



Institut Eurécom  
Networking and Security Department  
2229, route des Crêtes  
B.P. 193  
06904 Sophia-Antipolis  
FRANCE

Research Report RR-08-220

**An Experimental Comparison of Layer 2 and 3 Mechanisms  
for Improving User Perceived Performance of 802.11  
Wireless LANs**

May 15<sup>th</sup>, 2008

Martin Heusse\*, Guillaume Urvoy-Keller<sup>†</sup> and Andrzej Duda\*

\* Institut Eurecom

<sup>†</sup> Laboratoire d'Informatique de Grenoble

Email : Martin.Heusse@imag.fr, Guillaume.Urvoy@eurecom.fr,  
Andrzej.Duda@imag.fr

---

<sup>1</sup>Institut Eurécom's research is partially supported by its industrial members: BMW Group Research & Technology - BMW Group Company, Bouygues Télécom, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales.



# **An Experimental Comparison of Layer 2 and 3 Mechanisms for Improving User Perceived Performance of 802.11 Wireless LANs**

Martin Heusse, Guillaume Urvoy-Keller, and Andrzej Duda

## **Abstract**

We propose LAS-ACK (Least Attained Service with TCP ACKs), a new layer 3 scheduling mechanism for solving the TCP unfairness problem in 802.11 wireless LANs. We evaluate its performance through measurements of a realistic TCP workload on a wireless platform, using a variety of metrics that reflect the user perceived performance as well as the efficiency at the network layer. The proposed mechanism achieves even better results if combined with AAP (Asymmetric Access Point), a MAC layer access method that gives more capacity to the access point. We show that an efficient way to combine AAP and LAS-ACK is to deploy LAS-ACK at the wireless stations only. This is a desirable property as it allows to keep the access point fairly simple and shifts the computational complexity of LAS-ACK to the wireless stations that have much more resources than the access point in general.



## **Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>2</b>
<b>3</b>	<b>LAS-ACK scheduling discipline</b>	<b>3</b>
<b>4</b>	<b>Measurement setup</b>	<b>3</b>
4.1	Workload . . . . .	5
4.2	Performance metrics . . . . .	7
<b>5</b>	<b>Performance results</b>	<b>8</b>
<b>6</b>	<b>Conclusions</b>	<b>17</b>

## List of Figures

1	Measurement platform . . . . .	4
2	Cumulative distributions functions of the transfer sizes used in the experiments. . . . .	6
3	% of bytes carried by transfers of size less than $n$ MSS for all $n$ values . . . . .	7
4	Asymmetric load and Pareto distributed connection sizes, 20ms delay. . . . .	9
5	Asymmetric load and Pareto distributed connection sizes, 150ms delay. . . . .	9
6	RTT histograms for asymmetric load and Pareto distributed connection size, 20ms delay. . . . .	10
7	RTT histograms for asymmetric load and Pareto distributed connection size, 150ms delay. . . . .	10
8	Conditional throughput for asymmetric load and Pareto distributed connection size. . . . .	11
9	Measured aggregated throughput, Pareto distributed connection size. . . . .	12
10	Asymmetric load and exponentially distributed connection size, 150ms delay. . . . .	13
11	Cumulative distribution functions of the sizes of completed connections for asymmetric load and exponentially distributed connection size, 20ms delay. . . . .	14
12	Cumulative distribution functions of completed connections for asymmetric load and Pareto distributed connection size, 150ms delay. . . . .	15
13	Inter-arrival times at the access point queue for asymmetric load and Pareto distributed connection size, DCF/FIFO policy . . . . .	16
14	Measured aggregated throughput, exponentially distributed connection size. . . . .	17

# 1 Introduction

In current infrastructure mode 802.11 wireless LANs, TCP traffic may suffer from significant unfairness between upload and download connections, long delays, and significant packet loss. This *TCP unfairness problem* comes from the fact that the 802.11 access point does not benefit from enough radio channel share compared to wireless stations [1–9]. The standard IEEE 802.11 DCF (*Distributed Coordination Function*) [10] access method provides approximately equal channel access probability to all devices in a wireless cell. Thus, if there are  $N$  wireless stations in a cell, the access point only benefits from  $1/(N + 1)$  of channel access probability. In this case, the access point becomes a bottleneck that limits the overall throughput, which leads to lost frames due to buffer overflow. For upload connections, lost TCP ACKs sent to wireless stations do not raise much problems, because TCP ACKs are cumulative. Data segments of download connections fill up the access point buffer and are dropped unless it is large enough for the TCP sources to choke on the window size [3]. This behavior results in far better throughput obtained by uploading stations compared to the downloading ones. Moreover, it also impacts reactivity of short flows and interactive applications—the saturated queue at the access point results in long response times from the user point of view.

Several authors have proposed solutions to address the TCP unfairness problem at different layers: transport, network, or MAC (cf. Section 2). As the main cause of the problem is insufficient capacity of the access point, it seems promising to solve it by an appropriate solution at the MAC layer. Recent AAP (Asymmetric Access Point) proposal [9] is a pure MAC layer approach based on the Idle Sense access method [11, 12]. It gives special importance to the access point by allocating more capacity in a dynamic way: the access point always benefits from twice the access probability of that of all the stations present in the wireless cell. Thus, the downlink queue at the access point never builds up unless traffic becomes intensive and downlink unbalanced, but this is highly unlikely, because all transport protocols like TCP, DCCP, or SCTP mostly generate one ACK per every other data segment. Although AAP significantly improves fairness of TCP flows, it cannot provide an optimal solution, because it would require higher layer information, in particular about a given mix of upload/download connections.

LAS [13] (Least Attained Service) discipline schedules packets by giving greater priority to packets from flows that have generated less traffic so far. LAS drastically improves responsiveness of short connections. We adapt LAS for operating at the access point in a way that takes into account the half-duplex nature of the 802.11 wireless link: the packet priority depends on the total bi-directional traffic of each TCP connection (data and ACK segments). This is achieved by adding to each considered segment size the amount of data acknowledged by this segment. We call this variant LAS-ACK. LAS-ACK needs to maintain two counters per TCP connection: the last ACK sequence number observed and the total data size for the bi-directional connection. This idea is similar to the one at the root of VFQ (Virtual Flow Queueing) [14], that extends WFQ to take into account the progress of

TCP connections over half duplex wireless links. Note that LAS-ACK operates at the network layer, but it relies on the transport layer information (number of transmitted and acknowledged bytes).

In this paper we measure the performance of TCP flows over 802.11 WLANs to find the best mechanism for solving the TCP unfairness problem. We consider realistic synthetic TCP traffic with various types of load distributions and patterns. First, we compare FIFO packet scheduling with LAS-ACK over the standard 802.11 DCF access method. LAS-ACK drastically reduces the response time of short connections at the expense of long-lived ones. To alleviate this binary effect, we investigate coupling of LAS-ACK with AAP. As AAP moves the bottleneck to the uplink direction, it is sufficient to only apply LAS-ACK to wireless stations, which makes the access point fairly simple (constant contention window at the MAC layer proper to the AAP method and simple FIFO scheduling). It turns out that the combination of these two methods offers the best performance trade-off: it significantly improves performance by giving the right priority to the access point and properly scheduling flows sent by wireless stations.

Unlike many studies of TCP unfairness solutions usually evaluated via simulation or analytical modeling, we report on measurement data of TCP performance and UDP delays gathered on an experimental platform implementing the proposed mechanisms.

The paper is organized as follows. Section 2 overviews the related work. Section 4 presents the experimental platform and synthetic workload. Then, we report performance results in Section 5. Section 6 concludes the paper.

## 2 Related work

Several authors have studied the TCP unfairness problem. S. Pilosof *et al.* proposed to modify the receiver window in TCP ACKs to pace sources on wireless stations and provide in this way more bandwidth for the download traffic [3]. Some authors proposed to solve the unfairness problem by providing a suitable scheduling mechanism at the IP layer. D. Eckhardt *et al.* in [15] defined an *Effort Limited Fair* scheduling for wireless networks. Other authors proposed to use QoS support or service differentiation to cope with performance problems over WLANs [16,17]. J. Ha *et al.* address the unfairness problem with two distinct queues for data segments and ACKs at the access point [18]. They tune their relative priorities according to the number of flows in both directions and the corresponding offered window field in ACK segments. Finally, Many authors propose to solve the unfairness problem by using an adequate MAC access method. D.J. Leith *et al.* choose suitable parameters of IEEE 802.11e to provide fairness between competing TCP uploads and downloads [7, 8]. Other authors propose algorithms to enhance performance in asymmetric traffic load conditions by giving more priority to the access point. However, they rely on exchanging information between the access point and wireless stations [2, 4, 19]. Setting up differentiation parameters is not a simple task,



because the priority given to the access point needs to adapt to the current load of a cell and the number of active stations. AAP (Asymmetric Access Point) [9] sets its contention window to a constant value while wireless stations use the *Idle Sense* access method. In this way, AAP obtains a transmission capacity twice greater than that of the sum of all active stations independently of the number of contending stations.

### 3 LAS-ACK scheduling discipline

LAS (Least Attained Service) also called Foreground- Background (FB) or Shortest Elapsed Time (SET) first, is a preemptive scheduling policy that gives service to the job that has so far received the least service. If multiple jobs receive the same amount of service, they share capacity according to the processor-sharing policy. LAS favors short jobs without requiring the knowledge of job sizes. The impact of short jobs on the mean response time of large jobs under LAS highly depends on the job size distribution. In particular, for job size distributions with a high coefficient of variation, the small mean response times of short jobs comes at the expense of only a very small increase in the mean response times of the largest jobs.

LAS fits the Internet traffic particularly well, because flow duration follows long tail distributions with large coefficients of variation: most of the flows are short, while more than half of the bytes are carried by a small percentage of flows that are very long. A network flow is identified by the source and destination addresses and ports. In LAS scheduling, a router identifies the first and subsequent packets in a flow, adds up the amount of data transferred by the flow, and uses this sum to insert the packet into a priority queue. Packets are served in the order of the smallest volume of transferred data first, as this corresponds to the attained service. Packets with the same volume of transferred data are served in the FIFO order, which implies an approximate round-robin service.

When we want to apply LAS to schedule TCP flows in a wireless cell, we need to take into account not only data segments and their respective sequence numbers, but also ACKs as they make TCP sources advance in a data transfer. Note that the 802.11 wireless link is half-duplex and both data and ACK segments contend for channel access. In the variant called *LAS-ACK*, we thus assign to a TCP ACK a priority that depends on the total volume of data transferred in both directions by looking at the amount of data acknowledged by each segment. This requires maintaining two counters per TCP connection: the last ACK sequence number observed and the total data size for the bi-directional connection.

### 4 Measurement setup

We use a group of six wireless stations with one acting as an access point connecting the others to the wired part of the network (cf. Figure 1). Four stations

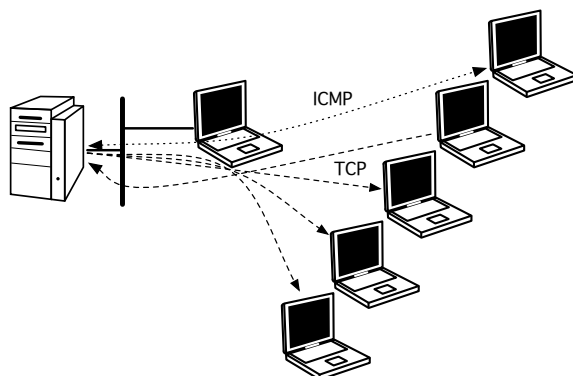


Figure 1: Measurement platform

generate TCP traffic and one monitors the delay and loss rate at the access point with `ping` at the rate of 5 packets per second.

All stations may use the standard 802.11 DCF access method, switch to a different microcode running Idle Sense, or configure a fixed contention window like in AAP. This means that we can set the MAC layer of the platform to the standard 802.11 DCF or AAP. At the packet level, the access point and stations can either use standard FIFO scheduling or LAS-ACK. In this way, we can measure four combinations of mechanisms at layer 2 and 3: DCF/FIFO, AAP/FIFO, DCF/LAS-ACK, and AAP/LAS-ACK. We have implemented LAS-ACK by modifying the BSD `dummynet` kernel module. We can also emulate different propagation delays over the wired part of the network in `dummynet`.

We use Intel IPW2915 wireless cards operating according to the 802.11a standard at the 12Mb/s rate. We have chosen this relatively low bit rate to operate in good channel conditions, but our results are still valid for higher bit rates, because the unfairness problem is related to the MAC layer behavior in saturated or near saturated conditions and does not depend on a particular bit rate.

We have disabled FreeBSD caching of TCP parameters and turned off the limitation of the TCP window based on Bandwidth-Delay product estimation (cf. [20] for more details). We have also tuned up buffer sizes at layer 2 and 3. On our FreeBSD wireless stations, the default layer 2 buffer size is 64 frames in addition to the layer 3 queue of 50 packets, which makes a total of 114 packets. We have changed these default settings to 1 frame at layer 2 and 20 packets at layer 3, which corresponds to usual settings of commercial access points. A longer layer 3 buffer would increase the time spent by packets at the access point queue for DCF/FIFO or at wireless stations for AAP/FIFO. A longer layer 2 buffer has the same impact and also makes any layer 3 queueing strategies other than FIFO ineffective. More generally, large buffers are detrimental to TCP performance as they inflate the round trip time (RTT) on which depends TCP reaction to network load variations.

## 4.1 Workload

To observe the impact of any IP and MAC scheduling policy, the overall load on the wireless medium must be large enough. In our experiments, we assume TCP connections arriving according to a Poisson process with rate  $\lambda$  adjusted such that the offered load on the wireless medium is equal to 10 Mb/s on average, which slightly overloads the wireless link operating at 12 Mb/s nominal rate. Since TCP controls transfers and losses can occur at the access point, the observed load is smaller than the offered load.

The workload consists of bulk TCP transfers of varying size. All TCP connections use 1500 bytes MSS. We draw the volume of data to transfer from a distribution with a fixed average value. We set the average connection size to 60 KB (40 packets of 1500 B), which is in line with flow sizes observed on typical campus WLANs [21]. Note that the rise of social networks and videocasters like YouTube [22] tend to increase the average size of bulk TCP transfers.

We consider two different distributions of the TCP connection size. The first one is Pareto denoted by  $P(k, \alpha)$ , where  $k$  is the minimum connection size and  $\alpha$  is the exponent of the power law. The density of this distribution is given by:

$$f(x) = \alpha k^\alpha x^{-\alpha-1}, \quad k \leq x, 0 \leq \alpha \leq 2. \quad (1)$$

It corresponds to realistic workload with long tail distribution of data volume transferred by TCP. We can tune its coefficient of variation (CoV—the ratio of the standard deviation to the mean) through parameter  $\alpha$ . We have chosen the value of CoV close to 6, which is in the range of common values for WLAN traffic (e.g. observed values of CoV are between 2 and 6 [21]).

For comparison purposes, we also consider a second distribution—an exponential one of parameter  $\mu$  with density function:

$$f(x) = \mu e^{-\mu x}, x \geq 0. \quad (2)$$

It is less realistic than the Pareto distribution, but the performance of LAS (and hence of LAS-ACK) strongly depends on the variability of the transfer size distribution [13]: LAS performs better for more variable distributions, because the probability mass corresponding to short flows is small enough to avoid starvation of long flows. Thus, the exponential distribution represents unfavorable conditions for the LAS-ACK policy.

To account for the fact that TCP connection sizes have typically a minimum and maximum size, we set the minimum size of  $k = 6$  MSS and maximum size  $P = 13,000$  MSS for both the Pareto and the exponential distributions. 13,000 MSS correspond to a maximum transfer size of about 20 MB, which is a reasonable value for a 802.11 WLAN.

Figure 2 presents the cumulative distribution function of the transfer sizes observed in our experiments. We can see that the maximum transfer size is 246 MSS for the exponential distribution and 5139 MSS for Pareto. These values are smaller

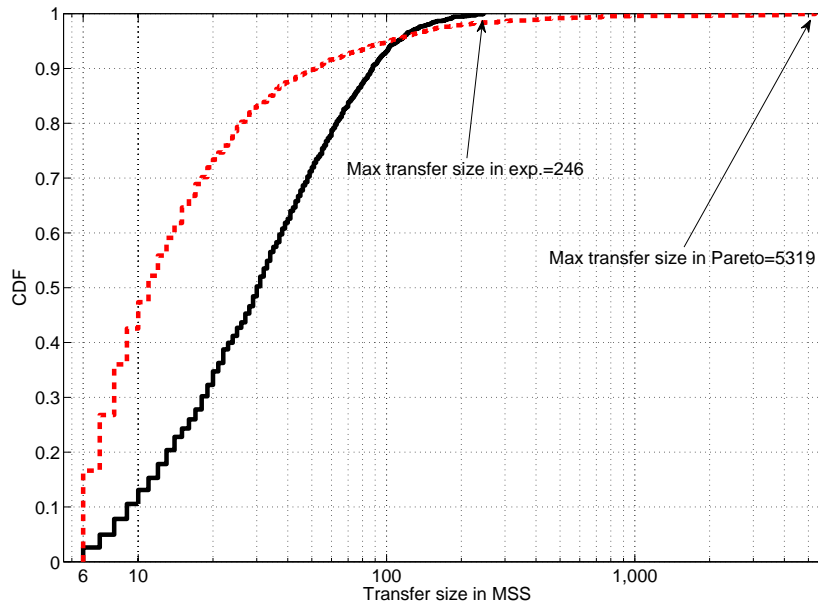


Figure 2: Cumulative distributions functions of the transfer sizes used in the experiments.

than 13,000 MSS, because we gather a finite sample for each distribution due to a limited duration of measurements. For the two distributions, the majority of TCP transfers are fairly short: 95% of them are smaller than 100 MSS. Short TCP transfers are known to be unresponsive to variations of network load. Note that as we use an advertised window of 64 Kbytes (the usual case for the Internet traffic), TCP needs to send 128 segments before entering the congestion avoidance phase if the delayed acknowledgement mechanism is used (the usual case again) and no losses occur.

Even if for both distributions 95% of connections are smaller than 100 MSS, the distributions are very different if we consider the volume of transferred data in bytes. Figure 3 shows that only about 38% of the total data volume is carried by transfers of size less than 100 MSS in the case of the Pareto distribution while it is 77% for the exponential distribution. These measurements indicate that the exponential workload is mainly composed of short TCP transfers conveying most of the data volume. It is less responsive than the Pareto one, because most of TCP transfers do not leave the slow start state. Note also that the distribution of finished transfers may be different from the generated distributions, because some transfers are not finished during the measurement period (cf. Section 5).

We consider two mixtures of upload and download connections: a *balanced* load with the same proportion of uploads and downloads, and an *asymmetric* one with 75% of downloads and 25% of uploads. For a given scenario, a wireless

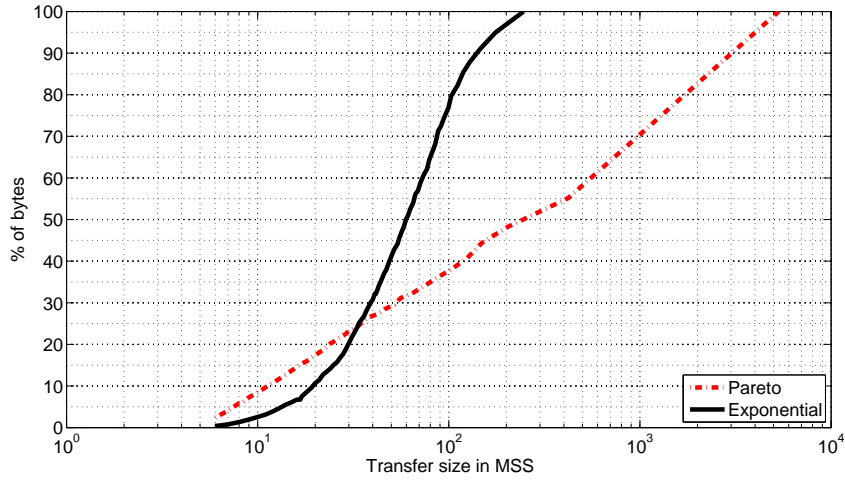


Figure 3: % of bytes carried by transfers of size less than  $n$  MSS for all  $n$  values

station acts either as a sender or as a receiver. In addition, we evenly share the download (resp. upload) load on all the stations that perform downloads (resp. uploads).

## 4.2 Performance metrics

We consider several performance metrics:

- the *conditional connection response time*: the time required for a TCP connection of a given size to finish a transfer,
- the *response time variability*: measured as the difference between the 90-th and 10-th quantile of the conditional connection response time,
- the *packet delay*: the round trip time of a packet,
- the *packet loss rate*: the proportion of lost packets,
- the *conditional connection throughput*: the throughput obtained by a TCP connection of a given size,
- the *aggregated throughput*: the throughput obtained by all TCP connections.

The conditional response time and its variability characterize user perceived performance—how long a TCP connection takes to transfer a given volume of data. Ping statistics estimate the packet delay and loss rate. This measure is also important for interactivity, because for instance losses of DNS requests or replies greatly impact web browsing. The conditional connection throughput quantifies network capacity used by a connection of a given size. Finally, the aggregated

throughput measures the effective usage of network resources: it may vary depending on a given combination of workload, packet scheduling discipline, and the WLAN access method. We measure the aggregated throughput on the wired part of the network.

Nevertheless, we want to point out that no single metric fully captures user perceived performance in a wireless environment. We believe that the conditional connection response time is a valuable gauge, especially for interactive traffic such as Web browsing. However, it fails to capture the lack of efficiency in network usage for some scheduling disciplines. Similarly, loss rate contributes to the user perceived response time, because SYN segments get retransmitted with the initial retransmission timeout. It also expresses the level of buffer saturation at the access point and may lead to extrapolation of what would happen to flows with various RTTs or types of payload (VoIP).

Each measurement experiment lasts for 100 s. Some connections are unfinished at the end of an experiment, either because of the elapsed time or an aborted transfer due to a high loss rate. We report performance results for connections that completed a transfer, if relevant.

## 5 Performance results

In this section, we present the measured performance results for TCP flows over 802.11 WLANs under different layer 2 MAC access methods and layer 3 scheduling disciplines:

- DCF/FIFO,
- DCF/LAS-ACK,
- AAP/FIFO,
- AAP/LAS-ACK.

Note that when the access point operates under the AAP method (resp. DCF), wireless stations use Idle Sense (resp. DCF). The default layer 3 policy at wireless stations is FIFO except for AAP/LAS-ACK: in this case the access point uses FIFO while wireless stations operate under LAS-ACK.

We first report the results for the realistic distribution of connection size (Pareto) under asymmetric load. Figures 4 and 5 present the conditional connection response time in function of the percentiles of connection sizes for the delay over the wired part of 20 ms and 150 ms, respectively (100 percentile corresponds to 5139 MSS).

*Observation 1: under DCF/FIFO TCP connections suffer from important response time and high variability. Moreover, download connections and especially the largest ones, take significantly more time to complete than the upload ones.*

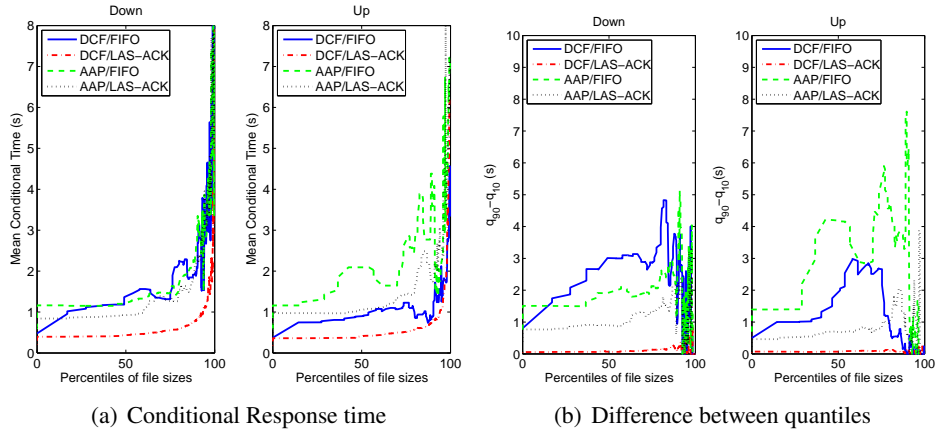


Figure 4: Asymmetric load and Pareto distributed connection sizes, 20ms delay.

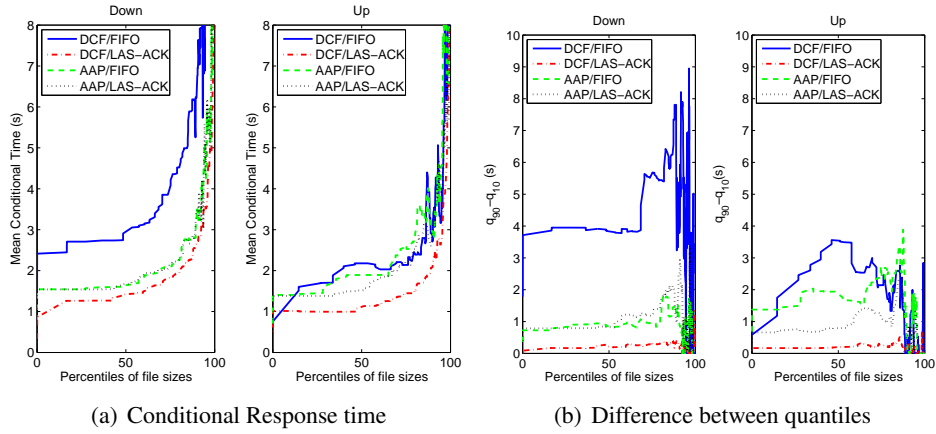


Figure 5: Asymmetric load and Pareto distributed connection sizes, 150ms delay.

For the delay of 20 ms, we can observe that under DCF/FIFO connections last of the order of 1 to 2 seconds even for small transfer sizes irrespectively of the direction. When the delay is longer (150 ms), the response time of DCF/FIFO becomes even worse with significant unfairness between download and upload connections. This extends the TCP unfairness problem of DCF/FIFO to the case of the TCP workload with flows of various sizes (it was studied before for persistent long-lived TCP flows). We can also observe high variability of the response time—Figures 4(b) and 5(b) illustrate how many connections may experience the response time of up to several seconds for downloading tens of TCP segments.

*Observation 2: applying LAS-ACK to give more priority to short flows dramatically improves the response time and lowers variability for most connections.*

We can see in Figure 4(a) that the response time under LAS-ACK is constantly reduced to a fraction of a second except for the largest flows. Figure 5(a) shows a similar effect for the delay of 150 ms. Even more importantly for the user, LAS-ACK lowers the variability of the response time (cf. Figures 4(b) and 5(b)).

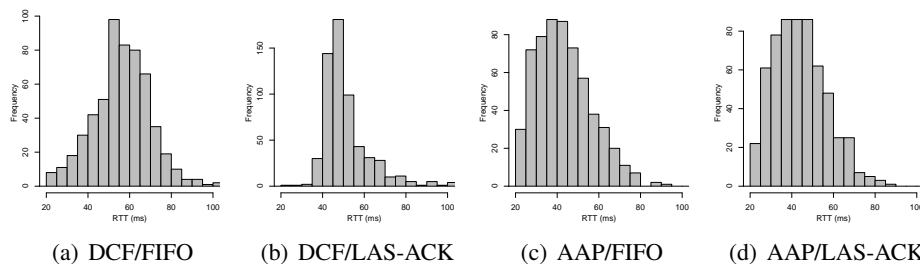


Figure 6: RTT histograms for asymmetric load and Pareto distributed connection size, 20ms delay.

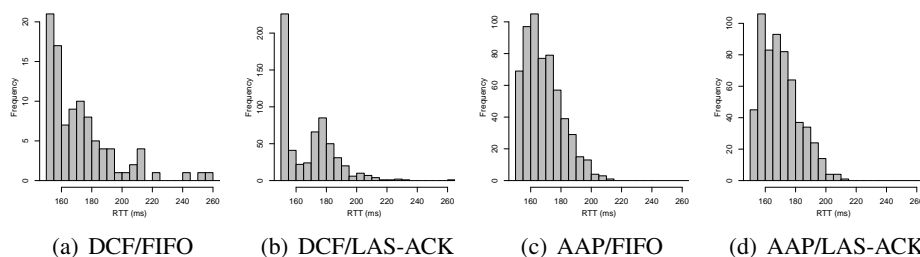


Figure 7: RTT histograms for asymmetric load and Pareto distributed connection size, 150ms delay.

Replacing DCF with AAP while still operating under FIFO shifts the point of congestion to upload stations. Consequently, we observe a marginally better response time for downloads only when the latency is significant and performance gets worse for uploads. This is due to the fact that all wireless stations are backlogged—packets suffer from a long delay in the output buffers at stations. However, the gain in variability for downloads is notable. Moreover, if a station had a lower demand than others, it would experience short response times as the shared buffer at the access point does not grow significantly. We can see this effect in Figures 6 and 7 presenting the packet RTT histograms: RTT is longer for DCF, because the buffer of the access point is full most of the time.

*Observation 3: LAS-ACK scheduling with AAP greatly improves up-load responsiveness and provides intermediate results between DCF/LAS-ACK and AAP/FIFO.*

AAP/LAS-ACK solves the performance problem of DCF/FIFO so that we can observe low variability of the response time (cf. Figures 4(b) and 5(b)). All stations



Table 1: Packet loss rate, Pareto distributed connection size

	DCF FIFO	DCF LAS-ACK	AAP FIFO	AAP LAS-ACK
Asym., 20ms	6%	0%	0%	1%
Asym., 150ms	9%	0%	2%	1%
Sym., 20ms	14%	0%	0%	0%
Sym., 150ms	9%	0%	1%	0%

can now benefit from the short queue at the access point. In this combination of mechanisms, LAS-ACK improves scheduling on greedy stations, while AAP results in better sharing of wireless capacity—the wireless cell does not suffer too much from congestion.

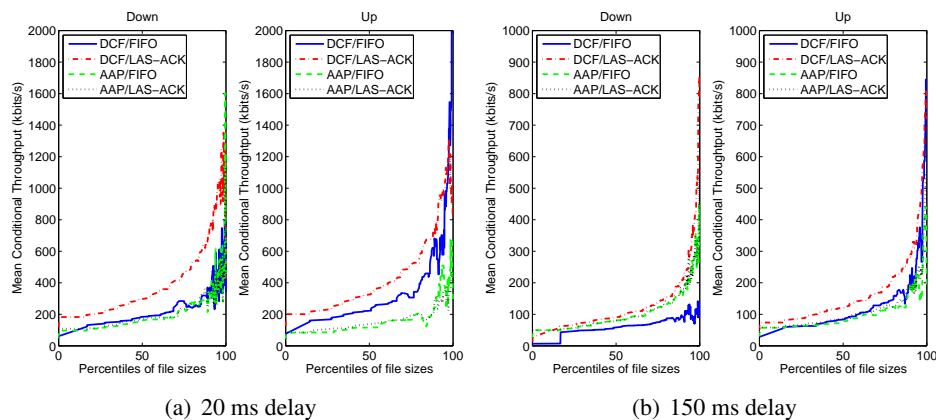


Figure 8: Conditional throughput for asymmetric load and Pareto distributed connection size.

*Observation 4: DCF/LAS-ACK consistently achieves the best per connection throughput for almost all connection sizes.*

Figure 8 presents the conditional throughput in function of the percentiles of connection sizes. We can observe that long uploads under DCF get higher throughput than downloads, while DCF/LAS-ACK achieves a consistent better performance for almost all connection sizes. For 20 ms delay, TCP connections under DCF/LAS-ACK obtain roughly twice the throughput of DCF/FIFO.

*Observation 5: Loss rate is important for DCF/FIFO while it stays reasonable for all other mechanisms, e.g. for 20 ms delay, the loss rates is 6% under DCF/FIFO and around 1% for all other mechanisms.*

Table 1 presents the loss rate for different mechanisms. We can observe significant loss rate for DCF/FIFO and low values for all other mechanisms. LAS-ACK

is beneficial to loss rate, because it gives priority to short connections that are generally unresponsive to losses—they do not adapt to congestion, but only delay their transmissions. Hence, by quickly servicing such connections, LAS-ACK makes the buffer of the access point shorter than in the case of DCF/FIFO (cf. Figure 7(b)) and thus limits the probability of losing packets. Similarly, AAP based policies quickly drain the buffer at the access point, which lowers loss rate.

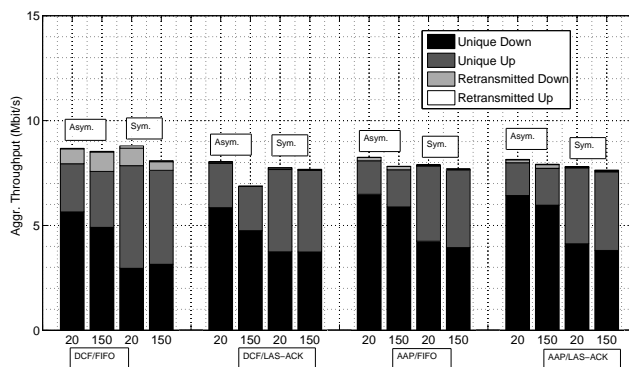


Figure 9: Measured aggregated throughput, Pareto distributed connection size.

*Observation 6: All considered mechanisms obtain almost the same level of aggregated throughput.*

Figure 9 shows the aggregate throughput of all mechanisms. Overall, we observe that all mechanisms have a similar efficiency at the network layer when looking at the aggregate throughput of uploads and downloads. When focusing on the relative share of uploads and downloads for the symmetric and asymmetric case, the unique advantages or drawbacks of each mechanism becomes visible again. DCF/FIFO consistently favors uploads at the expense of downloads, which becomes especially visible for symmetric traffic. DCF/LAS-ACK aims at evenly sharing the resources between uploads and downloads due to the way it treats the TCP ACKs from the uploads. As a consequence, under a symmetric workload, uploads and downloads achieve the same overall throughput. AAP based mechanisms tend to consistently favor downloads at the expense of uploads, which prevents them to achieve the same global rate even in the symmetric case.

We remark that the throughput plots for DCF/FIFO do not show an unfairness between uploads and downloads as marked as with long lived connections. We would have expected to observe a substantially higher rate for uploads than for downloads. The reason why download traffic is not shut down by the upload traffic is that the download traffic does not consist of persistent TCP flow, but rather of a large number of transfers, most of them being small. Those small TCP transfers are more aggressive than long transfers would be, and thus the unfairness problem is less obvious in terms of throughput shares. In our experiments, the proportion

of TCP packets that do not carry any information is never less than 40%, and this minimum is attained by DCF/FIFO, whereas long TCP connections generate only one third small packets due to the delayed ACK mechanism.

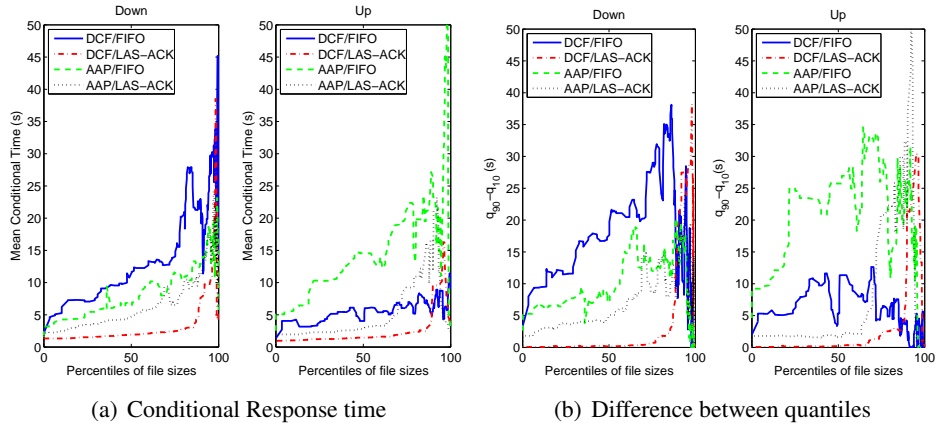


Figure 10: Asymmetric load and exponentially distributed connection size, 150ms delay.

*Observation 7: even for exponentially distributed connections sizes, LAS-ACK provides significant performance improvement in terms of the conditional response time, variability, packet delay, and loss rate. Nevertheless, the performance improvement comes at the cost of starvation of the longer connections.*

We consider the exponentially distributed workload to stress LAS-ACK based policies and see whether they are still able to improve performance in our experimental set up. Figure 10(a) presents the conditional response times for this workload and 150 ms delay. Note that now 100 percentile corresponds to 246 MSS.

The relative performance of all four combinations is quite similar the case of the Pareto distribution, although some aspects are more pronounced. In particular, the TCP unfairness problem is more visible for DCF/FIFO and DCF/LAS-ACK still provides very low response times as well as very low variability for most of the transfer sizes. However, this comes at some cost: long connections starve under DCF/LAS-ACK. Figure 11 presents the cumulative distribution functions of completed connections that clearly show how LAS-ACK privileges short connections at the expense of the large ones (the input dots show the cdf of the transfer sizes, cf. Figure 2). This result is in line with theoretical results on LAS showing that it is detrimental to large flows when the variability of the flow size distribution is small [23]. Note that it was not the case for the Pareto workload—the cumulative distribution functions of finished connections for all policies fully overlap (cf. Figure 12).

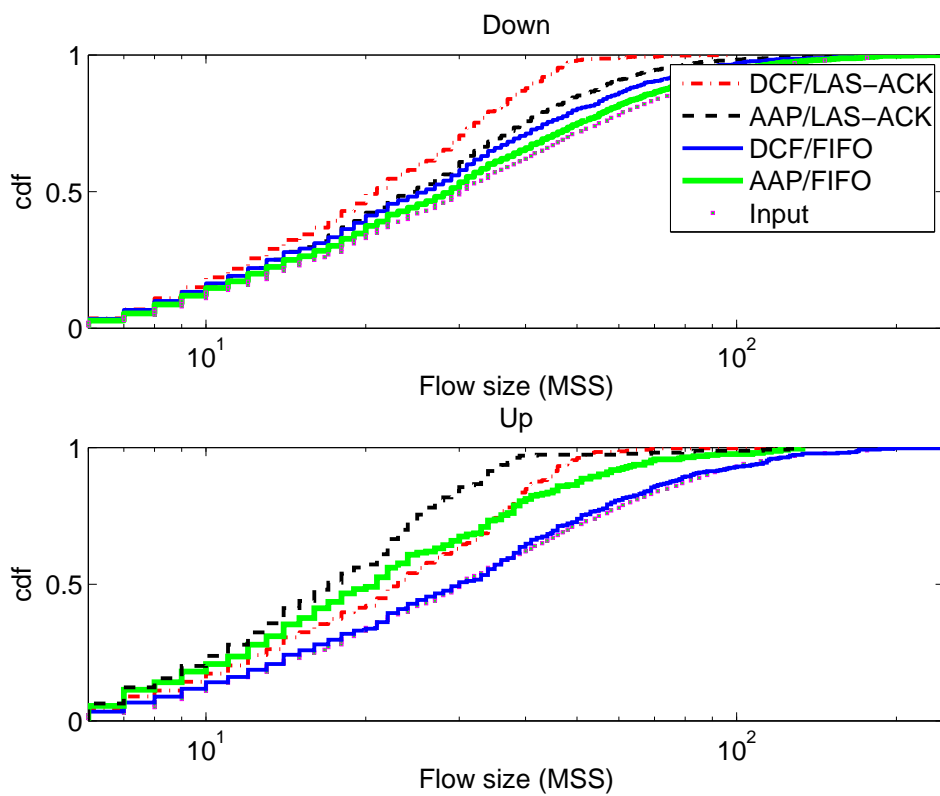


Figure 11: Cumulative distribution functions of the sizes of completed connections for asymmetric load and exponentially distributed connection size, 20ms delay.

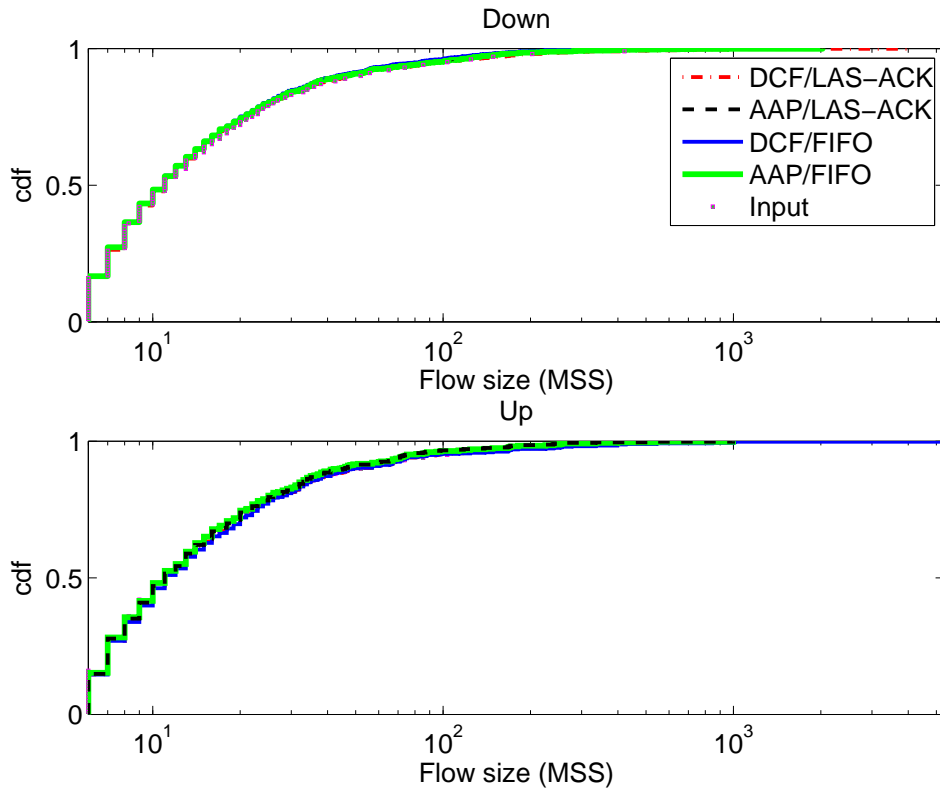


Figure 12: Cumulative distribution functions of completed connections for asymmetric load and Pareto distributed connection size, 150ms delay.

The performance of DCF/LAS-ACK depends the most on the connection size distribution, because LAS performs better for higher CoV of the connection size distribution [23]. This means that in our experiments DCF/LAS-ACK achieves the best performance under the Pareto workload. This is also the case for AAP/LAS-ACK, even though we notice smaller sensitivity of AAP/LAS-ACK to the connection size distribution compared with DCF/LAS-ACK. Hence, AAP/LAS-ACK not only obtains intermediate results between DCF/LAS-ACK and AAP/FIFO in general (whatever the metrics), but also it achieves more predictable performance.

Figures 10(a) and 5(a) (and also Figures 10(b) and 5(b)) show that the exponentially distributed workload implies longer response times (and a longer queue at the access point) compared to the Pareto distribution, because there are much more short connections in the exponential workload. As they mostly operate in the slow start state in which TCP sources are more bursty, the response times are longer. We can see from Figure 13 that the exponentially distributed workload is more bursty traffic in the case of the DCF/FIFO policy under asymmetric load: inter-arrival times tend to be smaller, which means more bursty traffic.

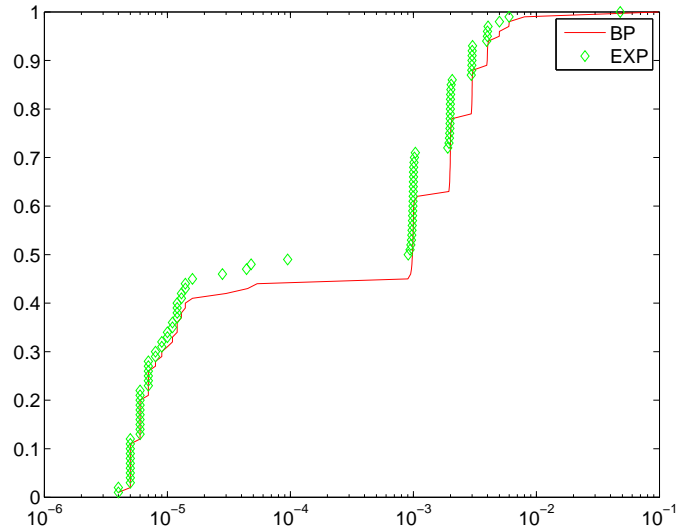


Figure 13: Inter-arrival times at the access point queue for asymmetric load and Pareto distributed connection size, DCF/FIFO policy

Table 2: Packet loss rate, exponentially distributed connection size

	DCF FIFO	DCF LAS-ACK	AAP FIFO	AAP LAS-ACK
Asym., 20ms	25%	0%	2%	1%
Asym., 150ms	26%	2%	4%	3%
Sym., 20ms	31%	0%	0%	0%
Sym., 150ms	29%	0%	2%	2%

The exponentially distributed workload also shows that under heavy load and DCF/FIFO, connections may take up to tens of seconds to complete, because loss rate is so high that connection establishment may take a long and variable time. Other policies efficiently mitigate this effect. The only exception is AAP/FIFO that saturates the output buffer of wireless stations, which leads to large transfer times for uploads, even though the response times of uploads under AAP/FIFO is similar to the ones of downloads under DCF/FIFO. In fact there is a tradeoff between an efficient use of the wireless network represented by the aggregated throughput and the response time, which is more a user oriented performance measure. In other words, DCF/LAS-ACK discards or slows down many connections, while under DCF/FIFO, the largest downloads take tens of seconds, which may be prohibitive from the user point of view.

Table 2 presents the loss rate for the exponentially distributed workload. We can observe higher loss rate for almost all cases and a significant increase for DCF/FIFO.

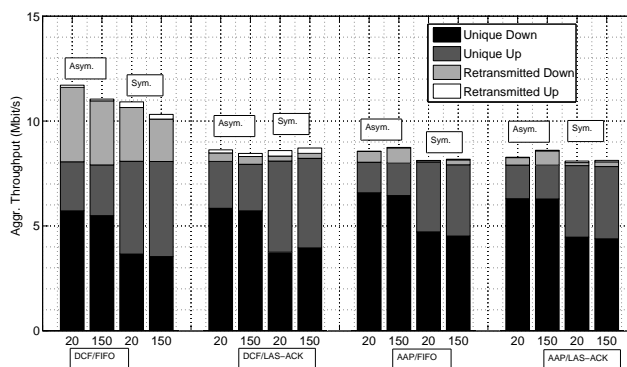


Figure 14: Measured aggregated throughput, exponentially distributed connection size.

Figure 14 show the aggregate throughput for the exponentially distributed workload. It stays at the similar level for all mechanisms, but we can observe an increased rate of retransmitted segments for DCF/FIFO.

## 6 Conclusions

The fundamental issue in 802.11 wireless LANs stems from the downlink packet queue that builds up when the access point does not benefit from sufficient radio channel capacity. This problem has two major consequences: it severely impacts the reactivity of short connections and interactive applications as well as results in significant unfairness between uploads and downloads. The two considered approaches (AAP and LAS-ACK) both attempt to drain the buffer along different angles of attack. AAP gives enough priority to the access point so that previously saturated buffer shared by stations disappears. However, the bottleneck moves to stations and uplink queues build up, which can only be mitigated by a suitable queueing strategy at layer 3. Applying LAS-ACK at the access point decreases the queue as expected and significantly improves performance especially for a realistic flow size distribution. However, its behavior can be very aggressive towards longer flows depending on the statistical characteristics of the transfer size distribution.

Taking into account three dimensions of the problem: reactivity, overall performance, and complexity of the solution, it appears that the combination of LAS-ACK and AAP offers the best tradeoff. It significantly improves performance by giving the proper priority to the access point and by properly scheduling flows sent by wireless stations. This combination also makes the access point fairly simple: it needs to operate with a constant contention window (fixed according to the AAP method) and schedule packets using standard FIFO.

## References

- [1] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of a Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in *Proc. of IEEE INFOCOM 2002*, vol. 2, June 2002, pp. 599–607.
- [2] X. Wang and S. A. Mujtaba, "Performance Enhancement of 802.11 Wireless LAN for Asymmetric Traffic Using an Adaptive MAC Layer Protocol," in *Proc. of IEEE VTC-Fall 2002*, vol. 2, September 2002, pp. 753–757.
- [3] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," in *Proc. of IEEE INFOCOM 2003*, vol. 2, March-April 2003, pp. 863–872.
- [4] M. Bottigliengo, C. Casetti, C. Chiasserini, and M. Meo, "Smart Traffic Scheduling in 802.11 WLANs with Access Point," in *Proc. of IEEE VTC-Fall 2003*, vol. 4, October 2003, pp. 2227–2231.
- [5] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," in *Proc. of ACM MOBICOM 2000*, August 2000, pp. 167–178.
- [6] A. Banchs and X. Perez, "Distributed Weighted Fair Queueing in 802.11 Wireless LAN," in *Proc. of IEEE ICC 2002*, vol. 5, April-May 2002, pp. 3121–3127.
- [7] D. Leith and P. Clifford, "Using the 802.11e EDCF to Achieve TCP Upload Fairness Over WLAN Links," in *Proc. of IEEE WiOpt 05*, April 2005, pp. 109–118.
- [8] D. Leith, P. Clifford, D. Malone, and A. Ng, "TCP Fairness in 802.11e WLANs," *IEEE Communications Letters*, vol. 9, no. 11, pp. 964–966, November 2005.
- [9] E. Lopez-Aguilera, M. Heusse, Y. Grunenberger, F. Rousseau, A. Duda, and J. Casademont, "An Asymmetric Access Point for Solving the Unfairness Problem in WLANs," *IEEE Transactions on Mobile Computing*, 2008, preprint. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/TMC.2008.44>
- [10] IEEE, "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
- [11] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," in *Proc. of ACM SIGCOMM 2005*, vol. 35, no. 4, August 2005, pp. 121–132.



- [12] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, "Experience with an Implementation of the Idle Sense Wireless Access Method," in *Proc. of CoNEXT 2007*, 2007.
- [13] I. A. Rai, G. Urvoy Keller, and E. W. Biersack, "Analysis of LAS scheduling for job size distributions with high variance," in *ACM SIGMETRICS 2003, International Conference on Measurement and Modeling of Computer Systems, June 10-14, 2003, San Diego, USA*, Jun 2003.
- [14] P. Starzetz, M. Heusse, F. Rousseau, and A. Duda, "Virtual Flow Queueing for Improving TCP Performance over IEEE 802.11 WLANs," in *Proc. of IEEE WCNC 2008*, Las Vegas, March-April 2008.
- [15] D. Eckhardt and P. Steenkiste, "Effort-limited Fair (ELF) Scheduling for Wireless Networks," in *Proc. of IEEE INFOCOM 2000*, vol. 3, March 2000, pp. 1097–1106.
- [16] M. Heusse, P. Starzetz, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Bandwidth Allocation for DiffServ Based Quality of Service over 802.11b," in *Proc. of IEEE GLOBECOM 2003*, vol. 2, December 2003, pp. 992–997.
- [17] D. Skyrianoglou, N. Passas, and A. Salkintzis, "Support of IP QoS over Wireless LANs," in *Proc. of IEEE VTC-Spring 2004*, vol. 5, May 2004, pp. 2993–2997.
- [18] J. Ha and C.-H. Choi, "TCP Fairness for Uplink and Downlink Flows in WLANs," in *Proc. of IEEE GLOBECOM 2006*, November 2006.
- [19] A. Banchs, A. Azcorra, C. Garcia, and R. Cuevas, "Applications and Challenges of the 802.11e EDCA mechanism: an experimental study," *IEEE Network*, vol. 19, no. 4, pp. 54–58, July/August 2005.
- [20] L. Stewart and J. Healy, "Tuning and testing the FreeBSD 6 TCP stack," Centre for Advanced Internet Architectures, Melbourne, Tech. Rep. 070717B, July 2007. [Online]. Available: <http://caia.swin.edu.au/reports/070717B/CAIA-TR-070717B.pdf>
- [21] X. G. Meng, S. H. Y. Wong, Y. Yuan, and S. Lu, "Characterizing flows in large wireless data networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 174–186.
- [22] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 15–28.
- [23] M. Nuyens and A. Wierman, "The foreground-background queue: A survey," *Perform. Eval.*, vol. 65, no. 3-4, pp. 286–307, 2008.