Multi-User Downlink with Single-User Uplink can Starve TCP

Peshal Nayak [*], Michele Garetto [†], Edward W. Knightly [*]
[*]Rice University, Houston, TX, USA [†]Università di Torino, Italy

***Abstract*—In this paper we present the first cross-layer analysis of wireless LANs operating under downlink multi-user MIMO (MU-MIMO), considering the fundamental role played by closed-loop (TCP) traffic. In particular, we consider an 802.11ac scenario in which the access point transmits on the downlink via MU-MIMO, whereas stations must employ single-user transmissions on the uplink. With the help of analytical models built for the different regimes that can occur in the considered system, we identify and explain crucial performance anomalies that can result in very low throughput in some scenarios, completely offsetting the theoretical gains achievable by MU-MIMO. We discuss solutions to mitigate the risk of this performance degradation and alternative uplink strategies allowing WLANs to approach their maximum theoretical capacity under MU-MIMO.**

I. Introduction

Downlink multi-user MIMO (DL MU-MIMO) is a promising physical-layer technology to boost the capacity of wireless LANs by transmitting data streams to multiple stations (STAs) concurrently, thus scaling up the achievable data rate by a factor equal to the number of antennas on the Access Point (AP). This approach is different from traditional single-user (SU) networks where only one STA gets served at a time.

With inclusion in the IEEE 802.11ac standard [1], [2], DL MU-MIMO has moved from theoretical research into the real world. However, we are still far from observing in practice the capacity gains promised by advanced physical-layer technologies such as MU-MIMO.

In this paper, we show that one root cause of disappointing WLAN performance is poor cross-layer design: making just the AP much more powerful in sending downlink traffic does not necessarily correspond to an equivalent gain in terms of throughput perceived by users at the transport layer, even if the vast majority of bytes are transmitted in the downlink direction, e.g., via download of large files via TCP. Specifically, we show that severe performance degradation can occur when DL MU-MIMO is coupled with a single-user uplink (as now standardized) under closed-loop traffic such as that generated by TCP, which still carries more than 80% [3] of the Internet traffic today. Surprisingly, performance can be worse than that achieved by a single-user single-antenna downlink (i.e., neither multi-user nor MIMO), even under ideal channel and network conditions.

Our work provides the following contributions: *(i)* we present, to the best of our knowledge, the first cross-layer performance evaluation study of MU-MIMO under closed-loop (TCP) traffic; *(ii)* we develop novel analytical techniques to compute the throughput of a WLAN operating under downlink MU-MIMO, and the standard channel access mechanism of 802.11; *(iii)* with the help of our models, we identify the fundamental reasons for the poor performance that can be observed in a realistic network scenario operating under 802.11ac compliant MU-MIMO; in particular, we show the crucial role played by frame aggregation for uplink transmissions by the stations and the intrinsic limitations due to suboptimal multiplexing gain resulting from random channel contention; *(iv)* we discuss different uplink strategies that can overcome the above limitations and approach the maximum theoretical performance.

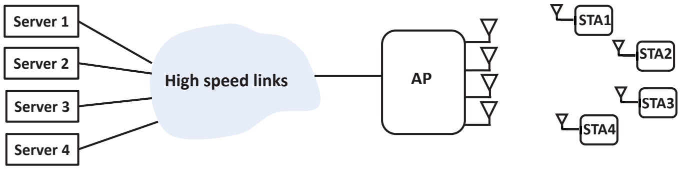


Fig. 1. Network topology for studying impact of closed loop traffic.

The rest of the paper is organized as follows. In Sec. II we present the network scenario considered in our work, including the necessary background on DL MU-MIMO. In Sec. III we describe our model of the considered system, and a simple high-level characterization of the different regimes that can occur. Detailed analytical models are developed in Sec. IV for the most significant cases, and validated by simulation. In Sec. V we compare different uplink strategies from a system-design perspective. We discuss related work in Sec. VI and conclude in Sec. VII.

II. Network scenario

A. Cross-layer Setup

To investigate the performance of DL MU-MIMO under closed-loop traffic, we consider the network scenario illustrated in Fig. 1. A set of users (or stations[1]) attached to a wireless LAN establish long-lived TCP flows to download bulk data from a set of servers located in the wired network. To isolate the targeted factors, we assume that data is sent only downlink, so that just TCP ACKs are sent in the uplink direction. Servers are connected to the AP over high speed links, which ensures absence of congestion and queueing delays in the wired portion of the network.

In this scenario, there are no losses in the backbone, therefore each TCP flow (discarding an initial transient) operates at the

This research was supported by Cisco, Intel, the Keck Foundation, and by NSF grants CNS-1642929, CNS-1514285, and CNS-1444056.

[1] In this paper we use the term user and station (STA) interchangeably.

maximum TCP congestion window size. As a consequence, TCP dynamics related to specific versions of the TCP protocol do not come into play in our scenario. Essentially, the only TCP feature that matters is the fact that data (ACK) packets are transmitted by TCP senders (receivers) in response to ACK (data) packets received in the opposite direction. This captures the closed-loop nature of the traffic generated by almost all versions of TCP.

Note that, while operating at the maximum congestion window size, TCP senders transmit one data packet in response to each TCP ACK (or two data packets, if the delayed ACK option is enabled [4]). We assume that all TCP flows traverse the same AP, which is equipped with multiple antennas and performs MU-MIMO transmissions on the wireless channel whenever possible, i.e., when the AP has backlogged traffic for more than one user.

As is the case with IEEE 802.11ac, uplink transmissions by the stations are instead single-user, i.e., the STAs transmit on the uplink one at a time as dictated by random access. In general, the STAs could also be equipped with multiple-antennas, and thus perform SU-MIMO by transmitting multiple streams to the AP simultaneously (we account for this in our analysis).

We will be especially interested in analysing the standard case in which channel access is governed by the fair 802.11 contention mechanism, which provides equal probability of contention victory to all nodes competing for transmission: each node that intends to transmit generates a random value for the backoff timer chosen uniformly from $[0, W_0 - 1]$ where $W_0 = 16$ is the minimum contention window size. While the channel is sensed idle, the node counts down with a slot duration of σ, and transmits when the backoff timer becomes zero.

Since the random channel access protocol of 802.11 can be responsible for severe throughput degradation of MU-MIMO under conditions that we will uncover in this paper, alternative channel access strategies will be considered later in Sec. V.

B. Background on 802.11ac compliant MU-MIMO

Here, we review the key components of the 802.11ac timeline for our analysis. When the AP obtains access to the channel by winning contention, it performs a transmission including three main phases:

Channel Sounding and feedback phase. The AP requires channel state information at the transmitter (CSIT) to limit interference among users. Consequently, it initiates a sounding process by transmitting a Null Data Packet Announcement (NDPA) which contains information that identifies the STAs that the AP intends to transmit data to on the downlink. Following this, the AP transmits a Null Data Packet (NDP) which contains the pilot sequence that the STAs use to estimate the CSI. The STAs process the CSI to calculate the angles ϕ and ψ that are used to build the transmit weight matrix at the AP [5]. The STAs transmit these in a compressed beamforming report (CBR), as polled by AP.

Data transmission phase. Data is transmitted simultaneously to the users, typically via zero-forcing beamforming using the collected CSIT. To amortize overhead and improve performance, the AP aggregates multiple frames destined to the same STA into the same data bundle. We emphasize that 802.11ac allows up to 1 MB to be aggregated per STA.

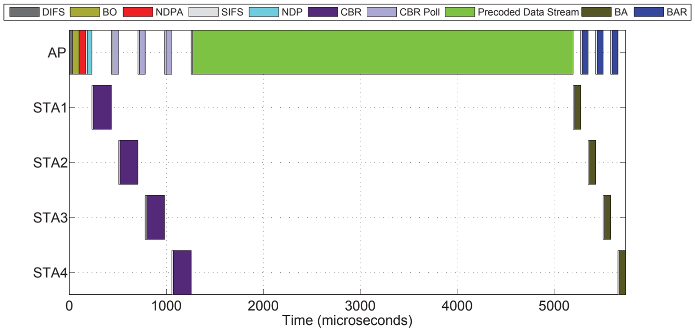


Fig. 2. An example of 802.11ac downlink transmission timeline in the case of an AP with 4 transmit antennas serving 4 single-antenna STAs.

Acknowledgement phase. After the AP transmits data, the first STA responds with a Block Acknowledge (BA). Following this, the AP subsequently transmits a block acknowledgement request (BAR) to other STAs, which then transmits their BA.

Fig. 2 shows an example 802.11ac downlink transmission for an AP with four transmit antennas serving four single-antenna STAs, in the case of channel bandwidth 20MHz, sub-carrier grouping of 4 and quantization bits for ϕ and ψ being 7 and 5 respectively. These values result in the minimum possible sounding and feedback phase duration at this bandwidth. Note that, even in this case, the total overhead due to channel sounding and feedback phases is about 1.5 milliseconds. During this time, roughly 10 data packets of size 1 KB could be transmitted using standard SISO. Therefore, aggregation of at least a few tens of frames (among all stations) is necessary to get any performance gain from MU-MIMO with respect to traditional SISO.

To validate the results obtained in this paper, we extended the simulator ns3 [6] to incorporate detailed behavior of 802.11ac compliant MU-MIMO WLANs.

III. System model

A. Assumptions and notation

Let K be the number of stations attached to the AP, which are destination of at least one long-lived TCP flow. Our goal is to compute the aggregate steady-state throughput Λ achieved by the set of all TCP flows.

In some of the scenarios that we will consider, the aggregate throughput will be limited by the TCP maximum window size $W_{\max}$ (expressed in number of segments). In those cases, we will assume for simplicity a symmetric traffic scenario: stations establish an equal number F_s of TCP flows, and all flows experience the same two-way propagation delay D in the fixed network.

To simplify the analysis, we further assume a perfect wireless channel (without errors) and a collision-free MAC protocol.[2]

[2] Under the 802.11 MAC protocol, the absence of collisions can be obtained (i.e., simulated with ns3) by assuming that the backoff extracted by a node is continuous, rather than discrete, and that nodes instantaneously freeze their backoff as soon as another node starts transmitting.

While these assumptions are simplifications of the real system, they enable us to capture macroscopic effects into a parsimonious analytical model. Channel errors and/or collisions could be incorporated in the analysis using well-established techniques [7], [8], but we do not do so here to keep the analysis focused on the joint impact of a closed-loop transport layer with a multi- and single-user MAC.[3]

We consider an AP implementing a work-conserving policy: when it has at least one packet to transmit, the AP starts contending for channel access. When it wins the channel, the AP employs multi-user MIMO whenever it has packets queued for at least two different stations (if it has packets destined to only a single station, the AP employs single-user MIMO). Note that the AP maintains a separate queue to store the packets destined to each attached station. Let N_{AP} be the number of antennas in the AP. Let N_{STA} be the number of antennas in each of the stations. If $N_{\text{AP}} < K$, it is possible that the number of stations for which the AP has a non-zero backlog is larger than the number of antennas at the AP. In this case, we assume that the AP will pick N_{AP} different stations with non-zero backlog uniformly at random. Let $A(h, b)$ be the channel holding time of the AP, which depends on two parameters: the number of non-empty queues h, and the largest backlog b of these queues.

Let B_{AP} be the maximum number of frames destined to the same station that can be aggregated and sent by the AP in one channel access. Note that B_{AP} will never constrain performance when $B_{\text{AP}} > F_s W_{\max}$, since in any case the AP cannot store a number of frames destined to the same station larger than the product of the TCP maximum window size times the number of flows per station.

Let B_{STA} be the maximum number of frames (TCP ACKs, in our case) destined to the AP that can be aggregated and sent by a station in one channel access.

We emphasize that the vast majority of existing performance evaluation studies of 802.11, focused on early versions of the standard, only consider the case $B_{\text{STA}} = B_{\text{AP}} = 1$. The impact of aggregation (in particular, possibly different levels of aggregation performed by the AP and by the stations) is instead fundamental to understand the performance of MU/SU MIMO systems.

B. High-level packet dynamics

In the absence of congestion in the backbone, each long-lived TCP flow reaches a steady-state condition with $W_{\max}$ outstanding packets in the network.

Note that these $W_{\max}$ packets can be travelling around the network either in the form of data packets or in the form of TCP ACKs. As a consequence the system behaves as a closed queueing network with a constant number of 'customers', where it is not really important to distinguish whether customers are data packets or TCP ACKs. Note that the aggregate system throughput essentially depends on how fast these customers circulate around the network.

Unfortunately, traditional methods (such as product-form solutions) cannot be applied here to solve the queueing network model of the system, due to correlated batch services and complex synchronizations induced by the wireless channel. Nevertheless, we can still apply bottleneck analysis to identify the component that determines the overall system performance.

Actually, a crucial question we may ask ourselves is: where are the customers of the system most likely to be found at a given point in time? By looking again at the topology in Fig. 1, we observe that system customers can only be in one of three places: i) stored in the AP (or being transmitted by the AP); ii) stored in the stations (or being transmitted by a station); iii) 'flying' in the backbone.

Consider, initially, the case in which the number of packets flying in the backbone reaches its maximum value. This case always occurs when D is very small (possibly zero), or when $W_{\max}$ is large enough that TCP flows completely 'fill the pipe'. Then a simple saturation throughput analysis, to be described next, allows us to understand where the rest of customers are primarily to be found (i.e., either in the AP or in the stations).

C. Saturation throughput analysis

Suppose to start from a condition in which the MAC queues of the AP, and the MAC queue of each station, have a large backlog. The AP moves packets down into the stations, while stations push up packets back into the AP (through the backbone). Who wins?

The key observation here is that contention for the wireless channel is fair among all nodes trying to transmit on it. Therefore, on average, for one downlink transmission performed by the AP, we will have K uplink transmissions performed by the set of all stations. Now, under the assumption that the AP employs multi-user MIMO (if $N_{\text{AP}} > 1$), whereas stations employ single-user MIMO, the AP will push down on average

$$S_{\text{down}} = B_{\text{AP}} \cdot \min\{N_{\text{AP}}, K \cdot N_{\text{STA}}\}$$

in each cycle of $K + 1$ transmissions. Indeed, the number of concurrent streams is given by the minimum between the number of antennas on the transmitting and receiving sides, and we can assume that the maximum allowed number of packets (equal to B_{AP}) is transmitted on each stream. During the same cycle of $K + 1$ transmissions, the stations will send up on average

$$S_{\text{up}} = K \cdot B_{\text{STA}} \cdot \min\{N_{\text{AP}}, N_{\text{STA}}\} \cdot T_{\text{F}}$$

effective TCP ACKs. Indeed, each station will have (on average) one opportunity to transmit B_{STA} packets using single-user MIMO, and we have accounted for the fact that TCP receivers might thin the feedback traffic to improve performance [9], by transmitting only one out of T_{F} (Thinning Factor) ACKs. For example, the standard delayed ACK option of TCP [4] corresponds to $T_{\text{F}} = 2$. For later purposes, let $S_{\text{sta}} = B_{\text{STA}} \cdot \min\{N_{\text{AP}}, N_{\text{STA}}\} \cdot T_{\text{F}}$ be the maximum number of (effective) TCP ACKs sent by a station in one access, so that $S_{\text{up}} = K S_{\text{sta}}$.

If $S_{\text{down}} > S_{\text{up}}$, the AP will eventually be able to move its backlog into the stations, maintaining its queues almost empty

[3] Further, collisions typically produce only a second-order effect, while they do not lead to closed-form expressions (i.e., they require numerical fixed-point solutions).

from that time on. If $S_{\text{down}} < S_{\text{up}}$, the stations will instead be able to drain their backlog, and most of the packets will be found in the AP. If $S_{\text{down}} = S_{\text{up}}$, the AP and the set of all stations will maintain on average an equal backlog.

We emphasize that existing analytical models of IEEE 802.11 have focused only on the case $S_{\text{down}} \leq S_{\text{up}}$. This can be explained by the fact that, prior to the introduction of multi-user technique, it was reasonable to assume $B_{\text{STA}} = B_{\text{AP}}$ (and in many models $B_{\text{STA}} = B_{\text{AP}} = 1$), and $N_{\text{AP}} \leq K \cdot N_{\text{STA}}$. Note that earlier versions of 802.11 (without MIMO) correspond to $N_{\text{AP}} = N_{\text{STA}} = 1$. In all cases above, the AP becomes the performance bottleneck under closed-loop (e.g., TCP) traffic.

Multi-user MIMO has changed the picture by making the AP much more powerful than the typical station. Not only can the AP be equipped with many more antennas than its attached stations (which by itself would not be enough to move the bottleneck to the uplink), but more importantly, the AP must employ significant frame aggregation ($B_{\text{AP}} \gg 1$) to amortize the overhead necessary to set up multi-user transmissions. As a consequence, the performance bottleneck can shift to the uplink, which is one novel scenario analysed in our work.

D. Fundamental regimes

When the propagation delay D is small enough that TCP flows are able to fill the backbone pipe, previous discussion leads us to distinguish the following three fundamental regimes:

- ***downlink bottleneck* regime**. This regime occurs when both $S_{\text{down}} \leq S_{\text{up}}$ and $KF_sW_{\max} \gg S_{\text{down}}$. Under the above conditions, the AP can be assumed to operate in saturation conditions, i.e., to be always fully backlogged. This is actually a desirable property to achieve the capacity gain of DL MU-MIMO.
- ***uplink bottleneck* regime**. This regime occurs when both $S_{\text{down}} > S_{\text{up}}$ and $F_sW_{\max} \gg S_{\text{sta}}$. Under the above conditions, each station can be assumed to operate in saturation conditions, i.e., to be always fully backlogged.
- ***full aggregation* regime**. This regime occurs when both $S_{\text{down}} \geq KF_sW_{\max}$ and $S_{\text{sta}} \geq F_sW_{\max}$. Under the above conditions both the AP and the stations perform a large enough packet aggregation to completely empty their buffers at each channel access. This regime is different from the others because no node transmitting on the channel operates in saturation conditions.

Note that the *full aggregation* regime is a limiting case of the downlink (uplink) bottleneck regime as we increase the aggregation level performed by the AP (the stations).

As we increase the backbone delay D, the system performance will eventually be limited by the wired network delay, rather than by wireless channel dynamics. In our analysis we will also (partially) explore the impact of the backbone delay D in the regimes described above.

Remark. One crucial observation that we can already make at this point is the following: the size of data packets, and that of TCP ACKs, plays no role in determining the regime in which the system operates, as one can check by inspecting the conditions listed above for each regime. Specifically, the fact that TCP ACKs are much smaller in size than a TCP data packet does not modify in any way the system bottleneck. This fact is in sharp contrast to a common misconception, according to which the impact of uplink traffic is negligible because TCP ACKs are "small" (in size). As we will see, instead, the uplink feedback process can determine the overall system performance, although the large majority of traffic volume flows only downstream.

E. Reference system

Although the models developed in this paper are quite general, to validate our analysis we will consider a reference system closely following the network topology illustrated in Fig. 1 and the 802.11ac settings described in Sec. II-B. Specifically, we will always assume an AP equipped with 4 antennas (equal to the maximum number of concurrent streams considered in 802.11ac), operating at 54 Mb/s physical data rate per stream.

Stations are instead assumed to have a single antenna, thus performing single-user SIMO transmissions in the uplink. Unless otherwise specified, we assume 4 stations in the network, so that all of them can potentially be served concurrently by the AP.

We further assume that each station establishes a single long-lived TCP flow with a server ($F_s = 1$). Unless otherwise specified, the maximum TCP congestion window size is $W_{\max} = 200$. The TCP segment size is 1024 bytes, and we enable the delayed ACK option ($T_F = 2$).

In the next section, we will compare analytical results (for each of the regimes in Sec. III-D) with detailed ns3 simulations obtained in our reference system. To put our throughput figures under the right perspective, it is important to keep in mind the following simple upper bounds on Λ.

Given a physical data rate of 54 Mb/s, and 4 antennas, clearly we cannot exceed the trivial upper bound $\Lambda^{(1)} = 54 \cdot 4 = 216$ Mb/s, corresponding to the unrealistic case of zero overhead everywhere. Under the constraint of adopting the best 802.11ac-compliant MU-MIMO in the downlink, we obtain a better (tighter) bound as $\Lambda^{(2)} = KF_sW_{\max}/A(K, F_sW_{\max})$, by assuming zero overhead in the uplink: after the AP sends down the aggregate of all system packets, all data is acknowledged in zero time by the TCP receivers. In our reference system with $K = 4$, $F_s = 1$, $W_{\max} = 200$, we obtain $\Lambda^{(2)} = 192.5$ Mb/s. At last, assuming that all system packets, after been dumped by the AP, are sequentially acked by the stations (actually, one ACK every 2 packets, since $T_F = 2$), we obtain $\Lambda^{(3)} = KF_sW_{\max}/[A(K, F_sW_{\max}) + T_{\text{up}}(KF_sW_{\max}/T_F)] = 172.5$ Mb/s, where $T_{\text{up}}(KF_sW_{\max}/T_F)$ is the channel time to send the TCP ACKs (400 ACKs, in our case).

IV. Analysis

A. Downlink bottleneck regime

Recall that in this regime we assume the AP to be always fully backlogged. We consider a discrete-time Markov Chain embedded at the time instants at which the wireless channel becomes idle (i.e., at the end of a transmission) – see Fig. 3.

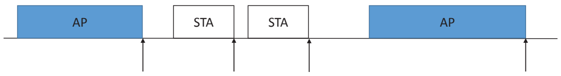


Fig. 3. Embedded discrete-time Markov Chain to analyse the downlink bottleneck regime.

The state of this Markov Chain is the set of queue lengths of the stations at the beginning of a cycle.

Standard renewal theory allows us to write the aggregate throughput Λ (in packets per seconds) as

$$\Lambda = \frac{\text{average number of packets sent in a cycle}}{\text{average cycle duration (s)}} \tag{1}$$

where packets can be *either* TCP data packets or (effective) TCP ACKs. Indeed, flow conservation (closed-loop traffic) implies that throughput in terms of data packets must be equal to throughput in terms of (effective) ACKs.

Any cycle is divided into two parts: a contention phase and a packet transmission phase. Let $\hat{K}$ be the random variable denoting the number of contending stations at the beginning of a cycle. To simplify the analysis, we assume that random backoffs are chosen according to an exponential distribution of mean $1/\mu$, instead of a uniform distribution in $[0, W_0 - 1]$ (in number of slots of duration σ). To match the first moment of the backoff distribution, we correspondingly set $\mu = 2/(W_0\sigma)$. Then the average duration of the contention phase, conditioned on having $\hat{K} = k$ contending stations ($k = 0, 1, \ldots, K$), is $\frac{1}{(k+1)\mu}$. If the AP wins the contention, which occurs with probability $\frac{1}{k+1}$, we have a downlink transmission of a data bundle by the AP consisting of S_{down} TCP data packets, occupying the channel for a duration $T_{\text{down}} = A(K, B_{\text{AP}})$. Instead, with probability $\frac{k}{k+1}$ the contention is won by a station, that will occupy the channel for a duration T_{up}.

An exact analysis of the system requires to track the queue lengths of the stations. However, following this approach would be an overkill, given that the system obeys flow conservation in the downlink and uplink directions. Actually, the only advantage of performing the above exact analysis would be to perfectly characterize the duration of the contention phase at the beginning of a cycle, which has however negligible impact on the overall throughput. Therefore, we adopt the following simplifying assumptions: i) a station always transmits $\min(B_{\text{AP}}, S_{\text{sta}})$ packets when it gets access on the channel; ii) the number $\hat{K}$ of contending stations, which is a random variable, is replaced by a constant value k^* obtained by flow conservation:

$$\frac{1}{k^*+1} S_{\text{down}} = \frac{k^*}{k^*+1} \min(B_{\text{AP}}, S_{\text{sta}})$$

which provides[4] $k^* = \frac{S_{\text{down}}}{\min(B_{\text{AP}}, S_{\text{sta}})}$. These might appear to be rough approximations but, to say it again, they only impact the computation of the average contention time at the beginning of a cycle, which has negligible impact on the throughput.

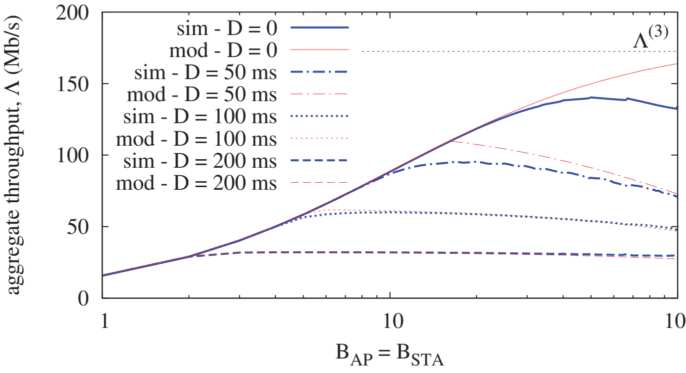


Fig. 4. Throughput comparison (model vs simulation) in the reference system, with $B_{\text{AP}} = B_{\text{STA}}$.

The above considerations allows us to derive the throughput according to (1):

$$\Lambda = \frac{\frac{1}{k^*+1} S_{\text{down}}}{\frac{1}{(k^*+1)\mu} + \frac{1}{k^*+1} T_{\text{down}} + \frac{k^*}{k^*+1} T_{\text{up}}} = \frac{S_{\text{down}}}{1/\mu + T_{\text{down}} + k^* T_{\text{up}}}$$

At last, we account for the fact that, as we increase the backbone two-way delay D, we will enter at some point the regime in which the backbone becomes the performance bottleneck. To do so, we adopt a simple approach based on the assumption that the queues of the AP are in one of two states: they are either empty, or they have sufficient backlog to send S_{down} packets in one channel access.

Let $\bar{C} = \frac{1}{\mu} + T_{\text{down}} + k^* T_{\text{up}}$ be the average time to send S_{down} packets downlink. Suppose that we start from a condition in which all $K F_s W_{\max}$ packets in the system are stored in the AP. If the backbone delay is too large, the queues of the AP will not get refilled in time to maintain it constantly backlogged. In particular, the AP will run out of packets if $\frac{D}{\bar{C}} > \frac{K F_s W_{\max}}{S_{\text{down}}}$. Moreover, to be sure that the AP sends S_{down} packets in each channel access, we assume that at least $g \cdot S_{\text{down}}$ packets have to be stored in its buffers, where $g \geq 1$ is a small constant playing the role of a guard factor (in our experiments, we set $g = 2$). If there are not enough packets in the system to fill the pipe and guarantee enough backlog in the AP, we simply assume that the AP remains completely idle for some time. Specifically, we consider the AP to be fully backlogged for a fraction of time $\frac{K F_s W_{\max}}{(g + \frac{D}{\bar{C}}) S_{\text{down}}}$, if this fraction is smaller than one.

The final formula for the throughput, valid whenever $S_{\text{down}} \leq S_{\text{up}}$, $K F_s W_{\max} \gg S_{\text{down}}$, becomes:

$$\Lambda = \frac{S_{\text{down}}}{1/\mu + T_{\text{down}} + k^* T_{\text{up}}} \cdot \min\left(1, \frac{K F_s W_{\max}}{\left(g + \frac{D}{\bar{C}}\right) S_{\text{down}}}\right) \tag{2}$$

Fig. 4 compares simulation results (blue, thick lines) against analytical prediction (2) (red, thin lines) in our reference system, as we vary the aggregation level employed by all nodes, for different values of backbone delay D. We do not show confidence intervals for simulation results since they are too narrow (at 95% level) to be visible.

Note that, with $B_{\text{AP}} = B_{\text{STA}}$, we are in the *downlink bottleneck* regime. As expected, the model is less accurate when

[4] The value of k^* computed in this way is, in general, not an integer, but we do not have to worry about this.

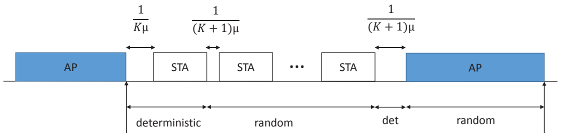


Fig. 5. Cycle analysis for the uplink bottleneck regime with $D = 0$.

D comes into play, or (for $D = 0$) when the assumption $KF_sW_{\max} \gg S_{\text{down}}$ (which here reads $800 \gg 4B_{\text{AP}}$) does not hold. Interestingly, there is an optimal aggregation level (strongly related to $F_sW_{\max}$) which maximizes throughput. This can be explained by the fact that, as we push B_{AP} close to $F_sW_{\max}$, we obtain diminishing returns from amortizing the overhead of setting up MU-MIMO, while increasing the probability that the AP completely empties one of its MAC queues, resulting is lower multiplexing gain. Unfortunately, such kind of optimization of the aggregation level requires knowledge of $F_sW_{\max}$, and can hardly be done in practice.

B. *Uplink bottleneck regime*

Recall that in this case we assume the stations to be always fully backlogged. In this paper, we will analyze this regime under two additional assumptions[5]: i) the backbone delay $D = 0$; ii) the AP completely empties its queues when it gets access on the channel. Assumption i) can represent the network scenario in which servers are located within the same LAN of the stations. Assumption ii) holds in the uplink bottleneck regime when $N_{\text{AP}} \geq K$.

The main difficulty of the analysis lies in the fact that now the AP, differently from the downlink bottleneck regime, is not fully backlogged, thus it typically aggregates only a limited number of packets, which can severely degrade the maximum theoretical throughput computed under saturation conditions.

Recall that the channel holding time $A(h, b)$ of the AP depends on both the number of non-empty queues h in the AP (hereinafter called user diversity) and their maximum backlog b. Let H be the random variable denoting the user diversity, and B the maximum queue length among the AP queues. Let $P(h, b) = \mathbb{P}[H = h, B = b]$ be the joint discrete distribution of the above two variables at the time instant at which the AP gets access on the channel. Note that since the AP has contended for channel access, we have $h \in \{1, \ldots, K\}$, $b \in \{1, \ldots, F_sW_{\max}\}$.

Suppose, for now, that $P(h, b)$ is known. In Appendix A we show how $P(h, b)$ can be analytically computed. The aggregate system throughput can then be derived by a simple cycle analysis, illustrated in Fig. 5.

This time we consider cycles delimited by time instants at which the AP releases the channel. Since the AP flushes out all its backlog, any cycle starts deterministically with a contention phase among K backlogged stations, of average duration $\frac{1}{K\mu}$, followed by the transmission of the winning station, of duration T_{up}. Now, since the backbone delay is zero, the ACKs sent up by this station will immediately create new data packet(s) in the AP, which will start contending as well. Before the AP will eventually win the contention, a random number of stations will be able to transmit. Actually, on average each station will be able to put one transmission on the channel before the AP wins. This result derives from the assumption that backoffs are exponential: by conditioning on the value x extracted by the AP, the number of transmissions made by a station is Poisson distributed of parameter μx. Deconditioning w.r.t. x, we obtain that on average each station makes one transmission before the AP, of duration T_{up}, preceded by a contention period of average duration $\frac{1}{(K+1)\mu}$.

The cycle ends deterministically with another contention period of average duration $\frac{1}{(K+1)\mu}$ (the one won by the AP) followed by the channel holding time by the AP, whose average duration is $\sum_{h,b} P(h, b)A(h, b)$. To compute the average number of packets sent in a cycle, it is convenient to express this number in ACKs, rather then data packets, since we have already shown that on average we see K transmissions by the set of all stations, plus the deterministic transmission at the beginning of the cycle.

Putting everything together, the usual renewal formula (1) provides the throughput for this scenario:

$$\Lambda = \frac{(K+1)S_{\text{sta}}}{\frac{1}{K\mu} + (K+1)\left(\frac{1}{(K+1)\mu} + T_{\text{up}}\right) + \sum_{h,b} P(h, b)A(h, b)} \tag{3}$$

To get insights into the resulting system performance, we compute here the marginal user diversity distribution $P(h) = \mathbb{P}[H = h]$ through an alternative method that does not require us to first derive the joint distribution $P(h, b)$. This computation leads indeed to a rather simple and instructive result that we will discuss later on.

We first isolate the impact of the initial deterministic ACK, computing the user diversity distribution $\hat{P}(h)$ produced by stations' transmissions following the first one. By conditioning on the backoff value x extracted by the AP, we can write:

$$\hat{P}(h) = \int_0^\infty \binom{K}{h} \left(1 - e^{-\mu x}\right)^h e^{-\mu x(K-h)} \mu e^{-\mu x} \, \mathrm{d}x$$

Integrating by parts, we get

$$\hat{P}(h) = \int_0^\infty \binom{K}{h} \frac{h}{K-h+1} (1 - e^{-\mu x})^{h-1} e^{-(K-(h-1))\mu x} \mu e^{-\mu x} \, \mathrm{d}x$$

Noticing now that $\binom{K}{h}\frac{h}{K-h+1} = \binom{K}{h-1}$, the above expression means that $\hat{P}(h) = \hat{P}(h-1)$. In other words, the distribution of $\hat{P}(h)$ is uniform over $h = 0, 1, \ldots, K$, hence $\hat{P}(h) = \frac{1}{K+1}$.

To compute the distribution $P(h)$, that includes the contribution of the first ACK, we observe that $H = h$ occurs in two possible ways: i) either the first ACK belongs to one of the h queues which are already non-empty for effect of subsequent transmissions of the stations, with probability $\frac{h}{K}$, or it increases by one the number $h - 1$ of non-empty queues produced by the other transmissions, with probability $\frac{K-(h-1)}{K}$. We obtain:

$$P(h) = \hat{P}(h)\frac{h}{K} + \hat{P}(h-1)\frac{K-(h-1)}{K} = \frac{1}{K}$$

meaning that $P(h)$ is also uniform over the set of possible values $h = 1, 2, \ldots, K$.

[5] Relaxing either of these two assumptions is analytically challenging, and we leave it to future work.

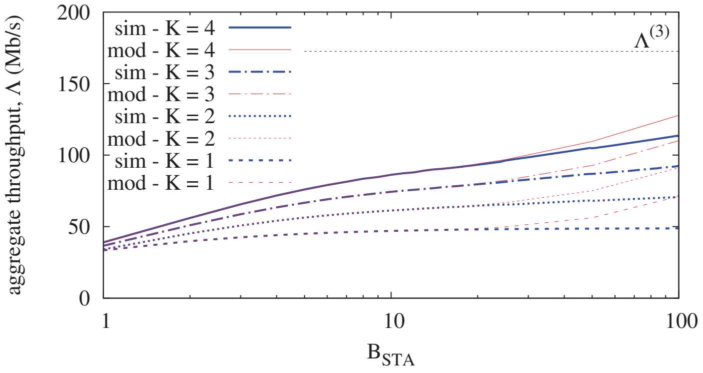


Fig. 6. Throughput comparison (model vs simulation) in the reference system, with $B_{\text{AP}} = \infty$, $D = 0$, as function of B_{STA}, for different number of stations.

This result has striking consequences on the efficiency of MU-MIMO, which strongly relies, in addition to the availability of large per-station backlog (to amortize the overhead), on large user diversity (i.e. multiplexing gain). Note that the optimal operating point of MU-MIMO is full diversity ($h \geq N_{\text{AP}}$), which naturally occurs in the downlink bottleneck regime.

In the uplink bottleneck regime, instead, wireless channel contention can result into a random user-diversity far from the optimal one. Under the scenario considered in this section ($K \leq N_{\text{AP}}$), the average user diversity is $(K+1)/2 \leq K$, which results roughly into a throughput reduction by factor $(K+1)/(2K)$, which in our reference system (with $K = 4$) equals 5/8 = 0.625. Note that the penalty introduced by such sub-optimal user-diversity is intrinsic to the random access nature of the channel, and thus unavoidable (in the uplink bottleneck regime). Instead, the overhead required to set up MU-MIMO in the downlink can be amortized by letting the stations perform packet aggregation in a way similar to what the AP does.

Fig. 6 compares the analytical prediction (3) against simulation in our reference system (with $D = 0$), as we vary the aggregation level B_{STA} and the number K of stations. Here we assume unlimited aggregation by the AP (actually, $B_{\text{AP}} \geq F_s W_{\max}$), bringing to system to operate in the *uplink bottleneck* regime. As expected, the model is less accurate when the assumption $F_s W_{\max} \gg S_{\text{sta}}$ (which here reads $200 \gg 2B_{\text{STA}}$) does not hold.

Focusing on the case $K = 4$, we observe severe throughput loss when B_{STA} is small, due to poor frame aggregation by the AP.[6] But even with unlimited aggregation by the stations (actually, the maximum level of aggregation by stations is already achieved with $B_{\text{STA}} = 100$) the throughput is only about 113 Mb/s[7], which is $0.65 \cdot \Lambda^{(3)}$ (see Sec. III-E), close to our analytical prediction of a throughput reduction by factor 0.625.

C. Full aggregation regime

Recall that in this case both the AP and the stations perform a large enough packet aggregation to completely empty their buffers at each channel access. Since the current 802.11 standards allow to adopt large levels of aggregation (around 1 MB), possibly larger than the TCP maximum window size, we believe this regime to be quite important in practice.

The main effect produced by large aggregation performed by both AP and stations is the following: all packets circulating in the system, and associated to the same station (under our assumptions, $F_s W_{\max}$ packets) cluster together and move as a single entity (a large batch) across the network. Note that this phenomenon does not depend on initial conditions nor on the value of backbone delay.

The above behavior allows us to develop a simpler analytical model than that in Sec. IV-B, accounting also for backbone delay. We start analyzing the case of $D = 0$. We adopt the same cycle analysis illustrated in Fig. 5. This time, however, we can have at most one transmission by each station in between two consecutive transmissions by the AP. Actually, we can directly exploit the computation of $P(h)$ done in Sec. IV-B, and conclude that the number of transmissions performed by stations in a cycle has the uniform distribution over $1, \ldots, K$. Let T_{up} be the time required by a station to send $F_s W_{\max}$ (effective) ACKs in the uplink. The usual renewal formula (1) provides in this case:

$$\Lambda = \frac{\sum_{h=1}^{K} \frac{1}{K} h F_s W_{\max}}{\frac{1}{\mu K} + \sum_{h=1}^{K} \frac{1}{K} \left(A(h, F_s W_{\max}) + h T_{\text{up}} + \sum_{j=0}^{h-1} \frac{1}{\mu(K-j)} \right)} \qquad (4)$$

Let us now consider a scenario in which the backbone delay is extremely small, but larger than the maximum channel contention time (i.e., slightly larger than $W_0 \sigma$). It happens here that the last batch of ACKs sent up by a station in a cycle cannot arrive at the AP in time to be immediately resent down in the following AP transmission (marking the end of the current cycle). Therefore, this last batch will be aggregated with those sent by the AP at the end of the next cycle. One important consequence of this fact is that, with non-zero delay, we never see the maximum value ($h = K$) of user diversity. This explains the sharp initial drop that we observe in the throughput as we step out of $D = 0$ (see Fig. 7).

One can actually compute the throughput in the case of small delay accounting for the fact that the last uplink batch is always sent down in the next cycle, through the formula (see [10]):

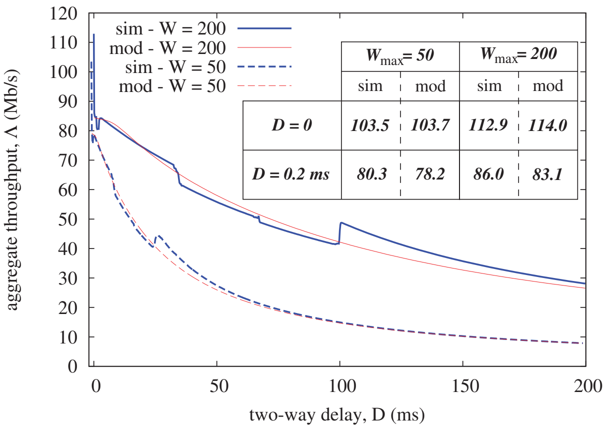
$$\Lambda = \frac{\sum_{h=0}^{K-1} \frac{1}{K} \max(1,h) F_s W_{\max}}{\sum_{h=0}^{K-1} \frac{1}{K} \left(A(\max(1,h), F_s W_{\max}) + h T_{\text{up}} + \sum_{j=0}^{h} \frac{1}{\mu(K-j)} \right)} \qquad (5)$$

leading to a throughput reduction roughly equal to $\frac{K^2-K+2}{2K^2}$.

To compute the throughput in the case of larger delays, we adopt a useful approximation which consists of assuming that the network delay D is exponentially distributed (instead of deterministic). Such approximation greatly simplifies the analysis, while providing an accurate throughput prediction. Indeed, the memoryless property of the exponential distribution allows us to embed a discrete-time Markov Chain at the boundaries of the cycles as in Fig. 5, with a bi-dimensional state (m_1, m_2) denoting (assuming $K > 1$): the number $1 \leq m_1 < K$ of batches

[6] without the delayed ACK option, with $B_{\text{STA}} = 1$ we would get $\Lambda = 23.9$ Mb/s, smaller than that of a DL SU system! (see Sec. V).

[7] This value requires exactly $D = 0$. Under more realistic conditions of small but not null delay, we would get $\Lambda = 86$ Mb/s, see Sec. IV-C.



	W_{max}= 50		W_{max}= 200	
	sim	mod	sim	mod
D = 0	103.5	103.7	112.9	114.0
D = 0.2 ms	80.3	78.2	86.0	83.1

Fig. 7. Throughput comparison (model vs simulation) in the reference system, under the full aggregation regime ($B_{\text{AP}} \geq W_{\max}$, $B_{\text{STA}} \geq W_{\max}$), for $W_{\max} = 50$ or 200, as function of backbone delay D.

transmitted by the AP at the end of previous cycle (recall that this number cannot be equal to K, with non-zero delay); the number $0 \leq m_2 \leq K - m_1$ of batches stored by the stations. Then the remaining batches $m_3 = K - m_1 - m_2$ are still 'flying' in the backbone, with remaining time to arrive at the AP exponentially distributed with mean D. Note that the total number of states, equal to $\frac{K^2+K-2}{2}$, is typically small (in the order of K^2).

We can easily express the transition probabilities among the above defined states, and use the stationary distribution of the Markov Chain to compute the throughput according to (1) (details can be found in [10]).

Fig. 7 compares analytical predictions obtained by our Markov Chain model against simulation as we vary the backbone delay D, for two different values of TCP maximum window size $W_{\max} = 50$ or 200. We observe that the analytical predictions (based on the exponential delay assumption) nicely interpolate the rather complex curves obtained from simulation under deterministic delay.

The table inserted on the plot also shows the accuracy of (4) (for $D = 0$) and (5) (for small but non null delay). Results for the latter (more realistic) case confirms that no more than 86 Mb/s can be achieved by full aggregation in the reference system with $W_{\max} = 200$, which is 50% of bound $\Lambda^{(3)} = 172.5$ Mb/s, as roughly predicted by factor $\frac{K^2-K+2}{2K^2}$, equal to 44%, with $K = 4$.

V. Comparison of uplink strategies

Since the traditional random access mechanism of 802.11 does not allow us to fully exploit the capacity gain of downlink MU-MIMO under closed-loop traffic, we may ask which alternative schemes (specifically intended for the uplink traffic) could be used to improve the throughput.

We will again focus on our reference system (under the best case $D = 0$), for which theoretical throughput bounds have been already computed in Sec. III-E.

A simple solution to avoid the performance degradation inherent to random channel access is to make the uplink operate under the AP's coordination. Consider, for example, a simple polling mechanism working as follows: right after transmitting down a data bundle, the AP polls each station to which it has transmitted data to send up a corresponding number of packets. Clearly, this scheme allows to achieve bound $\Lambda^{(3)} = 172.5$ Mb/s.

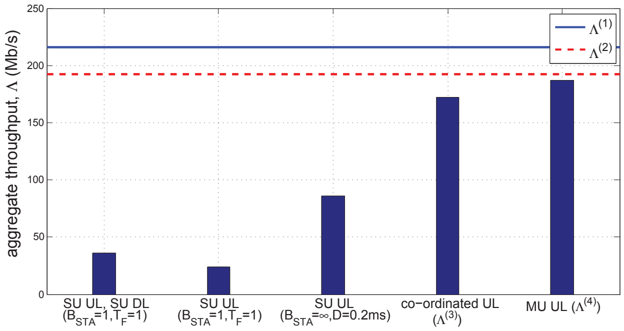


Fig. 8. Comparison of throughputs achieved in the reference system, with $D = 0$, $B_{\text{AP}} = \infty$, under different settings and access strategies.

Note that upper bound $\Lambda^{(2)} = 192.5$ could be approached in a similar way, if stations were also able to send up a single (small) cumulative ack for all data received from the AP. This could actually be obtained at the transport layer by increasing the thinning factor T_F. Note, however, that massive use of delayed ACK techniques (beyond the standard $T_F = 2$) has detrimental effects to TCP [9], and would require sophisticated cross-layer design to be implemented in a WLAN.

At last, we could employ multi-user transmissions also in the uplink (as is expected to be the case with the upcoming 802.11ax). In particular, consider a vanilla MU uplink with zero overhead[8], that allows backlogged stations to aggregate and concurrently send up many packets (TCP ACKs, in our case) in the uplink. Even employing the standard delayed ACK option ($T_F = 2$), such scheme would achieve, with full aggregation, throughput as high as $\Lambda^{(4)} = KF_sW_{\max}/[A(K, F_sW_{\max}) + T_{\text{up}}(F_sW_{\max}/2)] = 187.0$ Mb/s, where $T_{\text{up}}(F_sW_{\max}/2)$ is the channel time to send 100 TCP ACKs, in our case.

Fig. 8 visually compares the throughputs achieved in several interesting cases that we have analyzed and discussed so far, in our reference system (always with unlimited aggregation by the AP). The first bar shows that, in the case of $B_{\text{STA}} = 1$, $T_F = 1$, the throughput that we get by using SU DL is actually larger than what we get by enabling MU DL (second bar)! The third bar (related to the full aggregation regime) shows the huge throughput loss (around 50%) intrinsically due to random channel contention. The last two bars are related to the alternative uplink strategies discussed in this section.

VI. Related Work

The capacity gain of MU-MIMO has been widely investigated at the PHY layer, considering various schemes to acquire CSI

[8] Similar to DL MU-MIMO, a multi-user uplink transmission also requires some overhead to set up communication [16]. While a multi-user uplink is yet to be standardized in the upcoming 802.11ax standards, prior works such as [17] have demonstrated schemes to reduce this uplink overhead to as little as 100 μs which is approximately 10 times less as compared to the sounding overhead for DL MU-MIMO.

and different precoding techniques to enable simultaneous data transmission (e.g., [11], [12]). However, the impact of traffic dynamics on the achievable throughput performance is still not well understood.

MAC protocols [13], [14] that exploit the higher transmission capabilities of the advanced MU-MIMO PHY layer have been designed and evaluated with over-the-air experiments. In [15] authors propose a queueing model for MU-MIMO under open loop (non-saturated) traffic. Various user scheduling algorithms for poor channel quality avoidance are analyzed in [18]. However, crucial assumptions made in the papers above is that the AP is always fully backlogged, or that traffic is open loop only.

There exists a huge body of literature on modeling 802.11 (i.e., variations of [7]), considering also the impact of closed-loop traffic (TCP) (e.g., [8]). However, to the best of our knowledge, no work has explored so far the performance of MU-MIMO under closed-loop traffic and 802.11 contention.

VII. Conclusion

We have presented the first cross-layer analysis of an 802.11ac compliant WLAN where DL MU-MIMO is coupled with SU uplink, considering the impact of closed-loop (TCP) traffic. Despite the fact that the majority of traffic volume flows downlink, our analysis has revealed the emergence of a dichotomy between a downlink bottleneck regime and an uplink bottleneck regime, depending on several parameters such as number of stations/antennas, frame aggregation levels, thinning of feedback traffic. With the help of our analytical models we have identified crucial performance factors that offset the gains achievable by DL MU-MIMO, showing the intrinsic limitations due to random channel contention. We have also taken a system design view discussing strategies to mitigate this loss and allow MU-MIMO WLANs to achieve their theoretical capacity under closed loop traffic.

Appendix A
Computation of joint distribution $P(h, b)$

We can limit ourselves to the case in which stations send just one effective TCP ACK in each channel access ($S_{\text{sta}} = 1$). The extension to the case in which stations send $S_{\text{sta}} > 1$ effective ACKs in each channel access is trivial, since it just requires to scale the distribution obtained for $S_{\text{sta}} = 1$ accordingly.

To obtain an exact expression of $P(h, b)$ in the case of zero-delay backbone, we separately account for the impact of the initial deterministic ACK at the beginning of a cycle (see Fig. 5). So, let us first consider the distribution produced by uplink transmissions following the first one. For them, we actually compute the more detailed joint pdf $\hat{P}(h_1, h_2, b)$ where: b is the maximum queue length; $h_1 \geq 1$ is the number of queues having exactly length b; $h_2 \geq 0$ is the number of queues having length strictly less than b. By conditioning on the backoff value x extracted by the AP, we can write:

$$\hat{P}(h_1, h_2, b) = \int_0^\infty \binom{K}{h_1} \left[\frac{(\mu x)^b}{b!} e^{-\mu x}\right]^{h_1} \cdot$$
$$\binom{K-h_1}{h_2} \left(\sum_{j=1}^{b-1} \frac{(\mu x)^j}{j!} e^{-\mu x}\right)^{h_2} e^{-\mu x (K-h_1-h_2)} \mu e^{-\mu x} \, \mathrm{d}x \qquad (6)$$

Despite their ugly look, integrals of the form (6) have a closed-form expression, obtained by expanding them into a sum of contributions, each leading to an analytical solution. Just as an example, in the case of $K = 4$,

$$\hat{P}(2, 1, 3) = \binom{4}{2}\binom{2}{1} \frac{1}{(3!)^2} \left(\frac{7!}{1!\,5^8} + \frac{8!}{2!\,5^9}\right) = \frac{3024}{390625}$$

Note that $\hat{P}(h_1, h_2, b)$ are some 'universal' numbers that depend only on K, and that can be computed once and forall and made available through, e.g., a table lookup.

To derive the final joint pdf $P(h, b)$ we have to add the contribution of the first deterministic ACK:

$$P(h, b) = \sum_{h_1+h_2=h-1} P(h_1, h_2, b) \frac{K-h+1}{K} +$$
$$\sum_{h_1+h_2=h} P(h_1, h_2, b-1) \frac{h_1}{K} + \sum_{h_1+h_2=h} P(h_1, h_2, b) \frac{h_2}{K} \qquad (7)$$

References

[1] IEEE 802.11ac/D7.0, "Enhancements for Very High Throughput for Operation in Bands Below 6 GHz," 2013.

[2] O. Bejarano, E. W. Knightly and M. Park, "IEEE 802.11ac: from channelization to multi-user MIMO," *IEEE Communications Magazine*, vol. 51, no. 10, pg. 84-90, 2013.

[3] D. Murray and T. Koziniec, "The state of enterprise network traffic in 2012," *In Proc. IEEE APCC*, pg. 179-184, 2012.

[4] R. Braden, "RFC-1122: Requirements for internet hosts," *Request for Comments*, pg. 356-363, 1989.

[5] Perahia, Eldad and Stacey, Robert, "Next Generation Wireless LANS: 802.11n and 802.11ac," *Cambridge university press*, 2013.

[6] ns3, https://www.nsnam.org/, Accessed: 2016-07-28.

[7] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC*, 18(3), pg. 535–547, 2000.

[8] A. Kumar, et al.,"New Insights from a Fixed-point Analysis of Single Cell IEEE 802.11 WLANs," *IEEE/ACM Trans. Netw.*, 15(3), pg. 588-601, 2007.

[9] D. Miorandi, A. A. Kherani and E. Altman, "A Queueing Model for HTTP Traffic over IEEE 802.11 WLANs," *Comput. Netw.*, 50(1), pg. 63-79, 2006.

[10] Rice University ECE Technical Report TREE1701, January 2017.

[11] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE JSAC*, 24(3), pg. 528-541, 2006.

[12] G. Caire N. Jindal M. Kobayashi and N. Ravindran, "Multiuser MIMO achievable rates with downlink training and channel state feedback," *IEEE Trans. on Inf. Theory*, 56(6), pg. 2845-2866, 2010.

[13] N. Anand, et al., "Mode and user selection for multi-user MIMO WLANs without CSI," *In Proc. IEEE INFOCOM*, pg. 451-459, 2015.

[14] O. Bejarano, et al., "MUTE: sounding inhibition for MU-MIMO WLANs," *In Proc. IEEE SECON*, pg. 135-143, 2014.

[15] B. Bellalta, V. Daza and M. Oliver, "An approximate queueing model for multi-rate multi-user MIMO systems," *IEEE Communications Letters*, 15(4), pg. 392-394, 2011.

[16] K. Tan, et al., "SAM: enabling practical spatial multiple access in wireless LAN," *In Proc. ACM Mobicom*, pg. 49-60, 2009.

[17] A. Flores, S. Quadri and E. W. Knightly, "A Scalable Multi-User Uplink for Wi-Fi," *In Proc. 13th USENIX NSDI*, pg. 179-191, 2016.

[18] Chiung-jang Chen and Li-chun Wang, "Performance Analysis of Scheduling in Multiuser MIMO Systems with Zero-Forcing Receivers," *IEEE JSAC*, 25(7), pg. 1435–1445, 2007.