [67] utilize sender-side RSSI information (i.e., RSSI of received ACKs) to reduce the number of rate configurations that need to be probed. In some scenarios, they are able to improve rate adaptation by reducing the overhead of probing many configurations. The data collection methodology they use [67] requires repeatability across experiments. Therefore, all experiments conducted to measure the relationship between RSSI and transmission features are done in controlled environments with no mobility and no interference other than those introduced by the experimenters. Additionally, their methodology only obtains information about rates if and when they are selected by the RAA.

In contrast with all previous work, in Chapter 6, we characterize the relationship among rate configurations in order to determine how accurately the FER of one rate configuration can estimate the FER of another configuration. We show how these relationships can be used to provide new insights into the behaviour of different physical layer features (e.g., SGI and LGI). Additionally, we demonstrate how relationships might be used to reduce the

[5] OFDM sub-carriers suffer from different and independent fading, rendering the average SNR over all sub-carriers a very coarse and inaccurate metric for the quality of the received signal.

search space when choosing the best combination of physical-layer transmission features.

2.3 Chapter Summary

In this section, we provide background for the 802.11 physical and MAC layers, and review common methodologies for the performance evaluation of these networks. We consider realism, repeatability, representativeness, and comprehensiveness when reviewing the efficacy of these methodologies. Empirical evaluation, which is probably the most intuitive way of conducting performance evaluation, takes advantage of actual devices and wireless channels which makes it the gold standard in terms of realism. However, fast changing channel conditions make achieving repeatability very difficult if not impossible. In Chapter 3, we further study methodologies for conducting empirical evaluations due to its popularity and being the gold standard for realism. In addition, we design and evaluate a technique for achieving repeatability in some scenarios, despite highly variable channel conditions.

As opposed to the experimental approach, the simulation and emulation methodologies are fundamentally repeatable, since they can use the same channel model repeatedly to evaluate different alternatives. The negative aspect of these methodologies is the lack of realism of the channel models used in simulation or emulation. To combine the strength of the experimental and simulation (or emulation) approaches, the trace-based evaluation methodology has been designed. In this approach, realistic traces are collected from actual experiments, while repeatability is achieved by using the same identical trace for the evaluation of multiple alternatives. In Chapters 4 and 5, we propose and evaluate two trace-based frameworks for the performance evaluation of 802.11g and 802.11n networks, respectively.

Finally, we review studies that characterize 802.11n networks from different perspectives. Although all of these studies provide valuable insights about the performance of 802.11 networks, in Chapter 6, we characterize these networks from the view of relationships between transmission rates, which has not been done in previous work. Our characterization results can be utilized to improve the performance of 802.11 optimization algorithms, such as rate adaptation and frame aggregation algorithms.

Chapter 3

Empirical Evaluation of 802.11 Networks Using the Randomized Multiple Interleaved Trials (RMIT) Methodology

3.1 Introduction

An important aspect of any type of research is being able to fairly evaluate, compare and draw conclusions regarding the relative merits of multiple competing systems or techniques. In this work, we refer to *alternatives* as solutions or systems that depend on 802.11 networks. Some examples include comparing different versions of TCP, video streaming techniques, 802.11 rate adaptation algorithms, or frame aggregation algorithms.

In 802.11 networks, performing fair comparisons of different alternatives can be extremely challenging because channel conditions can vary significantly over time. As a result, analytic models, simulation, and channel emulation are enticing because they can be used to ensure that each alternative being compared is exposed to precisely the same channel conditions. However, as discussed in Section 2.2.2, wireless channel models usually lack realism.

Therefore, empirical evaluations are highly desirable because they can provide a level of realism and accuracy that is difficult to achieve otherwise. However, empirical evaluations are not a panacea. Because channel conditions can vary over time, conducting repeatable experiments can be very challenging. One of the goals of this chapter is to understand the possibilities and limitations of empirical measurements when using popular 802.11n MIMO networks. We study the efficacy of different methodologies for comparing multiple competing alternatives. We examine both tightly controlled, presumably repeatable channel conditions, and uncontrolled environments with highly variable channel conditions. Our intention is not to discredit past work that use these methodologies, since they may be effective given stable channel conditions. Instead, our goal is to understand these methodologies and their pitfalls.

In this chapter, we evaluate the efficacy of a methodology that we call (randomized) multiple interleaved trials. This technique has been used in other disciplines [76], but to the best of our knowledge, we are the first to directly study this methodology in the context of WiFi performance analysis.

3.2 Experimental Setup

We have created a small testbed for conducting experiments in a cubicle-based office space in a university campus building. In addition to the experiments conducted in this chapter, we utilize this testbed in Chapters 5 and 6. Our testbed consists of desktop systems containing TP-Link TL-WDN4800 dual-band wireless N PCIe adapters. These cards use the Atheros AR9380 chipset, contain three antennas, and support three streams (i.e., a 3x3:3 MIMO configuration). In stationary experiments, we use two desktops, 4 meters apart, with no line of sight. For mobile experiments, we use a laptop configured to use a TP-Link TL-WDN4200 dual-band wireless N USB adapter. This adapter contains an Ralink RT3573 chipset and also supports a 3x3:3 configuration. To maximize repeatability, for some experiments, we use a small electric train on which a laptop is placed. The train is operated at walking speeds.

We conduct experiments using both 2.4 and 5 GHz 802.11n networks. Unless otherwise specified, we choose channels that are not used by any other access points or devices. In our 5 GHz experiments, we enable the optional 40 MHz channel width to increase the available bandwidth. We ensure that there is at least 40 MHz of separation between the channel used by our network and those used by other networks. This helps to avoid channel leakage from adjacent channels, which can cause performance degradation [36]. In our 2.4 GHz experiments, we do not use 40 MHz channel widths in order to limit external interference in this spectrum. The long guard interval and the optional short guard interval features are enabled in all of our experiments. To ensure that unknown, or unwanted, interference from other devices is avoided, we continuously monitor all of our experiments using a spectrum analyzer (AirMagnet Spectrum XT [21]). This analyzer is able to detect and classify both

WiFi and non-WiFi interference.

On the desktop machines, as well as the laptop, we use Ubuntu 12.04 with Linux kernel version 3.13.0. The ath9k (Atheros) and rt2800usb (Ralink) device drivers are provided by the backports-3.14-1 package. We use iperf [54] to generate UDP traffic between the sender and receiver at as high a rate as possible, to fully utilize the network infrastructure. The sending device in all of our experiments uses the *Minstrel-HT* 802.11 rate adaptation algorithm, which is the default rate adaptation algorithm used by the Linux Ath9K driver. Although we could have used `tcpdump` to record much of the information reported in this study, we use detailed information obtained directly from the ath9k driver. The ath9k driver is modified to report the fate of each packet along with the transmission time and transmission configurations. MAC-layer frame aggregation is enabled to increase the efficiency of the 802.11n MAC layer.

3.3 Experimental Methodologies

One of our goals in this thesis is to better understand the efficacy of existing methodologies that have been used to empirically measure and compare the performance of 802.11 networks. We use the term *trial* to refer to one measurement, typically obtained by running a benchmark or micro-benchmark for some period of time (the length of the trial). An *experiment* can be comprised of multiple trials executing the same benchmark, where the results of the experiment are reported over the multiple trials (e.g., the average of the measurements obtained over the trials).

Because it is well known that channel conditions can vary over time, we are interested in understanding repeatability and degree of variability across multiple experiments. Clearly, if the goal of empirical research is to compare multiple alternatives in order to draw conclusions about which is the best choice, it is important that the conditions under which the different alternatives are tested do not change significantly, in order for the comparison to be fair.

The approaches that we examine in this work are:

- **Single Trial Experiments**: In this approach, an experiment consists of only a single trial. This is a surprisingly common approach used in existing work [85, 89, 101]. In most cases, multiple wireless environmental setups might be considered (e.g., mobile, stationary, with and without hidden terminals), however, comparisons are made and conclusions are drawn using only a single trial.

- **Multiple Consecutive Trials**: This approach recognizes that possible changes in channel conditions can lead to variability. As a result, multiple trials are used. All trials for the first alternative are run, followed by the second alternative and each of the remaining alternatives. Finally, the means and confidence intervals of all alternatives are compared.
- **Multiple Interleaved Trials**: This approach requires interleaving each of the alternatives being studied. One trial is conducted using the first alternative, followed as soon as possible by one trial with the second, and so on until each alternative has been run once. When one trial has been conducted using each alternative we say that one *round* has been completed. Rounds are repeated until the appropriate number of trials has been conducted to complete an experiment. If channel conditions are affected at regular intervals, and the intervening period coincides with the length of each trial, it is possible that some alternatives are affected more than others. Therefore, we recommend a random reordering of alternatives for each round which we call Randomized Multiple Interleaved Trials (RMIT). In essence, a randomized block design [82] is constructed where the blocks are intervals of time (rounds) and within each block all alternatives are tested, with a new random ordering of alternatives being generated for each block. In this chapter, to make this methodology easier to describe and understand and because we did not find it necessary to randomize trials, we use the same ordering for each round.

Next we evaluate the efficacy of these methodologies for fair and repeatable comparisons of multiple alternatives using throughput and RSSI (helpful for understanding channel conditions). When multiple trials are used, we include 95% confidence intervals computed using the Student's t-distribution.

3.4 Evaluating Repeatability

The results in Sections 3.4.1 through 3.4.3 are obtained by collecting data while one stationary sending device is communicating at as high a rate as possible to a single stationary receiver for 24 hours. Nothing changes over that time except for possibly the channel conditions. We then divide the results obtained over that entire time period into chunks of time and compare results over those time periods as though they are "different" alternatives. However, because in reality the same alternative is used in all experiments, the results obtained across all experiments should ideally be identical, or close enough so as to be statistically similar. That is, the results should be repeatable. Results that are not similar indicate that there is a flaw with the methodology.

3.4.1 Single Trial Experiments

Often when researchers wish to compare two or more alternatives, a single trial of each alternative is used. For example, running alternative A for 60 seconds, followed by alternative B for 60 seconds, and using these results to compare performance differences between A and B.

We collect throughput and RSSI measurements over 24 hours and then divide the data into 60-second experiments and compare all consecutive sets of two experiments. Figure 3.1 shows measured throughput (the top line) and RSSI (the bottom line) for each of these experiments. The x-axis represents time from midnight of one day until midnight 24 hours later. In this and many subsequent graphs, the left y-axis shows throughput and the right y-axis shows RSSI. In practice, someone using this methodology has only one data point for each alternative, rather than the hundreds presented here, hence our focus on only consecutive data points. We include 24 hours worth of data to see how this methodology works at different times of the day (i.e., with different degrees of variability).

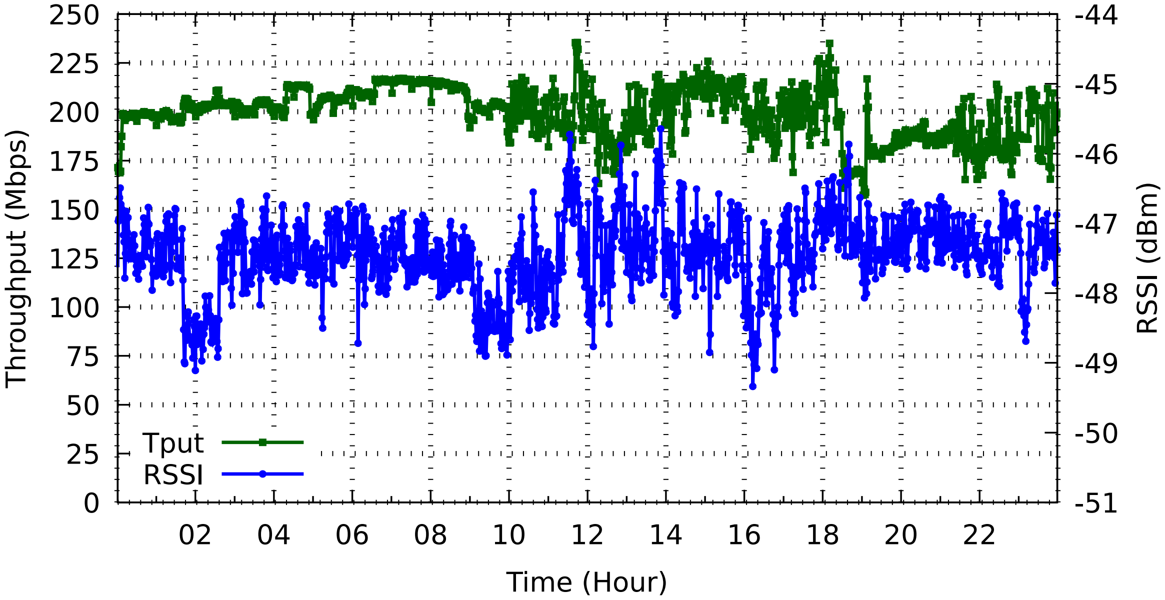


Figure 3.1: Consecutive experiment variability

For this methodology to be sound, each set of consecutive data points needs to be within some tolerance level based on the desired granularity at which differences in alternatives are to be compared. Figure 3.2 highlights (by zooming in on) two subsets of the 24 hours. The purpose is to provide a more detailed view of the results at the level of individual experiments and to examine the differences between these two periods of time.

The period of time from 04:00 - 06:15 uses the top x-axis, while the time from 17:00 - 19:15 is plotted using the bottom x-axis. These times were chosen because they include periods of relative stability (04:00-06:15) and variability (17:00-19:15). These subsets could be thought of as roughly corresponding to the middle of the night and some working hours (for the graduate students using this lab and these offices).

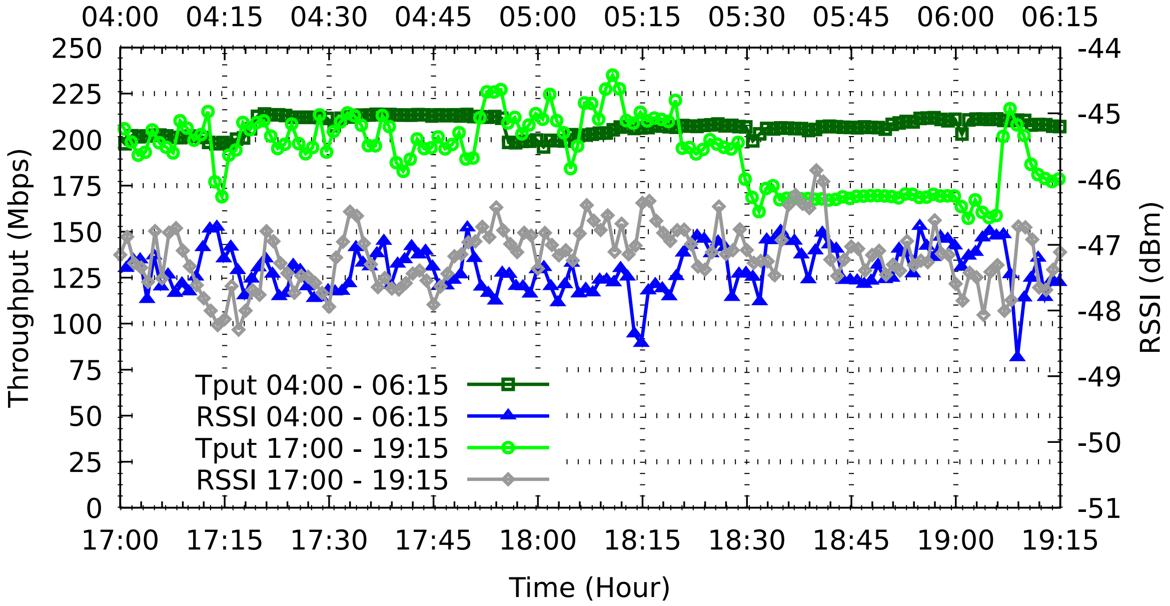


Figure 3.2: Consecutive experiment variability (04:00 to 06:15 and 17:00 to 19:15)

In Figure 3.2, throughput is roughly 160 Mbps at 19:06 and then jumps to 200 Mbps for the trial's immediate successor, 19:07. This is an increase of 25% in two consecutive trials whose start times differ by only 60 seconds. From this level of variation, it is evident that a single trial experiment could be misleading if one alternative were to be measured at time 19:06 and another at 19:07. In this case, a difference of 25% might be incorrectly attributed to the change in alternatives, rather than the change in channel conditions. This problem may be exacerbated by adding more alternatives, as channel conditions may change for each of the alternatives. For example, if several alternatives were examined between 18:00 and 18:15, each would be subject to different channel conditions (as can be seen by the changes in RSSI) and obtain different throughputs. While the variation observed between

04:00 and 06:15 in Figure 3.2 is lower than that observed during 17:00-19:15, there exist two consecutive trials where a difference of around 10% is observed (at 04:55). This suggests that, for large enough differences between alternatives, the single trial technique may be fine during the middle of the night (or a period of low variability) but rarely during the day, making it an unreliable technique overall. The fluctuations of the throughput during this experiment are mainly due to the movement of people working in nearby cubicles. Additionally, the environments experienced during both time periods were interference-free and the period from 04:00 to 06:15 was entirely stationary (no movement of devices or people), leading to a level of stability that may not be representative of the environments in which devices are most often used.

RSSI measurements in Figure 3.2 are similarly variable. During the day (17:00-19:15), RSSI in consecutive experiments differs by up to 0.92 dBm, while at night RSSI differs by up to 1.26 dBm. Even though RSSI is not a complete predictor of performance, it is an indication of the variability of channel conditions. Additionally, fluctuation is expected to increase in environments with interference and higher levels of mobility. Judd et al. [60] observed high levels of variability in RSSI in a stationary environment with lots of mobility, such as a lobby or a hallway. However, we observed that fluctuations occur even in fairly stable environments. Interestingly, and also noted by others [48], stronger signals do not necessarily correlate with higher throughput.

Guideline: *In order to draw conclusions using single trial experiments, channel conditions must be extremely stable. As a result, conclusions from these experiments cannot be extended to other environments possessing greater variability (which may be considered more representative of environments in which devices are actually used). More importantly, even if channel conditions are believed to be stable, we do not know of any way*[1] *to guarantee, or verify, that channel conditions did not change from one trial to the next. Thus, our recommendation is to avoid using this methodology.*

3.4.2 Multiple Consecutive Trials

As we saw in the previous section, a considerable amount of variability can exist even in interference-free, stationary environments. A possible, and commonly used, solution is to run experiments multiple times and report performance metrics using a mean and some indication of variability across experiments, instead of using a single measure. In practice, when more than a single trial is collected, experimenters often conduct all of the trials for a

[1] Although RSSI is an indicator of channel conditions, it is oblivious to WiFi and non-WiFi interference.

given alternative, before proceeding to the next. This makes sense, since there is generally some setup time involved in switching between alternatives. For example, changing WiFi cards or a software configuration (e.g., unloading and loading a device driver in order to change the rate adaptation algorithm).

To evaluate this approach, we again divide the same data collected during the 24-hour period into 60-second trials but now combine consecutive trials together to constitute an experiment. While experiments may not be performed in such a continuous fashion, we use this technique since it allows us to fairly and easily utilize the same data when comparing competing methodologies. We compute the throughput and average RSSI for different numbers of trials. Recall that during the 24-hour period, the only thing changing is channel conditions and, because we ensure that there is no WiFi or non-WiFi interference, channel conditions *should be* relatively stable (especially compared with more challenging scenarios that include interference). If the experiments are repeatable, we should see no statistical difference between the two consecutive alternatives A and B (i.e., if the 95% confidence intervals overlap, we consider the result to be repeatable). However, if a statistically significant difference is observed (the confidence intervals do not overlap), the measurement technique can potentially lead to erroneous conclusions. A conclusion may be drawn that performance differences are due to different alternatives, rather than changes in channel conditions. Even though, in this case, A and B are identical.

Figure 3.3 shows the results of applying the multiple consecutive trials technique to the 24-hour data that we used in the previous section. We focus on 4 hours of data (16:00 - 20:00) where higher channel variability was observed (again because we are interested in determining if experiments can be used to evaluate alternatives when channel conditions are variable). The three plots in this figure show the average throughput and RSSI with 95% confidence intervals for both alternatives when using 5, 10, and 15 trials (top, middle and bottom, respectively). We consider 5 trials to be small but include it because a number of previous studies [37, 92] examine 5 or even fewer trials. As in the previous section, we only compare two consecutive experiments. However, because several trials are used, we test for overlapping confidence intervals to assess repeatability.

In Figure 3.3, when examining throughput, we see that for 5 trials there are two consecutive non-overlapping confidence intervals for experiments at 16:50, 17:50 and 19:05. Increasing the number of trials to 10, we see non-overlapping confidence intervals at 16:00, 16:50, 17:50, 18:20 and 18:35. This may be due to the fact that more trials increases the amount of time between the start of consecutive experiments and, therefore, increases the likelihood that the environment changes between these two experiments. This can lead to problems if experiments happen to be run during these periods of variability. While it may be tempting to assume that an astute experimenter would recognize these periods of variability, in reality, it is unlikely that they have 24 hours of experiments comparing the two or more alternatives; instead, they are likely to have only one data point for each alternative. Unfortunately, we are not aware of any way to determine, before or after the experiments have been run, that the observed differences are due to different alternatives, rather than changes in channel conditions. Recall that in our case we know that all alternatives being examined are identical. These issues become more pronounced when comparing more than two alternatives, since the start times between compared alternatives are further apart. For example, consider 5 alternatives with 15 trials starting at time 18:12 (in Figure 3.3). Several of these results have non-overlapping confidence intervals. In the next section, we explain how a simple modification to this technique can overcome these issues.

Guideline: *In a controlled and interference-free environment, we were not able to reliably repeat experiments using the technique of multiple consecutive trials. Moreover, the confidence intervals obtained using this technique may provide a false sense of rigor and validity when comparing different alternatives. Because channel conditions may change over time and because we do not know of any way to guarantee or verify that they have not changed, our recommendation is to avoid using this methodology.*

3.4.3 Multiple Interleaved Trials

In the hope of addressing the shortcomings of the multiple consecutive trials technique, we now evaluate the technique we call multiple interleaved trials. To our knowledge, this method has not been explicitly used in previous studies evaluating performance when using 802.11 networks. We are the first to directly study this methodology in the context of WiFi performance analysis.

This perhaps obvious, but important, approach provides the advantage that trials are more closely situated in time, provided each trial does not run for too long (the length of trials is discussed in Section 3.4.3). The intuition is that by interleaving alternatives, they will be exposed to channel conditions that are more similar than when using multiple consecutive trials. This is particularly important as the number of alternatives being compared grows. For example, if two alternatives are compared using multiple consecutive trials, if a microwave-oven (or any device that generates interference) is used during A's trials but not during B's trials, A will likely experience unfairly low throughput. However, with multiple interleaved trials, if the trials are short relative to the length of time the microwave is on, both alternatives are likely to be subjected to the interference. If the trials are sufficiently long, such that one alternative is affected significantly more than the

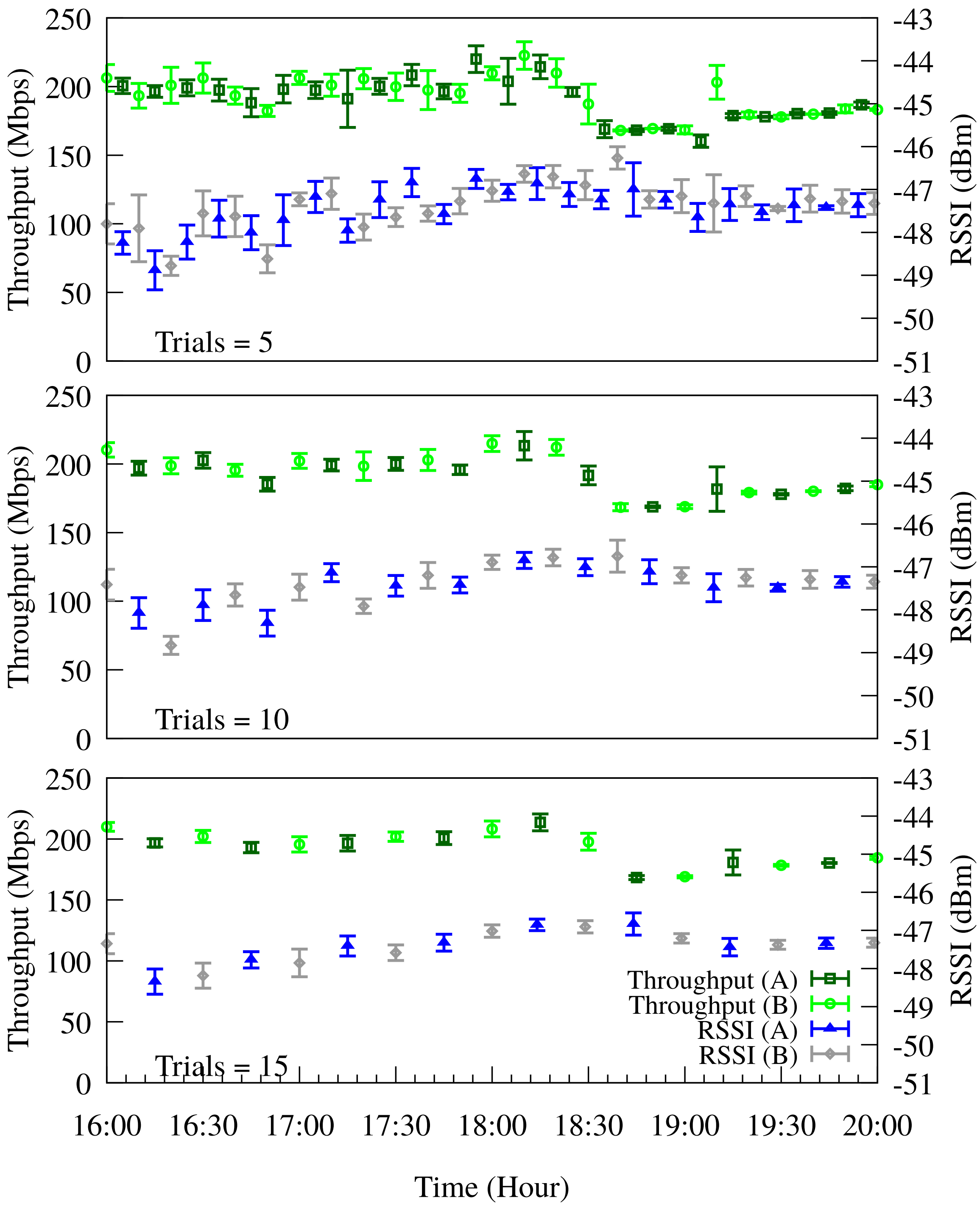


Figure 3.3: Multiple consecutive trials (2 alternatives)

other, it should show up as wider confidence intervals for that alternative.

Stationary: Two Alternatives

We begin by using the same 24-hour data used in the previous evaluations in Sections 3.4.1 and 3.4.2. This ensures that the same conditions are experienced using all three methodologies. We process the data by using the first 60 seconds of data for the first trial of the first alternative (A) and the next 60 seconds for the first trial of the second alternative (B). This completes the first round or trial. This process is repeated until N trials have been obtained. We then compute the mean throughput, RSSI, and confidence intervals from the N trials and plot that data. This process is repeated until all 24 hours of data have been used. We thus produce multiple pairs of data points comparing alternatives A and B over different times of the day. Since A and B are identical, we expect to see no statistical difference in any of the pairs of data points comparing the two alternatives.

Figure 3.4 shows these results for 5, 10, and 15 trials from a period of the day with the high variability (16:00 - 20:00). In all cases (including times not shown), confidence intervals for A and B overlap. This suggests that experiments using multiple interleaved trials are repeatable for two alternatives for the given data. Increasing the number of trials tightens the confidence intervals. Recall that with multiple consecutive trials, increasing the number of trials decreased the size of confidence intervals, however the results were not repeatable.

Stationary: Five Alternatives

To this point we have focused on comparing two alternatives. However, it is often desirable to compare multiple competing alternatives. Comparing more alternatives makes the evaluation more challenging because each round takes longer to complete and trials from the same alternative are farther apart in time. Consequently, trials may experience more disparate channel conditions. The different trials for each alternative can be separated in time either because of more alternatives, as is the case in this section, or perhaps because of setup time required to switch between alternatives. In Section 3.4.3, we also consider the interval between trials by examining trials of different lengths.

We again use the same 24-hour data and multiple interleaved trials to compare five alternatives. Data is processed in the same way as for two alternatives, except there are now five alternatives (so each round lasts 5 minutes). The results are shown in Figure 3.5. Each group of points, illustrated by different colors, represents the comparison of 5 alternatives.

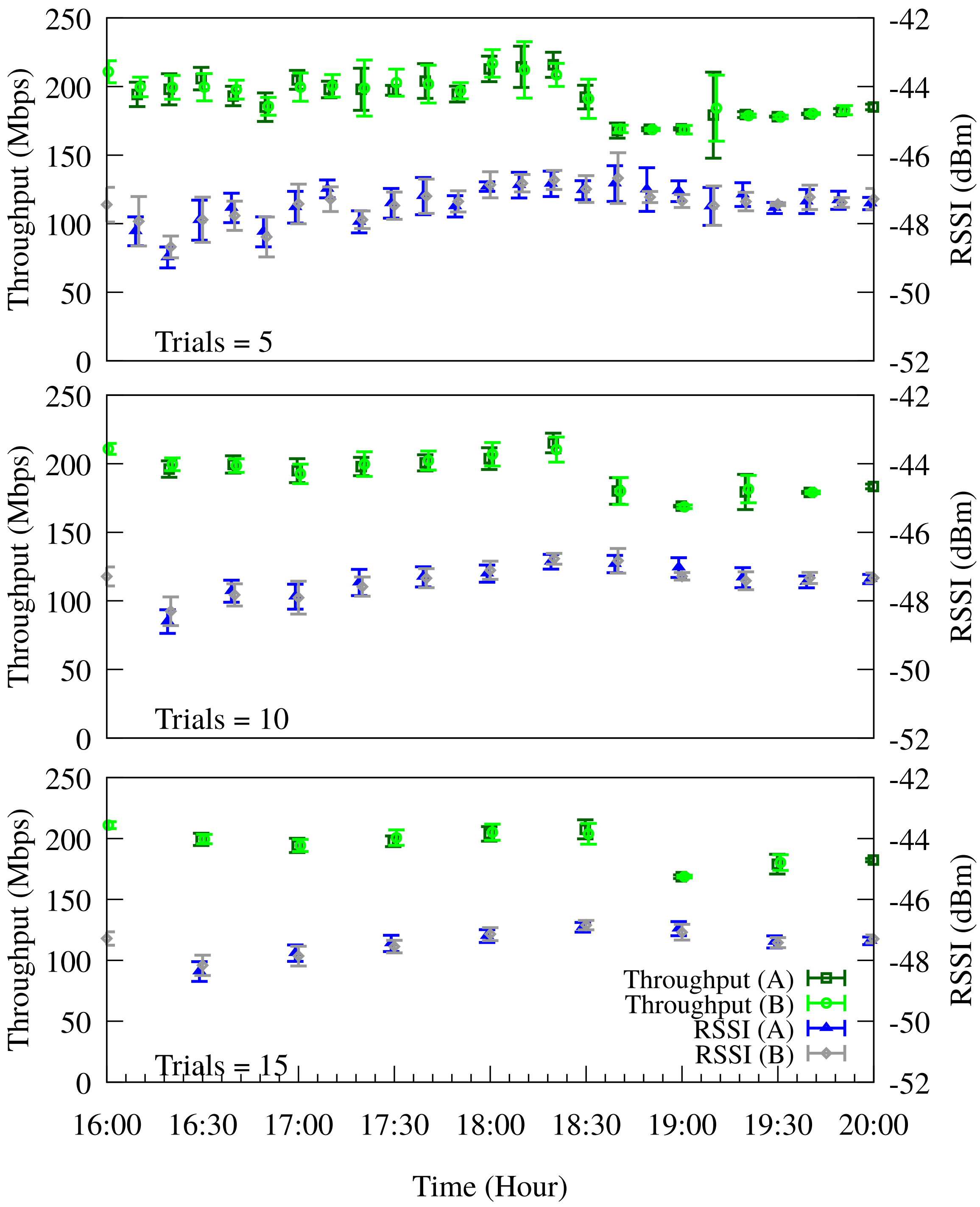


Figure 3.4: Multiple interleaved trials (2 alternatives)

This graph shows that the 95% confidence intervals for the mean throughput and RSSI for all five alternatives for 5, 10 and 15 trials overlap. The overlapping confidence intervals indicate that each alternative was subjected to roughly similar channel conditions, and demonstrates that for this environment, results are repeatable for five alternatives using multiple interleaved trials. For these experiments, increasing the number of trials decreases the size of the confidence intervals, allowing the study of finer differences between multiple alternatives. However, we expect that diminishing returns would come into effect as the number of trials grows.

Guideline: *The technique of using multiple interleaved trials increases the probability that the compared alternatives will be subject to similar channel conditions. This leads to repeatable experiments and should permit fair comparisons of different alternatives. Because multiple trials are being conducted, the size of the confidence intervals is a reliable indicator of the variability in channel conditions across different trials, and also provides a gauge as to whether or not differences in the observed means are statistically significant.*

Mobile: No Interference

Previous experiments were conducted using stationary devices with no interference. Since mobility increases the variability of wireless channels, we evaluate the efficacy of multiple interleaved trials in mobile environments where a WiFi receiver is moved by a motorized toy train or by a person.

We set up a toy train track in a roughly oval shape that allows for a fairly wide range of received signal strengths due to varying degrees of path obstruction and distances from the Access Point (AP). At the starting point, there is very little blocking the line of sight between the laptop and the AP. At the farthest point, there are several cubicle walls consisting of metal, wood and fabric in the way. Although the distance between the closest and furthest points on the track are roughly only 5 meters, path loss due to cubicle walls affect the 5 GHz signal propagation sufficiently to achieve a 15 dBm range of signal strength variation.

We place our laptop on the train and aim to maximize repeatability by starting the train from the same position in each trial. Each trial consists of two laps around the track at approximately walking speed. We find that throughput experiments are relatively repeatable in this environment. Figure 3.6 (top) shows the throughput (bottom cluster) and RSSI (top cluster) measured for each second. Each trial lasts 60 seconds and we plot the throughput obtained during each second of all 20 trials. We add wider dark lines

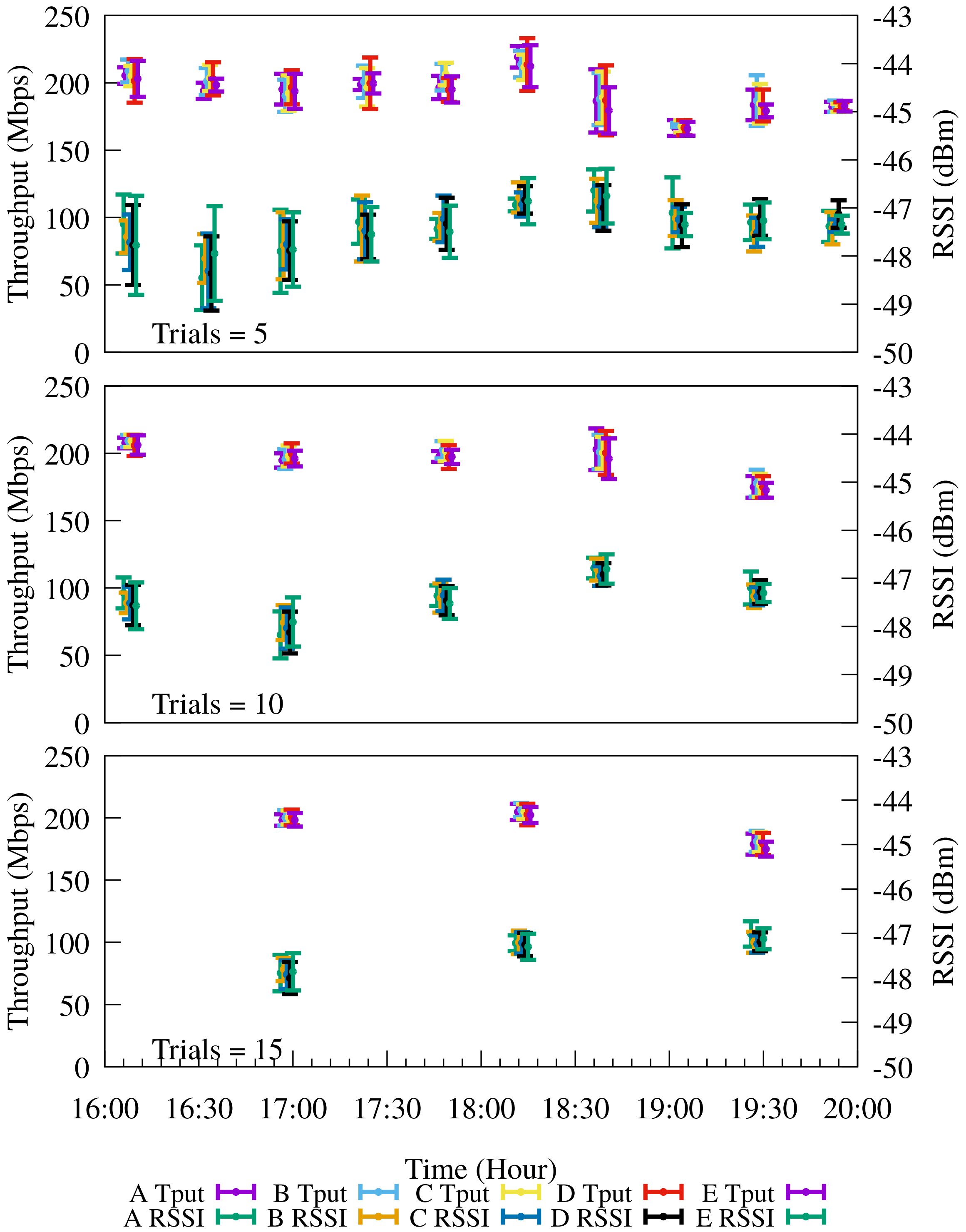


Figure 3.5: Multiple interleaved trials (5 alternatives)

indicating the maximum and minimum values obtained across all runs to better indicate the range of values obtained. As expected, each trial is different, however, all trials follow similar trends with throughput increasing near the AP (near the 0, 30, and 60 second marks) and decreasing as it gets further away.

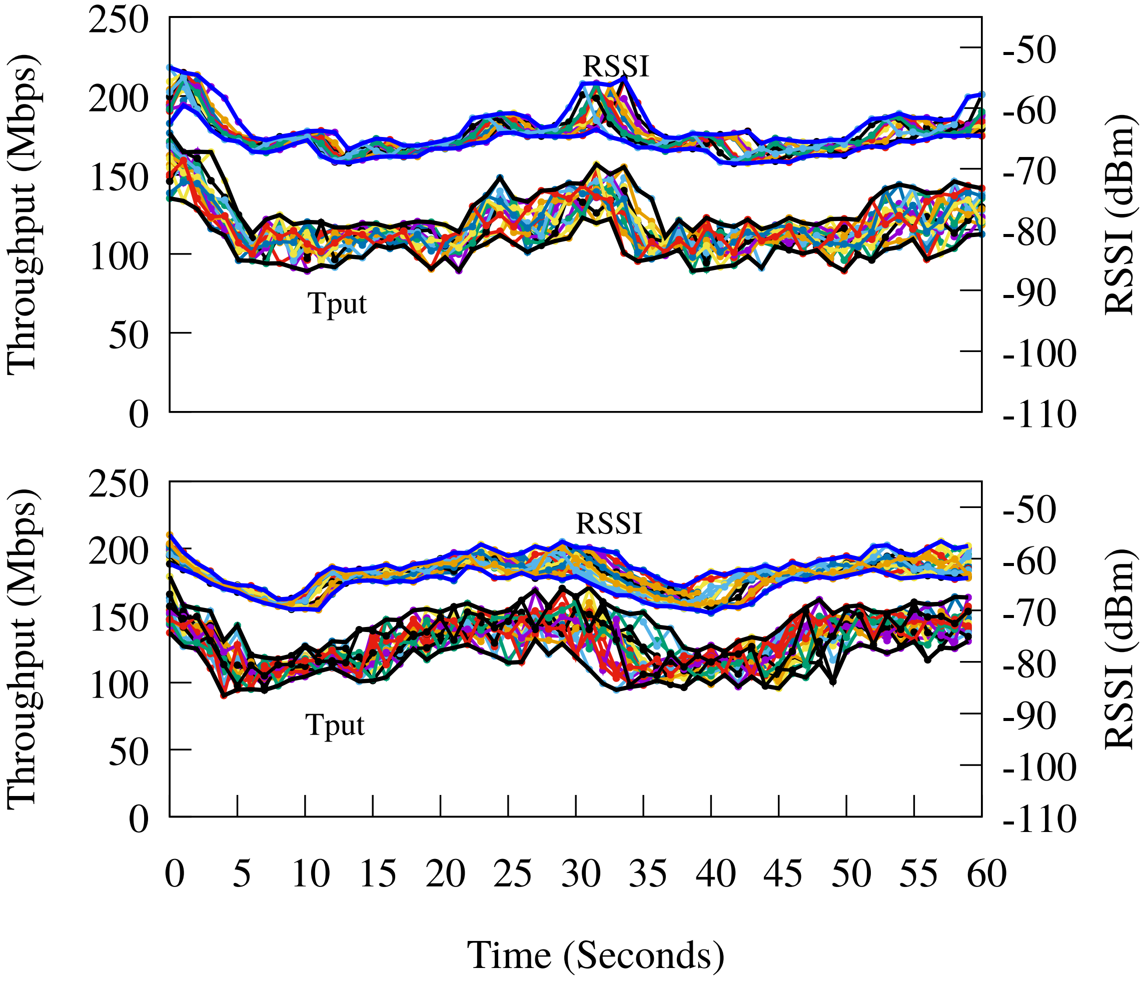


Figure 3.6: Mobile: toy train (top) and walking (bottom)

We now repeat the same experiments, with a person carrying the laptop. Markers on the floor are used to follow roughly the same path as that of the train. Holding the laptop at waist height, we measure throughput for 60 seconds; again making two passes around the track for each trial. Figure 3.6 (bottom) shows that the throughput (bottom cluster) and RSSI (top cluster) trends are fairly consistent across trials, although less consistent than those obtained with the train. This is expected since walking a path is naturally less accurate due to the difficulty of maintaining consistent speeds, positioning of the arms and body, and due to the body's impact on the received signals.

We should not (and do not) compare performance results obtained from the train and walking trials. The height of the laptop and the line of sight have changed between the two experiments. Instead, we use interleaved trials to analyze the collected data as though two alternatives are being studied in each of the toy train and walking experiments.

Figure 3.7 shows the mean throughput achieved using the multiple interleaved trials technique for different numbers of trials. The bars with 3 trials use data from the first 6 runs (two alternatives are interleaved), 5 trials - the first 10, and 10 trials - all 20 runs. These graphs are presented to examine differences in the variability and repeatability of results obtained during the train and walking experiments. Again, if the experiments are repeatable, we expect to see no statistically significant difference between these two alternatives (labeled A and B). While the confidence intervals are overlapping in all cases, indicating repeatability, the train experiments exhibit tighter confidence intervals than the walking experiments. This is an important finding since walking experiments are often used for mobile WiFi performance evaluation. If small differences between alternatives are to be measured, the use of a robot (or train) may help reduce variability and increase the likelihood of obtaining statistically significant differences. Note that results obtained with a train or robot may not be representative of situations where people are walking and carrying devices, due to elimination of the impact of human body on electromagnetic signals.

Guideline: *Human-controlled mobile experiments introduce significant variability, making it desirable to use automated tools such as an electric train or robot. The multiple interleaved trials technique may permit the use of experiments that involve people walking with mobile devices when automation is not feasible. However, it may be difficult to discern small differences between alternatives.*

Mobile: Person Walking with Interference

To further test the reliability of results obtained using multiple interleaved trials, we conduct a few experiments using a 2.4 GHz network where WiFi and non-WiFi interference are present. We first use a channel that overlaps with multiple other uncontrolled APs

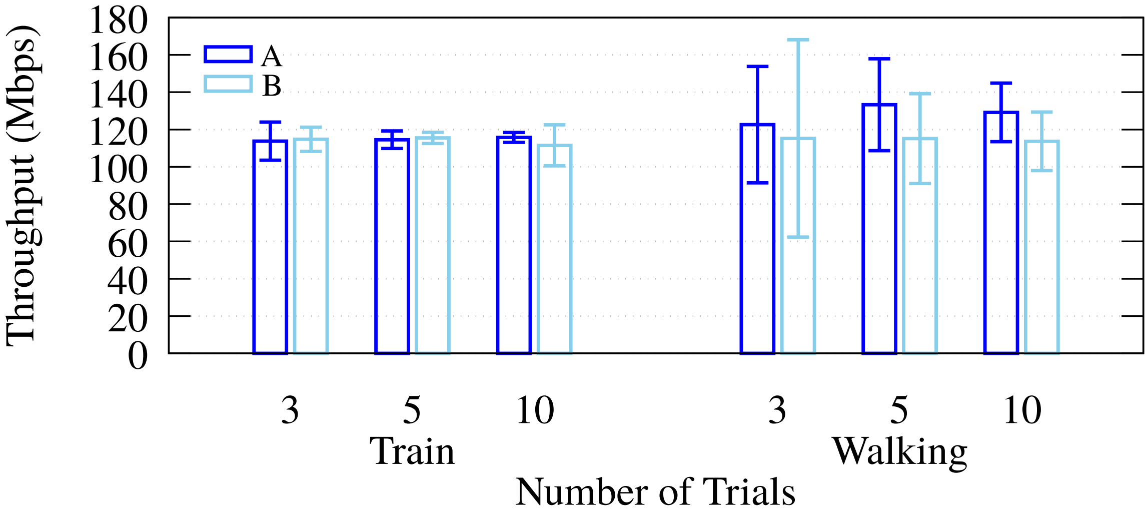


Figure 3.7: Mobile: no interference

(called the “Busy Channel”) and then a channel with neighboring, but non-overlapping, channels (called the “Unused Channel” because it is unused except for our experiments). We refrain from referring to this second channel as interference-free due to the potential for channel leakage effects described in Section 3.4.1 and [36]. Instead, these two experiments can be thought of as examining two different scenarios, one with significant amounts of WiFi interference and one with limited WiFi interference. Both channels are subject to non-WiFi interference. We follow the same procedures as the walking experiments, described in Section 3.4.3, using twenty 60-second trials. However, in this case we started much closer to the AP and walked to a point significantly farther from the AP. This was done in order to cover a much wider range of signal strengths.

We plot raw throughput and RSSI for each trial in Figure 3.8 and find that while there is significant variation across trials for both channels, the general trends are quite similar across the different trials. Figure 3.9 shows the average throughput with 95% confidence intervals. When considering two alternatives, we see that the confidence intervals overlap in all cases. This suggests that multiple interleaved trials can be used to conduct repeatable experiments, even in a challenging 2.4 GHz environment with significant interference. Note that the confidence intervals for these experiments are much smaller than the interference-free walking experiments shown in Figure 3.7. This counterintuitive observation is indicative of the variable and unpredictable nature of these types of experiments.

While it might be tempting to conclude that existing techniques could be used in this situation, we again emphasize that we know of no way to guarantee or verify that channel conditions do not change during the experiment in a way that creates unfair conditions for one or more alternatives.

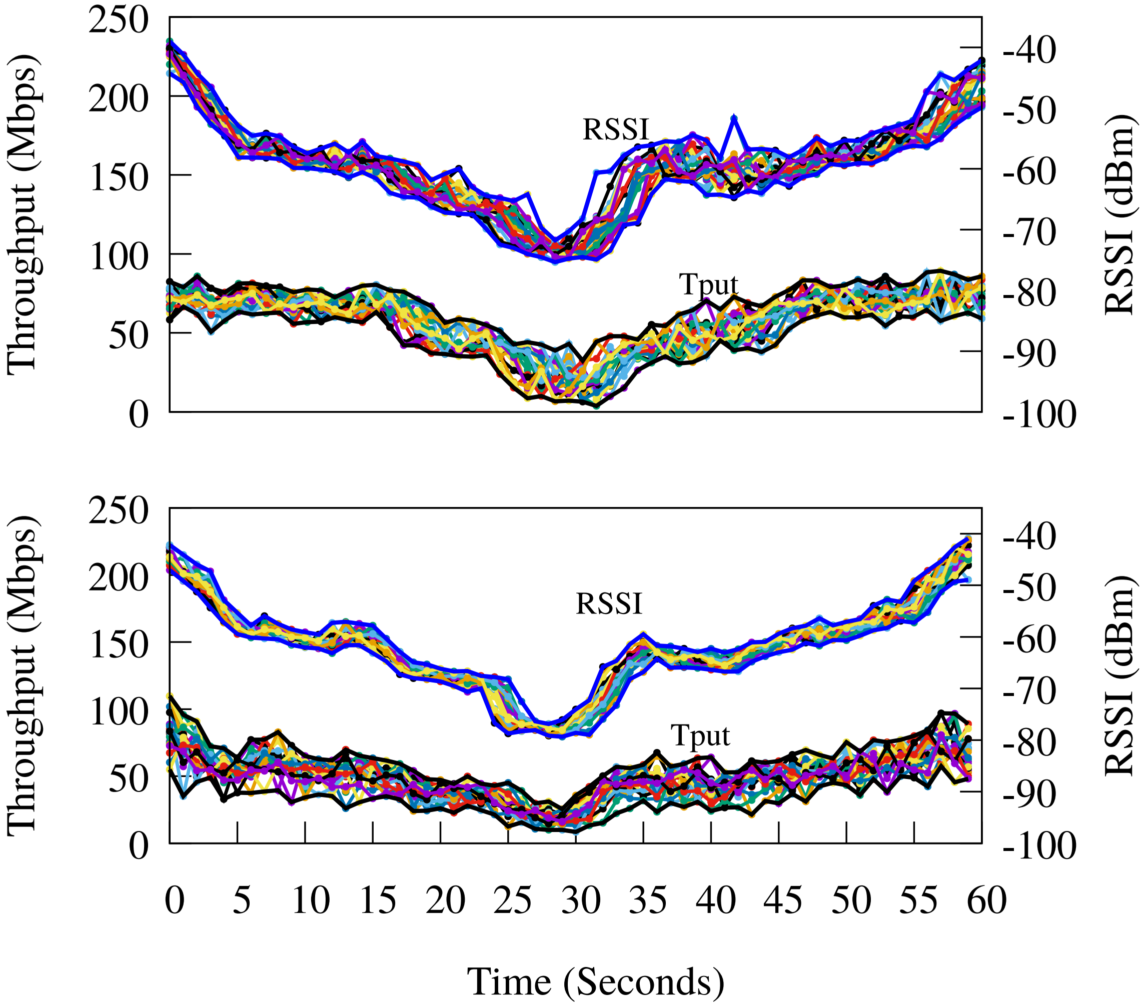


Figure 3.8: Mobile: unused (top) and busy (bottom) channels

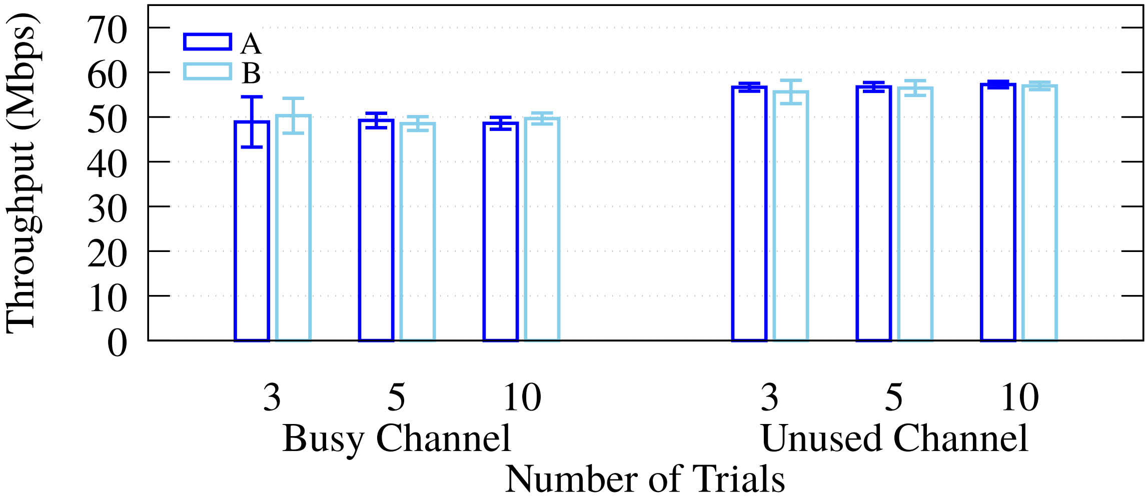


Figure 3.9: Mobile: 2.4 GHz, busy and unused channel

Trial Durations

In addition to ensuring that enough trials are run to obtain tight confidence intervals, an interesting consideration is the length of time required to conduct each trial. One issue is how long trials should be and another is whether or not trials of different lengths can be compared (e.g., when measuring data transfer time) While these are issues that we hope to investigate more deeply in future research, we now provide a brief examination. To study these issues we use the same 24-hour data used previously in Sections 3.4.1 and 3.4.2, and compare results obtained using trials of different lengths.

Figure 3.10 shows the results obtained by considering three alternatives (A, B and C) with trial lengths of 15, 60 and 240 seconds, respectively. In this case, each round is 315 seconds. The top, middle, and bottom graphs show, 5, 10, and 15 trials, respectively. To more easily compare the three alternatives, we have zoomed in on a range of time where variability was relatively high (in this case 12:00 - 20:00).

These results show that even when comparing alternatives that use different durations (15, 60, 240 seconds), all sets of experiments comparing these alternatives have overlapping confidence intervals. This demonstrates repeatability for this technique, for this set of data. We also note that with only 5 trials, the confidence intervals for alternative A (15

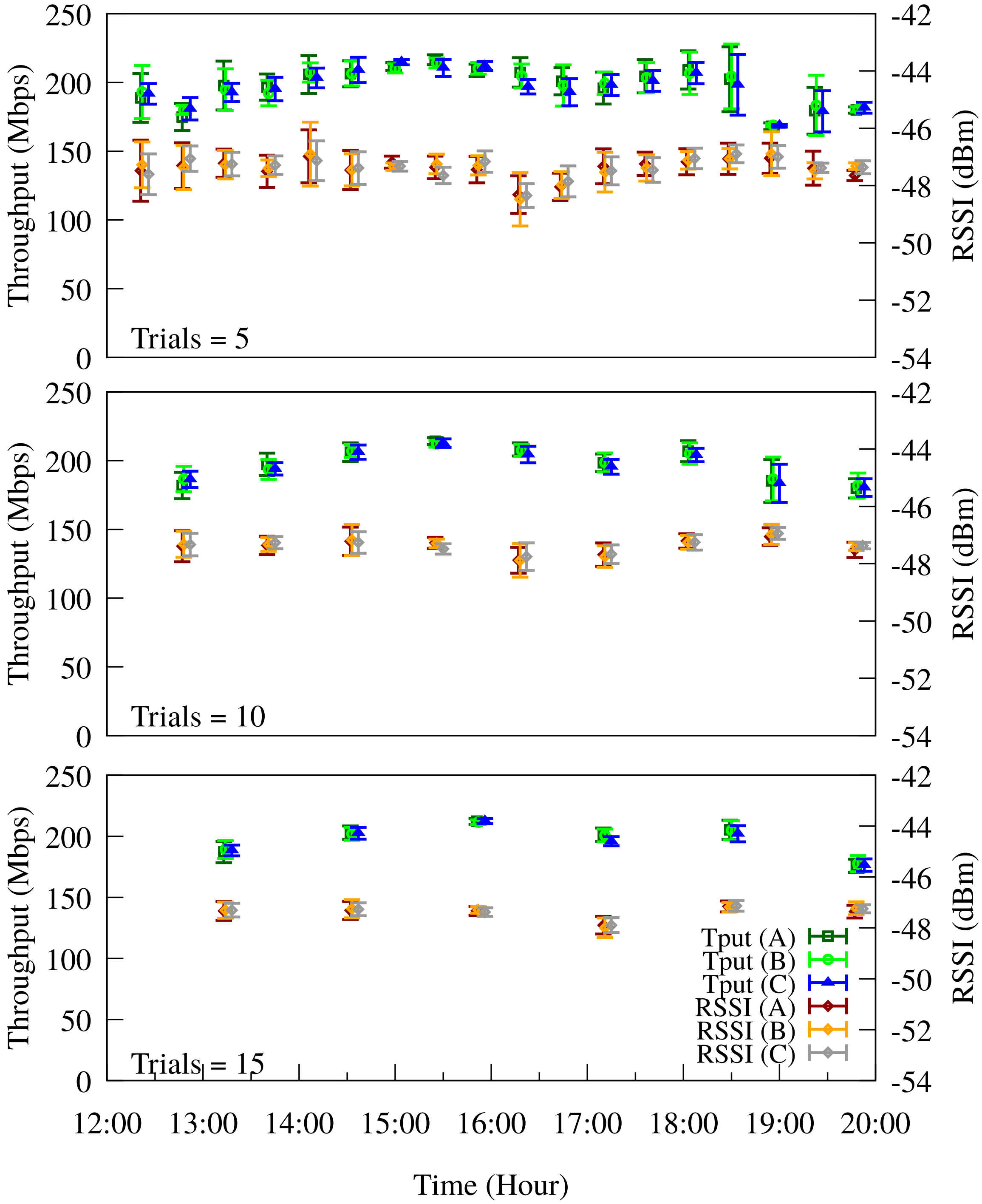


Figure 3.10: Different trial lengths 15 s (A), 60 s (B), 240 s (C)

seconds) may be slightly larger in some cases than those obtained for the alternatives with longer trial durations. However, as the number of trials increases, those differences seem to diminish. In Section 3.5, we use alternatives that each require different amounts of time to run in order to test if this methodology can distinguish between alternatives where differences are expected.

3.5 Distinguishing Differences

Thus far all experiments have compared two or more alternatives, each of the same configuration and setup, which allows us to establish repeatability. However, it is often desirable to perform experiments involving comparisons of different alternatives. Generally, we are interested in how a given system, configuration, or algorithm performs relative to other alternatives.

In order to study the granularity at which statistically significant differences can be determined between alternatives, we conduct experiments using the interleaved trials methodology. In this example, we vary the amount of the data transferred from a sender to a receiver, and measure the time required to complete the transfer. Relative to the baseline data size of 200 MB, we use transfer sizes of 100%, 110%, 120%, 130% and 140%, whose transfer times should ideally differ by exactly these factors (relative to the baseline case). We chose 5 alternatives in order to determine the degree to which the interleaved trials methodology can be used to distinguish between these different alternatives. While we perform 20 interleaved trials of each size, in both a noisy 2.4 GHz network and an interference-free 5 GHz network, we examine the data obtained after 5, 10, 15 and 20 trials. This is done in order to study the influence that the number of trials may have on the variability of the results and therefore the confidence intervals.

Figure 3.11 (top) shows the transfer times for the 2.4 GHz network for each of the 20 trials. As can be seen in this graph, the transfer times differ widely between trials. At 2.4 GHz, transfer times are noticeably longer near the beginning of our experiment. This reinforces the need to use the multiple interleaved trials technique; if consecutive trials were used then transfer times for data sizes measured early in the experiment could have appeared unfairly long. However, using the multiple interleaved trials technique, all alternatives (i.e., data sizes) are subjected to the unfavorable environment. Using the 5 GHz network, transfer times were fairly consistent for each data size, which is expected due to the lack of interference.

Figure 3.12 plots the average transfer times and 95% confidence intervals computed after 5, 10, 15 and 20 trials. As might be expected, when using the 2.4 GHz network with

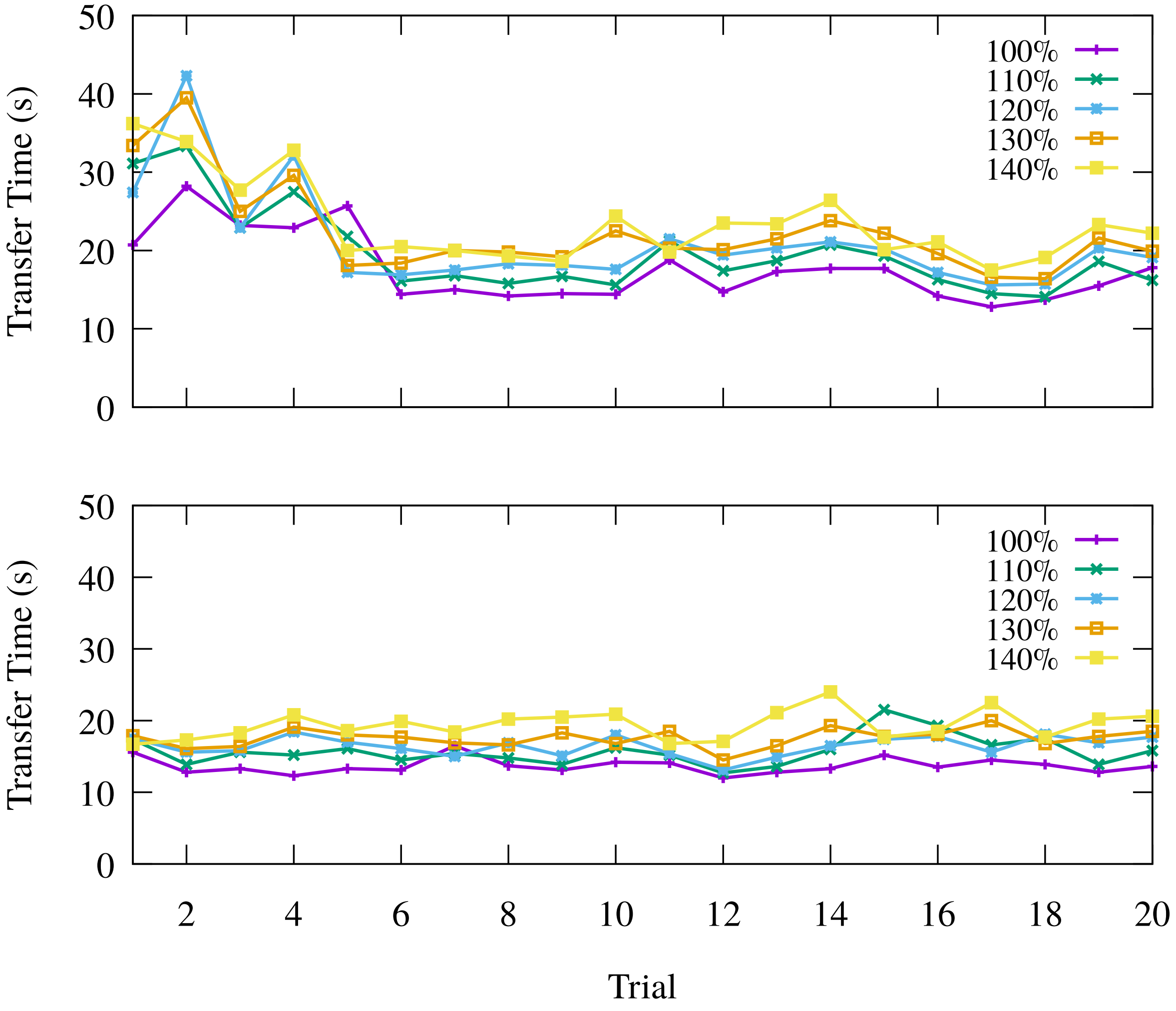


Figure 3.11: Raw transfer times: 2.4 GHz (top) and 5 GHz (bottom)

more dynamic channel conditions (e.g., due to WiFi and non-WiFi interference), small numbers of trials result in relatively large confidence intervals. This may make it difficult to draw conclusions about differences in performance between alternatives unless those differences are substantial.

Table 3.1 shows the means and confidence intervals for the 20-trial experiments in Figure 3.12. In this case, confidence intervals are noticeably wider when using the 2.4 GHz

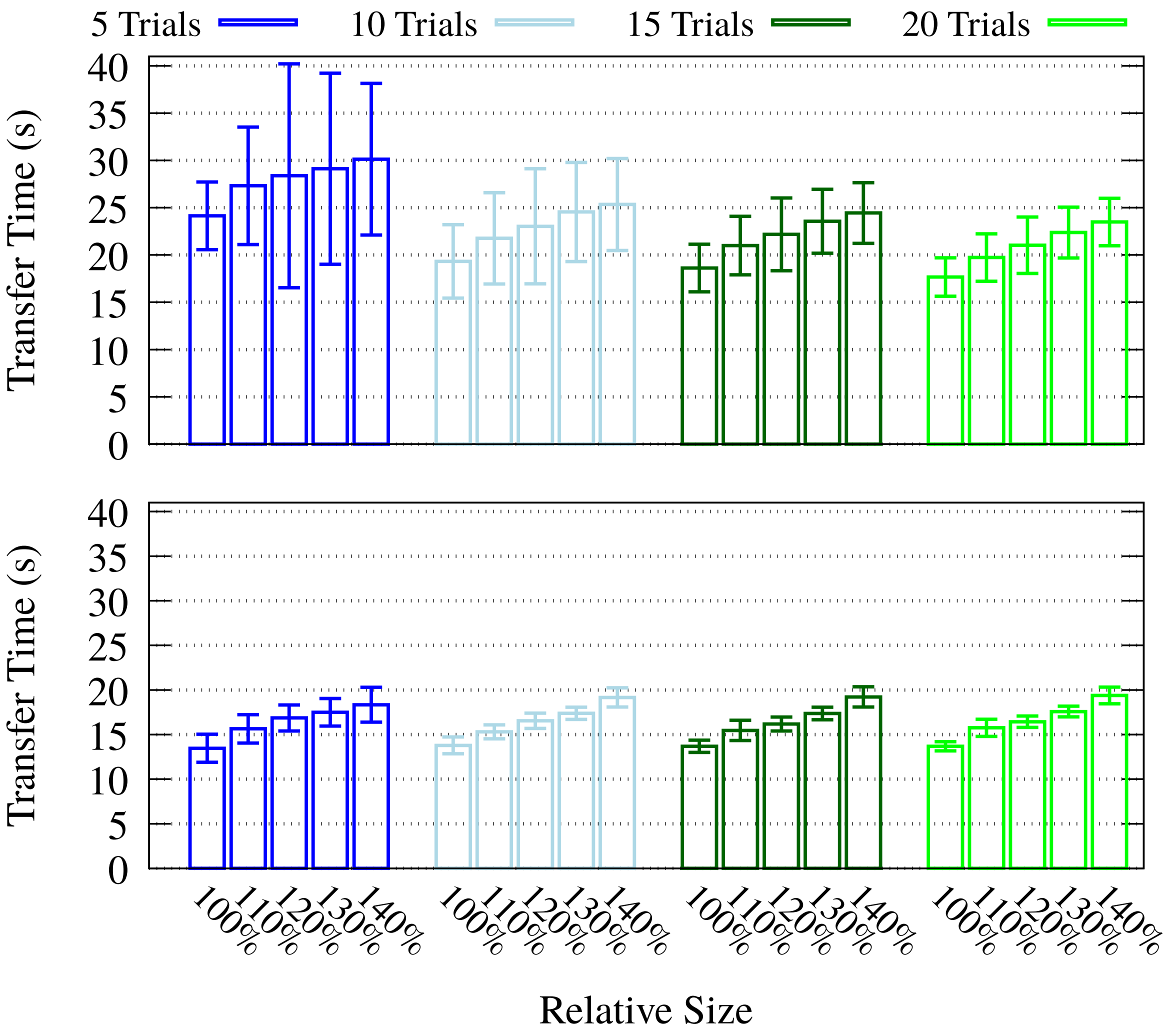


Figure 3.12: Average transfer times: 2.4 GHz (top) and 5 GHz (bottom)

network, compared to the 5 GHz network. As a result, for the 2.4 GHz experiment, only differences as large as 40% are considered statistically significant. However, for the 5 GHz experiment, differences as small as 10% can be distinguished. Generally, a rule of thumb cannot be established regarding the granularity of differences that can be detected, since it depends on the level of variability in the experiment. Note that this is not a limitation of the technique but rather the reflection of natural variation of experimental results. The multiple interleaved trials technique naturally and inherently captures variability across trials.

	2.4 GHz		5 GHz	
Relative size	Mean	Confidence interval	Mean	Confidence interval
100%	17.67	15.64 – 19.71	13.68	13.16 – 14.19
110%	19.73	17.22 – 22.24	15.76	14.80 – 16.72
120%	21.04	18.06 – 24.02	16.44	15.80 – 17.08
130%	22.38	19.69 – 25.07	17.58	16.98 – 18.18
140%	23.49	20.98 – 25.99	19.39	18.46 – 20.32

Table 3.1: Transfer times: 20 trials

Guideline: *When using the multiple interleaved trials methodology in some environments, a small number of trials may not be sufficient to allow one to determine statistically significant differences between multiple alternatives. It is critical to conduct multiple trials and to compute and report confidence intervals.*

3.6 Discussion

It is important to understand that the results we have obtained are unique to our environment and the times at which the experiments were run. These results should **not** be used to draw conclusions about the granularity of differences that can be distinguished in other environments. We acknowledge the difficulties involved in conducting multiple interleaved trials. It may be time consuming to manually make configuration changes between each trial. This stresses the benefits of automation or scripting of experiments.

Guideline: *In theory, channel conditions could be affected in a specific pattern, periodically. In such a case, it could be possible that some alternatives are affected more than others. For example, imagine that a WiFi device (interferer) downloads some data every 5 minutes. If a multiple interleaved trials experiment is designed with one-minute trials and 5 alternatives, it could be synchronized with the pattern of interference. In this case, the experiment might be biased towards one or more alternatives. Since it is not possible to tell before or after an experiment if such recurring interferences exist, we recommend using the Randomized Multiple Interleaved Trials (RMIT) methodology in which trials are randomly reordered to avoid the described problem.*

In an ideal world, all trials would be conducted at once to achieve perfect temporal correlation and identical channel conditions between trials. One approach, inspired by Judd *et al.* [58], would be to use splitters/combiners to capture identical signals at multiple receivers. Unfortunately, this has the significant drawback of reducing the received signal strength by $1/N$, where N is the number of splitters. Therefore, this approach may be of limited practicality and we leave its evaluation for future work.

3.7 Use in Other Contexts

We have found that the idea of using multiple interleaved trials predates our use in at least one other field. In an n-of-1 clinical trial, one individual is observed as different treatments (perhaps including a placebo) are administered over time with data being collected and analyzed to determine the best treatment. This technique has applications in individualized medicine [76].

We believe that multiple interleaved trials are well-suited to, but to our knowledge have not been used, when evaluating the performance of computer systems or networks when the system under test *cannot be guaranteed* to be subject to identical conditions across experiments. Some examples might include: experiments on other types of wireless networks, "live" systems like wide area networks or Web services, or when using cloud computing environments where CPU, memory, and network performance can all vary significantly [99] over time.

3.7.1 Cloud Computing Environments

Previous work has shown that benchmark and application performance in public cloud computing environments can be highly variable. Utilizing Amazon EC2 traces that include measurements affected by CPU, memory, disk, and network performance, we study commonly used methodologies for comparing performance measurements in cloud computing environments. The results show considerable flaws in these methodologies that may lead to incorrect conclusions [14]. For instance, these methodologies falsely report that the performance of two identical systems differ by 38% using a confidence level of 95%. We then study the efficacy of the Randomized Multiple Interleaved Trials (RMIT) methodology using the same traces. We demonstrate that RMIT could be used to conduct repeatable experiments that enable fair comparisons in this cloud computing environment despite the fact that changing conditions beyond the user's control make comparing competing alternatives highly challenging. Note that we did not find it necessary to randomize the trials when using the MIT methodology for our WiFi data sets. However, MIT failed to produce repeatable results in a few instances in our cloud data set and therefore using RMIT was actually required.

The results of this study have been published in the proceedings of the 8th ACM/SPEC International Conference on Performance Engineering (ICPE 2017) [14]. This study is not presented here due to being out of the scope of this thesis.

3.8 Chapter Summary

Experiments have long been used and will remain an important tool for evaluating the performance of 802.11 networks. Because channel conditions vary over time, the difficulty lies in obtaining repeatable and fair comparisons when evaluating competing alternatives. One approach used in the literature is to control for, and essentially eliminate, variability in channel conditions. However, such evaluations are not representative of more variable channel conditions, under which devices are likely to be used. More importantly, variations usually exist even in controlled environments.

In this chapter, we study the degree to which 802.11n MIMO network experiments can be repeated, with an emphasis on methodologies for comparing competing systems, configurations, or algorithms. We find that using existing methodologies, we were not able to reliably obtain repeatable results, even under controlled conditions where there is no WiFi or non-WiFi interference. As a result, we propose the use of the multiple interleaved trials methodology, which permits conducting experiments under variable channel conditions. The keys with this approach are: 1) it ensures that all alternatives are subject to similar channel conditions, and is therefore fair; and 2) it can be used to easily and explicitly measure the variability of the results possibly due to changing channel conditions. Using this technique, we are able to obtain repeatable results and distinguish differences among competing alternatives in several challenging scenarios, including mobile environments with both WiFi and non-WiFi interference. We recommend that multiple interleaved trials be considered when conducting 802.11n MIMO experiments and stress that, regardless of the methodology used, it is critical for researchers to understand, quantify, and report on the variability of their results.

Chapter 4

T-RATE: Trace-based Simulation of 802.11g networks

4.1 Introduction

The throughput of a device using a WiFi network depends on its rate adaptation algorithm's (RAA) ability to successfully infer the quality of the communication channel being used and to select the rate that maximizes the device's goodput. Rate adaptation algorithms attempt to create a balance between high transmission rates and low enough channel error rates to maximize throughput. Experimental evaluations and comparisons of RAAs are extremely challenging due to the variability of channel conditions. Non-WiFi devices, such as microwave ovens, cordless phones, and other devices operating in the same spectrum make the situation worse. On the other hand, the complexity of wireless channels makes the accurate simulation or emulation of RAAs extremely difficult (and possibly infeasible).

As a result, recent evaluations of RAAs use traces to increase the realism of simulators and emulators [111] [94] [116] [72] [58]. Traces are collected and used to capture *some* aspects of wireless channel conditions, such as frame loss due to path-loss and fading, in order to avoid having to simulate/emulate them. Because the current approaches to collecting traces are able to capture only some aspects of wireless channel conditions, the environments under which this approach can be used are currently limited. For instance, the most recent approach [111] is limited to interference-free environments for trace collection, making it difficult to use in some important scenarios that are representative of those in which devices are likely to be used.

Throughput is the main performance metric used to evaluate 802.11 RAAs. It can be affected by the sender's ability to *access* the channel, the *error rate* observed when using the channel, and the transmission rate. Figure 2.7 depicts the environmental factors that affect the throughput. The properties of a channel that have been used in previous trace-driven evaluations of RAAs are represented by the shaded area. In contrast, our work is driven by the insight that data regarding the important properties of a wireless channel that affect the throughput and comparison of 802.11 RAAs can be captured in traces collected during real experiments. In other words, we capture and utilize all of the information regarding the factors shown in Figure 2.7. As a result, we are able to simulate the operation of the 802.11 MAC layer (as it relates to RAAs) to precisely mimic the properties of the wireless channel that affect throughput. The result is a framework for the fair and accurate comparison of RAAs using scenarios that are more representative of actual use conditions than previously possible.

4.2 Trace-driven Framework

At a high-level, the goal of our approach is to enable traces to be collected with relative ease, under a variety of channel conditions. To evaluate the performance of rate adaptation algorithms, each trace is processed using the algorithms being studied. Because each algorithm uses exactly the same trace and each trace captures the channel conditions relevant to rate adaptation algorithms, this enables the fair, realistic, and repeatable comparison of algorithms. Figure 4.1 provides an overview of our proposed framework. First, in the trace collection phase, a trace of 802.11 transmissions are collected to capture different environmental factors that affect the throughput. Only information about a single frame with certain properties (e.g., 54 Mbps transmission rate) can be collected at time T. However, when we later simulate a rate adaptation algorithm, information about a frame with different properties (e.g., 48 Mbps transmission rate) might be required at time T. This frame is *missing* in the collected trace. Therefore, in the trace preparation phase, we estimate missing frames using collected frames and store them in complete traces. Finally, our trace-processing engine simulates the operation of the 802.11 MAC layer, UDP, and RAAs using the complete trace and reports performance results.

4.2.1 Trace Collection

The goal of the trace-collection phase is to obtain a trace of an actual experiment whose 802.11 frame transmissions can be used to generate a representative complete trace for use

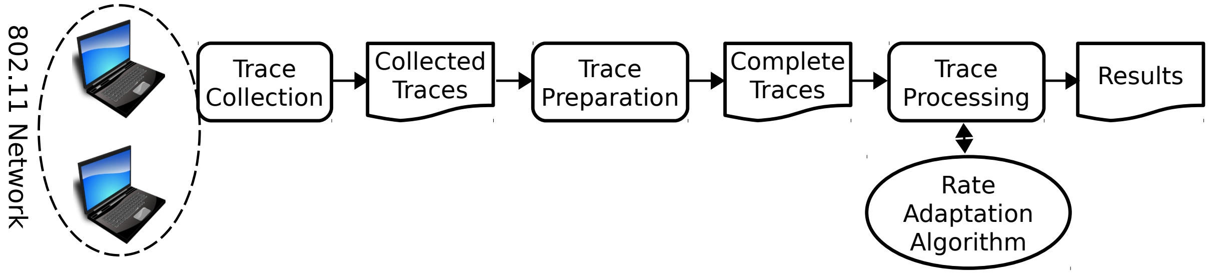


Figure 4.1: Overview of our trace-driven framework

by the trace-processing engine to evaluate RAAs. During this phase, an 802.11 network, consisting of an access point and a client (sender) is built. The sender saturates the link by transmitting as many 1470-byte UDP packets as possible to the access point. We now describe the techniques we have developed to capture the environmental factors shown in Figure 2.7.

Collecting channel access data

All 802.11 devices perform carrier sensing before the transmission of a frame. The delay caused by a busy channel lowers the effective throughput. We use two mechanisms to detect and capture the effect of WiFi and non-WiFi interference on channel access.

WiFi sources: We use TCPDUMP [106] to capture the traffic generated from a sending device in addition to traffic that does not belong to our experiment (third-party traffic). Third-party traffic is used by the trace-processing engine to mimic the delay caused by detecting WiFi transmissions during carrier sensing.

Non-WiFi sources:[1] Capturing the effect of undecodable signals from non-WiFi devices detected during carrier sensing (energy detection) is more challenging. We exploit some information reported by widely used Atheros-based chipsets to calculate and capture the delay caused by energy detection on the channel. This information has been previously used for localization [43] and channel utilization estimation [18]. To our knowledge, we are the first to use this information to estimate the amount of time a channel is busy due to non-WiFi interference. The Atheros-based chipsets can report Cycle Counter Information (CCI) that indicates the time the chip spends transmitting and receiving data. This

[1] In addition to non-WiFi devices such as cordless phones and microwaves, we treat WiFi devices operating on overlapping channels as non-WiFi interferers, because like non-WiFi sources their signal can not be decoded.

information is critical to obtaining accurate timing for packet transmission so we now describe the details of how it is obtained and used.

We have modified the Ath9k driver to report CCI information at the end of every transmission (i.e., when an ACK is received or is timed out). The chip uses a 40 MHz clock and in addition to the total number of cycles between transmissions (cycles), it reports the number of cycles spent receiving a frame (rx_frame), spent transmitting a frame (tx_frame), and the number of cycles during which the channel was busy (rx_busy).

In an interference-free environment, we expect to have: rx_busy = tx_frame + rx_frame. However, when the sender detects undecodable energy on the channel during carrier sensing (e.g., non-WiFi interference), we have: rx_busy > tx_frame + rx_frame, because in addition to the transmission of the frame and possibly receiving the ACK, the channel was busy due to non-WiFi interference. Moreover, non-WiFi interference causes additional delays, since the channel needs to remain idle for the duration of a DCF inter-frame space (DIFS) before the back-off countdown can be resumed. In general, data transmission is delayed by one DIFS and possibly a portion of a back-off time slot for each detected burst of energy. Figure 4.2 shows two different examples where the same amount of busy time can result in significantly different delays before being able to transmit data. The example in the top line shows data transmission being delayed by three DIFS because two bursts of energy were detected. The bottom line shows that despite energy being detected for the same amount of time, the packet is delayed by only two DIFS because only one burst of energy was detected.

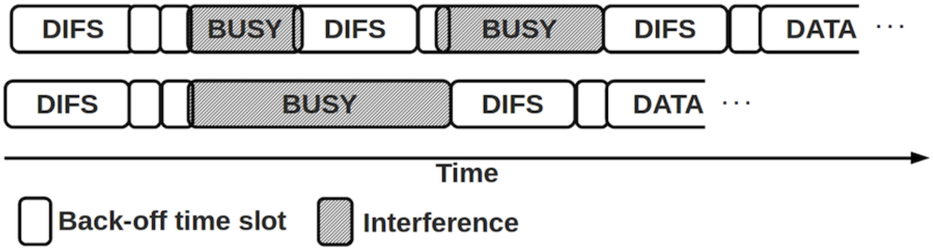


Figure 4.2: Interference and channel sensing

Energy bursts may also occur during DIFS, which can lead to losing partial DIFS. Therefore, we used the following formula to calculate the total delay caused by interference detected at the sender:

$$
\begin{aligned}
Delay = {} & \frac{cycles - (tx\_frame + rx\_frame)}{Clock\ Frequency} \\
& -(DIFS + BO \times BO\_time\_slot + SIFS)
\end{aligned}
$$

where BO is the back-off value used (i.e., the number of back-off time slots), and BO_time_slot is the time for each back-off. The first line of the formula (the fraction) calculates the total time not sending or receiving WiFi frames. The second line deducts the standard wait times defined by the 802.11 standard (even when there is no interference). As a result, $Delay$ will represent any additional delay caused by non-WiFi interference. Because BO is chosen by the physical layer and is not reported to other layers, it is not possible to determine the value used. Since the back-off values are chosen uniformly between 0 and 15, we use an average back-off value of 7.5 in this equation.

During the evaluation of our prototype implementation, we found that it was necessary to include CCI along with this heuristic to accurately capture the effect of non-WiFi interference on the sender.

Collecting channel error rate data

A key challenge of trace collection is that in order to later process a trace, we would like to be able to determine if a packet could be successfully received if it was sent at any available rate r at time t for all R available rates. However, while collecting a trace, we can send a packet using only one of the R supported rates, at any given time t. Unfortunately, sending packets from multiple devices on the same sender would cause packet collisions at the receiver.

Vutukuru et al. [111] propose sampling each of the available R rates in a strict round-robin fashion to get a sample from each rate r in a short time window W. The trace file records whether or not the frame transmission was successful for each sampled rate. Later during simulation, the fate of the single packet sampled at rate r in a particular window W_i is used to determine the fate of all packets that could be sent at rate r in the same time window W_i. One disadvantage of this approach is that during simulation it assumes the frame error rate is either 0% or 100% for all sampling windows. In some cases, tens of frames can be transmitted during a time window. Unfortunately, some rate adaptation algorithms are highly sensitive to the consecutive failure or success of frames and may react unrealistically if evaluated using this technique. Therefore, we use a larger time window (i.e., 100 ms) to obtain several samples for each rate. We then compute the expected frame error rate for each rate during each time window. We choose a time window that is smaller than the expected *channel coherence time*. The channel coherence time is the time over which the channel conditions remain relatively constant and the channel can be assumed to be stationary. For 802.11 networks operating in the 2.4 GHz spectrum, where mobility is limited to a walking pace[2], studies show that the channel coherence time is between 100 and 200 ms [98, 101]. Sadeghi et al. [98] report that the channel coherence time reduces to 24 ms, 12 ms, and 6 ms for speeds of 18, 36, and 72 km/hr, respectively. In our prototype implementation, we use a 100 ms time window (50 ms before and after the point in time being considered). Hence, we require the coherence time to be above 50 ms which is the case when the speed of a mobile device is limited to walking speed. Our evaluations show that the design choice of 100 ms for the sampling window produces accurate results for experiments where movement includes walking speed. Studying higher speeds such as those in vehicular networks are left for future work.

We use a different technique to scan rates than used previously [111]. During the course of our investigation, we observed that traces collected using a strict round-robin strategy [111] were not as realistic as desired. We observed that the length of each round can be synchronized with the duty cycle[3] of some non-WiFi interferers such as frequency-hopping spread spectrum (FHSS) devices like cordless phones. These devices transmit signals periodically at a particular frequency. Hence, some frames sent at a particular rate can experience interference with higher probability than others. This bias in error rate for some rates affects the realism of the results obtained from traces collected using a strict round-robin strategy. In some experiments, we observed that a strict round-robin strategy measures error rates with a difference as large as 21% when compared with base measurements. The base measurements were obtained using actual experiments that use a constant rate. By comparison, our pseudo-random approach obtained differences of less than 4.8%.

To our knowledge, a novel technique we use is the addition of scanning RTS/CTS protected frames. Previous work [111] performs sampling in interference-free environments, and therefore ignores RTS/CTS. However, our work targets environments where WiFi interference may exist. In this case, the frame error rate depends on the use of the RTS/CTS protection mechanism. Our goal is also to support RAAs that may selectively enable and disable RTS/CTS [114]. As a result, we scan every other round with the RTS/CTS option

[2] The channel coherence time is inversely proportional to frequency, therefore the channel coherence time of networks operating in the 5 GHz is lower (about 100 ms).

[3] The fraction of time a device occupies the spectrum by transmitting signals when the device is active.

enabled.

In order to increase the realism of collected traces, we considered shortening the time required to complete a round by sampling only a subset of all available rates. For example, consider two simple heuristics for reducing the number of samples. 1) Sample rates in order from higher rates to lower rates. If k consecutive rates are successful, assume all rates lower than the lowest sampled rate will be successful (because they use more robust encodings). 2) Sample rates in order from lower rates to higher rates. If k consecutive rates are unsuccessful, assume all rates higher than the highest sampled rate will also be unsuccessful.

When analyzing data collected in environments where there is no interference the accuracy of these techniques is relatively high (even when a device is mobile). Unfortunately, such heuristics do not work in general due to the presence of interference, which is prevalent in environments in which WiFi devices are used [79] [95].

To study the problem, we collected and examined several traces collected by sampling all rates and then post-processing the trace to determine whether or not the heuristics could accurately determine the fate of the packets that would not be sampled using the heuristic. Table 4.1 shows that the fate of a significant percentage of packets would be incorrectly predicted. As a result of these experiments, our prototype samples all rates. This provides highly accurate results in the scenarios we have examined (see Section 4.4).

k	High to Low	Low to High
2	14%	64%
3	12%	52%

Table 4.1: Errors in packet fate estimation heuristics

In future work, if we encounter scenarios where we are not able to sample all rates or need to increase the accuracy of T-RATE, we would consider using software-defined radios. For example, we could shorten rounds, as is done in AccuRate [100], by extracting physical-layer information about the constellation of a received frame to estimate the fate of other rates that use different modulation and coding schemes.

The trace data

All information is collected on the sending device, except for the SNR of the received frame (which is collected on the receiver). In addition to channel access and channel error rate data, we collect other information, such as the SNR of data frames and acknowledgments, to enable the trace-processing engine to evaluate RAAs that require this information (like RBAR [52] and CHARM [60]). Note that, if available, additional information could be collected. For example, physical-layer properties could be obtained via special purpose hardware or from software-defined radios [111].

The *collected traces* (Figure 4.1) are the Experiment Traffic Trace (ETT) and the Third-party WiFi Traffic Trace (TWTT). The ETT includes all frames transmitted by the sender and the per-frame delay to access the channel caused by non-WiFi sources. The data in the ETT includes:

- Source: Ath9k Driver.
  - Channel Cycle Information (CCI).
- Source: TCPDUMP sender side.
  - Timestamp: the transmission time of the packet.
  - 802.11 transmission rate.
  - Was the packet sent using RTS/CTS.
  - ACK: specifies if the packet was received.
  - SNR of ACK.
- Source: TCPDUMP receiver side.
  - SNR of Data frame.

The ETT includes channel access delay caused by non-WiFi sources only. The TWTT collects frames from interfering WiFi traffic that cause additional channel access delays at the sender. The data in the TWTT includes:

- Source: TCPDUMP sender side.
  - Timestamp: arrival time of the frame.
  - Frame length and transmission rate: used to calculate the time required to send this frame (carrier sensing).
  - Duration: channel time reserved for acknowledgements or next frames (virtual carrier sensing).

4.2.2 Trace Preparation

Channel error rate

Because at any point in time t, we can only scan a single rate r with RTS/CTS on or off, if a rate adaptation algorithm chooses to send a packet at time t using any rate other than r and/or using a different RTS/CTS configuration, we will not have a corresponding packet in the collected trace. Therefore, we need to determine or compute information about all packets that could have been sent starting at time t. We call this phase trace preparation, because we use information from the collected trace (which contains only information about rates that were used and packets that were sent) to prepare a complete trace (which contains information about all rates that could have been used).

To predict if a missing packet could be received or not, our prototype implementation of T-RATE uses a window of 100 milliseconds (50 before and after) to compute packet reception probability. This is computed based on the number of packets sent and received within that time window. At time t, for each rate r, and for RTS/CTS on and off, we consider only packets from the collected trace that are within the window, were sent at rate r, and which were sent with RTS/CTS on and off, respectively. The computed probability is then used to predict if a missing packet would be received or not. That is, for each rate r, we record whether or not the packet would have been received with RTS/CTS on and off. This information is recorded in the complete trace. In the complete trace, at time t, for each rate r, we compute and record the estimated fate of the frame (as just explained), the estimated delay for accessing the channel, and the estimated RSSI, which are explained next.

Channel access

The delay caused by non-WiFi interferers is stored in the ETT, while third-party WiFi traffic received by the sender is stored in the TWTT. The TWTT is used directly by the trace-processing engine. On the other hand, the trace-processing engine needs to know the delay caused by non-WiFi interference for a packet ready to be transmitted at time t. Unfortunately, we may not have such a sample in the ETT. We use the 100 ms time window to compute the expected delay at time t, using a technique similar to that used for estimating the fate of a missing packet (explained in the preceding paragraph). Our evaluations show that this technique accurately predicts the expected delay caused by non-WiFi interferers.

Additional data: Signal Strength

We currently estimate the signal strength of a missing packet using a window of two packets. We use a linear interpolation of the RSSI (Received Signal Strength Indicator) of the two surrounding packets. The Atheros chipset used in our prototype implementation of T-RATE has a noise floor of -95 dBm, so the SNR can be determined from the RSSI. To evaluate the accuracy of our RSSI estimation technique, we adapt the methodology used in [58]. We downsample the RSSI data (i.e., remove data) obtained from an actual experiment to create a trace that is similar to the collected trace (i.e., it is missing the same amount of information as a collected trace). Then, we apply our RSSI estimation algorithm to the downsampled trace to prepare a complete trace. We then compare the information from the complete trace with that from the actual experiment. In an experiment conducted using a mobile device moving at walking speed, we observed that over 85% of the estimated RSSI values are within 1 dBm of the actual values and that the error is generally bounded by a few dBm, as can be seen in Figure 4.3. Although the trace used for this experiment was obtained from a challenging mobile experiment with highly variable channel conditions, our heuristic is highly accurate.

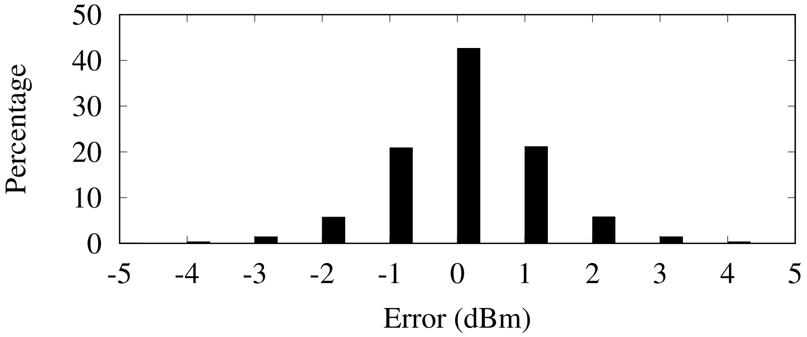


Figure 4.3: RSSI estimation error

Clearly, other approaches and different parameters could be used during trace collection and preparation. Although the evaluation of our prototype implementation of T-RATE in Section 4.4 demonstrates that our prototype produces highly accurate results for the scenarios we have tested, in future work we plan to explore some of these choices. For example, the choice of window length is a design decision that depends on the variability of the wireless channel conditions and different window sizes may be needed in environments with faster moving mobile devices or more transmission rates (e.g., in 802.11n).

4.2.3 Trace Processing

During the trace processing phase, a rate adaptation algorithm is simulated using the complete trace. The complete trace consists of the prepared experiment traffic trace (P-ETT) and the third-party WiFi traffic trace (TWTT). Trace processing involves tracking the current (simulated) time and determining which rate would be chosen by the given algorithm. Figure 4.4 illustrates the approach. Before the transmission of any frame, the simulator first checks the third-party traffic trace (TWTT) to simulate the delay caused by receiving WiFi frames at the sender. If any frame was observed at time T, the delay caused by receiving this frame is calculated. This includes the time the received frame is in the air (carrier sensing) plus any additional time for which the frame reserves the channel (virtual carrier sensing). Then, the simulation time (T) is updated and the TWTT trace is checked again for any unprocessed third-party frames at or before time T. We consider the unprocessed frames in the TWTT before time T, because third-party transmissions that started before T also affect channel sensing. When no such frames exist, the simulator proceeds to the transmission of the next frame.

The next step is to simulate the delay caused by non-WiFi devices while performing carrier sensing. This delay is directly read from the complete trace. Additional delays caused by the DCF protocol, such as back-off time, are computed and the simulation time is updated. The trace-processing engine implements DCF inter-frame spacing to achieve realistic timing and to ensure the accurate computation of throughput. The simulator looks in the complete trace to determine the fate of the frame sent at time T, rate r (specified by the RAA), and with the RTS/CTS option on or off. To accurately compute packet transmission times, we consider the structure of the 802.11g frame and the transmission time of each segment (i.e., header and payload). If the frame transmission fails, the frame is retransmitted until the frame is successfully transmitted or the retransmission limit is reached. The result is logged and is reported to the RAA. The logged data includes the rate used and whether or not the frame was received. We then post-process the log file to better understand an algorithm's behavior and to compute statistics related to performance, such as the average throughput over different time scales and the total number of bytes received.

4.2.4 Limitations

During the trace collection phase, we sample all rates as quickly as possible to maximize the number of collected samples. Although we are able to simulate delays before sending packets due to third-party traffic, when contention is too high the sender's access to the channel will be limited and may not be able to obtain enough samples to produce realistic complete traces. In some of our experiments, the sender's ability to access the channel is limited and fewer samples are obtained. Although we are able to obtain highly accurate results in several interesting scenarios in this study (see Sections 4.4.1 and 4.4.3), further work is needed to understand the number of samples required to obtain realistic results.

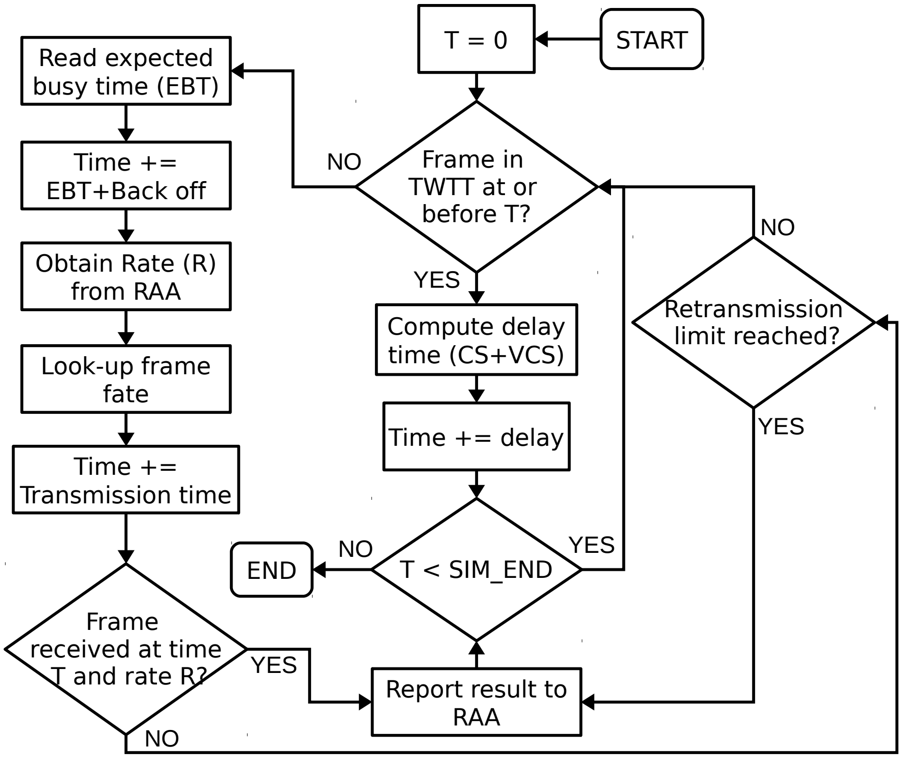

START
T = 0
Frame in TWTT at or before T?
NO
Read expected busy time (EBT)
Time += EBT+Back off
Obtain Rate (R) from RAA
Look-up frame fate
Time += Transmission time
Frame received at time T and rate R?
YES
NO
YES
Compute delay time (CS+VCS)
Time += delay
T < SIM_END
NO
END
YES
Report result to RAA
Retransmission limit reached?
NO
YES

Figure 4.4: Simulation flowchart

In this work, we utilize UDP to transmit data from the sender to receiver for trace collection and simulation. UDP is commonly used to evaluate the performance of rate adaptation algorithms [75,89]. The advantage of using UDP is that the performance is only impacted by the MAC and physical layer. This avoids potential complications introduced by other layers, such as TCP congestion control, making it difficult to attribute changes in performance solely to rate adaptation. Due to the wide use of TCP, it might be beneficial to understand interactions between rate adaptation algorithms and TCP. As a result, in future work, we will consider implementing TCP in our trace-based simulator. Also, during trace collection and simulation, we use a packet size of 1470 bytes and saturate the link, as is common practice when evaluating the performance of rate adaptation algorithms [18,37]. Studying other packet sizes and other application layer transmission rates is left for future work.

4.3 Evaluation Methodology

The purpose of our evaluation section is not to evaluate different RAAs but instead to evaluate the prototype implementation of T-RATE and to demonstrate that it can be used with a variety of RAAs. When possible, we compare the throughput obtained by T-RATE with throughput from an actual experiment. This requires us to be able to construct scenarios where we can ensure that the channel conditions are reasonably repeatable. Thus, we conduct some experiments in a controlled environment where there is no interference from devices other than those we use to generate interference.

For some evaluations, we compare the throughput obtained with an actual experiment using a fixed MAC data rate with that obtained by the trace-processing engine using that same rate. The reasons for this approach are: (1) At any point in time, a rate adaptation algorithm may choose any one of the available data rates. This ensures that for any rate chosen by any algorithm, the trace-processing engine will match the throughput obtained in the experiment. (2) It is easier to reason about the expected throughput for a fixed rate than for an algorithm that is constantly adjusting the rate. The rates available in 802.11g are 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. Most of our graphs show only rates of 6, 24 and 54 Mbps, because other rates show similar behaviour.

Despite using carefully controlled and monitored environments, we could not obtain repeatable results for some experiments (e.g., for some hidden terminal and mobile experiments). Additionally, we are interested in evaluating our prototype in uncontrolled environments where repeatability is not possible. To evaluate T-RATE, we would ideally like to compare the accuracy of T-RATE with that of an experiment. This requires running two experiments, one to collect a trace to be used by the simulator, and another using identical channel conditions which will be used as a basis for comparison. In an uncontrolled environment, if channel conditions change between these two experiments, T-RATE results should not be expected to match the experiment. We realized that the trace collection experiment can serve two purposes: (1) it is an experiment in which the rate adaptation algorithm follows a specific round-robin ordering; (2) it is also a T-RATE input trace, because it samples all rates in a round-robin ordering. If we simulate a rate adaptation algorithm that selects transmission rates in a round-robin order, we would expect the throughput of T-RATE to match that of the original experiment. Therefore, for the purpose of evaluating T-RATE, we simulate a round-robin rate adaptation algorithm that has the same expected throughput as the experiment but with a pseudo-random ordering of rates to force the simulator to estimate channel access and channel error rate, rather than simply replying the trace. This ensures that the trace-processing engine must use information from frames added to the complete trace (during the trace-preparation phase) that are not present in the collected trace.

4.3.1 Equipment and Software Used

We continuously monitor all of our experiments using an AirMagnet Spectrum XT [21]. This analyzer is able to find and classify WiFi and non-WiFi interference. Our 802.11g networks are constructed using netbooks and laptops running Linux with the Atheros Ath9k driver. We use a 2.4 GHz FHSS device (a controller for an RC helicopter) to generate non-WiFi interference. This device was chosen because it has a USB interface that allows us to programatically turn the device on and off at precisely the same time when repeating experiments. We use Iperf [54] to send 1470-byte UDP packets as fast as possible between the sender and the receiver. When plotting throughput versus time, each point shows the average and 95% confidence intervals of 10 half-second measurements.

4.3.2 Different Evaluation Scenarios

Figure 4.5 illustrates the different devices and their relative positions in our stationary and mobile scenarios. Not all devices are used in all experiments; the details of each experiment are provided in the evaluation section. In mobile scenarios, we use an electric train to carry a notebook computer (the mobile device). A train is used to ensure that the same path is traveled at the same speed (i.e., approximately walking speed) for multiple runs of the same experiment. The train moves on a U-shaped track. Every time the train reaches one end of the track the direction is reversed.

4.4 Evaluation

To conduct a comprehensive evaluation of our prototype, we consider the *environmental factors* that can affect the throughput of an 802.11 network as shown in Figure 2.7. In Section 4.4.1, we evaluate our prototype's ability to obtain the correct throughput when the sender's channel access is affected. In Sections 4.4.2 and 4.4.2, we evaluate the accuracy of T-RATE using scenarios where the channel error rate is affected due to interference.

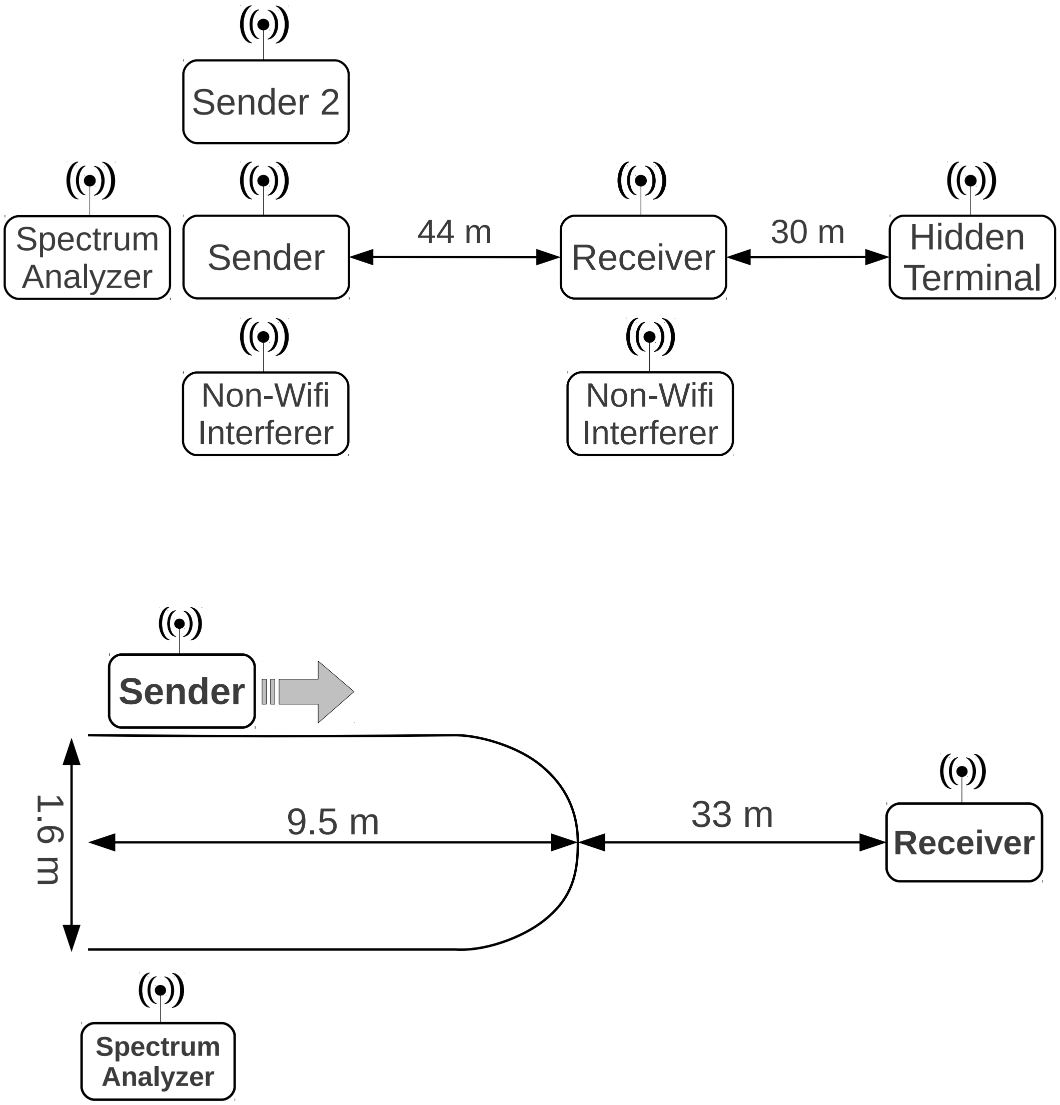
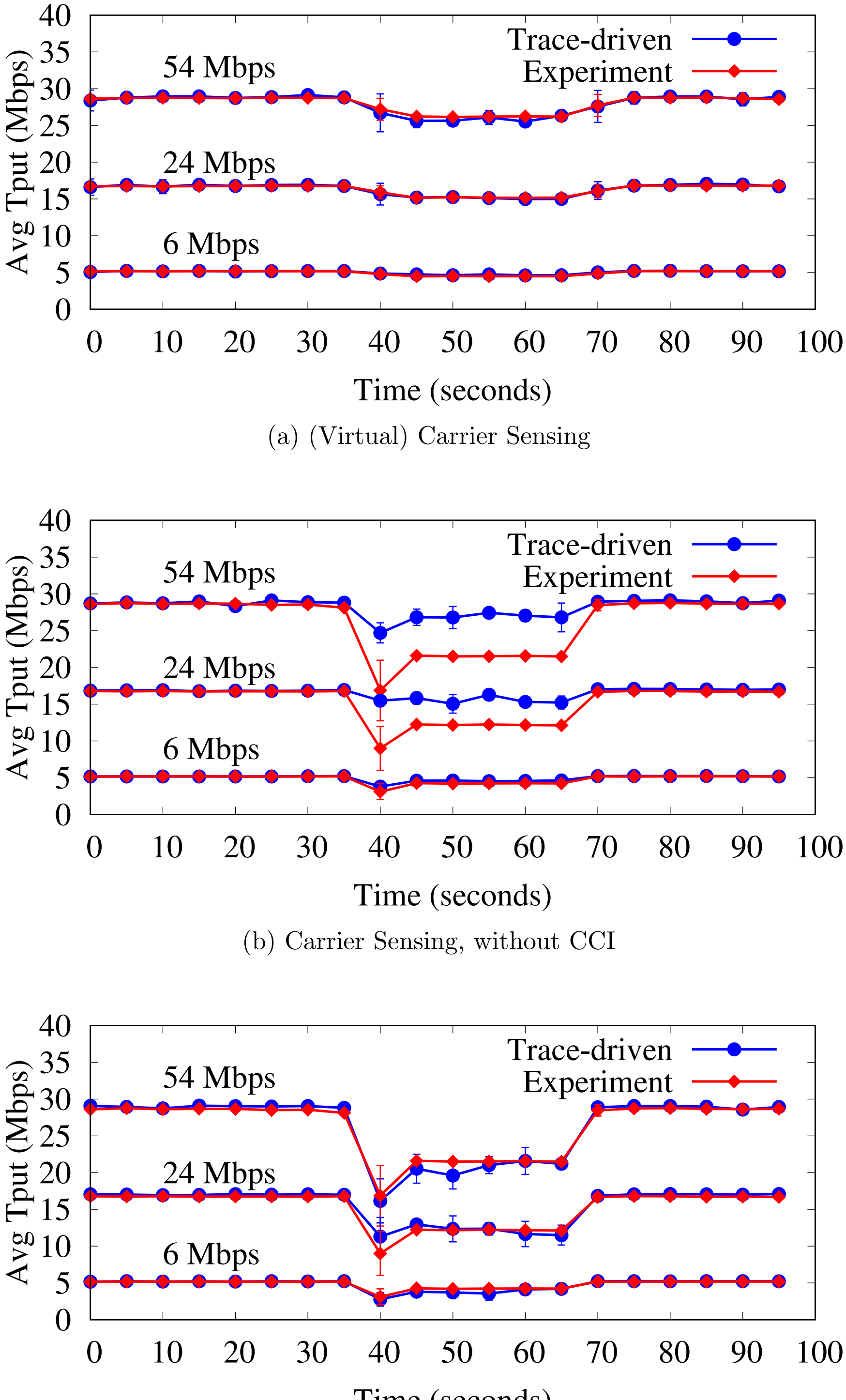


Figure 4.5: Stationary (top) and mobile (bottom) experiment setups



(a) (Virtual) Carrier Sensing

(b) Carrier Sensing, without CCI

(c) Channel Sensing

Figure 4.6: Channel Access

4.4.1 Channel Access

We now evaluate the realism and accuracy of T-RATE when channel access is limited because of WiFi and non-WiFi interference near the sender.

WiFi Source

We first test how well T-RATE is able to handle the effect of third-party WiFi traffic on channel access.

Experiment Setup: In addition to the Sender sending to the Receiver (Figure 4.5 – top), Sender 2 also sends UDP traffic to the same Receiver at a steady rate of 3 Mbps, roughly mimicking a video stream. This traffic is injected from time 40 to 70 seconds. Results of these experiments are shown in Figure 4.6(a).

This scenario mainly targets the processing engine's ability to handle the delay, which is observed during carrier sensing and virtual carrier sensing, caused by third-party traffic. Figure 4.6(a) is obtained by conducting four experiments. The environment is controlled and monitored to ensure that channel conditions are repeatable across each experiment. One experiment is run to collect a trace and the others are run to determine the ground truth for the three fixed rates. The throughput obtained using each of the three fixed rates is compared with that reported by the trace-processing engine when using those same fixed rates.

The tight match between the trace-driven and experimental results suggests that the processing engine is able to handle this scenario with high precision. As expected, the throughput of the rates 6 and 24 are only slightly affected while WiFi interference is present (from 40 to 70 seconds) while the throughput of the 54 Mbps experiment has a more noticeable 3 Mbps drop. Also note that when the environment is *interference-free* (prior to 40 and after 70 seconds), the throughput obtained for each rate matches the theoretical throughput expected.

Non-WiFi Source

Non-WiFi transmitters can also cause the carrier sensing protocol to report busy, which can decrease throughput by introducing delays before a packet can be sent. We now study the accuracy of handling the channel access delay caused by non-WiFi sources.

Experiment Setup: A non-WiFi transmitter (Figure 4.5 – top) near the Sender operates from time 40 to 70 seconds. Results from this experiment are shown in Figure 4.6(c).

Figure 4.6(b) and (c) show the importance of the channel cycle information (CCI). When CCI is not used (Figure 4.6(b)) the throughput obtained using T-RATE does not closely match the experiment. However, Figure 4.6(c) shows the excellent match obtained for all rates by including CCI in T-RATE.

4.4.2 Channel Error Rate

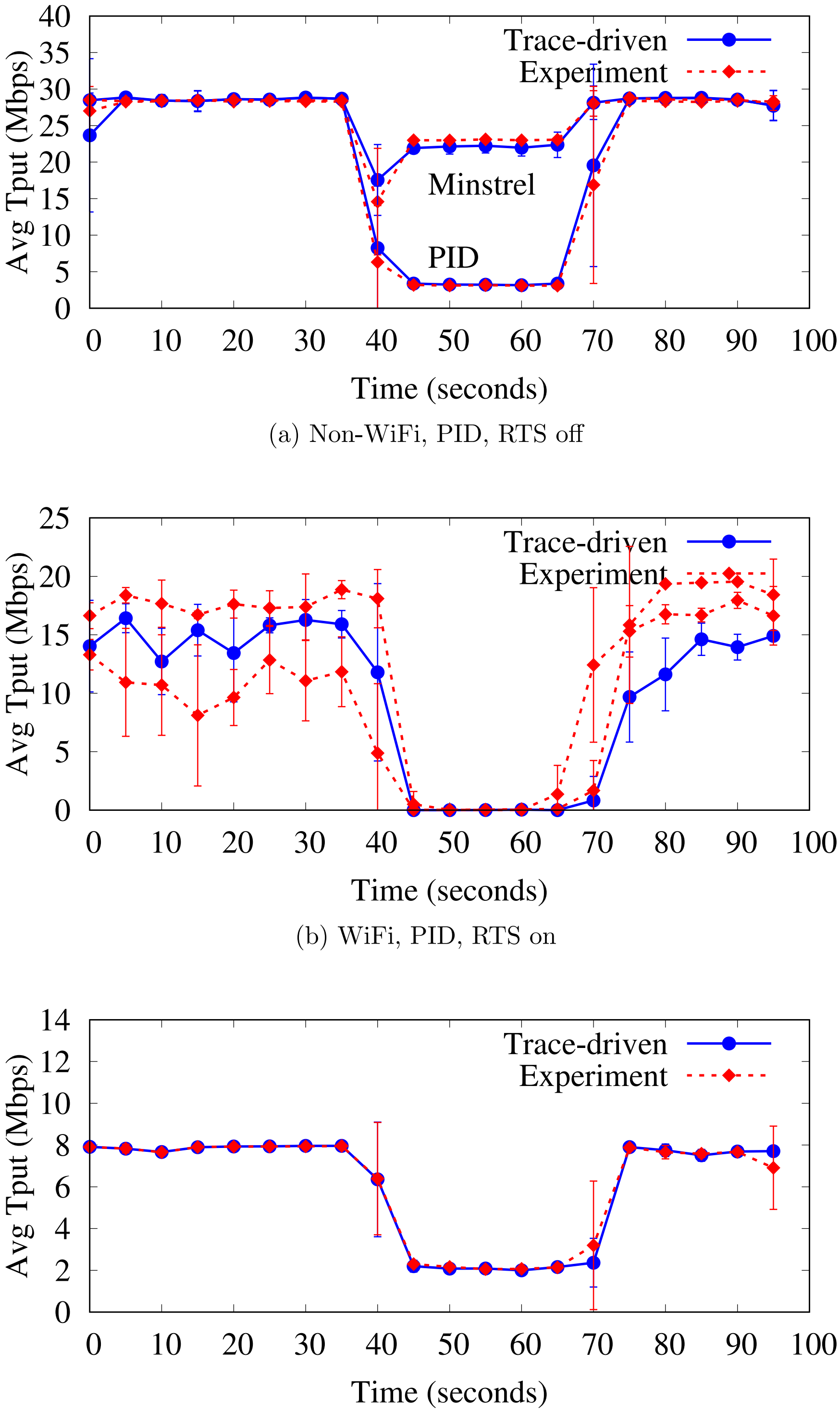
A sender's throughput also depends on the channel error rate. A channel's error rate can be influenced by two important factors: interference (which can come from WiFi and non-WiFi sources near the receiver) and signal propagation (i.e., path loss and fading). In this section, we examine the accuracy of T-RATE using scenarios where these influences are present.

Non-WiFi Interference

This experiment is designed to evaluate the accuracy of our framework when the channel error rate is affected by a non-WiFi interferer.

Experiment Setup: In this experiment, we place a non-WiFi interferer near the Receiver (Figure 4.5 – top) and out of range of the Sender, so that it does not affect channel access. The non-WiFi transmitter operates from time 40 to 70 and the results are shown in Figure 4.7(a).

Figure 4.7(a) compares the throughput obtained from real experiments using the PID (Proportional-Integral-Derivative) and Minstrel algorithms from the Linux Ath9K driver with throughput obtained using the trace-driven evaluation of these algorithms. These graphs show that for both algorithms, there is a tight match between the experiments and those obtained with T-RATE. The RAAs implemented in the trace-processing engine change rates at times that closely match the experiment. This suggests that the trace-processing engine correctly implements these algorithms and that the information required by and available to them in the complete trace is also accurate.



(a) Non-WiFi, PID, RTS off

(b) WiFi, PID, RTS on

(c) WiFi, Round-robin, RTS on/off

Figure 4.7: Channel error rate due to interference

WiFi Interference

We now evaluate the accuracy of T-RATE in the presence of WiFi interference generated by a hidden terminal source.

Experiment Setup: The Hidden Terminal WiFi transmitter in Figure 4.5 (top) transmits a 3 Mbps UDP stream for 30 seconds starting at time 40. The Hidden Terminal is configured to use a 6 Mbps MAC rate in this scenario in order to maintain connectivity with the access point. Results from this experiment are shown in Figures 4.7(b) and 4.7(c).

In the area of the building we used to conduct our experiments, we needed to lower the transmission power of the Sender in order to create a hidden terminal scenario. As a result, the throughput obtained when using fixed rates of 48 and 54 Mbps were consistently zero due to a low SNR. Unfortunately, for the same reason, the throughput obtained with 24 and 36 Mbps is very unstable and was not repeatable across experiments. Since RAAs may choose rates of 24 or 36 Mbps, the throughput is also very unstable when a real RAA is in place. Figure 4.7(b) shows that for two identical runs of an experiment using PID, the throughput can be significantly different. Thus, under this scenario, it is not possible to compare results obtained from experiments with those obtained from the trace-driven framework.

Because channel conditions change significantly from one experiment to the next, the experiments cannot be repeated. Therefore, we cannot expect the results obtained using the trace-driven framework to match those from experiments. Instead, we utilize the alternative method for evaluating our results described in detail in Section 4.3. That is, we compare the results obtained during the trace collection experiment with those obtained using the trace-driven framework while implementing a similar RAA. The excellent match between the experimental and trace-based throughput shown in Figure 4.7(c), suggests that T-RATE handles interference caused by a hidden terminal accurately. In environments where channel conditions change significantly, it is reasonable to collect multiple traces to capture different trajectories of the variable channel conditions in an attempt to achieve a more comprehensive evaluation.

Recall: *Traces are always collected using a round-robin ordering. From each trace, T-RATE can be used to evaluate different rate adaptation algorithms. In some cases, to evaluate the accuracy of T-RATE, we simulate a rate adaptation algorithm that changes the transmission rate in a round-robin fashion. This methodology is used when variable channel conditions prevent us from repeating experiments.*

Stationary Path Loss

In this section, we evaluate our trace-driven simulator when frames are lost due to path loss in mobile and stationary environments.

Experiment Setup: We periodically change the transmission power of the Sender in a controlled fashion (Figure 4.5 – top). The transmission power starts at 17 dBm and is reduced by 1 dBm every 3 seconds until it reaches 0 dBm. It is then increased by 1 dBm every 3 seconds until 17 dBm is reached. Figure 4.8 shows results from experiments conducted using this scenario.

This experiment is designed to evaluate T-RATE's ability to handle frame loss due to path-loss and to evaluate the trace-processing engine's implementation of RBAR [52]. RBAR proposes modifications to the 802.11 standard to transmit SNR information back to the sender. The sender then uses a table lookup to determine an appropriate rate for a given SNR. RBAR is implemented to demonstrate that T-RATE can accommodate SNR-based algorithms. To test the correctness of the operation of RBAR in our trace-processing engine, we have provided RBAR with the values shown in Table 4.2. We constructed this table to cover the RSSI ranges experienced in this experiment, and to force RBAR to select all rates. First, we conduct an experiment with a non-WiFi source of interference, then we study WiFi interference in an hidden terminal scenario.

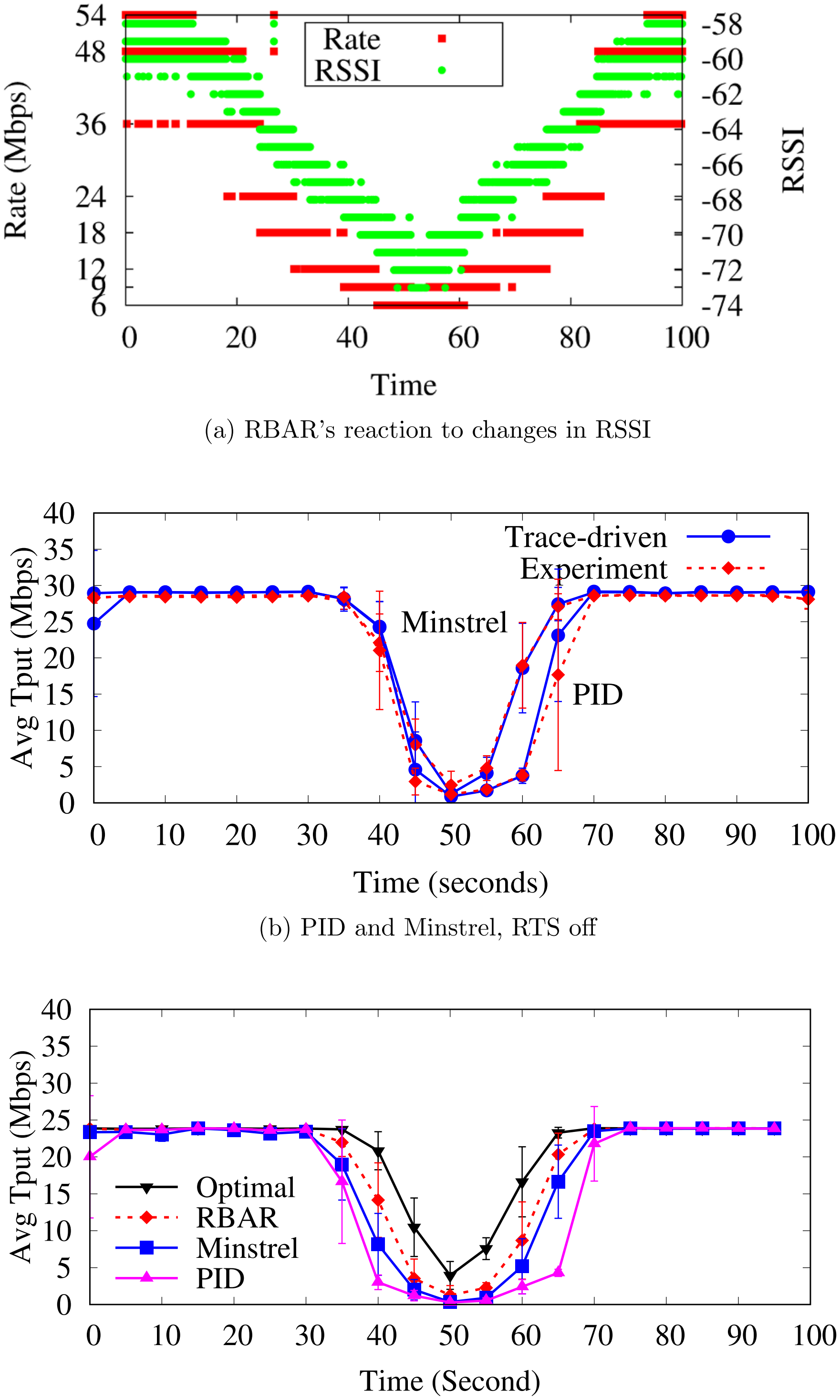
RSSI	Rate	RSSI	Rate	RSSI	Rate	RSSI	Rate
≥ -58	54	≥ -60	48	≥ -62	36	≥ -64	24
≥ -66	18	≥ -68	12	≥ -70	9	≤ -71	6

Table 4.2: RBAR rate selection table

Figure 4.8(a) shows how the RSSI of the received frames (right y-axis) changes as the result of changing the transmission power over time (the x-axis). The left y-axis shows the rates chosen by RBAR. This graph shows that the trace-processing engine's implementation of RBAR operates correctly based on Table 4.2.

Figure 4.8(b) shows results obtained using the PID and Minstrel algorithms both experimentally and using T-RATE. Recall that each point is the average throughput of 10 half-second measurements. The good agreement between the results obtained from the processing engine and experiments shows that T-RATE handles path-loss accurately for this scenario.

Figure 4.8(c) shows the throughput reported by T-RATE for the same experiment when RTS is on. As expected, throughput is lower than in Figure 4.8(b) where RTS is off. We



(a) RBAR's reaction to changes in RSSI

(b) PID and Minstrel, RTS off

(c) PID, Minstrel, RBAR, Optimal: RTS on

Figure 4.8: Path loss (Tx power changing, stationary)

compare the throughput achieved by the PID, Minstrel, and RBAR algorithms with an algorithm that chooses the best possible rate at each moment in time (Optimal). RBAR, which has now been properly calibrated for this environment, outperforms the loss-based RAAs. This is as expected because it can react to changes in signal strength faster. The ability to determine the optimal rate at any moment in time is an additional strength of T-RATE and should provide insights in some scenarios.

Mobile Path Loss and fading

We have evaluated our trace-driven simulator when handling frame loss due to path loss in stationary scenarios. In the previous stationary experiments, path loss was relatively controlled and sudden (by adjusting transmission power every 3 seconds). In a mobile environment, path loss continually changes as devices move. In addition, other effects such as the Doppler effect make mobile environments more dynamic and challenging. In this section, we examine path loss due to mobility. Ideally, the trace-driven simulator should support fast-changing wireless channels in mobile environments.

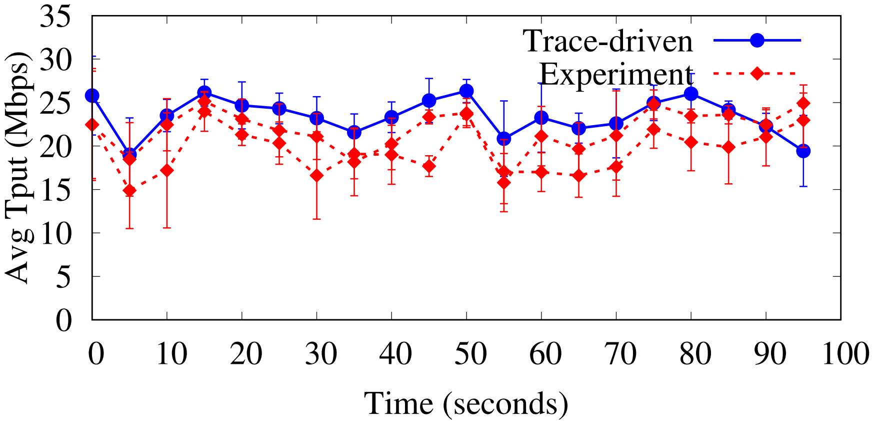
Experiment Setup: The scenario used is as shown in Figure 4.5. To repeat experiments, a mobile device (the Sender) is placed on an electric train that moves at approximately walking speed and results are shown in Figure 4.9(b).

Unfortunately, despite trying to control for all environmental factors, including the exact path traveled, these experiments are not repeatable. Figure 4.9(a) shows the variability across two experiments and results obtained using our framework when the Minstrel algorithm is used. Because of the inability to repeat experiments, we again rely on our alternative approach to evaluate T-RATE. These results are presented in Figure 4.9(b). The excellent match suggests that the results from the trace-driven framework are highly accurate despite the continual changes in signal strength that occur in this mobile scenario.

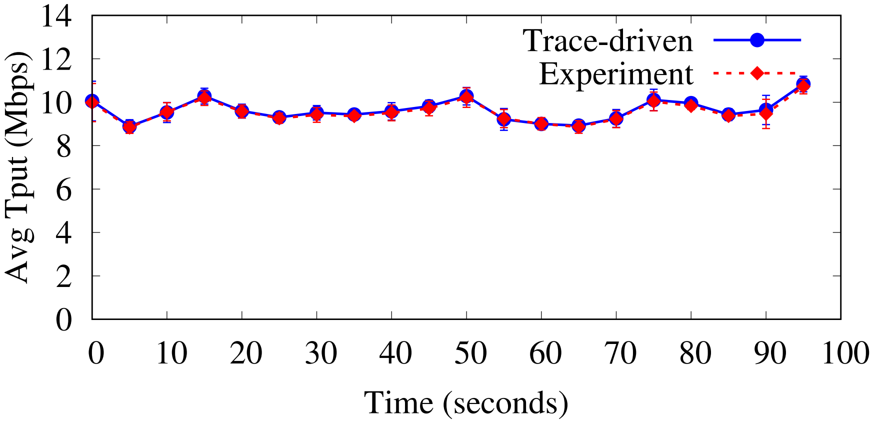
4.4.3 Field Trial

We now test our framework in an environment that is not under our control, and is subject to many sources of interference, path loss, and fading.

Experiment Setup: The receiver is placed in an office environment with multiple cubicles. The sending laptop is carried to different areas of the same office at walking speed. A spectrum analyzer is used to monitor uncontrolled interference (i.e., we do not inject



(a) Controlled Mobile, Minstrel



(b) Controlled Mobile, Minstrel

Figure 4.9: Path loss (mobile)

any controlled interference) from WiFi and non-WiFi sources. Other settings are kept the same as in our controlled experiments. Results are shown in Figure 4.10.

Figure 4.10 shows the throughput reported by the trace-driven framework and that obtained from an actual experiments where the transmission rate is changed in a round-robin fashion (recall that the trace-processing engine uses a different ordering from that used in trace-collection). The figure shows that the trace-driven results match those from the experiment.

We now further analyze the field trial results. The goal is to determine if T-RATE can produce accurate and realistic results when using algorithms other than round robin. Recall that if, for each available rate, at every point in time, the trace-processing engine produces the correct throughput, then no matter which rate is selected by an RAA it will

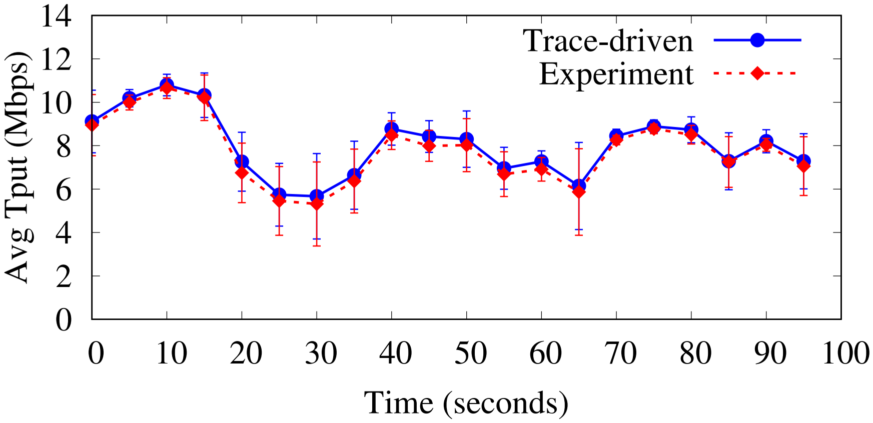


Figure 4.10: Uncontrolled mobile, Minstrel

obtain the correct throughput. For this reason, we examine results obtained via the trace-processing engine, PID and Optimal (as defined in the previous section). The top graph in Figure 4.11 shows the throughput reported by T-RATE using different algorithms. To make the graphs more legible, we have excluded some constant rates, all confidence intervals, and the Minstrel algorithm.

Despite the significant variation in the throughput over time, the 54 Mbps rate provides the best throughput of all the rates (excluding the optimal choice), except for at 65 seconds, where 48 Mbps provides the best throughput. To understand the variation in throughput, and why 54 Mbps provides the best throughput, we first consider the RSSI of received frames. The RSSI graph (second from the top in Figure 4.11) shows that the RSSI of the data frames decoded at the receiver are mostly greater than −60 dBm, so it is unlikely that any transmission rate will experience frame loss due to path loss. Therefore, any decrease in throughput is due to other factors such as limited channel access and/or frame loss due to interference. Indeed, the spectrum analyzer reports that a microwave oven was active at times during this experiment.

The bottom graph in Figure 4.11 shows the average error rate for fixed rates of 24, 48, and 54 Mbps. Comparing the error rate graph with the throughput graph, we see that for each rate, as the error rate increases, the throughput decreases. At time 10 the error rate of all fixed rates is close to 0 and the throughput of all rates is maximized. The cause of the changes in the error rate can be seen in the graph that is second from the bottom. This "interference graph" shows the average expected delay per packet sent, caused by non-WiFi and WiFi interference. Although there is WiFi traffic, the average delay imposed by this

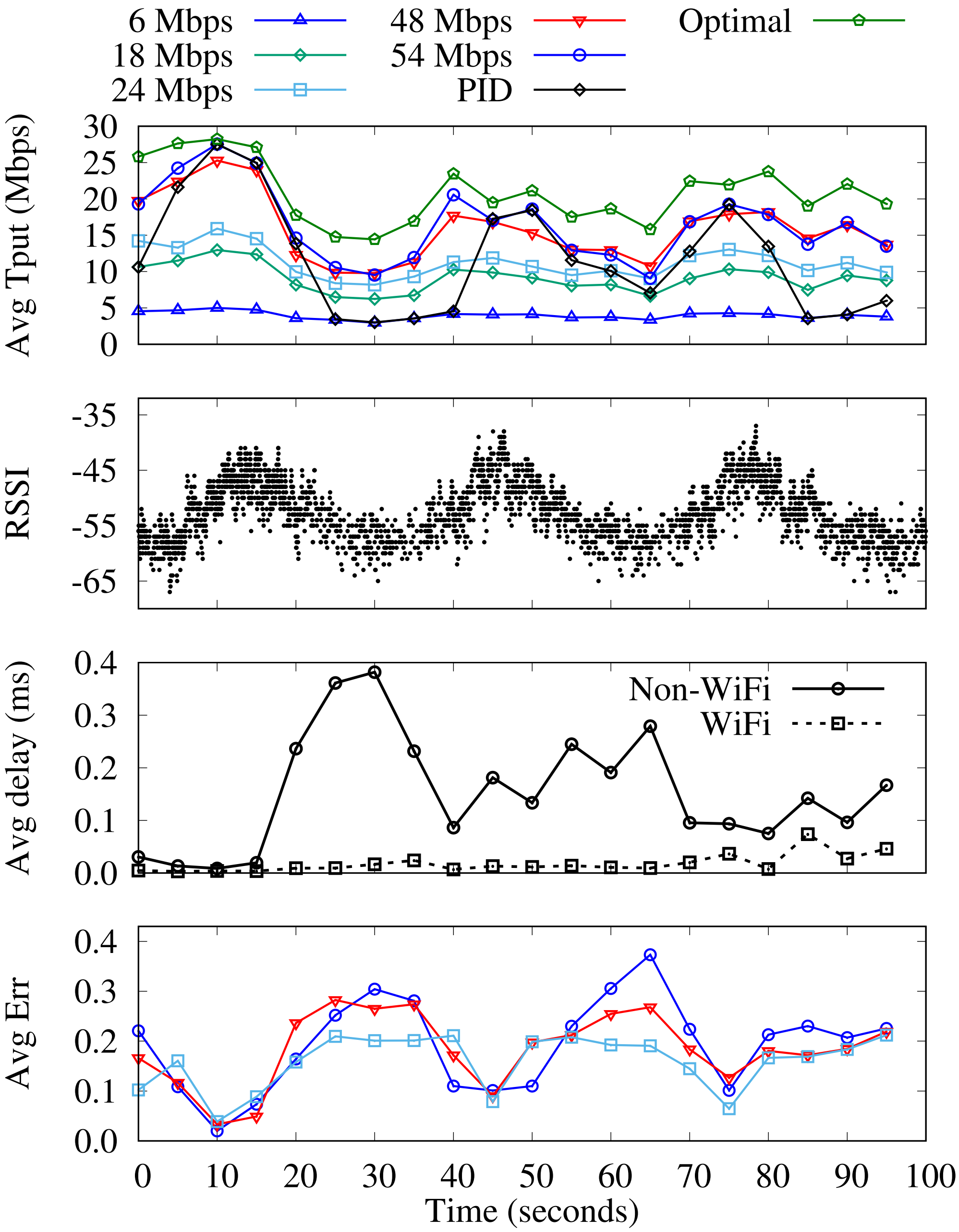


Figure 4.11: Throughput, RSSI, interference and error rate

traffic is low relative to that from non-WiFi sources. The shape of the average error rate graphs roughly corresponds to that of the non-WiFi interference graph.

The behavior of PID can be explained by examining the error rates. When the error rate exceeds a threshold, PID reduces the selected data rate. Because the cause of frame errors in this experiment is non-WiFi interference, reducing the transmission rate may not help. Figure 4.11, shows that as the error rate increases from 10 to 30 seconds, PID reduces its MAC data rate until it reaches 6 Mbps at time 30. It remains near 6 Mbps until after 40 seconds when the error rate starts to decrease and PID starts to choose higher rates.

Overall, the results obtained using our prototype and the behaviour of the algorithms are as would be expected given the environment and channel conditions under which the trace was collected. We believe that T-RATE is a significant step towards improving the evaluation of 802.11 RAAs.

4.5 Discussion

One of the strengths of our approach is that it captures data related to properties of the channel under conditions in which traces are collected. However, a trace is specific to the devices used when capturing the traces. We note that this is also an issue for existing trace-driven and experimental approaches for evaluating RAAs. We plan to study the degree to which results might be used across different devices and scenarios in future work. We also plan to: examine scenarios where mobile devices move at higher speeds; better understand and outline limitations that might be caused by limited channel access (e.g., due to dense WiFi use); collect traces using a wide variety of scenarios; and examine 802.11n and 802.11ac networks. In Chapter 5, we provide an overview of a trace-based framework for 802.11n networks along with a novel trace collection methodology for handling devices that support many transmission rates.

4.6 Chapter Summary

In this chapter, we present the design, prototype implementation, and evaluation of T-RATE, a trace-driven framework for evaluating 802.11 RAAs. We devise mechanisms that allow us to capture traces on communicating 802.11 devices, while conducting experiments under realistic conditions. We show that T-RATE can be used to conduct highly accurate evaluations of RAAs under a variety of channel conditions that are more representative of scenarios under which devices are likely to be used than previously possible. Moreover, our portable traces and trace-processing engine seamlessly couple with different algorithms to provide for easy, portable, repeatable, and realistic evaluations.

Chapter 5

T-SIMn: Trace-based Simulation of 802.11n Networks

5.1 Introduction

This chapter describes a new trace collection methodology that allows the 802.11n trace-based framework (T-SIMn) to handle devices that support many transmission rates. The initial part of the T-SIMn project [17] was done jointly with Andrew heard [50]. We first present an overview of this project required to the understand the new trace collection methodology (i.e., inferred trace collection) which is the main contribution of this chapter. In the joint project [17], we have developed a trace-based evaluation framework for the 802.11n standard. The fundamental goal of this project, namely achieving fair, repeatable, and realistic comparison of multiple competing algorithms, is similar to that of our T-RATE project. However, due to the differences in the physical and MAC layer of the 802.11g and 802.11n standards, the transition to 11n required several new significant research contributions. To start with, we had to completely rewrite the simulator component of our framework to accommodate the new features introduced in the 802.11n standard including MAC-layer frame aggregation. In T-RATE, our focus is on the performance evaluation of rate adaptation algorithms, while in our 802.11n trace-based evaluation framework, which we call T-SIMn, we broaden our focus to include link adaptation and frame aggregation algorithms. In this joint project, we design T-SIMn and evaluate it using an iPhone, which is representative of single-antenna devices such as many cell phones and tablets, which account for the vast majority of WiFi devices in use today. The reason we limited our evaluation to a device that supports 1 spatial stream is the challenges caused by many transmission rates and the MAC-layer frame aggregation. We now explain these challenges.

In contrast with 802.11g (which supports only 8 transmission rates), the 802.11n standard supports up to 128 rates. The trace collection methodology used in T-RATE and T-SIMn samples all transmission rates in a round-robin ordering. The idea is to attempt to measure the error rate of all transmission rates "simultaneously" (or in this case as closely in time as possible) by obtaining enough samples over the channel coherence window. This works easily for 802.11g networks with only 8 transmission rates but it becomes very challenging in 802.11n networks (when MAC-layer frame aggregation is used), since we might not be able to sample all transmission rates accurately within the channel coherence time, due to the large number of rates and the time required to send aggregated frames.

The 802.11n standard introduced MAC-layer frame aggregation, which allows longer frames to be transmitted by combining multiple subframes into a larger physical layer frame. Frame aggregation is a very important MAC-layer feature which significantly improves the efficiently of the 802.11n networks. Due to MAC layer overheads, without frame aggregation, the throughput of 802.11n networks cannot go above 50 Mbps regardless of the physical-layer bit rate (refer to Section 2.1.2 for more details). As a result, it is crucial to support frame aggregation in T-SIMn. However, frame aggregation makes trace collection even more challenging. The longer physical-layer frames (compared with no frame aggregation) further limits the number of samples that can be collected during the channel coherence window. We describe these challenges in detail in Section 5.4.

In the joint project with Andrew Heard [17], we consider only single-antenna devices which support 32 transmission rates. In this chapter, we propose and evaluate a novel trace collection methodology that enables the T-SIMn framework to handle devices that support many more transmission rates (we show that it works with up to 96 rates). Before presenting this methodology, we provide an overview of the T-SIMn framework [17]. Note that fundamental design goals and ideas in the T-RATE and T-SIMn frameworks are similar. However, we provide an overview for completeness and to point out the many technical differences between these frameworks.

5.2 Overview of the T-SIMn Framework

The main goal of T-SIMn is to achieve repeatability and realism when evaluating the performance of 802.11n networks. To achieve this goal, T-SIMn records information related to channel conditions that affect throughput in a trace and then uses this trace to simulate different 802.11n optimization algorithms such as rate adaptation and frame aggregation.

As a result, T-SIMn can be used to achieve repeatability by using an identical trace to evaluate different algorithms. In addition, it achieves realism since T-SIMn *relies on traces that are subject to and include information related to actual channel conditions rather than using wireless channel models*, which are known to lack realism [65, 74], as discussed in Section 2.2.2.

To simulate 802.11n networks with high fidelity, we need to accurately compute the transmission time of a frame and consider all factors that can affect throughput. Computing the transmission time for a frame is a relatively easy task, and is done very accurately in our simulator by using timing information available in the 802.11n standard. Environmental factors may affect 802.11 channel access (i.e., CSMA/CA) and channel error rate. If a WiFi or non-WiFi device operating at the same frequency is active during channel sensing, it forces a sender to back off and therefore limits the number of frames that can be sent. If WiFi or non-WiFi devices interfere with the receiver, the channel error rate may increase. In T-SIMn, to accurately simulate the time required for frame transmission we need to determine: the delay (overhead) imposed by channel sensing (i.e., CSMA/CA); how long it takes to transmit a frame, which depends on the transmission rate and the number of subframes in an aggregated frame, and must include ACK reception and DCF mandatory wait times. We also record whether or not the transmitted frame is received correctly.

T-SIMn uses two phases to simulate 802.11n networks. The first phase is trace collection, where a log containing the data necessary for accurately simulating an 802.11n experiment is collected. The second phase is simulation, where the trace is used to determine frame fates, transmission delays and throughput for any rate and any number of subframes at any point in time. This is required to compare link adaptation and frame aggregation algorithms. In our joint project, we show [17] that carefully considering all factors that affect throughput such as 802.11n transmission features, channel access, and channel error rate is necessary to accurately simulate this standard. More specifically, we show that the accurate handling of 802.11n frame aggregation is a key to obtaining realistic and highly accurate results. This work demonstrates [17] that the simulator portion of the project (SIMn) accurately simulates these factors (by comparing the simulator results with empirical measurements). We demonstrate that the T-SIMn framework can be used with single-antenna devices, which covers devices like most smartphones and tablets. A complete description and evaluation of the T-SIMn framework can be found in [17] and [50].

5.3 Testbed

For the experiments conducted in this chapter, we utilize our testbed in an office environment as described in Section 3.2. We create an 802.11n AP using Hostapd on the desktop machine (configured to use a TP-Link TL-WDN4800 PCIe card) and collect traces while that system sends packets using our modified ath9k driver. A laptop, configured to use a TP-Link TL-WDN4200 (3x3:3) dual-band wireless N USB adapter, is utilized as a mobile client. Although the AP could be used as the *sender* or the *receiver*, we use the computer designated as the AP as the *sender* in all experiments. The major advantage of this approach is that there are fewer requirements imposed on the *receiver*, which does not need to be capable of creating an AP. Most importantly, the *receiver* does not need to use a modified ath9k driver and, as a result, can be any 802.11n-capable device that runs Iperf. We create a network between the AP (*sender*) and a client (i.e., the laptop), which acts as a receiver. To collect a trace, the *sender* saturates the link by sending as many 1,470 byte UDP frames as possible using Iperf.

5.4 Trace Collection with Many Transmission Rates

As detailed previously, in order to increase the efficiency of the 802.11n protocol, a frame aggregation mechanism was introduced to combine multiple frames into a large physical frame. By utilizing frame aggregation, channel sensing and receiving an acknowledgment for each frame is amortized. Instead, channel sensing is done only once per aggregated frame and one block-acknowledgment is received that reports the fate of all subframes. As a result, the MAC layer efficiency increases significantly and much higher throughput can be achieved compared to when frame aggregation is not utilized. Figure 5.1 illustrates the structure of an Aggregated MAC Protocol Data Unit (A-MPDU), which is one of the two frame aggregation mechanisms supported by 802.11n. MAC Service Data Unit Aggregation (A-MSDU) is not widely supported by 802.11n cards [42], therefore, we study A-MPDUs. Note that all subframes in an aggregated frame are sent to the same destination.

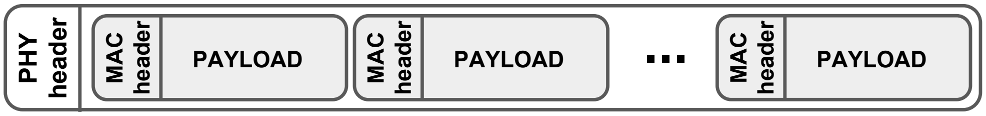


Figure 5.1: The structure of Aggregated MAC Protocol Data Unit (A-MPDU)

Although the MPDUs in an A-MPDU have their own MAC header and CRC, Byeon et al. [32] found that the FER of MPDUs depend on their location in an A-MPDU. Since hardware calibrations are done only once at the beginning of each frame, sub-frames at the end of an A-MPDU are more likely to fail due to an inaccurate estimation of the channel conditions. As a result, it is not accurate to consider a common FER for all MPDUs; instead, a per sub-frame error rate should be measured. In our project [17] with Andrew Heard, we designed the *direct measurement technique*, which measures the per-MPDU error rate by sending the longest possible A-MPDU for each transmission rate. All transmission rates are sampled using a round-robin ordering. If we sample all transmission rates within the channel coherence window, all rates experience the same channel conditions. The number of transmission rates determines the time required to complete a round of sampling and the number of samples that can be collected within the channel coherence window. For 802.11n devices that support many transmission rates, the number of samples collected by the direct measurement technique might be too low to accurately measure the error rate. We refer to this as the *lack of samples problem* for trace collection.

In that work [17], we show that the direct measurement technique can be used for devices that support 32 transmission rates. However, this technique may not be accurate for devices that support more transmission rates. For instance, for a device that supports up to 3 spatial streams (i.e., 96 rates), our measurements have determined that each round of sampling takes about 300 ms to complete when using the direct measurement technique. Since the channel coherence time in the 5 GHz spectrum is about 100 ms [22], the coherence window is 200 ms (i.e., 100 ms before and after a particular point in time). As a result, each transmission rate only has at most one (per sub-index) sample in a coherence window, which may not be enough to capture fast-changing channel conditions. While we have shown the direct measurement methodology is accurate for 32 rates, the maximum number of rates that this methodology can support depends on the required accuracy for a particular study and is a topic of future work.

However, in the rest of this section, we study the lack of samples problem with the goal of alleviating the shortcomings of the direct measurement technique. We show that a relatively intuitive approach that one might expect to work cannot be used, and argue that more sophisticated techniques are required. In Section 5.5, we propose a novel methodology that can be used to obtain a sufficient number of samples with devices supporting a larger number of rates (i.e., three antennas with 96 rates).

5.4.1 Dynamic Transmission Rate Elimination

One approach to increasing the number of samples within the coherence window is to dynamically eliminate "trivially predictable" transmission rates when collecting a trace using a round-robin ordering. The intuition is that if a particular transmission rate fails to successfully transmit a frame, faster transmission rates (which use less redundancy) will not be able to transmit the frame either (i.e., the FER will be 1). Similarly, if a particular transmission rate is successful, sampling slower transmission rates is not necessary since they are even more robust and will be successful (i.e., the FER will be 0). Therefore, sampling these transmission rates with known expected FER is wasteful and sampling all rates might be considered unnecessary and "oversampling" some rates. As a result, the round-robin trace collection mechanism might be able to skip many rates and collect a larger number of more useful samples by dynamically eliminating some rates.

This heuristic could solve the lack of samples problem if a large enough portion of the rates experience an error rate sufficiently close to 0 or 1. These rates that are trivially predictable over some short period of time (i.e., *trivial rates*) can be temporarily and dynamically excluded or included by the heuristic. Note that the set of trivial rates might change over time as the channel conditions change. To evaluate the potential of such a heuristic, we initially designed an experiment to examine the number of *non-trivial rates*, with FER$\neq$0 or FER$\neq$1, over a one second window. However, we soften those requirements to reduce the number of non-trivial rates by considering only rates with an FER between 0.05 and 0.95 as non-trivial. If the intuition behind the heuristic is correct, a small portion of rates will be non-trivial and the rest of them will be trivial and can therefore be eliminated from sampling.

In this experiment, a laptop (i.e., receiver) equipped with the wireless N USB adapter, which supports 96 transmission rates, is carried at walking speeds for 15 minutes in the office environment as described in Section 5.3. There exists no line-of-sight between the AP and client for most of the experiment, because the signal is blocked by obstacles such as metal cabinets, cubicle partitions, and walls. We use a 2.4 GHz channel, which is exposed to WiFi and non-WiFi interference. The distance between the AP and client ranges from 1 meter to about 20 meters. In this experiment, all 96 transmission rates are sampled using a round-robin ordering without frame aggregation. Frame aggregation is not used in this experiment to increase the number of samples collected for each transmission rate.

Figure 5.2 shows the number of non-trivial rates which *cannot be eliminated* by the heuristic over the 15 minute experiment. The figure indicates that there are several times when the number of non-trivial rates is more than 70. Therefore, at those times the heuristic can eliminate less than 26 of the 96 transmission rates. Sampling 70 transmission rates using frame aggregation takes over 200 ms to complete. As a result, each rate has at most one sample over the coherence window (i.e., 200 ms) and this heuristic, based on dynamic rate elimination, will also suffer from a lack of samples. Therefore, we dismiss this as a potential trace collection methodology. We believe that the reason for these counterintuitive results is WiFi and non-WiFi interference. We believe that interference might be strong enough to change the FER of most transmission rates regardless of the amount of redundancy encoded in the frames.

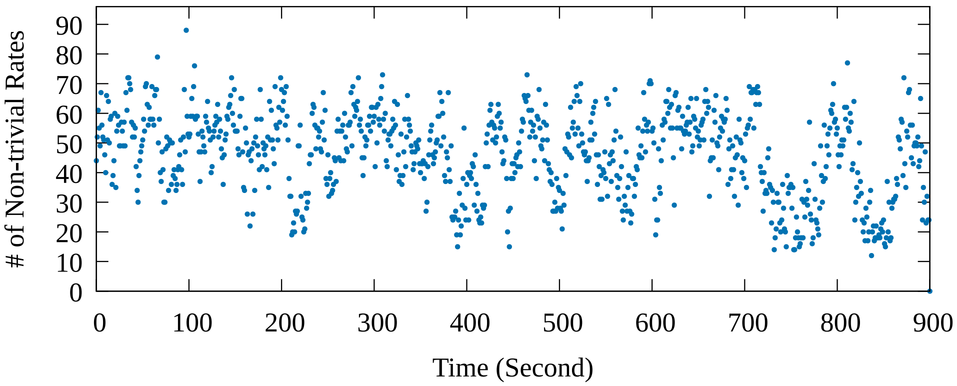


Figure 5.2: Mobile 2.4 GHz

To support this claim, we repeat the experiment in the 5 GHz spectrum on a channel with no WiFi or non-WiFi interference and present the results in Figure 5.3. Note that except the spectrum band, everything else in this experiment is unchanged. Figure 5.3 indicates that when the interference is eliminated from the experiment, the number of non-trivial rates decreases significantly.

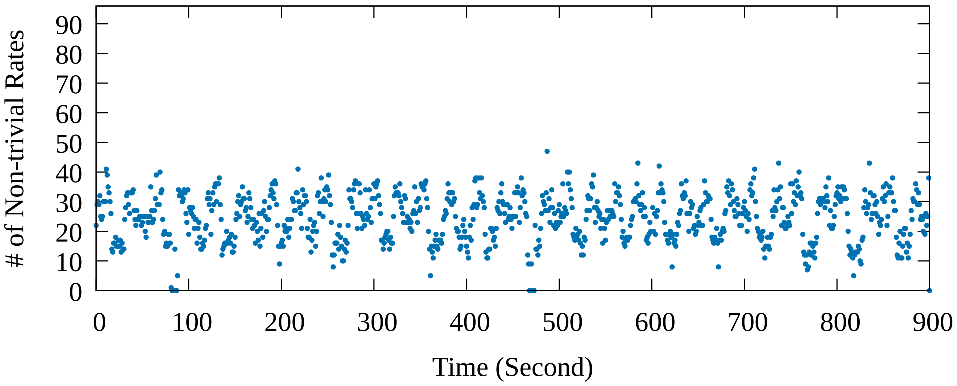


Figure 5.3: Mobile 5 GHz

In summary, the rate elimination heuristic might be able to skip many transmission rates when there is no interference. However, our goal is to collect traces under a variety of channel conditions including those that are affected by different sources of interference. As a result, this heuristic cannot be used for our purpose. In the next section, we characterize the MPDU delivery ratio patterns in an A-MPDU and then use the characterization to design a methodology for collecting traces with many transmission rates.

5.5 Inferred Measurement Technique

The Frame Error Rate (FER) is not identical for the MPDUs in an aggregated frame. For this reason, the direct measurement methodology [17] transmits the largest possible A-MPDU, during trace collection, to measure the FER of all MPDU indexes. However, using frame aggregation during trace collection leads to the lack of samples problem in 802.11n networks with many transmission rates as explained in Section 5.4. We now describe our new inferred measurement methodology, which avoids frame aggregation for trace collection to increase the number of samples collected over the channel coherence window. Note that when frame aggregation is disabled, only the error rate of the first MPDU can be measured. Since during simulation an A-MPDU with any arbitrary length might be transmitted, the inferred measurement methodology needs to accurately estimate the error rate of the missing MPDUs. To explain the design of this methodology, we need to understand the characteristics of the MPDU delivery ratio (MDR) in an aggregated frame. Therefore, we first study the effect of carrier frequency and the speed of movement on the changes to the MDR in an A-MPDU, then, we describe the design of the inferred measurement technique.

5.5.1 Changes in the MPDU Delivery Ratio

Byeon et al. [32] report that if channel conditions change rapidly (for instance when the sender or receiver is moving), subframes at the end of an A-MPDU experience higher error rates than those closer to the header of the frame. In addition, they show that higher speeds of movement intensify this phenomenon. In order to accurately estimate the MDR of subframes in an A-MPDU, we need to characterize and understand how the MDR of different subframes change within an A-MPDU. *In this section, the MDR of subframe i is denoted by MDR_i*

We design a mobile experiment where a laptop with the wireless N USB adapter is carried at different speeds from a very slow to a fast walking speed in the office environment described in Section 5.3. We divide the experiment into three parts, namely slow, normal, and fast walking speeds. Figure 5.4 shows the relative MPDU delivery ratio (relative MDR) for the physical rate of 117 Mbps for the three walking speeds. We present the relative

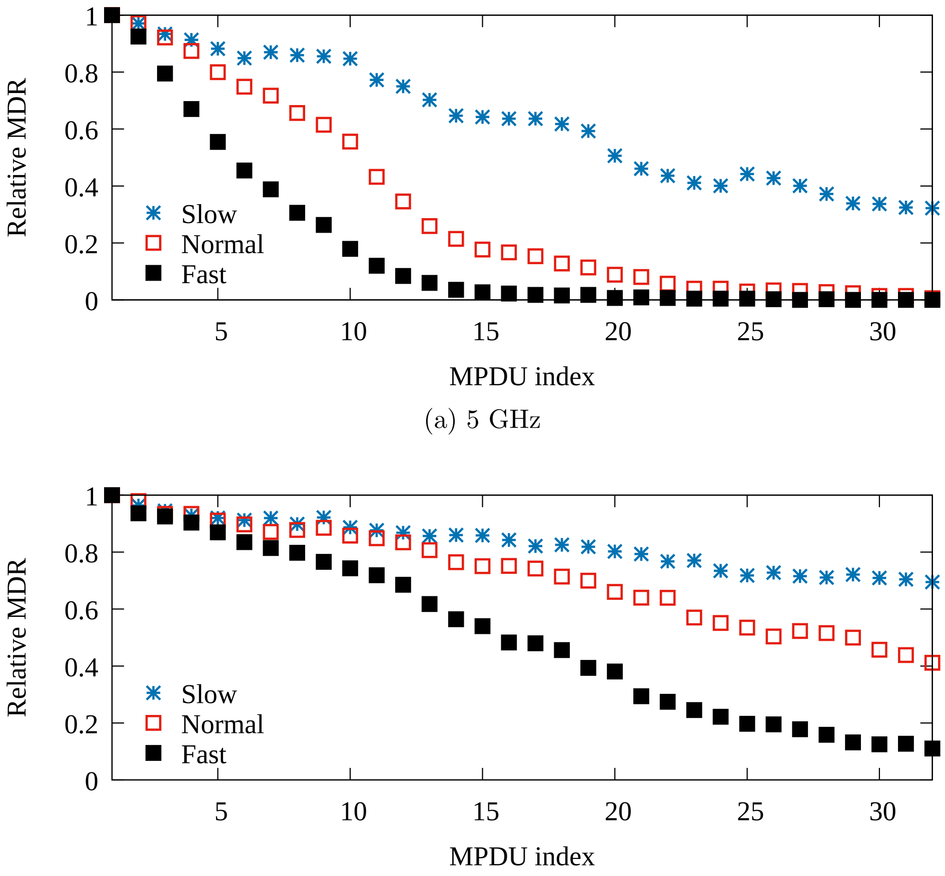
MDR for an easier comparison of the 2.4 and 5 GHz experiments. The relative MDR is computed by dividing the MDR of each index by the MDR of the first MPDU. For instance, the relative MDR of subframe i is $MDRi/MDR_1$. We can observe that in both spectrums, as the speed increases the delivery ratio drops more sharply due to more rapid changes in the channel conditions. This is because the initial calibrations done on the preamble are less representative of the channel state as the time since the preamble increases. The results show that the speed has a significant role in the decrease of the MDR, and it is more prominent in the 5 GHz spectrum. For instance, let's consider MPDUs with index 16, which are in the middle of the aggregated frames. In the 2.4 GHz spectrum, the relative MDR at the slow and fast speeds are 0.84 and 0.48, respectively. The relative MDR had dropped by 43%, while in the 5 GHz band, the relative MDR is 0.64 and 0.02 which is a 97% drop. Byeon et al. [32] also observed that the speed of movement intensifies the MDR decline phenomenon. However, their measurements are conducted in the 5 GHz spectrum only. Our results show that the 2.4 and 5 GHz frequency bands impact this phenomenon differently.

One reason for the faster decrease of the MPDU delivery ratio in the 5 GHz spectrum is explained by the Doppler shift. The following formula shows how the Doppler shift is directly related to the transmission frequency (f_0):

$$\Delta f = (\frac{\Delta v}{c})f_0 \tag{5.1}$$

where Δf is the frequency shift, Δv is the relative speed of the sender and receiver, and c is the speed of light. As Equation 5.1 indicates, for a given relative speed Δv, the frequency shift in the 5 GHz spectrum is almost twice that of the 2.4 GHz spectrum. The frequency shift due to the Doppler effect causes inter-subcarrier interference in OFDM communication systems which consequently increases the bit error rate leading to a higher FER. As a result, in the 5 GHz spectrum, the later MPDUs in an aggregated frame are more likely to observe a change in the channel conditions (i.e., frequency shift) due to a stronger Doppler effect.

Observation: *We observe that the rate at which the MDR declines is related to the speed of movement. If we can quantify the channel dynamics caused by the movement of the sender, receiver, or nearby objects, we might be able to estimate the MDR of all other MPDUs from the MDR of the first MPDU and the channel dynamics metric.*



(a) 5 GHz

(b) 2.4 GHz

Figure 5.4: Impact of speed and spectrum on MPDU delivery ratio

The inferred measurement methodology (explained later in this section) utilizes this idea by turning off frame aggregation during trace collection (i.e., measuring the MDR of the first MPDU only) and estimating the MDR of *missing* MPDUs. By not using frame aggregation, the inferred measurement methodology can collect more samples during the same time interval than the direct measurement technique. We show that the inferred measurement methodology is able to support 802.11n networks with many transmission rates (in this case up to 96). We next describe a novel technique for quantifying the channel dynamics, then we present the design of the inferred measurement methodology.

5.5.2 Channel Dynamics Indicator (CDI)

We have shown that the decline of the MPDU delivery ratio in an aggregated frame depends on the speed of movement. We now propose measuring the channel dynamics (which changes with the speed of movement) by using changes in the RSSI of ACKs. Note that we do not use RSSI to estimate the frame error rate, instead, we infer the channel dynamics from the changes in the measured RSSI.

We require a statistical measure that reflects the channel dynamics. In other words, this metric should change with the speed or amount of movement of the sender, receiver, and the surrounding objects. We will use this measure to estimate how the MDR declines as the subframe index increases (i.e., how later frames within the A-MPDU have higher error rates). We found that the variance of the difference between consecutive RSSI measurements, which we call *Channel Dynamics Indicator (CDI)*, is a suitable measure. Studying other statistical measures will be the subject of future work.

Figure 5.5 shows the channel dynamics indicator for the experiment that we described in Section 5.5.1. In this experiment, the speed of movement increases gradually during the experiment, therefore, we expect the channel dynamics indicator to change in conjunction with the speed of movement. The graph shows that our metric correctly increases with the movement speed. At the beginning of the experiment, when the movement speed is very low, the CDI is fairly close to zero. As the speed increases, the CDI also increases to reflect the more variable channel conditions. In the next section, we show how this metric can be utilized to accurately estimate the MPDU delivery ratios.

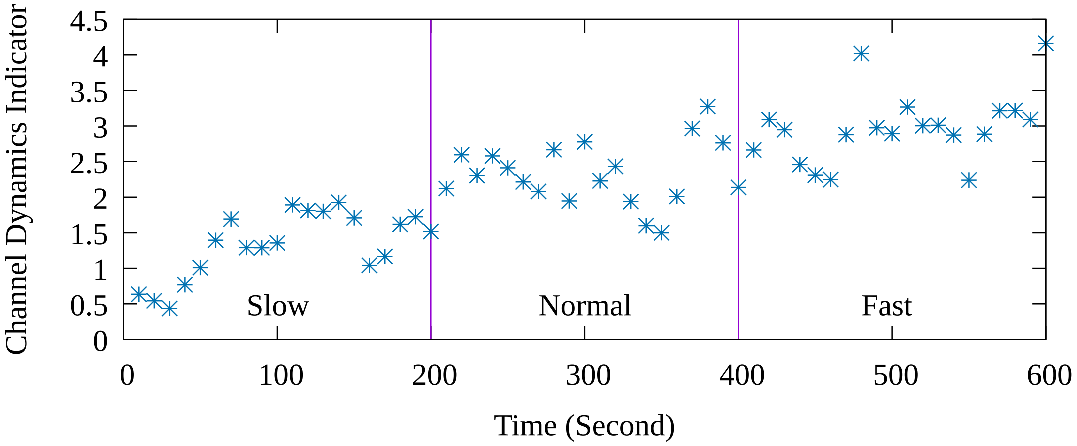


Figure 5.5: Channel dynamics indicator

5.5.3 Inferred Measurement Methodology

We observed that the channel dynamics indicator (CDI) can potentially be a good estimator for the fluctuations in the channel conditions caused by mobility. We have also shown that the MPDU delivery ratio decline in an A-MPDU depends on the speed of movement. As a result, we hypothesize that if the delivery ratio of the first MPDU of a given A-MPDU and the CDI can be determined, the delivery ratio of other MPDUs in that A-MPDU could be estimated. If this hypothesis turns out to be true, then we can collect traces without frame aggregation and infer the delivery ratio of missing MPDUs from non-aggregated frames (which correspond to the first MPDU in an A-MPDU) and CDI. In other words, we can increase the number of samples by not using frame aggregation, but still estimate the delivery ratio of all MPDUs, emulating the case where traces are collected using frame aggregation.

We now show that for the traces studied, the channel dynamics indicator and the delivery ratio of the first MPDU (MDR_1) can accurately estimate the delivery ratio of other MPDUs (MDR_i), where i is the MPDU index. In general, we observe that *for each transmission rate,* there exists a function f_i that maps the CDI and MDR_1 to MDR_i:

$$\forall i \in \{1, 2, ..., MAX\} \ \ \exists f_i : (MDR_1, CDI) \rightarrow MDR_i \tag{5.2}$$

where MAX is the maximum number of MPDUs that can be transmitted in 4 ms. Note that the ath9k driver further limits the number of MPDUs to 32. The 802.11n block acknowledgment mechanism allows up to 64 frames in a single transmission. However, the ath9k driver does not aggregate 64 subframes in an A-MPDU but instead creates two 32-subframe A-MPDUs. This way when the transmission of the first A-MPDU is finished the hardware can immediately start to transmit the second A-MPDU. While this frame is being transmitted, the driver has time to create the next 32-subframe A-MPDU. This is in contrast to using a 64-subframe A-MPDU which would require the NIC to wait for the driver to create the next A-MPDU which would waste useful transmission time and potentially decrease throughout.

Figure 5.6 illustrates how the channel dynamics indicator (CDI) and the MDR of the first MPDU (MDR_1) is related to MDR_{16} (bottom) and MDR_{32} (top), for the experiment described in Section 5.5.1. In this experiment, a laptop is carried at various speeds for 15 minutes in the office environment. The results for the physical layer transmission rate of 117 Mbps is presented in the graph. Similar results were observed for other transmission rates.

As depicted in the figure, when the channel dynamics indicator increases (i.e., corre-

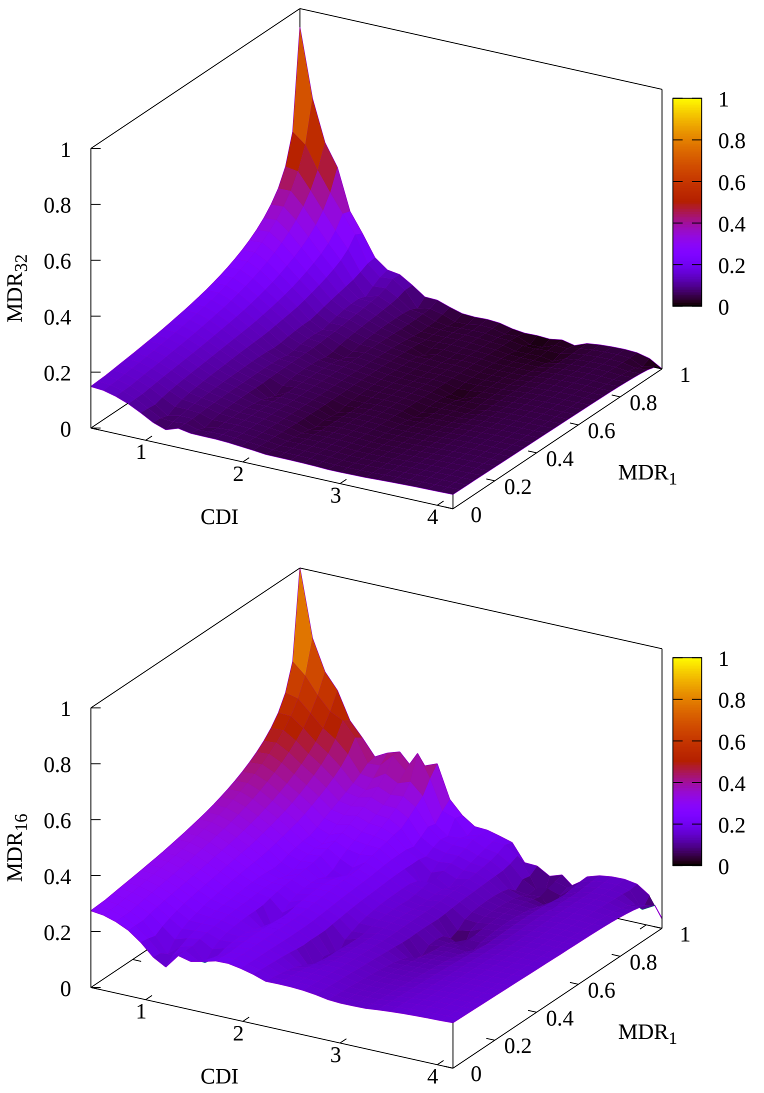


Figure 5.6: Relationship of MDR$_1$ and CDI with MDR$_x$

sponding to faster movement) the MPDU delivery ratios of MPDUs 16 and 32 decrease for any given MDR$_1$. This is consistent with our findings in Section 5.5.1, where we observed that the MDR drops more rapidly as the speed increases. MDR$_{16}$ and MDR$_{32}$ are relatively high when the receiver moves very slowly (i.e., CDI < 1) and MDR$_1$ is high (e.g., due to good channel conditions for the given modulation and coding scheme). On the other hand, when the receiver moves very fast (i.e., CDI $= 4$), MDR$_{32}$ is zero and MDR$_{16}$ is very low, regardless of MDR$_1$. Note that, for low values of CDI and MDR$_1 = 0$, MDR$_{32}$ and MDR$_{16}$

are greater than MDR_1 in the plot. This is simply an artifact of extrapolation for creating a surface from a series of points. As expected, MDR_{16} is generally higher than MDR_{32}, since MPDUs closer to the frame header have a higher delivery ratio.

Methodology Overview

In order to enable trace collection using devices that support many transmission rates, we have designed and implemented the inferred measurement methodology. The overview of our methodology (illustrated in Figure 5.9) is as follows:

1. A *calibration trace* is collected using frame aggregation, using the direct measurement technique explained in Section 5.4.1.

   This trace is used to find the relationships between the CDI and MDRs. To compute the MDRs, we use 10-second windows to obtain enough samples to accurately measure the MDRs (i.e., up to 30 samples inside the averaging window). Note that the goal is not to measure the MDRs at a particular time, but rather to compute the relationships between the CDI and MDRs. Therefore, in this case, the averaging window does not need to be smaller than or equal to the channel coherence time.

2. A *main trace* is collected without frame aggregation (i.e., $\texttt{FA}_{\texttt{COL}}\texttt{=1}$).

   This trace is called the main trace because it is a trace of the experiment and environment that will be later used in the simulation (i.e., it is the experiment that will be simulated). The MDR_1 and CDI information from this trace along with the mapping functions (obtained from the calibration trace) are used in the next step to find the fate of missing MPDUs.

3. For each transmission rate and subframe index i, the calibration trace gives us a set of 3-tuples, MDR_1, MDR_i, and CDI, which creates a 3d surface (i.e., *mapping function*) as previously shown in Figure 5.6. We use multi-variable regression, where MDR_1 and CDI are the estimator variables and MDR_i is the estimated variable, to determine this mapping function (details are presented later in this section). This procedure is repeated for all transmission rates and subframe indexes.

4. In order to estimate the fate of missing MPDUs in the main trace, we use the following procedure: for a given packet in the main trace at time t and transmission rate R, we consider a time window centered at time t and compute the average MDR of MPDU 1 (i.e., MDR_1) from the packets with rate R in this window. The time window we use

in this study is 200 ms to roughly approximate the channel coherence time. Similarly, we compute the CDI from the RSSI of all ACKs, regardless of the transmission rate, in this window. Note that the RSSI is a property of an electromagnetic signal not the transmission rate. Then, we use the computed MDR_1 and CDI as input into the mapping functions to estimate the expected fate of the MPDUs missing from the main trace. This procedure is repeated for all frames in the main trace and the results are stored in the *generated trace*. The generated trace is similar to the output of the direct measurement methodology, except that in this case the MPDU fates are estimated (or *inferred* by applying the models obtained from the calibration trace to the packets obtained in the main trace). This generated trace is used by T-SIMn to conduct performance evaluations.

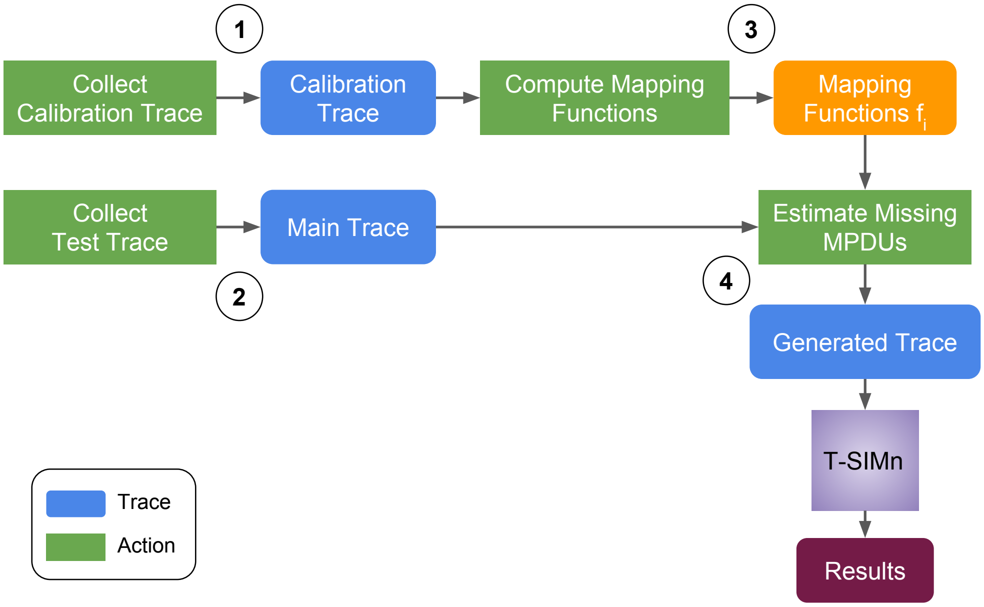


Figure 5.7: Inferred measurement methodology

Note that both calibration and main traces include the RSSI of ACKs needed to compute the CDI. In the inferred measurement technique, the calibration trace is only used to find the mapping functions. These mapping functions are used to estimate the fate of missing MPDUs in the main trace and can be used with different main traces in that environment, therefore, they do not need to be recalculated for each new main trace. Then, the main trace is actually used for performance evaluation. The advantage of this methodology is that the main trace contains many more frames per unit of time than can be collected using the direct measurement methodology. For example, in the inferred measurement methodology, if all 96 transmission rates (in a 3x3 MIMO system) are sampled using a round-robin sampling technique, we obtain one sample every 43 ms for each transmission rate (i.e., about 5 samples over a 200 ms coherence window). This is a large gain in terms of the number of samples collected when compared with trace collection using frame aggregation (i.e., direct measurement) where we sample each rate every 300 ms (i.e., less than one sample over a 200 ms coherence window). As a result, for devices that support 3 spatial streams (i.e., 96 transmission rates), the inferred measurement methodology increases the number of collected samples to 5 samples over the 200 ms coherence window, compared with only 0.7 samples that would be obtained if we were to use the direct measurement methodology. While 5 samples may seem low, in Section 5.5.4, we show that the inferred measurement methodology can accurately estimate the fate of missing MPDUs.

Equation 5.3 shows the regression model used in this study to obtain the mapping functions from the calibration trace. The model that we use contains two predictor terms MDR_1 and CDI. In addition, it has quadratic and interaction terms. We now explain why we include these terms in the model. Figure 5.6 shows that the relation between the predictor terms (i.e., MDR_1 and CDI) is curvilinear with MDR_i. As a result, we add the quadratic terms to the regression model. For the traces used in this study, we did not find higher order terms necessary in the regression model. In addition, as can be seen in Figure 5.6, there are interactions between the predictor terms. For example, when $CDI = 4$, the relation between MDR_1 and MDR_{32} is almost constant (i.e., $MDR_{32} \approx 0$). On the other hand, when $CDI = 0$, MDR_{32} is a curvilinear function of MDR_1. This observation confirms the existence of interactions between the predictors. As a result, we add the $MDR_1 * CDI$ term to Equation 5.3 to model the interactions between MDR_1 and CDI. We will show that the regression model in Equation 5.3 works well for the purpose of this study, however, we plan to examine other possible models in future work.

$$\begin{aligned} MDR_i = \beta_0 &+ \beta_1 MDR_1 + \beta_2 CDI \\ &+ \beta_3 {MDR_1}^2 + \beta_4 CDI^2 \\ &+ \beta_5 (MDR_1 * CDI) \end{aligned} \tag{5.3}$$

5.5.4 Methodology Evaluation

To evaluate the efficacy of the inferred measurement methodology, one can compare the outcome of this technique with that of an actual experiment (i.e., the ground truth).

However, it requires repeatability between the trace collection and the ground truth experiment, and achieving repeatability is very difficult, especially in environments that involve mobility and interference. As noted previously, our goal is to be able to simulate environments with mobility and interference. Therefore, we design a novel technique, illustrated in Figure 5.8, to study the accuracy of the inferred measurement methodology. Instead of using the main trace, which contains samples for all rates, we collect a trace (called a test trace) for a given transmission rate by alternatingly sending frames with and without frame aggregation. The frames without frame aggregation serve as the input to the inferred measurement methodology, while the aggregated frames are used to determine ground truth. The frames without frame aggregation are down-sampled before being fed to the inferred measurement methodology to match the number of samples obtained normally when sampling all transmission rates (i.e., matching the number of usable packets that would be available in the main trace). With this approach, we run an experiment in a way that gives us the ground truth data and the input to the inferred measurement methodology simultaneously. The inferred measurement methodology should be able to use only those non-aggregated frames to accurately estimate the fate of each MPDU in aggregated frames. Remember that with the inferred measurement methodology, in order to collect the main trace, we disable frame aggregation to increase the number of samples obtained per unit time. Then, we infer the fate of aggregated frames using the mapping functions (obtained from the calibration trace) and the main trace (which contains non-aggregated frames only). We now explain the procedure for evaluating the inferred measurement methodology step by step as illustrated in Figure 5.8. Note that some steps are identical to what is done in the inferred measurement methodology (marked as [IDENTICAL]).

1. Collect a calibration trace as explained before (i.e., round-robin all transmission rates with $\texttt{FA}_{\texttt{COL}}$=`MAX`). [IDENTICAL]

2. Collect a test trace for a given transmission rate R. Frame aggregation is turned on and off alternatingly (i.e., $\texttt{FA}_{\texttt{COL}}$=`MAX` and $\texttt{FA}_{\texttt{COL}}$=`1`) for each physical frame. Aggregated frames are extracted from the test trace and are stored in the *ground truth trace*, while non-aggregated frames are stored in the *single-rate main trace*. As opposed to the main trace, which is normally collected when using the inferred measurement methodology (containing the fate of all rates), this single-rate main trace contains the fate of rate R only. Note that for the purpose of evaluating the inferred measurement methodology, instead of sampling all rates, we sample only one rate so that we can also collect the ground truth data (i.e., aggregated frames) simultaneously.

3. For each transmission rate, mapping functions f_i are computed using the regression

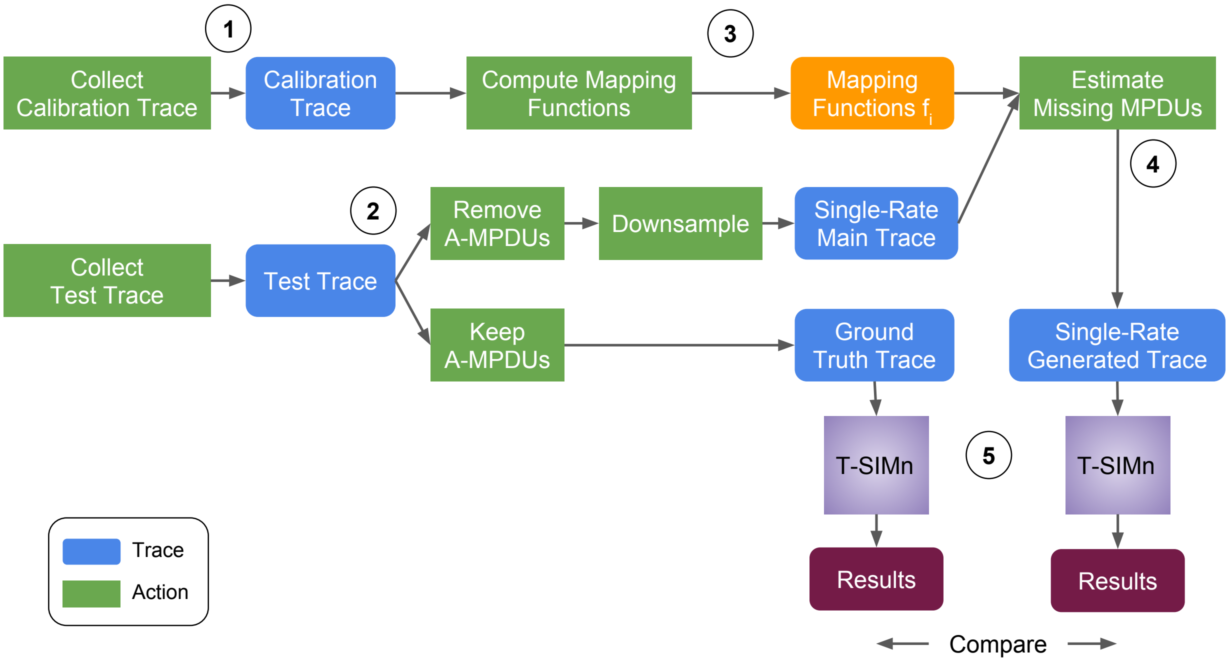


Figure 5.8: Inferred measurement methodology evaluation procedure

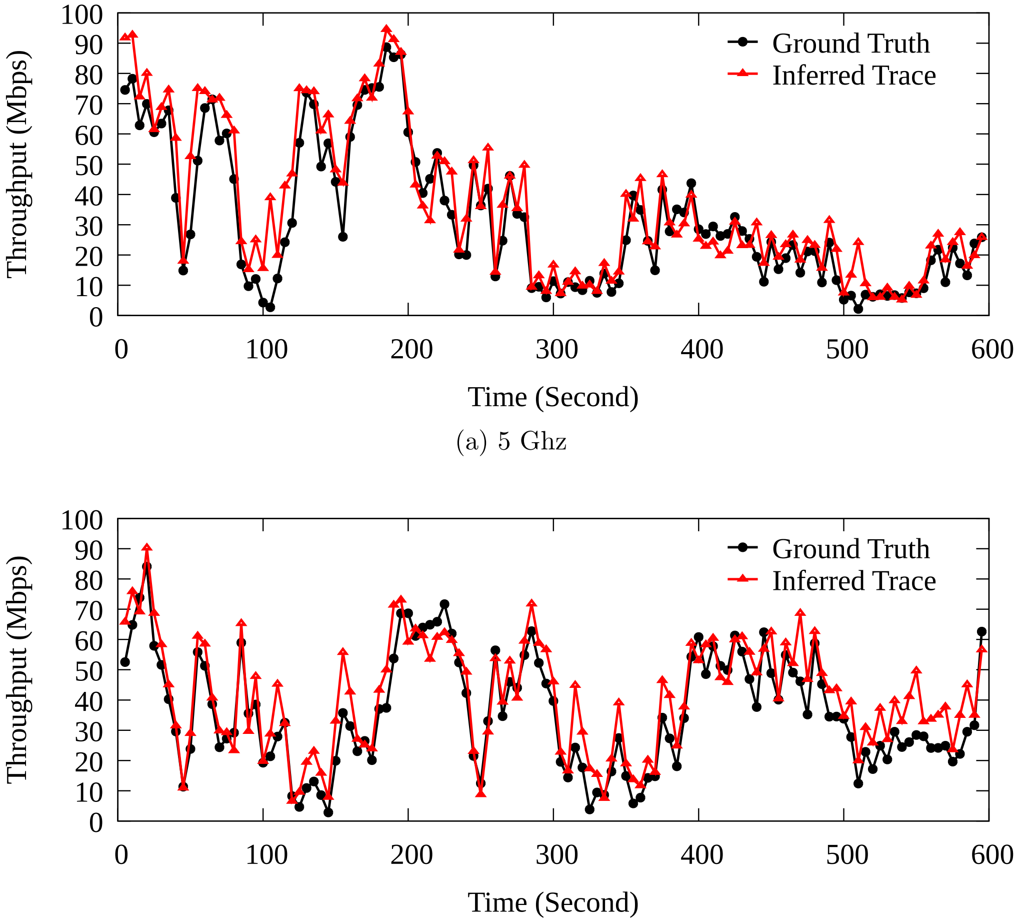
model in Equation 5.3 based on the calibration trace MDRs and CDI data. [IDENTICAL]

4. To estimate the MDR of missing MPDUs, the MDR_1 and CDI from the single-rate main trace are used as input into the mapping functions, which are used to produce the *single-rate generated trace*.

5. The single-rate generated trace and ground truth traces are fed separately to T-SIMn. T-SIMn is configured to saturate the link with a UDP stream using the constant physical layer rate R. We compare the throughput obtained from the generated and ground truth traces. If the inferred measurement technique is able to accurately estimate the MDR for A-MPDUs, then the obtained throughputs should match.

To evaluate the inferred measurement methodology using this procedure, we use a laptop with the wireless N USB adapter (which supports 96 transmission rates) as the mobile receiver in the office environment as described in Section 5.3. The laptop (i.e., receiver) is carried from a very slow to a very fast walking speed for 10 minutes to collect the calibration trace and once more to collect the test trace. The traces are collected at the sender, which is the stationary access point. In this experiment, there is a two week gap between the collection of the calibration and test traces to test the robustness of the calibration trace.

Figure 5.9 shows the throughput obtained from T-SIMn for the generated trace, produced by the inferred measurement methodology and the ground truth trace for experiments conducted using 2.4 and 5 GHz spectrums. Every point on the plots shows the average throughput computed over a 5-second window. The figures show that despite the highly dynamic environments, the throughput obtained from the inference methodology closely matches the ground truth data.



(a) 5 Ghz

(b) 2.4 GHz

Figure 5.9: Inferred measurement methodology evaluation

Software retransmission is disabled in T-SIMn in this experiment so that the block ACK window advancement does not limit the length of the transmitted A-MPDUs. As a result, the simulator always transmits full-length A-MPDUs to ensure that the estimated fates of all 32 MPDUs are used in the evaluation. Despite the two-week time gap between the collection of the calibration and test traces, and possible environment changes, the inferred measurement methodology produces accurate results. We speculate that if the calibration and main traces are collected more closely in time, more accurate results may be possible. We recommend that calibration traces include the same or bigger range of speeds of movement as the main trace so that a better fit can be achieved when performing the regression to compute the mapping functions. In future work, we intend to study if calibration is required for different environments.

Note that a particular transmission rate (i.e., 117 Mbps) is used in the *evaluation of the inferred measurement methodology*. This rate was chosen because we believe it would be a challenging choice for estimating the fate of missing MPDUs. To see if this is in fact a challenging choice, we consider the number of *non-trivial rates* (as introduced in Section 5.4.1). If a rate is *trivial* (i.e., the FER is almost 0 or 1) for most of the time during an experiment estimating missing MPDUs for this rate is a fairly easy task, since the FER of missing MPDUs is most likely 0 or 1. On the other hand, estimating the fate of missing MPDUs for a *non-trivial* rate is much more challenging.

Non-trivial FER ratio: *The ratio of the FER measurements for a transmission rate with the FER between 0.05 and 0.95 to the total number of FER measurements for that rate during an experiment.*

We consider a rate to be non-trivial at a given time, if the FER of this rate is between 0.05 and 0.95 over a one-second window centered at that time. To quantify how challenging transmission rates are for the purpose of evaluating our new methodology, we compute the ratio of time each rate is non-trivial. For instance, if the FER of a rate is between 0.05 and 0.95, for 40% of the measurements, the *non-trivial FER ratio* for this rate is 0.4 in that particular experiment. We compute the non-trivial FER ratio for all 96 rates in the calibration traces used in the 2.4 and 5 GHz experiments. In the 2.4 GHz trace, the rate we used in the evaluation (i.e., 117 Mbps) has the highest non-trivial FER ratio (i.e., 0.72). In the 5 GHz trace, this rate has a ratio of 0.81 and is the eighth highest non-trivial FER ratio (the highest ratio is 0.89). The high non-trivial FER ratios for the 117 Mbps rate make this rate a good candidate for the evaluation of our methodology, because estimating the missing MPDUs is challenging for this rate.

When collecting the test traces, we continuously monitor the FER to keep it in the non-trivial range. For instance, when the FER approaches 0.95, we start moving back towards the access point to lower the FER. As a result, the non-trivial FER ratio of the 117 Mbps rate in the 2.4 and 5 GHz test traces are 0.997 and 0.998, respectively. We believe that the rate used in this evaluation is a suitable choice for the purpose of this study due to the high non-trivial FER ratios and attempts made to keep the FER in the challenging non-trivial range.

Traces for the direct measurement methodology [17] are easy to collect and when applicable this methodology is preferred, however, it is applicable only to 802.11 networks with a smaller number of transmission rates. As mentioned before, in a joint project with Andrew Heard, we have shown that the direct measurement methodology can be used to collect traces with an iPhone that supports 32 transmission rates. To support networks with many more transmission rates such as MIMO 3x3:3 systems, we have designed the inferred measurement methodology. This technique utilizes the relationship between the MDR of MPDUs within an A-MPDU and the channel dynamics indicator to enable us to collect traces with frame aggregation disabled (thus increasing the number of samples collected per unit of time), and to infer the MDR for MPDUs that would have been sent with frame aggregation. Finally, we have utilized T-SIMn to evaluate the accuracy of the inferred measurement methodology. Our trace collection methodologies enable the T-SIMn framework to accurately measure the channel conditions in a variety of environments including those with WiFi and non-WiFi interference and mobility. When comparing the performance of two or more systems or algorithms using T-SIMn, utilizing traces guarantees that the competing alternatives are exposed to exactly the same channel conditions. Therefore, any differences observed in the performance are solely due to differences in those algorithms or systems and not due to differences in channel conditions. To evaluate the ability of traces to accurately capture fast-changing channel conditions, we demonstrate that the inferred measurement methodologies achieve a high degree of accuracy. It is worth noting that less accurate traces could still be used to evaluate competing alternatives as they would be compared using the same trace.

5.6 Chapter Summary

In this chapter, we provide an overview of a trace-based simulation framework for 802.11n networks, called T-SIMn [17]. Due to the limitations of the trace collection methodologies in [17], devices that support only one spatial stream can be handled. As a result, in this chapter, we design a novel trace collection methodology, called the inferred measurement methodology, to address this limitation. This methodology quantifies wireless channel fluctuations using the variation of the RSSI of received acknowledgment frames (called the channel dynamics indicator). Utilizing this metric, the inferred measurement methodology infers the fate of missing MPDUs when aggregation is avoided to increase the number of samples collected over the channel coherence window.

We show that the inferred measurement technique enables the framework to work with devices that support many transmission rates such as MIMO 3x3:3 systems. We believe that these important contributions have now improved T-SIMn to the point where it is a valuable tool for the realistic and repeatable performance evaluation of 802.11n networks.

Chapter 6

Characterization of 802.11n Networks

6.1 Introduction

Advancements in the 802.11 standard have made gigabit per second wireless communication possible by offering physical-layer transmission features such as denser modulations, channel bonding, and MIMO, in addition to MAC-layer frame aggregation. While these advancements help achieve higher data rates, efficient link adaptation in 802.11 networks is more challenging because of these options.

We use *rate configuration* (or simply *configuration*) to refer to a particular combination of physical-layer transmission features, such as the modulation and coding scheme, channel width, short/long guard interval (SGI/LGI), and the number of spatial streams. The 802.11g standard offers only 8 configurations. That number has increased to as many as 128 in 802.11n networks (our work examines the 96 configurations available on devices with 3 spatial streams) and up to 640 in 802.11ac networks. Because the 802.11 standard does not specify how to choose physical-layer transmission features, optimizing these choices is an active area of research. Examples of such research include channel bonding [36], rate adaptation [48,67], energy efficiency [115], QoS analysis [77], and STBC/SDM settings [67].

In rate adaptation studies, the combination of physical-layer features used for transmission is often chosen by sampling available configurations to determine their effective throughput. However, sampling can incur significant overhead [67] because probe packets are usually sent without frame aggregation. This conservative approach is used because probing often requires testing rates that may fail, and the failure of a large number of frames that have been aggregated can negatively impact application performance.

In this chapter, our hypothesis is that since several physical-layer transmission feature combinations (rate configurations) share common features (e.g., half use SGI and half use LGI), relationships may exist between the *average frame error rate (FER)* of different configurations. If it is possible to estimate the FER of one configuration from the measured FER of another configuration, algorithms that adapt configurations to changing channel conditions can be simpler and more effective. In this chapter, we first develop a methodology for characterizing the relationship between the FER of different configurations. Then we conduct experiments in a variety of settings and report on the relationships we observe.

6.2 Methodology

Our relationship analysis methodology consists of the following phases: (1) collect data, (2) compute the FER for each rate configuration, and (3) compute relationships between the FER of different configurations. As we demonstrate in Section 6.4, these steps can be used to characterize relationships between configurations.

6.2.1 Data Collection

To analyze the relationship between two rate configurations, the frame error rate (FER) of these rate configurations must be measured under identical channel conditions. Hence, previous work [66, 67] have conducted experiments at night without any movement in the environment using the 5 GHz band, while also ensuring that the only interference is controlled interference (e.g., co-channel and adjacent channel interference). Additionally, theses studies use an unmodified rate adaptation algorithm. Therefore, only those rate configurations chosen by the RAA are examined and may not properly cover all configurations.

In Section 2.2.1, we argue that repeating 802.11 experiments with identical channel conditions is difficult. More importantly, experiments from environments with only controlled interference and without mobility are unsuitable for understanding relationships between rate configurations in commonly encountered environments that include mobility and uncontrolled WiFi and non-WiFi interference. In contrast with previous approaches, we design an experimental methodology for collecting FER information in any environment that *also properly covers all configurations*. We now describe a methodology for collecting representative traces for our experiments in this chapter. This methodology is similar to the technique used in Chapter 4 to collect traces for an 802.11 trace-based simulator. Our methodology does not require repeatable channel conditions and can therefore be used in uncontrolled environments (including human movement and mobile devices operating in the 2.4 GHz band with WiFi and non-WiFi interference). With our technique, all configurations are sampled in a round-robin fashion. This process is continually repeated to collect information about changes in FERs over time.

Figure 6.1 shows a data collection example using a device with n rate configurations. Frames are sent with different configurations (denoted $R_1, R_2, ..., R_n$). The fate of each packet is denoted with 1 or 0, representing success or failure. Each sequence of n sampled configurations forms a round.

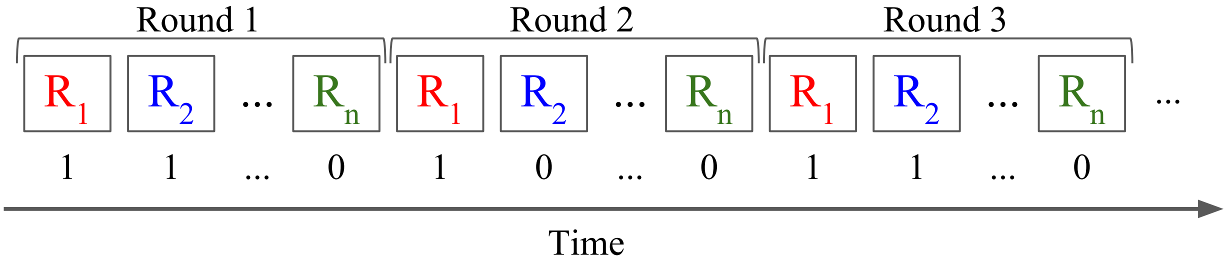


Figure 6.1: Round-robin data collection methodology

Since configurations in a round are subject to the same channel conditions (they are in the same channel coherence window), when interference does occur all configurations in a round experience the same conditions [16]. Changes, on average, impact each configuration equally. Since we are interested in physical-layer relationships, MAC-layer frame aggregation is disabled to increase the efficiency of the data collection mechanism. As explained in Section 5.5, each round of sampling when frame aggregation is not used takes about 43 ms, which is smaller than the channel coherence window in the 2.4 and 5 GHz frequency bands [98,101]. We implement round-robin sampling by modifying the device driver of the sending device used to collect data.

6.2.2 Frame Error Rate (FER) Computation

The frame error rate is computed for all n rates over multiple rounds of data collection, k. This is illustrated in Figure 6.2, where the average FER is denoted $\bar{E}$.

We now determine the number of packets required to compute the average FER. We use the following formula for calculating the minimum number of samples required to

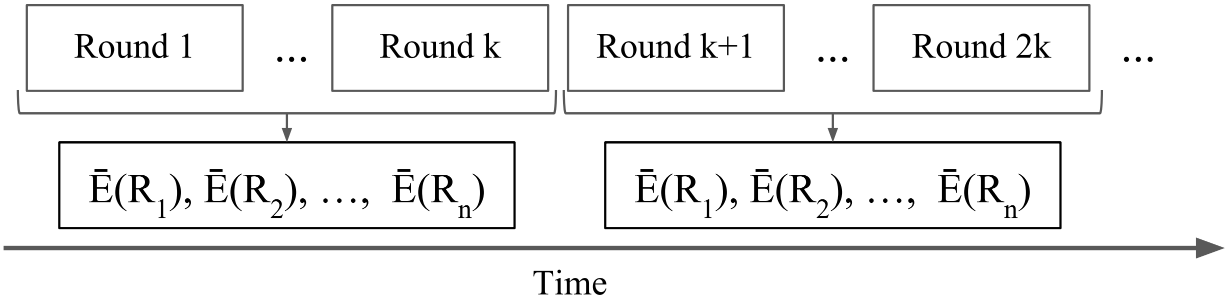


Figure 6.2: Error rate computation for transmission rates

determine the population mean with a specified level of confidence, when the population standard deviation is known:

$$k > \left(\frac{z * \sigma}{MOE}\right)^2 \tag{6.1}$$

For a 95% confidence level, $z = 1.96$, assuming that the underlying distribution is normal. The sample size k is maximized when the standard deviation (σ) is maximized. The value of σ is maximized (i.e., $\sigma = 0.5$) when half of the frames fail and the other half succeed. Using a 10% margin of error (MOE) and confidence level of 95% the minimum sample size required is 97 frames.

Since it takes approximately 43 *ms* to complete a round for all 96 rate configurations and we need a minimum of 97 observations (i.e., rounds), it takes about 4.2 seconds to collect enough samples to compute an average FER. If the channel access is delayed by WiFi and non-WiFi interference, it takes more than 43 *ms* to complete a round and more than 4.2 seconds to conduct enough observations to compute the FER. Therefore, we calculate the average FER using a 10-second window. The number of samples used for the window over which the FER is calculated in all experiments is 232.

6.2.3 Relationships and their Computation

Many methodologies could be used to assess the relationship between two rate configurations. We first define what we mean by *relationship* and then describe the methodology used in our study. Section 6.7 describes several methodologies that seem appropriate but are not suitable. Note that there exist several different connotations of the term relationship and what a relationship between rate configurations might mean. It is important to understand that for the purposes of this study, we are concerned strictly with the relationships as denoted by the following definition.

Relationships

Relationship: *We say that there exists a relationship between rate configurations R_1 and R_2 ($R_1 \mapsto R_2$) if the FER of rate configuration R_1 can estimate the FER of rate configuration R_2 with some expected degree of accuracy. In this case, we call R_1 the* estimator *and R_2 the* estimated *configuration. Note that relationships may or may not be reflexive.*

To provide the intuition behind our methodology, we first present an example of a relationship analysis between two transmission rate configurations. Using two stationary devices and the 2.4 GHz band, we collect data for all 96 rate configurations and compute the average FER over 10-second windows using the techniques described in Sections 6.2.1 and 6.2.2. Figure 6.4 shows the FERs for two of the 96 configurations, namely 2S-I6-LG-20M=104 (configuration R_1) and 2S-I7-LG-20M=117 (configuration R_2). The transmission rate configuration notation is described in Figure 6.3.

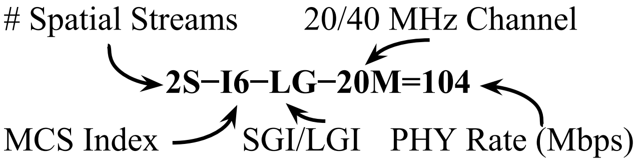


Figure 6.3: Transmission rate configuration notation

The changes in the FER over 30 minutes can be seen in Figure 6.4. One can see that the FERs of these two rate configurations seem to change together, suggesting the existence of a relationship between them.

Another way to examine the relationship between the FERs of two rate configurations (irrespective of time) is to use a scatter plot with the FERs for one configuration along the x-axis and the FERs of the other configuration along the y-axis. We remove the time component because our goal is to determine relationships that persist over time *even in the presence of changing channel conditions*. If at a point in time, t, in Figure 6.4 the FERs of configurations R_1 and R_2 are e_1 and e_2, respectively, these two points are represented on the scatter plot (Figure 6.5) by one point with x and y values of e_1 and e_2, respectively.

Rate configuration R_1 can *accurately estimate* rate configuration R_2 if the FER values of R_1 are mapped to *a relatively small range* of FERs of R_2. To determine whether or not rate configuration R_1 is an estimator for rate configuration R_2, we divide the data in the

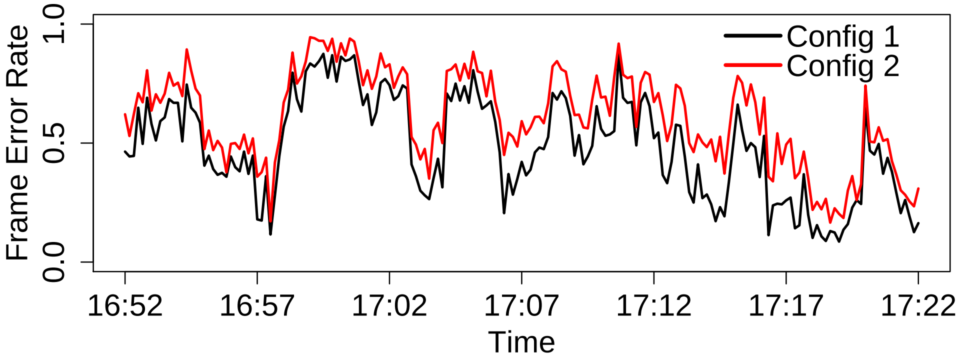


Figure 6.4: FER of two rate configurations changing over time

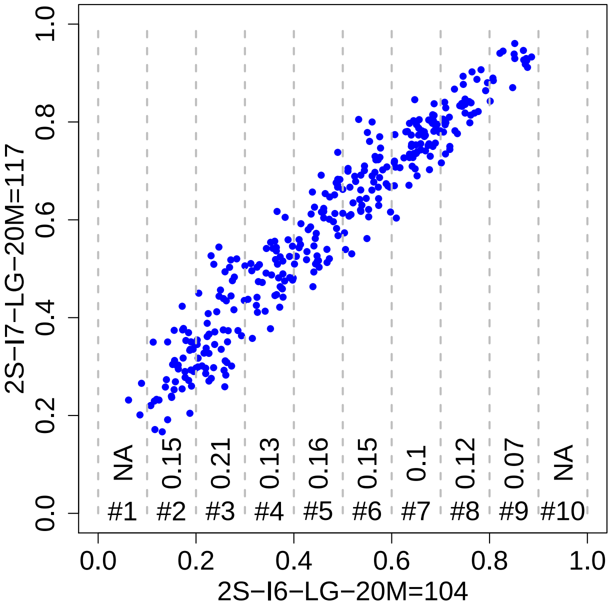


Figure 6.5: FERs, bins and estimation power for $R_1 \mapsto R_2$

scatter plot into bins, illustrated by dashed lines in Figure 6.5. For our analysis, we have chosen 10 bins; other values are possible and this is discussed in Section 6.7. Ideally, the dispersion of points (i.e., variation) in the vertical dimension is low in each bin, indicating that a reasonably accurate estimation of the FER of rate configuration R_2 from FER of rate configuration R_1 is possible. In this instance, an accurate estimate of the FER of R_2 when the FER of R_1 is 0.85 is possible, because the dispersion of the points in bin #9 is relatively low (i.e., 0.07). However, accurate estimation of the FER of R_2 when the FER of R_1 is 0.25, is not possible, because the corresponding FERs in bin #3 for R_2 have a fairly wide range (0.22 to 0.57).

Estimation Power

Ideally, we would like to summarize the strength of the relationship between two rate configurations with a single quantity. We start by quantifying the variation of data points within each bin. Statistical dispersion, which determines how "stretched" or "squeezed" the distribution of data points are, is one such suitable measure. There are several measures of statistical dispersion that could be used. Range and standard deviation are two common measures. However, these measures are highly sensitive to outliers, which may be common because of potentially high variation in frame error rates over time. On the other hand, other measures such as the mean absolute deviation and quartile deviation are more robust to a small number of outliers. We have considered several robust measures of dispersion including *median* absolute deviation (MAD), *mean* absolute deviation, and interquartile range (IQR). In this work, we have chosen to use the *interdecile range* which is the difference between the first and the last deciles (i.e., the first 10% and last 90%). This measure provides the characteristics desired for this study, such as excluding some but not too many outliers. We found that other measures can provide undesirable or misleading values (discussed in more detail in Section 6.7).

We now provide an example of how we apply the interdecile range to the FER data to obtain a *single* quantity that represents the strength of the relationship between two rate configurations. As depicted in Figure 6.5, in each bin the vertical dispersion of the values is calculated based on the interdecile range, which is written above the bin number. The interdecile range values quantify the dispersion in each bin and provide a measure of the relationship between two rate configurations based on the variation of FER in each bin. Note that bins with fewer than 5 values are ignored (labeled as NA, for not applicable), since they do not contain enough data points to provide a reliable measure of dispersion. We describe the importance of bins labeled NA and how we account for them in Section 6.2.3.

To examine relationships between 96 × 96 = 9,216 pairs of configurations, we devise a novel metric we call *estimation power*, which aggregates dispersion values from all bins for each pair of configurations.

Estimation Power: EP(R_1,R_2) *The estimation power of a relationship between rate configurations R_1 and R_2 is a measure of the expected ability of the FER of R_1 to estimate the FER of R_2. It is calculated as the fraction of bins with an interdecile range below a specified threshold.*

The total number of bins excludes those that do not have a sufficient number of data points for the interdecile range to be deemed reliable (we use 5). We use a threshold of 0.2 and discuss both choices in Section 6.7. In the example data in Figure 6.5, the estimation power of rate configuration R_1 to estimate rate configuration R_2, EP(R_1,R_2)= $\frac{7}{8}$.

Variability Indicator

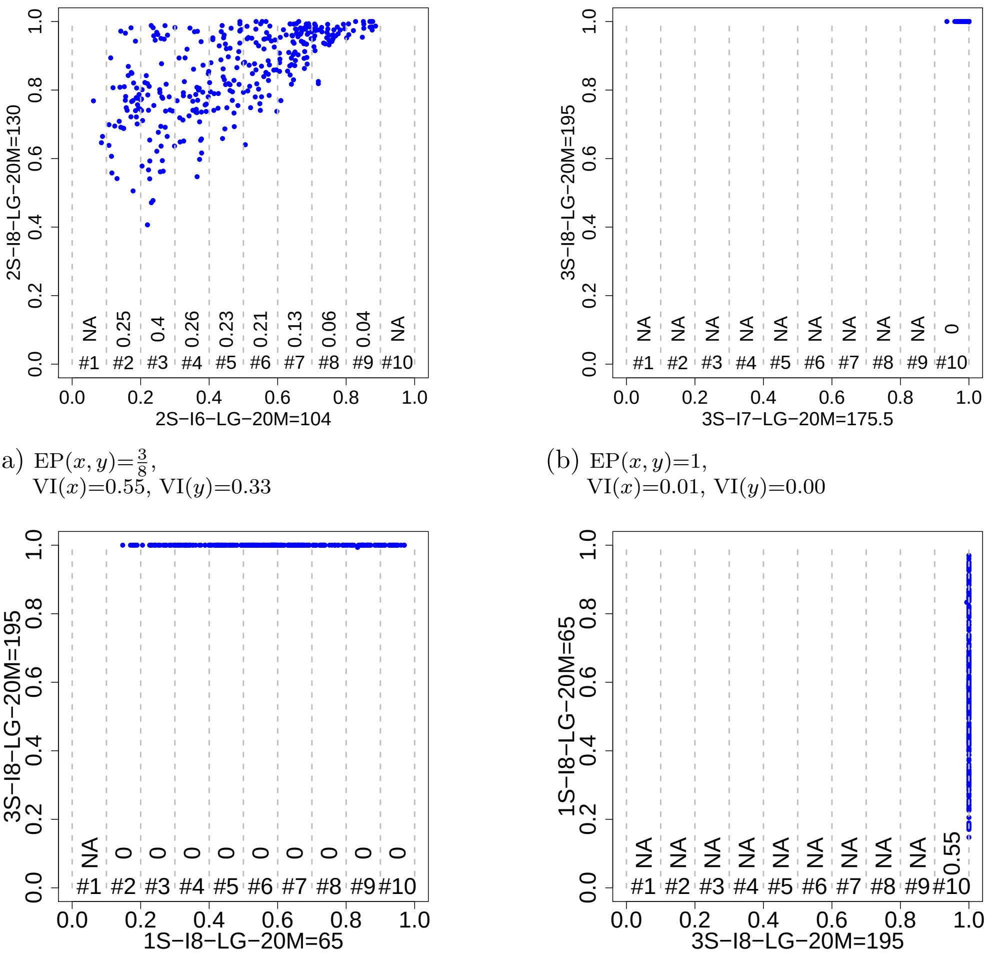
The estimation power (EP) is valuable for quantifying the relationship between two rate configurations. However, we found it beneficial to be able to differentiate types of relationships based on why they exist. We define a new metric called the *Variability Indicator* (VI) that quantifies the variation of the error rate of a rate configuration. Note that the variability indicator is not used to quantify the relationship nor to indicate the lack of a relationship, but rather to interpret and understand the EP.

Variability Indicator (VI): *The variability indicator is a measure of the variability of the frame error rate of a rate configuration. The metric we use is the interdecile range of the FERs of a given configuration over the course of an experiment.*

As discussed in Section 6.2.3, interdecile range is a suitable metric for quantifying the variation or dispersion of frame error rates. The variability indicator helps us to better understand the underlying reasons for the existence or non-existence of a relationship between two configurations. It is important to note that *relationships can exist regardless of the value of the variability indicator.* In Section 6.4.1, we use this measure to help explain the estimation power.

Understanding Our Metrics

To better understand the estimation power metric and the need for the variability indicator metric, we review some scatter plots (in Figure 6.6) for other pairs of configurations from



(a) EP(x, y)=$\frac{3}{8}$,
VI(x)=0.55, VI(y)=0.33

(b) EP(x, y)=1,
VI(x)=0.01, VI(y)=0.00

(c) EP(x, y)=1,
VI(x)=0.55, VI(y)=0.00

(d) EP(x, y)=0,
VI(x)=0.00, VI(y)=0.55

Figure 6.6: Estimation Power (EP) and Variability Indicators (VIs)

the same experiment as used in Figure 6.5. The caption for each subfigure shows the values for the estimation power and variability indicator metrics. The EP metric indicates the ability of the rate configuration on the x-axis to estimate the FER of the configuration on the y-axis. The VI metric is shown for the configurations on both the x and y axes.

Figure 6.6a shows that within several bins, the vertical dispersion of FERs is relatively high. This results in the low estimation power metric ($\frac{3}{8}$), which means that the x-axis configuration (2S-I6-LG-20M=104) is not able to estimate the y-axis configuration (2S-I8-LG-20M=130). Figure 6.6b and 6.6c show an example of one rate configuration (3S-I8-LG-20M=195, on the y-axis) with a constant FER (i.e., VI(y) = 0.00). The constant FER makes it possible to accurately estimate this configuration from the remaining 95 different configurations regardless of their variation in FER. For example, the estimator configuration in Figure 6.6b has low variation (i.e., VI(x) = 0.01), while the estimator configuration in Figure 6.6c has relatively high variation (i.e., VI(x) = 0.55).

Figure 6.6d demonstrates why an estimator configuration with a constant FER (i.e., FER is always 1) cannot estimate a configuration with highly variable FER. In this case, highly dispersed data points are all gathered in one bin (i.e., #10) making an accurate estimation impossible, as indicated by the EP value of 0. To emphasize that EP is a directional metric, Figure 6.6d shows the same configurations as Figure 6.6c, except we have switched the estimator and estimated configurations. As discussed in Section 6.7, symmetric measures that are oblivious to the direction of the relationship (e.g., correlation coefficient and R^2 obtained from a statistical regression) are not suitable for our purposes.

6.3 Experimental Environment

We utilize our testbed, which is housed within lab and office space (described in Section 3.2). Our access point and stationary clients are desktop systems, each containing a TP-Link TL-WDN4800 dual-band wireless N PCI-E adapter. For mobile experiments, we use a laptop configured to use a TP-Link TL-WDN4200 dual-band wireless N USB adapter. All of these 802.11n wireless cards support a MIMO 3x3:3 configuration.

To fully utilize the network infrastructure, we use iperf [54] to generate UDP traffic from the access point to a client at as high a packet rate as possible. We have modified the Ath9k device driver to implement a rate configuration selection algorithm that transmits using each configuration in a round-robin fashion as explained in Section 6.2.1. To record much of the information reported in this study, we use highly detailed information obtained directly from the Ath9k driver.

6.3.1 Different Scenarios Studied

We conduct experiments under a variety of channel conditions including stationary and mobile devices both with and without interference. In some experiments, we use the 5 GHz band to examine channels that are free of interference. In this case, we use a spectrum analyzer to ensure that there is no WiFi or non-WiFi interference. For other experiments, we use the 2.4 GHz band to ensure that the channel is exposed to different types of WiFi and non-WiFi interference. We intentionally selected channel 6, which overlaps with the channel used by the campus WiFi network to test our ability to study relationships in uncontrolled environments.

In mobile experiments, a laptop (i.e., receiver) is carried at walking speed for 15 minutes. Our mobile experiments are conducted in two environments (referred to as *office* and *hallway*), which we designed to exercise a variety of channel conditions. In the office environment, no line-of-sight exists between the AP and client for most of the experiment, since the signal is blocked by obstacles such as metal cabinets, cubicle partitions, and walls. In these experiments, the distance between the AP and client ranges from 1 meter to about 20 meters. In the hallway experiments, a line-of-sight exists between the AP and client, and the distance between them changes from 1 meter to 60 meters.

In the stationary experiments, the AP and client are placed in different rooms in an office environment with no line of sight. All experiments are conducted during the day with people moving in and between offices (this can cause signal attenuation and influence multipath propagation).

To better understand the experimental scenarios, we present some statistical characteristics of the collected data. We classify each of the 96 rate configurations into three categories. The first two, FER < 0.1 and FER > 0.9, indicate that all frame error rate measurements for that configuration are bounded by these values. The variability indicator is low for these categories. The final category captures all configurations that do not fit into the first two categories. Table 6.1 shows, for each scenario, the number of configurations in each category. We observe that in the stationary 5 GHz experiment, which is our most stable environment, the FER of 71 configurations (out of 96) are either less than 0.1 (i.e., most frames are received successfully), or greater than 0.9 (i.e., most frames are lost). The same stationary experiment (with constant transmission power) using the 2.4 GHz band shows different behavior due to WiFi and non-WiFi interference; even the most robust modulation and coding schemes experience errors in the presence of interference. Moreover, 11 configurations almost always fail in this scenario.

To introduce more variability in the FER for the 5 GHz stationary experiment, and to increase the variability indicator in that scenario, we conduct another experiment where the transmission power is changed. Transmission starts at the default maximum setting of 30 dBm and is decreased by 1 dBm every 30 seconds until it reaches 0 dBm. It then increases transmission power by 1 dBm until it reaches 30 dBm and repeats in a round-robin fashion. Table 6.1 shows that in this experiment, the FER of the majority of rate configurations are now variable and even the most robust configurations experience some errors.

	Office					Hallway	
	Stationary			Mobile		Mobile	
Scenario	1	2	3	4	5	6	7
Band (GHz)	2.4	5	$5^\dagger$	2.4	5	2.4	5
FER < 0.1	0	51	0	0	0	0	0
FER > 0.9	11	20	24	0	4	12	9
$0.1 \leq$ FER ≤ 0.9	85	25	72	96	92	84	87

Table 6.1: Characteristics of scenarios († = TX power cycling)

As mentioned previously, the mobile experiments were designed so that a variety of channel conditions are observed during the experiment. The data in Table 6.1 confirms that a majority of configurations experience a variable FER during the experiment. Note that in three of these four scenarios, the FER of some configurations are constantly above 0.9, even though the distance between the AP and the mobile client is about one meter during some points in the movement trajectory. A closer inspection of the raw data showed that these are the rate configurations that result in the highest physical data rates, which never find the perfect channel conditions they need.[1]

6.4 Characterization Results

In this section, we utilize the proposed methodology to examine relationships among 802.11n rate configurations.

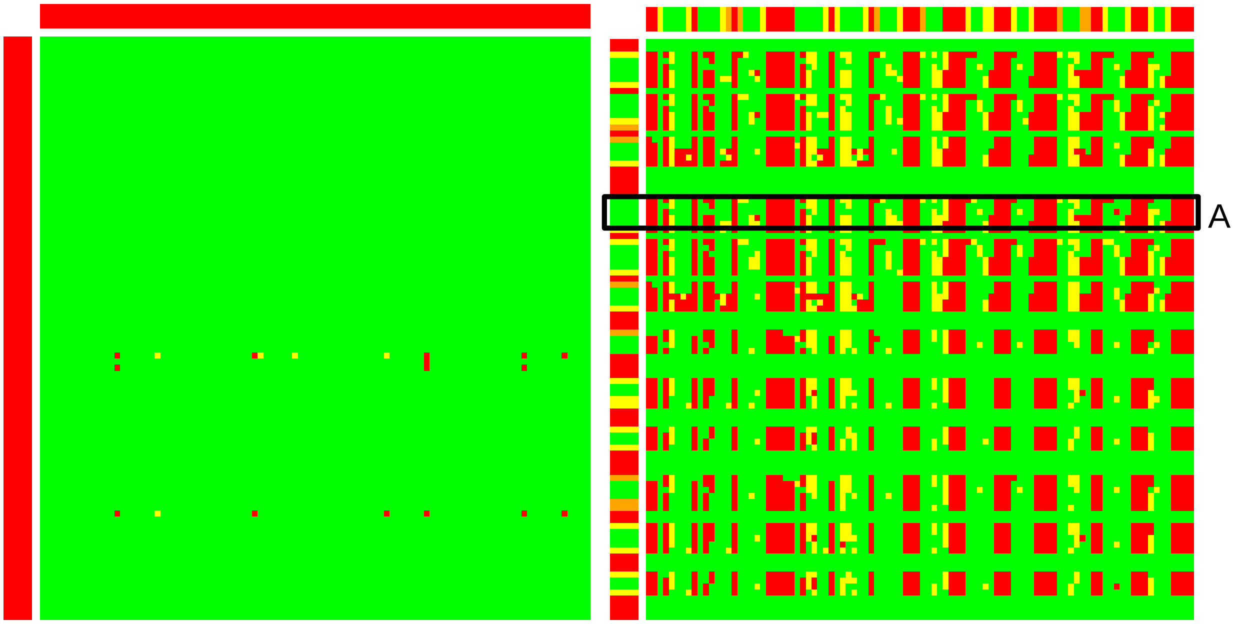
6.4.1 Examining Relationships

We first examine the *estimation power (EP)* and *variability indicator (VI)* of different configurations. Since 96 configurations are supported in our 802.11n cards, 9,216 (i.e., 96×96) relationships can be examined in each experiment.

Figure 6.7a illustrates the relationships between all 96 configurations for the 5 GHz office scenario with stationary devices (Scenario 2). The large square heat map (the EP

[1] This may be because, at short distances, the multipath field is insufficiently rich for the three antennas in the small USB WiFi adapter.

heat map) shows the estimation power of a configuration on the x-axis for estimating the FER of the configuration on the y-axis. Each row and column in this heat map represents a rate configuration. Note that there are 96 configurations on each axis, but labels are removed as they are unnecessary for the high level view we start with. Later, we present subsets of such heat maps in order to have a closer look at some particular relationships. The colors encode ranges for the estimation power (EP) as follows: high ($EP \geq 0.7$)[2] in green, medium ($0.5 \leq EP < 0.7$) in yellow, and low ($EP < 0.5$) in red. The 0.7 and 0.5 thresholds As depicted in Figure 6.7a, the estimation power of all pairs of configurations are very high. To understand why, we study the variability of the FER (i.e., the variability metric) of these rate configurations.



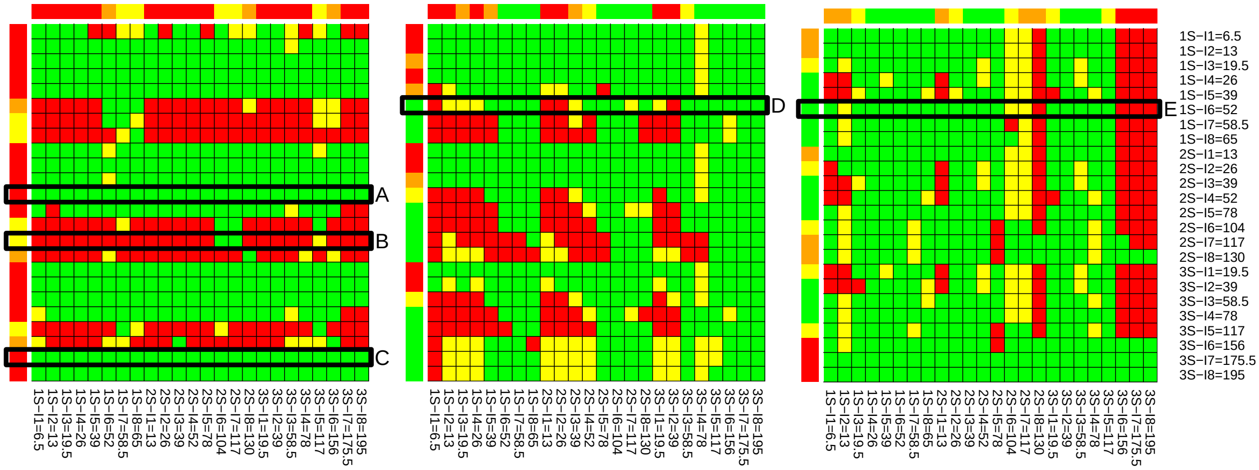
(a) Constant Power

(b) TX Power Cycle

Figure 6.7: Office: stationary, 5 GHz, 96 x 96

The thin heat maps at the top of and to the left of the EP heat map represent the variability indicator for the estimator and estimated configurations, and will be referred to as VI heat maps. The two VI heat maps are always identical but they are shown on both

[2] EP and VI are two different metrics. The 0.7 and 0.5 EP thresholds are chosen to indicate 7 and 5 out of 10 bins, respectively. The 0.75, 0.50, and 0.25 VI thresholds are chosen to represent the quartiles.



(a) Stationary, 2.4 GHz

(b) Mobile, 2.4 GHz

(c) Mobile, 5 GHz

Figure 6.8: Office: stationary/mobile, 2.4/5 GHz, 24 x 24

axes for readability. For the VI heat maps, the colors encode ranges for the variability indicator as follows: very high ($VI > 0.75$)[3] in green, high ($0.5 < VI \leq 0.75$) in yellow, medium ($0.25 < VI \leq 0.50$) in orange, and low ($VI \leq 0.25$) in red.

Interestingly, in Figure 6.7a, the variability indicator is quite low for all rate configurations. This indicates that the FERs do not change significantly. In this scenario, the main reason for so many strong relationships is because the FER of all configurations are mainly constant and are, therefore, easy to estimate. Note that in some of these cases, one configuration may consistently fail (e.g., 3S-I8-SG-40M=450) while the other is consistently successful (e.g., 1S-I1-LG-20M=6.5). Nevertheless, this is a strong relationship.

In an attempt to see if any relationships exist when the FERs are variable, we program the AP to change the transmission power as described in Section 6.3.1 (Scenario 3). The results are presented in Figure 6.7b. The differences between the VI heat maps in Figure 6.7a (which contain only red) and Figure 6.7b (which also contain some green, and some yellow) indicate that the changes in transmission power increase the variability of FERs. The EP heat map in Figure 6.7b shows that many rate configurations are still strongly related despite some FERs being highly variable. For example, although the few configurations in the rows outlined by the rectangle labelled “A” have highly variable FERs as depicted by green cells in the VI plot, we observe that there are many green cells in these

[3] Refer to footnote 2.

rows in the EP heat map. Each green EP heat map cell indicates that the corresponding x-axis configuration can accurately estimate the y-axis configuration. As can be seen in the rectangle “A”, many rate configurations can accurately estimate the y-axis configurations in those rows. Note that despite changing the transmission power, the variability of the FERs of some configurations is still low. By examining details of the collected FER data, we find that these configurations consistently fail in Scenario 2, which uses the maximum transmission power. As a result, lowering the transmission power (in Scenario 3) does not affect the FER of these configurations.

To examine some relationships in more detail, we now consider a subset of all rate configurations in the office scenarios. Figures 6.8a, 6.8b and 6.8c, show relationship results for the stationary (2.4 GHz) and mobile (2.4 and 5 GHz) scenarios, respectively (i.e., Scenarios 1, 4, and 5). In these scenarios, we examine only those configurations that use a long guard interval (LGI) and 20 MHz channels. This restricts the number of configurations to 24 (8 MCSes × 3 spatial streams) and the heat maps to 24 × 24 pairs of configurations.

In all graphs in Figure 6.8, we see patterns of colors suggesting the existence of relationships between estimation power and combinations of physical-layer transmission features. The green cells (indicating a high estimation power) are not scattered randomly on the EP heat maps, but rather clustered in specific patterns based on the physical layer configurations indicated on the x and y axes. We now examine these plots in more detail to better understand these results.

In the stationary scenario, by comparing the relationships in the 2.4 GHz and 5 GHz bands (Scenario 1 in Figure 6.8a and Scenario 2 in Figure 6.7a), we observe more red cells in the 2.4 GHz scenario indicating a decrease in the number of related configurations. The major difference between these experiments is the lack of interference in the 5 GHz band. Figure 6.8a shows that some configurations, such as 2S-I4=52 (see rectangle “A”), can be estimated accurately by many configurations. The variation of the FER of this configuration is low (i.e., indicated by the red cell in the left VI heat map). Therefore, it is easy to estimate. On the other hand, configurations such as 2S-I7=117 (see rectangle “B”) that have higher variation of FER (indicated by the yellow square in the left VI heat map), can be more difficult to estimate. In Figure 6.8a, the variability indicator is low for two groups of configurations: (a) configurations that consistently succeed (in this scenario), such as 2S-I4=52 (rectangle “A”); and (b) configurations that consistently fail (in this scenario), such as 3S-I7=175.5 (rectangle “C”). Other configurations, which are not in either of these groups, experience variable FER such as 2S-I7-117 (rectangle “B”).

Figures 6.8b and 6.8c show the relationships for a scenario where the client (i.e., receiver) device is moving at walking speed in an office environment using the 2.4 and 5 GHz bands (Scenarios 4 and 5). We consider these scenarios to study the effect of mobility on relationships. These are highly variable environments due to mobility. This can also be seen by the number of green cells in the VI heat maps when compared to their stationary counterparts. Figures 6.8b and 6.8c illustrate that strong relationships exist between many rate configurations. Interestingly, in both scenarios, for the estimated (y-axis) configurations with highly variable FERs (i.e., green or yellow cells in the left VI heat map), there are several (x-axis) configurations that can accurately estimate them. For example, configuration 1S-I6=52 has a highly variable FER in both scenarios (rectangles "D" and "E") as indicated by the corresponding green cells in the left VI heat maps. However, as depicted in Figures 6.8b and 6.8c, there are several green cells in the 1S-I6=52 rows ("D" and "E"), indicating the existence of several estimators for this configuration. The red cells in the 1S-I6=52 rows correspond mostly to the estimators with low variability FERs (i.e., indicated by red cells in the top VI heat map), since it is difficult for their relatively constant FER to estimate a variable FER. We have highlighted a few interesting scenarios and now present an overview of the relationships.

6.4.2 Overview of Relationships

To provide a high-level view of the number and strength of different relationships, we summarize the results from all experiments in Table 6.2. One measure of interest presented in this table is the count of the number of other rate configurations that can be used to estimate a particular rate configuration R, denoted $|* \mapsto R|$. Another metric that we present is the count of the number of other rate configurations that a particular rate configuration R can estimate, denoted $|R \mapsto *|$. To quantify if R_1 can estimate R_2, we use a threshold for the estimation power. In the heat maps shown in this chapter, we have used a threshold of 0.7 to indicate a very strong relationship. As a result, we also use this threshold when computing the data presented in Table 6.2. The intuition is that if the estimation power of R_1 when estimating R_2 is greater than or equal to 0.7, we presume that R_1 can estimate R_2. In addition, to study the impact of that threshold on our results, we also include computations using the most conservative threshold possible (1.0). Recall that this means that the dispersion metric in *all bins* must be no greater than the dispersion threshold (i.e., 0.2).

Table 6.2 shows a variety of information for the seven scenarios examined. For each scenario, we present the Min, Max and Min SC values for the defined measures $|* \mapsto R|$ and $|R \mapsto *|$. The Min and Max values are of interest in understanding the number of relationships that exist between different configurations (Min SC will be described later).

		Office: Stationary						Office: Mobile				Hallway: Mobile			
		Scenario 1		Scenario 2		Scenario 3†		Scenario 4		Scenario 5		Scenario 6		Scenario 7	
		2.4 GHz		5 GHz		5 GHz		2.4 GHz		5 GHz		2.4 GHz		5 GHz	
		60 minutes		60 minutes		60 minutes		15 minutes		15 minutes		15 minutes		15 minutes	
EP Thold	Stat	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$	$\lvert * \mapsto R\rvert$	$\lvert R \mapsto *\rvert$
0.7	Min	1	32	88	91	18	33	6	4	13	19	11	19	4	17
	Max	95	66	95	95	95	95	61	77	95	59	95	80	95	92
	Min SC	5 – 11		1		1		3		2		3		2	
1.0	Min	1	19	88	91	1	33	6	3	13	19	10	19	3	17
	Max	95	41	95	95	95	63	60	68	95	59	95	68	95	55
	Min SC	5 – 15		1		5		4		2		3		3	

Table 6.2: Summary of relationships. † indicates TX power cycling

The greater the number of relationships, the more likely we are to be able to accurately estimate the FER of one configuration from the FER of one or more other configurations.

We start by focusing on the values obtained with a threshold of 0.7. In the worst case (i.e., the lowest value) across all scenarios, there is only one estimator ($|* \mapsto R|$) for a particular rate, which occurs in Scenario 1 (see the row labelled "Min" and column $|* \mapsto R|$). When examining the number of configurations that can be estimated by a single rate ($|R \mapsto *|$), the minimum value is 4, which occurs in Scenario 4 (see the row labelled "Min" and column $|R \mapsto *|$).

We now consider the minimum number of rate configurations that are required to be able to estimate (i.e., cover) all other rate configurations. This problem is equivalent to the *minimum set cover* problem, which is NP-hard [63]. Interestingly, the minimum set cover in six (i.e., Scenarios 2 – 7) of the seven scenarios was small enough (the largest of these was 3), that we were able to find them using a brute force approach that examines all sets up to size 4. These small values indicate that there are a number of strong relationships that could potentially be exploited. In Scenario 1, we have used a heuristic search to determine that the set cover size is between 5 and 11. The results for all scenarios are shown in the row labelled Min SC in Table 6.2.

We now focus on the values obtained in Table 6.2 using the most conservative threshold of 1.0. Comparing the values obtained using the two different thresholds shows that in most instances the values do not change significantly. Furthermore, the size of the minimum set cover is unchanged in three of the seven scenarios and increases only slightly in the other four. Note that the precise numbers in these tables depend on the various parameters used to determine whether relationships exist or not. The impact of these choices is discussed in Section 6.7.

It is interesting to note that there are large numbers of strong relationships between a variety of different configurations in the experiments conducted using the interference-free 5 GHz band. This can be seen in Figure 6.7, and from the data provided in Table 6.2. Perhaps more compelling, however, are the surprisingly large numbers of relationships in the 2.4 GHz and mobile experiments. We believe that these are intriguing results considering: (1) the complexity of the interactions of the large number of possible configurations of physical layer features; (2) the constantly changing channel conditions due to WiFi and non-WiFi interference in the uncontrolled 2.4 GHz experiments (Scenarios 1, 4 and 6); (3) the continual fluctuations in signal quality due to mobility (Scenarios 4 – 7); and (4) the relatively long period of time over which these relationships hold (e.g., data for the stationary experiments covers 1 hour). In Section 6.7, we discuss possible limitations of these results, such as robustness of these relationships across different devices.

6.4.3 Changes in Relationships Over Time

To study if and how relationships might change over time, we start with a simple demonstration by dividing the one-hour stationary 2.4 GHz experiment (i.e., Scenario 1) into four 15-minute experiments and perform the same statistical analysis over each time window. Table 6.3 presents the relationship data for each sub-window (EP threshold = 0.7). We observe that various measures of the relationships change with time. For example, the size of the minimum set cover is 1, between 5 and 7, 5, and 1 for the four 15-minute sub-windows. Note that the minimum set cover was calculated to be between 5 and 11 over the full 1-hour experiment. These are significant reductions that demonstrate that a greater number of strong relationships may exist over shorter time intervals, and that relationships between rate configurations vary over time. The time window over which relationships will be computed in practice will depend on the application in which they are being used. However, because we have found evidence that relationships can be found over periods of 15 to 60 minutes in the scenarios we have examined, applications should not have to recompute relationships too frequently.

6.5 Studying PHY-Layer Features

In this section, we use our methodology to better understand the efficacy of short and long guard intervals, and then analyze the relationships that exist because of this feature.

	0–15 Min		15–30 Min		30–45 Min		45–60 Min		Overall	
Stat	$\lvert * \mapsto R \rvert$	$\lvert R \mapsto * \rvert$	$\lvert * \mapsto R \rvert$	$\lvert R \mapsto * \rvert$	$\lvert * \mapsto R \rvert$	$\lvert R \mapsto * \rvert$	$\lvert * \mapsto R \rvert$	$\lvert R \mapsto * \rvert$	$\lvert * \mapsto R \rvert$	$\lvert R \mapsto * \rvert$
Min	5	40	1	43	3	19	10	40	1	32
Max	95	75	95	77	95	72	95	95	95	66
Min SC	3		5 – 7		5		1		5 – 11	

Table 6.3: Changes in relationships over time, Scenario 1

6.5.1 Efficacy of LGI and SGI

Some people have argued that for indoor environments, the 800 *ns* guard interval (LGI) used in 802.11 protocols prior to 802.11n was more conservative than necessary [42, 78, 91, 109]. As a result, the shorter 400 *ns* guard interval (SGI) was introduced in the 802.11n standard. Because we are not aware of any empirical studies that examine if there is a need for the LGI in indoor 802.11n networks, we utilize our methodology to study this issue.

Suppose we examine the FER of all rate configurations that differ only in whether they use LGI or SGI (i.e., other features are the same). If their FERs are nearly identical, it indicates that shrinking the guard interval from 800 to 400 *ns* for these particular rate configurations does not have an adverse effect on the FER in the scenarios examined. Therefore, SGI should be used instead of LGI, since it provides higher throughput in situations where the FERs of LGI and SGI are the same.

The FERs of a pair of rate configurations, 1S-I2-LG-20M=19.5 and 1S-I2-SG-20M=21.7, are shown in Figure 6.9. The only difference between these configurations is the length of their guard intervals. The data was collected using the office, mobile scenario using both the 2.4 and 5 GHz bands (i.e., Scenarios 4 and 5). When using the 5 GHz band, the FER of the SGI configuration is generally higher than the LGI configuration. We also observe this behavior for many pairs of rate configurations that differ only in their use of SGI or LGI (not shown here) in the 5 GHz band office and hallway scenarios (i.e., Scenarios 5 and 7). Interestingly, when using the 2.4 GHz spectrum (i.e., Scenarios 4 and 6), the ratio of the FERs are close to equal.

We believe different results are seen in the 2.4 and 5 GHz bands because many building materials reflect 5 GHz signals orders of magnitude stronger than 2.4 GHz signals [113]. Thus, the delay spread can be longer for 5 GHz signals, increasing the FER of rate configurations using SGI because reflected signals arrive after the SGI and interfere with the next symbol.

We now study if the observed difference in FER is significant enough to result in the infe-

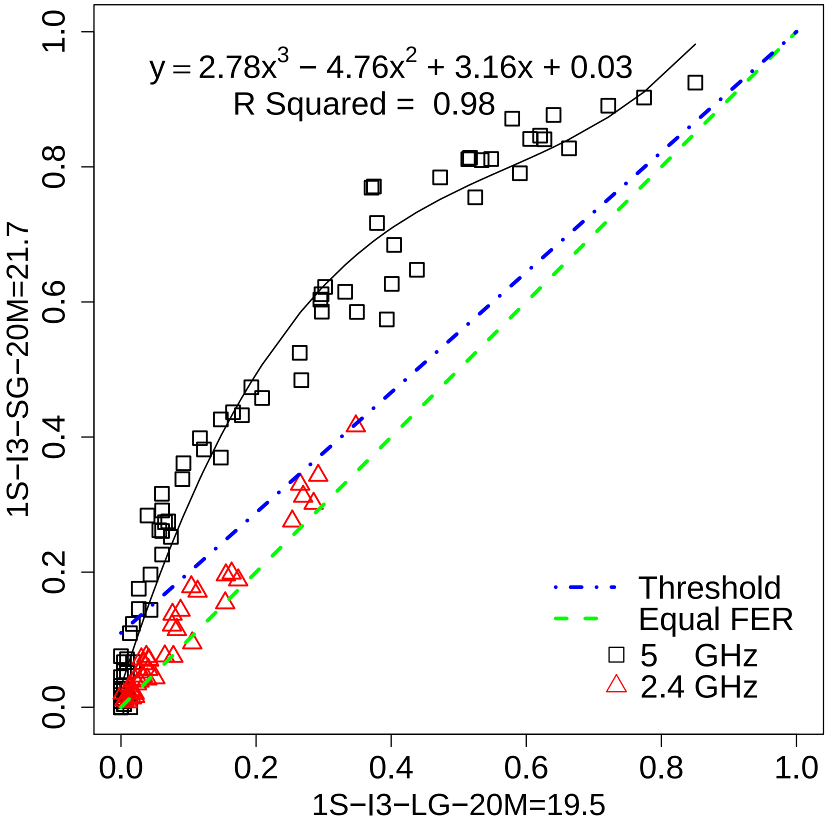


Figure 6.9: Office: mobile, 2.4 and 5 GHz

rior performance of the SGI configuration. The short guard interval provides a throughput increase of at most 11% when compared with the throughput of the corresponding LGI configuration. If the FER of the SGI configuration is too high, the extra throughput achieved from the SGI can no longer compensate for the higher FER. The blue dashed line (labeled "Threshold") shows this threshold. Points above this line indicate that the throughput of the LGI configuration is higher than the SGI configuration. As seen in Figure 6.9, in the 5 GHz band, many data points are above this line, indicating that the LGI configuration provides higher throughput than the SGI configuration. We observed similar results for other pairs of rate configurations and scenarios when comparing SGI and LGI configurations. Therefore, in the 5 GHz band, using the LGI could provide higher throughput for some configurations.

6.5.2 Relationship Between LGI and SGI

We now more closely examine the existence of relationships between LGI and SGI rate configurations. Figure 6.9 shows a non-linear relationship for the 5 GHz experiment. We found that a quadratic regression model (shown at the top of the figure) fits the data very well (i.e., $R^2 = 0.98$). To verify that LGI and SGI rate configurations are related for other combinations of physical-layer transmission rate features and other scenarios, we compute the estimation power from LGI to SGI configurations and from SGI to LGI configurations for all other scenarios and rate configurations (i.e., 48 configurations). Figure 6.10 shows the estimation power for all configurations for Scenarios 1, 2, 4, and 5 (from top to bottom, respectively). The graphs plot the rate configuration on the x-axis and the estimation power value on the y-axis for both the LGI $\mapsto$ SGI and LGI $\mapsto$ SGI relationships. Note that we only label every second rate configuration on the x-axis. The results show that the relationships are strong between all pairs of configurations that differ only by the guard interval length. Similar results were observed for the other scenarios. In the next section, we utilize the relationships between LGI and SGI configurations to demonstrate how a rate adaptation algorithm might utilize such relationships to optimize transmission rate feature configurations.

6.6 Impact on Applications

Many rate adaptation algorithms (RAAs), including the Minstrel HT algorithm implemented in the widely used Ath9K driver, rely on sampling to determine the best rate configuration. Unfortunately, the overhead of sampling is significant [67]. In order to reduce this overhead, the sampling frequency can be reduced by sampling a smaller subset of rate configurations and utilizing relationships between FERs to estimate the FERs of the remaining rates.

To demonstrate the impact of utilizing relationships in rate adaptation, we modify Minstrel HT to estimate the FER of LGI configurations from the SGI configurations (instead of sampling LGI configurations). In this way, we reduce the frequency at which different configurations are probed. We refer to this modified version as Minstrel HT Relationship. Our estimation function assumes that the FERs of two rate configurations are identical if the configurations only differ in the guard interval. Note that the RAA may still select LGI configurations, it just does not sample them. Although in Section 6.5.1 we show that in the 5 GHz bands the FER of the SGI configurations are sometimes higher, the performance results indicate that using the simpler estimation function is sufficiently accurate for this

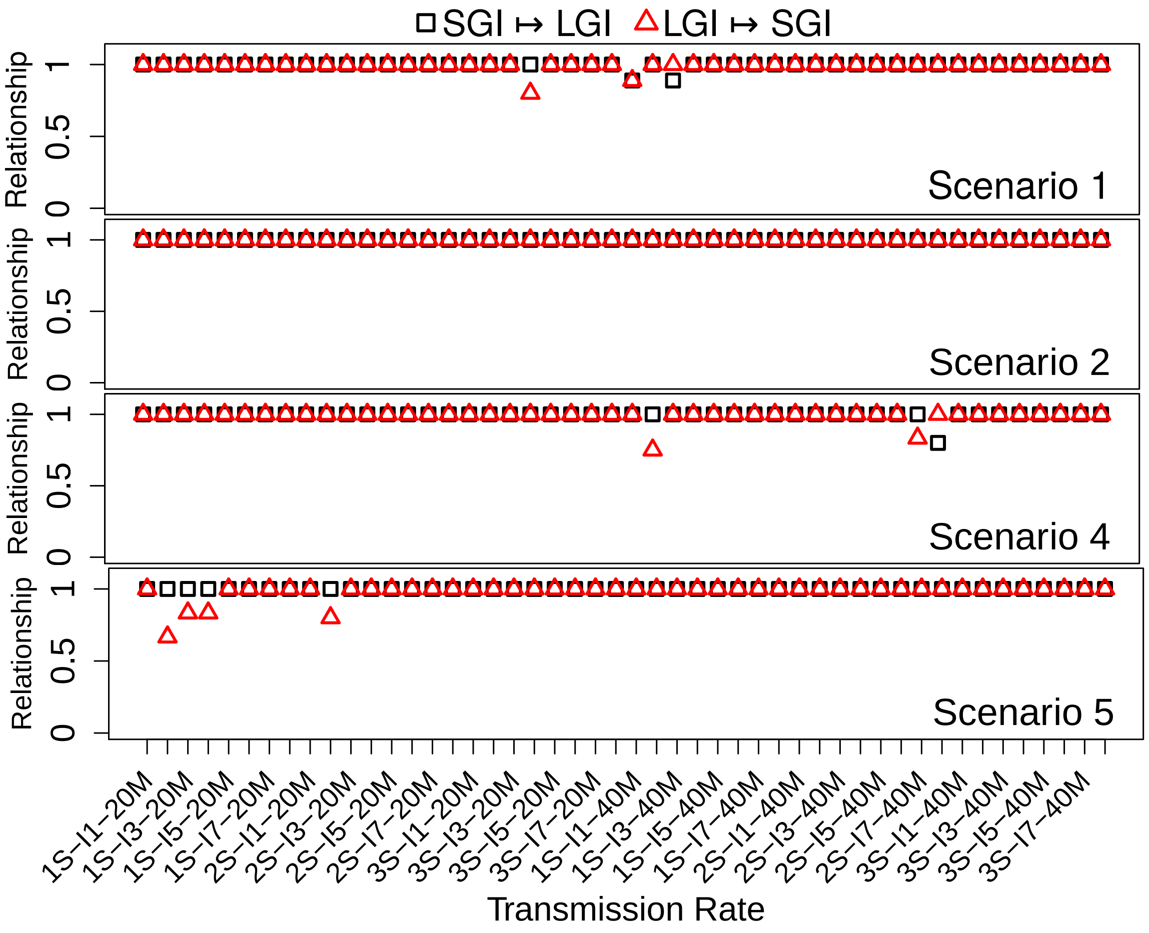


Figure 6.10: Estimation power of SGI/LGI rate configurations

illustration. We run each experiment five times and present the average throughput with 95% confidence intervals in Figure 6.11. The two pairs of outer bars show that Minstrel HT Relationship increases throughput when compared with the vanilla version by 28% and 17%, in the stationary 2.4 GHz and mobile 5 GHz scenarios, respectively. The middle bars in each grouping show the throughput obtained using Minstrel HT when the probing frequency is reduced (Low Sample Rate) to that used by Minstrel HT Relationship. The gaps between the "Low Sample Rate" bars and the "Relationship" bars demonstrate that throughput is improved as a result of using relationships, and does not come only from reducing the number of probed configurations.

These findings demonstrate the potential power of exploiting relationships among rate configurations in algorithms that optimize the selection of physical-layer transmission features. Designing and comprehensively evaluating an RAA to fully utilize relationships is outside the scope of this thesis and is left for future work.

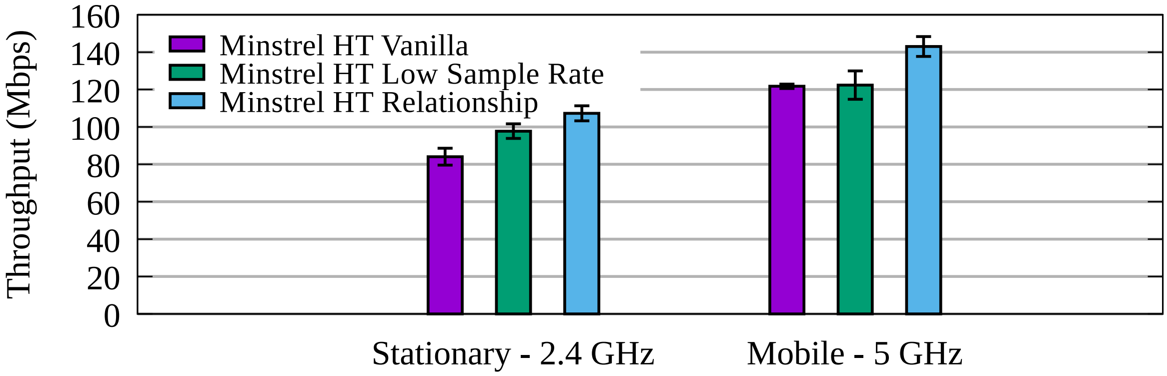


Figure 6.11: Impact of using relationships in RAAs

6.7 Discussion

Our results are limited to the scenarios and hardware used for these experiments. However, the number of strong relationships between many configurations across the scenarios examined suggests that interesting relationships are likely to exist under a variety of channel conditions.

Our methodology utilizes several parameters such as: the number of bins (10), the minimum number of data points required for the dispersion metric to be considered reliable (5), the threshold for the interdecile range for a bin to be considered acceptable for accurate estimation (0.2), and the value used to consider the estimation power metric to be good enough to consider the relationship as strong (0.7 for the heat maps). We chose these parameters based on visual inspection of significant amounts of data and by trying to find a set of parameters and a methodology that are fairly intuitive. We believe that the chosen parameters are fairly conservative. However, the best choice of these values will depend on the purpose of the particular characterization study or the application to which it is being applied. Topics for future research include: the choice of parameters, good estimator functions, and studying their accuracy.

To choose the most suitable methodology for characterizing relationships, we have considered many different methodologies including but not limited to correlation, conditional entropy, and parametric and non-parametric regression. As outlined in Section 6.2.3 our definition of relationship is, *by necessity* a directional measure. Unfortunately, many techniques such as correlation (e.g., the Pearson product-moment correlation coefficient), and R^2 of regression provide the same value for $R_1 \mapsto R_2$ and $R_2 \mapsto R_1$ relationships which is not true of the relationships that we believe are interesting for this study. In addition, correlation only detects simple linear correlations; however, we found non-linear relationships between rate configurations. For these key reasons, correlation and R^2 were not used in this study. While conditional entropy does consider the direction of the relationship, it is not defined for some special but critical cases we observe. For example, it is not defined when there is no variability in the FER of the estimator rate configuration. Therefore, it cannot be used to quantify estimation power.

We chose to focus on the 802.11n standard because it is widely used, it supports both 2.4 and 5 GHz bands, and because the open source Ath9k driver made it possible to implement our data collection methodology quickly and easily. Although we believe that many of our findings will apply in 802.11ac networks, which share many physical-layer features with 802.11n networks, studying 802.11ac networks is left for future work.

6.8 Chapter Summary

In this chapter, we design a methodology for evaluating the relationships between the FER of different physical-layer transmission feature combinations (rate configurations). We find that in all seven scenarios examined, a surprisingly small number of rate configurations can estimate the FER of all other configurations. Interestingly, although we demonstrate that relationships can change over time, relationships are observed in uncontrolled environments over periods of up to one hour. Although some people have suggested that the 800 ns guard interval (LGI) is too conservative, we find that an LGI can provide higher throughput than the 400 ns guard interval (SGI) for several pairs of rate configurations in the scenarios we have studied. Finally, by utilizing a small fraction of relationships, we provide a simple illustration of how the throughput of the widely used Minstrel HT algorithm can be increased by up to 28% in the uncontrolled environments tested. This demonstrates that there are significant opportunities for utilizing relationships between rate configurations in designing algorithms that must choose the best combination of physical-layer features from a large number of possibilities.

Chapter 7

Conclusions and Future Work

7.1 Thesis Summary

The 802.11 standard has become the dominant protocol for wireless local area networks. The ever-increasing demand for more bandwidth, along with the problems caused by interference over the shared spectrum, have made the performance evaluation and characterization of 802.11 networks very important and difficult. Simultaneous transmissions from two or more WiFi devices is likely to cause interference, which adversely impacts throughput. When a high number of WiFi devices are in the vicinity of each other, interference occurs more frequently. The rapid growth of WiFi-equipped devices increases the density of these devices, which triggers not only a higher demand for the overall network bandwidth, but also a higher chance of interference, and therefore creates challenges for achieving good performance in these networks.

The focus of this thesis is on the tools and techniques for the performance evaluation and characterization of 802.11 networks. Specifically, we study different methodologies for measuring the throughput of 802.11 networks for the purpose of comparing multiple competing algorithms or systems that depend on these networks. We review existing performance evaluation techniques for 802.11 networks and find serious flaws with these methodologies. In order to achieve repeatability when conducting empirical measurements of 802.11 networks, previous work usually avoids scenarios that include mobility or interference. However, these scenarios are not representative of the environments in which WiFi devices are typically used.

We propose two novel evaluation methodologies (one for conducting empirical evaluations and another for trace-based simulations). Our experimental methodology expands the scenarios under which repeatable experiments can be conducted for the purpose of comparing multiple competing alternatives. This methodology enables repeatable empirical performance evaluations for the purpose of fair and valid comparisons in variable channel conditions, including those affected by mobility and interference. Our trace-based simulation framework combines the realism of actual experiments with the repeatability of simulation. This framework provides the ultimate level of repeatability, since competing alternatives experience identical channel conditions at the granularity of the fate of a single packet. Even high variations in the channel conditions caused by mobility and interference do not limit the usability of this methodology. In addition, we characterize the relationships between 802.11n transmission rates in order to discover opportunities for improving optimization algorithms such as rate adaptation algorithms.

7.1.1 Experimental Evaluation of 802.11 Networks

Wireless channels change over time due to the movements of the sender, receiver, or nearby objects (due to large-scale and small-scale fading). In addition, interference from WiFi and non-WiFi devices in the shared spectrum increases fluctuations in the attainable throughput in 802.11 networks. A fair and valid empirical comparison of two or more competing algorithms or systems that depend on 802.11 networks can be extremely challenging due to changes in the channel quality. When a channel changes over time, it is not clear if an observed difference in the performance of two competing alternatives is due to the variable channel conditions or differences in those alternatives.

We examine different existing methodologies for conducting experiments to compare the performance of systems that use 802.11n MIMO networks. We show that some commonly used techniques for comparing the performance of different alternatives are flawed, even in highly controlled environments that are free from interference from other WiFi and non-WiFi devices. We find that even running the experiments multiple times to obtain an average with confidence intervals can produce misleading results. We propose and evaluate the Randomized Multiple Interleaved Trials (RMIT) methodology, which provides repeatable results and can be used to distinguish differences in performance, even with highly-variable channel conditions.

7.1.2 Trace-Based Evaluation of 802.11 Networks

We design a novel trace-based framework for the performance evaluation of 802.11g and 802.11n networks that achieves realism and repeatability at the same time. In addition, evaluating the performance of 802.11 networks using our trace-based simulator is typically much easier than conducting experiments.

The T-RATE framework focuses on the trace-based evaluation of 802.11g rate adaptation algorithms, since the choice of modulation and coding schemes is the only configuration that must be chosen during transmission (i.e., 8 transmission rates). We present the design and evaluation of the T-RATE framework. We show that T-RATE is able to accurately simulate the operation of different rate adaptation algorithms and provides realistic performance results. With the introduction and wide availability of 802.11n networks that support several configurable transmission features, link adaptation is no longer limited to selecting the best modulation and coding scheme (i.e., up to 128 transmission rates). Therefore, for 802.11n networks, we have broadened our focus to also include different optimization algorithms including frame aggregation and channel width adaptation algorithms. We present an overview of the T-SIMn framework, outline the challenges associated with collecting traces on 802.11 networks, and describe the design of a novel trace collection methodology that enables T-SIMn to handle devices that support many transmission rates. Our evaluations show that the proposed trace collection methodology can be utilized in the T-SIMn framework to accurately handle devices that support three spatial streams (i.e., 96 transmission rates).

7.1.3 Characterization of 802.11n Networks

Starting with the 802.11n standard, several configurable transmission features have been introduced which create many transmission rates. The number of transmission rates determines the search space of some optimization algorithms such as rate adaptation. A bigger search space can negatively impact the performance of the optimization algorithms due to a potentially longer search time to find the best transmission configuration.

We characterize 802.11n networks with the purpose of finding relationships between transmission rates in order to reduce the size of the search space for optimization algorithms. We find that strong relationships exist between most transmission rates even under highly-variable channel conditions, including those that are affected by WiFi and non-WiFi interference and mobility. As a result of determining these relationships, optimization algorithms can probe a small subset of all transmission rates and infer the error rates of other transmission rates instead of probing them. We show that a rate adaptation algorithm that implements a simple heuristic that takes advantage of the existing relationships can improve throughput by up to 28% compared to the vanilla version.

7.2 Future Work

In this section, we present some possible avenues for future work.

7.2.1 Evaluating the Efficacy of Numerous Transmission Rates

The 802.11 standard supports 8 transmission rates that correspond to different modulation and coding schemes. In the 802.11n and subsequent standards, other transmission features with selectable configurations such as guard interval and channel width were introduced. Different combinations of these transmission features create many transmission rates. Table 7.1 shows the rapid growth in the number of transmission rates in the 802.11 standard. The 802.11ax standard, which is going to be released in 2018, supports up to 1,152 transmission rates. Some optimization algorithms, such as rate adaptation algorithms, try to maximize throughput by trying to find the best combination of transmission features. The larger the number of rates, the more difficult it becomes to choose the best transmission rate.

Standard	MCS	Guard Intervals	Channel Width	Spatial Streams	Total Rates
802.11g	8	1	1	1	8
802.11n	8	2	2	4	128
802.11ac	10	2	4	8	640
802.11ax	12	3	4	8	1152

Table 7.1: The rapid growth in the number of transmission rates in 802.11

The trend of increasing number of transmission rates in the 802.11 standards motivates the need to study the cost-benefit analysis of supporting many transmission rates. In other words, how many transmission rates are actually required: can 802.11 networks perform as well or even better with fewer transmission rates? Studying the efficacy of supporting many transmission rates should be a fruitful avenue for future work.

7.2.2 Quantifying the Realism of Wireless Channel Models

As discussed in Section 2.2.2, wireless channel models used in the simulation or emulation of 802.11 networks are understood to be not very accurate. Although, it is commonly accepted [57,87,105] that such models suffer from a lack of realism, no study has quantified the inaccuracies of these models, to the best of our knowledge. It is important to determine the degree of inaccuracy of the simulators and emulators that use such models, in order to understand and be aware of their margin of error. An even more interesting question is that if wireless channel models are used instead of conducting actual experiments when comparing multiple alternatives, might these models provide misleading results regarding which alternative is better? Answering these question will provide valuable information regarding the extent to which wireless channel models should be relied on to evaluate the performance of 802.11 networks.

7.2.3 Utilizing T-SIMn to Design Optimization Algorithms

The 802.11 standard does not specify how to determine transmission configurations. As a result, device drivers or device firmware implement algorithms for rate adaptation, frame aggregation, or MIMO settings (e.g., if an antenna should be used in spatial multiplexing or Space-Time Block Code[1] (STBC) mode). Since the introduction of the 802.11 standard, researchers have been developing optimization algorithms to improve various performance metrics such as throughput, delay, and energy consumption. The process of designing and testing these algorithms can be very time-consuming and frustrating. Designing and testing a new algorithm usually requires multiple rounds of compiling the driver, loading proper modules, setting up the access point and the client(s), and conducting several experiments (as described in Chapter 3). This may include carrying a wireless device for a relatively long time. Each time a problem is found in the design, this procedure must be repeated, which can be very time-consuming.

The T-SIMn framework is a great tool for designing new algorithms, since researchers can implement their new ideas and easily measure and compare the performance of algorithms using many realistic traces. When something is changed in the algorithm, we can simply rerun the trace-driven simulations and compare the result with previous versions. The T-SIMn framework significantly improves the efficiency of the design and evaluation of optimization algorithms. Utilizing the T-SIMn framework, we intend to design new rate adaptation and frame aggregation algorithms using our findings from the characterization studies presented in this thesis.

[1] STBC transmits the same copy of data from two antennas. If the receiver fails to decode one copy, it has a second chance to recover the data.

7.2.4 Further Characterization Studies

Our characterization study presented in Chapter 6 has opened up many avenues for future work. We now briefly describe some of them. We have found strong relationships between many pairs of transmission rates. If these relationships are utilized, for example by a rate adaptation algorithm, to perform error rate estimation, how accurate must these estimations be? Quantifying the accuracy of the estimations obtained from relationships between transmission rates provides some guidance regarding the scope of applicability of these relationships.

In each scenario we studied, only a small subset of transmission rates are needed to estimate all other rates. Are there a unique subset of rates that can possibly estimate all other rates in all or many scenarios? Or do the estimator rates change with each scenario or over time? If a certain subset of rates can estimate all other rates across a variety of scenarios, it may have a big impact on rate adaptation algorithms that are based on probing, since they only need to probe this small subset of rates.

7.3 Concluding Remarks

The increasing demand for more bandwidth and contention over shared channels by numerous WiFi devices in close proximity have made the performance of WiFi networks an important research topic. Unfortunately, we have found some serious flaws in the current methodologies used for evaluating the performance of algorithms and systems that depend on 802.11 networks. We show that the experimental evaluation of 802.11 networks should be done with extreme care and provide some guidelines for researchers to consider when conducting experiments. We believe that the Randomized Multiple Interleaved Trials (RMIT) methodology *must be used* for all experiments that involve comparing two or more competing alternatives.

The performance evaluation of 802.11 systems and algorithms is usually done in a limited number of scenarios, sometimes not representative of those in which WiFi devices are actually used. As a result, papers often report that an algorithm or system A is better than B, while another study reports the opposite. We believe that our trace-based simulation framework should become the new standard to be used when comparing the performance of competing alternatives. Our vision is to create a large repository of traces collected in a wide variety of scenarios, so that researcher all over the world can contribute to and use these traces. When a sufficiently large repository is created, researchers can evaluate their new algorithms or systems in a variety of scenarios without having to conduct empirical evaluations. In addition, they can quickly and easily compare the performance of their new algorithm or system with many previously evaluated systems.

We believe that the results of our characterization study can be used in designing new optimization algorithms to shrink the size of their search space. Every new 802.11 standard increases the number of supported transmission rates (e.g., 1,152 rates in 802.11ax) which makes optimizing the transmission features more difficult. The relationships we have found between transmission rates can potentially improve the throughput of 802.11 networks by making optimization algorithms more efficient. Furthermore, we hope that the repository of traces could be used for future characterization studies to better understand how devices are used in practice.

Finally, we point out a relatively simple but powerful technique used in this thesis. In several parts of this thesis, we use mechanisms that perform different measurements using a round-robin sampling of different alternatives. In the trace collection techniques developed for T-RATE, T-SIMn, and the 802.11n characterization study, transmission rates are sampled in a round-robin ordering. In our RMIT experimental evaluation methodology, competing alternatives are measured in a (randomized) round-robin ordering. In the former cases, we switch to the next transmission rate on the order of microseconds, while in the latter case we switch to the next alternative on the order of minutes. However, the same fundamentally important intuition applies in all of these cases, which is exposing all transmission rates or alternatives to similar channel conditions to allow examination and comparison of their performance.

References

[1] ABI Reserach report. https://www.abiresearch.com/press/abi-research-anticipates-more-20-billion-cumulativ. Accessed: 2017-04-21.

[2] Global WiFi market report. http://www.marketsandmarkets.com/PressReleases/global-wi-fi.asp. Accessed: 2017-03-15.

[3] G*Power: Statistical Power Analyses for Windows and Mac. http://www.gpower.hhu.de/en.html. Accessed: 2016-03-27.

[4] IEEE 802.11g-2003. https://en.wikipedia.org/wiki/IEEE_802.11g-2003. Accessed: 2016-02-28.

[5] NS-2. http://www.isi.edu/nsnam/ns/. Accessed: 2016-03-10.

[6] NS-3. https://www.nsnam.org/. Accessed: 2016-03-10.

[7] NS3 current development. https://www.nsnam.org/wiki/Current_Development. Accessed: 2016-03-10.

[8] OMNeT++. https://omnetpp.org/. Accessed: 2016-03-10.

[9] OPNET. http://www.riverbed.com/products/steelcentral/opnet.html?redirect=opnet. Accessed: 2016-03-10.

[10] QualNet. http://web.scalable-networks.com/content/qualnet. Accessed: 2016-03-10.

[11] WiFi service market report. http://www.marketsandmarkets.com/PressReleases/wi-fi-as-a-service.asp. Accessed: 2017-03-15.

[12] Ali Abedi and Tim Brecht. T-RATE: A framework for the trace-driven evaluation of 802.11 rate adaptation algorithms. In *MASCOTS*, 2014.

[13] Ali Abedi and Tim Brecht. Examining relationships between 802.11n physical layer transmission feature combinations. In *MSWiM*, 2016.

[14] Ali Abedi and Tim Brecht. Conducting repeatable experiments in highly variable cloud computing environments. In *ICPE*, 2017.

[15] Ali Abedi, Tim Brecht, and Andrew Heard. T-SIMn: A trace collection and simulation framework for 802.11n networks. *Computer Communications*, (to appear) 2017.

[16] Ali Abedi, Andrew Heard, and Tim Brecht. Conducting repeatable experiments and fair comparisons using 802.11n MIMO networks. *SIGOPS Oper. Syst. Rev.*, 49(1), 2015.

[17] Ali Abedi, Andrew Heard, and Tim Brecht. T-SIMn: Towards the high fidelity trace-based simulation of 802.11n networks. In *MSWiM*, 2016.

[18] P.A.K. Acharya, A. Sharma, E.M. Belding, K.C. Almeroth, and K. Papagiannaki. Congestion-aware rate adaptation in wireless networks: A measurement-driven approach. In *SECON*, 2008.

[19] Mona Aghababaeetafreshi, Lasse K. Lehtonen, Toni Levanen, Mikko Valkama, and Jarmo Takala. IEEE 802.11ac MIMO transceiver baseband processing on a VLIW processor. *Journal of Signal Processing Systems*, 2016.

[20] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, 2004.

[21] AirMagnet. Fluke networks. http://www.flukenetworks.com/enterprise-network/wireless-network/AirMedic.

[22] Jeffrey G Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking*. Pearson Education, 2007.

[23] L. Angrisani and M. Vadursi. Cross-layer measurements for a comprehensive characterization of wireless networks in the presence of interference. *IEEE Transactions on Instrumentation and Measurement*, 2007.

[24] Mustafa Y. Arslan, Konstantinos Pelechrinis, Ioannis Broustis, Srikanth V. Krishnamurthy, Sateesh Addepalli, and Konstantina Papagiannaki. Auto-configuration of 802.11n WLANs. In *Co-NEXT*, 2010.

[25] Mahdi Asadpour, Domenico Giustiniano, Karin Anna Hummel, and Simon Heimlicher. Characterizing 802.11n aerial communication. In *ANC*, 2013.

[26] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Towards predictable datacenter networks. In *SIGCOMM*, 2011.

[27] Hitesh Ballani, Keon Jang, Thomas Karagiannis, Changhoon Kim, Dinan Gunawardena, and Greg O'Shea. Chatty tenants and the cloud network sharing problem. In *NSDI*, 2013.

[28] Sanjit Biswas, John Bicket, Edmund Wong, Raluca Musaloiu-E, Apurv Bhartia, and Dan Aguayo. Large-scale measurements of wireless network behavior. In *SIGCOMM*, 2015.

[29] Bastian Blywis, Mesut Güneş, Felix Juraschek, and Jochen H. Schiller. Trends, advances, and challenges in testbed-based wireless mesh network research. *Mobile Networks and Applications*, 2010.

[30] K. C. Borries, G. Judd, D. D. Stancil, and P. Steenkiste. Fpga-based channel simulator for a wireless network emulator. In *VTC Spring*, 2009.

[31] R. Burchfield, E. Nourbakhsh, J. Dix, K. Sahu, S. Venkatesan, and R. Prakash. RF in the jungle: Effect of environment assumptions on wireless experiment repeatability. In *ICC*, 2009.

[32] Seongho Byeon, Kangjin Yoon, Okhwan Lee, Sunghyun Choi, Woonsun Cho, and Seungseok Oh. MoFA: Mobility-aware frame aggregation in Wi-Fi. In *CoNEXT*, 2014.

[33] Michael Cardosa, Chenyu Wang, Anshuman Nangia, Abhishek Chandra, and Jon Weissman. Exploring MapReduce efficiency with highly-distributed data. In *MapReduce*, 2011.

[34] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 20162021. Technical report, 2017.

[35] Pradipta De, Ashish Raniwala, Rupa Krishnan, Krishna Tatavarthi, Jatan Modi, Nadeem Ahmed Syed, Srikant Sharma, and Tzi Chiueh. Mint-m: An autonomous mobile wireless experimentation platform. In *MobiSys*, 2006.

[36] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. The impact of channel bonding on 802.11n network management. In *CoNEXT*, 2011.

[37] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. Joint rate and channel width adaptation for 802.11 MIMO wireless networks. In *SECON*, 2013.

[38] L. Sandamali Dharmasena, P. Zeephongsekul, and Basil M. de Silva. Sample size determination for kernel regression estimation using sequential fixed-width confidence bands. *PIAENG International Journal of Applied Mathematics*, 38:129–135, 2010.

[39] Benjamin Farley, Ari Juels, Venkatanathan Varadarajan, Thomas Ristenpart, Kevin D. Bowers, and Michael M. Swift. More for your money: Exploiting performance heterogeneity in public clouds. In *SoCC*, 2012.

[40] Sachin Ganu, H. Kremo, R. Howard, and I Seskar. Addressing repeatability in wireless experiments using orbit testbed. In *Tridentcom*, 2005.

[41] Mattew S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2005.

[42] Mattew S. Gast. *802.11n: A Survival Guide*. O'Reilly, 2012.

[43] Domenico Giustiniano and Stefan Mangold. Caesar: Carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *CoNEXT*, 2011.

[44] Robert S. Gray, David Kotz, Calvin Newport, Nikita Dubrovsky, Aaron Fiske, Jason Liu, Christopher Masone, Susan McGrath, and Yougu Yuan. Outdoor experimental comparison of four ad hoc routing algorithms. In *MSWiM*, 2004.

[45] Sarthak Grover, Mi Seon Park, Srikanth Sundaresan, Sam Burnett, Hyojoon Kim, Bharath Ravi, and Nick Feamster. Peeking behind the nat: An empirical study of home networks. In *IMC*, 2013.

[46] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of RF interference on 802.11 networks. *SIGCOMM Comput. Commun. Rev.*, 37, 2007.

[47] D. Gupta, D. Wu, P. Mohapatra, and Chen-Nee Chuah. Experimental comparison of bandwidth estimation tools for wireless mesh networks. In *INFOCOM*, 2009.

[48] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *SIGCOMM*, 2010.

[49] J.A. Hartwell and A. Fapojuwo. Modeling and characterization of frame loss process in IEEE 802.11 wireless local area networks. In *VTC*, 2004-Fall.

[50] Andrew Heard. T-SIMn: Towards a Framework for the Trace-Based Simulation of 802.11n Networks. Master's thesis, University of Waterloo, 2016.

[51] C. Hepner, A. Witt, and R. Muenzner. In depth analysis of the ns-3 physical layer abstraction for WLAN systems and evaluation of its influences on network simulation results. In *SInCom*, 2015.

[52] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *MobiCom*, 2001.

[53] A. Iordache, C. Morin, N. Parlavantzas, E. Feller, and P. Riteau. Resilin: Elastic MapReduce over multiple clouds. In *CCGrid*, 2013.

[54] IPerf. http://sourceforge.net/projects/iperf/.

[55] D. Jiang, B.C. Ooi, L. Shi, and S. Wu. The performance of MapReduce: An in-depth study. *Proc. VLDB Endow.*, 3(1-2):472–483, 2010.

[56] David Johnson and Albert Lysko. Comparison of MANET routing protocols using a scaled indoor wireless grid. *Mobile Networks and Applications*, 2008.

[57] Glenn Judd and Peter Steenkiste. Repeatable and realistic wireless experimentation through physical emulation. *SIGCOMM Comput. Commun. Rev.*, 2004.

[58] Glenn Judd and Peter Steenkiste. A simple mechanism for capturing and replaying wireless channels. In *ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, 2005.

[59] Glenn Judd and Peter Steenkiste. Using emulation to understand and improve wireless networks and applications. In *NSDI*, 2005.

[60] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *MobiSys*, 2008.

[61] Hakyung Jung, Ted “Taekyoung” Kwon, Kideok Cho, and Yanghee Choi. REACT: Rate Adaptation using Coherence Time in 802.11 WLANs. *Computer Communications*, 34(11), 2011.

[62] Gideon Juve, Ewa Deelman, G. Bruce Berriman, Benjamin P. Berman, and Philip Maechling. An evaluation of the cost and performance of scientific workflows on Amazon EC2. *J. Grid Comput.*, 10(1), 2012.

[63] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[64] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *INFOCOM*, 2006.

[65] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM*, 2004.

[66] L. Kriara, M.K. Marina, and A. Farshad. Characterization of 802.11n wireless LAN performance via testbed measurements and statistical analysis. In *SECON*, 2013.

[67] Lito Kriara and Mahesh K Marina. Samplelite: A hybrid approach to 802.11n link adaptation. *ACM SIGCOMM Computer Communication Review*, 2015.

[68] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 rate adaptation: a practical approach. In *MSWiM*, 2004.

[69] Katrina LaCurts and Hari Balakrishnan. Measurement and analysis of real-world 802.11 mesh networks. In *IMC*, 2010.

[70] S. Lakshmanan, S. Sanadhya, and R. Sivakumar. On link rate adaptation in 802.11n WLANs. In *INFOCOM*, 2011.

[71] J.-H. Lee and C.-C. Cheng. Spatial correlation of multiple antenna arrays in wireless communication systems. In *Progress In Electromagnetics Research*, volume 132, 2012.

[72] K.C. Lee, J.M. Navarro, T.Y. Chong, Uichin Lee, and M. Gerla. Trace-based evaluation of rate adaptation schemes in vehicular environments. In *Proceeding of VTC '10*, pages 1 –5, 2010.

[73] M.R. Leek, T.E. Hanna, and L. Marshall. An interleaved tracking procedure to monitor unstable psychometric functions. *The Journal of the Acoustical Society of America*, 90(3):1385–1397, 1991.

[74] Vincent Lenders and Margaret Martonosi. Repeatable and realistic experimentation in mobile wireless networks. *IEEE Transactions on Mobile Computing*, 2009.

[75] Chi-Yu Li, Chunyi Peng, Songwu Lu, and Xinbing Wang. Energy-based rate adaptation for 802.11n. In *Mobicom*, 2012.

[76] E.O. Lillie, B. Patay, J. Diamant, B. Issell, E.J. Topol, and N.J. Schork. The n-of-1 clinical trial: the ultimate strategy for individualizing medicine? *Personalized Medicine*, 8(2):161–173, 2011.

[77] Ren Ping Liu, G.J. Sutton, and I.B. Collings. A new queueing model for QoS analysis of IEEE 802.11 DCF with finite buffer and load. *Wireless Communications, IEEE Transactions on*, 2010.

[78] J. Lorincz and D. Begusic. Physical layer analysis of emerging IEEE 802.11n WLAN standard. In *8th International Conference Advanced Communication Technology*, 2006.

[79] Aniket Mahanti, Niklas Carlsson, Carey Williamson, and Martin Arlitt. Ambient interference effects in WiFi networks. In *NETWORKING*, 2010.

[80] Piyush Mehrotra, Jahed Djomehri, Steve Heistand, Robert Hood, Haoqiang Jin, Arthur Lazanoff, Subhash Saini, and Rupak Biswas. Performance evaluation of Amazon EC2 for NASA HPC applications. In *ScienceCloud*, 2012.

[81] 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*, 2011.

[82] D.C. Montgomery. *Design and Analysis of Experiments*. Wiley, 2012.

[83] Calvin Newport, David Kotz, Yougu Yuan, Robert S. Gray, Jason Liu, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM*, 2004.

[84] Giao T Nguyen and Brian Noble. A trace-based approach for modeling wireless channel behavior. In *Simulation*, 1996.

[85] Konstantinos Nikitopoulos, Juan Zhou, Ben Congdon, and Kyle Jamieson. Geosphere: Consistently turning MIMO capacity into throughput. In *SIGCOMM*, 2014.

[86] Brian D. Noble, M. Satyanarayanan, Giao T. Nguyen, and Randy H. Katz. Trace-based mobile network emulation. In *SIGCOMM*, 1997.

[87] S. Papanastasiou, J. Mittag, E. G. Strom, and H. Hartenstein. Bridging the gap between physical layer emulation and network simulation. In *WCNC*, 2010.

[88] U. Paul, R. Crepaldi, Jeongkeun Lee, Sung-Ju Lee, and R. Etkin. Characterizing WiFi link performance in open outdoor networks. In *SECON*, 2011.

[89] Ioannis Pefkianakis, Yun Hu, Starsky H.Y. Wong, Hao Yang, and Songwu Lu. MIMO rate adaptation in 802.11n wireless networks. In *MobiCom*, 2010.

[90] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei. WiFi can be the weakest link of round trip network latency in the wild. In *INFOCOM*, pages 1–9, 2016.

[91] Ramjee Prasad, Sudhir Dixit, Richard van Nee, and Tero Ojanpera. *Globalization of mobile and wireless communications*. Springer Netherlands, 2011.

[92] Hariharan Rahul, Farinaz Edalat, Dina Katabi, and Charles G. Sodini. Frequency-aware rate adaptation and mac protocols. In *Mobicom*, 2009.

[93] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[94] Lenin Ravindranath, Calvin Newport, Hari Balakrishnan, and Samuel Madden. Improving wireless network performance using sensor hints. In *USENIX*, 2011.

[95] Shravan Rayanchu, Ashish Patro, and Suman Banerjee. Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. In *IMC*, 2011.

[96] O. Rensfelt, F. Hermans, P. Gunningberg, and L.-A Larzon. Repeatable experiments with mobile nodes in a relocatable WSN testbed. In *DCOSSW*, 2010.

[97] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, and John Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *E-WIND*, 2005.

[98] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate Ad Hoc networks. In *MobiCom*, 2002.

[99] Jörg Schad, Jens Dittrich, and Jorge-Arnulfo Quiané-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proceedings of the VLDB Endowment*, 3(1-2):460–471, 2010.

[100] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. AccuRate: Constellation based rate estimation in wireless networks. In *NSDI*, 2010.

[101] Wei-Liang Shen, Yu-Chih Tung, Kuang-Che Lee, Kate Ching-Ju Lin, Shyamnath Gollakota, Dina Katabi, and Ming-Syan Chen. Rate adaptation for 802.11 multiuser MIMO networks. In *Mobicom*, 2012.

[102] M. Stoffers and G. Riley. Comparing the ns-3 propagation models. In *MASCOTS*, 2012.

[103] Srikanth Sundaresan, Nick Feamster, and Renata Teixeira. *Home Network or Access Link? Locating Last-Mile Downstream Throughput Bottlenecks*. 2016.

[104] Elliott Suthers. 2016 WiFi shippment report. https://www.wi-fi.org/news-events/newsroom/wi-fi-device-shipments-to-surpass-15-billion-by-end-of-2016.

[105] Mineo Takai, Jay Martin, and Rajive Bagrodia. Effects of wireless physical layer modeling in mobile Ad Hoc networks. In *MobiHoc*, 2001.

[106] TCPDUMP. http://www.tcpdump.org/.

[107] Xiaozheng Tie, Anand Seetharam, Arun Venkataramani, Deepak Ganesan, and Dennis L. Goeckel. Anticipatory wireless bitrate control for blocks. In *CoNEXT*, 2011.

[108] Jiann-An Tsai, R.M. Buehrer, and B.D. Woerner. The impact of aoa energy distribution on the spatial fading correlation of linear antenna array. In *Vehicular Technology Conference*, 2002.

[109] Richard Van Nee, V. K. Jones, Geert Awater, Allert Van Zelst, James Gardner, and Greg Steele. The 802.11n MIMO-OFDM standard for wireless LAN and beyond. *Wireless Personal Communications*, 2006.

[110] T. Vilches and D. Dujovne. Gnuradio and 802.11: performance evaluation and limitations. *IEEE Network*, 2014.

[111] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-layer wireless bit rate adaptation. In *SIGCOMM*, 2009.

[112] Tao Wang, Guangyu Sun, Jiahua Chen, Jian Gong, Haoyang Wu, Xiaoguang Li, Songwu Lu, and Jason Cong. GRT: A Reconfigurable SDR Platform with High Performance and Usability. *SIGARCH Comput. Archit. News*, 2014.

[113] Robert Wilson. Propagation losses through common building materials, 2.4 GHz vs 5 GHz. http://www.am1.us/Protected_Papers/E10589_Propagation_Losses_2_and_5GHz.pdf. Accessed: 2016-05-12.

[114] Starsky H. Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *MobiCom*, 2006.

[115] Yunze Zeng, P.H. Pathak, and P. Mohapatra. A first look at 802.11ac in action: Energy efficiency and interference characterization. In *IFIP*, 2014.

[116] Jiansong Zhang, K. Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. A Practical SNR-Guided Rate Adaptation. In *INFOCOM*, 2008.

[117] Zengbin Zhang, Xia Zhou, Weile Zhang, Yuanyang Zhang, Gang Wang, Ben Y. Zhao, and Haitao Zheng. I am the antenna: accurate outdoor AP location using smartphones. In *MobiCom*, pages 109–120, 2011.

[118] Z. Zhou, K. Ishizawa, H. Kikuchi, M. Sengoku, and Y. Onozato. Generalized spatial correlation equations for antenna arrays in wireless diversity reception exact and approximate analyses. In *The Neural Networks and Signal*, 2003.