Per-Station Throughput Fairness in a WLAN Hot-Spot with TCP traffic

A. Detti*, N. Blefari-Melazzi*, I. Habib**, A. Ordine*
*University of Rome "Tor Vergata", {andrea.detti, blefari, alessandro.ordine}@uniroma2.it
**Department of Electrical Engineering, City University of New York, habib@ccny.cuny.edu

***Abstract*— The 802.11 DCF protocol was devised to achieve per-station fairness; conversely, an harsh per-station unfairness occurs when DCF is loaded by TCP traffic. We aim at extending theoretical knowledge on this fairness issue and at restoring per-station fairness. The contribution of the paper is threefold. First, we introduce two findings: i) changing the physical transmission rate of the stations does not impact the per-station fairness; ii) packet queuing occurs in the uplink buffer of a station and has a non-negligible impact on the fairness. Second, we derive a model for evaluating fairness performance. Our model accounts for the effect of several parameters and scenarios that no other model captures in a single analytic framework. Third, and finally, we propose a technique to enforce fairness, which is easy to deploy in real systems, without having to modify existing devices. Experimental results obtained with commercial devices confirm the validity of our solution and findings. Albeit the focus of this paper is on per-station (throughput) fairness, our technique can be easily modified e.g. to enforce time-based fairness, or per-direction or per-flow fairness.**

1 INTRODUCTION

A typical installation of an IEEE 802.11 Wireless LAN (WLAN) is called "hot-spot" (Fig. 1, upper part) and it is composed of an Internet Gateway (GW), an Access Point (AP) and wireless stations (STAs). The Gateway and Access Point are connected through a private network, usually an Ethernet. The Gateway takes care of IP forwarding between the public Internet and the private network. The Access Point performs frame switching between the Ethernet and the wireless stations. The Wireless Medium Access Control (MAC) is based on the 802.11 DCF protocol [1].

This paper deals with the issue of "per-station" throughput fairness. Perfect per-station throughput fairness occurs when all stations access the wireless channel with equal data rates [25], [27]. Per-station throughput fairness has been a design driver of DCF protocol; indeed, in case that a set of greedy stations access the channel, they fairly share transmission opportunities and (for equal packets' lengths) they obtain the same throughput. Conversely, when the data traffic is non-greedy but it is regulated by Transmission Control Protocol (TCP), the throughput achieved by a station strongly depends on the configuration of the station's TCP connections (i.e., number of connections, data direction, receiver window size). Hence stations with different configurations may achieve different throughputs. For instance, let us consider the simple case of a WLAN with two stations both operating at 11 Mbit/sec but with a different number of connections. Station A is performing a TCP download from a local wired server, whereas station B is performing two TCP downloads from the same server. The overall throughput achieved by station A is about 1.6 Mbit/sec versus the 3.2 Mbit/sec of station B. Therefore stations do not enjoy a fair throughput service. Our objective is to remedy this shortcoming.

It is plain having different opinions on what is the best fairness achievement: i.e., “per-station” fairness, “per-flow” fairness [2] or “per direction” fairness [6] or “time-based fairness” [7], [8]. In this work, we deal with per-station fairness because we simply believe that such a kind of farness is the closer one with the original DCF expectation. However, our technique to restore fairness can be easily modified to enforce time-based, per-direction or per-flow fairness.

The structure of this paper is as follows. In section 2, we recall the phenomena that cause unfairness. We also introduce two novel results that are important to understanding fairness in two specific scenarios: stations with different physical transmission rates, and stations with multiple upstream TCP connections. In section 3, we propose an original analytical method to evaluate the “per-station fairness level (i,j)” which is defined as the ratio between the useful data-rate of station (i) and that of station (j). However, our model is able to evaluate also other fairness indexes, such as up-down fairness index [2] and Jain’s fairness index [3] (see Appendix III of [17]). The need for this model is mainly motivated by the fact that existing analytical models (e.g., [2][4][5][23]) do not consider 802.11 multi-rate environments. They are also not adequate to analyzing stations with multiple upstream TCP connections since those models neglect the occupancy of the MAC (uplink) buffer contained in each station [28].

In Section 4, we use our understanding of the performance of fairness, obtained via our model, to design a viable technique for enforcing per-station throughput fairness, named Virtual Shared Bottleneck (VSB). The need for a new technique stems from the fact that existing proposals are rather difficult to deploy: such proposals require either changes in the Access Point [26], or the wireless station ([25] and [27]) or both [21]. On the contrary, our VSB is a software module that is logically located between the Gateway and the Access Point (see Fig. 1 lower part). It thus enforces fairness without any changes in existing devices. In section 5, we present an experimental test-bed that we used to confirm the validity of our analytical model and of our proposed VSB solution and provide some numerical results. Furthermore, we show that our mechanism does not waste WLAN bandwidth. In section 6, we compare our work vis-à-vis other related works in this area in order to emphasize and further clarify the significance of our contribution. In section 7 we draw our conclusions.

2 State-of-the-Art and Two New Results

We present the state-of-the-art knowledge on fairness in WLANs by using experimental measurements obtained in a WLAN without fairness enforcement (see Fig. 1.a). We consider five mobile stations having active TCP connections with a Fixed-Host. The TCP receiver window of all connections is the default one of Microsoft Windows Vista [12] (about 64 kbytes). The Access Point is a Cisco Aironet 1200 with a downlink buffer space allocated by the operating system of 75 packets[1]. The physical transmission rate is 11 Mbit/sec for all stations unless otherwise stated. We focus our analysis on the WLAN thus we neglect the eventual impact of fixed network characteristics (loss and delay) on the TCP performance, as commonly done in the literature; the Fixed-Host/Gateway and Gateway/AP connections are realized by means of Ethernet cables. We examine four different TCP connection scenarios:

1. upstream-vs-downstream (UP vs DW): station 1 has one upstream TCP connection (i.e., TCP data transfer from the wireless station to Fixed-Host); the other four stations have one downstream connection each.
2. downstream-vs-downstream (DW vs DW): station 1 has six downstream connections; the other four stations have one downstream connection each.
3. upstream-vs-upstream (UP vs UP): station 1 has six upstream connections; the other four stations have one upstream connection each.
4. downstream-vs-downstream multirate (DW vs. DW multirate): station 1 has six downstream connections; the other four stations have one downstream connection each; in this case the stations have different physical transmission rates (details are in the following sub-section).

In the upper half of Fig. 2 we plot the *goodput* of each of the five stations in the four different scenarios. The goodput of a station is the sum of the average useful throughputs of all TCP connections active on that station. In the lower half of Fig. 2 we plot a measurement of the *fairness level* of station 1 vs. the other stations in the four different scenarios. We define the fairness level of station 1 as the ratio between the goodput of station 1 and the goodputs of the station *#id* as reported in abscissa.

The first macroscopic observation is that station 1 always enjoys more goodput than other stations (upper plots); its fairness level (lower plots) ranges from 2 up to 6. We explain this observation in the following. We start the discussion with the upstream-vs-downstream case since it has already been thoroughly analyzed in the literature (e.g., [2][4][5][6][15][21][22][23]). The literature agrees on the following three properties that constitute the state-of-the-art knowledge on this phenomenon:

Property 1: the Access Point downlink buffer is the network bottleneck; the MAC 802.11 gives to the Access Point the same channel access opportunity of any wireless station, even if the Access Point has to handle more traffic than any single station. This characteristics is called "MAC-induced unfairness" in [21] and leads to the following consequences [1].

Property 2: most packet loss occurs in the Access Point buffer.

Property 3: the Access Point queuing delay makes up a large part of the overall TCP round-trip (having neglected the eventual fixed network delay).

A consequence of these properties is that all TCP connections experience almost the same round-trip-time delay and the same packet loss. However, when upstream flows lose a packet in the Access Point buffer their data-rate is not significantly reduced since the loss affects a TCP ACK. On the contrary, when downstream flows lose a packet in the Access Point buffer their data-rate is reduced since the loss affects a TCP segment. This asymmetry implies that upstream flows starve downstream flows when packet loss occurs in the Access Point. This phenomenon is the well-known [2] upstream/downstream unfairness; it is called "TCP-induced unfairness" in [21], and it is exactly what happens in our upstream-vs. downstream scenario: the Access Point is losing packets as station 1, which is uploading traffic, perceives a higher data-rate than the downloading stations.

In the downstream-vs-downstream scenario all TCP connections are downloading traffic; these

---

[1] The physical memory size of the AP is 16 MB, but accessing the OS through a serial connection we find an interface queue of 75 packets.

connections all experience the same round-trip-time and the same segment loss [11] and consequently they perceive the same data-rate. In this scenario the system offers "per-flow" fairness: station 1 has 6 connections and its goodput is six times greater than the one perceived by each of the other stations that have only one connection active each. This behavior is explained by the three properties reported above.

In the upstream-vs-upstream case, the three properties reported above would imply that the goodput of station 1 should be *six* times greater than the one perceived by each of the other stations. Furthermore, according to these properties, all connections should experience the same round-trip-time and no "segment" loss thus leading to per-flow fairness. On the contrary, Fig. 2 tells us that this is not true: the goodput of station 1 is about *two* times greater than the one perceived by each of the other stations. We found out that the reason of this behavior [28] is that property 3 is not always true; when a station has a significant number of upstream connections, the occupancy of its uplink buffer may assume large values, even greater than that of the Access Point. Therefore, it is not correct to neglect the occupancy of the station buffer and its contribution to the overall TCP round-trip-time, as was done in the literature, e.g., see [2][4][5][23]. For instance, in this upstream-vs-upstream scenario, the average buffer occupancy of station 1 is about 170 packets, whereas that of the Access Point is only 75 packets. The TCP connections of station 1 experience a round-trip-time significantly longer than the TCP connections of other stations whose uplink buffer is practically empty. Since station 1 has 6 connections its goodput tends to be larger than that of the other stations; on the other hand, each of station 1's connections has a data-rate lower than those of the other stations because of their longer round-trip-time. The outcome of these two contrasting effects is a fairness level of 2, instead of the expected value of 6 [2].

The last scenario is the downstream-vs-downstream multi-rate; station 1 has a physical transmission rate of 2 Mbit/sec, while the other stations' rate is 11 Mbit/sec. As expected [16], the goodputs of all stations are seriously degraded by the presence of the low-rate station. However, if we look at the fairness level we observe that the fairness level of the downstream-vs-downstream scenario in which all stations have a rate of 11 Mbit/sec is almost equal to the fairness level of this multi-rate environment. This means that the fairness level does not vary as a function of the physical transmission rates. Although this is not a surprising result for the case of greedy traffic [16], in case of TCP traffic such a behavior has never been presented before and, as it occurs for per-station fairness, it is not obvious that something occurring in case of greedy traffic, occurs in case of TCP traffic too.

The analysis of the latest two scenarios implies that we have to reformulate the 3rd property quoted above and add a fourth property:

Property 3: The TCP round-trip-time is made up mainly of the queuing delay of the Access Point *plus* the delay encountered in the buffer of the station. The latter contribution is *not* negligible when the station has several upstream connections;

Property 4: The MAC physical transmission rate does not affect the fairness performance (be it measured with our fairness level, up-down fairness index [2] or Jain's fairness index [3]).

3 ANALYTICAL MODEL

In this section we present a novel analytical model to evaluate the fairness performance; the model results will be discussed and compared to experimental measurements in Section 5.

The analytical model uses the following inputs: i) the AP downlink buffer size (B) and, for each wireless station i, ii) the number of upstream (Nup_i) and iii) downstream (Ndw_i) connections, iv) the receiver window of TCP connections active on that station (W_i); v) the channel access priority of that station with respect to the Access Point (β_i). The output of the model is the fairness level $\eta_{i,j}$ between station #i and station #j, which is defined as the ratio between the goodput of station #i and the one of station #j. The goodput of the station #i is defined as the sum of the average useful data-rates of all its connections. We believe that the adopted merit figure – the fairness level - is very effective in explaining our test-bed results; however, our methodology allows easily deriving also other performance figures such as those defined in [2] and [3] (see Appendix III of [17]).

Our model is flexible enough to analyze all possible configurations of TCP connections arising in a WLAN. In this sense, our model extends the seminal one proposed in [2], which is able to analyze only the case of one connection per station with all connections having the same TCP receiver window. Another advantage of our model is that it takes into account the occupancy of the uplink buffer of the

---

[2] We observe that in the downlink-vs-downlink scenario no queuing occurs on the stations, so the fairness level is 6.

wireless station, instead of neglecting it, as commonly done in the literature [2][4][5][23]. This feature requires an innovative treatment that makes our model significantly different from the one in [2].

To make the analysis easily tractable, but still accurate, we make some simplifying assumptions, which are reported in Tab. 1, together with their consequences on the accuracy of the model.

The long-term steady-state goodput of the i-th STA (segments per second) is equal to the ratio between average useful number of segments exchanged by TCP connections during a round-trip-time and average round-trip-time:

$$GPsta_i = \frac{Nup_i \cdot W_i}{E[D_i] + E[D_{ap}]} + \frac{Ndw_i \cdot Tn(p, W_i)}{E[D_i] + E[D_{ap}]} \tag{1}$$

The average round-trip-time is the sum of the average queuing delay suffered in the AP, E[D_{ap}], and in the uplink station buffer, E[D_i]. The average queuing delay E[D_{ap}] (E[D_i]) is the average time interval between the instant in which a packet is stored in the AP (STA) buffer and the instant in which the MAC ACK of that packet is received by the AP (STA). This parameter includes all relevant MAC layer delays.

The first addend of Eq. (1) is the part of goodput due to the Nup_i upstream connections. As a consequence of assumptions a1, a2 and a7, these connections do not experience segment loss and fully open their congestion window up to the receiver window W_i. Thus, upstream connections exchange W_i segments every round-trip-time. The second addend of Eq. (1) is the part of goodput due to the Ndw_i downstream connections, which experience a segment loss probability p on the AP buffer; thus, the average number of the useful segments exchanged in a round-trip-time by a connection is less than W_i and equal to a quantity which is a function of W_i and p, denoted as $Tn(p, W_i)$. The expression $Tn(p,W)$ is derived by Eq. (37) of [11] multiplied by RTT, and assuming T_0=RTT (see assumption a6). At this point the fairness level η_{ij} between station i-th and station j-th can be written as:

$$\eta_{ij} = \frac{GPsta_i}{GPsta_j} = \frac{Nup_i \cdot W_i + Ndw_i \cdot Tn(p, W_i)}{Nup_j \cdot W_j + Ndw_j \cdot Tn(p, W_j)} \cdot \frac{E[D_j]/E[D_{ap}] + 1}{E[D_i]/E[D_{ap}] + 1} \tag{2}$$

As discussed in [28], the ratio E[D_i]/E[D_{ap}] is equal to the ratio Q_i/Q_{ap}, where Q_{ap} is the average occupancy of the AP buffer and Q_i is the average occupancy of the i-th STA buffer. This result is due to the fact that, since the MAC layer operates as a per-packet fair scheduler among backlogged devices (see assumption a7 in Tab. 1), all backlogged stations and the AP obtain the same packet-rate λ, independently of their transmission rate [16][32]. From Little's law [29] $Q_i = E[D_i]\lambda$ and $Q_{AP} = E[D_{AP}]\lambda$; therefore $E[D_i]/E[D_{ap}] = Q_i/Q_{ap}$.

It is worth noting that the equality $E[D_i]/E[D_{ap}] = Q_i/Q_{ap}$ has an important consequence: the fairness level η_{ij} is independent of the WLAN physical transmission rate (see Property 4 in section 2). If we substitute $E[D_i]/E[D_{ap}]$ with Q_i/Q_{ap}, we remove from Eq. (2) the dependence from the parameters $E[D_{ap}]$ and $E[D_i]$, which are the only parameters related to the WLAN physical transmission rate. As a matter of fact, the queue sizes Q_i and Q_{ap} do not depend on any temporal parameters (as can be seen from the following Eqs. (4) and (5)).

The equality $E[D_i]/E[D_{ap}] = Q_i/Q_{ap}$ holds if all devices behave in the same way and fully respect standard protocols. However, the results that we obtained in a real environment show that devices produced by different manufacturers have different "priority" in accessing the wireless medium, in contrast with 802.11 specifications. These differences are due to behavior of both the device driver and the firmware that implements the 802.11 baseband functions, an explanation of this phenomenon is provided in [10]. To take into account these differences among devices produced by different manufacturers we introduce a correction factor β_i, which is defined as the ratio between the medium access probability of the i-th STA and the medium access probability of the AP. If β_i is greater than one, then the i-th STA has a priority greater than the AP and vice versa.

The equality $E[D_i]/E[D_{ap}] = Q_i/Q_{ap}$ must then be rewritten as: $E[D_i]/E[D_{ap}] = Q_i/(\beta_i Q_{ap})$. It follows that Eq. (2) has to be rewritten as:

$$\eta_{ij} = \frac{GPsta_i}{GPsta_j} = \frac{Nup_i \cdot W_i + Ndw_i \cdot Tn(p, W_i)}{Nup_j \cdot W_j + Ndw_j \cdot Tn(p, W_j)} \cdot \frac{Q_j/(\beta_j Q_{ap}) + 1}{Q_i/(\beta_i Q_{ap}) + 1} \qquad (3)$$

It now remains to evaluate Q_i, Q_{ap} and p. In [28] we evaluated these parameters in a simplified scenario where we did not consider: i) the TCP delayed-ack mechanism, which is instead implemented in most operating systems; ii) the factor β_i, iii) the fact that different stations may have different receiver windows W_i. In this paper we remove these limitations. The new evaluation of Q_i, Q_{ap} and p is reported in

Appendix II of [17]. In the main text that follows we report only the final results.

We evaluate Q_i, Q_{ap} and p in two different cases: *lossless* AP buffer and *lossy* AP buffer obtaining two sets of formulas. To decide which set of formulas is to be used in a given situation, first we evaluate Q_{ap} by means of the *lossless* set; if Q_{ap} is less than the AP buffer size, B, then we use the *lossless* set of formulas; otherwise we use the *lossy* set of formulas.

The evaluation of the *lossless* set of formulas requires the numerical solution of a system of 2*M+1 equations (4) (for all the possible value of i, $1 \le i \le M$):

$$\begin{aligned}
Qup_i &= \max\left(0, Nup_i \cdot W_i - \frac{Nup_i}{Nup_i + 0.5 \cdot Ndw_i}\beta_i \cdot Qap\right)\\
Qdw_i &= \max\left(0, \frac{Ndw_i \cdot W_i}{2} - \frac{Ndw_i}{2 \cdot Nup_i + Ndw_i}\beta_i \cdot Qap\right)\\
Q_{AP} &= \sum_{i=1}^{M}\left[\left(\frac{Nup_i \cdot W_i - Qup_i}{2}\right) + \left(Ndw_i \cdot W_i - 2 \cdot Qdw_i\right)\right]
\end{aligned} \tag{4}$$

where M is the number of wireless stations and Qdw_i (Qup_i) is the average number of packets belonging to downstream (upstream) connections stored in the buffer of the i-th station; therefore, $Q_i = Qdw_i + Qup_i$.

The evaluation of the *lossy* set of formulas requires the numerical solution of a system of 2*M+1 equations:

$$\begin{aligned}
p &= 1 - \frac{B}{\displaystyle\sum_{i=1}^{M} \frac{0.5 \cdot (Nup_i \cdot W_i) + Ndw_i \cdot NSR(p, W_i)}{1 + \frac{Q_i}{\beta_i \cdot B}}}\\
Q_i &= \max\left\{0, Nup_i \cdot W_i + \frac{1}{2} \cdot Ndw_i \cdot NSR(p, W_i) \cdot (1-p) - \beta_i \cdot B\right\} \cong \max\{0, Nup_i \cdot W_i - \beta_i \cdot B\}
\end{aligned} \tag{5}$$

where *NSR(p,W)* is the number of segments emitted by a TCP connection during a RTT in presence of a segment loss probability p, with a receiver window W and in presence of delayed ACKs. *NSR(p,W)* is derived by Eq. (32) of [11], with T_0=RTT and b=2.

We conclude with two important considerations: i) queuing phenomena in the STA buffer is essentially due to upstream connections; ii) in the AP lossy case the buffer occupancy of a STA is greater than zero only when $Nup_i\ W > \beta_i\ B$ (see Eq. (5)); in the AP lossless case the buffer occupancy of a STA is greater than zero when the upstream segments of that STAs are a considerable fraction of the overall traffic (see

Condition (14) in Appendix II of [17]).

4 Restoring Per-Station Fairness

4.1 The key-idea

TCP data-rate depends on the resources available in the network bottleneck, which in our case is the WLAN. Thus, it would be possible to control the TCP rate by implementing a suitable scheduler in the AP. Since we want to avoid modifying existing hardware and software, and remain back compatible with installed devices, we introduce the so-called Virtual Shared Bottleneck (VSB), located between the Gateway and the AP (see Fig. 1, lower part). As depicted in the upper part of Fig. 3, the VSB forces uplink and downlink traffic to share a common *virtual interface* whose output bit-rate is a bit lower than the WLAN capacity. Therefore, the network bottleneck is "moved" from the WLAN to the virtual interface. This is very handy, as we can freely and easily implement the scheduling policy most fit for our purposes at the virtual interface (e.g., by using a Linux box), instead of modifying AP or stations[3].

In this paper we exploit this idea to provide per-station fairness and therefore we simply implement a per-station Fair Queuing (STA-FQ) packet scheduler at the virtual interface. If we want to provide time-fairness then we just need to implement a per-station Weighted Fair Queuing where the weights are proportional to the physical bit rates of the stations (that should be retrieved by the AP). If we want to provide "per-flow fairness" [2] then we just need to implement a per-flow FQ. If the goal is "upstream/downstream fairness" [6] (i.e. total upstream goodput equal to total downstream goodput), the scheduler will be a FQ between overall upstream and downstream packets[4].

We stress that thinking to a per-station fair queuing approach to restore per-station fairness is straightforward. Nevertheless, we believe being not trivial the proposal of *a common-queue for all the uplink and downlink traffic coming from a station* (Fig. 3 lower). This novel approach is an effective enabler for any choice of per-station fair queuing scheduler; indeed, it equalizes the different

[3] We note that if a set of APs and a Gateway are connected via a switch, then a single device implementing a "bank" of VSBs can be placed between the Gateway and the switch. Each VSB of the bank handles the traffic of a specific AP. To identify the traffic of a specific AP, the VSB device has to know the IP addresses of the STAs associated to the AP; then, by inspecting the IP header of entering packets it can decide if a packet belongs to a given AP or not. The knowledge of the IP addresses of the STAs associated with a given AP is obtained by retrieving from the AP either the IP addresses of the associated STA or their MAC addresses; in the latter case, the local arp cache is then used to map MAC addresses to IP addresses. For instance, in the case of a Cisco Aironet AP, the IP addresses of the STAs associated with an AP can be retrieved by accessing the CISCO-DOT11-ASSOCIATION-MIB (OID name ciscoDot11AssociationMIB) by means of SNMP.

aggressiveness of upstream and downstream traffic.

4.2 The Virtual Shared Bottleneck in a Linux box

In the lower part of Fig. 3 we show our Linux implementation of the Virtual Shared Bottleneck; all related documentation and software are available in [18]. We exploit the QDISC and FILTER traffic control tools of the iproute2 package. Ingress filters located in the two network interfaces (eth1, eth2) redirect the entering TCP downlink and uplink packets (as a function of the IP addresses) toward the virtual interface IFB0, which is an Intermediate Functional Block module of the Linux Kernel[5]. On the IFB0 we set up a FIFO queue of size 200 packets for each station; the FIFO queue assigned to each station stores both uplink and downlink packets of that station. The scheduling mechanism is a Hierarchy Token Bucket (HTB) [31]: there is a HTB *root* class that outputs packets at *C* Mbit/sec and each FIFO queue is associated to an HTB *leaf* class with guaranteed rate equal to *C/M*, where *M* is the number of STAs. Overall, this class structure implements a work-conserving, per-station, fair queuing discipline.

For the VSB to work, the overall bit rate *C* must be such that the WLAN remains not congested; thus the value of *C* must be estimated run-time so as to be slightly less than the WLAN capacity. This run-time-estimation is the job of the Wireless Capacity Estimator (WCE), described in 4.2.1.

4.2.1 The Wireless Capacity Estimator

The estimation of the available wireless capacity is not a major goal of this paper; however, we present a simple and effective algorithm based on the PING tool, which resembles the one used by TCP delay-based congestion control algorithms [33], like TCP Vegas [34][35].

A detailed stability and sensitivity analysis of this algorithm is out of the scope of this paper. The BASH code of our Wireless Capacity Estimator (WCE) is available in [18]; here we describe only the main concept of operation. The WCE periodically updates *C* in such a way the WLAN is not congested. The congestion status of the WLAN is deducted by measuring the round-trip-time between the VSB and a randomly-selected wireless station. The measurement is performed by sending PING messages, which do not pass through the VSB scheduler. If the round-trip-time sample overcomes a certain *RTT threshold*

---

[4] In Appendix V of the extended version of this paper [17]. we give an example of how the Linux VSB can be programmed to enforce different types of fairness.

(i.e., 40 ms), the algorithm concludes that the current value of *C* overloads the WLAN and then the value of *C* is linearly decreased (of a fixed amount at each step); otherwise *C* is linearly increased (of a fixed amount at each step). The *RTT threshold* represents the "base" RTT [35] that we would have in an unloaded WLAN between the AP and a STA. Furthermore, to speed up the convergence time, when the round-trip-time sample is greater than a *high RTT threshold* (i.e., 200 ms), the value of *C* is decreased more rapidly, by means of an exponential backoff mechanism. A symmetric mechanism is applied if the round-trip-time sample is smaller than a *low RTT threshold* (i.e., 5 ms).

In Fig. 4 we report an example of the output of the WCE measured in a real test-bed. Before time instant 25s, there is no traffic in the WLAN and the WCE returns its maximum possible value (set equal to 6 Mbit/sec). At the 25 seconds time instant, a station with physical transmission rate of 11 Mbit/sec starts a downstream TCP connection. The WCE timely senses an increase of the occupancy level of the AP buffer and decreases the VSB capacity to about 5.4 Mbit/sec. At time instant 57 s, another station starts a downstream TCP connection with a physical rate of 2 Mbit/sec. The performance-anomaly [16] phenomenon reduces the available WLAN transfer capacity and the AP buffer fills up. Thus, the WCE reduces the VSB capacity at about 2.4 Mbit/sec and the AP buffer empties out. When the connection at 2 Mbit/s is seizes, the WCE goes back to the 5.4 Mbit/s rate.

5 PERFORMANCE EVALUATION

We assess the accuracy of the analytical model described in Section 3 and the effectiveness of the VSB enforcer by means of an experimental test-bed made up of five wireless laptops (playing the role of user stations), a Cisco Aironet 1200 Access Point (having 75 packets of buffer space), and three PCs (playing the role of Fixed-host, Gateway and Virtual Shared Bottleneck) [6]. Wired links are realized by means of 100 Mbit/s Ethernet cables. The 802.11 transmission mode is 802.11b.

We consider three main connection scenarios, as done in Section 2: upstream-vs-downstream; downstream-vs-downstream and upstream-vs-upstream. For each scenario, we vary a specific TCP

[5] For scenarios with STAs supporting greedy UDP sources that last for a valuable amount of time, the VSB and the WCE logic are more complex, and they are discussed in Appendix IV of [17].

[6] We also succeeded in integrating the VSB module within an embedded Linux AP (MIPS CPU at 183 Mhz), based on the OpenWRT Kamikaze 7.05 distribution. The AP CPU architecture is MIPS at, the RAM is 16 MB and the Wi-Fi chip is Atheros AR2315. We do not report in this paper the results of the experiments performed with this equipment both for space limitations and for highlighting the fact that the VSB perfectly works outside the AP.

parameter of station 1 (i.e., number of connections or TCP receiver window) while maintaining that parameter fixed on the remaining four stations. This means that i) all fairness levels $\eta_{1,x}$ between station 1 and any other station x (x=2,3,4,5) are equal among themselves; ii) stations 2, 3, 4 and 5 are fair among themselves: $\eta_{y,x}=1$ for any x,y greater than 1. Consequently we will report only results regarding one of these fairness level, i.e. $\eta_{1,2}$ between station 1 and station 2.

The measurements are made by using the “iperf” tool (with a packet size of 1500 bytes). Each measurement lasts three minutes and is repeated ten times. The 95% confidence interval is reported in all plots; when it is not visible, it simply means that the interval is too small. With respect to the parameters β_i used in the model of Section 3, we observe that we use Wi-Fi cards of the same type; therefore $\beta_i=\beta$ for each value of i. To evaluate β we set up a greedy downstream UDP connection and a greedy upstream UDP connection, whose end-points are STA 1 and the Fixed Host; then we measure the ratio between the goodput of the upstream UDP and that of the downstream UDP. We repeated this measurement ten times and obtained an average value of about 1.2 [7].

In addition to the previous “laboratory” tests, in Section 5.4 we report the result of a “real” test carried out by using a public Internet server as fixed host.

5.1 Upstream versus Downstream

We consider a number of upstream TCP connections of station 1 varying from 1 to 6; the other stations have only one downstream connection; the TCP receiver window of all connections is equal to 42 packets (about 64 kbytes). All stations have a physical transmission rate of 11 Mbit/sec.

Fig. 5 shows $\eta_{1,2}$ versus the number of upstream TCP connections of STA 1. We plot three curves: i) *meas. 802.11*: measurements obtained in a WLAN without fairness enforcer (scenario depicted in Fig. 1, upper part); ii) *model 802.11*: analytical results obtained with the model presented in Section 3, with B=75 and β=1.2; iii) *meas. 802.11+VSB*: measurements obtained in a WLAN with the VSB fairness enforcer (scenario depicted in Fig. 1, lower part). As expected, without fairness enforcer, upstream

[7] We observe that we have been unable to analytically evaluate/fix this parameter because the firmware code of the Cisco Access Point and of the DLINK Wi-Fi adapter are not open-source. For this reason we had to resort to such a reverse-engineering measurement approach. Moreover, the value of β may also vary during a test, as a function of different WLAN condition (e.g., collision probability), even if this did not happen during our tests. If β varies, modeling results may slightly differ from measurements, since in the model we assume a static value of β.

connections of STA 1 perceive a goodput higher than the downstream connections of STA 2; the fairness level increases from 2 to 16 as the number of upstream connections of STA 1 increases from 2 to 16. This is an example of the upstream/downstream unfairness discussed in Section 2. More precisely, as the number of upstream connections of STA 1 increase, the goodput of STA 1 increases, the AP packet loss probability increases and the goodput of the downlink connection of STA 2 decreases. However, when the number of upstream connections of STA 1 is greater than or equal to three (see considerations at the end of Section 3), the uplink buffer of STA 1 starts to fill up: the buffer occupancy increases linearly with the number of uplink connections [28]. As the uplink buffer occupancy gets filled-up, the upstream connections of STA 1 experience a longer round-trip-time compared to the connection of STA 2. This effect counter balances the effect of the upstream/downstream unfairness on the fairness level. The curve *meas. 802.11+VSB* shows that the VSB attains the goal of providing a per-station fairness, i.e. $\eta_{1,2} = 1$. As shown in the following figures, this goal is achieved in all scenarios considered in the paper [8].

To show that our VSB does not waste WLAN bandwidth, we report in a box of Fig. 5 the "average cumulative goodput" obtained in a WLAN without fairness enforcer (labeled "GP 802.11") and the same parameter evaluated in a WLAN with the VSB fairness enforcer (labeled "GP 802.11+VSB"). The average cumulative goodput is calculated by summing all the goodputs of the stations relevant to each of the considered values of the x-axis (in the case of Fig. 5, for each value of the number of the upstream connections), we then take the average of all these sums. For the case of Fig. 5, the average cumulative goodput with VSB (4.9 Mbit/sec) is slightly lower than the case of a WLAN without fairness enforcer (5.1 Mbit/sec). This is because the VSB plays the role of an additional bottleneck and thus causes a (limited) loss of goodput. However, we will show that in some cases (sections 5.2.1 and 5.2.2) the VSB may even lead to small goodput gains.

5.2 Downstream versus Downstream

In this section we consider stations having downstream connections only and analyze the fairness level as a function of the number of connections and of the TCP receiver window size.

[8] We observe that when our fairness level is equal to one for all stations, then also Jain's index [3], $\eta_{ij} = \left(\sum x_i\right)^2 / M \sum x_i^2$, is equal to one (where x_i is the goodput of the *i*-th station, $GPsta_i$ and *M* is the number of stations).

5.2.1 Varying the number of downstream TCP connections per STA

We consider a number of downstream connections on STA 1 varying from 1 to 6; the other STAs have only one downstream connection; the TCP receiver window of all connections is equal to 42 packets (about 64 kbytes). All STAs have a physical transmission rate of 11 Mbit/sec.

Fig. 6 shows the value of the ratio $\eta_{1,2}$ versus the number of downstream connections of STA 1. In a WLAN without fairness enforcer, there are no queuing phenomena in the STAs buffers, since there are only downstream connections (see considerations at the end of Section 3). All TCP connections experience the same RTT and the same packet loss probability; consequently, each connection has the same goodput. In this scenario, the WLAN without fairness enforcer provides per-flow fairness, thus when STA 1 has X connections active, its goodput is X-time greater than the goodput of STA 2, which has only one connection active; i.e. $\eta_{1,2} = X$.

The average cumulative goodput with VSB (5.0 Mbit/sec) is slightly greater than the one without VSB (4.6 Mbit/sec). The reason for this difference is that with the VSB the TCP send-rate is more stable, since the overall system has more buffer space available and the loss of segments is smaller. As a matter of fact, in the WLAN without fairness enforcer packets are stored in the AP while in presence of the VSB packets are stored in the VSB itself which has a greater buffer space than the AP.

Fig. 7 shows the same performance parameters of Fig. 6 in a multi-rate environment. STA 1 has a physical transmission rate of 2 Mbit/sec; the other four STAs have a rate of 11 Mbit/sec. As expected, the level of fairness does not change; however, we observe that the average cumulative goodput with VSB is about 3.8 Mbit/sec versus a value of 2.6 Mbit/sec in absence of the VSB. This improvement is due to the fact that the VSB limits the bit-rate of the slower STA thus improving the time utilization of the wireless channel.

5.2.2 Varying the size of the receiver window

We vary the TCP receiver window of STA 1 from 6 packets (i.e., about 8 kbytes) to 42 packets (i.e., about 64 kbytes). The TCP receiver window of the other STAs is fixed to 42 packets[9]. All STAs have a physical bit rate of 11 Mbit/sec. The fairness level is reported in Fig. 8. In a WLAN without fairness enforcer there is no queuing phenomena in the stations and each TCP connection experiences the same RTT and segment loss. Consequently, since all stations have the same number of connections, per-station unfairness only shows up when the average TCP *congestion* window of the connections of a given station is different than that of another station. Segment losses occur in the AP buffer and these losses increase as the TCP receiver window of STA 1 increases because more segments are "in-fly". Depending on the amount of segment losses (i.e., on the size of the receiver window of STA 1), we observe two distinct different behaviors of the fairness level:

1- If the TCP receiver window of STA 1 is greater than 36 kbytes, segment loss is so severe that the steady-state TCP congestion window of all connections mainly depends on the loss probability. Consequently, all connections, and all stations in this scenario, have the same goodput and the system provides per-station fairness ($\eta_{1,2}$=1).
2- If the TCP receiver window of STA 1 is lower than 24 kbytes, the receiver window limits the dynamics of the TCP congestion window. Consequently, the smaller is the receiver window of STA 1, the smaller is its goodput with respect to other stations and per-station unfairness appears.

The model results are in good agreement with experimental data. For example if we consider the range of receiver window sizes between 24 kbytes and 36 kbytes, we observe some inaccuracies of the results of the mode. This is due to the approximation used in the formula used to evaluate the average congestion window (i.e., T_0=RTT). The average cumulative goodput with VSB (5.0 Mbit/sec) is greater than that of a WLAN without fairness enforcer (4.8 Mbit/sec), since segment loss is fewer.

5.3 Upstream versus Upstream

In this section we consider stations with upstream connections only and analyze the fairness level versus the number of connections and the TCP receiver window size.

5.3.1 Varying the number of upstream TCP connections per STA

In Fig. 9, we report results for stations with a physical rate of 11 Mbit/sec. In a WLAN without fairness enforcers there are no queuing phenomena in STAs. 2,3,4,5. With respect to STA.1, the queuing

[9] We performed the same test on STA n 2,3,4,5 with a receiver window of 6 packets and we obtained similar results; the model is still accurate and the VSB is effective in enforcing fairness.

phenomena shows up when the number of connections is equal to or greater than three (see considerations at the end of Section 3). As discussed in the upstream-vs-downstream case, this causes an increase of the RTT that results in a leveling of the curve of the fairness level [10].

The average cumulative goodput with VSB (5.0 Mbit/sec) is slightly lower than the case of a WLAN without fairness enforcer (5.4 Mbit/sec). In fact, in both cases the TCP connections fully open their congestion windows. However in a WLAN without fairness enforcer, upstream TCP connections loose only ACKs in the AP buffer. This ACK loss has no negative consequence; on the contrary it slightly improves the WLAN performance since less overhead bits are transferred to the radio interface [6]. This improvement is not obtained with the VSB. We also verified that different physical transmission rates do not result in changes of the fairness level (see Fig. 10).

5.3.2 Varying the size of the receiver window of upstream connections

The results are reported in Fig. 11. Without VSB, the buffers of all STAs are always empty (see considerations at the end of Section 3). Therefore, with all RTTs are equal, no segment loss occurs and the ratio $\eta_{1,2}$ is equal to the ratio of W_1/W_2. The average cumulative goodput with VSB (5.0 Mbit/sec) is lower than the case of a WLAN without fairness enforcer (5.4 Mbit/sec) for the same reasons stated in reference to Fig. 9.

5.4 Real Internet testbed

In this section, we describe a test performed with a real Internet server playing the role of fixed host. We have two STAs that download a big file (the latest version of Ubuntu Linux distribution from the main server: http://releases.ubuntu.com/). The STA n.1 uses a download accelerator that sets up 4 parallels TCP connections with the server, the STA n.2 does not use the accelerator, i.e. it only sets up a TCP connection with the server. Fig. 12 reports the goodput perceived by the two STA versus time, i.e. the download speed of the file. From 0 to 165 seconds the VSB is not present, after 165 seconds the VSB becomes active. The STA without the accelerator starts the download at time 0s, the STA with accelerator starts the download at time 60s. We observe that without the VSB (i.e. before 165s) the STA with the accelerator has a download speed of about 4 Mbit/s and the STA without the accelerator has a

[10] The asymptotic value is $\beta_i B/W_i$,, which can be easily derived by taking the limit of $\eta_{1,2}$ in Eq. (3) for Nup_1 going to infinity.

download speed of about 1 Mbit/s, therefore the measured fairness level $\eta_{1,2}$ is about 4 and this result is consistent with the theoretical result of Eq. (3). At time 165s the VSB is activated and both STAs share in a fair way the WLAN bandwidth.

6 Related Work

In this section we briefly report and compare the latest literature studies on 802.11/TCP fairness modeling as well as solutions proposed to provide per-station fairness. In this section we also provide our reasoning for yet another 802.11 TCP fairness model and the added-value of our VSB fairness enforcer over existing techniques.

6.1 Review of models on TCP fairness in 802.11 Hot-Spots

The fairness performance can be evaluated as a ratio between goodputs of individual connections. Thus, to model fairness performance one can proceed in two ways: i) model the goodput of individual connections and then take a ratio between such goodputs; ii) model directly the fairness performance as we did in Section 3.

As far as the first approach is concerned, a complete and up-to-date modeling of the TCP throughput is given in [19]. There, the authors perform a valuable effort both in revising the flaws of previous literature TCP models and in devising their model.

However, in [19] (and also in the papers quoted in [19] and in [20]), the authors make a strong effort in modeling the 802.11 MAC behavior but use a rather simple model of the TCP behavior. Indeed, the AP is assumed to be lossless; this assumption limits the accuracy of computing fairness since it ignores AP packet loss. The authors of [19] assumed that the AP was lossless, as packet loss never occured in their test-bed campaign. We argue that this may be because they used an operative system (e.g., Microsoft Windows XP or previous versions of Windows) with a small TCP receiver window (e.g., 12 pkts); such values of receiver window prevent the overflow of the AP buffer up to a reasonable number of active TCP connections. On the contrary, recent operating systems (e.g., Microsoft Windows 7 and Linux Kernel 2.6) use larger receiver window (e.g., 42 pkts) and commercial AP buffers start to lose packets even with few connections (e.g., 3).

Other papers [5][4][23] do account for AP packet loss but assume that the occupancy of the uplink buffer of the wireless stations is negligible [11] (*empty-buffer* conjecture) which is also another short-coming compared to our recent findings [28]. As to the approaches based on the direct derivation of fairness performance, a seminal paper is [2]. To the best of our knowledge, [2] is the only paper that proposes a direct analytical model to evaluate TCP fairness performances[12]. In that paper the authors showed for the first time upstream/downstream unfairness phenomena, as well as proposed an analytical model and a technique that restores "per-flow" fairness (i.e., each TCP connection has the same data-rate). We improve the model in [2] by i) analyzing what happens in case of stations with different number of TCP connections and/or different receiver window (although the model in [2] could be extended to allow the same analysis); and ii) removing the *empty-buffer* conjecture.

6.2 Review of literature proposals for providing per-station fairness

A valuable review of techniques to restore TCP fairness in WLAN is given in [24]. In [24] the authors classify the available techniques in two categories: i) per-flow fairness; ii) per-station fairness. In this paper, we focus on per-station fairness. Thus in this section we extend the review performed in [24] with respect to the techniques aimed at providing per-station fairness by adding two recent papers: [21] and [27]. We briefly describe the main concepts of these recent works and briefly recall what has been said in [24] with respect to papers on per-station fairness (see [25][26]). We summarize assumptions and system requirements of those four papers in Tab. 2.

In the Distributed Access Time Control (DATC) [25] each station controls the time spent in accessing the channel; if it is greater than a pre-defined value, then it drops TCP segments. The scheme requires: i) an hacking of the MAC code of the station, to have a run-time feedback on the amount of time it is using the channel and, ii) an additional user-space software on the station to regulate the packet drop rate versus the MAC feedback on current access time. The Access Time Control (ACT) proposed in [26] is similar to DACT but all the work is performed by the AP.

[11] In section 3.1 of [5] the authors do not include in their round trip time evaluation the queuing delay on the STA buffer. In section 4 of [23] (that is an extension of [4]) the authors say that the round trip time of TCP connections (T_d) "…can be approximated by the average queuing delay at the AP downlink buffer".

[12] It should be noted that other two recent papers [21][22] deal with such matters; however, in [21][22] the authors propose an analytical model for UDP traffic over MAC 802.11, and then they use these results to "qualitatively" explain the interaction between TCP and MAC layer, without giving analytical formulas for TCP performance.

In [21] the authors propose an IP-MAC cross-layer solution. To achieve fairness, the AP communicates to stations their allowed percentage usage of the wireless channel. When a station receives the value of its allowed percentage usage, that station sets the IP Explicit Congestion Notification Bit on the receiving packets in such a way that the TCP senders opportunely adapt their rate to reach the allowed percentage usage of the wireless channel. This solution requires i) an hacking of the MAC code of the AP and of the station, in order to handle the new subtype field of the MAC control header, ii) an hacking of the station operative system; in fact, this solution requires the use of TCP explicit congestion notification, which is not "activated" by default on Windows Vista and on Linux Kernel 2.6, while it is completely absent in previous Microsoft OSs.

Finally the technique in [27] requires that the wireless stations control the TCP receiver window to avoid AP buffer overflows. In addition, the receiver window is scaled by the number of connections of the station. The scheme requires additional software (in the AP and in wireless stations) to set at run-time the TCP receiver window of the wireless stations and to communicate to them the AP buffer size.

Summing up, the above schemes require modifying the Access Point or the wireless stations or both. Our solution leaves the AP and user devices without modifications; this was verified in implementation in a test-bed made up of commercial devices. Furthermore, our solution does not exploit 802.11e features to allow backward compatibility toward non-802.11e devices. However, even if we think of a future in which all devices will be 802.11e compliant, our solution in not in contrast with the standards of 802.11e to enforce fairness. For instance, in [30] it is proposed that TCP ACKs and TCP segments going in downlink direction be differently prioritized; this implies the use of different MAC queues (EDCA) for TCP ACK and TCP segments, altering the original 802.11e intended use of EDCA queues (voice, video, best-effort, background) and preventing the effectiveness in case of IPSec. Instead, in our approach the IP best effort traffic would still be handled by the best-effort EDCA queue, leaving background, video and voice MAC queues available to transfer background, video and voice traffic, as the standard assumes. Moreover, we do not need to inspect IP packet payload.

7 CONCLUSION

In this paper we have improved the state-of-the-art on per-station fairness with the following results: i) queuing phenomena on station buffers cannot be neglected; ii) the fairness level does not depend on the physical bit-rate configurations of the stations. Also, we have devised an analytical model that allows analyzing all possible TCP (Reno) connection configurations that may occur in a WLAN Hot-Spot. Finally, we have proposed a viable technique that allows enforcing fairness without the need of modifying deployed Access Points or user devices.

REFERENCES

[1] IEEE Std 802.11-2007, pp.C1-1184, June 12 2007.
[2] S. Pilosof, R. Ramachandran, D. Raz,, Y. Shavitt, S. Prasun, "*Understanding TCP fairness over wireless LAN*," in *Proc.* IEEE INFOCOM 2003, pp. 863-872.
[3] R. Jain, W. Hawe, D. Chiu, "A Quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems," DEC-TR-301, September 26, 1984.
[4] W. Yi, N. Zhisheng, Z. Janfeng, "*Upstream/downstream unfairness issue of TCP over wireless LANs with per-flow queueing*," in *Proc.* IEEE ICC 2005, pp. 3543-3547.
[5] C. Burmeister, U. Killat, J. Bachmann, "*TCP over Rate-Adaptive WLAN - An Analytical Model and its Simulative Verification*," in *Proc.* IEEE WOWMOM 2006, pp. 339-348.
[6] N. Blefari-Melazzi; A. Detti; I. Habib; A. Ordine; S. Salsano, "*TCP Fairness Issues in IEEE 802.11 Networks: Problem Analysis and Solutions Based on Rate Control*," IEEE Transactions on Wireless Communications, vol.6, no.4, pp. 1346-1355. April 2007.
[7] Dennis Pong, Tim Moors, "Fairness and Capacity Trade off in IEEE 802.11 WLANs," LCN 2004.
[8] Godfrey Tan, John Guttag, "Time based Fairness Improves Performance in Multirate WLANs", The USENIX Annual Technical
[9] Conference, Boston, MA, June 2004.
[10] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, I. Tinnirello, "*Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards*," in *Proc.* IEEE INFOCOM 2007, pp. 1181-1189.
[11] J. Padhye, V. Firoiu, D.F. Towsley, J.F. Kurose, "*Modeling TCP Reno performance: a simple model and its empirical validation*," IEEE/ACM Transactions on Networking, Vol.8, No. 2, April 2000, pp. 133 –145.
[12] Joseph Davies, "*The Cable Guy: TCP Receive Window Auto-Tuning,*" Microsoft TechNet Magazine, January 2007, available at "http://www.microsoft.com/technet/technetmag/issues/2007/01/".
[13] R. Braden, "*Requirements for Internet Hosts -- Communication Layers,*" IETF RFC 1122, October 1989.
[14] Forsythe, G. E., M. A. Malcolm, and C. B. Moler, "*Computer Methods for Mathematical Computations,*" Prentice-Hall, 1976
[15] N. Blefari-Melazzi, A. Detti, A. Ordine, S. Salsano, "*A mechanism to enforce TCP Fairness in 802.11 wireless LANs and its performance evaluation in a real test-bed,*" in *Proc.* of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, 2007 (WoWMoM 2007) vol., no., pp.1-7, 18-21 June 2007
[16] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "*Performance anomaly of 802.11b,*" IEEE Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2003, 30 March - 3 April 2003, vol.2, pp. 836-843
[17] A. Detti, N. Blefari, I. Habib, A. Ordine "Per-Station Throughput Fairness in a WLAN Hot-Spot with TCP traffic (Extended Version)" available at http://netgroup.uniroma2.it/Andrea_Detti/VSB/STAFAIR_ext.pdf
[18] A. Detti et al, Linux Implementation of the Virtual Shared Bottelneck available at http://netgroup.uniroma2.it/Andrea_Detti/VSB/vsb.tar.gz
[19] R. Bruno, M.Conti, E. Gregori, "*An accurate closed-form formula for the Throughput of long-lived TCP connections in IEEE 802.11 WLANs,*" Elsevier Computer Networks, Volume 52, Issue 1, 18 January 2008, Pages 199-212
[20] J. Yu, S. Choi "*Modelling and Analysys of TCP dynamics over IEEE 802.11 WLAN*", in *Proc.* of, Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on , vol., no., pp.154-161, 24-26 Jan. 2007
[21] E.C. Park, D.Y. Kim, H. Kim , C. Choi " *A Cross-Layer Approach for The Per-Station Fairness in TCP over WLANs*", IEEE Transaction on Mobile Computing, Vol. 7, No.7, July 2008
[22] E.C. Park, D.Y. Kim, C. Choi, "*Analysis of Unfairness between TCP Uplink and Downlink Flows in Wi-Fi Hot Spots*" IEEE GLOBECOM '06, vol., no., pp.1-5, Nov. 27-Dec. 1 2006
[23] Y. Wu, Z. Niu, J. Zheng, "Study *of TCP upstream/downstream unfairness issue with per-flow queuing over infrastructure-mode WLANs*", Wiley, Wireless Communications and mobile computing, DOI : 10.1002/wcm.303
[24] M. Seyedzadegan, M. Othman, S. Subramaniam, Z. Zukarnain, "*The TCP Fairness in WLAN: A Review*", In proc. of IEEE International Conference on Telecommunications, 14-17 May 2007, Penang, Malaysia.
[25] D.-Y. Kim, E.-C. Park, and C.-H. Choi, "*Distributed Access Time Control for Per-station Fairness in infrastructure WLANs*," IEICE Trans. on Communications , Vol. E89-B, No. 9, pp. 2572-2579, Sep. 2006.
[26] D. Kim, "*Access time control for TCP fairness in WLANs*," Ph.D. Dissertation, Seoul National University, 2006.

[27] M. Seyedzadegan, M. Othman, "*Providing Per-Station TCP Fairness in WLANs using Weighted Window Method*", Proceedings of Communication Systems and Networks (AsiaCSN 2008), April 2 – 4, 2008, Langkawi, Malaysia.

[28] A. Detti, N. Blefari, "*On the buffer occupancy of an IEEE 802.11 station in a Hot-Spot,*" IEEE Transactions on Wireless Communications (letter), in press. Available for reviewers at http://netgroup.uniroma2.it/Andrea_Detti/VSB/STA_buffer.pdf

[29] Little, J. D. C: "*A Proof of the Queueing Formula* $L = \lambda W$", Operations Research, 9, 383-387 (1961).

[30] D.J. Leith; P. Clifford, "*Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links,*" IEEE Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005, pp. 109-118, 3-7 April 2005

[31] HTB for Linux, http://luxik.cdi.cz/~devik/qos/htb

[32] A. V. Babu, L. Jacob, "*Fairness Analysis of IEEE 802.11 Multirate Wireless LANs*", IEEE Transaction on Vehicular Technology, vol. 56, No. 5, Sept. 2007

[33] G. McCullagh, "Exploring delay based TCP congestion control," Master's thesis, Hamilton Institute, NUI Maynooth, Ireland, 2008.

[34] La, R.J., Walrand, J., Anantharam, V.: Issues in TCP Vegas. Electronics Research Laboratory, University of California, Berkeley, UCB/ERL Memorandum, No. M99/3 (January 1999).

[35] Lawrence S. Brakmo and Larry L. Peterson: "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on selected Areas in communications,1995, Vol.13, pages 1465—1480.

TAB. 1
ASSUMPTIONS AND THEIR CONSEQUENCES ON THE ACCURACY OF THE MODEL

	ASSUMPTION	CONSEQUENCE
a1	The latency and the packet loss of the wired part are neglected (i.e. the wireless part is the bottleneck)	Neglecting the packet loss in the wired part seems reasonable, as loss phenomena mainly occur in the wireless part. By contrast, the delay suffered in the wired part cannot be always neglected. However, in [24] (footnote 5) we say how to modify the model to take into account the wired part delay.
a2	The STA uplink buffer is large enough as to avoid packet loss	This is reasonable, since the operating system of a wireless host usually allocates a large amount of memory to its network interfaces (e.g. 1000 packets)
a3	The TCP version is Reno with delayed ACK	This TCP version is the most usual implementation in current operating systems
a4	The packet loss probability at the AP downlink buffer is small enough that: i) packet losses do not prevent the startup of TCP connections; ii) the impact of ACKs loss on the congestion window dynamic is negligible	This is a limitation of our model. We assume that all TCP connections start and reach a steady-state behavior. In [5] we show that this assumption is not always true. In case of heavy losses, some TCP connections may be completely starved. We are not able to capture this critical-starvation phenomenon; thus, our model is valid only when the loss probability is such that critical starvation does not occur. When critical starvation does occur the model results can be considered as best-case in terms of fairness.
a5	When the AP has a steady-state packet loss probability greater than zero, then its buffer is considered as always full	This assumption is needed to avoid evaluating the average value of the occupancy of the AP buffer, which is a complex task. In the lossless case, this assumption is unnecessary. The assumption is reasonable since the AP buffer is the network's bottleneck and TCP tends to fill the bottleneck buffer.
a6	At the steady-state, the AP packet loss process resembles a Bernulli one.	This assumption enables us to adopt the well-known Padhye model [11] in evaluating the throughput of downstream TCP connections. Albeit we are aware that Padhye model is valid for a single TCP flow in a "bath of noise", we have observed that its usage yields theoretical results very tight with experimental ones.
a7	The wireless channel is error free and the MAC Retry Limits is never reached; therefore, the MAC layer behaves as a fair per-packet scheduler among backlogged stations	This assumption is widely used in the literature (e.g., [2][3][4][12][16][17][24][27]) to avoid modeling wireless channel impairments. The assumption is justified by the fact that automatic rate control techniques adopted by the wireless device strongly limits the number of transmission failures due to channel error with respect to the number of successful transmissions. Moreover, the value MAC Retry Limit is usually large enough to successfully cope with all consequent packet errors/collisions. Under assumption a6, it is well-known ([12] [27]) that the MAC layer behaves as a fair per-packet scheduler among backlogged devices, providing equal long-term channel access probability to backlogged devices.

TAB. 2
REVIEW OF PER-STATION FAIRNESS SOLUTIONS

Name & reference	**Hacking of MAC code**	**Hacking of TCP/IP code**	**Location of the enforcer software**	**Performance evaluated by means of**
DACT [25]	YES, on user devices.	NO	User device	Simulations
ACT [26]	YES on the Access Point.	NO	Access Point	Simulations
WW [27]	NO	NO	User devices and Access Point	Simulations
CL-F [21]	YES, on Access Point and user devices	YES, on user devices	User devices and Access Point	Simulations
Our VSB solution	NO	NO	Ad hoc device located between gateway and AP	Experimental Test-bed

FIGURES

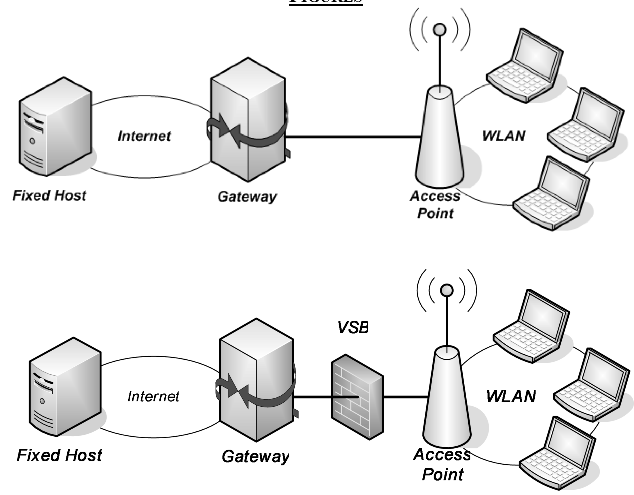

Fig. 1 – WLAN Hot-spot scenario (upper part: without fairness enforcer, lower part with the VSB fairness enforcer)

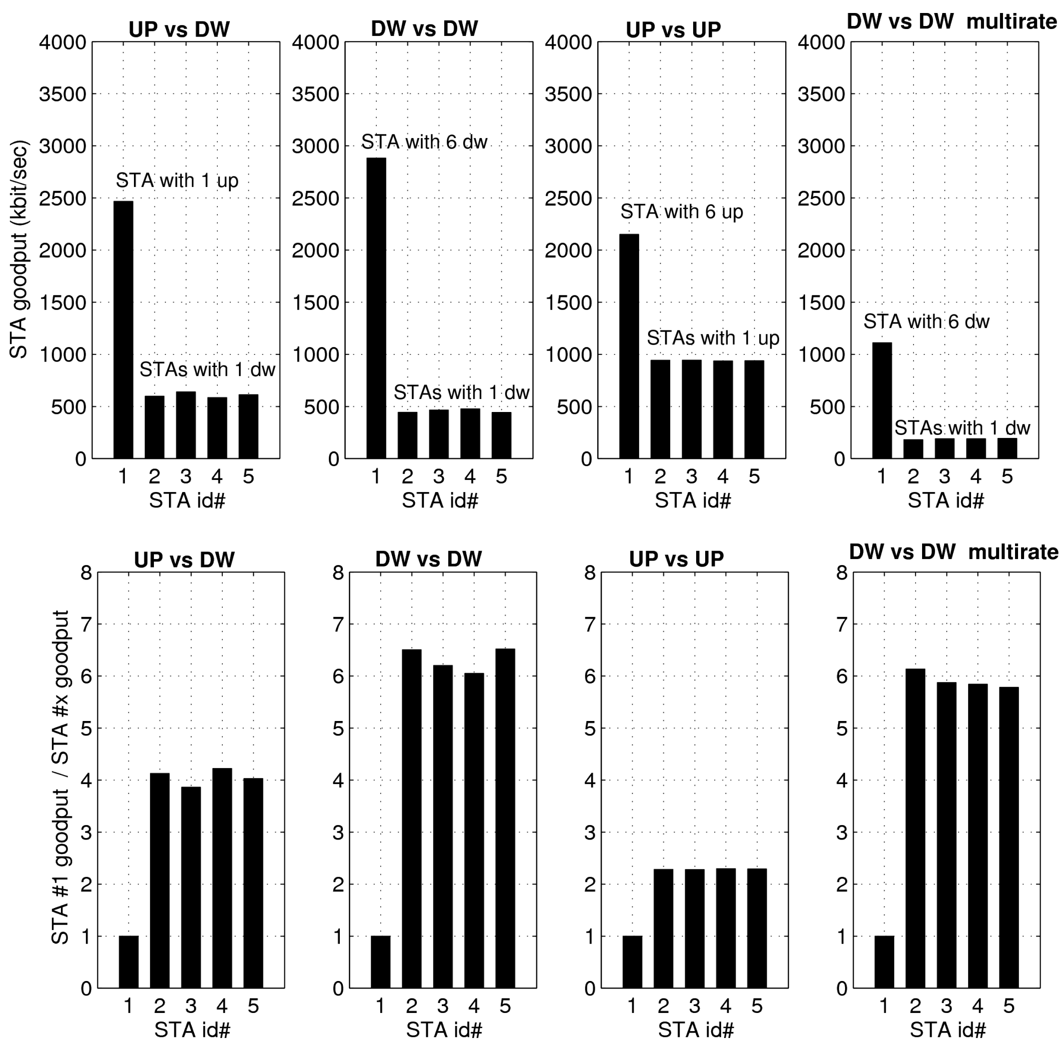

Fig. 2 – STA goodput (upper plots) and related fairness level (lower plots).

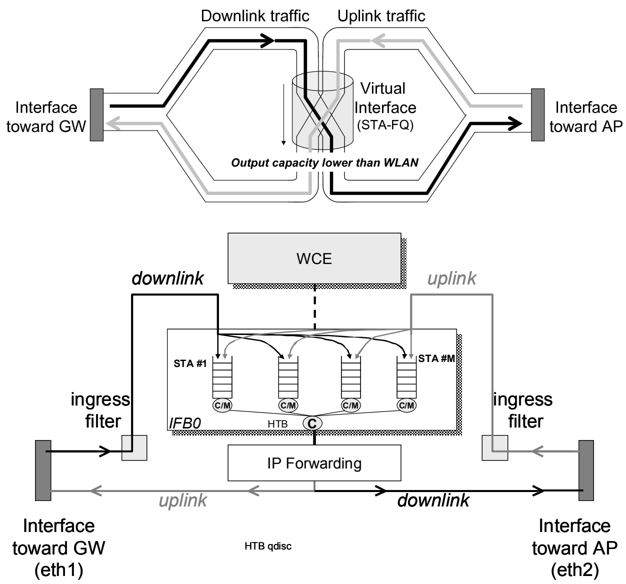

Fig. 3 – Traffic handling in the Virtual Shared Bottleneck (upper part) and our Linux implementation (lower part)

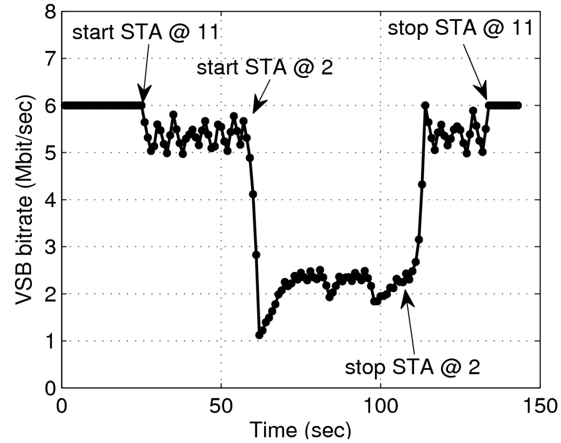

Fig. 4 – VSB bit rate as estimated by the WCE versus time in case of two stations with different physical transmission rates.

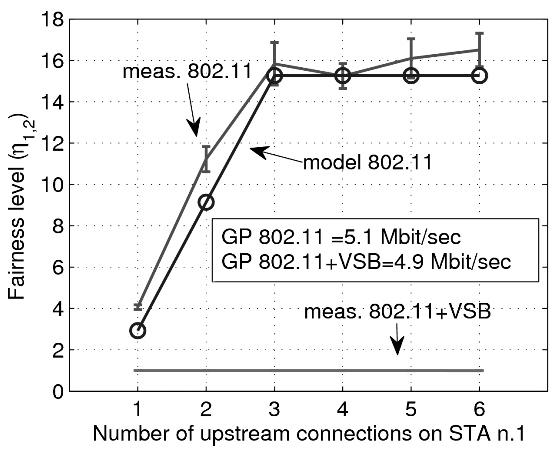

Fig. 5 – Fairness level $\eta_{1,2}$ versus the number of upstream connections on STA 1; the other STAs have 1 downstream connection; all the STAs have a physical transmission rate of 11 Mbit/sec and a TCP receiver window of 42 pkts

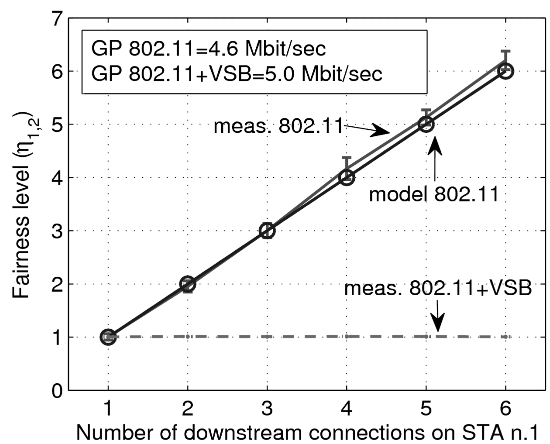

Fig. 6 – Fairness level $\eta_{1,2}$ versus the number of downstream connections on STA 1; the other STAs have 1 downstream connection; all the STAs have a physical transmission rate of 11 Mbit/sec and a TCP receiver window of 42 pkts (64 kbytes)

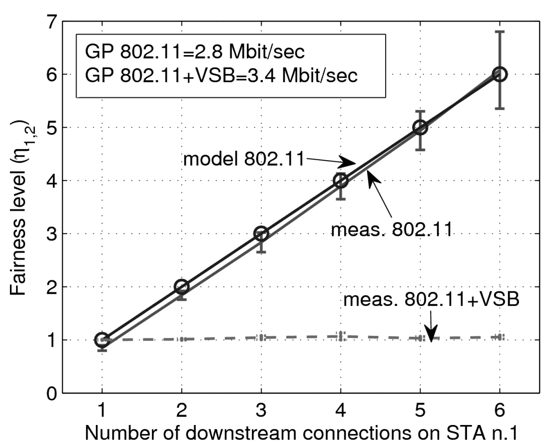

Fig. 7 – Fairness level $\eta_{1,2}$ versus the number of downstream connections on STA 1; the other STAs have 1 downstream connection. STA 1 has a physical transmission rate of 2 Mbit/sec, other STAs work at 11 Mbit/sec. The TCP receiver window is 42 pkts (64 kbytes).

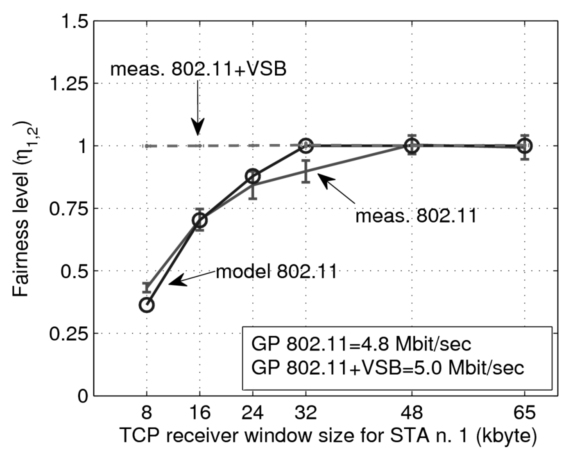

Fig. 8 – Fairness level $\eta_{1,2}$ versus the size of the TCP receiver window of STA 1; the other STAs have a TCP receiver window equal to 64 kbytes (i.e. 42 pkts). All the STAs have only one downstream connection and the physical transmission rate is 11 Mbit/sec.

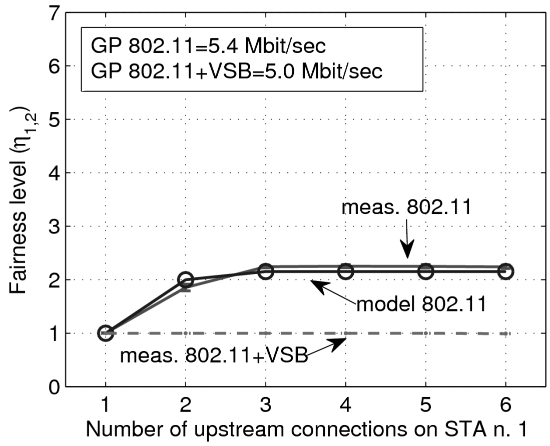

Fig. 9 – Fairness level $\eta_{1,2}$ versus the number of upstream connections on STA 1; the other STAs have 1 upstream connection; all the STAs have a physical transmission rate of 11 Mbit/sec and a TCP receiver window of 42 pkts (64 kbytes).

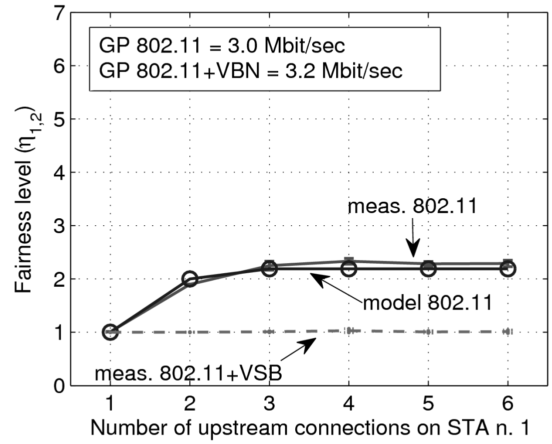

Fig. 10 – Fairness level $\eta_{1,2}$ versus the number of upstream connections on STA 1; the other STAs have 1 upstream connection. STA 1 has a physical transmission rate of 2 Mbit/sec; the other STAs work at 11 Mbit/sec. The maximum TCP receiver window is 42 pkts (64 kbytes).

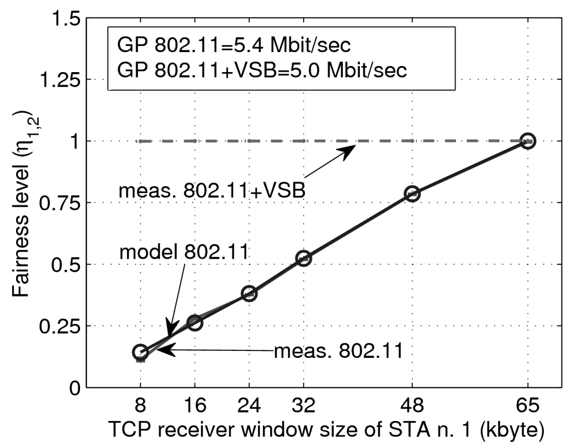

Fig. 11 Fairness level $\eta_{1,2}$ versus the size of the TCP receiver window of STA 1; the other STAs have a TCP receiver window equal to 64 kbytes (i.e. 42 pkts). All the STAs have only one upstream connection and the physical transmission rate is 11 Mbit/sec.

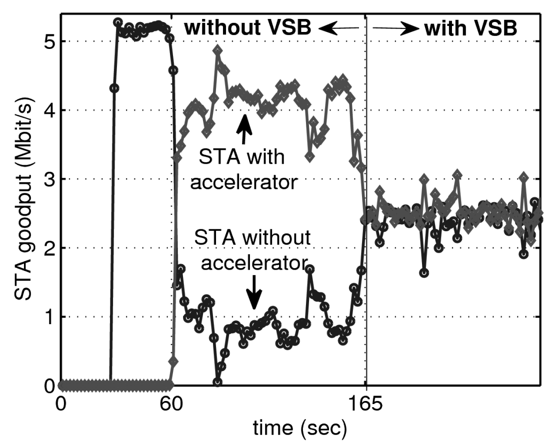

Fig. 12 Goodput perceived by two STAs with physical transmission rate 11 Mbit/sec during an actual Internet download in absence (before 165 sec) and presence (after 165 sec) of the VSB. A STA performs the download using a download accelerator, the other STA does not use the download accelerator.

TAB. 3
DEFINITIONS

d1	$\eta_{i,j}$: the fairness level between station #*i* and station #j, defined as the ratio between the goodput of station #*i* and the goodput of station #*j*.
d2	$GPsta_i$: *i*-th STA goodput defined as the “overall” useful data-rate of all its TCP connections
d3	*B*: the size of the AP buffer (in packets)
d4	*M*: the number of STAs
d5	Nup_i: the number of upstream connections (i.e., with data source on the STA) of the *i*-th STA
d6	Ndw_i: the number of downstream connections (i.e., with data source on the fixed host) of the *i*-th STA
d7	W_i: the maximum TCP congestion window of the *i*-th STA (in packets)
d8	*Qap:* the average occupancy of the AP buffer
d9	Qup_i: the average number of packets stored in the *i*-th STA buffer and belonging to an upstream connection
d10	Qdw_i: the average number of packets stored in the *i*-th STA buffer and belonging to a downstream connection
d11	Q_i: the average number of packets stored in the *i*-th STA buffer (i.e., $Qup_i + Qdw_i$)
d12	*NSR* (*p*,*W*): the number of segments emitted by a TCP connection during a RTT (i.e., $E[D_{AP}]+E[D_i]$) in presence of a segment loss probability *p*, with a receiver window equal to *W* and with delayed ACKs. This value is can be derived from Eq. (32) of [11], with T_0=RTT and *b*=2.
d13	*Tn*(*p*,*W*): the average value of the useful segments received in an RTT in presence of a segment loss probability *p* and in case of a maximum congestion window equal to *W*. This value can be derived from Eq. (37) of [11] multiplied by RTT and assuming T_0=RTT and *b*=2.
d14	β_i*:* the ratio between the medium access probability of the *i*-th STA and the medium access probability of the AP. If the MAC level is fair at packet level, then $\beta=1$; however, it may happen that a specific implementation of the 802.11 layer alters the per-packet fairness, giving to the STAs an advantage in accessing the medium with respect to the AP ($\beta>1$) or viceversa ($\beta<1$).
d15	$E[D_{ap}]$: the average queuing delay of the AP buffer; i.e., the average time interval between the instant in which a packet is stored in the AP buffer and the instant in which the MAC ACK of that packet is received by the AP. Thus, this parameter includes also all the relevant MAC layer delays.
d16	$E[D_i]$: the average queuing delay suffered in the i-th STA buffer, ; i.e., the average time interval between the instant in which a packet is stored in the STA buffer and the instant in which the MAC ACK of that packet is received by that STA. Thus, this parameter includes also all the relevant MAC layer delays.
d17	*p*: the AP packet loss probability

APPENDIX I

ASSUMPTIONS AND MODEL LIMITS

In this appendix we discuss the assumptions made in the paper and the ensuing model limits (se also Tab. 1).

Assumption a1 means that the wireless part is the bottleneck. Neglecting the packet loss in the wired part seems reasonable, as loss phenomena mainly occur in the wireless part. By contrast, the delay suffered in the wired part cannot be always neglected. However, in [28] (footnote 5) we say how to modify the model to take into account the wired part delay.

Assumption a2 is reasonable, since the operating system of a wireless host usually allocates a large amount of memory to its network interfaces (e.g. 1000 packets).

With respect to assumption a3, the Reno version with delayed ACK is the most usual implementation in current operating systems.

Assumption a4 is a model limitation. We assume that all TCP connections start and reach a steady-state behavior. In [6] we show that this assumption is not always true. In case of heavy losses, some TCP connections may be completely starved. We are not able to capture this critical-starvation phenomenon; thus, our model is valid only when the loss probability is such that critical starvation does not occur. When critical starvation does occur the model results can be considered as best-case in terms of fairness.

Assumption a5 allows us not to compute the average value of the occupancy of the AP buffer, by assuming full buffer occupancy whenever losses occur. In the lossless case, this assumption is unnecessary. The assumption is true as TCP flows tend to saturate the AP buffer being the network's bottleneck. Increasing the number of connections further validates a5 [(13)].

Assumption a6 This assumption enables us to adopt the well-known Padhye model [11] in evaluating

[13] We note that if the AP becomes much more aggressive than STAs (e.g., $\beta<0.5$ see definition d13), then a5 is not completely verified, as the average value of the occupancy of the AP buffer is a bit lower than the buffer size. However, we experienced this kind of problem only for very serious cases of MAC level unfairness. We succeed in reproducing these pathological situations only by means of simulations, as we never observed such limit behaviours by using real equipments (e.g., Cisco and Linksys as APs and DLink and INTEL Centrino as WLAN cards).

the throughput of downstream TCP connections. Albeit we are aware that Padhye model is valid for a single TCP flow in a "bath of noise", we have observed that its usage yields theoretical results very tight with experimental ones.

Assumption a7 is widely used in the literature (e.g., [2][4][5][16][20][21][28][32]) since it avoids the mathematical burden of modeling wireless channel impairments. The assumption is justified by the fact that automatic rate control techniques adopted by the wireless device strongly limits the number of transmission failures due to channel error with respect to the number of successful transmissions. Moreover, the value MAC Retry Limit is usually large enough to successfully cope with all consequent packet errors/collisions. Under assumption a6 it is well-known [16] [32] that the MAC layer operations behave as a fair per-packet scheduler among backlogged devices. Indeed, the MAC layer provides equal long-term channel access probability to backlogged devices.

APPENDIX II

DERIVATION OF p, Q_{AP} AND Q_I

In this appendix we derive the AP packet loss probability (p), the average buffer occupancy of the AP (Q_{ap}) and of a generic wireless station #i (Q_i). The analytical methodology that we follow is the one described in [28]. However, we extend our previous work by: i) considering the TCP delayed-ack mechanism; ii) including the factor β_i,and iii) considering the fact that different stations may have different receiver windows W_i. To make this appendix self-consistent we prefer to report the full analysis instead of describing only the improvements with respect to [28]. We point out that items (ii) and (iii) are simple modifications of the original formulas of [28]. On the other hand, the analysis of the delayed-ack mechanism has required significant modeling effort especially in the AP *lossless* case.

Before we proceed, we need additional definitions that we report in Tab. 4. The analysis is based on the notion of "round": *A round is the time interval needed to send out all the packets buffered at the AP wireless interface since the start of the round itself*. The i-th round starts at time t_i and the first round starts at time t_0. The time t_0 is any time instant after which the system can be considered in a steady-state.

For instance, if at time t_i the last packet in the AP buffer is the packet #x, then the i-th round ends when the packet #x is received by the destination STA[14].

To derive p, Q_{ap} and Q_i, we analyze both the *lossless* and *lossy* AP buffer cases. To decide which of the two models can be used in a given scenario we evaluate Q_{ap} by means of the *lossless* formulas. If Q_{ap} is less than the AP buffer size, B, then we use the same lossess formulas, otherwise we use the *loss*y ones.

7.1 Lossless AP buffer

Let us consider a generic STA buffer during a generic round k (i.e., the round that starts at time t_k). For each downstream TCP connection of the STA: every time two segments are received by a TCP sink of the STA, the TCP sink sends out only one ACK [15]. Thus the STA queues a downstream packet (i.e. an ACK of a downstream connection) every two receptions of downstream packets (i.e., two TCP segments of a downstream connection). For each upstream connection of the STA, every time an ACK is received by a TCP source of the STA, the TCP source sends out two segments. Thus the STA queues two upstream packets (i.e., two TCP segments of an upstream connection) for every reception of an upstream packet (i.e. an ACK of an upstream connection).

If we assume that packet emissions and transmissions during a round can be modeled with a fluid flow approach, we can write the number of packets of downstream ($Qdw_i(t_{k+1})$) and upstream ($Qup_i(t_{k+1})$) connections stored in the buffer of the i-th STA, at the end of round k (i.e., at the start of round k+1) as:

$$\begin{aligned} Qdw_i(t_{k+1}) &= \max\{0, Qdw_i(t_k) + 0.5 \cdot Pown\_dw_i(t_k) \cdot Qap(t_k) - \chi dw_i(t_k)\} \\ Qup_i(t_{k+1}) &= \max\{0, Qup_i(t_k) + 2 \cdot Pown\_up_i(t_k) \cdot Qap(t_k) - \chi up_i(t_k)\} \end{aligned} \tag{6}$$

where $Pown\_dw_i(t_k)$ ($Pown\_up_i(t_k)$) is the probability that at the beginning of round k a packet stored in the AP belongs to a downstream (upstream) connection of the i-th STA; $Qap(t_k)$ is the number of packets stored in the AP buffer at time t_k; $\chi dw_i(t_k)$ ($\chi up_i(t_k)$) is the number of downstream (upstream) packets leaving the i-th STA buffer during the round. The *max* operator accounts for the obvious fact that the buffer occupancy cannot be less than zero.

[14] As an exception to this definition, if at the end of the i-th round, no packet is contained in the AP buffer, then the i+1-th round is defined as a *void-round*. During a *void-round* all the backlogged STAs transmit a single packet toward the AP and the i+1-th round ends.

[15] We are assuming that the delayed ACK timer never expires

If we assume that the involved random processes are stationary, by taking the average of both members of Eq. (6) and by approximating E[max(X,Y)] with max(E[X],E[Y])[16] we obtain an approximation of the average number of downstream (Qdw_i) and upstream (Qup_i) packets stored in the i-the STA buffer:

$$Qdw_i = \max\{0, Qdw_i + 0.5 \cdot Pown\_dw_i \cdot Qap - \chi dw_i\} \tag{7}$$

$$Qup_i = \max\{0, Qup_i + 2 \cdot Pown\_up_i \cdot Qap - \chi up_i\} \tag{8}$$

where $Pown\_up_i$ ($Pown\_dw_i$) is the steady-state probability that a packet transmitted by the AP on the wireless interface belongs to an upstream (downstream) connection of the i-th STA; χdw_i (χup_i) is the average number of downstream (upstream) packets leaving the i-th STA buffer during a round.

We now calculate the average occupancy of the STA buffer Q_i=Qdw_i+Qup_i. Unfortunately, the *max* operator in Eq. (7) and Eq. (8) makes this evaluation difficult. For this reason, we first evaluate Q_i when it is greater than zero and then we take into consideration what happens when the STA buffer is empty.

When Q_i>0 Eqs. (7) and (8) may be particularized in three different ways, depending on the traffic scenarios: i) when there are only downstream connections, Eq. (8) becomes $Qup_i = 0$ while the *max* operator can be neglected in Eq. (7), since Q_i=Qdw_i>0; ii) when there are only upstream connections, Eq. (7) becomes $Qdw_i = 0$ while the *max* operator can be neglected in Eq. (8); iii) when there are both upstream and downstream connections, the *max* operators of Eqs. (7) and (8) may be neglected, since both upstream and downstream connections have packets in the *shared* STA buffer. In the following we derive the value of Q_i in the latter case. Nevertheless, it is easy to verify that the formula for Q_i that we obtain in this case (the following Eq. (12)) will be valid also for the two cases of unidirectional data traffic.

Since Q_i>0 the station is backlogged and the number of packets that can leave the STA buffer during a *round* is equal to the number of packets emitted by the AP multiplied by β_i, since for each packet transmitted by the AP a backlogged STA is able to transmit, on average, β_i packets [17]. This said, we evaluate the parameter χup_i as the average number of packets leaving the STA buffer during a *round* (i.e.,

[16] We note that this approximation results in an underestimation of Qi, which decreases as Qi increases.

[17] We recall that in a perfect 802.11 environment β_i=1, as the station has the same transmission opportunity of the AP. Nevertheless firmware or software difference among commercial products may lead to value of β_i different than the perfect 1.

β_i Qap) multiplied by the probability that such packets belong to upstream connections of the i–th STA. The latter probability is equal to the ratio between the average value of upstream packets in the STA buffer and the average value of all packets in the STA buffer. A similar derivation can be repeated for the parameter χdw_i, thus obtaining [18]:

$$\begin{aligned} \chi up_i &= \frac{Qup_i}{Qdw_i + Qup_i} \cdot \beta_i \cdot Qap \\ \chi dw_i &= \frac{Qdw_i}{Qdw_i + Qup_i} \cdot \beta_i \cdot Qap \end{aligned} \tag{9}$$

Now we deal with the evaluation of $Pown\_dw_i$ and $Pown\_up_i$. The probability $Pown\_dw_i$ is equal to the ratio between the number of packets of downstream connections of the i-th STA contained in the AP buffer and the total number of packets contained in the AP buffer (Qap). Since the system is lossless, the connections fully open their congestion window up to the receiver window W_i. Within the STA buffer there are Qdw_i TCP ACKs; consequently the number of TCP segments in the AP buffer belonging to downstream connections of the STA i-th is $Ndw_i\ W_i - 2\ Qdw_i$, where the factor 2 accounts for the delayed ACK [19]. A similar derivation can be repeated for the upstream direction, concluding that the number of ACKs contained in the AP buffer and belonging to upstream connections of the STA i-th is $(Nup_i\ W_i - Qup_i)/2$. Thus, we can write:

$$\begin{aligned} Pown\_dw_i &= \frac{(Ndw_i \cdot W_i - 2 \cdot Qdw_i)}{Qap} \\ Pown\_up_i &= \frac{(Nup_i \cdot W_i - Qup_i)/2}{Qap} \end{aligned} \tag{10}$$

By considering Eqs. (7), (8), (9) and (10) as an equation system in the six unknowns Qup_i, Qdw_i, $Pown\_dw_i$, $Pown\_up_i$, χdw_i, χup_i, and by solving such system we have:

[18] In Eq.(9) we do not take into account the possibility of having a *void-round*. To take into account a *void-round* the value of β_i Qap in this equation should be replaced by $\max\{\beta_i\ Qap, 1\}$, where the *max* operator accounts for the fact that during a void-round (i.e. Qap=0) at least one segment is sent out. This assumption may lead to a physically meaningless solution of the system Ω (described in the following) for which Qap=0. In this case to obtain a meaningful solution we can approximate $\max\{\beta_i\ Qap, 1\}$ with $\beta_i\ Qap+1$.

[19] If the RTT of the fixed network is not negligible, then ACKs are contained also in the fixed network's pipe; in that case, we must subtract a further quantity equal to two times the number of ACKs contained in the fixed network's pipe. The same operation must be done for upstream connections. From this point onward, the derivation can follow the same steps also in the lossy case (see also [28]).

$$Qup_i = Nup_i \cdot W_i - \frac{Nup_i}{Nup_i + 0.5 \cdot Ndw_i} \beta_i \cdot Qap$$
$$Qdw_i = \frac{Ndw_i \cdot W_i}{2} - \frac{Ndw_i}{2 \cdot Nup_i + Ndw_i} \beta_i \cdot Qap \tag{11}$$

The last equations give the average occupancy of the STA buffer when the STA buffer is greater than zero. To account for the general case we must write:

$$Qup_i = \max\left(0, Nup_i \cdot W_i - \frac{Nup_i}{Nup_i + 0.5 \cdot Ndw_i} \beta_i \cdot Qap\right)$$
$$Qdw_i = \max\left(0, \frac{Ndw_i \cdot W_i}{2} - \frac{Ndw_i}{2 \cdot Nup_i + Ndw_i} \beta_i \cdot Qap\right) \tag{12}$$

Now we are left with the evaluation of Qap, which can be expressed as:

$$Q_{AP} = \sum_{i=1}^{M} \left(\frac{Nup_i \cdot W_i - Qup_i}{2}\right) + \left(Ndw_i \cdot W_i - 2 \cdot Qdw_i\right) \tag{13}$$

We now have $2M+1$ equations: Eq. (13) plus the $2M$ Eqs. (12). The unknowns are $2M+1$: Qap and Qup_i, Qdw_i for all the possible value of i, $1 \le i \le M$. The resulting system of equations will be referred to in the following as Ω. The *max* operator in Eqs. (12) makes the system Ω not-linear and difficult to solve analytically. Thus we resort to a classical numerical technique (see e.g., [14]), which searches the solution by varying Qap in the range $0 \le Qap \le \sum_{i=1}^{M} \left((Nup_i \cdot W_i)/2 + Ndw_i \cdot W_i\right)$, where the upper limit of the range is the number of packets contained in the AP buffer when all STA buffers are empty.

In addition, when the MAC layer implementation does not introduce unfairness in the medium access (i.e., $\beta_i=\beta=1$), we are able to solve Ω analytically, as follows. From Eqs. (7) and (8) we note that if the i-th STA is backlogged then $Pown_i=Pown\_dw_i+ Pown\_up_i \ge 0.5$. This implies that only one STA can be backlogged. Then we can strongly simplify the system Ω by assuming that the backlogged STA is the i-th one and exploiting the fact that $Q_j=0$ for all other STAs. The result is that the i-th STA is backlogged if and only if:

$$Nup_i \cdot W_i \ge \gamma + Ndw_i \cdot W_i \tag{14}$$

where,

$$\gamma = \left(\sum_{\substack{j=1 \\ j \neq i}}^{M} Nup_j \cdot W_j + 2 \cdot Ndw_j \cdot W_j \right) \quad (15)$$

When the i-th STA verifies the backlogging condition now expressed as Eq. (14), then the average occupancy Q_i ($=Qdw_i+Qup_i$) of its buffer can be written as:

$$Q_i = \max\left\{0, \frac{(Nup_i + 0.5 \cdot Ndw_i) \cdot (Nup_i \cdot W_i - Ndw_i \cdot W - \gamma)}{Nup_i - Ndw_i}\right\} \quad (16)$$

7.2 Lossy AP buffer

The following derivation is a simple extension of the analysis in [28]. The *lossy* case differs from the *lossless* one in the following points.

- With respect to downstream connections, the loss of a segment in the AP buffer reduces the TCP congestion window. Hence, the congestion window of a downstream connection is not constant, as in the *lossless* case, but it depends on the packet loss probability of the AP buffer.
- With respect to upstream connections, the loss of ACKs in the AP buffer implies that, when a TCP source located in a STA receives an ACK after a sequence of ACK losses, that TCP source sends out a burst of TCP segments (given the fact that ACKs are cumulative). The size of this burst is equal to the number of segments cumulatively acknowledged by the received ACK. It follows that the STA may queue in its buffer more than one packet for each received ACK.
- *Qap* is no more an unknown: it is equal to *B*, given assumption a5.

To evaluate p and Q_i in the lossy case, we first consider the impact on the STA buffer of downstream connections and then that of upstream connections. As for downstream connections, every time that the TCP sink of a STA receives two segments, the TCP sink sends out the respective ACK[20]. As a consequence Eq. (7) holds also in the *lossy* case, for downstream packets.

As for upstream connections, given assumption a4, the congestion window is always equal to the receiver window, W_i. It follows that the overall number of in-fly segments is equal to $Nup_i\ W_i$. Given assumption a1, these packets can be in the AP buffer, or in the STA buffer, or they are lost at the AP

[20] This is strictly true in absence of losses; when a segment loss occurs during the fast recovery phase, a duplicated ACK is sent for each segment. We do not consider this behavior in our model and we will assess the impact of this approximation via test-bed measurements.

buffer. Packets in the STA buffer are TCP segments, whereas packets lost or packets in the AP buffer are TCP ACKs. This implies that, at the start of round k, the number of in-fly ACKs of the i-th STA is equal to $(Nup_i\ W_i - Qup_i(t_k))/2$. Given the fact that ACKs are cumulative, at the end of round k, $Nup_i\ W_i - Qup_i(t_k)$ segments will be acknowledged and TCP senders of the i-th STA will queue in the STA buffer the same number of segments.

This said, in the *lossy* case the Eqs. (7) and (8) can be rewritten as

$$Qdw_i = \max\{0, Qdw_i + 0.5 \cdot Pown\_dw_i \cdot B - \chi dw_i\} \tag{17}$$

$$Qup_i = \max\{0, Qup_i + (Nup_i \cdot W_i - Qup_i) - \chi up_i\} = \max\{0, Nup_i \cdot W_i - \chi up_i\} \tag{18}$$

As done in the *lossless* case, to evaluate the average occupancy of the STA buffer we first evaluate this quantity when it is greater than zero and then we take into account what happens when the STA buffer is empty. Moreover, as done in the *lossless* case, we assume that a station has both upstream and downstream connections; nevertheless the obtained formulas (see Eq. (22)) hold in unidirectional cases too.

If the occupancy of the STA buffer is larger than zero, and there are both upstream and downstream connections; then the *max* operator in Eqs. (17) and (18) can be neglected. Moreover, we observe that we cannot use the value of $Pown\_dw_i$ evaluated in the lossless case (i.e., Eq. (10)) since packet losses may now occur. Thus, we evaluate $Pown\_dw_i$ as the ratio between the number of downstream packets (i.e., segments) of the i-th STA entering the AP buffer in the unit time and the overall number of packets entering the AP buffer in the unit time. If we take as time unit the average packet delay in the AP buffer $E[D_{AP}]$ (i.e., the average duration of a round), we have:

$$Pown\_dw_i = \frac{\dfrac{Ndw_i \cdot NSR(p,W_i)}{E[D_{AP}] + E[D_i]} \cdot E[D_{AP}]}{B \cdot \left(\dfrac{1}{1-p}\right)} = \frac{Ndw_i \cdot NSR(p,W_i) \cdot (1\text{-}p)}{B + \left(\dfrac{Q_i}{\beta_i}\right)} \tag{19}$$

where $NSR(p,W)$ is the number of segments emitted by a TCP connection during a RTT (i.e., $E[D_{AP}]+E[D_i]$) in presence of a segment loss probability p, with a receiver window W and in presence of delayed ACKs. $NSR\ (p,W)$ is evaluated by the classical expression obtained in [11], specifically Eq. (32)

of [11], where we assume T_0=RTT and b=2. Therefore, the numerator Ndw_i $NSR(p,W)$ E[D_{AP}] / E[D_{AP}]+E[D_i] is exactly equal to the number of TCP segments of the i-th STA entering the AP buffer in $E[D_{AP}]$ seconds. Moreover, during E[D_{AP}] seconds, the AP sends out B packets (see assumption a5), thus $B/(1-p)$ is the overall average number of packets entering the AP in $E[D_{AP}]$ seconds. Finally, the last equality is obtained by using the relation E[D_i]/E[D_{ap}] =$Q_i/(\beta_i B)$.

By substituting the expressions for χup_i and χdw_i given in Eq. (9) in Eqs. (18) and (17) (without the *max* operator, since we are assuming Q_i>0) we obtain:

$$Qup_i = Nup_i \cdot W_i - \frac{Qup_i}{Qdw_i + Qup_i} \cdot B \cdot \beta_i \tag{20}$$

$$0.5 \cdot Pown\_dw_i \cdot B = \chi dw_i = \frac{Qdw_i}{Qdw_i + Qup_i} \cdot B \cdot \beta_i \tag{21}$$

By combining Eqs. (20) and (21) we obtain:

$$Q_i = Qdw_i + Qup_i = \beta_i \cdot \frac{Nup_i \cdot W_i}{(\beta_i - Pown\_dw_i/2)} - B \cdot \beta_i \tag{22}$$

If we put in the last equation the expression for $Pown\_dw_i$ given in Eq. (19), we obtain a quadratic equation in the unknown Q_i. Solving this equation, we get a single meaningful solution, given by: $Nup_i\ W_i + 0.5\ NSR(p,W_i)(1\text{-}p)\text{-}\beta_i B$. The last expression gives the average occupancy of the STA buffer when it is greater than zero. Thus, to account for the general case we write:

$$Q_i = \max\left\{0, Nup_i \cdot W_i + \frac{1}{2} \cdot Ndw_i \cdot NSR(p,W_i) \cdot (1-p) - \beta_i \cdot B\right\} \tag{23}$$

We are now left with the evaluation of p. By definition, p is equal to one minus the ratio between the traffic leaving the AP buffer and the traffic offered to the AP buffer:

$$p = 1 - \frac{\dfrac{B}{E[D_{AP}]}}{\sum_{j=1}^{M} \dfrac{0.5 \cdot (Nup_j \cdot W_j) + Ndw_j \cdot NSR(p,W_j)}{E[D_j] + E[D_{AP}]}} = 1 - \frac{B}{\sum_{j=1}^{M} \dfrac{0.5 \cdot (Nup_j \cdot W_j) + Ndw_j \cdot NSR(p,W_j)}{1 + \dfrac{Q_j}{\beta_j \cdot B}}} \tag{24}$$

If we combine Eq. (24) with the M Eqs. (23), we obtain a non-linear system of M+1 equations with M+1 unknowns: Q_i with 1≤i≤M and p. This system of equations will be referred to in the following as Ψ.

We solve the system Ψ via a classical numerical technique (see e.g., [14]), which searches the solution by varying p in the range $0 < p < 1$.

Finally, we evaluate a useful approximation of the average buffer occupancy of a STA by neglecting the contribution of downstream connections in Eqs. (23). The rationale of this approximation derives from the fact that when queuing phenomena occur on a STA, the AP buffer is heavy loaded and the downstream connections operate with a very small congestion window. Thus, the average buffer occupancy of a STA can be approximated by:

$$Q_i = \max\{0, Nup_i \cdot W_i - \beta_i \cdot B\} \tag{25}$$

TAB. 4
ADDITIONAL DEFINITIONS

d18	$Qdw_i(t_k)$: the number of packets stored in the i-th STA buffer at time t_k (i.e. at the start of round k) and belonging to a downstream connection
d19	$Qup_i(t_k)$: the number of packets stored in the i-th STA buffer at time t_k and belonging to an upstream connection
d20	$Q_i(t_k)$: the number of packets stored in the i-th STA buffer at time t_k, i.e. $Qdw_i(t_k) + Qup_i(t_k)$;
d21	$Qap(t_k)$: the number of packets stored in the AP buffer at time t_k; this quantity is equal to the number of packets emitted by the AP during the k-th round
d22	$Pown\_dw_i(t_k)$: the probability that a packet transmitted on the wireless interface by the AP during the k-th *round* belongs to a downstream connection of the i-th STA.
d23	$Pown\_up_i(t_k)$: the probability that a packet transmitted on the wireless interface by the AP during the k-th *round* belongs to an upstream connection of the i-th STA.
d24	$Pown_i(t_k)$: the probability that a packet transmitted on the wireless interface by the AP during the k-th *round* is directed toward the i-th STA; i.e. $Pown_i(t_k)= Pown\_up_i(t_k) + Pown\_dw_i(t_k)$
d25	$Pown\_up_i$: the steady-state probability that a packet transmitted by the AP on the wireless interface belongs to an upstream connection of the i-th STA. $Pown\_up_i$ is also equal to the probability that a packet stored in the AP buffer belongs to an upstream connection of the i-th STA
d26	$Pown\_dw_i$: the steady-state probability that a packet transmitted by the AP on the wireless interface belongs to a downstream connection of the i-th STA. $Pown\_dw_i$ is also equal to the probability that a packet stored in the AP buffer belongs to a downstream connection of the i-th STA

d27	$Pown_i$: the steady-state probability that a packet transmitted by the AP on the wireless interface belongs to a connection of the i-th STA, i.e. $Pown_i = Pown\_up_i + Pown\_dw_i$
d28	$\chi dw_i(t_k)$ ($\chi up_i(t_k)$): the number of downstream packets leaving the i-th STA buffer during the k-th round
d29	$\chi up_i(t_k)$: the number of upstream packets leaving the i-th STA buffer during the k-th round

APPENDIX III

UP/DOWN FAIRNESS INDEX

Pilosof et al. in [2] define the "up/down" fairness parameter $R = R_{up}/R_{dw}$, where R_{up} (R_{dw}) is the goodput of all upstream (downstream) connections. In this appendix we derive R by using the analytic approach presented in Section 3:

$$
\begin{aligned}
R &= \frac{R_{up}}{R_{dw}} = \frac{NormR_{up}}{NormR_{dw}} = \frac{\sum_{i=1}^{M} (Nup_i \cdot W_i) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)}{\sum_{i=1}^{M} (Ndw_i \cdot Tn(p, W_i)) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)} \\
NormR_{up} &= R_{up} / (R_{up} + R_{dw}) = \frac{\sum_{i=1}^{M} (Nup_i \cdot W_i) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)}{\sum_{i=1}^{M} (Nup_i \cdot W_i + Ndw_i \cdot Tn(p, W_i)) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)} \\
NormR_{dw} &= R_{dw} / (R_{up} + R_{dw}) = \frac{\sum_{i=1}^{M} (Ndw_i \cdot Tn(p, W_i)) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)}{\sum_{i=1}^{M} (Nup_i \cdot W_i + Ndw_i \cdot Tn(p, W_i)) / \left(1 + \left(Q_i / (\beta_i \cdot Q_{ap})\right)\right)}
\end{aligned}
\tag{26}
$$

where $NormR_{up}$ ($NormR_{dw}$) is the goodput R_{up} (R_{dw}) of all upstream (downstream) connections normalized with respect to the overall goodput $R_{up} + R_{dw}$.

JAIN'S FAIRNESS INDEX

Jain et al. in [3] define the Jain's fairness index as $J = \left(\sum_{i=1}^{n} x_i\right) \Big/ n\sum_{i=1}^{n} x_i^{\,2}$. If we substitute the goodput of a station $GPsta_i$ to x and the number of stations M to n, to be coherent with our notation, and exploiting our model (see Section 3) we get:

$$J=\frac{\left(\sum_{i=1}^{M}x_i\right)^2}{M\sum_{i=1}^{M}x_i^{\ 2}}=\frac{\left(\sum_{i=1}^{M}GPsta_i\right)^2}{M\sum_{i=1}^{M}GPsta_i^{\ 2}}=\frac{\left(\sum_{i=1}^{M}(Nup_i\cdot W_i+Ndw_i\cdot Tn(p,W_i))/(1+(Q_i/(\beta_i\cdot Q_{ap})))\right)^2}{M\sum_{i=1}^{M}((Nup_i\cdot W_i+Ndw_i\cdot Tn(p,W_i))/(1+(Q_i/(\beta_i\cdot Q_{ap}))))^2} \tag{27}$$

APPENDIX IV

VSB DESIGN IN PRESENCE OF UDP TRAFFIC

The handling of UDP traffic requires an extension to the structure of the VSB and to the logic of the WCE presented in section 4.2. The code of such extension is given in http://netgroup.uniroma2.it/Andrea_Detti/VSB/vsb_usb.tar.gz.

As regards the VSB, we argue that in presence of UDP traffic, uplink UDP packets must not be sent to the VSB, whereas UDP downlink packets must be sent to the VSB. If UDP traffic were sent to the IFB0, the rate control performed by the VSB would not change the amount of the WLAN capacity consumed by UDP uplink traffic; hence, the rate control is not effective in enforcing STA fairness and it is useless to drop or delay UDP uplink packets in the VSB. Conversely, if UDP downlink traffic is sent to the VSB, the rate control performed by the VSB does change the amount of WLAN capacity consumed by the UDP downlink traffic and therefore the rate control is effective in enforcing STA fairness.

As regards the WCE, we observe that in presence of uplink UDP greedy sources, if the PING were performed on STAs that run greedy UDP uplink sources, the WCE would detect high RTT values, which are insensitive to capacity *C*; consequently the WCE would perform a critical reduction of *C* that jeopardizes the exchange of TCP traffic. Therefore, it is fundamental to not consider RTT measurements that involve STAs supporting UDP uplink traffic. To find out which STAs are supporting UDP uplink traffic we can insert UDP TC FILTERS per STA in the scheduler and then monitor run-time the number of packets that these filters captured.

Furthermore, we observe that the presence of STAs supporting greedy UDP sources increases the channel access delay and the “base” RTT between AP and STAs with TCP traffic. As we stated in Section 4.2.1, the *RTT threshold* of the WCE is related to such base RTT, therefore in case of UDP uplink sources we need an automatic mechanism that adapts the *RTT threshold* by estimating base RTT. We observe that the estimation of the base RTT is a well-known issue of delay-based TCP congestion control and the literature provides different solutions to this problem [33]. With this regard, we extended the version of the WCE presented in 4.2.1 with a mechanism that estimates the base RTT. This estimation mechanism resembles the one of [34], i.e. every *N* (e.g. 8) samples of RTTs, the minimum value of RTT among these samples becomes the base RTT.

APPENDIX V

This section exemplifies how the Linux VSB can be programmed to enforce different types of fairness. In this example, we consider the case of up/down fairness and per-flow fairness.

Fig. 13 depicts the structure of a VSB devised to enforce up/down fairness, i.e. the WLAN capacity is fairly shared between the overall uplink TCP traffic and the downlink TCP traffic. The scheduler is composed of an HTB *root* class that outputs packets at *C* Mbit/sec, and two HTB leaf classes with guaranteed rate equal to *C/2* Mbit/sec. An HTB leaf class serves uplink traffic and the other HTB leaf class serves the downlink one.

Fig. 14 depicts the structure of a VSB devised to enforce per-flow fairness, i.e. the WLAN capacity is fairly shared among TCP flows. The scheduler is composed of an HTB *root* class that outputs packets at *C* Mbit/sec. The HTB root class drains packets from an inner scheduler that is a Stochastic Fair Queuing (SFQ).

From these examples, we can draw a conclusion on how to make the VSB "programmable" to different goals and working conditions: we set up an HTB *root* class drained at the rate defined by the WCE and, within this class, we insert a set of leaf classes or schedulers that implement the desired fairness goal.

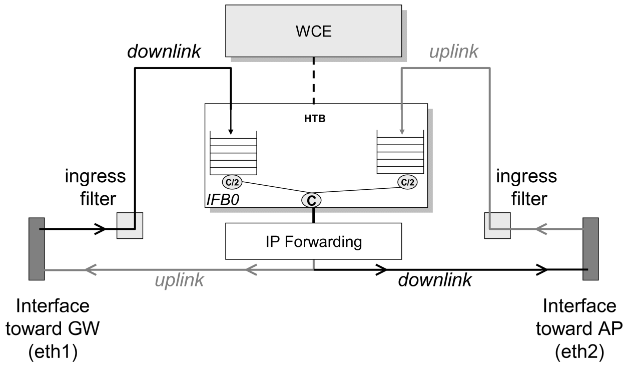

Fig. 13 Linux VSB implementation for up/down fairness.

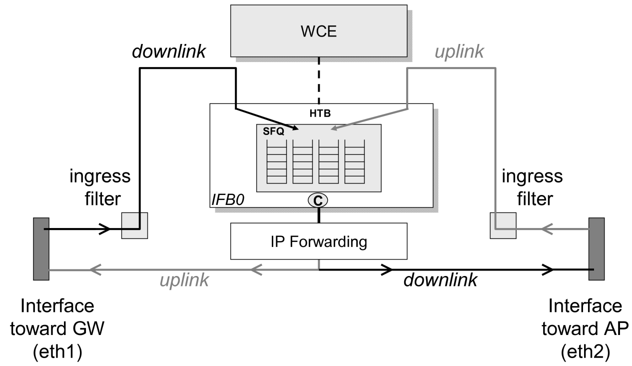

Fig. 14 Linux VSB implementation for per-flow fairness.