Overcoming Performance Pitfalls in Rate-Diverse High Speed WLANs

Martín Zubeldía*, Andrés Ferragut, Fernando Paganini

Universidad ORT Uruguay, Montevideo, Uruguay

Abstract

Recent developments on the IEEE 802.11 family of standards promise significant increases in speed by incorporating multiple enhancements at the physical layer. These high modulation speeds apply to the data portion of the transmitted frames, while headers must remain at lower speeds; this has motivated the use of *frame aggregation* to increase data payloads in the newer standards. However, this simple method may still utterly fail to deliver the promised speeds, due to a series of cross-layer effects involving the transport and multiple access layers: the downward equalization of throughputs imposed by TCP under physical rate diversity, the excessive impact of the TCP ACK stream, or the unreasonable fraction of access opportunities taken by uplink flows when competing with the more numerous downlink connections. A first contribution of this paper is to demonstrate these impediments and isolate their causes through a series of experiments with the ns3 packet simulator, on the 802.11n and 802.11ac protocol versions.

Our analysis leads us to propose a desirable resource allocation for situations of rate-diverse competition, and an architecture for control at the access-point to achieve it. Our implementation is compatible with the standard, involving a combination of known techniques: packet aggregation, multiple queues with TCP-ACK isolation, and control of the MAC contention window. The main contribution here is to provide a practical, comprehensive solution that imposes the desired efficiency and fairness model addressing all the previously indicated limitations. We demonstrate analytically and through extensive simulation that our method is able to provide significant enhancements in performance under a variety of traffic conditions.

Keywords: IEEE 802.11, Rate-diversity, Cross-layer analysis, Performance evaluation

1. Introduction

Wireless local area networks (WLANs) based on IEEE 802.11 [1] are present in nearly every networking deployment around the world. WLAN hotspots are shared by multiple users at a time through Medium Access Control (MAC) protocols, and newer versions of the standard have progressively upgraded the available physical channel speeds.

For instance, in the IEEE 802.11n version [2], many new enhancements in modulation and transmission techniques (OFDM, MIMO) have been incorporated to allow stations to transmit at rates reaching 600 Mbps. In future releases such as IEEE 802.11ac [3], modulation speeds are expected to grow up to 7 Gbps, a 10-fold increase. It is clear, however, that these higher data rates are only achievable in the best channel conditions, and thus stations are allowed to transmit at lower data rates if necessary to reduce frame transmission errors. The net effect of this adaptation is that multiple users with diverse data rates coexist in the same cell. This fact is not taken into account in the medium access control layer, where typically the Distributed Coordination Function (DCF) mechanism is used for channel access. This mechanism provides roughly equal access opportunities to all stations, regardless of their physical rate; as we demonstrate below, this leads to severe inefficiency. Channel access differentiation is allowed in the Enhanced Distributed Channel Access (EDCA) function of the standard, but its intended use is to differ-

*Corresponding Author

Email address: zubeldia@ort.edu.uy

Postal address: Cuareim 1451, 11.100

Tel./Fax: (+598) 29021505 / 29081370

entiate traffic classes, not individual station data rates.

Physical rate diversity also affects the efficiency impact of protocol overheads. In particular, a Physical Layer Convergence Protocol (PLCP) is required to provide synchronization and indicate the data rate of the forthcoming frame. This header, which must be sent at the basic (lowest) data rate, can occupy a significant amount of time in comparison to the data frame at a high physical rate. To mitigate this, recent standards have enabled *packet aggregation*, in which a single channel access by a station is used to transmit multiple higher layer packets, whether in a single frame (A-MSDU) or in multiple contiguous frames (A-MPDU). Frame aggregation is known to achieve almost 100% channel utilizations in point to point communications. However, the use of frame aggregation in rate diverse environments, as well as the implications it has on higher layer protocols has received less attention. We review some of the previous work on the subject in Section 2.

The purpose of this paper is to explore the effect of packet aggregation and rate diversity on TCP connections, explaining the often dismal performance, and proposing a comprehensive solution. We begin in Section 3 by identifying various reasons why the packet aggregation mechanisms alone may fail to deliver the promised speeds: lack of proper attention to the bidirectional nature of TCP; inefficient allocation of transmission opportunities between rate-diverse stations sharing a common queue; asymmetric competition between uplink and downlink flows. Our packet simulations exhibit some striking inefficiencies in the use of the wireless medium, which are due to features of the 802.11 family in all its versions (a, g, n, ac), but have more severe impact at high modulation speeds.

In Section 4 we describe our proposal to overcome these limitations. We first argue for what we believe is the proper assignment for rate diverse cells, a proportionally fair allocation between data flow rates. This allocation has been postulated several times in the wired [4] and cellular [5] cases, and is associated (see [6]) with the fair distribution of channel time [7]. Next, we proceed to describe an architecture that combines queueing, packet aggregation, and control of contention windows to achieve this allocation. The proposed architecture may be implemented at the access point, relying only on locally available information, does not require substantial modifications at the stations, and is well suited to current and future versions of the standard. The method is initially developed for the downlink case, but later extended to mixed downlink-uplink traffic scenarios, still based on control at the AP. We validate its performance through packet-level simulations with TCP connections, initially taken to be permanent.

In Section 5 we consider the more realistic traffic scenario of a varying number of connections under a stochastic model for traffic demand. We provide a theoretical model for predicting flow level throughputs, and show that the proposed algorithm enables a flow level throughput allocation that is both efficient and robust to different job size statistics.

We give conclusions in Section 6. Partial results included in this paper were presented in [8].

2. Related work

The performance analysis of 802.11 cells was pioneered by [9], who used a Markov chain analysis of the collisions/backoff process in the DCF algorithm to evaluate the throughput achieved by uplink stations in a single-rate cell. This analysis was extended in [10], in particular to rate diverse cells; noting that the DCF resource allocation is largely oblivious to the modulation rate, a downward equalization of effective rates is found to occur. [11] extended Markov-based models to include the rate adaptation (ARF) mechanisms in multi-rate LANs, and a model of TCP connections over it. It is recognized in many works that equal distribution of channel *times* avoids the aforementioned inefficiency, and a variety of mechanisms have been proposed to achieve it (see [7] and references therein).

Frame aggregation as an overhead reduction scheme was analyzed empirically in [12], before these mechanisms were defined in the 802.11n standard (see [13] for an overview of these features). Markov-based models that incorporate this feature are given in [14]. For a study of aggregation in a multi-hop environment see [15]. Methods of aggregation in a lossy environment, where fragments must be retransmitted are proposed in [16, 17].

The unfairness between uplink and downlink TCP connections, due to the asymmetric impact of losses for data packets and ACKs, has been studied in [18, 19], and proposals are given to address the issues based on (real or virtual) differentiated queues, and/or control of the DCF contention window parameter.

While our paper revisits many of the issues raised in these previous works, and also employs similar control mechanisms (aggregation, differentiated queueing, backoff window adaptation), it is to our knowledge the first to propose a comprehensive solution to the issues of efficiency and fairness for TCP flows under rate diversity in both downlink and uplink scenarios.

Parameter	Value
Slot time	$9\mu s$
SIFS	$16\mu s$
DIFS	$43\mu s$
PLCP Header	$32\mu s$
PHY rates	$\{6.5, \ldots 65\}$ Mbps (802.11n)
	$\{58.5, \ldots 780\}$ Mbps (802.11ac)
CW_{min}	15
CW_{max}	1023

Table 1: IEEE 802.11 parameters and physical layer rates.

The flow level performance of wireless cells is thoroughly analyzed in [20], as well as in our previous papers [21, 6].

3. Inefficiencies in 802.11 cells with TCP and packet aggregation

The purpose of this Section is to explore by simulation the effects of packet aggregation on the throughput achieved by the newest versions of the IEEE 802.11 standard. All simulations were performed in the network simulator `ns3` [22], which we modified to include the 802.11n and 802.11ac physical layer rates, as well as all the other MAC behavior and time parameters included in the standard. We focus here mainly on non-MIMO channels, due to simulator limitations; nevertheless as we shall see, the main conclusions of these experiments do not depend on the explicit physical rates involved. The relevant parameters are summarized Table 1. All simulations involve a single cell consisting of an Access Point (AP) and one or several client stations (STAs). Furthermore, all simulations use a noisy channel with Rayleigh fading and the STAs use the AARF rate adaptation mechanism. Rate diversity is achieved by placing the STAs at different distances from the AP so as to get different PHY rates.

3.1. Aggregating UDP frames in the AP

Consider first a single transmission in the downlink sense. A UDP traffic source, with enough bitrate to saturate the channel, is directly connected to the AP and transmits over the wireless link to the STA. The source generates packets of standard length $L = 1500$ bytes, that are aggregated in the AP using A-MPDU, which enables frames of size up to $64KB$ in 802.11n, or $1MB$ in 802.11ac. Taking into account the protocol overheads, the effective transmission time when aggregating n packets is given by:

$$T_n^0 = DIFS + H + \frac{nL}{PHY} + SIFS + H + \frac{L_{mac}}{PHY},$$

where H is the time to transmit the PLCP header, L is the packet size, PHY is the modulation rate and L_{mac} is the MAC layer ACK length.[1] Before each packet, the AP performs a random backoff stage. The time spent in this stage greatly depends on the collision probabilty, which is zero because all packets are sent from the AP when using UDP. Furthermore, the retransmissions due to channel errors are reduced thanks to the A-MPDU, that isolates the packet which has errors and only retransmits that one. As a result, this time can be estimated by $\overline{T}_{bo} = \frac{CW_{min}}{2} T_{slot}$. The average throughput is then given by:

$$Thr = \frac{nL}{T_n^0 + \overline{T}_{bo}} = \frac{nL}{\frac{nL}{PHY} + const}.$$

As the aggregation factor n increases in the above formula, we see that the expected throughput improves, reducing the impact of the fixed time overheads; it should eventually approach the value of the PHY rate. Let us test this fact by simulation with an uncontrolled UDP traffic source that generates packets at the PHY rate. To make the test in the worst-case scenarios, the PHY rate of the station is fixed at 65 Mbps in IEEE 802.11n and 780Mbps for 802.11ac, where overheads are more relevant. Results are shown in Figure 1 where it can be seen that aggregation indeed has the predicted impact on the UDP throughput.

3.2. Aggregating TCP frames in the AP

Now we consider a single transmission in the downlink sense as before, but with a TCP source, which regulates its transmission rate based on a stream of TCP ACKs. This modifies the throughput performance with respect to the UDP case, in two different ways.

The first, simpler reason is that the channel medium must accommodate ACK packets in the uplink sense[2], and although these are small (40 bytes), they still require time for transmission of wireless overheads at the basic rate. Indeed, this leads to a loss in efficiency of TCP with respect to UDP even in the case of no packet

[1] We assume L_{mac} fixed for ease of exposition, in practice there are minimal differences between ACKs due to the use of Block ACKs.

[2] In our simulations, one TCP ACK is generated for each packet, to simplify the analysis of TCP effects. Typical TCP implementations also send one ACK every two packets, and our results can be adapted to that situation.

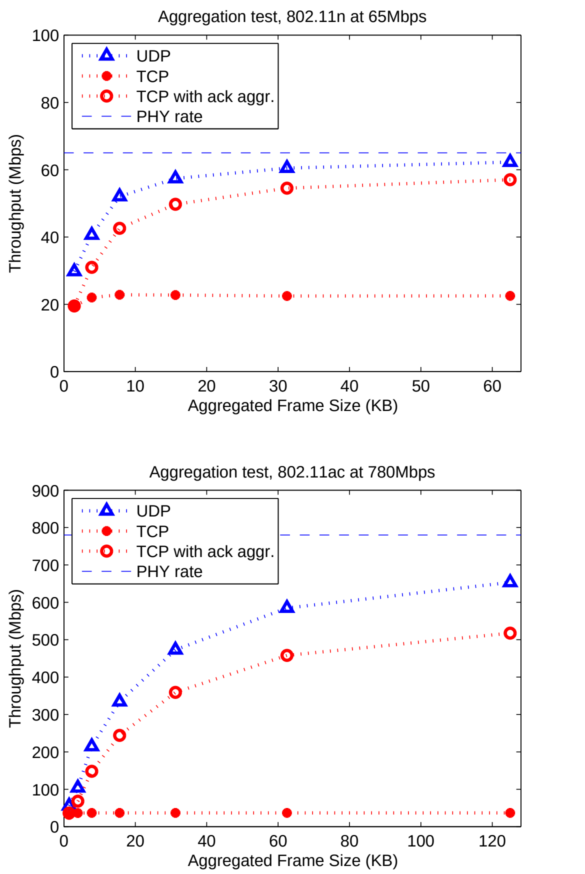

Figure 1: Effect of the aggregation on UDP and TCP throughput at maximum *PHY* rates. Above: 802.11n. Below: 802.11ac.

aggregation (see [6]). The second effect stems from the ACK-clocking of TCP transmissions, that impacts the number of packets that the source makes available at the AP queue, a determinant factor in whether frame aggregation will be successful.

To investigate this second issue we first consider the case when there is no aggregation in the station that is sending the TCP ACKs. The points marked "TCP" in the graphs of Figure 1 show the throughput obtained in this situation, as a function of the aggregation factor. We see that the TCP flow benefits far less from packet aggregation than the UDP flow, reaching a maximum efficiency of 35% at 65 Mbps, and only 5% at 780 Mbps.

The reason for this severe inefficiency is the following: for every aggregate frame sent with n packets, the receiver generates n TCP ACKs; since the STA performs no aggregation, sustaining a steady flow of aggregate frames would require n channel accesses of the STA for every AP access. But the DCF mechanism gives the AP and the STA equal channel access opportunities whenever there are packets in both queues. The net result is that the AP queue empties, while the TCP flow waits for the uplink ACKs to generate replacement packets. In Figure 2 we plot the AP queue in 802.11n for an A-MPDU limit of $64K$, and the STA queue of TCP ACKs. Note that most of the time only one or two packets can be aggregated at the AP, and the congestion is experienced by the TCP ACKs. The maximum aggregation is not achieved, with the throughput saturating well below the maximum. In the Appendix we provide a Markov chain analysis that matches this observed behavior.

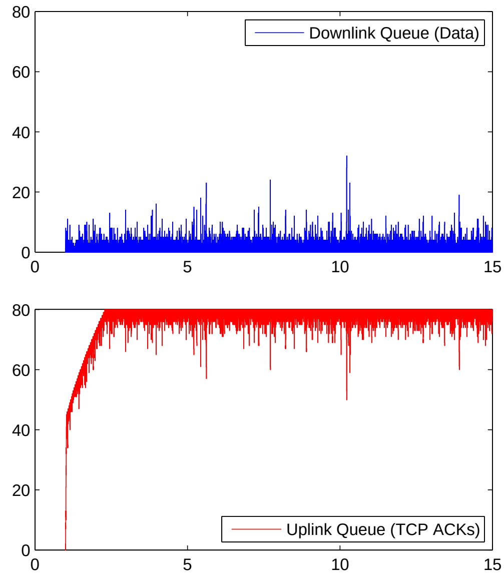

Figure 2: AP (above) and STA (below) queue for a single TCP flow with aggregation in the AP.

This problem can be readily solved by enabling aggregation on the STA, such that the TCP ACKs also get bundled in a single MAC frame, and thus require only one channel access to be transmitted. Once aggregation is enabled in the STA, the throughput increases considerably as can also be seen in Figure 1. At maximum aggregation, TCP improves its efficiency, and the remaining difference with UDP is only due to the TCP ACKs transmission time.

The main conclusion of this analysis is that *aggregation must be implemented in both directions to give real benefits*.

3.3. Rate-diverse cells and differentiated aggregation

An effect already observed (c.f. [10, 6]) in WiFi cells is that slow stations slow down the whole network. This

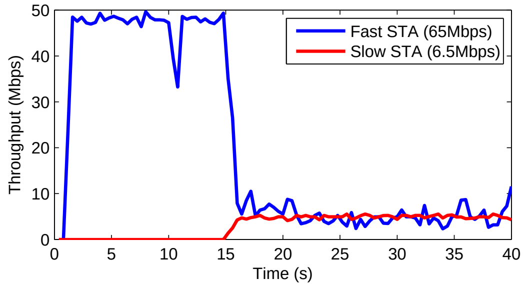

Figure 3: Throughput equalization under rate diversity due to common downlink queue. Aggregation is in use in both STAs.

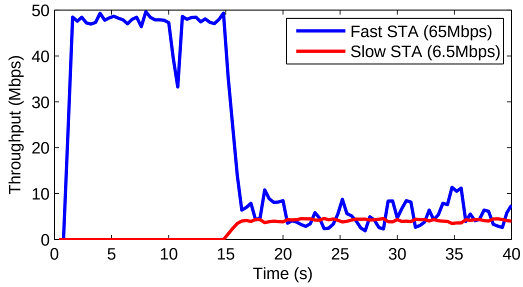

Figure 4: Throughput equalization under rate diversity due to common downlink queue. Rates are equalized despite aggregation is in use only at the fast STA.

is explained as follows: while transmission of a single frame to a fast station takes a short time, it must subsequently wait for a long time while the slow station is serviced; the net effect is that throughputs get equalized to a level below the physical rate of the slowest station.

A more efficient allocation would involve equalizing the channel allocation *times*, as pursued by various techniques (see [7, 16] and references therein). In this regard, it would seem that packet aggregation provides a very simple tool to address this issue: by enabling differentiated packet aggregation in proportion to the physical rates, one could equalize the effective data transmission times. This could be done by using TxOP and aggregation at the same time or by just aggregating a certain amount of packets, proportional to the PHY rate of their destinatary.

However, when the transport layer with TCP comes into play, a new cross-layer issue appears which we will illustrate with the following example. Consider a scenario with two stations downloading data from the AP. For definiteness, assume they are operating at *PHY* rates 65 and 6.5 Mbps respectively. In the simulation, each STA establishes a single downlink TCP connection, with the slow one entering after some time. Aggregation is enabled in both ways, so TCP ACKs do not become a bottleneck, as discussed before. As depicted in Figure 3, the fast STA suffers greatly from the presence of the slow one, going from a throughput of 50 Mbps when alone to one of less than 6 Mbps in this case.

If we modify the queueing algorithm in the AP queue to aggregate packets *only for the fastest station* in a $10 : 1$ ratio, we would expect the fast station to be protected. However, as shown in Figure 4, aggregation has a very modest influence in correcting this outcome. This happens independently of the aggregation factor used, we refer to [8] for further simulation scenarios.

An explanation can be found in the fact that TCP connections are controlled through packet losses that occur in arrivals to a *single*, common AP queue. Packets of slow and fast flows see the same loss probability when arriving at this queue, and therefore TCP congestion control will roughly equalize the mean congestion windows of both flows [23]. Since rate equals window/round-trip-time, the only chance at throughput differentiation would come from RTT differentiation; some of that is observed, and is consistent with the advantage of "jumping the queue" that the fast station gets when its packets are aggregated, but by no means this can achieve the desired level of throughput differentiation.

The main conclusion of this experiment is that *aggregation alone cannot differentiate throughputs in a rate diverse environment*, due to the closed loop behavior of the TCP protocol.

The situation is totally different if we give each flow a different queue in the AP. The EDCA mechanism in the standard enables us to implement this, although here we establish no class priorities: both queues are given equal access parameters, with aggregation only performed in the fast station. Results are shown in Table 2. The aggregation factor n (ratio between the A-MPDU limit and the base packet length of 1500 bytes) now has a significant impact on the rate allocation: indeed, there is a roughly linear relationship between the relative throughput (between both flows) and the aggregation factor. Clearly, in this last scenario we have found a suitable "knob" to affect the resource allocation; the fair way to use it is discussed in Section 4.

3.4. Competing uplink traffic

A third major issue in 802.11 cells is the resource allocation between downlink and uplink traffic. In typi-

Agg. factor n (Fast STA)	Thr. (65Mbps)	Thr. (6.5Mbps)	Thr. ratio
1	4.43	3.39	1.31
2	8.17	3.55	2.30
5	16.49	2.95	5.59
10	25.77	2.27	11.35
21	34.86	1.60	21.78
43	41.33	1.11	37.23

Table 2: Bandwidth sharing with differentiated aggregation and separated queues.

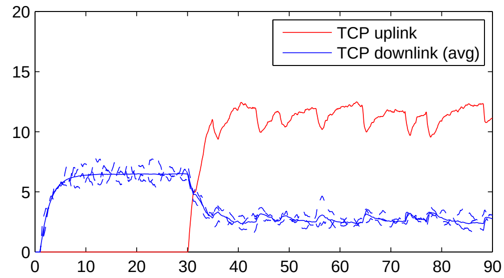

Figure 5: Throughputs for 3 downlink connections and a competing uplink, all stations at 65 Mbps.

cal Internet access settings, most of the traffic is downlink and the AP is serving client stations. However, once uplink traffic is present, the downlink traffic can be severely affected with uplink flows getting an unfair share of the bandwidth. This imbalance between downlink and uplink flows can be explained as a combination of two issues.

To begin with, consider a scenario in which we have multiple STAs downloading data from the AP and a single STA uploading data. In this case there are two queues sending data packets, the AP and the uplink STA, and the DCF does not discriminate between them in the channel access opportunities. Hence there would be a roughly 50-50 split between the uplink and total downlink throughput[3], which is not sensitive to the different number of *flows* served by each of the queues.

We illustrate this effect using 802.11n in the following simulation example, where *PHY* rates are now homogeneous at 65 Mbps. Initially we have 3 downlink stations, and later on a fourth station opens an uplink connection. No aggregation is used. Results are shown in Figure 5. When the downlink stations are sharing the medium throughputs are equalized, each connection getting approximately 7 Mbps, a third of the throughput they would get alone using TCP without aggregation. Once the uplink connection starts, the allocation changes, with the downlink connections getting approximately 3.5 Mbps each, while the uplink obtains around 11 Mbps. This is clearly an inconvenient result, which is not related to the particular version of the standard being used. Also matters can be worse if we add a rate-diverse environment.

We conclude that for the case of multiple STAs accessing a medium, *access opportunities should be related to the number of flows the STA (or the queue within a STA) is handling*.

[3]This explanation oversimplifies matters since ACK traffic is not considered, but captures nevertheless the essence of the problem.

The second reason for imbalance between the uplink and downlink flows is the asymmetry in TCP congestion control regarding packet and ACK losses. Note that, independently of the number of STAs or the number of uplink and downlink flows, it is clear that the AP is responsible for half of the medium accesses when there is no aggregation in place: For each data packet that the AP sends, it will receive a TCP ACK in return and for each data packet that it receives, it will send a TCP ACK in return. This implies that when there is more than one STA in the cell, the queues of these stations must empty at times in order to let the AP have half of medium accesses. As a result, *the most congested queue will be the one in the AP*, leading to the drop of some data packets of the downlink flows and TCP ACKs of the uplink flows. Now, TCP congestion control will always react to data packet loss, but ACK losses can be corrected by the subsequent ACK. This asymmetry implies downlink flows will perceive a higher drop probability thus lowering their relative throughput. This second phenomenon was already noted in [18] and [19], and solutions were proposed based on queue separation in the AP, one for the data packets and another for the TCP ACKs, regulated by a somewhat complex cross-layer information exchange. In our solution below we offer a simpler method to regulate these queues.

4. Rate Based Queueing and Aggregation

From the discussion in Section 3, it should be clear that aggregation and differentiated queueing can have an impact in the resource allocation achieved by TCP flows in an 802.11 environment if performed correctly. Moreover, access opportunities should take into account the number of flows a given node is offering to the network, whether in the downlink or uplink sense. This is particularly important to protect the AP from having less chances to access the medium when handling multiple downlink flows. The purpose of this section is to devise and test a queueing and aggregation algorithm that the AP can perform in order to find a proper resource allocation.

4.1. *The target allocation*

Consider several stations that want to communicate over the wireless link, say in the downlink sense although this is not a restriction. Assume moreover that, when transmitting alone, station *i* can achieve a throughput C_i, with the overheads taken into account. When two or more STAs compete for the medium, it is reasonable to allocate the rates x_i such that:

$$\frac{x_i}{x_j} = \frac{C_i}{C_j}. \tag{1}$$

This way, STAs which are more effective in using the medium are rewarded with higher rates. Alternatively, the *time-proportions* x_i/C_i allocated for each STA are equalized by (1): transmission time is equally shared between all stations.

This notion of fairness can also be related to the theory of Network Utility Maximization, where it coincides with the familiar notion of *proportional fairness* introduced by [4] for wired networks and in [5] in the wireless case. In this formulation, rates are chosen to solve the following convex optimization problem:

Problem 1. *Maximize* $\sum_{i=1}^{N} \log(x_i)$*, subject to*

$$\sum_{i=1}^{N} \frac{x_i}{C_i} \leqslant 1. \tag{2}$$

This differs from the standard case of [4] by the capacity constraint: in a rate-diverse situation, (2) states that the sum of time proportions in the medium can be no larger than unity. For completeness, we briefly derive[4] the solution of Problem 1, by introducing the Lagrangian

$$\mathcal{L}(x,p) = \sum_{i=1}^{N} \log(x_i) + p\left(\sum_{i=1}^{N} \frac{x_i}{C_i} - 1\right).$$

Here $p \geqslant 0$ is the Lagrange multiplier associated with constraint (2), and the Karush-Kuhn-Tucker (KKT) conditions for optimality imply that

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{1}{x_i} - \frac{p}{C_i} = 0. \tag{3}$$

From (3), we deduce that $x_i/C_i = 1/p$ for all *i*, verifying the equality of time-proportions mentioned before. Using the constraint (2) yields

$$x_i^* = \frac{C_i}{N}, \tag{4}$$

N being the total number of flows. In particular, rates are allocated proportionally to the effective capacities C_i.

Remark 1. *The allocation defined by* (4) *verifies the following attractive property: whenever a given flow in a cell changes its radio conditions, the allocated rate changes only for that flow. This is especially important in rate-diverse environments. If several flows are transmitting at the maximum possible rate, and one of them changes to a lower rate, in a typical 802.11 cell this will downgrade the rates of all flows. If* (4) *is used, faster flows are protected and as a result, the total throughput of the cell will be significantly higher.*

Having discussed our target resource allocation, we now turn out attention to implementation. We first describe the downlink case for simplicity, and then show how to adapt the solution to include the uplink traffic.

4.2. *Implementation: Downlink traffic*

The implementation question is how to drive the system to allocation (4) using modern additions to 802.11 capabilities. In order to achieve (4), we should:

- Give each flow equal channel access opportunities.
- Allow each flow to transmit during the same amount of time during a channel access.

This could in principle be implemented by putting each flow in a separate queue, with equal access opportunities (i.e. each flow has a single EDCA queue with the same AIFS and backoff parameters), and use the same TxOP time for each queue.

The first part of the solution is not practical due to the potentially large, and variable number of flows. The second part may be practical, but is very inefficient in the use of the medium. We propose instead the following Rate Based Queueing and Aggregation architecture (RBQA), which consists of three ingredients:

Queue management

The AP maintains one queue for each *PHY* data rate. This is implemented using the EDCA algorithm, but in principle each queue has the same AIFS parameter, and therefore, transmission opportunities. We use the MAC destination address to determine the current *PHY* rate and put the packet in the corresponding queue.

[4] More details are found in [6].

Remark 2. *With so many PHY rates available in 802.11n and 802.11ac, a reasonable implementation to keep the number of queues at a reasonable level, is to group similar PHY rates in the same queue. This keeps the number of queues low and still manages to approximate the desired allocation for substantially different PHY rates.*

Other implementation issue that arises, is that the PHY rate of a STA is not constant in time because of the rate adaptation algorithms that are in place. Our solution to this problem is to keep an exponential moving average of the PHY rate, and send the packets to the queue with the closer PHY rate assigned.

Frame aggregation

To achieve the desired time-fairness, each queue implements A-MPDU aggregation. The aggregation factors as a function of the physical rates are indicated in Table 3.

For the case of 802.11n, we use an aggregation limit of 1500 bytes in the slowest *PHY* rate, which amounts to 1 packet when data transfers are in place. As the *PHY* rate increases, the A-MPDU limit increases in proportion, reaching a maximum 10 : 1 ratio. Since the fixed overheads are the same for all rates, this amounts to equalizing channel usage times. This could also be achieved setting the maximum possible aggregation level on every queue and setting an appropiate TxOP time to get the same results. Table 3 also indicates the corresponding effective rates a single flow would get when alone in the cell (considering all MAC layer and TCP ACK overheads), which correspond to the C_i of equation (4). Note that these are proportional to the *PHY* rates, which means that the resulting allocation has the following property:

$$\frac{x_i}{x_j} = \frac{C_i}{C_j} = \frac{PHY_i}{PHY_j}. \tag{5}$$

This means that the resulting throughput is not only proportional to C_i but also to PHY_i, which we consider to be the fair allocation.

Of course, higher C_i's for all classes could be achieved by scaling all aggregation factors by a common number; we have refrained, however, from using aggregations beyond 10 packets for 802.11n to keep our buffering and transmission time requirements in check.

For the much faster 802.11ac we used a minimum aggregation factor of 3, because the gain in throughput is much more significant than in 802.11n and the transmission times are still small. The maximum aggregation factor of 40 respects the proportionality of 13.3 : 1 between *PHY* rates for 802.11ac.

PHY (Mbps)	A-MPDU limit (bytes)	C_i (Mbps)
6.5	1500	4.87
13	3000	9.82
19.5	4500	14.8
26	6000	19.7
39	9000	29.6
52	12 000	39.5
58.5	13 500	44.4
65	15 000	49.4

PHY (Mbps)	A-MPDU limit (bytes)	C_i (Mbps)
58.5	4500	36.8
117	9000	73.5
175.5	13 500	109
234	18 000	146
351	27 000	216
468	36 000	282
526.5	40 500	316
585	45 000	349
702	54 000	414
780	60 000	458

Table 3: PHY rates, aggregation and maximum effective TCP Rates for 802.11n and 802.11ac physical layers.

Channel access for multiple flows

To give flows equal access opportunities without having to resort to per-flow queues, the proposal is to control the aggressiveness of channel access of each AP queue j in proportion to the number of connections n_j present in it. We assume for simplicity that each STA has a single connection, and thus we can identify n_j with the number of MAC addresses present in the queue[5], something that can be tracked by the AP.

We wish to regulate the frequency τ_j of channel accesses of queue j, in proportion to n_j. Therefore we want to obtain this frequency as a function of the backoff parameters. In fact, this frequency is the inverse of the expected value of the backoff time se it is natural to think that this is inversely proportional to the base backoff window. Based on further analysis, seeing the backoff as a renewal process (see Appendix B), we get that this frequency can be approximated by:

$$\tau_j \approx \frac{2(1-2\gamma)}{CW_{min\,j}} \tag{6}$$

[5]With this approach our fairness model is established between STAs rather than TCP flows, a valid alternative.

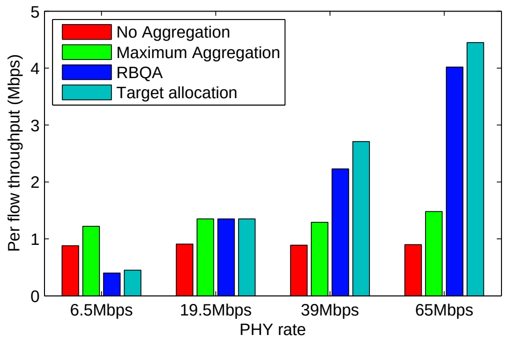

Figure 6: Per-flow throughput comparison for IEEE 802.11n in a rate diverse scenario

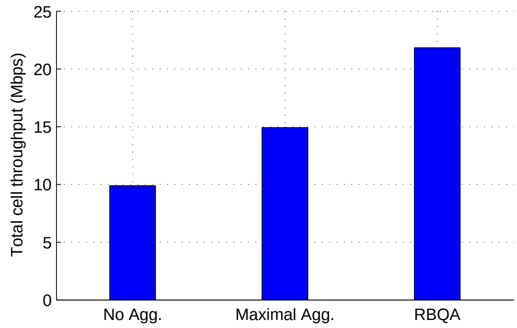

Figure 7: Total cell throughput comparison for IEEE 802.11n in a rate diverse scenario

Where γ is the collision probability seen by a queue. Taking this into account, we choose to adapt the minimum contention window parameter CW_{min_j} associated with each queue to control the frequency. The backoff adaptation algorithm is thus defined by

$$CW_{min_j} = CW_0 \frac{n_{\max}}{n_j}, \tag{7}$$

where $n_{\max} = \max_j n_j$ and CW_0 is a base contention window setting, used by the queue with most connections, which we set to $CW_0 = 16$ slots. The remaining queues have less aggressive backoff processes. By using (7), we ensure that channel access frequencies τ_j are set proportionally to the number of stations with flows traversing queue j. Thus we approximate per-flow queueing with an architecture that keeps the number of queues to a minimum and fixed in time, thereby simplifying implementation.

4.3. Simulation results: Downlink

The complete set of algorithms was implemented at the MAC layer of the AP in `ns3`. To test its performance, we simulated a rate-diverse scenario consisting of 3 STAs connected at a *PHY* rate of 65 Mbps, 1 at 39 Mbps, 5 at 19.5 Mbps and 2 at 6.5 Mbps. In Figure 6 we present the measured per-flow throughputs achieved by these connections under different algorithms: first with standard 802.11n with no aggregation and second with full length ($64KB$) A-MPDU aggregation in use. In both cases, the throughputs are roughly equalized across all classes, with aggregation providing a slight increase in performance, as discussed in Section 3. We also show the results in the same scenario, when our proposed RBQA algorithm is in place. For comparison purposes, we include the corresponding proportionally fair desired target $x_i^* = C_i/N$.

The measured throughputs are now clearly different between classes, approximating the desired proportionally fair allocation. In regard to total cell throughput, depicted in Figure 7, we have 9.9 Mbps in the standard case, and 14.9 Mbps with maximal aggregation. Instead, the RBQA algorithm achieves a total throughput of 21.8 Mbps, a 120% increase in efficiency with respect to standard 802.11n. We note that examples could be given where the increase in efficiency is even more dramatic; the above scenario was chosen to exhibit what we found to be a representative case.

4.4. Uplink traffic throttling and global solution

Up to now we have considered downlink traffic, and improved the resource allocation of the cell through queueing and aggregation. However, as we already discussed in Section 3.4, if STAs open uplink connections, the channel access algorithm will give them an important share of the resources. We would like to enhance our algorithm in order to throttle the uplink sources from the AP side, without modifying the STAs, which typically cannot be controlled directly.

Our approach here, already considered in [18] for the single rate case, is to use the TCP feedback behavior in order to force the STAs to regulate themselves by controlling the number of TCP ACK packets going in the downlink sense. This implies using separate queues for TCP ACK packets at the AP, in our case as many as the available *PHY* rates; access probabilities for the ACK queues should also be made proportional to the number of flows, i.e. we use (7) to set their contention window.

In a multi-rate environment, the remaining question is what aggregation to use in the ACK queues to reach the proportionally fair allocation (4) between all flows (uplink or downlink) in the cell. The answer is that TCP ACKs should use the aggregation factor corresponding

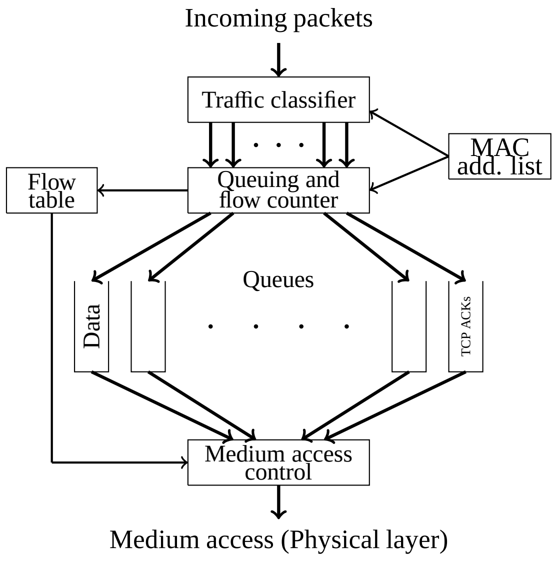

Figure 8: RBQA architecture for the AP.

to their *PHY* rate, *as if* they were data packets. For instance, to regulate 65 Mbps sources in 802.11n, we aggregate up to 10 TCP ACKs on each transmission. Note that this is different from aggregating 15 000 bytes and in the case of using TxOP, we would have to set a different time limit for the TCP ACK queues. This is why the number of packets is a better measure for us than the total time, considered when using TxOP.

The effect of the proposed aggregation is the following: the transmission rate in ACKs/sec from the AP back to the source STA will be equal to the packets/sec allocated to a downlink flow of the same *PHY* rate. Since the TCP source throttles its transmission to this ACK stream, its uplink rate in data packets/sec, and hence in Mbps, will equalize to that of downlink flow of the same *PHY* rate, as desired.

The complete RBQA architecture for the AP is shown in Figure 8. It can be implemented in the AP resorting only to local information already at its disposal. The only necessary modification to the STAs is to enable aggregation with a high A-MPDU limit.

4.5. Simulation results: Uplink and Downlink

To test the performance of our proposed algorithm, we revisit the uplink example of Section 3.4. Three downlink TCP connections at $PHY = 65$ Mbps are established and some time later, an uplink connection of the same *PHY* rate enters. Results are shown in Figure 9. Note that, due to the use of aggregation, downlink performance is improved. Most importantly, once the

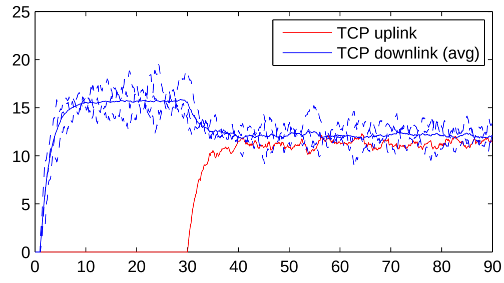

Figure 9: Throughputs for 3 downlink connections and a competing uplink, all stations at 65 Mbps and RBQA in place.

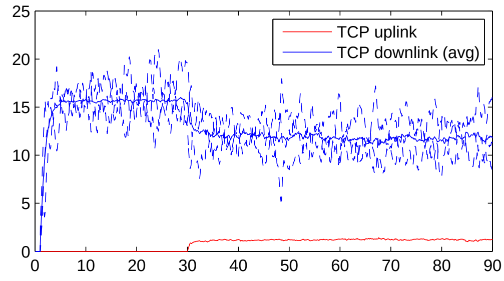

Figure 10: Throttling of slow uplink connection with RBQA.

uplink connection is started, it is throttled so the downlink connections are not unduly penalized, with all connections getting the same share.

To test rate diversity, we simulate the same scenario, but the uplink connection now enters with a *PHY* rate of 6.5 Mbps. Results are shown in Figure 10. In equilibrium, the resulting rates are approximately 12.5 Mbps for the faster flows, and 1.2 Mbps for the slow uplink flow, which coincides with the desired proportionally fair allocation (4).

As a final example of the improvements obtained by RBQA, we simulate a cell with a total of 8 stations: 4 with downlink connections and 4 with uplink ones. *PHY* rates of 175 Mbps and 780 Mbps are present in both directions. In Figure 11 we compare the per-flow throughputs obtained by the different classes when no aggregation is used, with 64*K* aggregation, and with the RBQA algorithm in place. When standard aggregation techniques are used, we observe the aforementioned imbalance between in favor of uplink classes; and within each class (uplink or downlink), throughputs are equalized independent of the *PHY* rate. As a result, total cell throughput is hindered.

The RBQA algorithm protects the fast users, and reg-

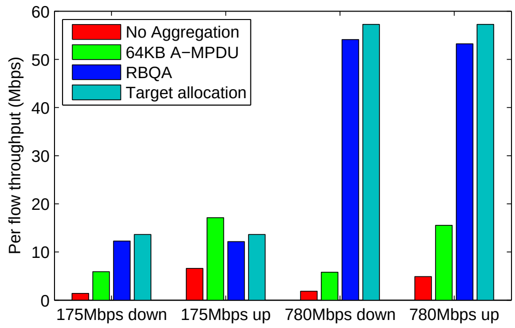

Figure 11: Per-flow throughput comparison for IEEE 802.11ac in a rate diverse scenario

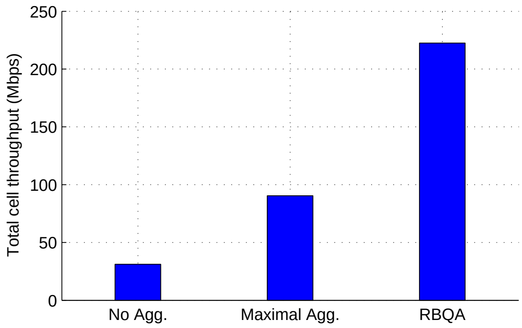

Figure 12: Total cell throughput comparison for IEEE 802.11ac in a rate diverse scenario

ulates the uplink and downlink bandwidth to the same value, which coincides with the desired proportionally fair allocation. In Figure 12 we plot the total cell throughputs in each case showing that the RBQA algorithm is much more effective in using the wireless medium.

5. Flow level performance

In Section 4 we proposed the RBQA algorithm to enforce a proportionally fair allocation of rates between permanent TCP connections. However, typical cells are subject to more random traffic conditions. We would like to incorporate more realistic traffic to our model, and analyze the behavior of our proposal in such environment with a time-varying number of ongoing flows.

A frequently used model [24] for this situation is to consider that new TCP connections of class i, associated with PHY rate PHY_i, and effective rate C_i, arrive as a Poisson process of intensity λ_i. Each connection brings a random amount of workload, independently and exponentially distributed with mean $1/\mu$. When the RBQA algorithm is in place, we may assume that connections are allocated proportionally fair service rates as in (4). This amounts to a time-scale separation assumption, where RBQA and congestion control both operate on a faster time scale than connection arrival and departure, which is a standard assumption in the literature to keep the model tractable.

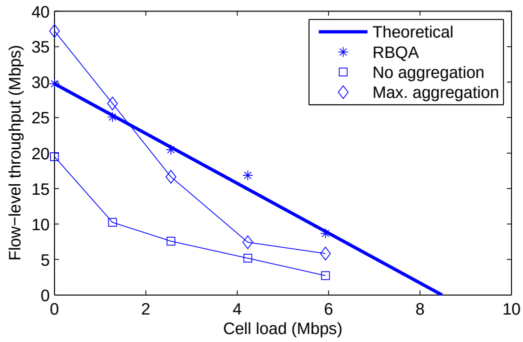

Figure 13: Flow level throughputs under different setups for the 39 Mbps class with random downlink traffic.

In this scenario, we define the *load* of the system as:

$$\rho = \sum_i \frac{\lambda_i}{\mu C_i}$$

Further analysis, based on modeling the system as a Processor Sharing Network (see Appendix), gives us the average throughput perceived by a typical connection of class i:

$$Thr_i = C_i(1-\rho). \tag{8}$$

Note that the system provides a flow-level throughput proportional to the effective rates C_i, and only coupled with the remaining classes through the total cell load, which is a desirable result. A second remark is that, due to the insensitivity properties of reversible Processor Sharing Networks [25], equation (8) still holds for general job size distribution with mean $1/\mu$. Therefore, the flow-level throughputs achieved by the system are robust, in the sense that they not depend on how job sizes are drawn.

5.1. Packet level simulations

To begin with, we simulate a single cell with downlink traffic in which incoming connections are equally split between two classes of PHY rates 39 and 6.5 Mbps respectively. Job sizes are exponentially distributed with average $3MB$, and the arrival rates are varied such that the load goes from 0 to 1. To show how RBQA

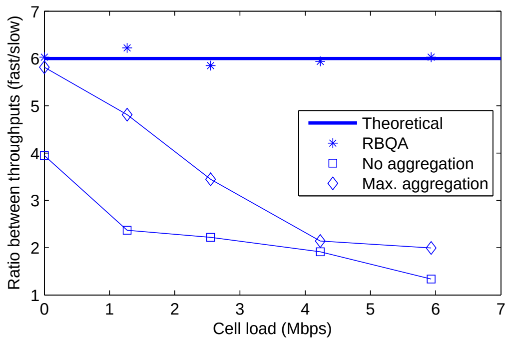

Figure 14: Ratio between the flow level throughputs between 39 Mbps and 6.5 Mbps stations with random downlink traffic.

compares to the standard 802.11n implementations, we simulate the cell under the same conditions and with the same loads[6], for the RBQA, for the case of plain 802.11n without aggregation and with full $64KB$ A-MPDU. In Figure 13 we can see that with the proposed algorithm faster stations can achieve higher throughputs even compared with the case of maximum aggregation, specially at higher loads. With very low loads it seems preferable to use the maximum aggregation instead of RBQA. This is because of the tradeoff we made between pure speed and fairness, delay and buffer requirements when we decided to use less aggregation than we could. In Figure 14 we show that RBQA maintains the same ratio between the flow-level throughputs (the ratio between physical rates) regardless of the load of the cell, which corresponds to the ideal proportionally fair allocation.

Furthermore, to show the robustness of our proposal, we also simulated the system with different job sizes, in particular Pareto (heavy tailed) and deterministic distributions, for a fixed value of the load. Results are shown in Table 4, which shows the predicted insensitivity of the allocation.

As a final example, we also simulate an IEEE 802.11ac cell exhibiting rate diversity, as well as uplink and downlink traffic. In the simulation, the RBQA algorithm is in place, controlling channel access for downlink stations and throttling the uplink flows via TCP ACKs. All traffic is random, and is split between several stations. Downlink traffic represents 75% of the load and uplink traffic the other 25%, and it is equally split between slow and fast stations. We simulate independent runs of our system for different values of the

[6]The cell load is in Mbps because each setup would yield a different load ρ.

Job size distr.	Thr_{39Mb} (Mbps)	$Thr_{6.5Mb}$ (Mbps)
Exponential	8,68	1,44
Pareto	9,30	1,57
Deterministic	9,23	1,60
Theoretical:	8,93	1,48

Table 4: Flow level throughput for two PHY classes with Poisson arrivals and different job size distribution.

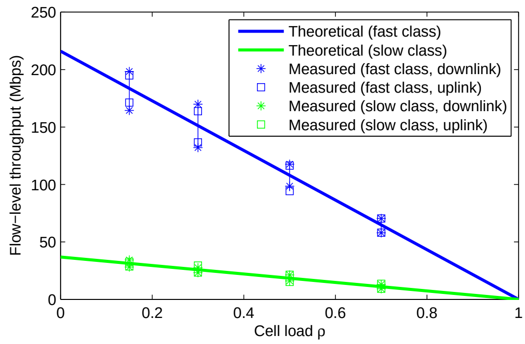

Figure 15: Flow level throughputs for a 802.11ac cell with random uplink and downlink traffic.

total cell load ρ, so as to get confidence intervals of 2σ. In Figure 15 we see that the RBQA algorithm correctly drives the system to the desired allocation, protecting the throughput of fast flows, as well as equalizing the throughput of uplink and downlink. Furthermore, we can see that the system indeed keeps a connection-level throughput differentiation across all loads, showing good fit against the theoretical predictions (equation (8)) even with uplink connections.

6. Conclusions and future work

In this paper, we analyzed several performance issues related to Wireless Local Area Networks based on the modern versions of the IEEE 802.11 standard. We showed that to be effective, packet aggregation and medium access must take into account cross-layer issues regarding the upper layer protocols. We proposed a Rate Based Queueing and Aggregation architecture that can be implemented in the Access Point, and that ensures that all data flows receive a proportionally fair share of bandwidth allocation and improves the total throughput of the cell. This algorithm relies only on locally available information and is also able to throttle the uplink flows. Moreover, we provided a packet-level implementation and simulations that validate its behavior in several settings.

In future work, we plan to analyze how to improve our algorithm to take into account other classes of traffic, such as real time or streaming, which should themselves be protected from data transfers. Note that, as designed, our algorithm is well suited to combined implementation with current priority-based mechanisms based on 802.11e queues. We also plan to implement the architecture in a real network deployment.

Appendix

TCP ACKs performance issue

In this appendix we provide a theoretical model to explain the performance issues in the scenario of one STA downloading data from the AP using TCP, with packet aggregation enabled *only* in the AP for data packets, not for the ACK queue at the STA. We model the number of packets in the AP queue as a discrete time Markov chain, defining time as the interval between successful transmissions, and making the following simplifying assumptions:

- The TCP congestion window is very large and constant (W).
- The aggregation limit is high enough to send all packets at once each time it transmits data (thus emptying the queue).
- Both the AP and the STA queues are big enough to contain W packets.

These assumptions may seem unrealistic, but are adequate approximations for long enough connections and lead to an accurate model nonetheless.

For illustration, the state diagram for the case $W = 4$ is depicted in Figure 16. Note that at state W, there can be no packets at the STA so only the AP attempts transmissions; similarly at state 0 only the STA will transmit. This explains the transitions with probability 1 from those states. At all other states, there is equal chance of successful transmission for both AP and STA, and in the former case the queue is emptied through packet aggregation.

Using the balance equations, we can find the invariant measure π of the process, and taking $W \to \infty$ we get

$$\pi(0) = \frac{1}{3},$$
$$\pi(n) = \frac{1}{3}\left(\frac{1}{2}\right)^{n-1} \qquad \forall n \geq 1.$$

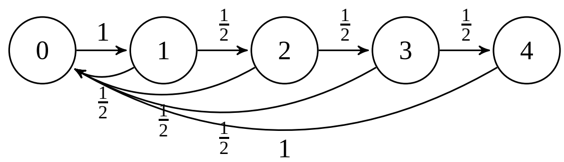

Figure 16: State diagram for the discrete time Markov chain for the AP queue.

First of all, we can see from the invariant measure that the queue is actually empty one third of the time. Moreover, we find that the average number of packets in the queue is $\frac{4}{3}$ and that the average number of packets aggregated in a single transmission is 2. This behavior is consistent with our previous simulations and explains why the throughput of the network does not improve using more aggregation: the amount of packets in the queue limits it.

As a last remark, we note that the AP queue is almost empty most of the time which implies that the actual data packets are never dropped. In a real situation with a variable TCP congestion window, it will just keep growing, filling the STA queue with TCP ACKs. This is also consistent with the results obtained through simulations.

Approximation of access frequency

We want to control de access frequency of each queue that is virtually competing for the medium inside the AP. In particular, we want to express the access frequency τ_j of queue j as a function of the backoff parameters of the queue. For this purpose we refer to the analysis in [10] for the backoff process, where the channel access attempt rate is obtained to be

$$G_j(\gamma_j) = \frac{\sum_{i=0}^{K_j} \gamma_j^i}{\sum_{i=0}^{K_j} \frac{CW_{min\,j}}{2}(2\gamma_j)^i}.$$

Here γ_j is the collision probability of the queue j and K_j is the number of backoff stages of the queue j. Multiplying this for the probability of success $(1 - \gamma_j)$ we get the actual frequency τ_j for the queue

$$\tau_j = \frac{2\left(1 - \gamma_j^{K_j+1}\right)(1 - 2\gamma_j)}{CW_{min\,j}\left(1 - (2\gamma_j)^{K_j}\right)} \approx \frac{2(1 - 2\gamma_j)}{CW_{min\,j}}.$$

We see that τ_j is inversely proportional to CW_{min_j}, but the proportionality constant is different for every queue.

The difference lies in the fact that these collision probabilities are the probability that other queue or STA attempts to transmit in the same slot. Typically we have many queues and STAs contending for the medium, which means that the collision probabilities γ_j are all very similar and thus we can approximate them by the average collision probability γ. Taking this into account, we get the following expression for the access frequency

$$\tau_j \approx \frac{2(1-2\gamma)}{CW_{min\,j}}. \tag{9}$$

This is the approximation that we use to design the algorithm.

Connection level theoretical model

In this appendix we provide a theoretical model that gives us the expected performance of the network under random traffic. The hypothesis are that new TCP connections of class i (effective rate C_i) arrive as a Poisson process of intensity λ_i, and that each connection brings a workload independently and exponentially distributed with mean $1/\mu$. When the RBQA algorithm is in place, we may assume that connections are allocated as in (4). This amounts to a time-scale separation assumption, where RBQA and congestion control both operate on a faster time scale than connection arrival and departure.

When job sizes are exponentially distributed, this type of model was studied in [6]. The vector valued process $n(t) = (n_i(t))$ recording the number of ongoing connections in each class constitutes a continuous time Markov chain with transition rates:

$$q_{n,n+e_i} = \lambda_i \tag{10a}$$

$$q_{n,n-e_i} = \mu C_i \frac{n_i}{\sum_j n_j}, \tag{10b}$$

where e_i is the i-th coordinate vector. Equation (10a) accounts for the job arrivals in class i. In turn, equation (10b) assigns a rate to class i departures, of the form $\mu n_i r_i(n)$, with $1/\mu$ the average job size, n_i the current number of connections on class i, and $r_i(n)$ the rate of an individual connection in the present state, which is given by the proportionally fair allocation (4).

The Markov chain defined by (10) is a particular case of a *Discriminatory Processor Sharing* queue [26], with equal weights for all classes. In particular, if we define the load of the system by $\rho = \sum_i \frac{\lambda_i}{\mu C_i}$, then the flow-level queue is stable only if $\rho < 1$, i.e. the time proportions needed to serve all flows on average are less than unity.

This particular case can also be solved explicitly, with the average number of flows in equilibrium on class i satisfying:

$$E[n_i] = \frac{\rho_i}{1-\rho},$$

where $\rho_i = \frac{\lambda_i}{\mu C_i}$ is the load of class i. By applying Little's law we can estimate the average job completion time $E[T_i]$ for class i as:

$$E[T_i] = \frac{1}{\lambda_i} E[N_i] = \frac{1/(\mu C_i)}{1-\rho}.$$

An estimation of the throughput perceived by a typical connection of class i can be derived by normalizing the above by the average job size $1/\mu$. We get:

$$Thr_i = C_i(1-\rho).$$

In this context, $(1-\rho)$ is called the *slowdown* of the processor sharing queue. This throughput is effectively proportional to C_i and is decreasing with the cell load.

References

[1] IEEE 802.11-2012, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.

[2] IEEE 802.11n-2009, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, 2009.

[3] IEEE 802.11 task group AC, Very High Throughput below 6GHz, 2012.

[4] F. Kelly, A. Maulloo, D. Tan, Rate control in communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society 39 (1998) 237–252.

[5] P. Viswanath, D. Tse, R. Laroia, Opportunistic beamforming using dumb antennas, IEEE Tran. on Inf. Theory 48 (2002) 1277–1294.

[6] A. Ferragut, F. Paganini, Resource allocation over multirate wireless networks: A network utility maximization perspective, Computer Networks 55 (2011) 2658–2674.

[7] K. Kashibuchi, A. Jamalipour, N. Kato, Channel occupancy time based TCP rate control for improving fairness in IEEE 802.11 DCF, IEEE Trans. on Vehicular Tech. 59 (2010) 2974–2985.

[8] M. Zubeldia, A. Ferragut, F. Paganini, Averting speed inefficiency in rate-diverse wifi networks through queueing and aggregation, in: IEEE Globecom.

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

[10] A. Kumar, E. Altman, D. Miorandi, M. Goyal, New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs, IEEE/ACM Transactions on Networking 15 (2007) 588–601.

[11] J. Choi, K. Park, C. kwon Kim, Cross-layer analysis of rate adaptation, DCF and TCP in multi-rate WLANs, in: IEEE Infocom.

[12] Y. Kim, S. Choi, K. Jang, H. Hwang, Throughput enhancement of ieee 802.11 wlan via frame aggregation, in: IEEE 60th. Vehicular Technology Conference.

[13] D. Skordoulis, N. Qiang, H.-H. Chen, A. Stephens, L. Changwen, A. Jamalipour, IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs, IEEE Wireless Communications 15 (2008) 40–47.
[14] B. S. Kim, H. Y. Hwang, D. K. Sung, Effect of frame aggregation on the throughput performance of IEEE 802.11n, in: IEEE WCNC.
[15] J. Karlsson, A. Kassler, A. Brunstrom, Impact of packet aggregation on TCP performance in wireless mesh networks, in: IEEE WoWMoM.
[16] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, T. Turletti, Aggregation with fragment retransmission for very high-speed WLANs, IEEE/ACM Transactions on Networking 17 (2009) 591–604.
[17] G. Banaghe, D. Raychaudhuri, I. Seskar, Backlogged queue based MAC frame aggregation, Pervasive and Mobile Computing 7 (2011) 449–466.
[18] J. Ha, C.-H. Choi, TCP fairness for uplink and downlink flows in WLANs, in: IEEE Globecom.
[19] Q. Xia, X. Jin, M. Hamdi, Active queue management with dual virtual proportional integral queues for TCP uplink/downlink fairness in infrastructure WLANs, IEEE Trans. on Wireless Communications 7 (2008) 2261–2271.
[20] T. Bonald, S. Borst, N. Hegde, M. Jonckheere, A. Proutière, Flow-level performance and capacity of wireless networks with user mobility, Queueing Systems 63 (2009) 131–164.
[21] A. Ferragut, F. Paganini, A connection level model of IEEE 802.11 cells, in: IFIP/ACM Latin American Networking Conference (LANC'09).
[22] NS-3Project, NS-3 Manual, 2011.
[23] R. Srikant, The Mathematics of Internet Congestion Control, Birkhäuser, Boston, MA, 2004.
[24] T. Bonald, A. Proutière, Flow-level stability of utility-based allocation in non-convex rate regions, in: Proc. of the 40th Annual Conference on Information Systems and Sciences (CISS 2006), pp. 327–332.
[25] T. Bonald, A. Proutière, Insensitivity in processor-sharing networks, Performance Evaluation 49 (2002) 193–209.
[26] E. Altman, K. Avrachenkov, U. Ayesta, A survey on discriminatory processor sharing, Queueing Systems: Theory and Applications 53 (2006) 53–63.