Characterizing and Improving WiFi Latency in Large-Scale Operational Networks

Kaixin Sui[†‡], Mengyu Zhou[†], Dapeng Liu[†‡], Minghua Ma[†‡],
Dan Pei[†‡*], Youjian Zhao[†‡], Zimu Li[†], Thomas Moscibroda[§]
[†]Tsinghua University [§]Microsoft Research
[‡]Tsinghua National Laboratory for Information Science and Technology (TNList)

ABSTRACT

WiFi latency is a key factor impacting the user experience of modern mobile applications, but it has not been well studied at large scale. In this paper, we design and deploy WiFiSeer, a framework to measure and characterize WiFi latency at large scale. WiFiSeer comprises a systematic methodology for modeling the complex relationships between WiFi latency and a diverse set of WiFi performance metrics, device characteristics, and environmental factors. WiFiSeer was deployed on Tsinghua campus to conduct a WiFi latency measurement study of unprecedented scale with more than 47,000 unique user devices. We observe that WiFi latency follows a long tail distribution and the 90th (99th) percentile is around 20 ms (250 ms). Furthermore, our measurement results quantitatively confirm some anecdotal perceptions about impacting factors and disapprove others. We deploy three practical solutions for improving WiFi latency in Tsinghua, and the results show significantly improved WiFi latencies. In particular, over 1,000 devices use our AP selection service based on a predictive WiFi latency model for 2.5 months, and 72% of their latencies are reduced by over half after they re-associate to the suggested APs.

Keywords

Wireless Network; WiFi Latency; Measurement

1. INTRODUCTION

WiFi latency is a critical performance measure for modern real-time, interactive Internet applications, such as online gaming, web browsing, and live collaboration applications. These applications have strict requirements on end-to-end latency [28, 33, 53]. For example, [53] shows that 20 ms last-mile latency will be amplified into about two to three seconds of web page load time, so slow that is beyond 47% of web users' expectation [19]. While the latency on the wired Internet is relatively stable and well-engineered (*e.g.*, via CDN), the last-hop WiFi latency is often unpredictable and a dominating factor in the end-to-end latency. For example, our measurement results in §3.1 show that, while the median WiFi latency is as low as 3 ms, its 90th and 99th percentiles are around 20 ms and 250 ms, respectively.

Characterizing WiFi latency using WiFi-related metrics (*e.g.*, channel utilization) is a key first step towards mitigating these problems. Understanding the reasons for high WiFi latency occurrences can lead to better designs of wireless protocols/hardware as well as better diagnosis and improvements in operational WiFi networks. However, measuring and characterizing WiFi latency at large scale is challenging, and to the best of our knowledge, there exists no comprehensive attempt at this problem. The reasons are manifold. First, measuring WiFi latency directly in an enterprise- or campus-scale WiFi network is difficult because i) WiFi latency is not reported in commonly used wireless products; ii) sniffer-based measurement methods are too costly or inconvenient to deploy in large-scale networks; iii) methods based on modifying firmware or software of access points (APs) take significant time to be adopted in commercial deployments; and iv) vanilla measurement methods based on pings are influenced by the device's energy saving modes (§2.1). Secondly, while directly measuring WiFi latency is thus challenging, characterizing WiFi latency by means of reducing it to other more easily measurable WiFi-related metrics is also challenging. Specifically, it turns out that there is no simple correlation between WiFi latency and any other such readily available metric. Instead, as we show in this paper, WiFi latency is related to other radio parameters and metrics in subtle and non-linear ways, revealing a complex interplay between multiple WiFi metrics, protocol parameters and environmental factors.

To tackle these challenges, we design **WiFiSeer** (Fig. 1), a framework for *measuring* and *characterizing* WiFi latency at large-scale, as well as *improving* WiFi latency in large operational Enterprise Wireless Local Area Networks (EWLAN) [9, 10], such as in campuses, shopping malls, airports, and hotels.

WiFiSeer first makes use of a new ping based methodology, called `ping2` (§2.1), to measure the WiFi latency of devices connected to the EWLAN. At the same time, WiFiSeer periodically polls the readily available data using SNMP (Simple Network Management Protocol) to derive potential impacting factors of WiFi latency in the EWLAN. Data from `ping2` and SNMP are associated and form the WiFi latency dataset of WiFiSeer. In a one-week exemplary dataset of real deployment in the EWLAN of Tsinghua University (THU-WLAN), hundreds of millions of WiFi latency records are collected based on `ping2`. Each of these records associates a specific WiFi latency with 11 industry-standard radio factors (*e.g.*, channel utilization), 3 protocol factors (*e.g.*, channel number), and 6 environmental factors (*e.g.*, buildings) from the SNMP data (§2.2). More than 2,700 APs, 114 buildings, and more than 47,000 mobile devices are observed in the dataset.

[*] Dan Pei is the corresponding author.

MobiSys'16, June 25-30, 2016, Singapore, Singapore
© 2016 ACM. ISBN 978-1-4503-4269-8/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2906388.2906393

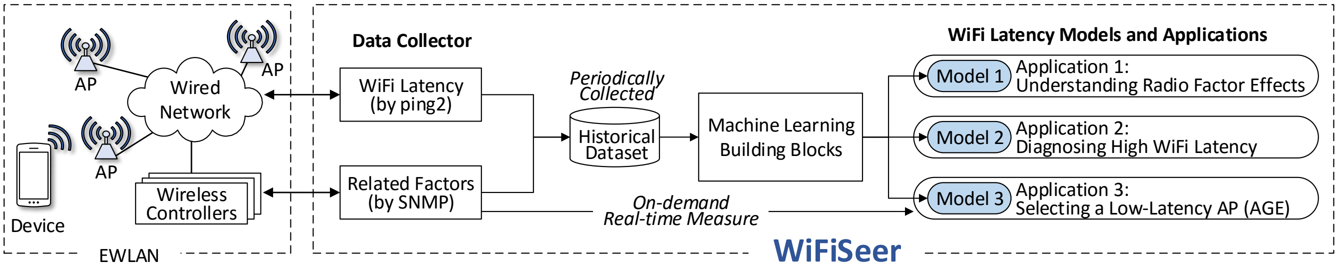

AP
AP
AP
Wired Network
Device
Wireless Controllers
EWLAN
Data Collector
WiFi Latency (by ping2)
Related Factors (by SNMP)
Periodically Collected
Historical Dataset
Machine Learning Building Blocks
On-demand Real-time Measure
WiFi Latency Models and Applications
Model 1
Application 1: Understanding Radio Factor Effects
Model 2
Application 2: Diagnosing High WiFi Latency
Model 3
Application 3: Selecting a Low-Latency AP (AGE)
WiFiSeer

Figure 1: Overview of WiFiSeer.

The above large-scale dataset enables a panoramic view on the WiFi latency in the wild. First, some common perceptions about unsatisfying WiFi latency are quantitatively confirmed: i) WiFi latency follows a long tail distribution: its 50th, 90th and 99th percentiles are around 3 ms, 20 ms and 250 ms, respectively; ii) 75% of user devices experience at least one unsatisfying episode which has the average latency of over 20 ms for more than one minute in a week. Second, straightforward visual inspection also reveals some interesting observations (§3.2): i) the 5 GHz band is still largely underutilized and offers lower latency than 2.4 GHz; ii) the default RSSI-based band selection of devices is biased towards 2.4 GHz; iii) RSSI is not as highly correlated with WiFi latency as channel utilization for example.

In order to understand the reasons behind these observations and mitigate the high WiFi latency in operational networks, we model WiFi latency based on WiFi related factors, which are easily measurable, but interdependent, and related to WiFi latency in non-linear or even non-monotonic ways. WiFiSeer uses machine learning to provide modeling building blocks and then applies them to build three different models tailored for three different practical applications, which in turn enable us to make more interesting observations and insights.

In the first application, WiFiSeer correlates industry-standard radio factors (*i.e.*, easily measurable WiFi performance metrics) to WiFi latency. This model can help vendors and protocol designers to evaluate potential solutions. The model shows that the three most informative indicators for WiFi latency are channel utilization, the number of online devices, and the signal-to-noise-ratio (SNR), while the commonly considered RSSI used for AP selection is not. The model also describes that these metrics can relate to WiFi latency in non-intuitive and intricate ways.

In the second application, WiFiSeer provides a WiFi latency model to help operators diagnose high WiFi latency in operational EWLANs. Specifically, this model identifies where and when WiFi latency is high, and provides potential impact factors for these "bad apples". Based on the results of this application, we accordingly deployed two concrete mitigation approaches (adding a 5 GHz only SSID and dense AP deployment) in THU-WLAN, which turned out to be effective in practice (§5.2).

In the third application, WiFiSeer implements AGE (**A**ssociate to **G**ood **E**nterprise APs), a latency-oriented AP selection service composed of a helper app installed on user's devices and a centralized controller running a predictive model. AGE controller predicts the WiFi latency of APs near a device based on readily-available performance metrics and environmental factors. Then the AGE client app guides the device to associate to the AP with the lowest predicted latency. Deployment of AGE on over 1,000 Android devices for 2.5 months shows that after AGE re-associations, 72% of the latencies are reduced by more than half.

To summarize, our main contributions are:

- WiFiSeer framework is general enough to enable researchers or operators worldwide to study their own networks. It comprises a systematic methodology for modeling the complex relationships between WiFi latency and a diverse set of WiFi performance metrics, device characteristics, and environmental factors.
- We present an enterprise-scale measurement study of WiFi latencies, the largest reported so far, complementing prior WiFi studies that have focused on interference [51] and throughput [46]. Our results quantitatively confirm some anecdotal perceptions, disapprove some others (*e.g.*, the impact of RSSI on latency), and reveal some previously unknown insights (*e.g.*, AP selection mechanisms are inadequate and not collaborative). We believe these results have implications to all players in the mobile Internet ecosystem.
- We deploy three practical solutions for improving WiFi latency in an operational EWLAN. The results show significantly improved WiFi latencies.

The rest of the paper is organized as follows. §2 describes how WiFiSeer collects WiFi latencies and related factors. §3 shows a large-scale measurement study based on the dataset collected from THU-WLAN. §4 introduces how WiFiSeer builds comprehensive WiFi latency models using machine learning techniques. §5 presents three typical applications based on the models in §4 to characterize and improve WiFi latency in large-scale operational networks. §6 discusses several directions for extensions of WiFiSeer. §7 reviews related work and §8 concludes the paper.

2. DATA COLLECTION

In this section, we describe how WiFiSeer collects the dataset of WiFi latencies and related factors. The measurement studies (§3) and model building (§4) are both based on the dataset collected in this way.

2.1 Measuring WiFi Latency

In an operational large-scale EWLAN, systematically measuring WiFi latencies at low cost is challenging. First, WiFi latency is not readily available in common wireless products. For example, the APs of Cisco do not monitor WiFi latency in the SNMP data. Second, sniffer-based methods [31, 32, 47] require deploying extra wireless sniffers, and are thus inconvenient or too costly for a large EWLAN. Third, methods based on modifying the firmware or software of APs [12, 46, 51] cannot be used immediately because it takes a long time for vendors to adopt them in commercial products. Finally, the standard ping methods, issued from a mobile device to an AP (*e.g.*, as used in MobiPerf [39] and Speedtest [17]) or issued from an AP side to a mobile device, can in principle estimate WiFi latency [37], but in practice are unusable (see the evaluation of ping2 later) because they are impacted by the device's *energy saving modes* [6, 7, 14, 16], such as the low power suspend mode of the processor [48] and the power saving mode of the WiFi chip [35]. In particular, if the device is in the energy saving mode, the ping latency will be significantly inflated by the device wake-up time and is thus inaccurate. Because there lacks a feasible WiFi latency measurement tool, researchers and operators rarely know accurate WiFi latencies in an EWLAN.

Design of ping2: To obtain more accurate WiFi latency, we develop ping2, a ping based light-weight WiFi latency measurement tool. The key idea of how ping2 avoids the additional delay introduced by the energy saving mode is intuitive. As shown in Fig. 2, ping2 runs on a server on the wired network in the EWLAN, and sends two consecutive pings from the server to a user's device. The first ping is used to wake up the device. After the first ping is returned, the second ping is immediately issued. Since the device default active mode time (*e.g.*, about 200 ms for Android devices [36] and 100 ms for iPhone [35]) is usually longer than the RTT (*e.g.*, in Fig. 4, 98% of RTT is less than 200 ms, and 97% of that is less than 100 ms), the second ping, in most cases, avoids the device's energy saving mode, and is able to measure WiFi latency more accurately. Thus, the second ping (RTT 2 in Fig. 2) is deemed as the measured result by ping2. Note that the wired part latency (from the ping2 server to the AP) is negligible because the 99th percentile of wired part latency is less than 1 ms in THU-WLAN according to our measurement.

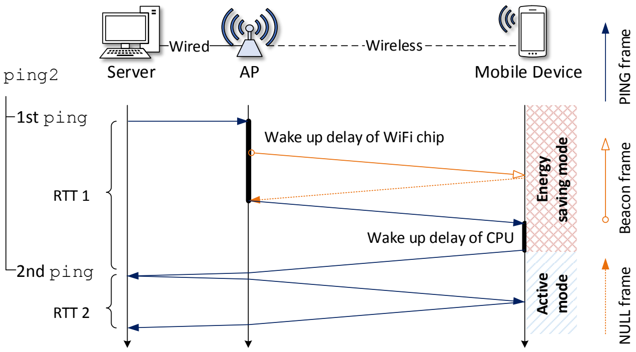


Figure 2: Time line of ping2.

We use ping2 to measure the WiFi latency of users' devices that connect to the EWLAN. In order to obtain a stable latency, we use ping2 to measure the latency of each device every 10 seconds and take the average of six measured results in a minute as one WiFi latency record in the dataset. The IP addresses of the connected devices are recorded by the EWLAN and can be obtained from the SNMP data. To reduce bandwidth cost, ping2 uses ping packets without any payloads.

Evaluation of ping2: We now evaluate ping2 in terms of its accuracy, server overhead, and battery overhead of users' devices. In order to validate the accuracy of ping2, we conduct controlled experiments on three different types of mobile devices in a small testbed. The testbed has an AP, a Linux server connected to the AP in a wired way, and three mobile devices including an Android smartphone (LG Nexus 5), a Windows tablet (Surface Pro 3), and a Linux laptop (Lenovo ThinkPad X1 Carbon). The testbed is set up in an isolated environment where almost no traffic is on the wireless channel we used. We restore the three devices to their factory settings such that the devices will not be waken up by additional installed softwares. Each time, we let only one mobile device connect to the AP via WiFi, and then use the following methods to measure WiFi latency in turn: M_1 uses ping2 to measure latency every 10 s; M_2 pings from the server to the mobile device every 10 s; M_3 pings from the mobile device to the server every 10 s; M_4 pings from the mobile device to the server every 1 ms; M_5 is MobiPerf, an app that is installed only for the smartphone and pings from the smartphone to the server every 0.5 s and uses the average of 10 pings as a latency result. We assume that WiFi latency remains the same during the experiment, and consider the result of M_4 as the ground truth of WiFi latency since the ping of that high frequency prevents devices entering energy saving mode [14, 16]. M_4 is a more general method to obtain the ground truth of WiFi latency than manually turning off the energy saving mode of devices, because the later depends on or may even not be supported by devices. We use each method to measure latency 120 times and show their CDFs in Fig. 3. Among these methods, M_1 (ping2) is the most similar to M_4 for different devices, which indicates that ping2 can measure WiFi latency more accurately. On the other hand, M_3 and M_5 introduce extra latency because if the device is in energy saving mode, it needs several milliseconds to wake up before sending the ping; M_2 is even worse because according to the power saving strategy, waking up a device from the AP side requires waiting for an AP beacon, which can be delayed for as long as a beacon cycle [29], *e.g.*, 100 ms in THU-WLAN.

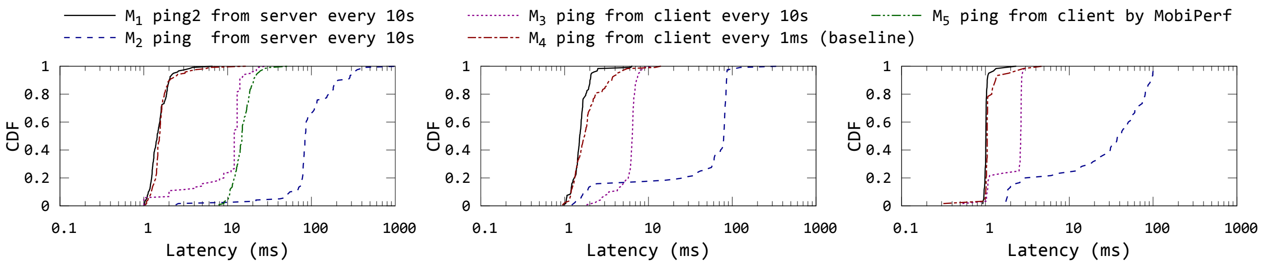
In terms of server overhead, ping2 is extremely light-weight. We implement ping2 using C++ multithreading, and run it on an ordinary Linux server (with two Intel E5-2420 CPUs, 32 GB memory, and a gigabit ethernet card). At peak hour, ping2 is capable of measuring the WiFi latency of more than 15,000 online users' devices using less than 10% CPU utilization and 1MB/s bandwidth (also including the overhead of SNMP data collection).

In addition, ping2 introduces relatively low and occasional overhead on the batteries of users' devices. We conduct A/B tests on 6 different pairs of phones in our small testbed. For each pair, one is measured by ping2 and the other is not. To evaluate the worst case, that is, let ping2 wake up the phones from the energy saving mode as much as possible, we set all the phones to their factory settings and standby mode, so they are in energy saving mode in most cases during the test. We find that ping2 costs at most 7%–10% of the battery volume for different phones in a persistent test of 24 hours (Table 1). In practice, the impact of ping2 could be much less because users' phones themselves are not always in energy saving mode. Specifically, they are woken up periodically by the background traffic of mobile apps [49], such as the transmission of app logs and keep-alive packets. More importantly, ping2 does not have to run all the time. WiFiSeer only needs to periodically collect a dataset, *e.g.*, for a week every quarter, to build WiFi latency models.

In summary, ping2 is an accurate and low-cost tool for measuring WiFi latencies in a large EWLAN. It also is very general and can be plugged into an operational EWLAN without any modifications to the existing APs or users' devices.

Table 1: Maximum battery costs of ping2 in a persistent A/B test of of 24 hours.

ID	Brand	Model	Battery costs of ping2
1	Huawei	Honor 6	7%
2	Xiaomi	Mi 4	8%
3	Vivo	X5 Pro	8%
4	Oppo	R7	9%
5	Samsung	Galaxy Note 3	10%
6	Meizu	MX5	10%



(a) Android smartphone (b) Windows tablet (c) Linux laptop

Figure 3: CDFs of latency measured by different methods. M_5 (MobiPerf) is only for the smartphone.

2.2 Collecting Related Factors

SNMP data is a commonly-used data source for monitoring large-scale EWLANs. Most vendors provide a large range of useful SNMP data on their wireless controllers. To avoid causing overload of the wireless controllers, WiFiSeer polls the SNMP data once per minute. The wireless performance related factors we collect are listed in Table 2. They generally fall into three broad categories: radio factors, protocol factors, and external factors. Table 2 also shows the value range of each factor observed in our dataset. The applications (§4.1) that each factor is used for are also marked.

Radio factors: Radio factors include typical universal wireless performance metrics of APs and user devices, for example channel utilization, interference utilization, receiver/transmitter utilization, and #devices (per radio).[1] See [4] for details of these factors. These radio factors describe the properties of the wireless environment and may fundamentally affect WiFi latency or be correlated to it.

Protocol factors: Protocol factors represent spectrum and frequency properties such as the 5 GHz band. Protocol factors can be described by radio factors. For example, the 5 GHz band has less devices and lower channel utilization. Thus, protocol factors can be deemed as specific collections of radio factors.

External factors: External factors include temporal and spatial factors as well as hardware characteristics of APs and user devices. They often affect WiFi latencies indirectly through radio factors. For example, the rush hour implies a high channel utilization and may cause high WiFi latencies. The five AP models are Cisco AIR-CAP{3602I-C, 702W-C, 3502E-C, 3702I-H, 3502I-C}-K9. The WiFi chip manufacturers are derived from the organizationally unique identifier (OUI) in the device's MAC address. The building type is derived from the usage of the building: classroom buildings, libraries, dormitories, cafeterias, gyms, administrative buildings, departments, hotels, elementary school, and others. Day of week and time of day are derived from the SNMP data timestamp. Some of these external factors, such as the AP models and the buildings, are specific to our THU-WLAN deployment.

Table 2: Wireless performance related factors.

For Application 2, factors of $\checkmark_1$ are used in the first stage of the model, and factors of $\checkmark_2$ are used in the second stage (§5.2).

ID	Radio factor	Value range	Used in app		
			1	2	3
1	Antenna gain	≥ 0 dBm	✓		✓
2	Noise power	≥ -128 dBm	✓		✓
3	Interference power	≥ -128 dBm	✓	$\checkmark_2$	✓
4	Transmit power	≥ -1 dBm	✓	$\checkmark_2$	✓
5	#Devices (per radio)	≥ 0	✓	$\checkmark_2$	✓
6	Channel utilization	0 – 100%	✓	$\checkmark_2$	✓
7	Interference utilization	0 – 100%	✓	$\checkmark_2$	✓
8	Receiver utilization	0 – 100%	✓		✓
9	Transmitter utilization	0 – 100%	✓		✓
10	RSSI	≥ -128 dBm	✓		✓
11	SNR	≥ -128 db	✓		✓

ID	Protocol factor	Value range	1	2	3
12	Channel number	1, 6, 11, 13; 36, 149, 153, 157, 161, 165		$\checkmark_2$	✓
13	Band	2.4 GHz, 5 GHz		$\checkmark_2$	✓
14	Protocol	802.11a, b, g, n, ac		$\checkmark_2$	

ID	External Factor	Value range	1	2	3
15	AP model	5 models			✓
16	WiFi chip manufacture	152 manufactures			✓
17	Building	114 buildings		$\checkmark_1$	✓
18	Building type	10 types		$\checkmark_1$	✓
19	Day of week	Mon, Tue, ..., Sun		$\checkmark_1$	✓
10	Time of day	24 hours		$\checkmark_1$	✓

3. LARGE-SCALE MEASUREMENT STUDY

We have deployed all components of WiFiSeer in THU-WLAN. Tsinghua campus covers an area of about 4 km^2 with more than 53,000 students, faculties, and staff. More than 2,700 dual band Cisco APs governed by 14 Cisco 5508 wireless controllers provide a dual-band SSID "Tsinghua" in 114 buildings for WiFi access. We use WiFiSeer to collect one week of data of WiFi latency and related factors. Our dataset contains hundreds of millions of records (up to 500 GB) from more than 47,000 unique devices. To the best of our knowledge, this dataset thus represents a WiFi latency measurement of unprecedented scale. It provides us a unique opportunity to understand WiFi latency, and its relationship to a diverse set of related factors in the wild. There are several high-level findings that generalize beyond the scope of THU-WLAN: (1) WiFi latency has a long tail; (2) the 5 GHz band is largely underutilized and provides lower latency than 2.4 GHz; (3) AP selection mechanisms are inadequate and not collaborative; (4) Different WiFi chipsets differ greatly in their WiFi latency. We provide insights into these and other results in this section.

3.1 Distributions and Relationships

WiFi latency: Fig. 4 shows the distribution (CDF) of WiFi latency in THU-WLAN. While the median WiFi latency is low (about 3 ms), the 90th percentile is around 20 ms and the 99th percentile is even more than 250 ms. As mentioned in the introduction, this long tail is critical because it can lead to multiple seconds of application-level or user-perceived latency [15]. For example, according to [53], when last-hop latency increases only 10 ms,

[1] Channel utilization is the percentage of time used by all traffic of this channel; interference utilization is the part of channel utilization used by other 802.11 networks (*e.g.*, rogue APs [52]) on the same channel; receiver/transmitter utilization is the percentage of time the AP receiver/transmitter is busy operating on packets; #devices (per radio) is the number of devices connected to the specific band of the AP.

web page load time will increase hundreds of milliseconds; about 20 ms last-hop latency can result in about two to three seconds of page load time for popular websites (*e.g.*, Google and Yahoo), a time that is beyond 47% of users' expectation [19]. Because of this multifold amplification of WiFi latencies to user perceived latencies, the long-tail distribution we observe is alarming.

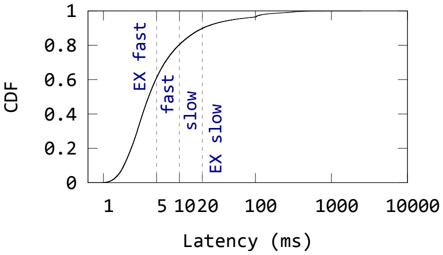


Figure 4: WiFi latency distribution.

For better conceptualization, we categorize WiFi latency into four classes based on the end-to-end latency requirements of Internet services (Table 3). With this classification, we see that in Fig. 4, about 20% of measured WiFi latencies belong to classes slow and EX slow. We also find that on Tsinghua campus, 75% of user devices experience at least one unsatisfying episode of EX slow WiFi latency for more than one minute in a week. The above results demonstrate that high WiFi latency not only exists but is also quite common in practice.

Table 3: Classes of WiFi latency.

WiFi latency class (EX: extremely)	Range	Referred Internet service end-to-end latency requirement
EX fast	< 5 ms	Online gaming [33]
fast	6 – 10 ms	Online collaborative show [28]
slow	11 – 20 ms	Quick Web surfing [53]
EX slow	> 20 ms	-

Related factors: Fig. 6 shows the distribution (left Y-axis) of some factors.[2] The numeric factors are binned by 1 unit (*e.g.*, 1% and 1 dBm). Our dataset includes a wide variety of factors (radio and environmental factors, protocol parameters, *etc.*). The results present a subset of these values that are of interest with regard to WiFi latency. For example, the median of interference power is less than -80 dBm; noise power is under -80 dBm in most cases; the median of #devices (per radio) is below 5; the median of channel, interference, and receiver/transmitter utilization is below 30%, 5%, and 10%, respectively; the median of RSSI and SNR of devices are -66 dBm and 29 db; 802.11n is the dominant protocol.

Relationship to WiFi Latency: To understand the qualitative relationship between WiFi latency and the above related factors, we also plot the average WiFi latency against (binned) values of each factor (right Y-axis of Fig. 6). The trends visually confirm that some factors, such as channel utilization, correlate with WiFi latency well. To quantify these relationships, we consider two common measures: the Kendall rank correlation coefficient (Kendall score for short) and the information gain. These two measures are complementary [41]. While the Kendall score quantifies the monotone relationships (increasing or decreasing), the information gain quantifies how well we can predict WiFi latency if we know a factor. Table 4 shows the results for the radio factors whose values are numeric. The factors are sorted in descending order of the absolute Kendall scores. We see that channel utilization is the most important factor, ranked first for both measures. Interestingly, we notice that these two measures can be conflicting for some factors. For example, RSSI is ranked high for the Kendall score, but its information gain is very low. It means that although RSSI negatively correlates well with WiFi latency, it does not help in predicting it. The reason for this counter-intuitive result is that 96% of RSSI values concentrate on the range between -85 dBm and -45 dBm, where WiFi latency does not change much with RSSI. These results demonstrate the complex relationships between WiFi latency and other diverse factors. We will show how to systematically model WiFi latency in §4.

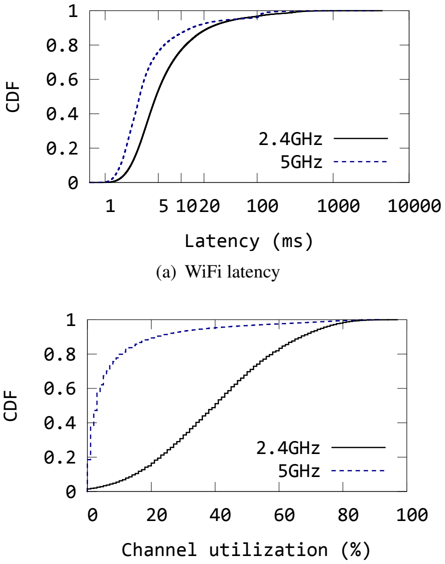
Table 4: Correlation and information gain.

Radio factor	Kendall score	Information gain
Channel utilization	0.886	0.1708
#Devices	0.799	0.0355
Interference power	0.489	0.0629
RSSI	-0.472	0.0008
Noise power	0.230	0.0336
Transmit power	-0.214	0.0047
Antenna gain	0.200	0.0081
Interference utilization	0.132	0.0462
SNR	-0.116	0.0020
Receiver utilization	0.075	0.0294
Transmitter utilization	-0.056	0.0588

3.2 Observations

Our dataset provides ample material to study WiFi latency. We make three important observations, the first one is specific to the measurement environment but also representative, and the other two are due to the protocol implementation.

5 GHz has lower latency: On Tsinghua campus, the average WiFi latency of 5 GHz is 35% lower than that of 2.4 GHz. Furthermore, Fig. 5(a) also shows that the WiFi latency distribution in 5 GHz is better. A potential reason is that only half as many devices use 5 GHz than 2.4 GHz, and thus channel utilization in 5 GHz is much lower (Fig. 5(b)). This observation is consistent with [26]. See §5.2 for how we improve WiFi latency using 5 GHz in THU-WLAN.



(a) WiFi latency

(b) Channel utilization

Figure 5: 5 GHz vs. 2.4 GHz.

[2] Due to the limitation of space, we only show 11 radio factors, 3 protocol factors, and one external factor (time of day) .

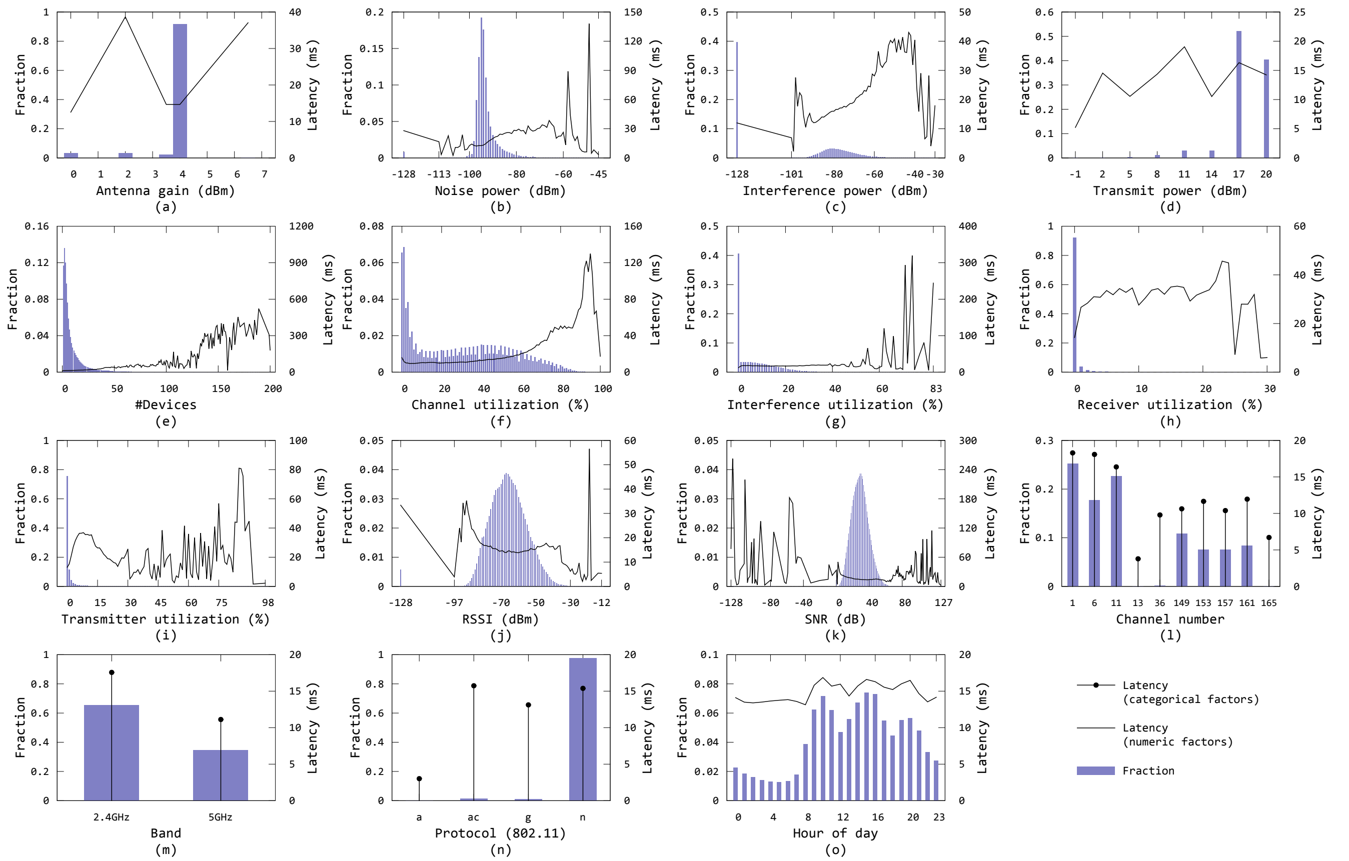


Figure 6: Distribution of factors, and their relationships with WiFi latency. Some peaks of latency are caused by the lack of enough latency data for corresponding factor values.

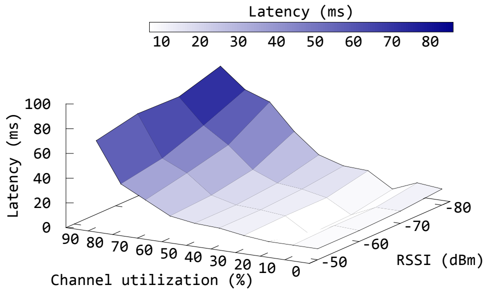


Figure 7: Joint effect of RSSI and channel utilization.

Table 5: Major WiFi chip manufacturers in Tsinghua.

WiFi chip manufacturer	#Devices	Latency (ms)		
		Mean	Median	90^{th}-%ile
Intel	1153	9.63	3.53	15.15
Xiaomi	2626	11.58	3.8	16.56
Apple	21210	12.88	3.31	18.57
HonHai	848	13.75	3.36	18.84
Samsung	3735	14.52	4.02	20.08
Huawei	2536	17.16	4.54	27.21
Meizu	1061	43.83	4.28	131.3

AP selection mechanisms are inadequate and not collaborative: Given the above findings, we find that, surprisingly, dual-band devices (supporting both 2.4 GHz and 5 GHz) are 1.6× more frequent than single-band devices (supporting only 2.4 GHz), and over 50% of the dual-band devices connected to 2.4 GHz. A natural question is that since 5 GHz provides a lower WiFi latency, why are dual-band devices nevertheless connecting to 2.4 GHz? The reason is that the device AP selection heavily depends on RSSI, and since 5 GHz signals attenuate faster, devices are biased towards 2.4 GHz [26]. In contrast, RSSI is not a very important factor for WiFi latency: good RSSI does not guarantee low WiFi latency. For example, Fig. 7 shows that RSSI has a trivial effect on WiFi latency when channel utilization is low. Moreover, when channel utilization exceeds 50%, even high RSSI (*e.g.*, -50dBm) cannot achieve low latency. Interestingly, we also find that AP vendors such as Cisco provide a mechanism (which is also used in THU-WLAN) to direct dual-band devices to the 5 GHz band by delaying probe responses of 2.4 GHz to make 5 GHz more attractive [11]. However, our measurement shows that device vendors do not collaborate well with this ad hoc mechanism, because a user device can still discover the 2.4 GHz of APs by listening to their beacons, leading to crowded 2.4GHz and little-utilized 5GHz bands.

WiFi chipsets are not equal: Table 5 shows the WiFi latency of the major WiFi chip manufacturers that have more than 800 devices in our dataset, sorted by the mean of their WiFi latency. We notice that these manufacturers differ a lot in the tail part (90^{th} percentile). It is interesting to note that in addition to radio factors, device factors also potentially impact WiFi Latency.

4. MODELING METHODOLOGY

The measurements in the previous section show that many factors impact WiFi latency. In this section, we present how WiFiSeer builds more comprehensive WiFi latency models based on machine learning for three representative applications (§6 discusses some other applications.).

4.1 Applications and Challenges

Applications: We begin by describing the model requirements of the three applications in WiFiSeer: (a) Application 1 is to systematically understand how radio factors affect WiFi latency in EWLAN, *i.e.*, which combination(s) of factors impact WiFi latency and how? (b) Application 2 is to help operators diagnose high WiFi latency in the EWLAN by locating where and when WiFi latency is high, and then identifying potential impact factors that operators can take action to improve. (c) Application 3 is to associate user devices to low-latency APs by predicting WiFi latency based on readily-available factors.

All the three applications need effective models to characterize the relationship between WiFi latency and different factors. However, they differ significantly in their requirements with respect to the models (Table 6). First, the model usages are different. The first two applications both aim at providing human-readable insights, thus they need the models to be intuitive enough to interpret rather than black boxes; in contrast, the third application merely needs the model to predict WiFi latency as accurately as possible. Secondly, the applications make use of different related factors. While Application 1 is interested in radio factors, Application 2 cares mostly about environmental factors such as temporal and spatial factors, since they provide specific locations and times for troubleshooting. Besides, because AP selection happens before a device connects to a certain AP, Application 3 cannot use factors that are only available after the connection is established. Finally, the applications focus on different WiFi latency ranges. Application 2 is to diagnose high WiFi latency, Application 3 seeks to find low WiFi latency APs, and Application 1 pays attention to both.

Motivated by these considerations, rather than building a single unified WiFi latency model, we argue that the model should be tailored for the specific application.

Challenges: However, modeling WiFi latency imposes several challenges: First, the relationships between WiFi latency and related factors are complex. As mentioned in §3.1, these relationships are non-linear, or even non-monotonic. As a result, simple approaches such as linear regression that assume a linear or monotonic relationship are unlikely to work. Secondly, various factors are naturally interdependent of each other. For example, Fig. 8 visually shows that SNR and RSSI are highly correlated; Fig. 5(b) shows that channel utilization is quite different in 2.4 GHz and 5 GHz. With such subtle interdependencies, the individual relationship could be a combined effect of several factors. Finally, the distribution of WiFi latencies is highly skewed. For instance, Fig. 4 shows that high WiFi latency is much rarer. Modeling such imbalanced data often has biases (*e.g.*, ignoring high WiFi latency), and thus are less representative [38].

4.2 Machine Learning Algorithms

To address the challenges outlined above, WiFiSeer uses supervised machine learning based methods to model WiFi latency. At a high level, we begin by categorizing WiFi latency into different classes (§3.1), and then build classification models to express the WiFi latency as a function of the related factors[3]. These models can be used to explain or predict WiFi latencies based on the related factors. All models are built offline using the dataset in §3.

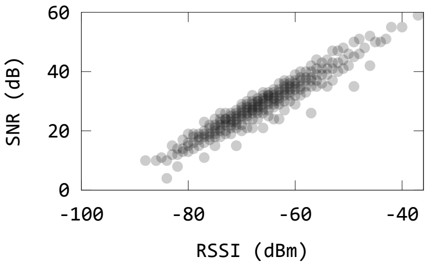


Figure 8: Interdependency between SNR and RSSI.

We need to be careful when choosing machine learning algorithms. The algorithms should be able to not only tackle the complex relationship and the interdependencies, but also satisfy the different requirements of the applications (Table 6). We conducted some pilot experiments on commonly used machine learning algorithms, including k-nearest neighbor (KNN), supporting vector machine (SVM), naive Bayes, neural networks, decision trees, and random forests (see [13] for details). We use these algorithms to do binary classification on our dataset (`slow/EX slow` WiFi latency as one class and `fast/EX fast` WiFi latency as the other class). The dataset is undersampled to deal with the imbalanced class problem [38]. The main idea is to randomly remove a part of majority class data to make classes balanced. We evaluate the classification accuracy using precision-recall curves [38]. Fig. 9 shows that random forests outperform other algorithms by achieving higher recall ($\frac{\text{\# of reported true positives}}{\text{\# of true positives}}$) for the same precision ($\frac{\text{\# of reported true positives}}{\text{\# of reported positives}}$). As a result, we select random forests to build the predictive model for Application 3. In addition, we find that among these learning algorithms, decision trees have a unique advantage of being easy to interpret and also reasonably accurate. Thus, we select decision trees to build the descriptive models for Application 1 and Application 2.

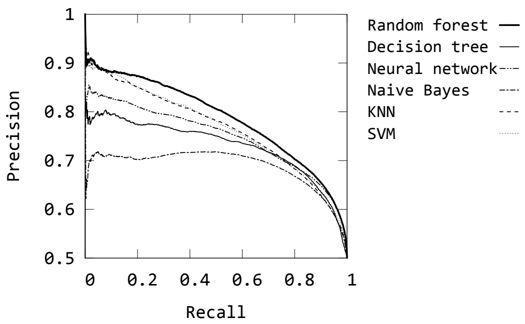


Figure 9: Comparing different machine learning algorithms (Y-axis starts from 0.5).

Table 7 summarizes the model and its configuration for each application. To determine a proper parameter setting for each model, we use 10-fold cross-validation to select the parameter setting that best achieves the goal of the model. We describe these models and how they are used for each application in §5.

[3] We also test regression models (*e.g.*, regression trees) in our pilot experiments, but its accuracy is low. Moreover, classification models are often more straightforward and usable for practitioners [24, 25, 54].

Table 6: Requirements of WiFi latency models for different applications.

	Application 1 Understanding effects of radio factors	Application 2 Diagnosing high WiFi latency	Application 3 Selecting a low-latency AP
Model usage	Intuitive descriptive model	Intuitive descriptive model	Accurate predictive model
Factors interested in (see Table 2)	Industry-standard radio factors	Environment-specific factors and factors actionable for operators	Factors that are available before devices connect to APs.
The WiFi latency interested in	Both high and low latency	High latency	Low latency

Table 7: Configurations and performance of WiFi latency models for different applications.

	Model 1 for Application 1	Model 2 for Application 2	Model 3 for Application 3
Modeling method	Decision tree	Decision trees	A random forest
Classification & Classes (see Table 3)	Multiclass classification: 1. `EX slow` 2. `slow` 3. `fast` 4. `EX fast`	Binary classification: 1. `EX slow` 2. `not EX slow`	Binary classification: 1. `slow/EX slow` 2. `fast/EX fast`
Factors used	Column 4 in Table 2	Column 5 in Table 2	Column 6 in Table 2
How to solve imbalanced data	Undersampling	Adjust the decision threshold	Undersampling
Model goal	Maximizing accuracy	Achieving acceptable precision of class 1 and maximize recall of class 1	Achieving high precision of class 2 and maximize recall of class 2
Performance	Accuracy = 0.42 (multiclass classification)	Precision = 0.59, recall = 0.64 (median)	Precision=0.76, recall=0.67

5. APPLICATIONS

Based on the three models we build in §4, WiFiSeer provides three applications to characterize and improve WiFi latency in large-scale operational networks.

5.1 App 1: Understanding Radio Factors

To understand the effects of the 11 radio factors on WiFi latency, we build a decision tree (Fig. 10) for the four WiFi latency classes. The specific configuration is shown in the second column of Table 7. The tree has an accuracy of 0.42, which is reasonable for multiclass classification [25]. Fig. 10 shows that among all the 11 radio factors, only 4 factors appear in the decision tree. Intuitively, because the decision tree puts important factors near the root to better classify the data, factors close to the root has more effect on WiFi latency than factors close to the leaves, than factors not appearing in the tree. In particular, we find that channel utilization and #devices (in the top three levels of the decision tree) are the two most important radio factors for characterizing WiFi latency. On the other hand, SNR and the transmit utilization only take effect on lower-level branches, *e.g.*, when channel utilization $> 47.5\%$ and #devices ≤ 11.5. Other radio factors that do not show up in the decision tree play a relatively less important role for WiFi latency.

The decision tree also shows the thresholds of how these radio factors affect WiFi latency. These thresholds quantitatively describe conditions in which typical wireless problems are likely to occur. The main takeaways are as follows:

First, for channel utilization, the `EX slow` class only appears when channel utilization $> 47.5\%$, and the `EX fast` class only appears when it $\leq 47.5\%$. Therefore, channel utilization greater than 47.5% can be deemed as the ***heavy load problem***. The decision tree suggests that the heavy load problem is a key cause of `EX slow` WiFi latency, and should be avoided if one wants to achieve `EX fast` WiFi latency. The threshold 47.5% quantitatively confirms previously suggested rules-of-thumb that channel utilization should be kept under 50% for good WiFi performance [5, 21].

Second, we observe that even when channel utilization is very low, a large #devices can still impact WiFi latency. For example, the leftmost `slow` node occurs where channel utilization is ≤ 22.5 and #devices > 35.5. Previous studies suggest that this is caused by the ***local contention problem***: a large number of concurrent senders increase the data collision probability and the backoff waiting time, and decrease the achievable channel utilization [27, 40]. Besides, each device has to wait for its turn to receive the data from the AP. The waiting time becomes long when the AP is connected by many devices.

Third, for SNR, we find that the split points of SNR nodes in the decision tree are from 21.5 dB to 25.5 dB. This suggests that SNR less than 20 dB is low, and could impact WiFi latency. Specifically, the ***fading and noise problem*** increases the bit error rate and thus increases the MAC layer frame retry times; or decreases the PHY rate and thus increases the transmission time, which both in turn inflate the WiFi latency.

The above observations and the tree model characterize how radio factors affect WiFi latency in a typical large-scale EWLAN, and can help further evaluations on the inherent trade-offs in the design of protocols and hardware.

5.2 App 2: Diagnosing High WiFi Latency

For diagnosing high WiFi latency, we consider a binary classification (`EX slow` and `not EX slow`) since the operators (*e.g.*, those we worked with) care more about `EX slow` cases. In the diagnosis, the time and place is important because operators need to know when and where high WiFi latency is likely to occur so that they can take further actions, such as troubleshooting in the field. However, since the decision tree does not guarantee that these factors appear, we use a two-stage method in Application 2.

We first split the data based on the combinations of temporal factors (time of day and day of week) and spatial factors (buildings and building types). In order to provide high-level results, we categorize the time of day into six meaningful intervals: morning (08:00 – 11:00), lunch (11:00 – 13:00), afternoon (13:00 – 17:00), dinner (17:00 – 19:00), evening (19:00 – 22:00), sleep (22:00 – 08:00); day of week into two types: weekdays and weekends. We define a *problem time and place* as any spatial-temporal combination whose 1) data is more than 0.1% of the total and

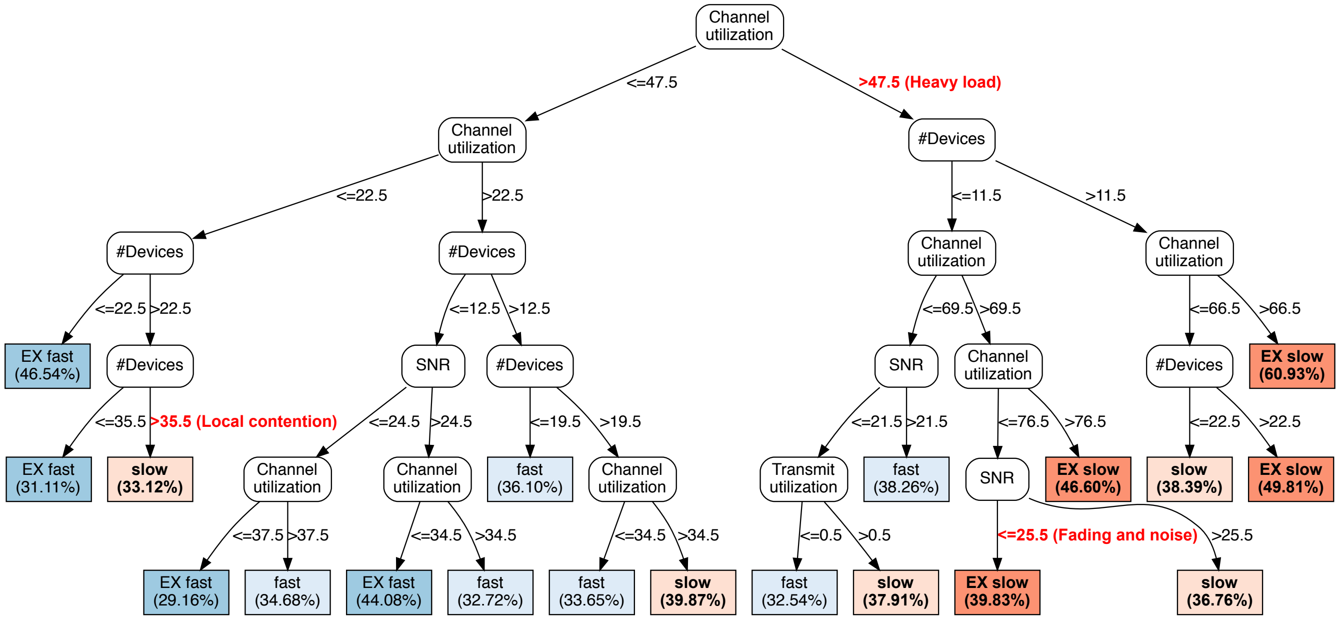


Figure 10: Decision tree for modeling the effects of radio factors on WiFi latency. The percent of the major class in the node is shown.

Table 8: Examples: problem time and places of 6Jiao.

Building-Building Type-Day-Time	%[EX slow]	%Data
6Jiao-Classroom-Weekday-Morning	36%	0.80%
6Jiao-Classroom-Weekday-Afternoon	35%	0.82%
6Jiao-Classroom-Weekday-Evening (Its decision tree is shown in Fig. 11)	51%	0.28%
6Jiao-Classroom-Weekend-Morning	33%	0.13%

2) %[EX slow] (percentage of EX slow) is significantly high ($\geq$1.5× the global %[EX slow] [41]). The global %[EX slow] is about 10% in THU-WLAN (Fig. 4). In this way, we found 92 problem time and places out of 1368 on Tsinghua campus. For example, Table 8 shows the problem time and places regarding a classroom building named 6Jiao (the sixth classroom building, also the largest one in Tsinghua).

Then, for each problem time and place, we build a decision tree (that is, 92 decision trees in total) based on factors that operators can possibly take action on to affect latencies. The specific configuration of the decision trees is shown in the third column of Table 7. Overall, the median precision and recall of those decision trees are 0.59 and 0.64, respectively.

A decision tree identifies a *high WiFi latency condition* as the combination of the factors and their values on the path from the root to an EX slow node. For example, Fig. 11 shows the decision tree of the problem time and place "6Jiao-Classroom-Weekday-Evening". The two high WiFi latency conditions, represented by two bold paths, are "channel utilization $>$ 66.5% and #device $>$ 13.5" and "50.5% $<$ channel utilization $\leq$ 66.5% and #device $>$ 41.5". As such, the operators should focus their efforts on investigating and solving the heavy load and local contention problems in the weekday evening of 6Jiao.

Table 9 shows the prevalent patterns of high WiFi latency conditions, that is, ignoring the specific factor values in the conditions. We see that high channel utilization, a large number of devices, and high interference power are the three dominant factors across the high WiFi latency conditions in THU-WLAN.

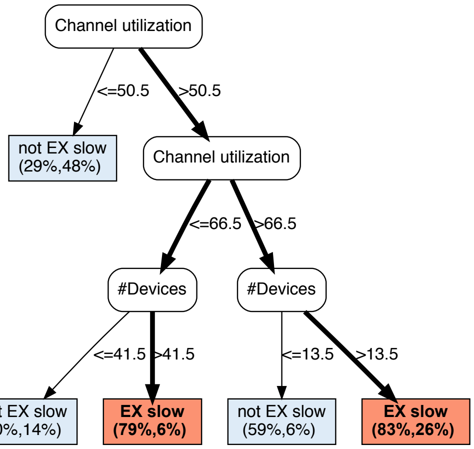


Figure 11: Decision tree for the problem time and place "6Jiao-Classroom-Weekday-Evening". A tuples in a leaf node means ($\frac{\texttt{EX slow}\text{ data in the node}}{\text{data in the node}}$, $\frac{\text{data in the node}}{\text{data in the tree}}$). Bold paths represent two high WiFi latency conditions.

Table 9: Prevalent patterns of high WiFi latency conditions that appear in more than five problem time and places.

High WiFi latency condition pattern	Frequency
Channel utilization $> x$ and #devices $> y$	21
Channel utilization $> x$	18
#Devices $> x$	11
Channel utilization $> x$ and #devices $\leq y$	8
Channel utilization $> x$ and $y <$ #devices $\leq z$	8
#Devices $> x$ and $y <$ channel utilization $\leq z$	6
Interference power $> x$ and #devices $> y$	6
Interference power $> x$ and channel utilization $> y$ and #devices $\leq z$	6

Practical solutions: To solve those common issues, we have deployed two solutions in practice to optimize the WiFi latency in THU-WLAN. First, we intend to guide more dual-band user devices to connect to the 5 GHz to avoid high WiFi latency conditions, because the 5 GHz band has fewer devices and lower channel utilization (Fig. 5(b)), and thus can provide lower latency (Fig. 5(a)). To this end, we add a 5 GHz only SSID "Tsinghua-5G" in addition to the original dual-band SSID "Tsinghua". This gives a chance for the owners of dual-band devices to manually select the 5 GHz band, which in turn solves the aforementioned problem that the default band selection of devices is biased towards 2.4 GHz. The number of user devices connected to "Tsinghua-5G" keeps growing and hit 6,000 at the end of Oct. 2015. We find that, on average, the WiFi latency of "Tsinghua-5G" is 22% lower than that of the original SSID "Tsinghua".

Second, we propose a dense AP deployment solution to avoid high WiFi latency conditions, and deploy this solution in Dorm#31. In particular, for 78 rooms from the 1st floor to the 4th floor in Dorm#31, we deploy *one AP in each room* in which live three students. Such dense deployment significantly reduces the number of devices per radio. Besides, the channel and the power of each AP are also well-engineered to mitigate the interference among APs, and no ethernet ports are available in the rooms so that the interference of rogue APs can be eliminated as much as possible. Compared with other dorms with regular AP deployment on Tsinghua campus — *e.g.*, Dorm#17 where two students live in a room and about *10 rooms share one AP* — the AP in Dorm#31 has fewer devices, less interference, lower channel utilization, and thus lower WiFi latency as shown in Fig. 12(a-d). On average, our dense AP deployment improved WiFi latency by 36%.

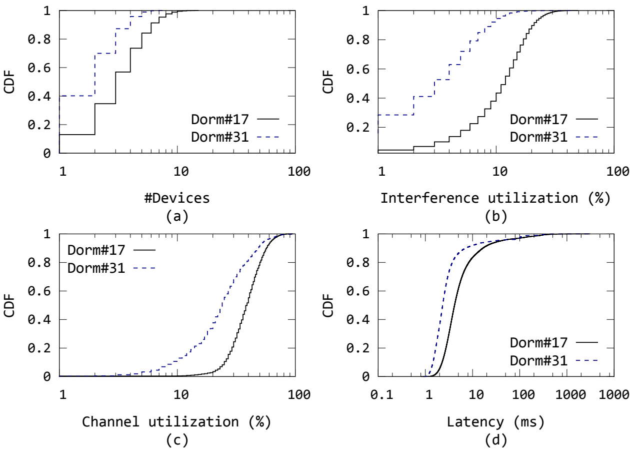
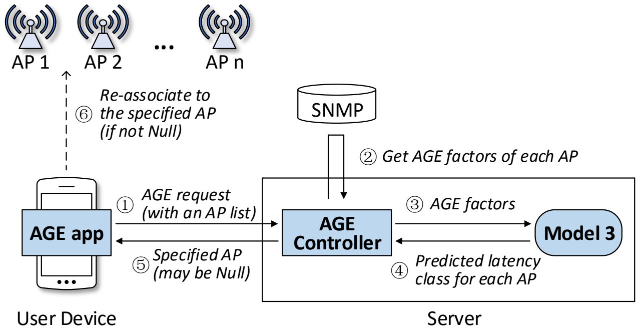


Figure 12: Dense AP deployment (Dorm#31) *vs.* regular AP deployment (Dorm#17).

5.3 App 3: Selecting a Low-Latency AP

An EWLAN often deploy many APs to improve WiFi availability and support device roaming. Therefore, a user device typically hears multiple APs in its vicinity [43], *e.g.*, user devices can hear on average 5 APs in THU-WLAN at the same time. However, deciding which AP to associate to for low latency is a challenging task because neither the device nor `ping2` can measure WiFi latency before the device associates to an AP. Thus, one needs to select a low-latency AP based on other WiFi metrics that are measurable and available before the connection being established, such as RSSI and channel utilization. However, the relationship between WiFi latency and WiFi metrics is complex (Fig. 6), and the default RSSI-based AP selection cannot assure low latency (§3.1). To solve this problem, WiFiSeer provides machine learning based latency-oriented AP selection, called **AGE** (**A**ssociating to **G**ood **E**nterprise APs), to help devices associate to low-latency APs nearby. AGE controls the AP association of user devices by a mobile app we developed, so that AGE can be easily deployed without any changes to device OS/drivers or AP firmware. AGE has already been deployed in THU-WLAN and is used daily by over 1,000 user devices. The deployment shows that AGE reduces WiFi latency substantially (shows later).

Design of AGE: Fig. 13 shows the high-level overview of AGE. It contains two main components: the AGE helper app installed on user devices to associate devices to the AP selected; the AGE controller which uses WiFi latency Model 3 (a random forest) to predict WiFi latencies of APs around a device based on readily-available WiFi factors from the SNMP data. Model 3 is built offline with the historical dataset (last column of Table 7). Its precision and recall are 0.76 and 0.67, respectively. Note that, to avoid battery overhead for devices, AGE does not use `ping2` to measure WiFi latency of devices. WiFiSeer only uses `ping2` for infrequent historical data collection, not for online AGE prediction.



AP 1
AP 2
...
AP n
⑥ Re-associate to the specified AP (if not Null)
SNMP
② Get AGE factors of each AP
AGE app
① AGE request (with an AP list)
⑤ Specified AP (may be Null)
AGE Controller
③ AGE factors
④ Predicted latency class for each AP
Model 3
User Device
Server

Figure 13: AGE overview.

In particular, AGE works as follows:

Step 1: When a device is connected to the EWLAN, the AGE app periodically[4] sends an AGE request with a list of nearby APs it scanned to the AGE controller.

Step 2: For the APs in the list, the AGE controller collects their related factors (last column of Table 2) from the SNMP data, called AGE factors here.

Step 3 & 4: Based on those AGE factors, Model 3 generates the likelihood that an AP is `fast/EX fast` or `slow/EX slow`. Then, the AP is classified as `fast/EX fast` or `slow/EX slow` if the likelihood is higher than 0.6; otherwise, the AP is deemed as `not sure`. We use 0.6 as the threshold because it best achieves the goal of Model 3 (Table 7) in offline 10-fold cross-validation.

Step 5: With the classification results of Model 3, the AGE controller decides whether the device should re-associate to another AP instead of the current one. Specifically, if the current connected AP is predicted as `slow/EX slow`, and there exists at least one nearby AP being predicted as `fast/EX fast`, the AGE controller will respond with the AP that has the highest likelihood of `fast/EX fast`; otherwise, the AGE controller returns `Null`.

Step 6: The AGE app re-associates the device to the specified AP (if not `Null`) by modifying the device WiFi configurations (*e.g.*, using the Android APIs [8]).

[4] For our deployment, the interval is set to 5 minutes when the device screen is on, and 20 minutes when it is off.

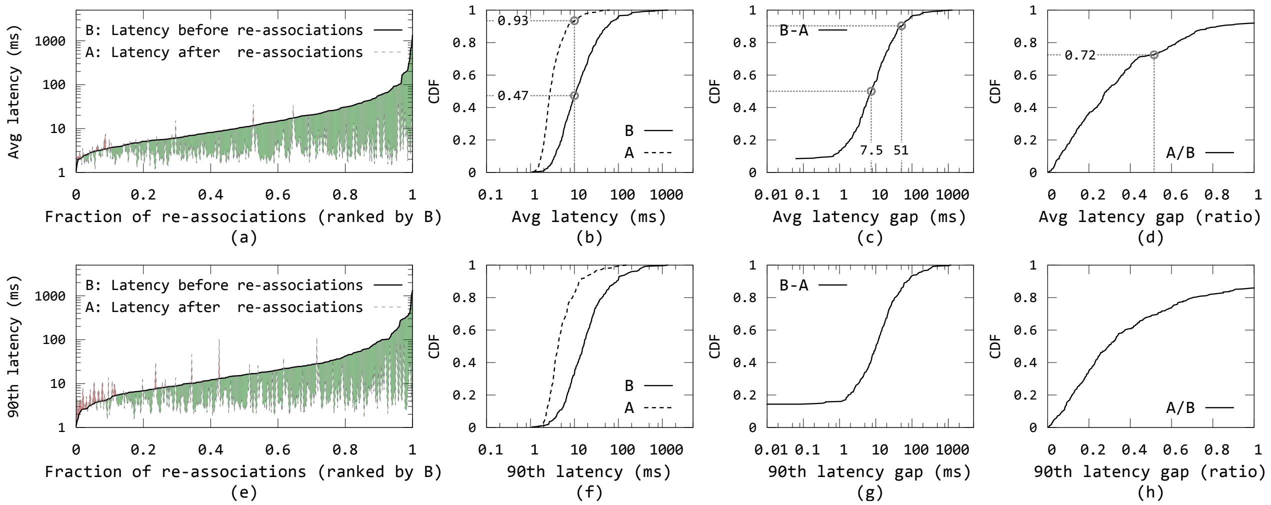


Figure 14: Comparison of WiFi latencies before (B) and after (A) AGE re-associations.

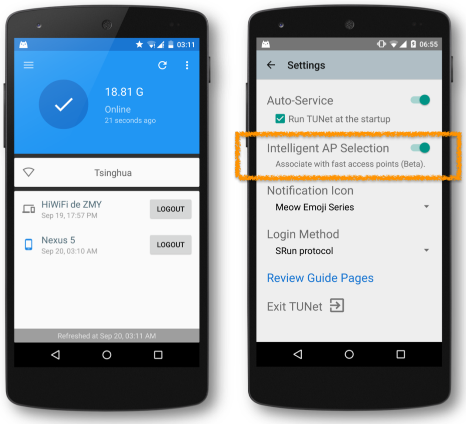
Large-scale deployment of AGE: We implemented the AGE app as a component of the Android version TUNet[5] as shown in Fig. 15. AGE automatically runs in the background, so users do not have to manually run it. 1165 devices have installed AGE since Sept. 2015, and have used it for 2.5 months.

To evaluate AGE, we measure the WiFi latency of these devices using `ping2` (This measurement is only for evaluation, it is not a part of the design of AGE). Fig. 14 shows the performance of AGE re-associations, that is, one-minute latency before and after devices re-associate to the APs suggested by AGE. For these re-associations, AGE significantly reduces both the average and the 90th percentile of WiFi latency. For example, focusing on the average latency in Fig. 14(a-d), 92% of the re-associations reduce the latency (green areas below the solid line in Fig. 14(a)); the fraction of `fast/EX fast` ($\leq$ 10 ms) has been improved from 47% before re-associations to 93% after re-associations (Fig. 14(b)); after re-associations, 50% of latencies are reduced by more than 7.5 ms and 10% of latencies are reduced by more than 51 ms (Fig. 14(c)), and relatively, 72% of the latencies are reduced by more than half (Fig. 14(d)). The results of the 90th percentile (Fig. 14(e-h)) are similar.

Our large-scale real-world deployment demonstrates that AGE is much more effective at selecting low-latency APs than the default AP selection of devices. Thus, AGE provides a promising framework for EWLANs to improve their WiFi latency.

6. DISCUSSION

Generality of WiFiSeer: WiFiSeer can be easily adopted by network operators in typical operational EWLANs. For the measurement part, WiFiSeer only uses traditional SNMP data and the light-weight `ping2` method, both of which are general methods for EWLANs. Furthermore, for the running of WiFiSeer, continuous running of data collection is not required by WiFiSeer. For modeling and the three applications, WiFiSeer takes advantage of widely used machine learning algorithms, which are well-known for being general and able to deal with diverse data. The three applications are independent of each other and thus can be run separately on demand.



HiWiFi de ZMY
Sep 19, 17:57 PM
LOGOUT

Nexus 5
Sep 20, 03:10 AM
LOGOUT

Auto-Service ●
☑ Run TUNet at the startup

Intelligent AP Selection ●
Associate with fast access points (Beta).

Notification Icon
Meow Emoji Series

Login Method
SRun protocol

Figure 15: TUNet with the AGE app as a component.

Other WiFi latency related applications: The machine learning approaches in WiFiSeer can be used in other applications. For example in EWLAN optimizations, due to the lack of knowledge on how different actions will influence performance, operators face hard decisions on whether certain actions will work or not. In other words, it is difficult to estimate beforehand their impact on network performance. WiFiSeer can provide a predictive model to help operators estimate the performance improvement of different actions. Based on the operational experience, operators can roughly estimate the impact of an action on the radio factors. For example, adding an AP for client-concentrated areas reduces the number of devices of nearby APs by 20%. By using our model, the potential benefit on latencies can be predicted before taking any real-world actions. Thus, our modeling approach can also help operators make decisions to balance cost/benefit trade-offs.

Potential improvements of AGE: In the AGE deployment, we notice that some devices do not collaborate well with the AGE app. In particular, although AGE suggests re-associations, some devices may not authorize the AGE app to do this, or some devices will change back to the original APs after AGE re-associations because of the conflicts between the default device AP selection algorithm and AGE. It shows that AGE has the potential to bring more benefits if it can be implemented on device hardware/OS levels or in protocols as we discuss next.

[5] TUNet is a mobile app which can help users automatically login the captive portal of THU-WLAN. The version with AGE can be downloaded from [18].

Future 802.11k/v integration: The real-world deployment (§5.3) of AGE shows that EWLAN users could have a much better latency experience. The strategy of AGE can be an important complement to the traditional 802.11k/v protocol, which is designed to assist AP selection when devices roam. The current implementation of 802.11k/v only provides a way to exchange several data (*e.g.*, channel loads, neighbor reports, and link measurements) between APs and devices, but leaves the AP selection to the user devices. This task, however, is challenging as elaborated in §4.1. AGE can be integrated into 802.11k/v to systematically solve those problems, and make 802.11k/v more intelligent in the future. Still, for legacy EWLANs and user devices, where 802.11k/v is not supported, our application-level attempt is a viable way for WiFi latency optimization.

Other WiFi performance metrics beyond latency: Different mobile applications are often sensitive to different WiFi performance metrics, *e.g.*, latency we studied in this work, throughput, or both. Also, it is possible that the good conditions for latency could conflict with that for throughput. Therefore, a promising extension of AGE is to build predictive models not only for latency but also for other metrics, such as throughput or even application layer metrics (*e.g.*, web page load time). Then, according to user's preference, AGE can select the AP that has either reasonable performance for all those metrics or the best performance for the active application, *e.g.*, high throughput for online videos. We will explore this direction in our future work.

7. RELATED WORK

WiFi measurements: Analyzing packet traces is a costly way to monitor a wireless network. It is hard to modify the hardware of enterprise APs to capture and record wireless frame traces like WiSe [46] and PIE [51]. Additional hardware, *e.g.* extra sniffers in Jigsaw [32] and Wit [44], are also inappropriate for large scale measurements. Meraki [26] monitors the wireless link at large scale using the customized AP with dedicated radio hardware (*e.g.*, Meraki MR18). In contrast, within WiFiSeer framework, WiFi latency measurement method of `ping2` and traditional SNMP are low-overhead and deployable in large-scale EWLANs.

Modeling: There are some prior studies of modeling key performance indicators in different domains, such as video QoE [25, 41, 50] and web QoE [24]. They also take advantage of machine learning algorithms, such as decision trees, to model the complex relationships. However, machine learning building blocks cannot be set once and for all. In this paper, we explore how to tailor models for three different applications (§4).

AP selection: There are many methods for AP selection. Currently most devices only utilize part of the available information to make choices with simple assumptions of wireless performance. Preferential association to APs with stronger RSSI is a common approach [55], but stronger signal strength does not assure better performance [42]. Some implementations utilize more historical or actively measured client-side information besides RSSI [3, 45]. However, it is time-consuming, battery-draining and sometimes bothering for a user device to test each nearby AP only by itself [1].

Some approaches exchange information of the wireless environment between the EWLAN and its clients — specifically, implementations of 802.11k/v [22, 23] standards. *E.g.*, Cisco enterprise APs can send neighbor AP report (defined in 802.11k, called "AP assisted roaming" by Cisco [2]) to 802.11k equipped iOS devices [20], which helps speed up scanning, reduces power consumption, and makes efficient use of air time. However, the information currently exchanged neither contains direct measurement of WiFi latency nor provides enough knowledge for 802.11k/v-capable devices to actually select best performing APs.

In addition, some work [30, 34, 43] attempts to associate a user device to multiple APs *simultaneously*. It requires the device to be equipped with more than one physical WiFi interface card or support virtual interfaces. Nevertheless, these solutions still need a way to decide which APs nearby have better performance. WiFiSeer provides a methodology for modeling and predicting WiFi performance, which is complementary to those solutions.

8. CONCLUSION

In this paper, we present WiFiSeer, a general and practical framework for measuring and characterizing WiFi latency in large-scale operational EWLANs. WiFiSeer was deployed in Tsinghua to conduct a WiFi latency measurement study of unprecedented scale. Our machine learning-based characterization results provide important insights to network operators and 802.11 protocol designers. Among commonly suspected reasons for unsatisfying WiFi latency, heavy traffic load, local contention, noise and fading, and interference are indeed quantitatively confirmed as top reasons, whereas RSSI (often used for AP selection) is not. Our machine learning based modeling reveals the complex but quantitative impacts of various factors on real-world WiFi latency, which are hard to obtain through theoretical analysis or simulations. The real-world performance gains resulting from two deployed mitigation approaches suggested by WiFiSeer highlight the promise of data-driven re-engineering of operational networks.

Our results also show that existing AP selection mechanisms are inadequate. Ad hoc solutions from AP and OS vendors do not collaborate with each other, resulting in poor utilization of available resources, *e.g.*, crowded 2.4 GHz and little-utilized 5 GHz bands. Through adaptively re-associating a device to better performing APs, our widely deployed AGE client greatly helps to mitigate these problems on Tsinghua campus. Encouraged by this result, we advocate that protocol designers, OS vendors, AP vendors, and researchers should work together to push 802.11k/v deployment, and to improve the methodologies, such as WiFiSeer, to generate better data-driven intelligence to drive the 802.11k/v protocols.

Acknowledgments

We thank our shepherd Kamin Whitehouse and the anonymous reviewers for their thorough comments and valuable feedback. We are grateful to Lab μ for their efforts on developing TUNet and for them allowing us to implement AGE as a component of it. We appreciate Weichao Li, Fan Ye, Hewu Li, and Chunyi Peng for their helpful suggestions, as well as Yousef Azzabi for his proofreading.

This work was partly supported by the Key Program of the National Natural Science Foundation of China under grant 61233007, the National Natural Science Foundation of China (NSFC) under grant 61472214 & 61472210 & 61033001 & 61361136003, the National High Technology Research and Development Program of China (863 Program) under grant 2013AA013302, the National Basic Research Program of China (973 Program) under grant 2013CB329105 & 2011CBA00300 & 2011CBA00301, the Tsinghua National Laboratory for Information Science and Technology key projects, the Global Talent Recruitment (Youth) Program, and the Cross-disciplinary Collaborative Teams Program for Science & Technology & Innovation of Chinese Academy of Sciences-Network and system technologies for security monitoring and information interaction in smart grid.

References

[1] 802.11 association process explained. https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_Association_process_explained.

[2] 802.11r, 802.11k, and 802.11w deployment guide, cisco ios-xe release 3.3: 802.11k assisted roaming. http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/5700/software/release/ios_xe_33/11rkw_DeploymentGuide/b_802point11rkw_deployment_guide_cisco_ios_xe_release33/b_802point11rkw_deployment_guide_cisco_ios_xe_release33_chapter_010.html.

[3] Android framework classes and services. https://android.googlesource.com/platform/frameworks/base/. Accessed: 2015-01-23.

[4] Cisco snmp. http://tools.cisco.com/Support/SNMP/do/BrowseOID.do.

[5] Designing for voice / tech tools. http://wirelesslanprofessionals.com/wlw-010-designing-for-voice-tech-tools/.

[6] Edimax wifi adapter seems to go to sleep if not used. https://www.raspberrypi.org/forums/viewtopic.php?t=61665.

[7] Edison wifi delay and power saving. https://communities.intel.com/message/255890.

[8] enablenetwork of wifimanager in android. http://developer.android.com/reference/android/net/wifi/WifiManager.html#enableNetwork(int,boolean). Accessed: 2015-09-22.

[9] Ewlan market. http://www.delloro.com/news/enterprise-wireless-lan-market-to-expand-70-percent-by-2018.

[10] Ewlan overview. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/emob41dg/emob41dg-wrapper/ch1_Over.html.

[11] Load balancing and band select on the cisco wireless lan controller. https://supportforums.cisco.com/document/81896/load-balancing-and-band-select-cisco-wireless-lan-controller.

[12] Openwrt. https://openwrt.org/.

[13] Orange. http://docs.orange.biolab.si/reference/rst/index.html.

[14] Ping on mavericks on late 2013 macbook pro slow and variable compared to windows. http://apple.stackexchange.com/questions/110777/ping-on-mavericks-on-late-2013-macbook-pro-slow-and-variable-compared-to-windows.

[15] Round-trip time is important. https://marketo-web.thousandeyes.com/rs/thousandeyes/images/ThousandEyes_Use_Case_Enterprise_and_IT_Operations.pdf.

[16] Slow ping on raspberry pi. http://procrastinative.ninja/2014/11/19/slow-ping-on-raspberry-pi/.

[17] Speed test. http://www.speedtest.net/.

[18] Tunet. http://netman.cs.tsinghua.edu.cn/wp-content/uploads/2016/04/TUNet-Android-v3.2.4.apk_.zip.

[19] What is the business value of application speed and performance to you? https://s3.amazonaws.com/cisco-partnerpedia-production/product_files/78904/original/Speed_Matters.pdf.

[20] Wi-fi network roaming with 802.11k, 802.11r, and 802.11v on ios. https://support.apple.com/en-us/HT202628.

[21] Wi-fi tip: Channel utilization best practice. http://www.revolutionwifi.net/revolutionwifi/2010/08/wi-fi-tip-channel-utilization-best.html.

[22] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac)and physical layer (phy) specifications amendment 1: Radio resource measurement of wireless lans. *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)* (June 2008), 1–244.

[23] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 8: Ieee 802.11 wireless network management. *IEEE Std 802.11v-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11w-2009, IEEE Std 802.11n-2009, IEEE Std 802.11p-2010, and IEEE Std 802.11z-2010)* (Feb 2011), 1–433.

[24] Balachandran, A., Aggarwal, V., Halepovic, E., Pang, J., Seshan, S., Venkataraman, S., and Yan, H. Modeling web quality-of-experience on cellular networks. In *MoBicom 2014*.

[25] Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., and Zhang, H. Developing a predictive model of quality of experience for internet video. In *ACM SIGCOMM Computer Communication Review* (2013), vol. 43, ACM, pp. 339–350.

[26] Biswas, S., Bicket, J., Wong, E., Musaloiu-E, R., Bhartia, A., and Aguayo, D. Large-scale measurements of wireless network behavior. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (2015), ACM, pp. 153–165.

[27] Bononi, L., Conti, M., and Gregori, E. Design and performance evaluation of an asymptotically optimal backoff algorithm for ieee 802.11 wireless lans. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on* (2000), IEEE, pp. 10–pp.

[28] Chafe, C., Gurevich, M., Leslie, G., and Tyan, S. Effect of time delay on ensemble accuracy. In *Proceedings of the International Symposium on Musical Acoustics* (2004), vol. 31.

[29] Chandra, P., and Lide, D. *Wi-Fi Telephony: Challenges and solutions for voice over WLANs*. Newnes, 2011.

[30] Chandra, R., Bahl, P., and Bahl, P. Multinet: connecting to multiple ieee 802.11 networks using a single wireless card. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* (March 2004), vol. 2, pp. 882–893 vol.2.

[31] Cheng, Y.-C., Afanasyev, M., Verkaik, P., Benko, P., Chiang, J., Snoeren, A. C., Savage, S., and Voelker, G. M. Shaman automatic 802.11 wireless diagnosis system.

[32] Cheng, Y.-C., Bellardo, J., Benkö, P., Snoeren, A. C., Voelker, G. M., and Savage, S. *Jigsaw: solving the puzzle of enterprise 802.11 analysis*, vol. 36. ACM, 2006.

[33] Claypool, M., and Claypool, K. Latency can kill: precision and deadline in online games. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems* (2010), ACM, pp. 215–222.

[34] Croitoru, A., Niculescu, D., and Raiciu, C. Towards wifi mobility without fast handover. Usenix NSDI.

[35] Ding, N., Pathak, A., Koutsonikolas, D., Shepard, C., Hu, Y. C., and Zhong, L. Realizing the full potential of psm using proxying. In *INFOCOM, 2012 Proceedings IEEE* (2012), IEEE, pp. 2821–2825.

[36] Ding, N., Wagner, D., Chen, X., Pathak, A., Hu, Y. C., and Rice, A. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement*

and Modeling of Computer Systems (New York, NY, USA, 2013), SIGMETRICS '13, ACM, pp. 29–40.

[37] GRIGORIK, I. *High Performance Browser Networking: What every web developer should know about networking and web performance*. " O'Reilly Media, Inc.", 2013.

[38] HE, H., GARCIA, E., ET AL. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on 21*, 9 (2009), 1263–1284.

[39] HUANG, J., CHEN, C., PEI, Y., WANG, Z., QIAN, Z., QIAN, F., TIWANA, B., XU, Q., MAO, Z., ZHANG, M., ET AL. Mobiperf: Mobile network measurement system. Tech. rep., Technical report, Technical report, Technical report). University of Michigan and Microsoft Research, 2011.

[40] JI, X., HE, Y., WANG, J., WU, K., YI, K., AND LIU, Y. Voice over the dins: improving wireless channel utilization with collision tolerance. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on* (2013), IEEE, pp. 1–10.

[41] JIANG, J., SEKAR, V., STOICA, I., AND ZHANG, H. Shedding light on the structure of internet video quality problems in the wild. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2013), CoNEXT '13, ACM, pp. 357–368.

[42] JUDD, G., AND STEENKISTE, P. Fixing 802. 11 access point selection. *ACM SIGCOMM Computer Communication Review 32*, 3 (2002), 31–31.

[43] KATEJA, R., BARANASURIYA, N., NAVDA, V., AND PADMANABHAN, V. N. Diversifi: Robust multi-link interactive streaming. CoNEXT.

[44] MAHAJAN, R., RODRIG, M., WETHERALL, D., AND ZAHORJAN, J. Analyzing the mac-level behavior of wireless networks in the wild. In *ACM SIGCOMM Computer Communication Review* (2006), vol. 36, ACM, pp. 75–86.

[45] NICHOLSON, A. J., CHAWATHE, Y., CHEN, M. Y., NOBLE, B. D., AND WETHERALL, D. Improved access point selection. In *Proceedings of the 4th international conference on Mobile systems, applications and services* (2006), ACM, pp. 233–245.

[46] PATRO, A., GOVINDAN, S., AND BANERJEE, S. Observing home wireless experience through wifi aps. In *Proceedings of the 19th annual international conference on Mobile computing & networking (MobiCom 2013)*, ACM, pp. 339–350.

[47] PAUL, U., KASHYAP, A., MAHESHWARI, R., AND DAS, S. R. Passive measurement of interference in wifi networks with application in misbehavior detection. *Mobile Computing, IEEE Transactions on 12*, 3 (2013), 434–446.

[48] PENG, G., ZHOU, G., NGUYEN, D. T., AND QI, X. All or none? the dilemma of handling wifi broadcast traffic in smartphone suspend mode. *Power (mW) 300*, 400, 500.

[49] QIAN, F., WANG, Z., GAO, Y., HUANG, J., GERBER, A., MAO, Z., SEN, S., AND SPATSCHECK, O. Periodic transfers in mobile applications: network-wide origin, impact, and optimization. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 51–60.

[50] SHAFIQ, M. Z., ERMAN, J., JI, L., LIU, A. X., PANG, J., AND WANG, J. Understanding the impact of network dynamics on mobile video user engagement. In *The 2014 ACM international conference on Measurement and modeling of computer systems* (2014), ACM, pp. 367–379.

[51] SHRIVASTAVA, V., RAYANCHU, S. K., BANERJEE, S., AND PAPAGIANNAKI, K. Pie in the sky: Online passive interference estimation for enterprise wlans. In *NSDI* (2011), vol. 11, p. 20.

[52] SUI, K., ZHAO, Y., PEI, D., AND ZIMU, L. How bad are the rogues' impact on enterprise 802.11 network performance? In *Computer Communications (INFOCOM), 2015 IEEE Conference on* (April 2015), pp. 361–369.

[53] SUNDARESAN, S., FEAMSTER, N., TEIXEIRA, R., MAGHAREI, N., ET AL. Measuring and mitigating web performance bottlenecks in broadband access networks. In *ACM Internet Measurement Conference* (2013).

[54] WANG, X. S., BALASUBRAMANIAN, A., KRISHNAMURTHY, A., AND WETHERALL, D. How speedy is spdy. In *NSDI* (2014), pp. 387–399.

[55] XU, F., TAN, C. C., LI, Q., YAN, G., AND WU, J. Designing a practical access point association protocol. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.