# Predicting home Wi-Fi QoE from passive measurements on commodity access points

Diego Neves Da Hora

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité : "Informatique et Réseaux"

École doctorale : EDITE

réalisée

à l'INRIA-Paris

présentée par

Diego NEVES DA HORA

pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

# Predicting Home Wi-Fi QoE from Passive Measurements on Commodity Access Points

soutenue le 27/04/2017

devant le jury composé de :

| | | |
|---|---|---|
| Mme. | Isabelle Guérin Lassous | Rapporteur |
| M. | Salah Eddine el Ayoubi | Rapporteur |
| M. | Serge Fdida | Examinateur |
| Mme. | Renata Teixiera | Directeur de thèse |
| M. | Karel van Doorselaer | Co-Directeur de thèse |

# Contents

# Chapter 1

# Introduction

Recent years have seen widespread penetration of residential broadband internet access [4]. Many modern consumer electronics use internet access such as laptops, smartphones, video-games and smart-tvs. A home network connects these devices to the internet. Figure 1.1 illustrates a typical home network setup. Devices can connect to a home access point (AP) using wired (e.g. Ethernet, PLC) or wireless (most often Wi-Fi). Users obtain internet access by signing a contract with an Internet Service Provider (ISP) and the home AP connects to the ISP network through the *access link*. More complex home network scenarios exist: another networking device can have the access link or multiple APs can provide access to the home network. In this example, one AP both connects Wi-Fi devices to the home network and provides internet access. This gives the home AP a centralized view of all devices in the home.

Wi-Fi is the preferred technology to implement home networks. The majority of mobile devices, such as tablets and smartphones, can only communicate over wireless and prioritize using the Wi-Fi network. Wired connectivity requires physical installation and gateways often have a limited number of ports, whereas many devices can connect to the same Wi-Fi network. The average home network has more stations connected through Wi-Fi than Ethernet and only a minority of homes uses all available Ethernet ports [39]. Wi-Fi traffic is predicted to grow to almost half (49%) of total IP traffic by 2020 from 42% in 2015 [55].

Wi-Fi networks can disrupt end-to-end application performance. Wi-Fi networks use radio bands reserved internationally for radio frequency use. Since many other radio applications share the radio band, electromagnetic interference may disrupt Wi-Fi traffic. In the 2.4Ghz band, for example, non Wi-Fi radio frequency (RF) devices such as microwaves, cordless phones and wireless cameras can severely degrade Wi-Fi

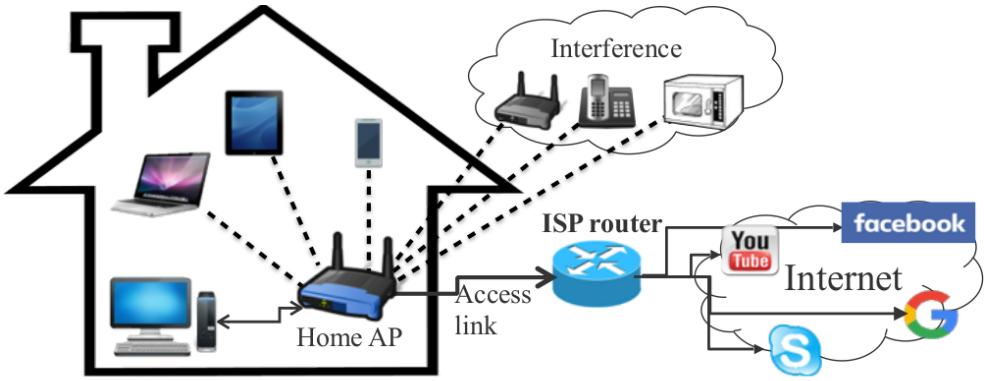



Figure 1.1: Example of home network access to internet applications.

performance [96]. Other Wi-Fi devices also compete for medium access, although the medium access control (MAC) protocol coordinates medium sharing between stations. In dense urban neighborhoods it is typical to see tens of competing Wi-Fi networks [91], alongside many non Wi-Fi RF devices [96]. In these cases, the Wi-Fi performance can vary. A poorly located AP may leave stations with weak signal, degrading performance as they communicate through low speed links.

Poor end-to-end application performance can degrade users' Quality-of-Experience (QoE). The concept of QoE combines user perception, experience, and expectations with quality of service (QoS) metrics. Figure 1.2 shows the relationship between network and application QoS and QoE. Network-layer QoS problems (e.g. throughput bottlenecks, packet losses, jitter) can degrade application-layer QoS metrics (e.g. video bitrate, page load time, audio artifacts), which in turn affects users' quality perception. In Wi-Fi networks, poor Wi-Fi quality can degrade network QoS. Several studies have shown that network QoS and application QoS metrics can have a considerable impact on QoE [36].

Diagnosing QoE degradations is challenging. The end-to-end path between the user device and application server traverses different networks, each of which can impair performance and degrade QoE. For example, if a YouTube video shows poor image quality and frequently stalls, the problem can be located in the home network, in the access link, in the ISP network, or the server network itself. Users often do not know whether there is a problem in the home, ISP or server network. Moreover, end users often blame the ISP for any problem in their internet connectivity, including the home Wi-Fi network. A practical way to diagnose QoE degradations is to monitor QoS metrics. ISPs spend considerable effort monitoring their network, while content /

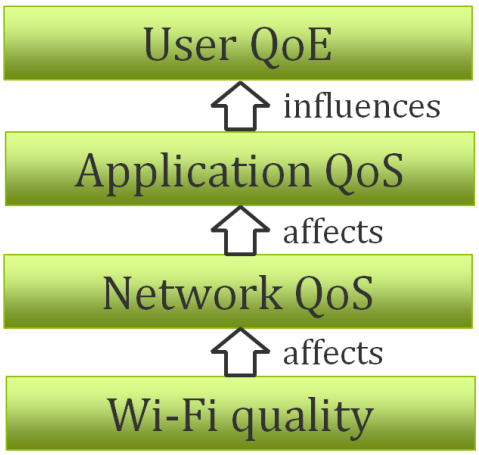



Figure 1.2: Relationship between QoS at different layers and QoE on Wi-Fi networks.

application providers monitor the service side. Home networks, however, are manage by end users who most often have limited to none network management expertise [33]. Monitoring home Wi-Fi quality is sufficient to detect most performance problems located in the home network, since wired connectivity is rarely problematic. The ISP in many cases provides and control the home AP, and it can use it to monitor Wi-Fi quality on behalf of users.

This thesis designs and evaluates techniques to passively monitor Wi-Fi quality on commodity access points and predict when Wi-Fi quality degrades internet application QoE. The home AP is the ideal monitoring vantage point since it has visibility on most home Wi-Fi devices and is often controlled by the ISP. In our approach, the home AP periodically makes Wi-Fi quality measurements at every ISP's customers. A Wi-Fi monitoring system collects AP measurements and infers Wi-Fi quality per station over time. We propose methods to infer Wi-Fi link capacity and available bandwidth from Wi-Fi QoS metrics. Since users typically only notice network problems when internet application QoE degrades, we also propose predictors of the effect of Wi-Fi quality on QoE. Our goal is to build a system that ISPs can deploy at scale on commodity APs. This goal brings some restrictions. We assume commodity APs that ISPs can deploy at no additional cost. This excludes solutions that rely on APs with specialized hardware or multiple Wi-Fi interfaces. We also exclude solutions that use other monitoring hosts within the home network. We want to avoid any disruption to user activities. This precludes solutions that do not require any installation on end users' devices. We avoid creating traffic in the home network, and create little access link traffic overhead. We do not use active measurements, since they may disrupt user traffic. The APs only

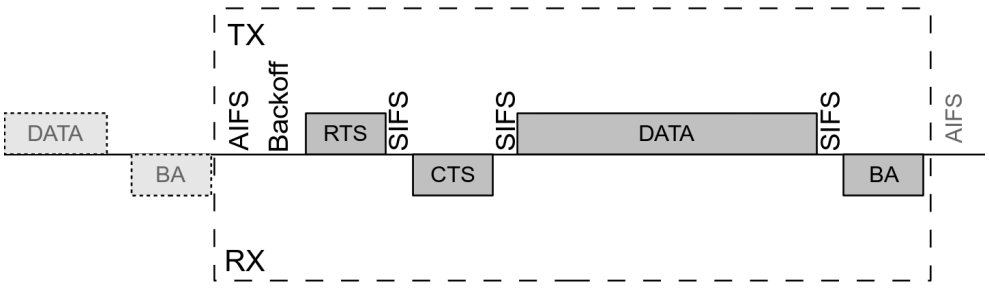



Figure 1.3: A-MPDU frame exchange with RTS/CTS protection

send aggregated statistics of measured Wi-Fi metrics in order to reduce overhead.

The rest of this chapter introduces the background and challenges on measuring Wi-Fi quality and predicting Quality-of-Experience. Then, we present the research contributions of this thesis, followed by the thesis outline.

## 1.1 Wi-Fi networks

### 1.1.1 Background

**802.11 MAC protocol.** The IEEE 802.11 Medium Access protocol uses carrier sense multiple access with collision avoidance (CSMA/CA). Nodes only transmit after they sense the medium idle for the duration of Arbitration Interframe Space (AIFS) plus a backoff timer. If Clear Channel Assessment detects the medium as busy (due to presence of Wi-Fi or non-Wi-Fi signal), then nodes exponentially increase the contention window size, and defer transmission for a random backoff counter, chosen between 0 and the contention window size. Frames need to be acknowledged by the receiver through an ACK frame, otherwise they are retransmitted, and each transmission is spaced by a Short Interframe Space (SIFS). The Request to Send (RTS) and Clear to Send (CTS) handshake is an optional mechanism used to reduce frame collisions and to mitigate the hidden node problem. It uses the network allocation vector in RTS/CTS messages to indicate for how long the medium will be busy due to impending transmissions.

**Medium Sharing and Frame Aggregation.** 802.11n introduces frame aggregation, which reduces MAC overhead by allowing delivery of multiple aggregated Mac Protocol Data Units (A-MPDUs) in a single medium access. The number of A-MPDUs per medium access depends on the PHY rate used and we represent this number by AGG(P), for a given PHY rate, $P$. The maximum A-MPDU size used by the transmit-

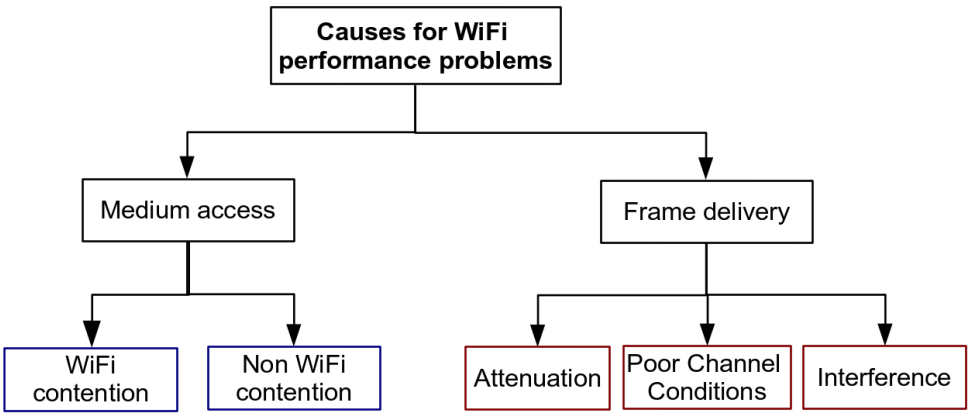



Figure 1.4: Causes of Wi-Fi performance problems

ter ($MAX_{agg}$) is implementation dependent, but it is mainly bounded by the maximum Rx A-MPDU length advertised by the receiver on control messages, which can assume the values: 8, 16, 32 or 64 Kbytes. Figure 1.3 illustrates a typical A-MPDU exchange with RTS/CTS protection.

**Physical rate and rate adaptation algorithm.** An 802.11n device is able to use any of the physical (PHY) rates introduced by 802.11n as well as legacy 802.11g and 802.11b rates. The maximum PHY rate for 802.11n using one antenna and 20MHz channel bandwidth and long guard interval is 65Mbps. Devices with two or more antennas can increase the PHY rate with Multiple-Input-Multiple-Output (MIMO). It is also possible to double PHY rate using 40Mhz channel bandwidth. Channel conditions determine the best PHY rate to use: if they are good, higher PHY rates maximize performance; otherwise, lower PHY rates increase the probability of reception. The rate adaptation algorithm is responsible for selecting the PHY rate for each frame. Even though rate adaptation algorithms are not specified by the standard, popular implementations are Adaptive Multi-Rate Retry, Onoe and Sample Rate [130].

### 1.1.2 Performance impairments in Wi-Fi networks

We classify performance impairments in Wi-Fi networks onto two broad classes according to their causes: *medium access* problems, which comprise problems that prevent Wi-Fi stations from accessing the medium; and *frame delivery* problems, which comprise problems that reduce the efficiency of frame delivery. Figure 1.4 summarizes the taxonomy of causes for Wi-Fi performance problems. Multiple problems can simultaneously impair Wi-Fi network performance. *Wi-Fi contention* occurs in Wi-Fi

channels with many active stations or when a sub-set of stations occupy a significant amount of airtime. The rate anomaly problem, presented by Heusse et al. [41], is a type of Wi-Fi contention problem where one station greatly reduces others' available bandwidth. *Non Wi-Fi contention* occurs when a radio frequency device, such as a microwave, cordless phone, or wireless camera, blocks the Wi-Fi channel for a large fraction of time. Rayanchu et al. [96] proposed a method to detect the presence of non Wi-Fi RF devices. Their results show that non Wi-Fi RF devices can limit throughput and increase MAC layer losses. Frame delivery problems occur when stations use the airtime available inefficiently. *Attenuation* causes stations to receive frames with low signal, forcing the use of lower but more robust PHY rates to ensure reception. For example, a station which only receives frames at PHY rates 6.5 Mbps and below could receive over 20× higher throughput at PHY rate 130 Mbps. *Poor channel conditions* causes frame errors at some PHY rate modulations, preventing efficient frame delivery. For example, some Wi-Fi links show high frame loss when using MIMO, due to the lack of spatial diversity. Frequency selective fading also causes high bitrate error at some subcarriers [128]. *Interference* from Wi-Fi and non Wi-Fi devices can also cause frame delivery problems. This happens when the transmitter's CCA cannot detect but there is interference affecting the receiver, causing collisions and frame losses. Due to the ambiguity between collisions and low signal, the rate adaptation algorithm often interprets frame losses as low signal and decreases the PHY rate, lowering frame delivery efficiency [95]. An example is the hidden terminal problem, which is partially resolved using RTS/CTS protection.

## 1.2 Measuring Wi-Fi quality

We can quantify Wi-Fi quality by monitoring metrics at different layers. We can infer Wi-Fi quality problems by measuring network QoS metrics. Alternatively, link-layer metrics can also indicate Wi-Fi quality.

The IETF has several metrics for quantifying end-to-end network performance, such as bandwidth, network delays and packet losses [90]. In Wi-Fi networks, we highlight three key IP-layer network performance metrics. The Wi-Fi link's *available bandwidth* represents the IP datarate the Wi-Fi link can sustain over a time interval. Available bandwidth helps identify, for example, if the Wi-Fi link bottlenecks internet access. The Wi-Fi link *packet loss* is the fraction of IP packets lost in a Wi-Fi link due to bottlenecks or frame losses. Frame losses due to collision or low signal are common

occurrences in Wi-Fi networks. MAC layer retransmission avoids IP packet losses, but successive frame losses causes IP packet losses. The Wi-Fi link *latency* is the one-way delay of IP packet transmissions in the Wi-Fi link. Busy Wi-Fi channels incur additional Wi-Fi *link latency*. Frame retransmissions increase latency and introduce jitter.

Link-layer metrics can also indicate Wi-Fi network quality. The received signal strength indicator (RSSI) measures the signal strength of frames the AP receives from the station and is an indication of attenuation. A Wi-Fi link that uses high PHY rates during data transfers typically represents good Wi-Fi quality. Wi-Fi links with low frame delivery ratio (FDR) experience jitter and reduced bandwidth. Some Wi-Fi cards can breakdown airtime availability into the fraction of time the CCA detects the medium as available, busy due to Wi-Fi traffic from other devices, or busy due to non Wi-Fi RF devices. These metrics detect the existence of Wi-Fi and non Wi-Fi contention.

We can measure network performance metrics using active or passive measurements. Active measurements send probes to quantify network performance. We can use ICMP pings, for example, to measure link latency and packet loss. Passive measurements passively observe link-layer and network metrics, without injecting probes. We can measure these metrics from the home AP, end user devices or other vantage point.

Our goal of developing a practical solution that ISPs can deploy at no additional cost at scale brings some restrictions. Active measurement solutions [67, 117] can help in on-demand Wi-Fi diagnosis, but they are unfit for large scale ISP deployments, since continuous active measurements may disrupt users' traffic and drain stations battery. The solution must use a passive measurement approach. Passively measuring some network QoS metrics requires packet inspection. For example, we can measure Wi-Fi link round-time-trip (RTT) as the elapsed time between data segments and corresponding ACKs on TCP traffic to the station. Although previous studies have instrumented APs to infer IP-layer QoS metrics [114, 89, 92], we avoid per-packet processing since the processing overhead can affect network performance on commodity APs. Finally, we want our solution to work on commercial APs, so we rely on standard link-layer metrics already exposed by Wi-Fi drivers. This excludes solutions with additional specialized hardware, multiple Wi-Fi cards, or driver modifications [76, 89].

We propose to sample link-layer Wi-Fi metrics to infer Wi-Fi quality. We define the *Wi-Fi link capacity* as the maximum UDP throughput a device can achieve assuming

full medium availability. We develop a method for estimating the Wi-Fi link capacity from passive link-layer Wi-Fi metrics and show that we can use the link capacity in combination with counters of medium availability to estimate the Wi-Fi link available bandwidth. We also propose a method that uses the Wi-Fi link capacity to diagnose Wi-Fi performance problems into medium access and/or frame delivery problems.

## 1.3 Quality-of-Experience

The research community has made significant efforts into defining and understanding Quality-of-Experience (QoE). While there is some consensus on general rules and best practices, there is no agreed framework to study and understand user QoE. In this section, we give background on QoE research and define the QoE terminology we adopt.

**QoE definition.** We use the QoE definition proposed by ITU Telecommunication Standardization Section (ITU-T) as "the overall acceptability of an application or service, as perceived subjectively by the end-user", which "includes the complete end-to-end system effects" and "may be influenced by user expectations and context" [62]. We refer to this as user QoE or simply QoE.

A number of authors agree that many factors influence QoE. Skorin-Kapov et al. [106] classify QoE influence factors into application, resource, context and user space using the *ARCU* model. We adopt the terminology proposed by Reiter et al. [98], which classify QoE influence factors between human, context, and system influence factors. While authors may differ on how to classify influence factors, they often agree on what factors may influence QoE. A few examples of QoE influence factors are user demographics (e.g., age, gender, cultural background), the physical context (e.g., at home/work/airport), and user device (e.g., mobile/stationary, screen size).

These influence factors are inter-dependent. For example, Balanchandran et al. [32] show that, for the majority of cases, video bitrate has a strong and positive relationship with QoE, except during live sports events, where users may play the video in background. Given the existence of these numerous influence factors, some of which are hardly measurable, it is very difficult to study and measure QoE. Most QoE studies are, thus, limited to one or a few key influence factors. In this dissertation, we focus on the Wi-Fi quality and how it affects QoE on web browsing, YouTube, audio and video RTC.

Detecting when Wi-Fi quality degrades QoE using passive Wi-Fi metrics is chal-

lenging. First, we have no information about the applications users are running at any given time. ISPs avoid looking into user traffic because of privacy considerations. Thus, we must estimate the effect of Wi-Fi quality on QoE of popular applications, which most customers are likely to run. This thesis details the effect of Wi-Fi quality on four popular internet applications: web browsing, YouTube, audio and video real time communication (RTC). Second, factors other than the Wi-Fi quality may degrade user QoE (e.g., poor Internet performance or an overloaded server). While monitoring any QoE degradation is desirable, the AP vantage point prevents us from having the end-to-end view necessary for this task. Finally, Wi-Fi metrics available in APs are coarse aggregates such as the average PHY rate or the fraction of busy times. It is unclear how to map these coarse metrics into QoE.

## 1.4 Contributions

This thesis makes the following contributions.

1. **Design and evaluation of a method for passive Wi-Fi link capacity estimation.**

   We define and validate simple models to estimate Wi-Fi link capacity in 802.11n networks. Previous models [66] are for 802.11a/b/g, which did not have frame aggregation. First, we define a model under the assumptions of fixed PHY rate and no frame losses. Then, we extend this model to the realistic case where the PHY rate varies and frame losses occur. We validate our model with controlled experiments. Our results show that we can estimate the Wi-Fi link capacity with errors similar to the state of the art, but without the need to tune station-specific parameters. Finally, we show that the Wi-Fi link capacity can help us distinguish between medium access and frame delivery problems with two case studies.

2. **Design and evaluation of predictors for the effect of Wi-Fi quality on QoE for web browsing, YouTube, audio and video RTC.**

   We show how to use passive link-layer Wi-Fi metrics available in commodity APs to predict the effect of Wi-Fi quality on QoE. Evaluating the effect of Wi-Fi quality on QoE on a great number of Wi-Fi scenarios is challenging. We perform automated controlled experiments on a great number of Wi-Fi scenarios in a Wi-Fi testbed and monitor application QoS metrics. We avoid intrusive user studies by using state of the art QoS / QoE models to map application QoS

into estimated QoE. We use support vector regression (SVR) to train predictors, and show that Tx PHY rate, frame delivery ratio, and counters of Wi-Fi and non Wi-Fi occupation can predict estimated degradation mean opinion scores (DMOS, in range [1-5]) with root mean square errors between 0.643 and 0.777. We validate predictors with experiments in an uncontrolled environment, under real Wi-Fi conditions, and we observe root-mean square errors between 0.740 and 0.938. These results allow ISPs to remotely monitor Wi-Fi quality at commodity APs and identify when Wi-Fi quality degrades QoE.

3. **Design of a method to identify poor QoE events.**

   We leverage the QoE predictors for web, YouTube, and audio and video RTC to identify instances where Wi-Fi metrics indicates that user experience should be poor across applications. We use K-means clustering to identify groups of Wi-Fi quality metrics that capture different QoE behavior. In particular, we identify clusters where all applications perform poorly. We then classify poor QoE events as short, consistent, and intermittent poor QoE events. We evaluate how often stations show each kind of poor QoE event and find that the majority of stations presents a mix between short, intermittent, and consistent poor QoE events, but stations with a larger fraction of poor QoE samples contain more samples on consistent poor QoE events.

4. **Characterization of Wi-Fi quality in the wild.**

   We analyze data from 6086 stations connected to 832 APs of customers of a large Asian-Pacific residential ISP, to characterize Wi-Fi quality in the wild. We find that Wi-Fi quality is generally good for internet applications. In the large majority of collected samples, Wi-Fi does not impair application quality. We classify only 9% of all samples as poor QoE samples, and 60% of stations have less than 5% poor QoE samples. We also find that link utilization is below 10% on 90% of all samples. We also observe instances of reduced Wi-Fi quality. We find that Avg Tx PHY rate is below $6.5 Mbps$ on 8.9% of all samples, and that medium occupancy is above 50% on 7.4% of all Wi-Fi samples. We find that 10.4% of stations have more than 25% poor QoE samples, with some poor QoE events stretching over many hours. The existence of a large number of short and intermittent poor QoE events highlights the necessity of continual monitoring to detect and diagnose Wi-Fi quality.

## 1.5 Thesis outline

This thesis is organized as follows. Chapter 2 presents related work on Wi-Fi networks and QoE of internet applications. Chapter 3 designs and evaluates the method for estimating station link capacity from passive Wi-Fi metrics. Chapter 4 builds predictors of the Wi-Fi quality effect on QoE. Chapter 5 applies the link capacity method and QoE predictors to characterize Wi-Fi quality in the wild. We conclude in Chapter 6.

# Chapter 2

# Related work

In this chapter presents, we review the state of the art on monitoring, characterization, and diagnosis of Wi-Fi networks. Then, we review the methodologies for studying Quality of Experience, as well as QoS/QoE models for several internet applications.

## 2.1 Wi-Fi networks

In this section, we discuss related work on Wi-Fi network link capacity and bandwidth estimation. Then we discuss methods for diagnosis performance problems in Wi-Fi networks. Then, we review studies that characterize Wi-Fi performance across different environments.

### 2.1.1 Link capacity and bandwidth estimation

We review the state of the art on methods to measure available bandwidth and estimate Wi-Fi link capacity. Most work on available bandwidth estimation on Wi-Fi networks uses active measurements [77, 75]. While these techniques can estimate link bandwidth, they are unfit for large scale ISP deployments, since active measurements can disrupt users' traffic, drain stations battery and require cooperation between endpoints. There are methods for estimating maximum MAC throughput achievable under ideal Wi-Fi network conditions. These methods does not measure network metrics, only calculate maximum throughput values based on protocol specification. Jun et al. [66] calculate the theoretical maximum throughput of IEEE 802.11a, b and g based on protocol parameters. Skordoulis et al. [105] compare the different frame aggregation mechanisms proposed in IEEE 802.11n, giving simulation results for the maximum throughput. These models are useful for giving link capacity upper bounds, since we can calculate

maximum throughput per device given its Wi-Fi capabilities. These models typically consider fixed PHY rate usage and although we can calculate maximum throughput per PHY rate, PHY rate values vary often in practice, due to the rate adaptation algorithm. Patro et al. [89] proposed *Witt*, which uses passive measurements of PHY rate samples to estimate link capacity and available bandwidth. We propose a hybrid method that combines maximum throughput estimations, considering IEEE 802.11n parameters, and passive PHY rate and frame delivery measurements to estimate link capacity when PHY rate varies. We combine link capacity with counters of medium availability to estimate the Wi-Fi link available bandwidth.

### 2.1.2 Diagnosis

Diagnosing the root cause of Wi-Fi problems is a challenging task. We classify Wi-Fi diagnosis methods between active and passive. Active methods employ active measurements to distinguish between possible root causes. Syrigos et al. [117] propose the use of active measurements with fixed PHY rate to diagnose common 802.11 pathologies. Kanuparthy et al. [67] proposes using user-level probing to diagnose common WLAN performance problems. They are appropriate for troubleshooting, when the presence of a Wi-Fi problem is known or expected. They are not fit for continual monitoring, since measurements may disturb local traffic and/or drain battery of Wi-Fi stations. Passive methods overcome these limitations by passively monitoring quality metrics. Rayanchu et al. [95] propose a method to diagnose frame collision from weak signal. This method requires modifications to the Wi-Fi kernel of both the AP and the device. Lakshminarayanan et al. [76] propose using a second NIC to diagnose medium usage, allowing accurate diagnosis of Wi-Fi and non Wi-Fi interference. Shravan et al. [96] developed Airshark, a system which detects different non Wi-Fi interference sources by using energy samples from the Wi-Fi card. These methods are able to diagnose Wi-Fi problems with a great degree of precision, but require a second NIC for dedicated spectrum monitoring, limiting deployment There is work on diagnosing Wi-Fi performance problems in enterprise and campus networks, where multiple APs are managed by the same entity, using a mix of active and passive measurements [24, 11]. These solutions often require combining the multiple points of view to diagnose performance problems, while we can only rely on the single home AP. We propose passive diagnosis of Wi-Fi problems between *medium access* and *frame delivery* problems, as shown in Figure 1.4. Kanuparthy et al. [76] and Syrigos et al. [117] propose similar classification of Wi-Fi performance problems.

### 2.1.3 Characterization

There are prior studies characterizing Wi-Fi performance in the wild using the AP point of view. Pei et al. [92] studies the latency in the Wi-Fi hop using 47 APs deployed in a university campus. They found that the Wi-Fi hop latency is larger than 20ms on more than 50% of TCP packets. Ioannis et al. [91] use passive Wi-Fi performance measurements from 167 homes on a period of four months. They found that a small number of devices accounts for most of the observed traffic, and suggest that most poor performance reports are caused by network interference. Biswas et al. [11] characterize Wi-Fi network usage and performance on over 20 thousand APs in industrial networks. They found that Wi-Fi network usage is growing quickly, the 2.4Ghz spectrum is more saturated than 5.0 Ghz, and that there is no direct relationship between the number of nearby SSIDs and channel utilization. Each of these studies adds to our understanding of Wi-Fi quality in different settings, but none of them addresses the issue of how Wi-Fi quality affects QoE as we do here.

## 2.2 Quality of Experience

In this section, we classify the methods to study QoE. Then, we present background and related work on web browsing, YouTube, and audio and video RTC.

### 2.2.1 QoE assessment methodologies

Studying and measuring QoE is an inherently difficult task, due to its subjective nature. We categorize methods to study QoE into three classes.

**Subjective quality assessment.** In this methodology, researchers evaluate the quality of a system, application or service through the subjective evaluation of users during controlled experiments. This is often the most informative approach, with researchers using questionnaires and interviews to directly obtain user opinion [99]. A common metric for summarizing the opinion of multiple users is the Mean Opinion Score (MOS). We obtain MOS by averaging the opinion of multiple users, informed through a numerical quality scale such as the Absolute Category Rating (ACR), where 1 = Bad, 2 = Poor, 3 = Fair, 4 = Good, 5 = Excellent.

Many influence factors may alter the user quality perception, such as ambient noise, screen quality, or user expectation. This requires researchers to carefully control test conditions and decide which factors to consider in their experiments. This, in addition

to the large number of user opinions necessary to obtain statistically relevant results, makes this methodology expensive and difficult to scale.

Several studies uses subjective quality assessment to provide a mapping between network or application QoS and MOS [99, 2, 5, 34, 122, 123, 124, 125, 61, 36]. Some authors refer to these mappings as QoS/QoE models.

**User engagement.** An alternative to subjective quality assessment is to implicitly collect user feedback through user engagement metrics. Balanchandran et al. [6] studies the effects of video QoS metrics on user video viewing time. They find that users are likely to abandon videos which show a high number of buffering events or have low bitrate. These finding were later confirmed by subjective studies [121]. Implicit feedback studies can reveal fundamental relationships between application QoS metrics and QoE but cannot propose QoS/QoE mappings. While implicit feedback is a known technique in the information retrieval community [70], it is rarely used to study QoE of internet applications.

**Objective quality assessment.** In this methodology, researchers evaluate the quality of a system by monitoring application QoS metrics that are known to highly correlate with user QoE. Depending on how much information is available from the original source, we classify objective quality assessment methods as *full reference*, *reduced reference*, or *no reference*. Full reference methods are the most accurate, since they compare the original and transmitted signals to calculate an objective quality metric, but are hard to employ as the original signal is not always available. No reference methods estimate quality using only the received signal, while reduced reference methods use also meta-data from the original signal to better estimate quality. Full reference methods are often used when the original signal is available, as is the case during some controlled experiments.

Objective quality assessment methods are important components of QoE monitoring and QoE management systems. QoE monitoring systems focus on detecting instances where QoE is good/bad, subject to constraints imposed by the system point of view. There are QoE monitoring proposals from different points of views, from end-devices [18, 68], ISPs[17, 30], and content providers [73, 6]. QoE management systems uses a combination of QoE monitoring and prediction to arbitrate resources and optimize user QoE [73, 42].

Several studies show the effect of network quality impairments on QoE for different internet applications [19, 108, 69, 64, 99]. To the best of our knowledge, no work studies the effect of Wi-Fi quality on QoE as we do here. We use objective quality assessment

to study the relationship between Wi-Fi quality and QoE. Then, we propose a QoE monitoring method that predicts user QoE based on Wi-Fi quality.

### 2.2.2 QoE of internet applications

There's extensive work studying QoE influence factors and proposing QoS/QoE models for different internet applications. Below, we present background and related work on web browsing, YouTube, and audio and video RTC.

**Web browsing.** There is recent interest into understanding web browsing QoE and its influence factors. Varela et al. [119] studies the influence of page load time, visual appeal and ease-of-use on user QoE. Bocchi et al. [12] analyzes how different application QoS metrics correlate to one another, proposing easier to obtain application QoS metrics. While these show that web browsing QoE is the result of many influence factors, end user waiting time is consistently identified as the main system influence factor for QoE.

The perception of end user waiting time on web requests is also subject to debate. The time between the page request, marked by an explicit "click" or entering a url, and page load completion can be measured using the *onload* browser event, triggered when all HTTP requests are answered. This time duration may differ from the user perception of page load time, as users may find the page "ready" before all page requests complete. We refer to these waiting times as "PLT" and "user perceived PLT", respectively. Varvello et al. [120] used crowdsourcing to directly compare user perceived PLT with several web QoS metrics. The found that PLT closely resembles user perceived PLT, user perceived PLT usually over-estimates PLT. It is important to note that user perceived PLT tends to differ from PLT for pages with content "below the fold", not normally visible without scrolling the page, as observed by Schatz et al. [99].

There are different web browsing QoS/QoE models based on the type of web activity: opening a single page, browsing through an extended session, reading e-mails, or downloading a file [99]. Considering the simpler case of single-page retrieval, the ITU-T g 1030 proposes a MOS model based on subjective studies, mapping PLT to MOS considering fast, medium, and slow network contexts [61]. This model considers a logarithmic relationship between PLT and MOS, which is in line with most recent web QoE studies [35, 110]. Alternatively, there's work mapping PLT and MOS using an exponential relationship, but results tend to only differ for exceptionally large PLT [36]. We use the ITU-T g 1030 model to map PLT measured by the *onload* event

into MOS.

**Audio RTC.** Audio RTC allows internet voice communication between two or more participants. There are many methods to estimate the perceived quality of a voice service. While many influence factors may affect QoE, most methods consider a listen only context of an average user, not considering conversational dynamics (e.g., reading text vs. free speech) or economic expectations (e.g., free vs. paid service) [99]. We identify two main approaches. Signal based methods analyze the received signal to estimate listening quality using full reference analysis (e.g., Perceptual Speech Quality Measure (PESQ) [58]) or no reference analysis (e.g., ITU-T P.563 [59]). Parametric models, such as the E-model (ITU-T G.107), estimate listening quality based on several impairment factors such as one way delay, codec distortions, and others. There are also hybrid approaches, combining parametric and signal based methods, covering a wider range of quality impairments [111]. In our work, we use PESQ to estimate audio quality impairments and the E-model to estimate one-way delay impairments, similarly to Sun et al. [111].

**Video streaming.** Video streaming is a popular internet application, with IP video accounting for 70% of all consumer internet traffic [54]. Video encoding is a key aspect of video streaming, as uncompressed videos are very bandwidth demanding. We can obtain videos with different bitrates depending on quality parameters set during the encoding process, such as frame rate, resolution, and quantization parameter. Videos with higher bitrate are often associated with better QoE [104, 6]. Higher is not always better: video streaming typically requires a connection bandwidth larger than the video bitrate, otherwise packet losses and delays may incur in significant quality impairments.

There are two very different approaches to deliver video on the internet. Packet loss tolerant transport protocols, such as Real time Transport Protocol (RTP) and RTP Control Protocol (RTCP), are mainly used to deliver real time audio and video on the internet. They accept losing some packets in order to timely deliver content between peers, thereby minimizing latency and jitter. RTP and RTCP are widely used by IPTV services [127], as well as real time communication applications using WebRTC [85]. Alternatively, HTTP video streaming uses TCP to reliably deliver content between peers. While this guarantees content fidelity, packet retransmissions may cause latency and jitter. This is the preferred approach for streaming video on-demand content, and was largely popularized by YouTube.

**RTP video streaming.** Different quality impairments may occur during RTP

video streaming. Since RTP does not directly employ error recovery mechanisms, packet losses directly degrades the video quality through visual artifacts. Some systems may employ error concealment techniques to reduce perceived degradation, for example by freezing the video until a completely valid frame is available [72]. In cases where there's not enough bandwidth for video transmission, we will observe severe quality degradation due to packet losses. In the case of real time encoding, the sender may dynamically adapt the video bitrate to reduce bandwidth requirements.

PSNR, SSIM and VQM are three full reference quality metrics widely used for video quality assessment, analyzing image quality distortion frame by frame between reference and transmitted videos. PSNR and SSIM are image quality assessment metrics, comparing two images to produce an error/similarity score. SSIM is the preferred choice, since PSNR was shown to not correlate well with human perception [123]. Wang et al. [124] proposes a QoS/QoE model that maps average video SSIM to MOS. There are other well performing QoS/QoE models, such as the VQM model [94] and the MOVIE-index [100]. In our work, we use full reference analysis to calculate average SSIM per video sample during video RTC experiments. Then, we calculate MOS values using the mapping proposed by Wang et al. [124].

**HTTP video streaming.** In its simpler version, HTTP video streaming consists of downloading a single video file. The video playback contains an initial joining time, where the video player fills the playback buffer, followed by the video execution with possible rebuffering events, video playback interruptions to refill the video playback buffer when it empties. Intuitively, video playback should occur smoothly as long as video download rate is faster than playback rate. The video server may provide videos with different bitrate options, allowing clients to choose the desired video quality.

The development of Adaptive Video Streaming allows video bitrate adaptation to occur during playback. Each video file is encoded at different bitrates (notably, with different resolution) and each video file is partitioned into several "chunks" with equal duration. The client downloads sequential chunks to play the video, with the possibility of choosing the bitrate of each chunk. Video player can dynamically adapt to variations in available bandwidth by reducing/increasing video chunk bitrate.

We identify three key application QoS metrics on HTTP adaptive video streaming: *join time*, *rebuffering events*, and *video bitrate*. Many researchers agree that *join time* has the lowest impact on QoE [121, 6, 30]. Wamser et al. [121] shows that the number, duration and frequency of rebuffering events can significantly degrade QoE, proposing a QoS/QoE model that estimates MOS based on these metrics. The average

video bitrate and number of bitrate changes are also important factors on video QoE quality [6, 48]. We measure video quality using two models. For each experiment, we calculate average video SSIM to calculate image quality impairments on video executions at lower bitrates, and obtain MOS values using the model proposed by Wang et al. [124]. We also measure the number and duration of buffering events to calculate quality impairments due to rebuffering events using the model proposed by Wamser et al. [121]. Finally, we estimate quality impairments as the minimum MOS estimation between both models.

## 2.3 Summary

Home Wi-Fi networks is an important topic of research given the prevalence of Wi-Fi connectivity in home networks. Prior work on link capacity and available bandwidth estimation used active measurements or assumed unrealistic PHY rate utilization. We build a method that estimates Wi-Fi link capacity and available bandwidth by passively measuring link-layer Wi-Fi metrics. There is work using active measurements to diagnose Wi-Fi pathologies. They are appropriate for Wi-Fi troubleshooting, but we cannot continually execute them in home networks. There is work using multiple Wi-Fi cards such that one functions as AP and the other monitor and diagnose the RF spectrum. We avoid the use of multiple Wi-Fi cards since we want solutions that ISPs can deploy at no additional cost. We propose passive diagnosis of Wi-Fi problems between *medium access* and *frame delivery* problems.

Impairments in the home Wi-Fi network can degrade user QoE on internet applications. Prior work shows the effect of network quality impairments on QoE for different internet applications. To the best of our knowledge, no work studies the effect of Wi-Fi quality on QoE as we do. We use objective quality assessment to study the relationship between Wi-Fi quality and QoE. Then, we propose a QoE monitoring method that predicts user QoE based on Wi-Fi quality.

# Chapter 3

# Passive Wi-Fi link capacity estimation

In this section, we present a method to estimate the link capacity of Wi-Fi links using passive metrics Wi-Fi drivers commonly expose. We define the *link capacity* as the maximum UDP throughput between the AP and the device assuming full medium availability. Our contribution is to define and validate simple models to estimate link capacity in 802.11n networks. Previous models [66] are for 802.11a/b/g, which did not have frame aggregation. First, we define a model under the assumptions of fixed physical data rate (or *PHY rate*) and no frame losses (§3.2). Then, we extend this model to the realistic case where the PHY rate varies and frame losses occur (§3.3).

Link capacity is useful for Wi-Fi diagnosis. As illustrated in Figure 1.4, a device's throughput may be limited because of *medium access* problems (i.e., Wi-Fi or non-Wi-Fi contention prevents access to the medium) or *frame delivery* problems affecting the link capacity (i.e., when the channel quality is poor) [76]. In §3.4, we show with two case studies how to use the link capacity to identify Wi-Fi performance bottlenecks and distinguish between those caused by medium access or frame delivery problems.

We tuned and validated our models experimentally on a commercial AP with a Broadcom NIC, which is a popular NIC in the APs ISPs provide. Our solution is broadly applicable, and we are starting to implement it on other chipsets. The results of our controlled experiments show that we can estimate the link capacity with estimation errors similar to the state of the art, but without the need to tune station-specific parameters.

## 3.1 Experimental setup

This section describes the setup of our experiments to validate the link capacity models. We run experiments in two settings: an anechoic chamber, where we know there is no external source of contention and interference; and our lab in Paris, which is a more realistic environment with other Wi-Fi networks and other sources of non-Wi-Fi interference.

**Testbed.** We study primarily one Technicolor AP with a Dual core Broadcom MIPS 400 MHz processor, 256 MB DDR RAM and a Broadcom BCM6362 NIC with 802.11n 2x2 technology. This AP chooses the $MAX_{agg}$ based on the maximum Rx A-MPDU length of the station, advertised on control messages. Particularly, we observe that $MAX_{agg}$ is equal to 4, 8, 16 and 32 when the *Maximum A-MPDU length* is 8, 16, 32 and 64 Kbytes respectively. For brevity, we consider that the AP only uses long guard interval and 20Mhz bandwidth, since this is the default configuration for the AP. Although most of our evaluation focuses on one AP, it has the Broadcom driver wl, which is popular among the APs ISPs provide to their customers. We also test the Broadcom brcmsmac driver and the Atheros ath9k driver in §3.2.2. We observe different behaviors among the drivers, but it is possible to tune the parameters of the model to account for these differences.

We perform link capacity tests using two devices: an Android tablet, with an 802.11n 1x1 NIC with $MAX_{agg} = 8$, and a MacBook pro, with a Broadcom 802.11n 2x2 NIC with $MAX_{agg} = 32$. We use two applications with the android tablet: *iPerf for Android* to use iperf in server mode, and *Wake Lock* to prevent the device from entering in sleep mode. We use iperf client on a Lenovo laptop with Ubuntu 12 to generate traffic from the AP to the devices. We use a sniffer, a MacBook pro in monitoring mode, to capture frames the AP sends and receives. We look at the retry bit to infer frame losses at the station and jumps on the MAC sequence number to infer frames not captured by the sniffer. This gives us ground truth on the link capacity and frame delivery ratio.

**Fixed PHY rate experiments.** In this scenario, we use a driver utility tool on the AP to saturate the link to the device using a chosen fixed PHY rate. We use packets with MAC payload of 1500 bytes, since it is the default Ethernet MTU. We execute one experiment per PHY rate with 5 minutes duration, for both the tablet and the MacBook. We measure the link capacity by calculating the UDP throughput to the device using packet logs from the sniffer, discounting IP and UDP packet headers.

**Varying PHY rate experiments.** In this scenario, the Lenovo laptop (con-

Table 3.1: Description of metrics measured on the access point.

| | Metric | Granularity | Description |
|---|---|---|---|
| AP metrics | $BUSY_{\text{WiFi}}$ | 2 s | % of time receiving Wi-Fi traffic |
| | $BUSY_{\text{NonWiFi}}$ | 2 s | % of time medium busy due to non-Wi-Fi signal |
| | A-MPDU chain size | any | Size of last transmitted A-MPDU chain |
| Station metrics | TX/RX data rate | any | Kilobits sent / received |
| | TX/RX PHY rate | any | PHY rate of last non-management frame sent / received |
| | FDR | any | Fraction of frames successfully delivered to station |

nected through a gigabit interface to the AP) uses iperf to generate UDP traffic to the station. We perform link capacity tests in the anechoic chamber, using a metal box to generate stable attenuation between the AP and the station. We perform one test without any attenuation and three tests with the device inside the metal box in different positions, to obtain different levels of link quality. Experiment duration is 20 minutes per scenario, and we measure the link capacity by calculating the UDP throughput to the station using packet logs from the sniffer.

**AP sampling.** Our link capacity solution can be implemented by periodically sampling AP parameters, as shown in §3.3.2. Table 3.1 describes the metrics we sample from the AP. We use wlctl, a Broadcom utility program, to sample the AP.

$BUSY_{\text{WiFi}}$ and $BUSY_{\text{NonWiFi}}$ are measured by the Wi-Fi driver over a period of 2 seconds. The other metrics can be sampled at any granularity, restricted only by the sampling overhead. It is possible to discover $MAX_{agg}$ of connected stations by monitoring the AMPDU-chain size on periods when only one station is transmitting.

## 3.2 Link Capacity under ideal conditions

This section proposes a model to estimate the link capacity assuming that the PHY rate is constant and there are no losses (i.e., 100% frame delivery). We remove these simplifying assumptions in the next section.

### 3.2.1 Model

Our model of link capacity extends the model of Jun et al. [66] to work under 802.11n MAC improvements, including frame aggregation. We calculate the link capacity for a given PHY rate, $P$, by dividing the UDP payload of the A-MPDU by its transmission time:

$$LC(P) = \frac{AGG(P) \times \text{UDP payload}}{\text{A-MPDU TxDelay (P)}} \times (1 - B_o) \tag{3.2.1}$$

where $B_o$ is the fraction of time the AP is busy sending beacons.

**A-MPDU UDP payload.** We calculate the UDP payload of the A-MPDU frame exchange by multiplying the UDP payload per IP packet by the number of MPDUs sent at $P$, $AGG(P)$. We show how to obtain $AGG(P)$ in §3.2.2.

**A-MPDU transmission delay.** To compute the A-MPDU transmission delay, we use 802.11 protocol parameters to model an A-MPDU exchange of $N$ packets of size $S$ using PHY rate $P$. Figure 1.3 shows our A-MPDU frame-exchange model, which uses Enhanced Distributed Channel Access (EDCA). We compute the A-MPDU transmission delay as:

$$\text{TxDelay(N,S,P)} = T_{AIFS} + T_{BO} + 3 \times T_{SIFS} + T_{RTS} + T_{CTS} + T_{ACK} + \text{DATA(N,S,P)} \tag{3.2.2}$$

With no frame loss assumption, the backoff timer does not exponentially increase. The backoff timer is chosen using a uniform distribution between 0 and $CW_{min}$, giving the expected value of $\frac{CW_{min}}{2}$. We estimate the delay to transfer the data block as:

$$\text{DATA(N,S,P)} = T_{PH} + \frac{22 + N \times (S_{MH} + S)}{P} \tag{3.2.3}$$

where $T_{PH}$ is the transmission delay of the PHY header, and $S_{MH}$ is the MAC header size. We add 22 trailing bits (16 + 6) to form the OFDM symbols. This is an approximation since we don't consider padding bits.

For the purposes of calculating the link capacity, we use $S = 1500$ bytes (default MTU for Ethernet networks), UDP payload of 1472 bytes and N = $AGG(P)$. Other model parameters are defined in Table 3.2. We consider frame exchanges using Best Effort Access Category, since it is the default configuration for bulk traffic transfer, and the use of implicit Block Ack Request. We consider control frames with PHY rate $\in \{1, 2, 6, 12, 24\}$ Mbps, and assume that the PHY rate to transmit a control frame is lower than that of a data frame. We use the transmission delay of control frames proposed by Jun et al [66] (shown in Table 3.3).

### 3.2.2 Parameter tuning

Our model in §3.2.1 has two parameters that we must estimate for the particular AP under study: $B_o$ and $AGG(P)$. In practice, ISPs work with relatively few models

Table 3.2: Parameters for the model instance

| Parameter | Value |
|---|---|
| $T_{SIFS}$, $T_{PIFS}$, $T_{AIFS}$ | $16, 25, 43\mu$s |
| $T_{BO}$ | $139.5\mu$s |
| $T_{PH}$ | $20\mu$s |
| $S_{MH}$ | 38 bytes |
| $CW_{min}$ | 31 |

Table 3.3: Timing of control frames

| Modulation | $T_{RTS}$ | $T_{CTS}$ | $T_{ACK}$ |
|---|---|---|---|
| OFDM 24 | $28\mu$s | $28\mu$s | $32\mu$s[1] |
| OFDM 12 | $36\mu$s | $32\mu$s | $44\mu$s[2] |
| OFDM 6 | $52\mu$s | $44\mu$s | $68\mu$s[2] |
| DSSS 2 | $272\mu$s | $248\mu$s | $248\mu$s |
| DSSS 1 | $352\mu$s | $304\mu$s | $304\mu$s |

of APs and estimating these parameters is simple. We show how to obtain $B_o$ by using the beacon interval and the number of SSIDs advertised. Even though $AGG(P)$ is implementation dependent, we show that the two most used Broadcom 802.11n drivers use the same function $AGG(P)$.

**Beacon overhead ($B_o$).** We estimate the $B_o$ as the fraction of time the AP is busy sending beacons. We calculate the delay to transmit a single beacon as its transmission time + $T_{PIFS}$. We calculate how many beacons are sent each second as the inverse of the beacon interval, and multiply it by the number of advertised SSIDs.

The AP under study advertises 3 SSIDs using 242-byte beacons at PHY rate 1 Mbps every 100 ms. We calculate that the AP sends 30 beacons each second, each beacon with duration of 1981 $\mu s$. Since the AP spends 59.430 $ms$ every second sending beacons, we have $B_o = 5.943\%$.

**Frame Aggregation per PHY rate ($AGG(P)$).** Either the driver or the hardware decides the aggregation of frames into A-MPDUs, with most modern 802.11n devices opting for the latter [71]. Broadcom NICs capable of IEEE 802.11n can use either the closed proprietary wl driver or the open-source driver brcm80211.

We setup a small controlled experiment to understand how our AP, which uses the wl driver, selects the number of frames per A-MPDU for different PHY rates. In this

Table 3.4: Estimated A-MPDU size (AGG), temporal duration (dur) in $\mu s$ and link capacity (LC) in Mbps per PHY rate.

| **PHYrate** | **$MAX_{AGG} = 8$** | | | **$MAX_{AGG} = 32$** | | |
|---|---|---|---|---|---|---|
| **(Mbps)** | **AGG** | **DUR** | **LC** | **AGG** | **DUR** | **LC** |
| 6.5 | 2 | 4150.35 | 5.34 | 2 | 4150.35 | 5.34 |
| 13.0 | 5 | 5045.81 | 10.98 | 5 | 5045.81 | 10.98 |
| 19.5 | 7 | 4730.32 | 16.39 | 7 | 4730.32 | 16.39 |
| 26.0 | 8 | 4075.35 | 21.74 | 10 | 5021.81 | 22.06 |
| 39.0 | 8 | 2813.40 | 31.50 | 15 | 5021.81 | 33.08 |
| 52.0 | 8 | 2182.42 | 40.60 | 21 | 5258.42 | 44.23 |
| 58.5 | 8 | 1972.10 | 44.93 | 23 | 5126.97 | 49.69 |
| 65.0 | 8 | 1803.84 | 49.12 | 26 | 5211.10 | 55.26 |
| 78.0 | 8 | 1551.45 | 57.11 | 31 | 5179.55 | 66.29 |
| 104.0 | 8 | 1235.96 | 71.69 | 32 | 4075.35 | 86.97 |
| 117.0 | 8 | 1130.80 | 78.36 | 32 | 3654.70 | 96.98 |
| 130.0 | 8 | 1046.67 | 84.66 | 32 | 3318.18 | 106.82 |

experiment, the AP sends data to the android tablet or to the MacBook using fixed PHY rate, as described in §3.1. The number of frames per A-MPDU used by the tablet and the MacBook is shown in Columns 2 and 5 of Table 3.4, respectively. We see that, when transmitting packets of the same size, the A-MPDU size varies per PHY rate, with more packets per A-MPDU at higher PHY rates up to $MAX_{AGG}$.

The inspection of brcmsmac's source code helps explain this behavior. The transmitter limits transmission duration to a threshold *txop*. It estimates how many bits can be sent during *txop* and then computes the maximum number of frames per A-MPDU with the equation:

$$AGG(P) = min\left(\left\lfloor \frac{P \times txop}{\text{MAC frame length}} \right\rfloor, MAX_{agg}\right) \tag{3.2.4}$$

The MAC frame length is given by MAC payload + $S_{MH}$. Considering a MAC payload of 1500 bytes and the default value of *txop* in brcmsmac of 5 ms, we were able to correctly estimate $AGG(P)$ for all PHY rates in Table 3.4. This suggests that both wl and brcmsmac uses the same *txop* threshold values.

While analyzing the sniffer logs, we observe an artefact on how the AP handles frame aggregation. Between PHY rates 52 Mbps to 78 Mbps, the A-MDPU

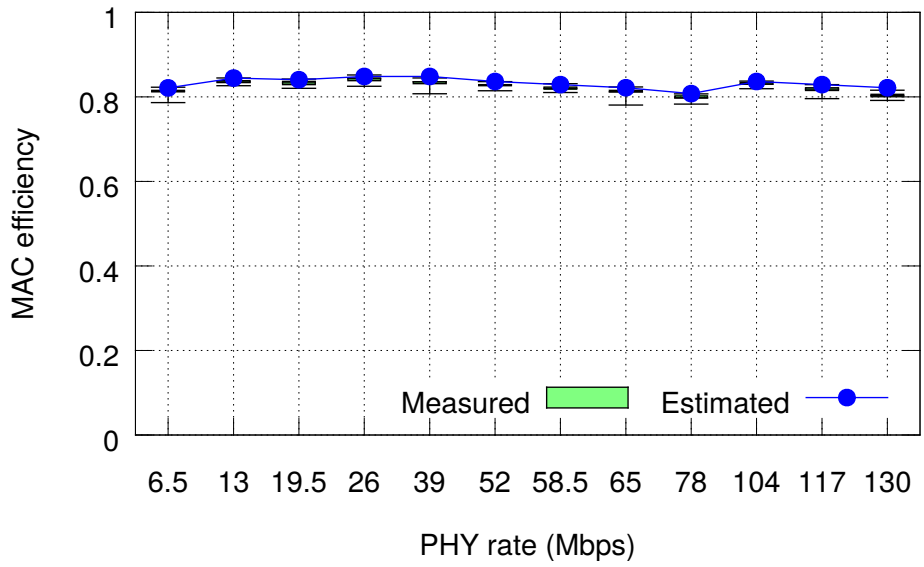



Figure 3.1: Estimated vs. measured link capacity of MacBook

size of transmitted frames successively alternates between two values, $AGG(P)$ and $MAX_{AGG} - AGG(P)$. The most extreme case is on the MacBook at PHY rate 78 Mbps, where we observe alternating A-MPDUs of sizes 31 and 1, resulting in reduced throughput in comparison with the usage of back-to-back transmissions of A-MPDUs of size 31. We consistently observed this artifact on all transmissions of the AP. We conjecture that the NIC is internally splitting outbound packets in blocks of size $MAX_{AGG}$ and then attempting to deliver all packets inside this block before moving to the next.

We perform a fixed PHY rate experiment with a Unix machine using an Atheros card with ath9k driver to check whether we see the same behavior. With the Atheros card we observe no instances of A-MPDUs with alternating sizes. This result indicates that this artifact is specific to the Broadcom NIC. Further tests are necessary to confirm whether this behavior happens on other models of Broadcom NICs.

### 3.2.3 Experimental validation

In order to validate the proposed model, we perform controlled experiments in an anechoic chamber using fixed PHY rate capacity tests as described in §3.1, comparing the measured link capacity with the estimated link capacity. This step should highlight any discrepancies between the link capacity model and the achieved UDP throughput. At PHY rates 52 Mbps to 78 Mbps we consider frame exchanges with A-MPDU sizes of 16, since the transmitter uses 2 medium accesses to deliver 32 packets.

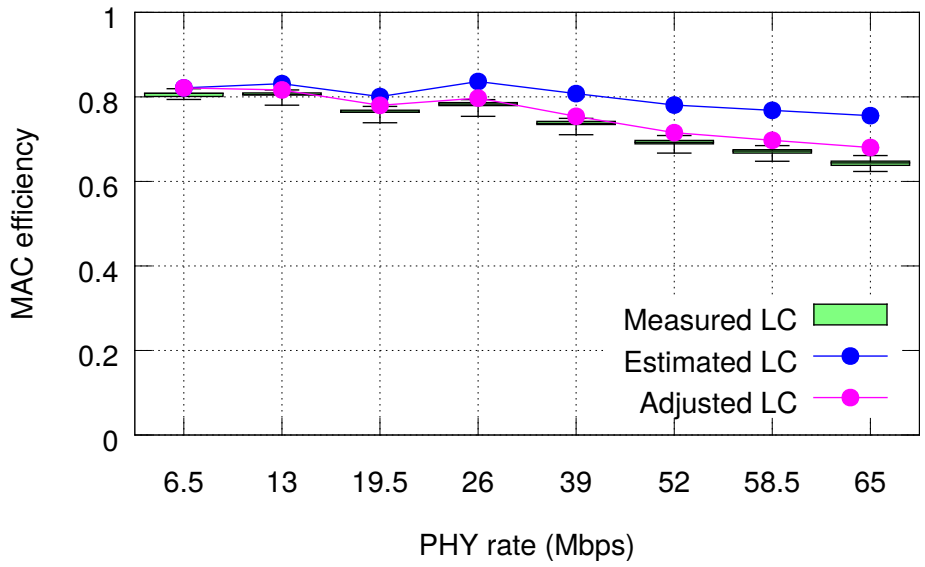



Figure 3.2: Estimated vs. measured link capacity of tablet

Figure 3.1 shows the *MAC efficiency*, the ratio between the link capacity and the PHY rate, during tests with the MacBook. Under fixed A-MDPU sizes, the AP deliver more frames at higher PHY rates, resulting in more medium accesses and larger MAC overhead. We see this behavior between PHY rates 39 Mbps and 78 Mbps with $AGG(P) = 16$, and between PHY rates 104 Mbps and 130 Mbps, with $AGG(P) = 32$. We are able to accurately estimate the link capacity at all PHY rates. However, the link capacity estimation is slightly positively biased (1.925%, on average), due to the extremely optimistic model assumptions.

Figure 3.2 shows the MAC efficiency for tests with the Android Tablet. A-MPDU size reaches maximum value when PHY rate $\geq$ 26 Mbps, thus the downward MAC efficiency between PHY rates 26 Mbps and 65 Mbps. We observe that estimated LC differs from measured LC when PHY rate $\geq$ 26 Mbps, showing a positive bias of up to 17% at PHY rate 65 Mbps. Our analysis of sniffer's packet logs reveals that the AP takes additional time between A-MPDUs of size 8 (around 200$\mu s$). We made additional tests with other android devices with $MAX_{agg}$ of 8 and 32, but we only observe this when $MAX_{agg} = 8$. When including the additional delay in the model, we obtain a smaller positive bias (3.96%) on the link capacity (adjusted LC). We consider in the next section two link capacity models: one using purely protocol information (LC: original) and a second (LC: adjusted), which includes the observed additional delay for devices with $MAX_{agg} = 8$. Also, we consider the presence of a positive bias of 4%,

and we deduce it before usage.

## 3.3 Link Capacity in Practice

This section adapts the model from §3.2 to work in practice. Rate adaptation algorithms frequently change the selected PHY rate [74] and frames may be lost (i.e., frame delivery $< 100\%$). We first adapt the model from §3.2 to take these issues into account. Then, we discuss how to obtain the inputs for the model. Finally, we validate our model using controlled experiments and comparing it with the state of the art.

### 3.3.1 Model

As Wi-Fi link capacity varies over time, our model estimates the link capacity for a given time interval $[t_0,t_0+\tau]$. Even though the PHY rate changes over time, the AP uses only one PHY rate for each frame. Thus, we can obtain "instant" link capacity measurements by applying the model from Equation 3.2.1. Let $P(t)$ be the PHY rate used at $t$ and $FDR(t)$ be the frame delivery rate at $t$. We estimate the link capacity for the time interval $[t_0,t_0+\tau]$ as follows:

$$LC(t_0,\tau) = \frac{1}{\tau}\int_{t_0}^{t_0+\tau} FDR(t) \times LC(P(t))dt. \tag{3.3.1}$$

### 3.3.2 Model inputs

The model in Equation 3.3.1 takes four inputs:

1. The initial estimation time, $t_0$, is simply the time operators will run link capacity estimation. In our existing trials with ISPs, we report the link capacity estimates periodically, but we can imagine scenarios where operators will request the estimate on demand as well.

2. The estimation interval, $\tau$, depends on how frequently and fine-grained operators want to estimate link capacity. Values of $\tau$ that are too large will average link capacity over a long time interval and may miss variations of link capacity that are important to diagnose Wi-Fi performance. On the other hand, if $\tau$ is too small (for instance, less than a second) the variations of link capacity estimates in short periods of time become harder to interpret and to map to user's Wi-Fi performance.

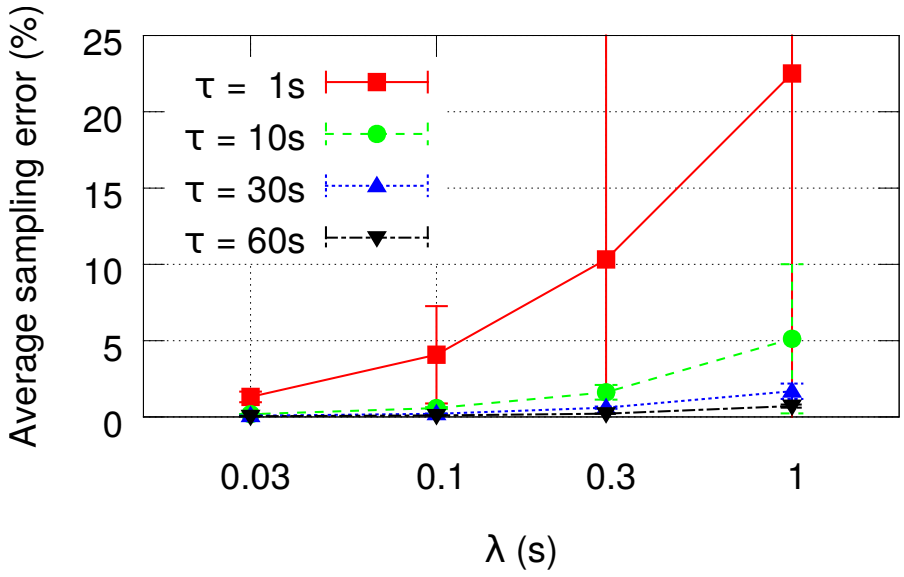



Figure 3.3: Average error when sampling the Link Capacity.

3. The function of PHY rate over time, $P(t)$. We obtain $P(t)$ by periodically polling the Wi-Fi driver. Ideally, we would get the PHY rate per frame as previous work has done for OpenWrt APs [89]. However, per-frame measurements impose high load on the system, particularly during moments of high network load. We show that we can obtain low sampling error by periodically sampling the driver for the PHY rate of last transmitted data frame.

4. The frame delivery ratio over time, $FDR(t)$. Similar to $P(t)$, we obtain $FDR(t)$ by polling the Wi-Fi driver. Unfortunately, $FDR(t)$ is not available for upstream traffic. We can approximate upstream $FDR(t)$ based on the downstream values, but this deserves further investigation, which we leave for future work.

We evaluate the sampling error of different $\tau$ and $\lambda$ parameters as follows. We emulate PHY rate sampling using the sniffer logs of the UDP iperf capacity tests, as described in §3.1. We periodically sample the PHY rate of the last data frame, with periodicity $\lambda$ over an estimation period $\tau$. We consider PHY rate sampling with periodicity 1 $ms$ as the ground truth. We compare the link capacity sampled with parameters $\lambda \in \{1\ s, 0.3\ s, 0.1\ s, 0.03\ s\}$ and $\tau \in \{1\ s, 10\ s, 30\ s, 60\ s\}$ with the ground truth, obtaining the sampling error. We execute 100 runs, randomly choosing the starting time and report average sampling error and standard deviation.

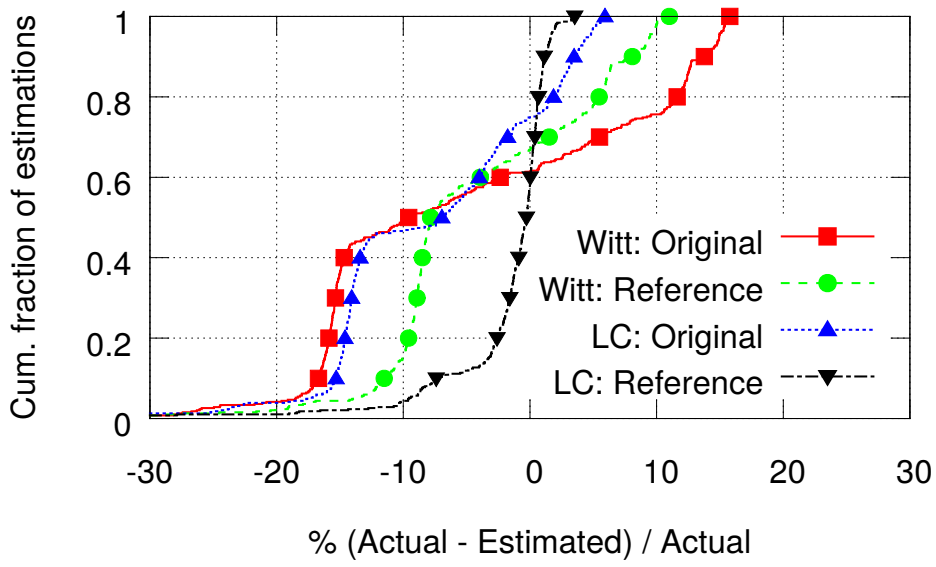



Figure 3.4: Comparison between throughput estimation methods

We can see in Figure 3.3 that sampling error decreases for larger $\tau$ and smaller $\lambda$. This is expected since estimation error tends to decrease with more PHY rate samples, which is given by $\tau/\lambda$. Therefore, there is a trade-off between sampling overhead, imposed by $\lambda$, and estimation granularity, imposed by $\tau$. We see that sampling error is, on average, below 2% when $\tau/\lambda \geq 30$.

### 3.3.3 Validation

In order to evaluate our algorithm, we performed controlled experiments in the anechoic chamber. We performed UDP iperf capacity tests as described in §3.1, using $\tau = 10\ s$.

**Comparison method.** We compare our method with the state of the art in Wi-Fi throughput estimation "Wi-Fi based TCP throughput" (Witt) [89]. Witt is calculated by linearly fitting a custom metric, *link experience*, with TCP throughput ground truth data. Link experience and Witt are given by:

$$\text{link\_exp} = (1-a) \times (1-c) \times \sum_i \frac{s_i.r_i}{p_i} \tag{3.3.2}$$

$$Witt = \beta_1 \times \text{link\_exp} + \beta_0 \tag{3.3.3}$$

where $a \in [0,1]$ is a percentage of airtime utilization used by external sources, and $c \in [0,1]$ accounts for local contention. While originally Witt is used to predict TCP throughput, we fit Witt to predict UDP throughput.

Both LC and Witt have very high correlation coefficients with the throughput, respectively .997 and 0.996. A key difference between both methods is that Witt finds the ratio between PHY rate and data rates (the MAC efficiency) by fitting $\beta_1$ on a data set and minimizing errors with the intercept $\beta_0$. We calculate the MAC efficiency per PHY rate. We see this when comparing Witt's original $(\beta_0, \beta_1)$ parameters $(-0.494, 0.733)$ with parameters found by fitting only the tablet data $(2.502, 0.629)$ and only the MacBook data $(0.875, 0.783)$. Witt cannot distinguish .11n devices with low and large frame aggregation usage, causing less accurate throughput predictions.

We compare the estimated link capacity using 4 methods: 1) Witt with original $\beta$ values; 2) Witt with $\beta$ values found by fitting this device's fixed PHY rates tests (Witt: reference); 3) LC with original values (LC: Original) and; 4) LC considering the AP instance inefficiencies at high PHY rates, as seen in in §3.2 (LC: Reference).

Figure 3.4 shows the distribution of estimation errors of the 4 different approaches. As expected, training the prediction method with reference data from the device under test significantly reduce prediction errors. However, training per station is impractical in operational environments. Over 90% of the predictions had and error below 15% for original LC, and over 95% presented error below 5% when using LC with reference data. We conclude that, in order to obtain more accurate link capacity estimations, the model should be tuned to properly incorporate performance inefficiencies on devices with restricted A-MPDU capabilities (i.e., $MAX_{AGG} \in \{8, 16\}$) as shown in §3.2. Fortunately, this has to be done once per AP model.

## 3.4 Diagnosis of throughput bottlenecks

In this section, we present how we can use the proposed link capacity to diagnose downstream throughput bottlenecks. We show that, by using the estimated link capacity defined in §3.3 in conjunction with other AP metrics, we can estimate the available bandwidth, which helps identifying throughput bottlenecks. Further, we show how we can diagnose instances of reduced available bandwidth.

### 3.4.1 Medium Access and Frame delivery losses

Figure 3.5 illustrates how the MAC overhead, frame delivery losses (FD), and medium access losses (MA) explain the difference between the nominal physical link capacity and the available bandwidth.

The link capacity defined in §3.3 estimates how much bandwidth the link supports

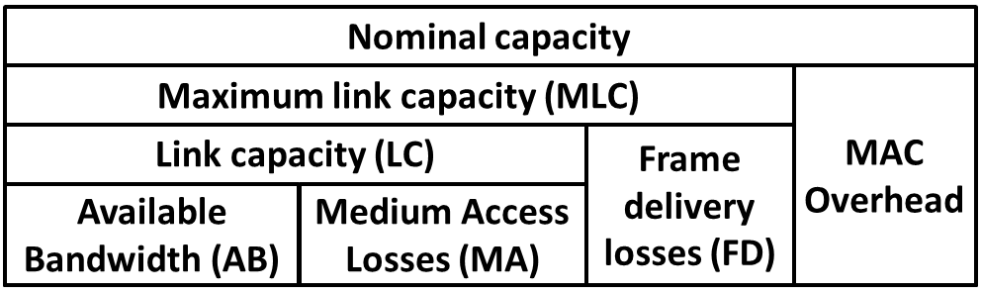



Figure 3.5: Breakdown of nominal capacity into: maximum link capacity, link capacity, available bandwidth, medium access losses and frame delivery losses

assuming full medium availability, but in reality we share the unlicensed medium with many Wi-Fi and non-Wi-Fi sources. Wi-Fi cards can export, for a given period, the percentage of time the medium was busy due to the reception of nearby Wi-Fi frames ($BUSY_{\text{WiFi}}$) or to the presence of high noise ($BUSY_{\text{NonWiFi}}$). We use these metrics to account for medium sharing. The AP we study report these counters with a resolution of 1%. We define medium access losses ($MA$) as the fraction of $LC$ lost due to busy medium, and available bandwidth ($AB$) as the fraction of $LC$ available for usage.

$$MA = LC \times (BUSY_{\text{WiFi}} + BUSY_{\text{NonWiFi}}) \tag{3.4.1}$$

$$AB = LC \times (1 - BUSY_{\text{WiFi}} - BUSY_{\text{NonWiFi}}) \tag{3.4.2}$$

Consider an ideal Wi-Fi link, with no frame loss nor medium sharing. This link would present the maximum $LC$ value ($MLC$), limited only by the maximum PHY rate available between endpoints. If $LC$ is lower than $MLC$, the channel quality is poor and we lose capacity due to a frame delivery problem. Therefore, we consider the difference between $MLC$ and $LC$ a frame delivery loss.

$$FD = MLC - LC \tag{3.4.3}$$

### 3.4.2 Case studies

Here we show two case studies, where we apply the metrics defined in §3.4.1 to diagnose the source of the reduced available bandwidth.

**Non-Wi-Fi interference.** We investigate how the available bandwidth varies under microwave interference with an experiment in the anechoic chamber. We use

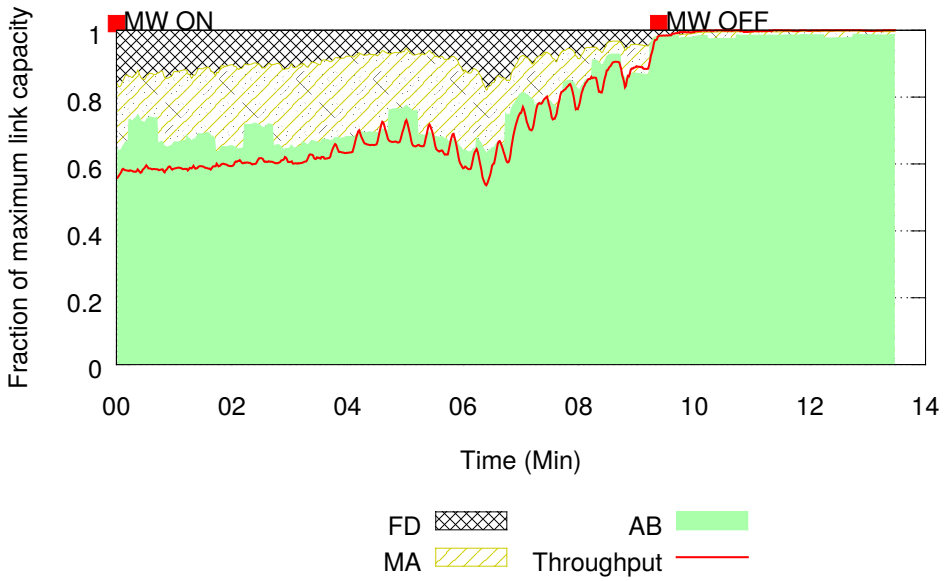



Figure 3.6: Available bandwidth under microwave interference

iperf to generate UDP traffic from the AP to the tablet as described in §3.1. The AP is configured to use channel 11, the most impacted by microwave interference. The experiment lasts 14 minutes and the microwave is ON during the first 10 minutes.

Figure 3.6 shows AB, MA, and FD, normalized by the maximum link capacity. From minutes 1 to 10 we see that the throughput is roughly 60% of the maximum link capacity, due to the Microwave duty cycle of approximately 40%. We observe that, after 7 minutes, the gateway manages to achieve more and more throughput. This seems to be a microwave mechanism to avoid overheating by periodically reducing the number of active cycles. Our method is able to predict the general trend of the microwave interference. Since in this scenario $BUSY_{\text{WiFi}} = 0$, we can further diagnose the interference source as non-Wi-Fi.

**Wi-Fi performance in uncontrolled environments.** Here, we diagnose the available bandwidth of a transmission in a real world uncontrolled scenario. We use iperf to generate traffic from the AP to the tablet in our office space. We put the AP on the extreme end of a long corridor, and the tablet at the other end, 40 meters apart.

Figure 3.7 shows the estimated available bandwidth as well as the causes for throughput losses. Between minutes 1 and 2, the throughput goes down to 3 Mbps. During this period, the tablet only received transmissions with PHY rate 5.5 Mbps. This result indicates a frame delivery problem either due to high noise, which prevents higher modulation usage, or narrow band interference, which prevents usage of OFDM

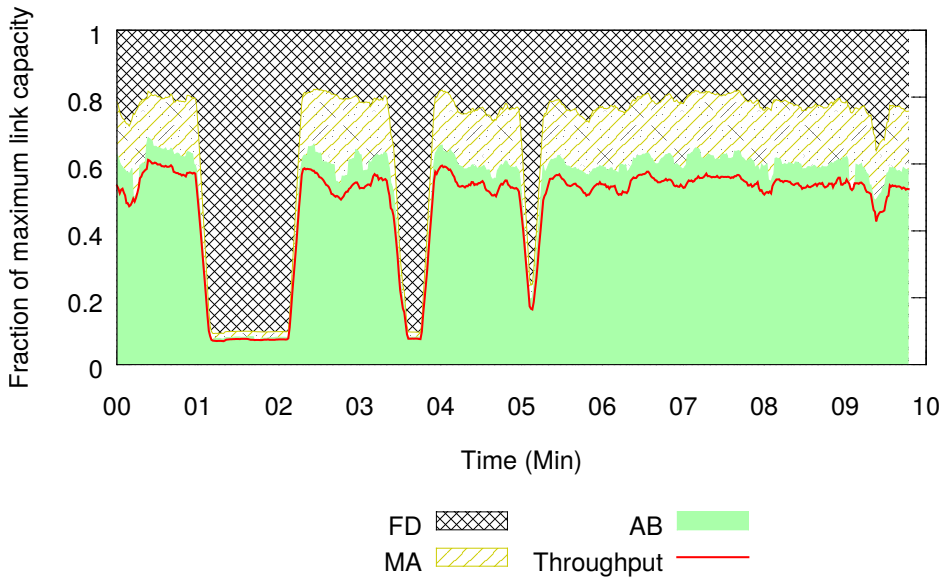



Figure 3.7: Available bandwidth prediction on the wild

modulation. Our method allows us to identify that the throughput loss during this period was due to a frame delivery loss.

## 3.5 Summary

In this section, we presented an algorithm to estimate the link capacity based on passive metrics from APs, which is ready to be deployed at scale. We show that it is possible to estimate the link capacity per PHY rate based on a limited set of parameters related to the particular AP instance. Then, we extend the initial model to estimate the link capacity when the PHY rate varies. We measured the link capacity in different link quality conditions and found that more than 90% of the estimations present error below 15% without prior parameter tuning, and more than 95% present estimation error below 5% with appropriate parameter tuning using fixed PHY rate tests. Also, our method achieves below 2% sampling errors when the ratio $\tau/\lambda \geq 30$. This method allows ISPs to passively monitor Wi-Fi quality on commodity APs. The simple diagnosis approach proposed allows helpdesk operators diagnose users' Wi-Fi problems, distinguishing medium access and frame delivery problems.

# Chapter 4

# Predicting the effect of Wi-Fi quality on QoE

In this chapter, we develop a model that predicts the effect of Wi-Fi quality on four internet applications: web browsing, YouTube, audio and video real time communication (RTC). We selected these applications because they are popular among residential users [11]. Evaluating the effect of Wi-Fi quality on application QoE is challenging. We want to execute experiments on a wide range of Wi-Fi scenarios, with multiple tests per experiment. To span all these settings requires thousands of individual tests per application. With subjective quality assessment, we ask multiple users to evaluate application QoE per test to obtain the Mean Opinion Score (MOS). The combination of user tests makes this approach unpractical. Instead, we use objective quality assessment to evaluate QoE by measuring application specific QoS metrics per test (§4.1). Then, we estimate quality impairments using state of the art QoS/QoE models to translate application QoS into mean opinion scores (MOS) ( §4.2). We evaluate the accuracy of proposed QoS/QoE models with small user studies. Then, we generate training sets by combining the Wi-Fi metrics and estimated QoE, and use regression analysis to build predictors of the effect of Wi-Fi quality on QoE (§4.3).

We use cross-validation and grid optimization to find model parameters, with MOS predictions showing root-mean square errors (RMSE) between 0.643 and 0.777. We perform validation tests with data collected in our lab (with realistic Wi-Fi conditions), where we collect both Wi-Fi metrics and application metrics, and we find RMSE between 0.740, and 0.938 (§4.3.3).

## 4.1 Experimental setup

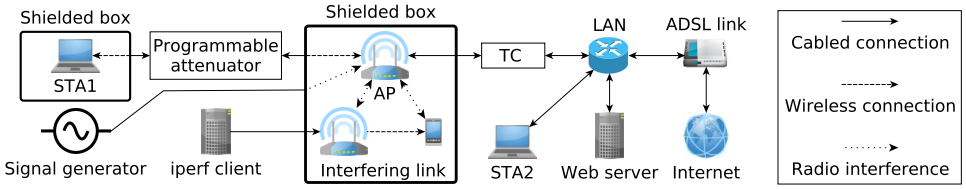



Figure 4.1: Wi-Fi testbed used in the study.

To understand how Wi-Fi quality affects user QoE, we perform controlled experiments on each of the four applications: web browsing, YouTube streaming, and audio / video RTC. The typical method to assess quality of experience is to request users to explicitly rate the quality of internet applications, obtaining mean opinion scores (MOS) [99]. We deem this approach impractical, since we want to systematically cover a wide range of Wi-Fi configurations. Indeed, we obtain over ten thousands data points across applications (over 27 hours of application time), each requiring the rating of multiple users. Instead, during each experiment we measure application specific QoS metrics and use state-of-the-art QoS/QoE models to obtain estimated MOS values.

### 4.1.1 Testbed

We emulate a Wi-Fi home network using the testbed shown in Figure 4.1, where we execute experiments on internet applications under controlled Wi-Fi conditions. We use a Technicolor AP with a Broadcom BCM6362 NIC with 802.11n 2x2 technology, and two MacBook pro stations (STA1 and STA2) with 802.11n 2x2 technology. This AP is used by some residential ISPs and has the same capabilities of those used in the field study, shown in Section 5. The Wi-Fi testbed is located in Belgium, and it connects to the internet through an Ubuntu 12 gateway using TC to emulate bandwidth restrictions of a home network access link, with 16 Mbps available bandwidth for download traffic. Our Web and iperf servers are Ubuntu 12 computers. We also use an Android tablet with 802.11n 1x1 technology, using iperf in server mode to receive traffic that interferes with the AP under test. We use the default AP Wi-Fi configurations: 20MHz channel width and long guard interval PHY rates only, on 2.4 GHz. We restrict Wi-Fi physical layer (PHY) rates to only use one-spatial-stream, due to the difficulty of reliably enabling MIMO communication in shielded boxes. While we do not study PHY rates above 65 Mbps, we expect QoE impairments to be similar to experiments

where there is no attenuation and one-spatial-stream PHY rate is maximum at 65 Mbps.

### 4.1.2 Experiment parameters

We vary the Wi-Fi conditions over two axis:

**Link speed.** When the link quality between AP and STA is good, the AP can use fast and efficient PHY rates. When the link quality is poor, however, the rate adaptation algorithm uses slow but robust PHY rates to ensure frame reception. We vary the link speed by introducing six increasing levels of attenuation in the path between AP and STA using a programmable attenuator.

**Medium availability.** The IEEE 802.11 medium access protocol specifies that nodes only transmit after sensing the medium idle. We vary the medium availability sensed by the AP by introducing interference from Wi-Fi or non-Wi-Fi sources. To introduce Wi-Fi interference, we use the iperf client to generate competing Wi-Fi traffic in the "interfering link", as shown in Figure 4.1. The AP from the interfering link is configured to only use PHY rate 5.5 Mbps, consuming more airtime per frame. We found that by generating constant bit rate UDP traffic of 0.7 Mbps, 1.1 Mbps, 2.1 Mbps and 2.9 Mbps we obtain, respectively, 70%, 50%, 30% and 15% medium availability, measured by the CCA counters on the AP.

To generate interference from a non Wi-Fi source, we use a signal generator to inject a narrowband sinewave to block the AP's CCA. We sweep the interfering sinewave in and out of the spectral CCA range of the AP every 200ms. By varying the percentage of time the sinewave is inside and outside this range, we create scenarios with medium availability of 100%, 75%, 50%, 25%, and 12.5%. We verified that Wi-Fi counters on the AP observe the intended medium availability with error below 5%.

### 4.1.3 Applications

For each Wi-Fi scenario, we execute automated tests on four internet applications: web browsing, YouTube streaming, and audio / video RTC, while we monitor Wi-Fi, network, and application QoS metrics. Below, we detail the experiment execution for each application.

**Web browsing.** STA1 uses PhantomJS, a headless browser, to sequentially access 10 times a set of 10 pages on the internet. We chose a mix of pages figuring in the Alexa TOP 20 pages in Belgium. We access the front pages of the following domains: bing.com, google.be, twitter.com, live.com, wikipedia.org, facebook.com, yahoo.com,

amazon.fr, nieuwsblad.be, and hln.be. We use JavaScript to record the page load time (PLT) triggered by the *onload* event.

**YouTube.** STA1 uses Google Chrome to access an instrumented YouTube page, which downloads and plays each YouTube video using DASH, during two minutes. We access three YouTube videos per experiment: a politician speech, a movie trailer, and a music clip. We choose these clips to obtain diversity in video content. We use JavaScript to record join time, buffering events, and video resolution.

**Audio and video RTC.** We use WebRTC to perform Audio and Video RTC tests. We implemented a simple WebRTC application that allows two peers to communicate in an audio and/or video call. STA1 and STA2 access a WebRTC page stored on the web server. Once connected, they each open a WebRTC session to the server, waiting for the other to connect. When both successfully connect, they initiate a WebRTC audio or audio/video call. We use the default OPUS plugin on WebRTC to encode audio and the VP8 plugin to encode video with default options. We use the *–use-fake-device-for-media-stream* command line option to perform audio/video real-time communication tests which allow us to control the audio/video sent by each WebRTC peer. This closely resembles a real user environment, since the audio/video processing will go through the same processing stack as a real microphone/camera. We use javascript setup to record the received audio/video streams by each WebRTC peer.

We use a set of 20 different audio samples recommended by ITU-T for speech quality assessment [60], each with an approximate duration of eight seconds. We create a single audio file with all the audio samples, which is then sent to Google Chrome's fake audio input device. To prevent echo canceling from interfering with the audio samples, we only examine cases where STA2 sends audio and STA1 is silent. After the experiment, we manually extract the individual audio samples from the received audio feed. For video, we use standard reference videos featuring human subjects from the Xiph collection [129], namely: *FourPeople*, *Johnny*, *KristenAndSara*, *Vidyo1*, *Vidyo3* and *Vidyo4*. We downscaled the original video format to $640 \times 480$, 30 fps to match common webcam capabilities. We merged all the video samples, each contains 300 frames (10 seconds), into a single video file, separated by 15 "black" frames to mark the transition between samples. After the experiment, we extract individual video samples from the received video using the "black" frames to detect sample transition, as well as manual verification (in rare cases, none of the "black" frames were received).

## 4.2 QoE from QoS metrics

We use objective quality assessment to measure how QoE varies across Wi-Fi scenarios. In this section, we present the QoS/QoE models used to quantify user QoE for each application. We select QoS/QoE models available in the literature for each application and adapt them to work on today's internet services. We validate our model choices with a small user study.

### 4.2.1 Approach overview

For each application, we define a function $f : x \rightarrow y$, where $x$ is a vector of application QoS metrics and $y$ is a MOS score. Absolute MOS values are not adequate to estimate the effect of Wi-Fi quality on QoE. A number of models exist to obtain $f$ for each of the applications in study. These models output the average absolute quality rating given by users in the study. For example, the simple act of encoding a video for real time transmission will reduce its quality. Yet, users today are used to the quality loss due to encoding. Moreover, our goal is to predict the Wi-Fi contribution to MOS degradation. Hence, we normalize the output of $f$ to obtain degradation MOS (DMOS) instead of absolute MOS. We calculate DMOS values, in range $[1 - 5]$, using the formula:

$$\bar{f}(x) = 4.0 \times \frac{f(x) - MOS_{min}}{MOS_{max} - MOS_{min}} + 1.0 \tag{4.2.1}$$

where $MOS_{min}$ is given by the minimum $MOS$ value from experiments, and $MOS_{max}$ is the average MOS during baseline experiments (i.e, no Wi-Fi impairment). This way, DMOS on baseline scenarios is always 5, whereas DMOS on the worst scenario is always 1. It is possible to find $MOS_{max}$ and $MOS_{min}$ by respectively maximizing and minimizing $f(x)$ for the set of application QoS parameters expected to be found in the target scenario. We use Equation 4.2.1 to obtain DMOS on all application models.

### 4.2.2 Video RTC QoE model

There is extensive work on objective video quality assessment using three types of metrics to capture video quality: full reference, reduced reference, and no reference. Since we have access to the original video signal in our testbed, we use full reference metrics, which are the most accurate. Popular full reference metrics are PSNR, SSIM, and VQM. We opt to use Structural Similarity (SSIM), since it was shown many times to have strong correlation with human perception.

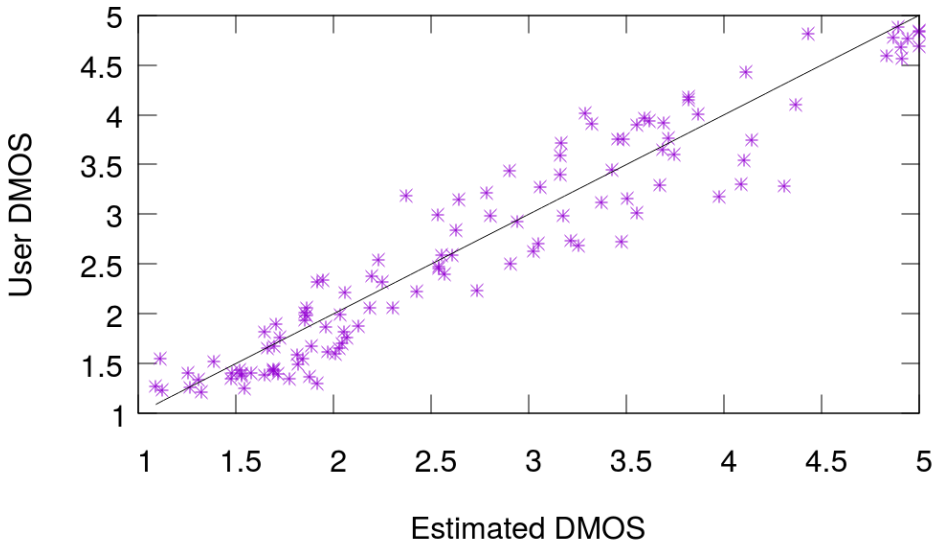



Figure 4.2: Validation of video RTC QoE model

We compare each video clip received by STA1 (*transmitted* sample), with the original sent by STA2 (*reference* sample), and compute per-frame SSIM. We calculate the video's average SSIM (s) and map it to the video MOS score using Equation 4.2.2, proposed by Wang et. al for video quality assessment [124].

$$f_v(s) = 129.25s^3 - 64.76s^2 + 22.08s \tag{4.2.2}$$

We noticed that under low network capacity, WebRTC often suppresses frames to reduce bandwidth, sending as little as 10 frames per second in extreme conditions. We deal with skipped frames using the strategy employed by Zinner et. al [132], by comparing reference skipped frames with the last received frame, producing reduced SSIM whenever the degraded sample presents skipped frames or freezes.

On some experiments, we observed instances where STA1 was able to associate to the AP and communicate to STA2 through WebRTC, but could not sustain the video call due to high packet loss. Since WebRTC prematurely terminates the call, we could not obtain valid transmitted samples. For every experiment where we observe a WebRTC connection for 10 second but no video traffic, we generate a sample with DMOS = 1.0, since there is no service.

**Model validation.** We conduct a small user study to see how DMOS correlates with user opinion. For each one of the six reference samples, we select 20 different transmitted samples with varying levels of video quality. We explain to users that they

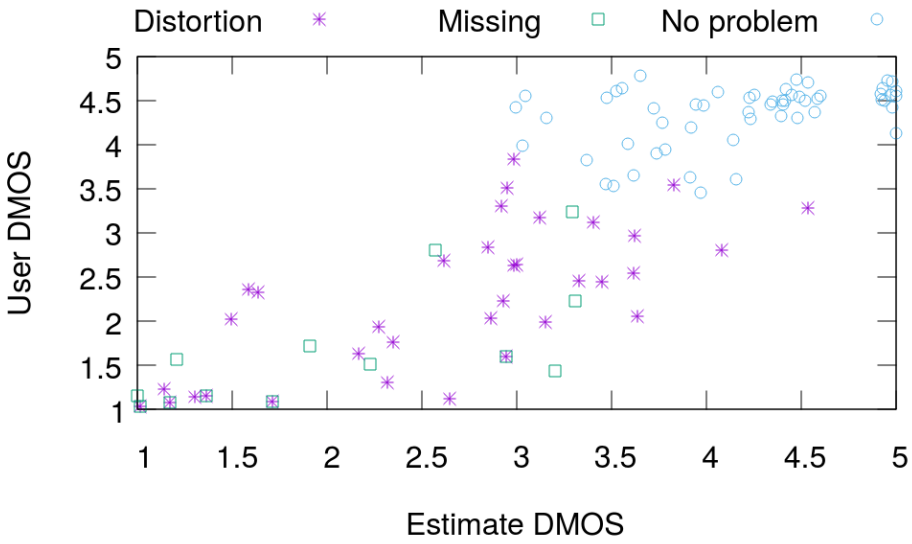



Figure 4.3: Validation of audio RTC QoE model and manual classification of audio impairments.

should consider a scenario of a real-time internet video call, show them the reference sample, and ask them to rate the quality of five transmitted samples. We give them the degradation category rating scale, from 1 to 5, to describe the perceived degradation: 1, very annoying; 2, annoying; 3, slightly annoying; 4, perceptible but not annoying; and 5, imperceptible. Each user rates a total of 30 samples, so tests require between 6 and 10 minutes to complete.

We requested for volunteers for this study through friends, at Technicolor labs and our lab mailing list, with answers from 40 users. We use a score normalization procedure similar to Wang et. al [123], to deal with users which do not use the whole range of grading options. First, we apply normalization by standard deviation per user. Then, we re-scale all normalized scores into range [1-5], obtaining user DMOS. Figure 4.2 shows compares user DMOS and estimated DMOS, with a strong correlation of 0.9547.

### 4.2.3 Audio RTC QoE model

There are many methods to estimate the perceived quality of a voice service. Signal based methods analyze the distorted signal to estimate listening quality, by either searching for known distortion patterns (e.g., ITU-T P.563) or by comparing it to the undistorted signal (e.g., Perceptual Speech Quality Measure (PESQ) [58]). Parametric

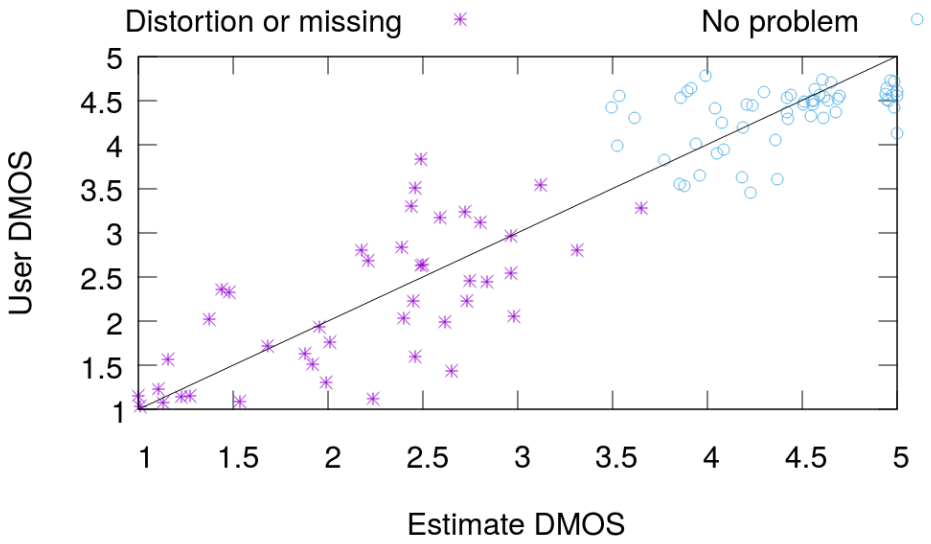



Figure 4.4: Audio RTC QoE model using adjusted PESQ MOS with $\delta = 0.5$

models, such as the E-model (ITU-T G.107), estimate listening quality based on several impairment factors such as one way delay, codec distortions, and others [99]. Here, we combine PESQ and E-model to account for both audio distortions and one way delays, following the approach of Lingfen et. al [111]. We obtain PESQ MOS by comparing the reference and received test samples, and one way delay by monitoring the median latency of WebRTC packets.

Similarly to video WebRTC experiments, we observed instances where STA1 was able to associate to the AP and communicate to STA2 through WebRTC, but could not sustain the audio call. For every experiment where we observe a WebRTC connection for 10 second but no audio traffic, we generate a sample with DMOS = 1.0, since there is no service.

**Model validation.** To understand how audio quality correlates with user opinion, we conduct a small user study. We select 12 reference audio clips, and 10 transmitted samples for each audio clip with different levels of audio quality. We explain users that they should consider a scenario of a real-time internet audio call, give them the reference sample and ask them to rate the quality of five transmitted samples. We give them the degradation category rating scale, from 1 to 5, to describe the perceived degradation. Each user rates a total of 60 samples, so tests require between 10 and 15 minutes to complete.

We requested native Dutch speaking volunteers (as the original audio clips are in

Dutch) for this study among friends and at Technicolor labs, with answers from 20 users. We apply the raw score normalization procedure described on §4.2.2 to obtain user DMOS. Figure 4.3 compares user DMOS and estimated DMOS per audio sample. We see that while for the majority of cases estimated DMOS is in line with user DMOS, there are cases where user DMOS is significantly higher than estimated DMOS. We listened to all degraded samples from this study and found two main impairments: voice distortion, such as when the voice "digitalizes"; and missing audio, such as when a word or phrase is partially or completely absent, as shown by the different markers on Figure 4.3. Our analysis of instances where none of the above impairments are present and user DMOS > estimated DMOS shows that the vast majority of transmitted samples are slightly fast/slower than original. This analysis suggests that PESQ is detecting audio distortions which users did not notice or care. We advise researchers to carefully review PESQ results, since PESQ may bias results if the application alters playback rate as part of its adaptation mechanism.

To counter this issue, we alter raw PESQ MOS values using a small $\delta$ parameter. We listen to all audio samples and increase PESQ MOS by $\delta$ whenever there is no voice distortion nor missing audio, or decrease PESQ MOS by $\delta$ otherwise. We find $\delta = 0.5$ minimizes the RMSE between user and estimated DMOS. Figure 4.4 shows estimated and user DMOS on the adjusted audio QoE model, with a correlation of 0.9191.

### 4.2.4 Web browsing QoE model

While there is extensive work showing that web browsing QoE is a result of many influence factors, end user waiting time is consistently identified as the main system influence factor for user QoE. Also, there are different QoE models based on the type of web activity: browsing through an extended session, opening a page, reading e-mails, or downloading a file [99]. Here, we consider the simple case of opening a single web page.

Most single-page web-QoE models consider the page load time (PLT) as the main application QoS metric. The PLT, however, is hard to measure since it is not always clear when users consider the page "ready". Here we use the PLT as measured by the *onload* event, since for the majority of pages, the *onload* event coincides with the user perception of the page being ready [120]. There is work suggesting that expectation [53] and page aesthetics [119] significantly influence web QoE. Since Wi-Fi quality cannot directly affect those, we remain confident on modeling MOS as a function of PLT.

We employ the ITU-T G 1030 single page web QoE model [61], which maps PLT

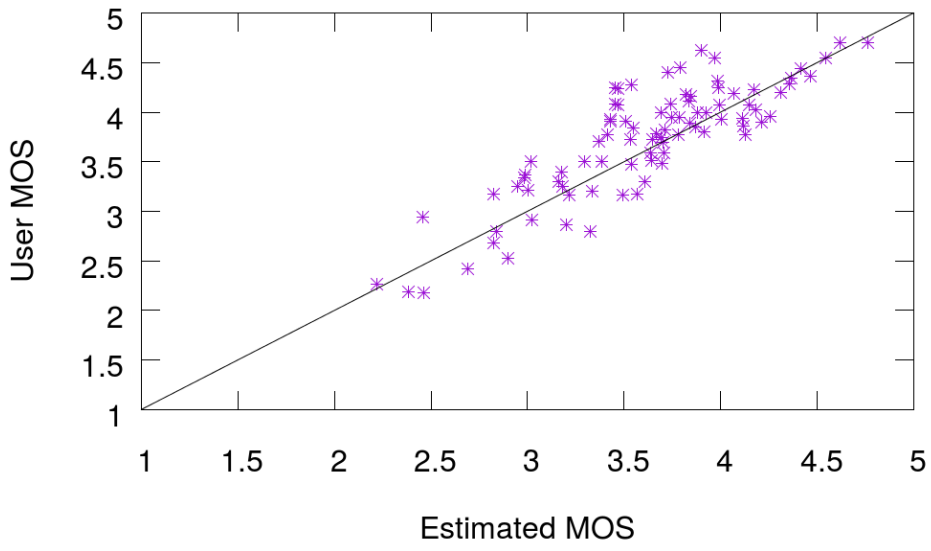



Figure 4.5: Validation of Web QoE model

into MOS in range [1-5] for fast, medium, or slow networks contexts. The network context expresses the user expectation of network speed, with users more tolerant to delays on slower network contexts. The ITU-T G 1030 proposes a logarithmic relationship between PLT and MOS, which is in line with most recent web QoE studies [35, 110]. We map PLT to MOS using Equation 4.2.3. We consider the medium network context, where $Min = 0.395$ and $Max = 38$, since home internet speed is usually lower than backbones or enterprise networks.

$$f_w(PLT) = 4.0 \times \frac{log(PLT/Min)}{log(Min/Max)} + 5.0 \tag{4.2.3}$$

During web browsing experiments, we limit PLT to a maximum of 10s. In case the web page does not finish loading within 10s, we do not have the precise PLT. In these cases, if the main HTML finished loading, we consider PLT = 10s. Otherwise, we consider that the service failed and set DMOS = 1.0.

**Model validation.** We validate the proposed model with data from a user study on Web QoE made by Bocchi et. al [12], which contains over 4000 PLT and user scores from 146 users across 41 websites and 8 network scenarios. We only consider users with at least 10 scores, and only pages with at least 10 scores. We aggregate results for each combination of page and network scenario to calculate MOS and mean PLT, which we use to obtain estimated MOS. Figure 4.5 shows the relationship between user and estimated MOS, with a correlation of 0.8529. This result suggests that PLT has a

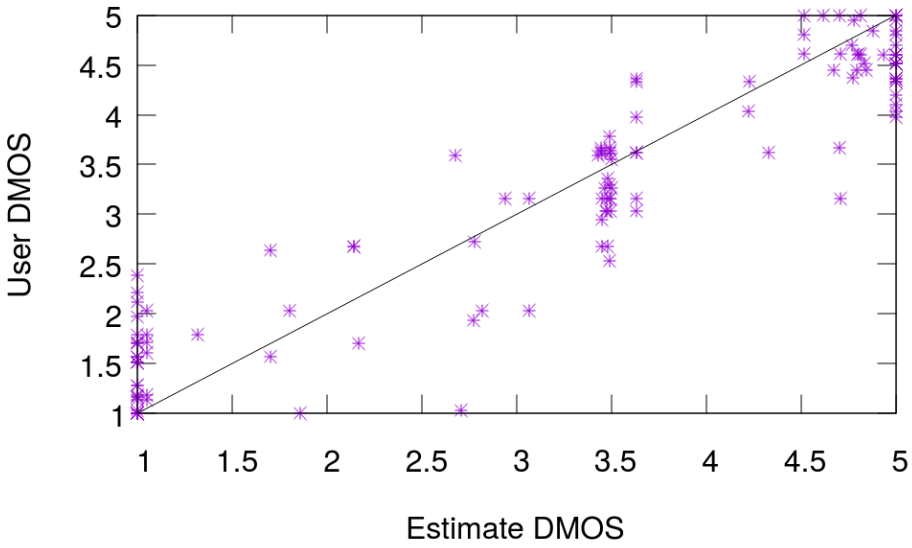



Figure 4.6: Validation of YouTube QoE model

strong correlation with user opinion, even though it is clearly not the only influence factor. Indeed, we verified that our model under-estimates MOS for pages with content "below the fold", not visible until you scroll the page.

### 4.2.5 YouTube QoE model

Video streaming is a popular internet application and YouTube is the most used video streaming service on the internet. While network impairments may cause frame losses on video RTC, YouTube uses TCP to reliably transmit content. YouTube uses a playback buffer to smoothly play the video and executions typically occur smoothly as long as video download rate is faster than playback rate. *Buffering events* occur when the playback buffer empties, when the player stops to refill the buffer. These instances are so disruptive that, in default configuration, YouTube attempts to prevent buffering events with adaptive bitrate streaming. In this scenario, the video player dynamically selects video bitrate during playback.

There are three main factors influencing QoE: average bitrate, join time, and buffering events [6, 121]. Most studies agree that join time does not drastically affect QoE, as users are more tolerant to initial delays. Therefore, we model YouTube QoE as a function of buffering events and video bitrate.

There is work on mapping individual video QoS metrics to MOS, but there is no model that maps multiple video QoS metrics to MOS. We consider two specialized

Table 4.1: $a_i, b_i, c_i$ parameters for Equation 4.2.4

| i | $\lambda$ | $a_i$ | $b_i$ | $c_i$ |
|---|---|---|---|---|
| 1 | $\lambda < 0.05$ | 2.97 | 0.74 | 2.03 |
| 2 | $0.05 < \lambda < 0.10$ | 3.07 | 0.96 | 1.93 |
| 3 | $0.10 < \lambda < 0.20$ | 3.17 | 1.55 | 1.83 |
| 4 | $0.20 < \lambda < 0.50$ | 3.21 | 1.66 | 1.79 |
| 5 | $\lambda > 0.50$ | 3.24 | 1.79 | 1.76 |

MOS models for YouTube quality: one using buffering events and the other using video bitrate.

**Buffering events.** We use the model proposed by Wamser et. al, which estimates MOS values given the number and duration of buffering events [121], shown in Equation 4.2.4. It considers $n$ as the number of buffering events, $\lambda$ as the ratio between total stalling time and total video elapsed time, and $a_i$, $b_i$, $c_i$ parameters according to Table 4.1.

$$f_{ys}(n, i) = a_i \times e^{-b_i \times n} + c_i, \forall i = 1, 2, 3, 4, 5 \tag{4.2.4}$$

**Video bitrate.** Similarly to Zinner et. al [132], we use SSIM to obtain estimated MOS scores to quantify the impact of video resolution. We compare the video at each resolution with the maximum resolution available ($1080hd$) to obtain video SSIM per resolution. Then, for a given video playback execution, we obtain a weighted average SSIM given by the fraction of time played at each resolution. We estimate MOS using Equation 4.2.2 and normalize it with Equation 4.2.1.

We model YouTube QoE as the minimum MOS between the estimated MOS of the models. By using the minimum MOS, we remain faithful to each model when only one type of impairment is present. We note that both types of impairment happen together in less than 10% of samples in our dataset.

**Model validation.** We validate the proposed model using a small user study on YouTube quality, made by Katsarakis et. al [69], where 16 users rated a total of 128 YouTube video executions under different network impairments. We apply the raw score normalization procedure described on §4.2.2, and compare normalized user scores and estimated DMOS in Figure 4.6. We find a correlation of 0.9268 using our proposed model, in comparison to 0.7272 and 0.8323 when using only video bitrate or buffering events respectively. This result suggests that it is necessary to account for

Table 4.2: Wi-Fi metrics measured on the access point.

| | Metric | Period | Feature name |
|---|---|---|---|
| AP metrics | % of time the medium is busy | 2 s | BUSY |
| | % busy due to Wi-Fi traffic | 2 s | $\text{BUSY}_{\text{WiFi}}$ |
| | % of busy due to non Wi-Fi traffic | 2 s | $\text{BUSY}_{\text{nonWiFi}}$ |
| Station metrics | PHY rate of last frame sent | 1 s | AvgTxPhy |
| | Frames sent / retransmitted to STA | 1 s | FDR |
| | Received signal strength indicator | 1 s | RSSI |

both types of video impairments.

## 4.3 Predicting the effect of Wi-Fi quality on QoE

This section shows how we predict the effect of Wi-Fi quality on QoE. We formulate the problem of predicting QoE as a regression problem. This way, we can predict the MOS degradation caused by Wi-Fi quality impairments. First, we build a training set where each sample is a vector that contains Wi-Fi metrics we passively collect from the AP labeled with the estimated QoE for each application. Then, we train models to predict the estimated QoE based on Wi-Fi metrics only. Finally, we validate the models with experiments in an uncontrolled environment, where we compare predicted and estimated QoE.

### 4.3.1 Building the training set

During each experiment, we passively measure the Wi-Fi metrics shown in Table 4.2 on the AP. For each application sample, we build a vector where each feature contains Wi-Fi metrics aggregated over an interval $T$ during the application execution, and the estimated QoE using the models in §4.2. We choose $T$ based on the duration of the application execution. For audio and video experiments, we create one feature vector per audio / video sample, with $T = 10s$ since the duration of audio / video samples is $\approx 10s$. For web browsing, we create one feature vector per web page access, using $T = 10s$ since this is the maximum page load time. For YouTube, we create one feature vector per video playback, using $T = 120s$ since we play each video for two minutes. We calculate the features BUSY, $\text{BUSY}_{\text{WiFi}}$, $\text{BUSY}_{\text{nonWiFi}}$, and AvgTxPHY using simple averages. For FDR, we calculate the fraction of non re-transmitted frames sent and the number of frames sent over the interval $T$. For RSSI, we use the last RSSI

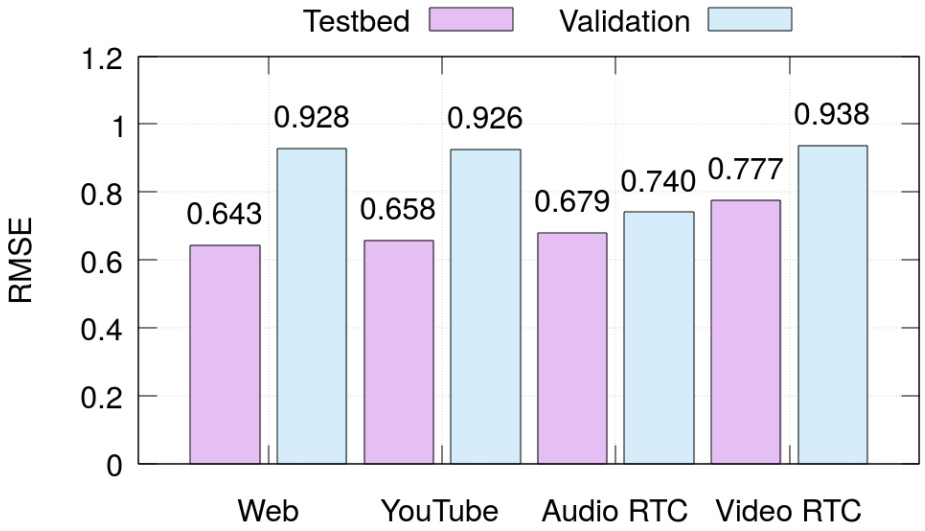



Figure 4.7: Validation

value. We use this choice of features to match the aggregation used in the deployment, described in Chapter 5.

### 4.3.2 Building the predictor

We opt to use support vector regression (SVR), since it e produces robust predictive models with low error for a wide range of problems. Since SVR is not scale invariant, we normalize features based on minimum / maximum values from dataset. We evaluate prediction accuracy using root mean squared errors (RMSE), as it is common practice on regression models.

Table 4.3: Frequency of feature in the best predictor with $K$ features, for $K \in [1-6]$

| **Feature** | **Video** | **Audio** | **Web** | **YouTube** |
|---|---|---|---|---|
| AvgTxPhy | 6 | 3 | 5 | 6 |
| FDR | 5 | 6 | 5 | 3 |
| BUSY | 2 | 3 | 4 | 3 |
| $\text{BUSY}_{\text{nonWiFi}}$ | 3 | 3 | 3 | 4 |
| $\text{BUSY}_{\text{WiFi}}$ | 3 | 4 | 2 | 4 |
| RSSI | 2 | 2 | 2 | 1 |

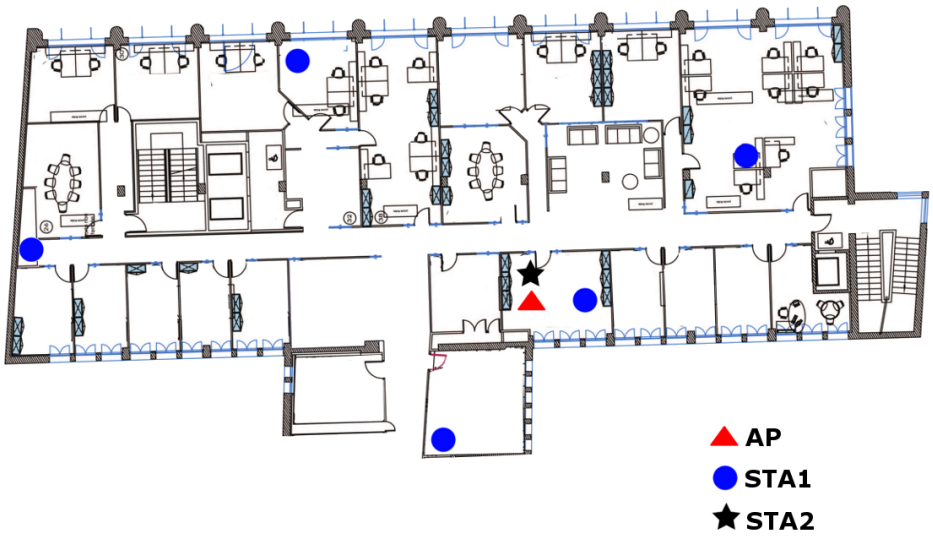
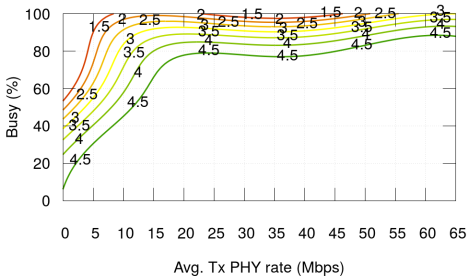


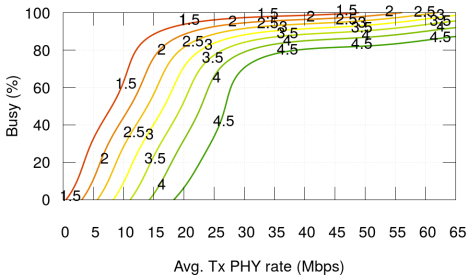

Figure 4.8: AP / STA1 position on validation dataset

**Features' predictive power.** We analyze which features hold more predictive power for each application. For each application, we train one SVR predictor for each possible combination of the six features, and evaluate its accuracy through ten folded cross-validation. Then, we observe how often each feature appears in the best predictor with $K$ features, $K \in [1-6]$, as shown in Table 4.3. AvgTxPHY is the most important feature on all applications except audio. We believe that's because audio QoE impairments only happen when not only AvgTxPhy is poor but also FDR. RSSI is the least important feature in the set. We believe this is because AvgTxPhy provides a better link speed estimation than RSSI (in line with findings from Patro et. al [89]). FDR is the most relevant feature for audio and the second most relevant for video and web, but it does not seems as relevant for YouTube. This is because low FDR causes jitter due to frame retransmissions, which may affect QoE of audio and video RTC, as well as web. On the other hand, YouTube should not suffer from these issues, since it can buffer during temporary delays.

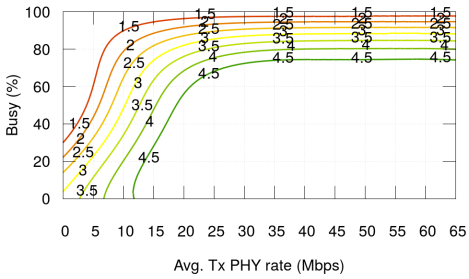
**Parameter selection.** We configure three SVR parameters. SVR uses the $C$ parameter to penalize misclassified samples and $\gamma$ parameter to set the range of influence of each support vector in the model. High values of $C$ and $\gamma$ are known to over-fit the dataset. The $\epsilon$ parameter regulates the error margin over which predictions incur into no penalty. We find the best combination of parameter through grid optimization, with tests where $C \in [1, 1000]$, $\epsilon \in [0.01, 1]$, and $\gamma \in [0.1, 100]$ at regular intervals. First, we use RMSE from ten fold cross-validation to track which combination of SVR



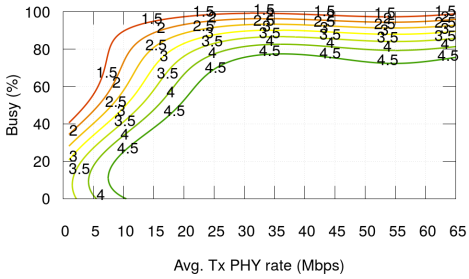

(a) Audio RTC




(b) Video RTC




(c) YouTube




(d) Web: average

Figure 4.9: Visualization of SVR models with two features: AvgTxPhy and BUSY.

parameters works better for each application predictor. We found that $\epsilon = 0.3$ works well for all applications, but several combinations of $\gamma$ and $C$ produce low RMSE values. Since SVR models with high $\gamma$ tend to generate high-variance models, we choose $\gamma = 3$ alongside the $C$ parameter which minimizes RMSE.

**Web page complexity.** For web browsing, we build one predictor per page. This is because the baseline PLT of different web pages vary by more than one order of magnitude (e.g. 0.287s on *bing.com*, 2.849s on *facebook.com*). For simplicity, we summarize web QoE results using three different pages that vary significantly in terms of complexity: google.be (web simple), facebook.com (web average), and amazon.fr (web complex).

**Model visualization.** To understand how Wi-Fi metrics influence the decision of predictors, we show the decision curves for SVR models using two features: "$AvgTxPhy$ and $BUSY$" on Figure 4.9. The SVR models learn similar boundary conditions. When Wi-Fi conditions are perfect, with BUSY near 0 and "Avg. Tx PHY rate" at maximum, we observe a predicted MOS of 5 on all predictors. Similarly, when either AvgTxPhy is close to 0 or BUSY close to 100%, we observe a predicted MOS of 1.

Figures 4.9b and 4.9c shows the SVR models for video RTC and YouTube QoE. We

observe that both models present reduced QoE when Avg Tx PHY rate below 15 Mbps. This indicates that these applications are sensitive to Wi-Fi bottlenecks. We also see that YouTube is more resilient to Wi-Fi impairments. This is due to adaptive bitrate selection dynamically adjusting video bitrate to match available bandwidth, avoiding buffering events. Figure 4.9a shows the SVR model for predicting audio QoE. Audio impairments only happen when Wi-Fi conditions are extremely poor. For Web QoE, we show decision curves for the SVR predictor considering pages of average complexity (Figure 4.9d). We observe that web QoE is sensitive to both Avg. Tx PHY rate and medium occupancy.

### 4.3.3 Validation

To validate our predictors, we generate the "Office" dataset in an uncontrolled environment (i.e. our lab in Paris), over four weeks in 2016. We vary the position of STA1 according to Figure 4.8 and automatically execute experiments every 30 minutes, from 7am to 10pm, in order to observe different levels of link speed and interference. Each experiment consists of tests of two applications, web and YouTube or audio/video RTC, and consists of multiple runs as described in §4.1.3. Then, we select 22 experiments where we observe the most diverse set of Wi-Fi parameters. We follow the same procedure as to generate the testbed dataset to obtain QoE estimations.

We evaluate prediction errors on the Wi-Fi testbed dataset through ten-fold cross-validation, using the best parameters found on §4.3.2. For the validation dataset, we learn one predictor using the testbed dataset and apply it on the office dataset. Figure 4.7 shows prediction errors for both Wi-Fi testbed and validation datasets. We observe validation RMSE below 0.94 for all predictors, while testbed RMSE is below 0.78 for all predictors.

## 4.4 Summary

In this chapter, we study the effects of Wi-Fi quality on QoE of four applications: web browsing, YouTube streaming, and audio / video RTC. We create QoE predictors based on Wi-Fi metrics commonly available in commercial APs. We generate the training set by performing controlled experiments in a Wi-Fi testbed, covering a wide range of Wi-Fi configurations, while monitoring Wi-Fi and application QoS metrics. We review and carefully select QoS / QoE models to obtain MOS estimations and validate each model with a small user study. We implement SVR models to predict QoE from Wi-

Fi metrics and analyze the impact of individual features on the prediction task. We evaluate predictors' accuracy using cross-validation and find RMSE between 0.643 and 0.777 on the testbed dataset, and RMSE between 0.740 and 0.938 on the validation dataset.

# Chapter 5

# Characterization of Wi-Fi quality in the wild

This chapter analyzes the effects of Wi-Fi quality on QoE in the wild. We analyze Wi-Fi measurements collected by a Wi-Fi monitoring system developed by Technicolor. First, we overview the system architecture and describe the dataset. Then, we use the models from Chapter 3 and predictors from Chapter 4 to evaluate how this affects Wi-Fi quality in the wild.

## 5.1 System overview

Figure 5.1 shows the architecture of the Wi-Fi monitoring system developed by Technicolor. The ISP owns the home AP, which includes a module that uploads Wi-Fi metrics including radio and station stats to the Wi-Fi monitoring system. The AP samples connected stations every second, including the Wi-Fi metrics shown in Table 4.2. It does not sample the FDR metric, since the AP does not export it per station. The AP uploads the set of samples to the monitoring server every 30s, including the station identifier, timestamp, Tx/Rx datarates, and station capabilities.

The AP uploads samples for inactive stations even though some Wi-Fi metrics need user traffic. Therefore, the system filters out invalid station samples taken at such periods of inactivity. The system aggregates valid samples over 30s bins per station, storing the aggregated 30s station samples. These station samples are processed to infer characteristics of Wi-Fi quality. They are annotated with post-processing metrics like link capacity, available bandwidth, etc. From here onward, we refer to these aggregated 30s station samples as *samples*, unless otherwise noted. They are presented

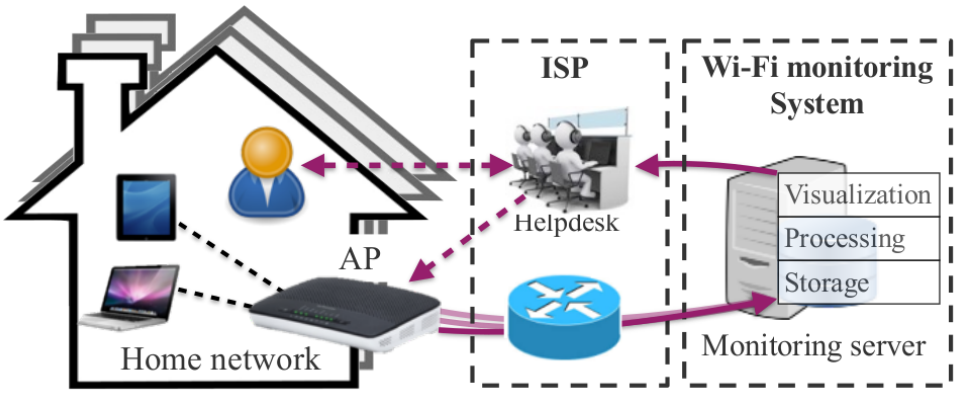



Figure 5.1: Wi-Fi monitoring system architecture.

in a visualization system, which allows helpdesk operators to detect and diagnose poor Wi-Fi quality.

## 5.2 Deployment dataset

We collect station samples from 832 APs of customers of a large Asian-Pacific residential ISP, with 6086 stations connected to these APs, during the month of September, 2016. We analyze a total of 9 million station samples.

First, we look at deployment characteristics regarding station usage and technology. Figure 5.2 shows the number of samples per stations and per AP, where one AP sample contains one or more valid station samples. We find stations with very little activity: we have less than 10 samples for 7.8% of stations. This is because some home Wi-Fi appliances seldom transfer data over Wi-Fi networks, such as "smart scales", and there is not enough traffic to provide valid samples. There are also APs with a small number of samples (2.16% had less than one hour of data), although we rightfully observe significantly more samples per AP than per station.

We investigate how frequently stations are active. Figure 5.3 shows the cumulative fraction of the number of active days per stations / APs. We consider one station *active* if we observe at least 5 minutes of activity over 24 hours (similarly for the AP). 23.6% of stations have one or less active days, which indicates a device belonging to visitors or sporadically used.

We focus our study on APs and stations which actively use the Wi-Fi network, so we filter stations and APs which are not regularly used. We only consider stations with at least two active days per week (68.44%) and APs with at least five active days

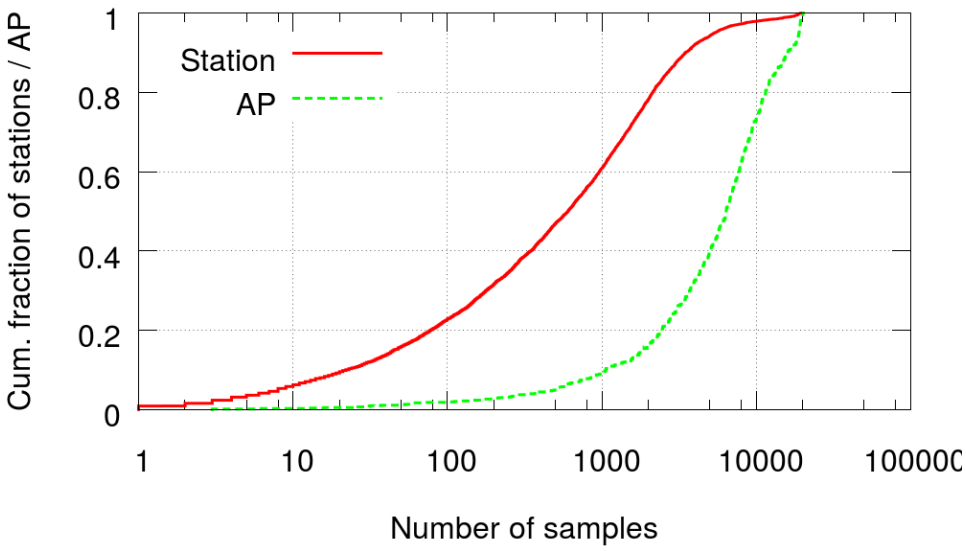



Figure 5.2: Number of samples per station and per AP.

Table 5.1: Station Wi-Fi technology on deployment.

| Technology | Before filtering | | After filtering | |
|---|---|---|---|---|
| | stations | samples | stations | samples |
| .11n 1x1 | 3531 | 3793K | 2599 | 3694K |
| .11n 2x2 | 2384 | 5009K | 1903 | 4936K |
| .11g | 187 | 197K | 100 | 191K |
| **Total** | **6102** | **9000K** | **4602** | **8821K** |

per week (92.29%). Table 5.1 shows the number of stations observed per technology before and after filtering, as well as the number of samples. Notice that we retain 98.01% of all samples, although we filter 24.58% of stations. Notice that over 46% of .11g stations have less than 2 active days per week, suggesting that stations with legacy technology are not frequently used.

## 5.3 Wi-Fi quality metrics

In this section, we analyze the distribution of link-layer Wi-Fi quality and network QoS metrics.

**Medium occupation.** We use the APs' CCA counters to measure periods where the medium is sensed busy. BUSY is the fraction of time the medium is sensed busy. We

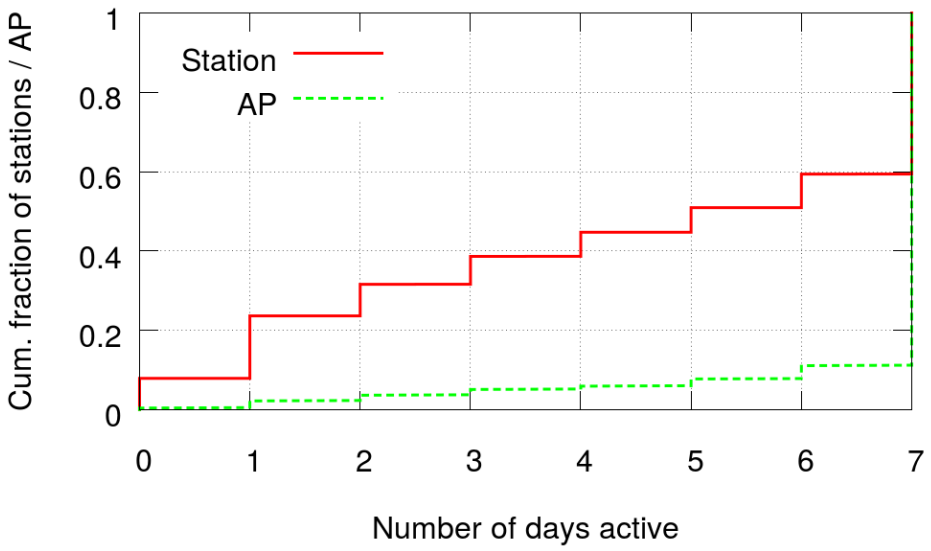



Figure 5.3: Number of days with collected data per stations / AP

further distinguish on the cause of the medium occupation. $\text{BUSY}_{\text{WiFi}}$ is the fraction of time the medium is busy due to decodable Wi-Fi traffic, and $\text{BUSY}_{\text{nonWiFi}}$ accounts for other reasons (e.g., non decodable Wi-Fi, non WiFi RF devices). For brevity, we refer to these cases as *Wi-Fi interference* and *non Wi-Fi interference*. Figure 5.4 shows the distribution of BUSY, $\text{BUSY}_{\text{WiFi}}$, and $\text{BUSY}_{\text{nonWiFi}}$ in the deployment. We observe that $\text{BUSY}_{\text{WiFi}} > 2\%$ on 99.3% of samples. This is likely due to management traffic from nearby APs such as beacon announcements. The median values for $\text{BUSY}_{\text{WiFi}}$ and $\text{BUSY}_{\text{nonWiFi}}$ are similar: 6 and 5%, respectively. The $90^{th}$ percentile, however, is very different: 15% for $\text{BUSY}_{\text{WiFi}}$ and 28% for $\text{BUSY}_{\text{nonWiFi}}$. This indicates that non Wi-Fi interference can degrade Wi-Fi quality with greater intensity. Non Wi-Fi interference is more often the root cause during contention. We find medium occupation above 50% on 7.4% of samples, with only 1% of AP samples with $\text{BUSY}_{\text{WiFi}} \geq 50\%$ and 4.2% with $\text{BUSY}_{\text{nonWiFi}} \geq 50\%$.

**Average Tx PHY rate.** Stations' average Tx PHY rate represents the link speed to the AP. A high average Tx PHY rate indicates good signal and channel conditions. The station capabilities dictate the PHY rate values available. Figure 5.5 shows the distribution of average Tx PHY rate for 802.11n 1x1, 802.11n 2x2, and 802.11g technologies. We observe many samples where the average Tx PHY rate is equal to existing PHY rate values. We observe 56%, 44%, and 20% of samples where average Tx PHY rate is maximum for stations with .11n 1x1, .11n 2x2, and .11g

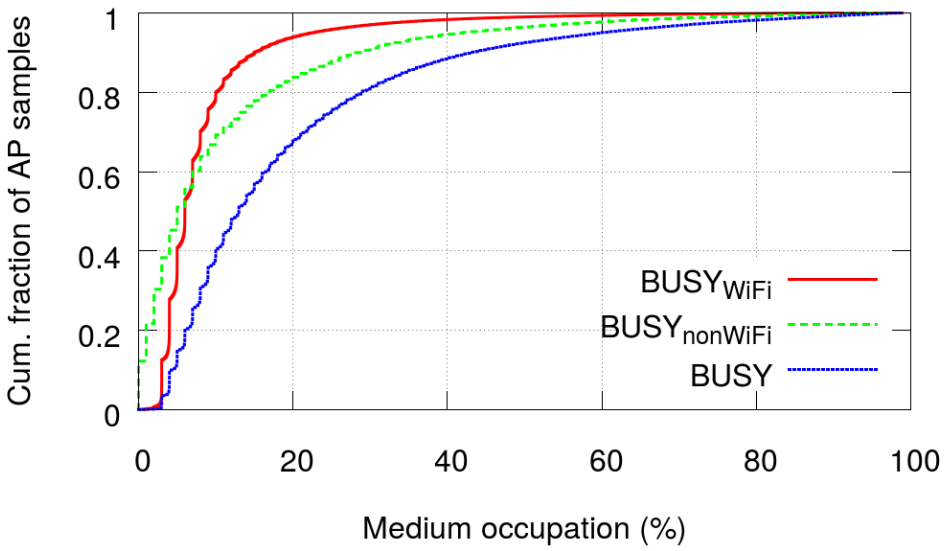



Figure 5.4: Distribution of medium occupation in the deployment dataset.

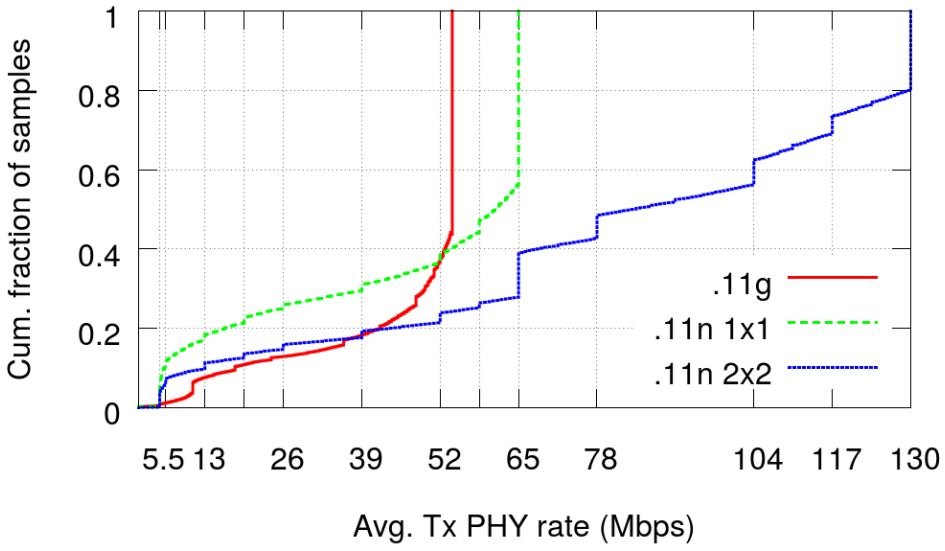



Figure 5.5: Distribution of Avg. Tx PHY rate for .11n and .11g stations.

technologies, respectively. Comparing stations with 802.11n 1x1 and 2x2 technology, we see significantly less samples with maximum PHY rate for .11n 2x2. This is because MIMO requires not only good channel conditions but also spatial diversity to enable communication on two spatial streams. On the other hand, overall speed is much higher for .11n 2x2 stations: 72% of samples have Avg. Tx PHY rate $\geq 65 Mbps$ on .11n 2x2. We observe 11.5% and 7.3% of samples with average PHY rate $\leq 6.5 Mbps$

for stations with .11n 1x1 and 2x2 technology. 6.5 Mbps is the lowest .11n modulation, but the AP can use legacy .11b rates (1, 2, 5.5 Mbps) to increase reception probability. Most of these samples are for PHY rate = 5.5 Mbps. This is because this AP rarely uses PHY rate < 5.5 Mbps when communicating with .11n stations. We also observed this behavior during controlled experiments in the Wi-Fi testbed.

**Link utilization.** Figure 5.6 shows the stations' datarate distribution for both TX (station download) and RX (station upload). Overall, we observe that the majority of samples have a small amount of traffic. We observe 51% of download samples with less than 10 Kbps, and 69% with less than 100 Kbps. This is because we collect samples even when the device is unattended and only background processes generate traffic. We observe more download than upload traffic. We find that most samples contain upload datarate below 1 Mbps (99.91%) and download datarate below 10 Mbps (99.72%). This is likely due to rate limits on the access link.

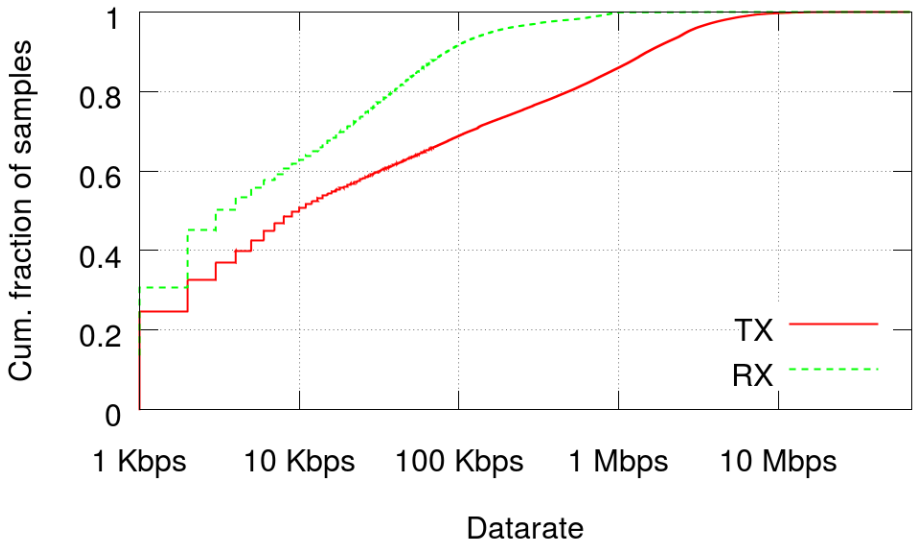



Figure 5.6: Distribution of stations' TX / RX datarates.

For each sample, we calculate link capacity and available bandwidth using Equation 3.4.2. Then, we divide the download datarate by the available bandwidth to calculate the link utilization. Figure 5.7 shows the link utilization for samples with datarate > 100 Kbps for different station technologies. We observe that .11n stations have lower link utilization, since they have greater link capacity. We find that 2.6%, 2.1%, and 1.7% of samples from .11g, .11n 1x1, stations have link utilization above 70%. In these cases, it is likely that the Wi-Fi link bottlenecks internet access. Assum-

ing a 10 Mbps access link, we estimate that the Wi-Fi link limits available bandwidth on 12.6%, 22.4%, and 12.9% of .11g, .11n 1x1, and .11n 2x2 stations' samples.

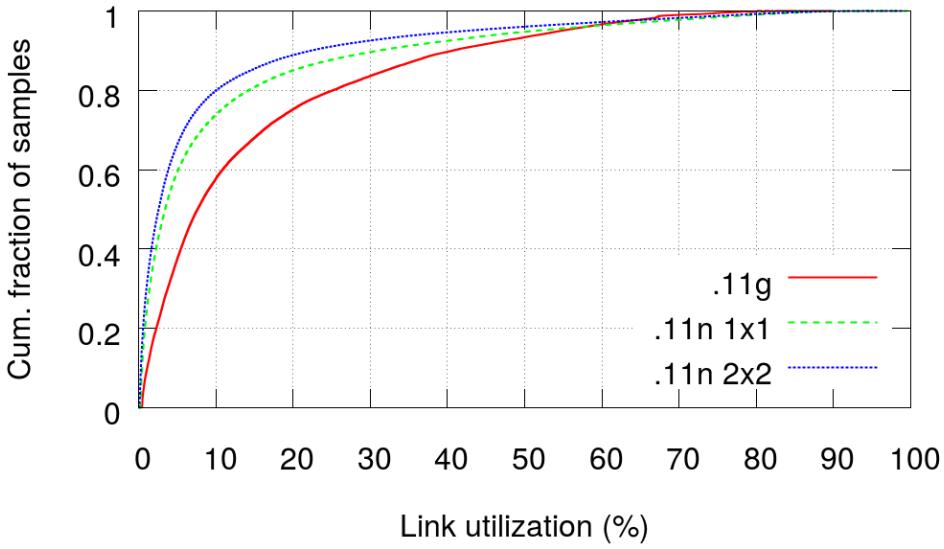



Figure 5.7: Distribution of link utilization of .11n and .11g stations for samples with TX datarate > 100 Kbps.

## 5.4 QoE predictions per application

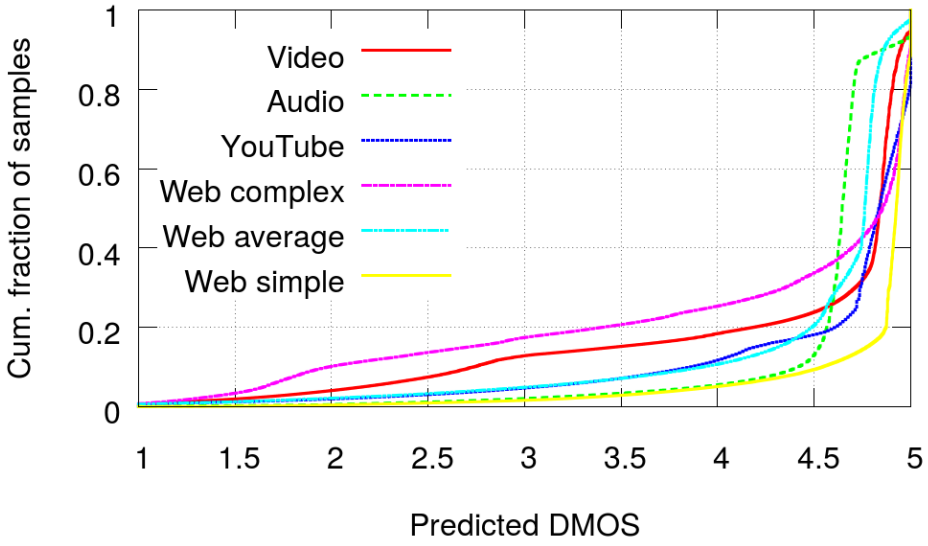



Figure 5.8: Predicted DMOS frequency in the wild.

Unfortunately, this dataset does not contain FDR. Therefore, we train SVR predictors using Avg. Tx-PHY, WiFi, and nonWiFi, instead. We build predictors with SVR parameters found in §4.3.2 and find RMSE between 0.780 and 0.842. We build predictors for YouTube, web browsing, and audio and video RTC. To account for web page complexity, we build three web predictors: web simple, representing very lightweight pages such as search engines front pages, we use samples of page loads of google.com for building this predictor; complex, representing pages with many objects / images (amazon.fr in our case); and average, representing pages of intermediary complexity (facebook.com).

Figure 5.8 shows the cumulative fraction of predicted DMOS per application in the deployment. We observe that over 66% of samples contain predicted QoE above 4.5 for all applications. This is reassuring as most of the time, Wi-Fi works well for all applications. We observe instances where predicted DMOS $< 3.0$ on all applications. We predict DMOS below 3.0 for web complex and video RTC on 17.4% and 12.8% of samples, for YouTube and web average on 4.9% and 4.7% of samples, and for audio RTC and web simple on 2.1% and 1.6% of samples.

The difference in frequency between applications is due to the application sensitivity to Wi-Fi impairments observed in the deployment. All samples with avg. phy rate up to 6.5 Mbps show predicted DMOS below 3.0 for web complex and video RTC, on a total of 9.14% of all deployment samples. On other predictors, this only occurs when we also observe interference.

## 5.5 QoE across applications

We saw that some applications require better Wi-Fi quality, with a larger fraction of samples where predicted DMOS $< 3.0$. Now, we would like to investigate how the same Wi-Fi quality affects QoE for different applications. Figure 4.9 hints that there should be similarities among QoE predictors: predicted DMOS is close to 1.0 as medium occupation gets closer to 100% for all applications. We would like to automatically detect the similarities and differences of QoE across applications.

We generate a synthetic set by sweeping all combinations of Wi-Fi parameters, namely: AvgTxPhy, $\text{BUSY}_{\text{WiFi}}$ and $\text{BUSY}_{\text{nonWiFi}}$. We do not generate samples where $\text{BUSY}_{\text{WiFi}} + \text{BUSY}_{\text{nonWiFi}} > 100$. For each Wi-Fi combination, we generate a sample with six features, one for the predicted DMOS for each application. Then, we apply a clustering algorithm to discover clusters of samples. Finally, we analyze the cluster

centroids to manually label each cluster.

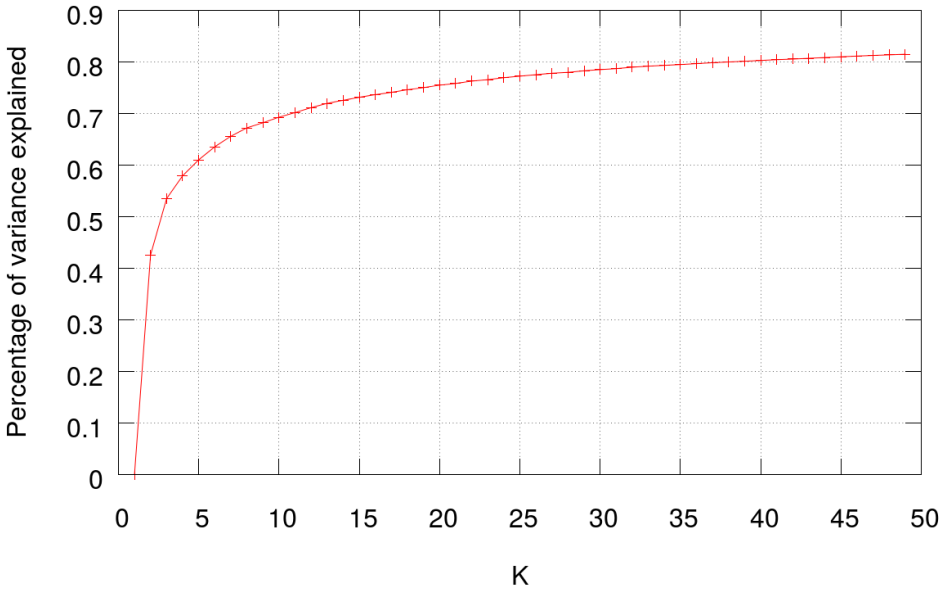


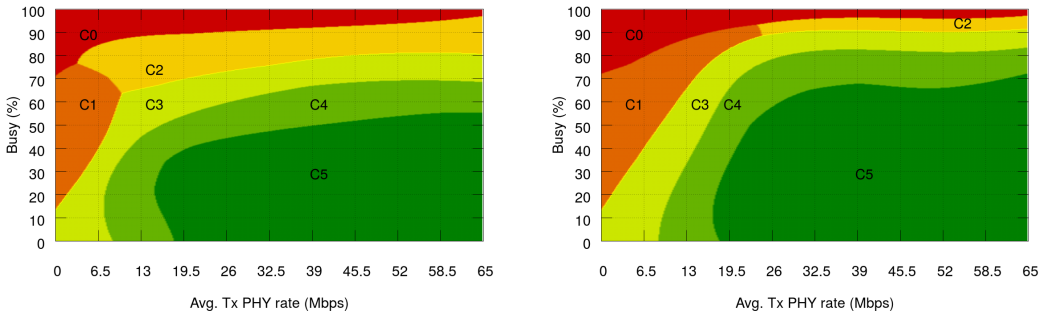

Figure 5.9: Percentage of explained variance for clustering with different K values

We find clusters using the K-means algorithm. Choosing the number of clusters (K) is difficult in this case. We do not expect to find rigid borders between clusters, since the QoE predictors produce continuous values between 1 and 5. Figure 5.9 shows the fraction of explained variance for K values between 2 and 30. This metric should monotonically increase as we increase K. One should choose the number of clusters so that adding another cluster does not significantly improves the model. We see that this metric increases very slowly for K>6. Furthermore, we find clusters with K=6 which we can manually label without much difficulty.

Table 5.2 show cluster centroids for K=6, ordered by the Euclidean norm. We manually analyze each cluster to identify if there is a semantic meaning for each cluster. Cluster C5 has DMOS $\geq$ 4.3 across applications, whereas cluster C0 has DMOS $\leq$ 1.8 across applications. Fortunately, nearly 83% of deployment samples belong to cluster C5, and only 0.39% to cluster C0. Cluster C4 has DMOS $\geq$ 3.9 across applications, although estimated DMOS for web complex $\approx$ 3.0. Cluster C3 has estimated DMOS $\leq$ 3.6 for all but two applications: audio and web simple. Cluster C2 and C1 has DMOS $\leq$ 3.0 on five applications, with the exception in audio for C2 and web simple for C1. Figures 5.10a and 5.10b shows cluster disposition on Wi-Fi QoS space, considering that only Wi-Fi interference or non Wi-Fi interference generates medium occupancy,

Table 5.2: Cluster centroids in QoE space

| | C0 | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|---|
| **Video** | 1.42 | 1.42 | 2.98 | 3.63 | 4.55 | 4.71 |
| **Audio** | 1.62 | 2.62 | 3.61 | 4.27 | 4.73 | 4.83 |
| **YouTube** | 1.35 | 1.49 | 2.51 | 3.37 | 4.43 | 4.87 |
| **Web complex** | 1.14 | 1.36 | 1.55 | 2.18 | 2.98 | 4.29 |
| **Web average** | 1.19 | 1.82 | 1.52 | 2.74 | 3.90 | 4.91 |
| **Web simple** | 1.74 | 4.48 | 2.70 | 4.30 | 4.74 | 4.98 |
| **Freq.** | 0.39% | 0.83% | 0.51% | 7.21% | 8.13% | 82.93 % |




(a) Wi-Fi interference

(b) NonWi-Fi interference

Figure 5.10: Clusters on Wi-Fi QoS space

respectively.

**Poor QoE samples.** The DMOS predictors help us understand if a given Wi-Fi scenario is adequate for executing one application. However, we would like to obtain a predictor that detects if Wi-Fi quality degrades QoE in general. The output of this predictor should simply be if Wi-Fi quality is good or not. We use the clusters to define this predictor. Clusters C3,C2,C1, and C0 contain samples where DMOS $<$ 4.0 on multiple applications. Therefore, we classify samples from clusters C3,C2,C1 and C0 as *poor QoE* samples. We call samples from cluster C4 and C5 as *good QoE* samples. Using this definition, we classify 8.94% of deployment samples as poor QoE samples, and 91.06% as good QoE samples.

Figure 5.11 shows the fraction of poor QoE samples per station. The majority of stations (60%) have less than 5% poor QoE samples, therefore, Wi-Fi is unlikely to impair user experience in those cases. 14% of stations, however, have more than 25%

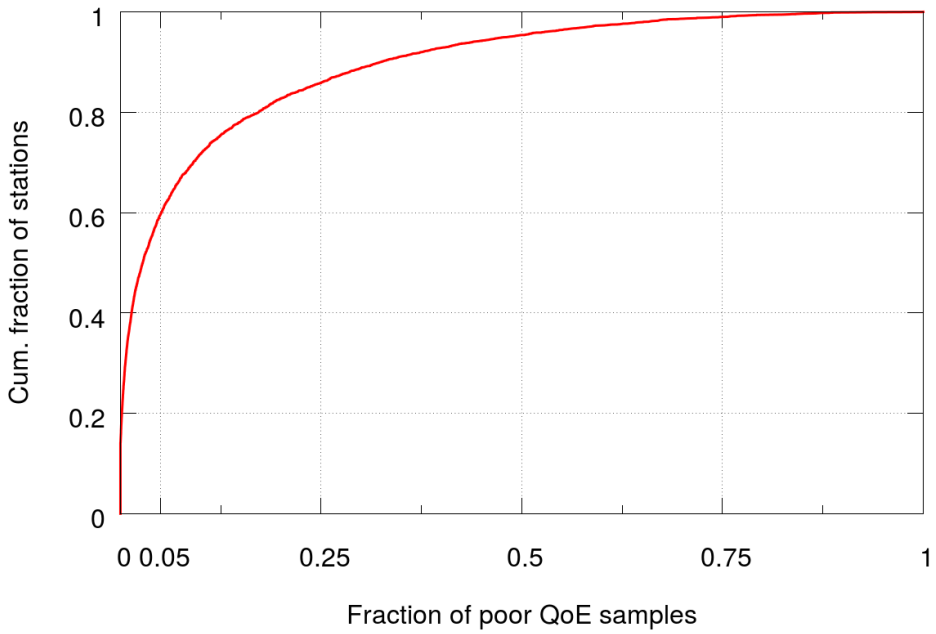



Figure 5.11: Fraction of poor QoE samples per station

poor QoE samples, which represent frequent Wi-Fi impairments. For the rest of this analysis, we focus on stations with at least 5% poor QoE samples.

## 5.6 Characterization of poor QoE events

We characterize how poor QoE samples occur over time. Figure 5.12 shows the inter-arrival time of samples and of poor QoE samples. In general, samples are not always evenly spaced due to stations' inactivity. 74% of all samples occur adjacently and 13% of samples have an inter-arrival time above 2 minutes. For poor QoE samples, 52% occur adjacent to another poor QoE sample, while 28% have inter-arrival time above 2 minutes. This indicates that poor QoE samples, in general occur in groups.

**Poor QoE event.** We define a poor QoE event as a series of poor QoE samples spaced by no more than a time threshold $T$. We use Figure 5.12 to decide the threshold $T$. We choose $T = 2$ minutes since a large fraction of poor-QoE samples have inter-arrival time $\leq 2$ minutes, but this fraction increases slowly for larger inter-arrival times. Notice that some good QoE samples may occur in-between poor QoE samples. Even in these cases, users are likely to experience poor quality during the poor QoE event.

Figure 5.13 shows the duration of poor QoE events, for T = 2 minutes. The majority of poor QoE events are short: 78% have duration of two minutes or lower.

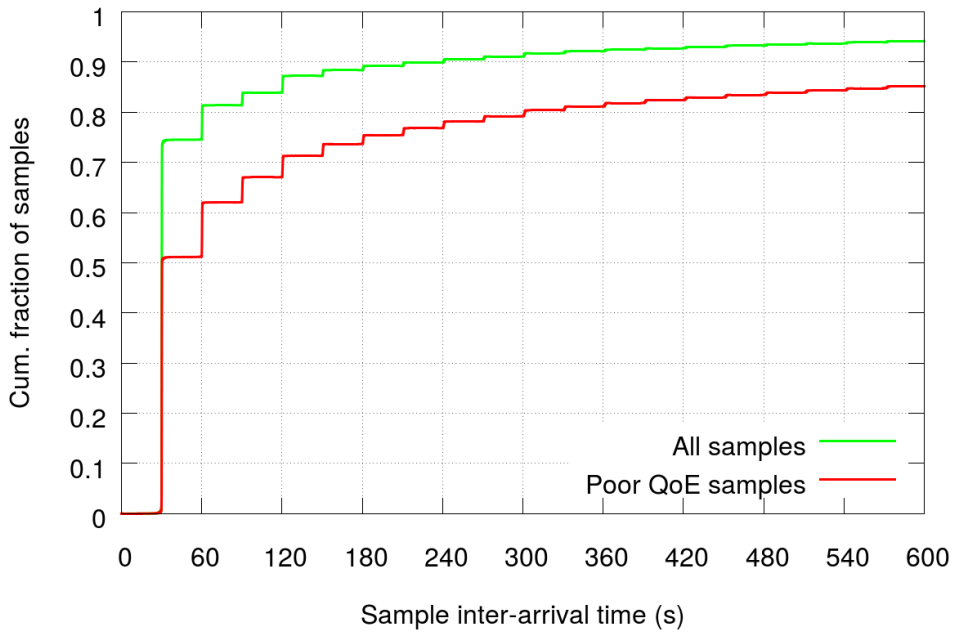



Figure 5.12: Sample inter-arrival time.

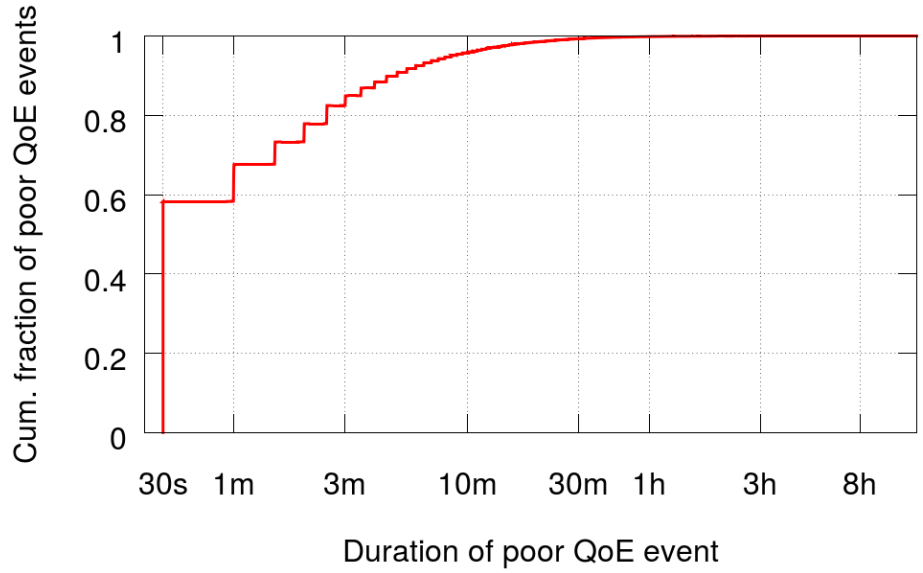



Figure 5.13: Duration of poor QoE events for $T = 2\ min$.

There are many possible reasons for short-lived poor QoE events. Mobile stations going in or out of Wi-Fi range are likely to produce short-lived poor QoE events. These can also be due to temporary Wi-Fi impairments, such as those caused by a peak of Wi-Fi interference. We observe a number of long poor QoE events, some stretching over

several hours. These more persistent Wi-Fi problems can happen when either the station is poorly located or the AP is using a busy Wi-Fi channel.

We categorize poor QoE events into three distinct classes. Short-lived poor QoE events are those with duration below or equal 2 min. We divide other events between consistent and intermittent. Intermittent events have good QoE samples in-between poor QoE samples. If the fraction of poor QoE samples during a poor QoE event is greater or equal to 80%, we call it a consistent poor QoE event, otherwise it is an intermittent poor QoE event.

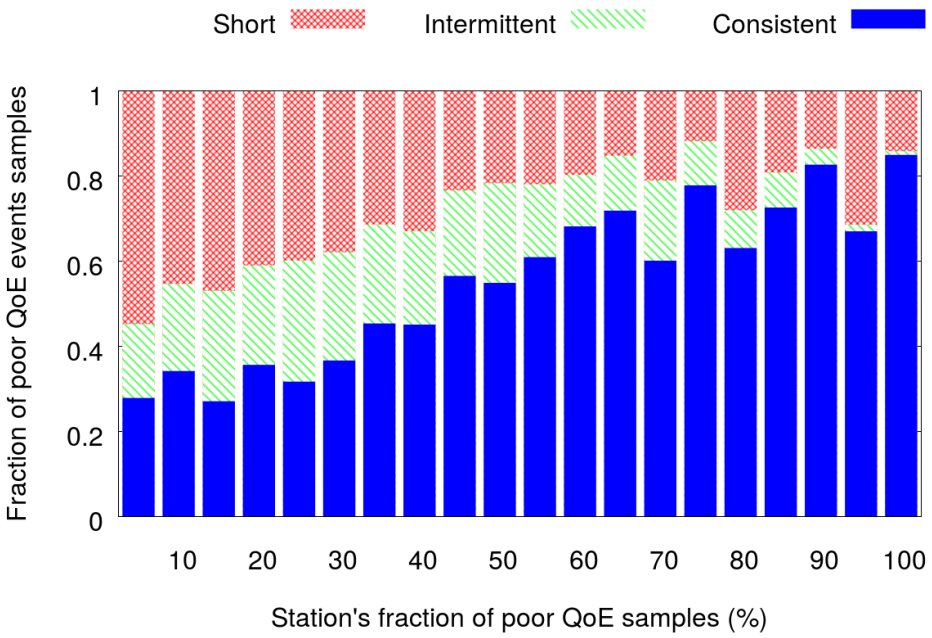



Figure 5.14: Classification of poor QoE samples per group of station.

Figure 5.14 shows the division between short, intermittent, and consistent poor QoE events. We group stations based on their fractions of poor QoE samples. We observe that stations with a small fraction of poor QoE samples have many short poor QoE events. For stations with fraction of poor QoE samples below 5%, 55% of poor QoE samples belong to short poor QoE events. Short-lived poor QoE events are difficult to detect and diagnose, since it is unclear if the root cause was fixed or temporarily absent. These stations require a long-term monitoring approach to detect and diagnose Wi-Fi problems. We also observe that stations with a larger fraction of poor QoE samples have less intermittent events and more consistent poor QoE events. For stations with fraction of poor QoE samples above 50%, we observe, on average, 68% of poor QoE samples on consistent poor QoE events. While this general rule holds

true for the majority of stations, there are some exceptions. Some stations with a very high fraction of poor QoE samples have nearly all samples on short poor QoE events. We verified these cases and found that this occurs because they generate traffic for very short periods. This prevents us from classifying poor QoE samples into long poor QoE events.

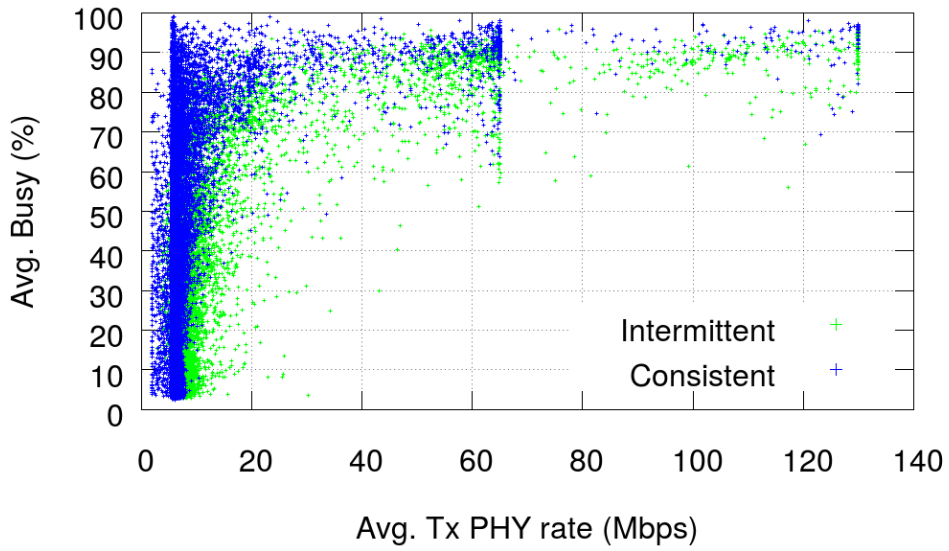



Figure 5.15: Scatter plot of Avg. Tx PHY rate and Busy for consistent and intermittent poor QoE events.

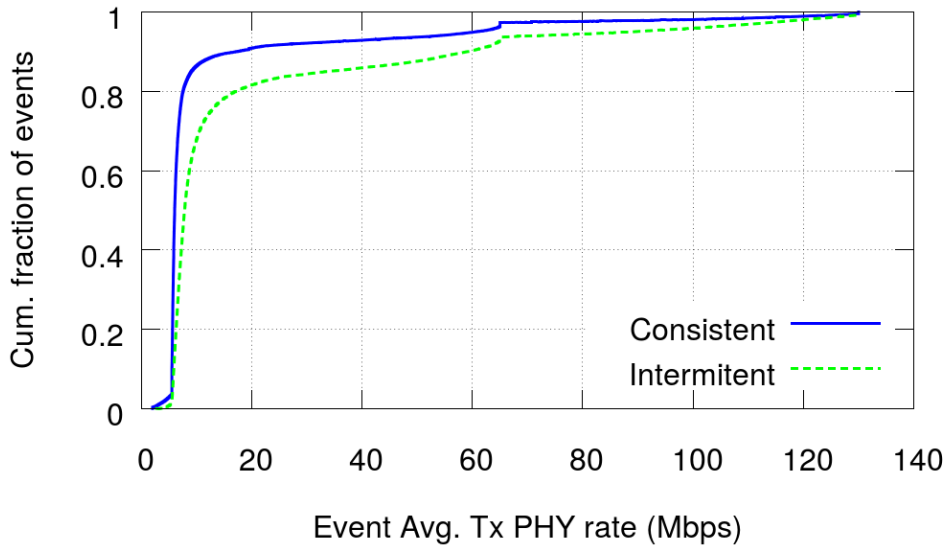



Figure 5.16: Distribution of poor QoE events' average PHY rate.

**Diagnosis of poor QoE events.** Next, we investigate what is the most likely cause for intermittent and consistent poor QoE events. Figure 5.15 shows average PHY rate and BUSY for consistent and intermittent poor QoE events. In general, poor QoE events have either Avg. Tx PHY rate $<$ 15 Mbps or Avg. Busy $>$ 60%. Figure 5.16 shows the distribution of the average PHY rate of consistent and intermittent poor QoE events. We see that 88.5% and 78.3% of consistent and intermittent poor QoE events have Avg. Tx PHY rate $<$ 15 Mbps. We diagnose these events as caused by low Avg. Tx PHY rate. Intermittent poor QoE events have Wi-Fi metrics closer to the decision boundary between clusters C3-C4. We find 7.42% and 25.49% of consistent and intermittent poor QoE samples with avg. Tx PHY rate between 8 and 15 Mbps.

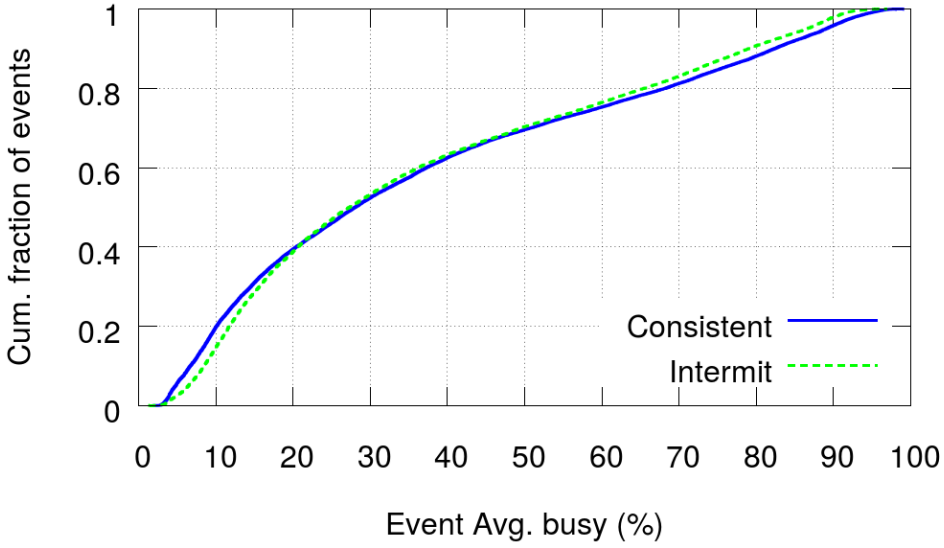



Figure 5.17: Distribution of poor QoE events' average BUSY.

Figure 5.17 shows the average medium occupation of consistent and intermittent poor QoE events. We see that both distributions are similar. We observe that 24.8% and 23.6% of consistent and intermittent poor QoE events have BUSY$\geq$ 60%. We diagnose these events as caused by high medium occupancy. We investigate cases where we find both low Avg. Tx PHY rate and high medium occupation, and found that this occurs on 14.2% and 5.0% of consistent and intermittent poor QoE events. This occurs more often on consistent poor QoE events because both metrics needs to improve to obtain a good QoE sample.

We further investigate the source of medium occupancy for the cases where we don't detect low Avg. Tx PHY rate. Figure 5.18 shows the distribution of average $\text{BUSY}_{\text{WiFi}}$

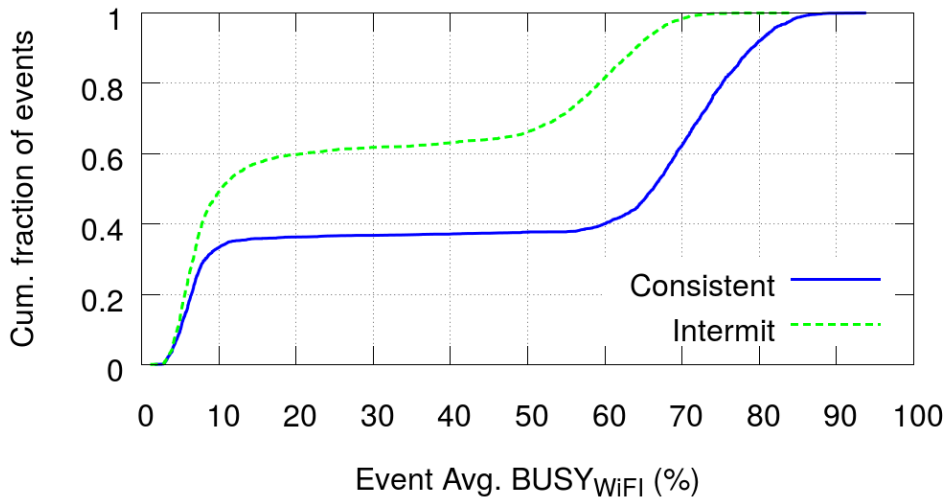



Figure 5.18: Distribution of poor QoE events' average $\text{BUSY}_{\text{WiFi}}$, where Avg. Tx PHY $>$ 15 Mbps.

for poor QoE events where Avg. Tx PHY rate $>$ 15 Mbps. We observe that 62.5% and 33.8% of consistent and intermittent poor QoE events have $\text{BUSY}_{\text{WiFi}} > 50\%$. We diagnose these events as caused by Wi-Fi interference. We also observe that 36.4% and 59.6% of consistent and intermittent poor QoE events have average $\text{BUSY}_{\text{WiFi}} < 20\%$. In these cases, Wi-Fi interference is unlikely the source cause. We verified these cases and found that, in these cases, $\text{BUSY}_{\text{nonWiFi}} > 40\%$. Therefore, we diagnose these events as caused by non Wi-Fi interference.

Table 5.3 summarizes the findings from the previous analysis. The main cause for poor QoE events of both consistent and intermittent poor QoE events is low PHY rate. The majority of poor QoE events have low average PHY rate. For the others, we detect a balance between Wi-Fi and non Wi-Fi interference, with non Wi-Fi interference

Table 5.3: Summary of the diagnosis of poor QoE events

| **Diagnosis** | **Consistent** | **Intermittent** |
|---|---|---|
| Low Avg. Tx PHY rate | 88.5% | 78.3% |
| Wi-Fi interference | 7.2% | 7.3% |
| Non Wi-Fi interference | 4.2% | 12.9% |
| Others | 0.1% | 1.5% |

occurring more frequently on intermittent poor QoE events. A small fraction of poor QoE events presented both Avg. Tx PHY rate > 15 Mbps and BUSY < 60%. We name the root cause simply as *others* and the poor QoE events are caused by a combination of low Avg. Tx PHY rate, Wi-Fi interference and non Wi-Fi interference.

## 5.7 Summary

We apply our predictors on Wi-Fi metrics collected in the wild from 832 APs over a period of 7 days. We first study link-layer Wi-Fi quality and network QoS metrics, and find that medium occupation is above 50% on 7.4% of samples, while Avg. Tx PHY rate is below 6.5Mbps on 11.5% and 7.3% of samples from .11n 1x1 and .11n 2x2 stations. We find that only 31% of download samples have datarate above 100 Kbps, but when they do Wi-Fi link utilization is above 70% on 2.6%, 2.1%, and 1.7% of .11n 2x2, .11n 1x1, and .11g stations samples. In these cases, the Wi-Fi link is likely bottlenecking end-to-end bandwidth. We that Wi-Fi quality is mostly good across applications — in more than 66% of samples Wi-Fi quality does not degrade QoE of any application. We use clustering to aggregate results across applications and find two clusters where Wi-Fi quality does not significantly degrade QoE across applications. We classify samples from the remaining cluster as poor QoE samples. We find 8.94% poor QoE samples in the deployment, with 60% of stations with a fraction of poor QoE samples below 5%. We study poor QoE samples over time to define short, consistent, and intermittent poor QoE events. We find that stations with small fraction of poor QoE samples have more short-lived poor QoE eventsm while stations with higher fraction of poor QoE samples have more consistent poor QoE events. Short-lived poor QoE events are difficult to detect and diagnose, since it is unclear if the root cause was fixed or temporarily absent and require a long-term monitoring approach. We diagnose consistent and intermittent poor QoE events and find that the large majority is caused by low average PHY rate, but consistent poor QoE events more often have both low average PHY rate and high medium occupation. We also find that, when average PHY rate is > 15 Mbps, non Wi-Fi interference is more often the cause for intermittent poor QoE events and Wi-Fi interference is more often the cause for consistent poor QoE events.

# Chapter 6

# Conclusion and future work

This thesis designs and evaluates techniques to passively monitor Wi-Fi quality on commodity access points and predict when Wi-Fi quality degrades internet application QoE. We make the following contributions.

1. **Design and evaluation of a method for passive Wi-Fi link capacity estimation.**

   We proposed an algorithm to estimate the link capacity based on passive metrics sampled from commodity APs. Our results show that it is possible to estimate the link capacity per PHY rate based on a limited set of parameters related to the particular AP instance. Link capacity is useful both for estimating Wi-Fi link available bandwidth as well as diagnosing Wi-Fi performance impairments. We show two case studies where we use link capacity to detect the presence of frame delivery and medium access problems. This method allows ISPs to continually monitor users' Wi-Fi link capacity and available bandwidth. The ISP would then be able to quantify how often the Wi-Fi link bottlenecks internet connectivity. Also, the diagnosis approach proposed here can be the first step to guide helpdesk operators when diagnosing users' Wi-Fi problems.

2. **Design and evaluation of predictors for the effect of Wi-Fi quality on QoE for web browsing, YouTube, audio and video RTC.**

   We studied the effects of Wi-Fi quality on QoE of four applications: web browsing, YouTube streaming, audio and video RTC. One of the main challenges in this research is that we want to cover a wide range of Wi-Fi scenarios while also obtaining the user perception of quality. In this thesis, we present a methodology to build the training set despite this challenge. We use application QoS to QoE

models to obtain MOS estimations and validate each model with a small user study. We show that we can build predictors using support vector regression, and we analyze the impact of individual features on the prediction task. These QoE predictors estimate how Wi-Fi quality affects each application and can be used for real time QoE monitoring. ISPs can manage customers' QoE by using QoE predictions to decide which home should receive more resources (e.g., helpdesk operators' attention) to improve overall customer satisfaction.

3. **Design of a method to identify poor QoE events.**

   We leverage the QoE predictors for web, YouTube, and audio and video RTC to identify instances where Wi-Fi metrics indicates that user experience should be poor across applications. This definition is useful in cases where we do not know the user application but we want to know whether Wi-Fi quality can degrade QoE. We study how poor QoE samples occur over time and define short, intermittent, and consistent poor QoE events. We found poor QoE events stretching over many hours. We found that stations differ in the type of poor QoE events they have: stations which rarely show poor QoE samples have more short poor QoE events, while stations with a large fraction of poor QoE samples have more consistent poor QoE events. We also show how we can diagnose poor QoE events using poor QoE events' Wi-Fi metrics. We can describe stations' quality by number, duration, and type of poor QoE events. Short-lived poor QoE events typically require a long-term monitoring approach for detection and diagnose, while consistent and intermittent poor QoE events may have different impact on user experience.

4. **Characterization of Wi-Fi quality in the wild.**

   We analyze data from 6086 stations connected to 832 APs of customers of a large Asian-Pacific residential ISP, to characterize Wi-Fi quality in the wild. We found that, more often, stations have low PHY rate than high medium occupation. We also found that non Wi-Fi interference was the main cause for high levels of medium occupation. We estimate that Wi-Fi quality limits access link bandwidth in up to 22% of stations' samples. The home Wi-Fi link already limits end-to-end internet performance in many scenarios, and this is likely to become more common as faster broadband internet access becomes accessible. We use the QoE predictors to estimate when Wi-Fi quality degrades application QoE. We find that home Wi-Fi quality, in most cases, does not degrade application QoE,

which is reassuring. We find that the majority of stations have a small fraction of poor QoE events. Some stations, however, very frequently show poor QoE events and are clear cases where the ISP could intervene to improve user experience. This characterization should provide insight on the occurrence and characteristics of poor QoE events in the wild.

This thesis is the first to establish relationships between link-layer Wi-Fi metrics from commodity APs and application QoE. Further research is required to improve quality of predictions and to integrate predictors into an online QoE monitoring system.

**Explore other features for the prediction task.** We create features using simple aggregation techniques (notably, averages) to coincide with the current implementation of the Wi-Fi monitoring system, but other options exist. It would be interesting to study how other aggregation metrics, such as median, percentiles, and standard deviations could improve predictors' accuracy. Also, future home APs should have processing power to measure per-packet processing metrics. Therefore, it would be interesting to study the accuracy of predictors which use measurements of local and external latency, for example.

**Study of other applications.** We are interested into studying the effects of Wi-Fi quality on other applications. It would be interesting to quantify the effects of Wi-Fi quality on online gaming, for example. Internet gaming is a popular form of home entertainment where network impairments may highly degrade user experience. This research is challenging: the experimental methodology used here is unlikely to apply as there is no QoS/QoE model that closely resemble user opinion on online gaming.

**Integration with Wi-Fi monitoring system.** We plan to integrate the QoE predictors and the clustering method into the Wi-Fi monitoring system. Efficiency is a very relevant factor. The Wi-Fi monitoring system currently collects measurements from over 5 thousands APs and over 30 thousands stations, and it is under expansion. We plan to cache QoE predictions using hash tables and use linear interpolations to efficiently approximate SVR predictors. This is to avoid iterating over all support vectors for every QoE prediction.

# Bibliography


[1] B. Agarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker. Netprints: Diagnosing home network misconfigurations using shared knowledge. In *Proceedings of the USENIX conference on Networked Systems Design and Implementation*, 2009.

[2] M. Alreshoodi and J. Woods. Survey on qoe\ qos correlation models for multimedia services. *arXiv preprint arXiv:1306.0221*, 2013.

[3] D. Ammar, K. De Moor, M. Xie, M. Fiedler, and P. Heegaard. Video qoe killer and performance statistics in webrtc-based video communication. In *Communications and Electronics (ICCE), 2016 IEEE Sixth International Conference on*, pages 429–436. IEEE, 2016.

[4] ARCEP: Autorité de Régulation des communications électroniques et des postes. Rapport public d'activité de l'arcep, 2015.

[5] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. A quest for an internet video quality-of-experience metric. In *Proceedings of the ACM Workshop on Hot Topics in Networks*, 2012.

[6] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. Developing a predictive model of quality of experience for internet video. In *ACM SIGCOMM CCR*, 2013.

[7] D. Basak, S. Pal, and D. C. Patranabis. Support vector regression. *Neural Information Processing-Letters and Reviews*, 2007.

[8] J. G. Beerends, C. Schmidmer, J. Berger, M. Obermann, R. Ullmann, J. Pomy, and M. Keyhl. Perceptual objective listening quality assessment (polqa), the third generation itu-t standard for end-to-end speech quality measurement part

i—temporal alignment. *Journal of the Audio Engineering Society*, 61(6):366–384, 2013.

[9] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006.

[10] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547, 2000.

[11] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-e, A. Bhartia, and D. Aguayo. Large-scale measurements of wireless network behavior. In *Proc. ACM SIGCOMM*, 2015.

[12] E. Bocchi, L. De Cicco, and D. Rossi. Measuring the quality of experience of web users. In *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*. ACM, 2016.

[13] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: measurements, metrics, and implications. In *Proc. IMC*, 2011.

[14] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: Measurements, metrics, and implications. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference*, 2011.

[15] CAIDA: Center for Applied Internet Data Analysis. Caida performance measurement tools taxonomy. http://www.caida.org/tools/taxonomy/performance.xml.

[16] L. Carvalho, E. Mota, R. Aguiar, A. F. Lima, J. N. de Souza, and A. Barreto. An e-model implementation for speech quality evaluation in voip systems. In *Proceedings of the 10th IEEE Symposium on Computers and Communications*, volume 154. IEEE Computer Society, 2005.

[17] P. Casas, M. Seufert, and R. Schatz. Youqmon: A system for on-line monitoring of youtube qoe in operational 3g networks. *ACM SIGMETRICS Performance Evaluation Review*, 2013.

[18] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, and R. Schatz. Next to you: Monitoring quality of experience in cellular networks from the end-devices. *IEEE Transactions on Network and Service Management*, 13(2):181–196, 2016.

[19] P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli. On user-centric modular qoe prediction for voip based on machine-learning algorithms. *IEEE Transactions on Mobile Computing*, 15(6):1443–1456, 2016.

[20] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying skype user satisfaction. In *ACM SIGCOMM CCR*, 2006.

[21] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying skype user satisfaction. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2006.

[22] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Commun. ACM*, 49(11), Nov 2006.

[23] K.-T. Chen, C.-C. Tu, and W.-C. Xiao. Oneclick: A framework for measuring network quality of experience. In *Proceedings of IEEE INFOCOM*, 2009.

[24] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *Proc. ACM SIGCOMM*, 2006.

[25] E. Cooke, R. Mortier, A. Donnelly, P. Barham, and R. Isaacs. Reclaiming network-wide visibility using ubiquitous endsystem monitors. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, 2006.

[26] M. Csikszentmihalyi and R. Larson. Validity and reliability of the Experience-Sampling Method. *J Nerv Ment Dis*, 175(9):526–536, Sep 1987.

[27] D. da Hora, K. Van Doorselaer, K. Van Oost, R. Teixeira, and C. Diot. Passive wi-fi link capacity estimation on commodity access points. In *Proc. Traffic Monitoring and Analysis Workshop (TMA)*, 2016.

[28] L. DiCioccio, R. Teixeira, J. Kurose, and M. May. Pinpointing home and access network delays and losses using wifi neighbors. Technical report, UPMC, 2012.

[29] L. DiCioccio, R. Teixeira, and C. Rosenberg. Measuring and characterizing home networks. In *Proceedings of the ACM SIGMETRICS/PERFORMANCE joint*

*international conference on Measurement and Modeling of Computer Systems*, 2012.

[30] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki. Measuring video qoe from encrypted traffic. In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 513–526. ACM, 2016.

[31] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *Proceedings of the ACM SIGCOMM conference on Internet measurement*, 2007.

[32] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.

[33] W. K. Edwards and R. E. Grinter. At home with ubiquitous computing: Seven challenges. In *International Conference on Ubiquitous Computing*, pages 256–272. Springer Berlin Heidelberg, 2001.

[34] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler. Waiting times in quality of experience for web based services. In *QoMEX*. IEEE, 2012.

[35] S. Egger, P. Reichl, T. Hoßfeld, and R. Schatz. "time is bandwidth"? narrowing the gap between subjective time perception and quality of experience. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1325–1330. IEEE, 2012.

[36] M. Fiedler, T. Hossfeld, and P. Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2), 2010.

[37] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao. Youtube everywhere: Impact of device and infrastructure synergies on user experience. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference*, 2011.

[38] M. S. Gast. *802.11 ac: A survival guide*. " O'Reilly Media, Inc.", 2013.

[39] S. Grover, M. S. Park, S. Sundaresan, S. Burnett, H. Kim, B. Ravi, and N. Feamster. Peeking behind the nat: an empirical study of home networks. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 377–390. ACM, 2013.

[40] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A measurement study of a large-scale p2p iptv system. *Trans. Multi.*, 9(8), Dec 2007.

[41] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11 b. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 836–843. IEEE, 2003.

[42] T. Hobfeld, R. Schatz, M. Varela, and C. Timmerer. Challenges of qoe management for cloud applications. *IEEE Communications Magazine*, 50(4), 2012.

[43] O. Hohlfeld, E. Pujol, F. Ciucu, A. Feldmann, and P. Barford. A qoe perspective on sizing network buffers. In *Proc. IMC*. ACM, 2014.

[44] D. N. d. Hora, R. Teixeira, K. Van Doorselaer, and K. Van Oost. Predicting the effect of home wi-fi quality on web qoe. In *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pages 13–18. ACM, 2016.

[45] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In *Pattern Recognition (ICPR), International Conference on*. IEEE, 2010.

[46] T. Hoßfeld, S. Biedermann, R. Schatz, A. Platzer, S. Egger, and M. Fiedler. The memory effect and its implications on web qoe modeling. In *Proceedings of the 23rd International Teletraffic Congress*. International Teletraffic Congress, 2011.

[47] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz. Quantification of youtube qoe via crowdsourcing. In *Proc. International Symposium on Multimedia (ISM)*. IEEE, 2011.

[48] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner. Assessing effect sizes of influence factors towards a qoe model for http adaptive streaming. In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, pages 111–116. IEEE, 2014.

[49] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *A Practical Guide to Support Vector Classification*. National Taiwan University, April 2010.

[50] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE journal on Selected Areas in Communications*, 21(6):879–894, 2003.

[51] Y. Hu and P. C. Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2008.

[52] IANA. Iana list of well known ports. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, Sept 2013.

[53] E. Ibarrola, F. Liberal, I. Taboada, and R. Ortega. Web qoe evaluation in multi-agent networks: validation of itu-t g. 1030. In *Proc. Int. Conf. on Autonomic and Autonomous Systems*. IEEE, 2009.

[54] C. V. N. Index. Cisco visual networking index: Forecast and methodology, 2015-2020 white paper. *Technical Report, Cisco, Tech. Rep.*, 2016.

[55] C. V. N. Index. Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021. *Technical Report, Cisco, Tech. Rep.*, 2016.

[56] IPERF. A tcp, udp, and sctp network bandwidth measurement tool. http://sourceforge.net/projects/iperf/.

[57] ITU-T. Methods for subjective determination of transmission quality, Aug 1996.

[58] ITU-T. Recommendation P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, 2001.

[59] ITU-T. Recommendation P.563: Single-ended method for objective speech quality assessment in narrow-band telephony applications, 2005.

[60] ITU-T. Recommendation P.862 annex a: Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 and P.862.2, 2005.

[61] ITU-T Recommendation G.1030. Estimating end-to-end performance in ip networks for data applications, 2005.

[62] ITU-T Recommendation P.10/G.100. Vocabulary and effects of transmission parameters on customer opinion of transmission quality, amendment 2, 2006.

[63] M. Jain and C. Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. In *In Proceedings of Passive and Active Measurements (PAM) Workshop*. Citeseer, 2002.

[64] D. Joumblatt, J. Chandrashekar, B. Kveton, N. Taft, and R. Teixeira. Predicting user dissatisfaction with internet application performance at end-hosts. In *Proceedings of IEEE INFOCOM*, 2013.

[65] D. Joumblatt, R. Teixeira, J. Chandrashekar, and N. Taft. Hostview: Annotating end-host performance measurements with user feedback. *ACM SIGMETRICS Performance Evaluation Review*, 38(3):43–48, 2011.

[66] J. Jun, P. Peddabachagari, and M. Sichitiu. Theoretical maximum throughput of ieee 802.11 and its applications. In *Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on*, pages 249–256. IEEE, 2003.

[67] P. Kanuparthy, C. Dovrolis, K. Papagiannaki, S. Seshan, and P. Steenkiste. Can user-level probing detect and diagnose common home-wlan pathologies. *ACM SIGCOMM CCR*, 2012.

[68] M. Katsarakis, G. Fortetsanakis, P. Charonyktakis, A. Kostopoulos, and M. Papadopouli. On user-centric tools for qoe-based recommendation and real-time analysis of large-scale markets. *IEEE Communications Magazine*, 52(9):37–43, 2014.

[69] M. Katsarakis, R. Teixeira, M. Papadopouli, and V. Christophides. Towards a causal analysis of video qoe from network and application qos. In *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*. ACM, 2016.

[70] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, pages 18–28. ACM, 2003.

[71] Kernel.org. About mac80211. http://wireless.wiki.kernel.org/en/developers/documentation/mac80211. Accessed: 03/01/2016.

[72] H. J. Kim and S. G. Choi. A study on a qos/qoe correlation model for qoe evaluation on iptv service. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 2, pages 1377–1382. IEEE, 2010.

[73] S. S. Krishnan and R. K. Sitaraman. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE ACM Transactions on Networking*, 2013.

[74] M. Lacage, M. H. Manshaei, and T. Turletti. Ieee 802.11 rate adaptation: a practical approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 126–134. ACM, 2004.

[75] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 314–321. ACM, 2004.

[76] K. Lakshminarayanan, S. Seshan, and P. Steenkiste. Understanding 802.11 performance in heterogeneous environments. In *Proc. SIGCOMM workshop on Home networks*. ACM, 2011.

[77] M. Li, M. Claypool, and R. Kinicki. Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 374–381. IEEE, 2008.

[78] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the ACM SIGCOMM conference on Internet measurement conference*, 2009.

[79] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. IMC*. ACM, 2009.

[80] P. Meenan. How fast is your website? *Communications of the ACM*, 2013.

[81] V. Menkovski, A. Oredope, A. Liotta, and A. C. Sánchez. Predicting quality of experience in multimedia streaming. In *Proc. International Conference on Advances in Mobile Computing and Multimedia*. ACM, 2009.

[82] M. Mirza, P. Barford, X. Zhu, S. Banerjee, and M. Blodgett. Fingerprinting 802.11 rate adaption algorithms. In *INFOCOM, 2011 Proceedings IEEE*, pages 1161–1169. IEEE, 2011.

[83] R. K. Mok, E. W. Chan, X. Luo, and R. K. Chang. Inferring the qoe of http video streaming from user-viewing activities. In *Proceedings of the ACM SIGCOMM Workshop on Measurements Up the Stack*, 2011.

[84] S. Möller, W.-Y. Chan, N. Côté, T. H. Falk, A. Raake, and M. Wältermann. Speech quality estimation: Models and trends. *IEEE Signal Processing Magazine*, 28(6):18–28, 2011.

[85] A. Narayanan, A. Bergkvist, D. Burnett, and C. Jennings. "WebRTC 1.0: Real-time communication between browsers". *World Wide Web Consortium WD WD-webrtc-20120821*, 2012.

[86] S. Ostermann. Tcptrace. http://www.tcptrace.org.

[87] O. Oyman and S. Singh. Quality of experience for http adaptive streaming services. *Communications Magazine, IEEE*, 2012.

[88] M. Papadopouli, P. Charonyktakis, M. Plakia, and I. Tsamardinos. On user-centric modular qoe prediction for voip based on machine-learning algorithms. *IEEE TMC*, 2015.

[89] A. Patro, S. Govindan, and S. Banerjee. Observing home wireless experience through wifi aps. In *Proc. international conference on Mobile computing & networking*. ACM, 2013.

[90] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. RFC 2330: framework for ip performance metrics. http://www.rfc-editor.org/rfc/rfc2330.txt, 1998.

[91] I. Pefkianakis, H. Lundgren, A. Soule, J. Chandrashekar, P. Le Guyadec, C. Diot, M. May, K. Van Doorselaer, and K. Van Oost. Characterizing home wireless performance: The gateway view. In *Proc. IEEE INFOCOM*, 2015.

[92] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei. Wifi can be the weakestlink of round trip network latency in the wild. In *Proc. IEEE INFOCOM*, 2016.

[93] E. Perahia and R. Stacey. *Next Generation Wireless LANs: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[94] M. H. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on Broadcasting*, 2004.

[95] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee. Diagnosing wireless packet losses in 802.11: Separating collision from weak signal. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[96] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: detecting non-wifi rf devices using commodity wifi hardware. In *Proc. IMC*. ACM, 2011.

[97] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. *ACM SIGCOMM Computer Communication Review*, 36(4):51–62, 2006.

[98] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank. Factors influencing quality of experience. In *Quality of Experience*, pages 55–72. Springer, 2014.

[99] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger. From packets to people: quality of experience as a new measurement challenge. In *Data traffic monitoring and analysis*, pages 219–263. Springer, 2013.

[100] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. A subjective study to evaluate video quality assessment algorithms. In *IS&T/SPIE Electronic Imaging*, pages 75270H–75270H. International Society for Optics and Photonics, 2010.

[101] V. Shrivastava, S. Rayanchu, J. Yoonj, and S. Banerjee. 802.11 n under the microscope. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 105–110. ACM, 2008.

[102] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. W. Biersack. Performance limitations of adsl users: a case study. In *Proceedings of the International Conference on Passive and Active Network Measurement*, 2007.

[103] J. Simpson, CharlesRobert and G. Riley. Neti@home: A distributed approach to collecting end-to-end network performance measurements. In *Passive and Active Network Measurement*, 2004.

[104] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino. Quality of experience estimation for adaptive http/tcp video streaming using h. 264/avc. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 127–131. IEEE, 2012.

[105] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour. Ieee 802.11 n mac frame aggregation mechanisms for next-generation high-throughput wlans. *Wireless Communications, IEEE*, 15(1):40–47, 2008.

[106] L. Skorin-Kapov and M. Varela. A multi-dimensional view of qoe: the arcu model. In *MIPRO, 2012 Proceedings of the 35th International Convention*, pages 662–666. IEEE, 2012.

[107] J. Sommers, P. Barford, N. Duffield, and A. Ron. Accurate and Efficient SLA Compliance Monitoring. In *Proceedings of the ACM SIGCOMM conference*, 2007.

[108] T. Spetebroot, S. Afra, N. Aguilera, D. Saucez, and C. Barakat. From network-level measurements to expected quality of experience: the skype use case. In *M & N workshop*, 2015.

[109] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 39–44. ACM, 2003.

[110] D. Strohmeier, S. Egger, A. Raake, T. Hoßfeld, and R. Schatz. Web browsing. In *Quality of Experience*, pages 329–338. Springer, 2014.

[111] L. Sun and E. C. Ifeachor. Voice quality prediction models and their application in voip networks. *IEEE transactions on multimedia*, 8(4):809–820, 2006.

[112] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: a view from the gateway. In *Proceedings of the ACM SIGCOMM conference*, 2011.

[113] S. Sundaresan, W. De Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Measuring home broadband performance. *Communications of the ACM*, 2012.

[114] S. Sundaresan, N. Feamster, and R. Teixeira. Measuring the Performance of User Traffic in Home Wireless Networks. In *Proc. PAM*, 2015.

[115] S. Sundaresan, N. Feamster, and R. Teixeira. Measuring the performance of user traffic in home wireless networks. In *Passive and Active Measurement*, pages 305–317. Springer International Publishing, 2015.

[116] S. Sundaresan, N. Feamster, R. Teixeira, N. Magharei, et al. Measuring and mitigating web performance bottlenecks in broadband access networks. In *ACM Internet Measurement Conference*, 2013.

[117] I. Syrigos, S. Keranidis, T. Korakis, and C. Dovrolis. Enabling wireless lan troubleshooting. In *Proc. PAM*. Springer, 2015.

[118] I. Tsamardinos, A. Rakhshani, and V. Lagani. Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. *International Journal on Artificial Intelligence Tools*, 2015.

[119] M. Varela, L. Skorin-Kapov, T. Maki, and T. Hosfeld. Qoe in the web: A dance of design and performance. In *QoMEX*, pages 1–7. IEEE, 2015.

[120] M. Varvello, J. Blackburn, D. Naylor, and K. Papagiannaki. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 399–412. ACM, 2016.

[121] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, and T. Hossfeld. Modeling the youtube stack: From packets to quality of experience. *Computer Networks*, 2016.

[122] T. Wang, A. Pervez, and H. Zou. Vqm-based qos/qoe mapping for streaming video. In *Proc. IC-BNMT*. IEEE, 2010.

[123] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[124] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2):121–132, 2004.

[125] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. Ieee, 2003.

[126] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 146–157. ACM, 2006.

[127] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani. Internet protocol television (iptv): the killer application for the next-generation internet. *IEEE Communications Magazine*, 45(11), 2007.

[128] Y. Xie, Z. Li, M. Li, and K. Jamieson. Augmenting wide-band 802.11 transmissions via unequal packet bit protection. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[129] Xiph.org media. http://media.xiph.org/video/derf/.

[130] W. Yin, K. Bialkowski, J. Indulska, and P. Hu. Evaluations of madwifi mac layer rate control mechanisms. In *Quality of Service (IWQoS), 2010 18th International Workshop on*, pages 1–9. IEEE, 2010.

[131] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-area Services. In *Proceedings of the USENIX conference on Operating Systems Design and Implementation*, 2004.

[132] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hoßfeld. Impact of frame rate and resolution on objective qoe metrics. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pages 29–34. IEEE, 2010.

# List of Figures

# List of Tables

**Diego NEVES DA HORA** **27/04/2017**

# Sujet : Prédiction de la QoE de la Wi-Fi domestique avec mesures passives sur les points d'accès de base

**Résumé** : Une mauvaise qualité Wi-Fi peut perturber l'expérience des utilisateurs domestiques sur Internet, ou la qualité de l'expérience (QoE). Détecter quand le Wi-Fi dégrade la QoE des utilisateurs est précieux pour les fournisseurs d'accès internet (FAI), surtout que les utilisateurs domestiques tiennent souvent pour responsable leur FAI lorsque leur QoE se dégrade. Pourtant, les FAI ont peu de visibilité au sein de la maison pour aider les utilisateurs. Cette thèse conçoit et évalue des techniques de surveillance passive de la qualité Wi-Fi sur les points d'accès de base (APs) et prédit quand la qualité du Wi-Fi dégrade la QoE de l'application Internet. Notre première contribution est la conception et l'évaluation d'une méthode qui estime la capacité de liaison Wi-Fi. Nous étendons les modèles précédents, adaptés aux réseaux 802.11a/b/g, pour travailler sur des réseaux 802.11n en utilisant des mesures passives. Notre deuxième contribution est la conception et l'évaluation des prédicteurs de l'effet de la qualité Wi-Fi sur la QoE de quatre applications populaires: la navigation sur le Web, YouTube, la communication audio et vidéo en temps réel. Notre troisième contribution est la conception d'une méthode pour identifier les événements qui traduisent une mauvaise QoE. Nous utilisons un clustering de K-moyennes pour identifier les cas où les prédicteurs QoE estiment que toutes les applications visées fonctionnent mal. Ensuite, nous classons les événements traduisant une mauvaise QoE comme des événements QoE de courte durée, intermittents et cohérents. Enfin, notre quatrième contribution consiste à appliquer nos prédicteurs aux métriques Wi-Fi collectées sur une semaine de surveillance de 832 points d'accès de clients d'un grand fournisseur d'accès Internet résidentiel. Nos résultats montrent que la QoE est bonne sur la grande majorité des échantillons de ce déploiement, mais nous trouvons encore 9% des échantillons avec une mauvaise QoE. Pire, environ 10% des stations ont plus de 25% d'échantillons dont la QoE est médiocre. Dans certains cas, nous estimons que la qualité Wi-Fi provoque une QoE médiocre pendant de nombreuses heures, bien que dans la plupart des cas les événements traduisant une mauvaise QoE soient courts.

**Mots clés** : Wi-Fi, Réseau domestique, QoE, surveillance passive

## Subject : Predicting Home Wi-Fi QoE from Passive Measurements on Commodity Access Points

**Abstract** : Poor Wi-Fi quality can disrupt home users' internet experience, or the Quality of Experience (QoE). Detecting when Wi-Fi degrades QoE is valuable for residential Internet Service Providers (ISPs) as home users often hold the ISP responsible whenever QoE degrades. Yet, ISPs have little visibility within the home to assist users. This thesis designs and evaluates techniques to passively monitor Wi-Fi quality on commodity access points (APs) and predict when Wi-Fi quality degrades internet application QoE. Our first contribution is the design and evaluation of a method that estimates Wi-Fi link capacity. We extend previous models, suited for 802.11a/b/g networks, to work on 802.11n networks using passive measurements. Our second contribution is the design and evaluation of predictors of the effect of Wi-Fi quality on QoE of four popular applications: web browsing, YouTube, audio and video real time communication. Our third contribution is the design of a method to identify poor QoE events. We use K-means clustering to identify instances where the QoE predictors estimate that all studied applications perform poorly. Then, we classify poor QoE events as short, intermittent, and consistent poor QoE events. Finally, our fourth contribution is to apply our predictors to Wi-Fi metrics collected over one week from 832 APs of customers of a large residential ISP. Our results show that QoE is good on the vast majority of samples of the deployment, still we find 9% of poor QoE samples. Worse, approximately 10% of stations have more than 25% poor QoE samples. In some cases, we estimate that Wi-Fi quality causes poor QoE for many hours, though in most cases poor QoE events are short.

**Keywords** : Wi-Fi, home networks, QoE, passive monitoring