HAL
open science

Passive Wi-Fi Link Capacity Estimation on Commodity Access Points

Diego Neves da Hora, Karel van Doorselaer, Koen van Oost, Renata Teixeira, Christophe Diot

To cite this version:

Diego Neves da Hora, Karel van Doorselaer, Koen van Oost, Renata Teixeira, Christophe Diot. Passive Wi-Fi Link Capacity Estimation on Commodity Access Points. Traffic Monitoring and Analysis Workshop (TMA) 2016, Apr 2016, Louvain-la-Neuve, Belgium. hal-01292633

HAL Id: hal-01292633

https://inria.hal.science/hal-01292633v1

Submitted on 23 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

©IFIP, (2016). This is the author's version of the work. It is posted here by permission of IFIP for your personal use. Not for redistribution. The definitive version was published by Springer in Traffic Monitoring and Analysis Workshop, Louvain-la-Neuve, Belgium, April 2016 and is available at www.springerlink.com

Passive Wi-Fi Link Capacity Estimation on Commodity Access Points

Diego da Hora[1,2], Karel Van Doorselaer[1], Koen Van Oost[1], Renata Teixeira[2], Christophe Diot[3]
[1]Technicolor, [2]Inria, [3]SAFRAN
Email: {*First.Lastname*}*@Technicolor.com*[1], *renata.teixeira@inria.fr*[2], *christophe.diot@safran.fr*[3]

***Abstract*—Wi-Fi is the preferred way of accessing the internet for many devices at home, but it is vulnerable to performance problems. In this work, we propose a method to estimate the link capacity of a Wi-Fi link using physical layer metrics passively sampled on commodity access points. We build a model that predicts the maximum UDP throughput a device can sustain, which extends previous models to consider IEEE 802.11n optimizations such as frame aggregation. We validate our model through controlled experiments in an anechoic chamber. Over 95% of the link capacity predictions present errors below 5% when using our model with reference data. We show how link capacity estimation enables Wi-Fi diagnosis in two case studies where we predict the available bandwidth under microwave interference and in an office environment.**

I. Introduction

Wi-Fi is the preferred way of accessing the Internet at home; many devices today connect only via wireless. Unfortunately, Wi-Fi performance is highly variable. For example, in dense urban neighborhoods it is typical to see tens of competing Wi-Fi networks [11] and other non-Wi-Fi devices (e.g., microwave ovens), which will cause contention and interference. Sub-optimal installation of the Wi-Fi access point (AP) can also degrade performance. For example, the AP may be placed in a location that leaves devices with weak signal. Our discussions with residential Internet Service Providers (ISPs) indicate that often, when customers call to complain about poor performance, the problem is in the home Wi-Fi, not the ISP network. Studies of home networks confirm that Wi-Fi can cause poor performance [11], [17]. Diagnosing problems in the home Wi-Fi is challenging for ISPs due to the lack of visibility within the home network. In many cases, however, the ISP provides the home AP.

We believe that ISPs should instrument APs to monitor Wi-Fi performance and assist in diagnosis. Although previous studies have already instrumented APs to monitor wireless performance [8], [10], [11], [17], [18], our aim is to develop a practical solution that ISPs can deploy *at scale on commodity APs to diagnose user Wi-Fi problems*. This goal brings some restrictions. First, we exclude solutions that rely on APs with multiple Wi-Fi NICs [8], [10]. The use of multiple NICs is appealing because one NIC can implement the usual AP functionality, whereas the other can perform per-packet monitoring without interfering with users' traffic. We instead directly monitor the NIC that is acting as AP. Second, we only rely on passive measurements. Active measurement solutions [5], [18] can help in on-demand diagnosis, but periodic active measurements would disrupt users' traffic and drain the battery of mobile devices. Third, we exclude the use of per-packet metrics [17]. Processing metrics per packet introduces overhead in periods of high network load that negatively impact the AP performance. Finally, we want our solution deployed at large scale, so we only rely on standard metrics available on commercial APs, which require no hardware or significant driver modifications.

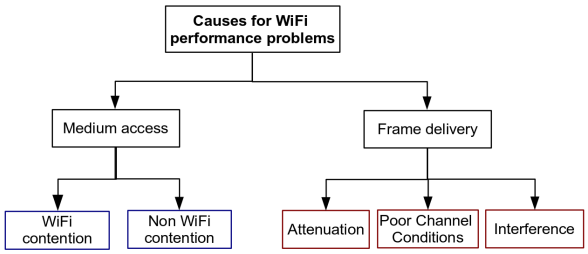

Fig. 1. Causes for Wi-Fi performance problems

In this paper, 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 [4] 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 (§IV). Then, we extend this model to the realistic case where the PHY rate varies and frame losses occur (§V).

Link capacity is useful for Wi-Fi diagnosis. As illustrated in Figure 1, 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) [8]. In §VI, 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. Our methods are part of a broader Wi-Fi diagnosis solution that is in trial with two major European ISPs and one major ISP in Asia-pacific. Each trial deployment is continually monitoring 30–50 APs.

II. Background

In this section, we present a brief background on Wi-Fi concepts, in particular of 802.11n.

802.11 MAC protocol. The IEEE 802.11 Medium Access protocol uses carrier sense multiple access with collision avoidance. Enhanced Distributed Channel Access is used to coordinate medium access: nodes only transmit after they sense the medium idle for a duration of Arbitration Interframe Space (AIFS) plus a backoff timer. Frames need to be acknowledged by the receiver through an ACK frame, and each transmission is spaced by a Short Interframe Space (SIFS). If the ACK is not received, the sender infers a collision, deferring transmission and exponentially increasing the contention window size if a retransmission is due.

RTS and CTS. 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. 802.11n stations are required to support HT-immediate Block ack, which uses Block Ack frames (BA) to acknowledge a set of MPDUs after the reception of an A-MPDU to improve MAC efficiency. The number of 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 transmitter (MAX_{agg}) is implementation dependent, but it is mainly bound by the field *Maximum A-MPDU Length Exponent* advertised by the receiver on control messages [12]. Figure 2 illustrates a typical A-MPDU with RTS/CTS protection. We study the frame aggregation impact on performance of commercial access points in §IV-B.

PHY rate and rate adaptation algorithm. An 802.11n device is able to use any of the PHY rates introduced by 802.11n as well as legacy 802.11g and 802.11b rates. The maximum PHY rate for 802.11n is 65Mbps[1] for a single spatial stream. Devices with two or more antenna chains can increase the PHY rate with Multiple-Input-Multiple-Output (MIMO). 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 aren't specified by the standard, popular implementations are Adaptive Multi-Rate Retry, Onoe and Sample Rate [19].

[1]Considering usage of long-guard interval.

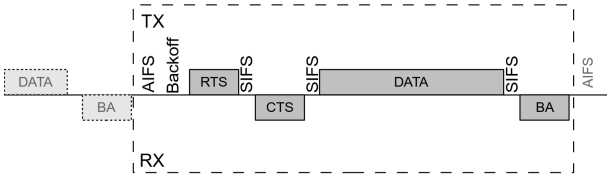

Fig. 2. A-MPDU frame exchange with RTS/CTS protection

III. 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 §IV-B. 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 (connected 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 §V-B. Table I 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.

IV. 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.

A. Model

Our model of link capacity extends the model of Jun et. al [4] 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:

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

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 §IV-B.

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 2 shows our A-MPDU frame-exchange model, which uses Enhanced Distributed Channel Access (EDCA). We compute the A-MPDU transmission delay as:

$$\text{TxDelay(N,S,P)} = T_{AIFS} + T_{BO} + 3 \times T_{SIFS} + T_{RTS} + T_{CTS} + T_{ACK} + \text{DATA(N,S,P)} \quad (2)$$

TABLE II
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 III
Timing of control frames

Modulation	T_{RTS}	T_{CTS}	T_{ACK}
OFDM 24	$28\mu s$	$28\mu s$	$32\mu s$[2]
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	$352\mu s$	$304\mu s$	$304\mu s$

We assume no frame losses (we remove this simplifying assumption in the next section). With no losses, the backoff timer does not exponentially increase. The backoff timer is chosen using an 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:

$$\text{DATA(N,S,P)} = T_{PH} + \frac{22 + N \times (S_{MH} + S)}{P} \quad (3)$$

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 II. 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 [4] (shown in Table III).

B. Parameter tuning

Our model in §IV-A 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 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.

[2]Duration of a Block ACK

TABLE I
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
Metrics per station	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

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 μs. 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 decide the aggregation of frames into A-MPDUs, with most modern 802.11n devices opting for the latter [1]. 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 experiment, the AP sends data to the android tablet or to the MacBook using fixed PHY rate, as described in §III. The number of frames per A-MPDU used by the tablet and the MacBook is shown in Columns 2 and 5 of Table IV, 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:

$$AGG(P) = min\left(\left\lfloor \frac{P \times txop}{\text{MAC frame length}} \right\rfloor, MAX_{agg}\right) \quad (4)$$

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 IV. 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 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.

TABLE IV
ESTIMATED A-MPDU SIZE (AGG), TEMPORAL DURATION (DUR) IN μs 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

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.

C. 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 §III, 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 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 4 shows the MAC efficiency for tests with the Android Tablet. A-MPDU size reaches maximum value when

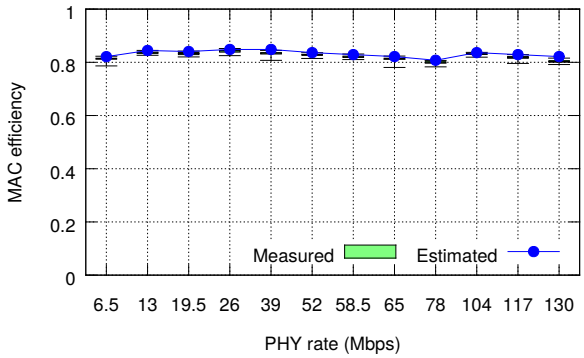

Fig. 3. Estimated vs. measured link capacity of MacBook

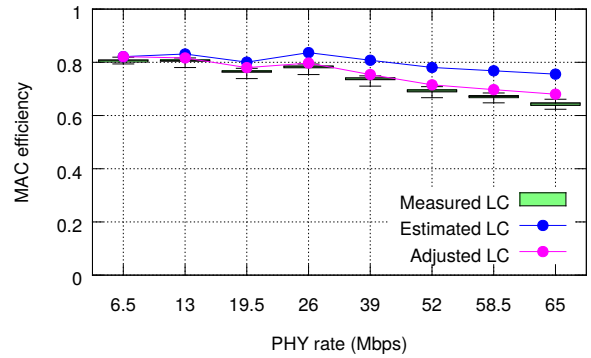

Fig. 4. Estimated vs. measured link capacity of tablet

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.

V. Link Capacity in Practice

This section adapts the model from §IV to work in practice. Rate adaptation algorithms frequently change the selected PHY rate [6] and frames may be lost (i.e., frame delivery $< 100\%$). We first adapt the model from §IV 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.

A. 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 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:

$$LC(t_0, \tau) = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} FDR(t) \times LC(P(t))dt. \quad (5)$$

B. Model inputs

The model in Equation 5 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, τ, depends on how frequently and fine-grained operators want to estimate link capacity. Values of τ 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 τ 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.
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 [10]. 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 τ and λ parameters as follows. We emulate PHY rate sampling using the sniffer logs of the UDP iperf capacity tests, as described in §III. We periodically sample the PHY rate of the last data frame, with periodicity λ over an estimation period τ. 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.

We can see in Figure 5 that sampling error decreases for larger τ and smaller λ. This is expected since estimation error tends to decrease with more PHY rate samples, which is given by τ/λ. Therefore, there is a trade-off between sampling

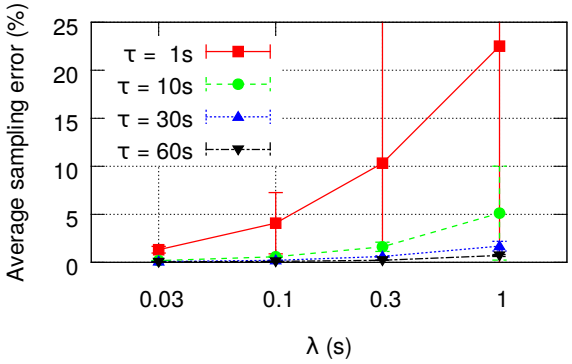

Fig. 5. Average error when sampling the Link Capacity.

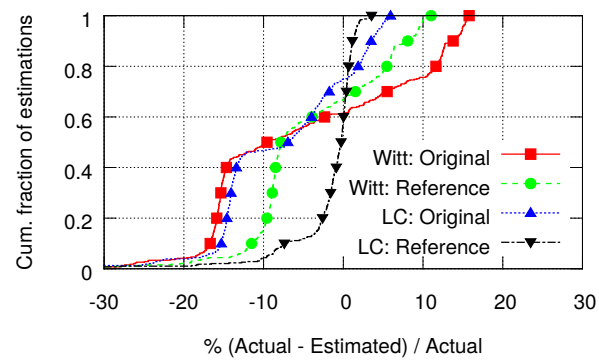

Fig. 6. Comparison between throughput estimation methods

overhead, imposed by λ, and estimation granularity, imposed by τ. We see that sampling error is, on average, below 2% when $\tau/\lambda \geq 30$.

C. Validation

In order to evaluate our algorithm, we performed controlled experiments in the anechoic chamber. We performed UDP iperf capacity tests as described in §III, 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) [10]. Witt is calculated by linearly fitting a custom metric, link experience, with TCP throughput ground truth data. Link experience and Witt are given by:

$$\text{link_exp} = (1-a) \times (1-c) \times \sum_i \frac{s_i.r_i}{p_i} \tag{6}$$

$$Witt = \beta_1 \times \text{link_exp} + \beta_0 \tag{7}$$

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 β_1 on a data set and minimizing errors with the intercept β_0. We calculate the MAC efficiency per PHY rate. We see this when comparing Witt's original (β_0, β_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 β values; 2) Witt with β 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 §IV (LC: Reference).

Figure 6 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 §IV. Fortunately, this has to be done once per AP model.

VI. 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 §V 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.

A. Medium Access and Frame delivery losses

Figure 7 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 §V estimates how much bandwidth the link supports 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.

$$MA = LC \times (BUSY_{\text{WiFi}} + BUSY_{\text{NonWiFi}}) \tag{8}$$

$$AB = LC \times (1 - BUSY_{\text{WiFi}} - BUSY_{\text{NonWiFi}}) \tag{9}$$

Consider an ideal Wi-Fi link, with no frame loss nor medium sharing. This link would present the maximum LC

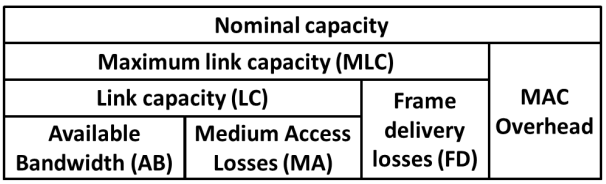

Fig. 7. Breakdown of nominal capacity into: maximum link capacity, link capacity, available bandwidth, medium access losses and frame delivery losses

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.

$$FD = MLC - LC \tag{10}$$

B. Case studies

Here we show two case studies, where we apply the metrics defined in §VI-A 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 iperf to generate UDP traffic from the AP to the tablet as described in §III. 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 8 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 9 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 modulation. Our method allows us to identify that the throughput loss during this period was due to a frame delivery loss.

VII. Related Work

There is a large body of work on throughput estimation techniques in Wireless networks. Jun et al. presents an analytical model to calculate the theoretical maximum throughput of IEEE 802.11a, b and g [4]. Skordoulis et al. compare the different frame aggregation mechanisms proposed in IEEE 802.11n, giving simulation results for the maximum throughput [15]. These models, however, always consider fixed PHY rate usage. We extend these models by considering IEEE 802.11n parameters and estimating the link capacity when PHY rate varies, while experimentally validating the results.

Active measurement methods have been proposed for estimating the available bandwidth in wireless networks. Lakshminarayanan et al. proposes Probegap, a probing technique to estimate bandwidth in multi-rate scenarios, such as Wireless networks [7]. Mingzhe Li et al. proposes WBest [9], a tool that uses packet-pairs and packet-trains to determine achievable throughput in IEEE 802.11 networks. Those techniques are unfit for large scale ISP deployments, since the active measurements can disrupt users' traffic as well as require cooperation between endpoints.

Many works have characterized wireless performance in the wild using the AP point of view [2], [11], [16], often relying on passive measurement collection. Patro et al. proposes Witt [10], a metric that estimates available bandwidth based on passive metrics from APs, using it to characterize the Wi-Fi quality in 30 homes. We obtain estimation errors similar to Witt, but our model is less sensitive to parameter tuning.

There is work on diagnosing wireless performance problems in enterprise and campus networks, where multiple APs are managed by the same entity [2], [3]. These solutions often require combining multiple points of view to diagnose performance problems, while we can only rely on the single home AP. Rayanchu et. al proposes a method to diagnose frame collision from weak signal [13]. This method requires modifications to the Wi-Fi kernel of both the AP and the device. Lakshminarayanan et al. proposes the use of a second NIC to allow users to diagnose the medium usage, allowing for accurate diagnosis of Wi-Fi and Non-Wi-Fi interferences [8]. Syrigos et al. proposes the use of active measurements with fixed PHY rate to diagnose common 802.11 pathologies [18]. Kanuparthy et al. proposes using user-level probing to diagnose common WLAN performance problems [5]. Shravan et al. proposes Airshark, a system which is able to detect different non-Wi-Fi interference sources by using energy samples from the Wi-Fi card [14]. These methods are able to diagnose Wireless problems with a great degree of precision, but either require the disruption of the user activity with active tests [5], [18] or a second NIC for dedicated spectrum monitoring, limiting deployment [8], [14]. We instead focus on providing high level diagnosis, distinguishing between *medium access* and *frame delivery* throughput losses.

VIII. Conclusions and Future Work

We presented an algorithm to estimate the link capacity based on passive metrics from APs, which is ready to be

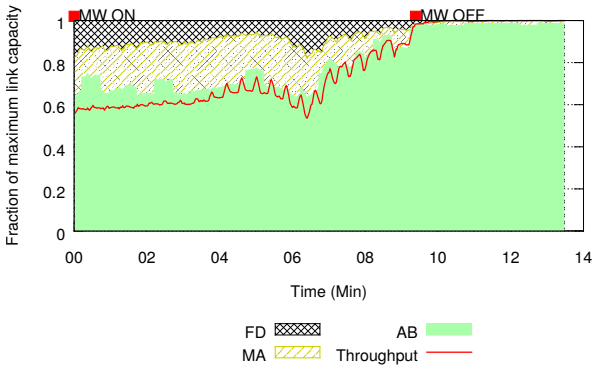

Fig. 8. Available bandwidth under microwave interference

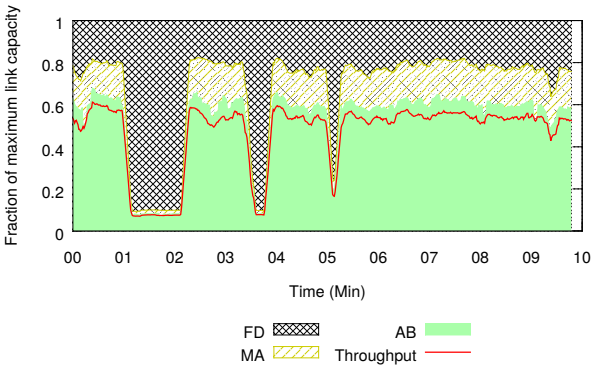

Fig. 9. Available bandwidth prediction on the wild

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 $\tau/\lambda \geq 30$.

We would like to use the proposed model to understand the home wireless performance of real users. Some of the monitoring capabilities described are already in trial in two large two major European ISPs and one major ISP in Asia-pacific. We plan to use this proposed model to perform a characterization of the Wireless performance in these home networks, answering questions such as "how often is wireless performance poor?" and "what is the most common root cause for reduced wireless performance?". Finally, we want to see how available bandwidth estimations correlate with QoE metrics, which more closely reflect the user experience.

Acknowledgements

We thank Gabriel Matei and Ali Louzir for helping with the anechoic chamber experiments. Thanks also to our shepherd Pedro Casas and the anonymous reviewers for their helpful comments. This work was supported by the European Community Seventh Framework Programme (FP7/2007-2013) no. 611001 (User-Centric Networking).

References

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

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

[3] 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*, volume 36. ACM, 2006.

[4] 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.

[5] P. Kanuparthy, C. Dovrolis, K. Papagiannaki, S. Seshan, and P. Steenkiste. Can user-level probing detect and diagnose common home-wlan pathologies. *ACM SIGCOMM Computer Communication Review*, 42(1):7–15, 2012.

[6] 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.

[7] 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.

[8] K. Lakshminarayanan, S. Seshan, and P. Steenkiste. Understanding 802.11 performance in heterogeneous environments. In *Proceedings of the 2nd ACM SIGCOMM workshop on Home networks*, pages 43–48. ACM, 2011.

[9] 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.

[10] A. Patro, S. Govindan, and S. Banerjee. Observing home wireless experience through wifi aps. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 339–350. ACM, 2013.

[11] 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 *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2713–2731. IEEE, 2015.

[12] E. Perahia and R. Stacey. Mac frame formats. In *Next Generation Wireless LANS: 802.11 n and 802.11 ac*, chapter 12. Cambridge university press, 2013.

[13] 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.

[14] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: detecting non-wifi rf devices using commodity wifi hardware. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 137–154. ACM, 2011.

[15] 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.

[16] 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.

[17] S. Sundaresan, N. Feamster, and R. Teixeira. Measuring the Performance of User Traffic in Home Wireless Networks. In *Passive and Active Network Measurement Conference*, Lecture notes in computer science, Mar. 2015.

[18] I. Syrigos, S. Keranidis, T. Korakis, and C. Dovrolis. Enabling wireless lan troubleshooting. In *Passive and Active Measurement*, pages 318–331. Springer, 2015.

[19] 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.