Experimental assessment of TCP performance over 802.11b

Francesco Vacirca and Francesca Cuomo
Infocom Department
University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome, Italy
Email: {vacirca,cuomo}@infocom.uniroma1.it

***Abstract*— This paper investigates performance behaviour of TCP sessions when supported over IEEE 802.11b Wireless LANs. We emphasize the interaction between uplink/downlink TCP fairness and link layer parameters or Access Point buffer provisioning. The novel aspect is that this investigation is fully made in an experimental environment. We prove that TCP suffers of some inequalities that derive to unfair bandwidth sharing between uplink and downlink. Our extensive experimental analysis shows the main effects of these inequalities on the TCP behavior and highlights some performance anomalies that are difficult to be measured via simulations. A simple, but effective, scheduling mechanism is then provided within the Access Point to alleviate the unfairness.**

I. Introduction

Wireless LAN standards are drawing the attention of the research and industrial community due to their potentialities in opening the market to the short range and high data rate wireless services in the local and hot spot areas. Technically speaking, the main strength of the most quoted standard, the IEEE 802.11, is the fully distributed nature of the access scheme, that provides cheap and easy-to-install components, able to operate in the unlicensed spectrum, still guaranteeing broadband capabilities.

Several works regarding 802.11 WLANs have been published: analytic models (e.g. [1]), simulation environments (e.g. [2] and [3]), experimental works (e.g. [3], [4], [5], [6] and [7]). In this paper we deal with experimental evaluation of 802.11b and we specifically point out the unfairness issue when uplink and downlink TCP flows compete in a WLAN.

The unfairness problem in a typical WLAN configuration made up of one Access Point (AP) and several mobile stations (STAs) has been highlighted by several works. Some papers stress the unfairness between uplink and downlink traffic and the disadvantages caused when the number of stations increases. The problem is mainly due to the fact that, while each station contends the medium to transmit its own traffic, the AP, with the same access mechanism, contends the medium to transmit the whole downlink traffic directed to the various STAs. To send the downlink traffic the AP relies on a unique MAC queue. The immediate conclusion is that, when the number of STAs increases, the downlink system performance decreases steeply because the AP transmission opportunities decrease inversely with the number of uplink competing flows. A proposal to reduce this drawback is given in [8] where authors operate at Logical Link Control (LLC) layer to solve the unfairness due to the 802.11b MAC mechanism. In the LLC AP a number of queues equal to the number of STAs is introduced; on the other hand, each STA is equipped with only one queue. A scheduling algorithm is then introduced to suitably pass the LLC frames to the MAC layer. In [8], to allow a fair share of the available bandwidth between uplink and downlink streams, AP MAC queue is provided with a lower contention window value than STAs' queues.

A controllable resource allocation method between uplink and downlink traffic flows has been proposed in [9]. This solution is based on measurements of the current load performed by the AP and on adapting some AP MAC parameters to control the fair sharing of bandwidth. The efficiency of the proposed method has been demonstrated by Markov analysis and computer simulations.

The unfairness between downlink and uplink becomes more critical when the flows exchanged in the WLAN are TCP-controlled. The combination of TCP mechanisms with an unfair bandwidth sharing increases the unbalancing between downlink and uplink flows giving rise to deep unfairness events. In [6] the TCP fairness over 802.11 is discussed by showing: i) the effect of the AP buffer size in an experimental test constituted by one mobile TCP sender and one mobile TCP receiver; ii) the up/down throughput ratio derived by carrying out an extensive simulation study. The main conclusions are that the buffer size in the AP plays a key role in the observed unfairness and that TCP throughput ratio between up/down could become very high ($\simeq 800$), thus giving rise to deep unfairness. Authors of [6] also propose a solution to alleviate the unfairness that is based on the manipulation of TCP advertised window. Simulative analysis of the proposed solution shows that the uplink/downlink fairness is achieved. Two problems however exist: 1) the solution is not tailored to TCP flows with different round trip times; 2) the advertised window manipulation requires that the AP is able to modify the TCP header fields and to re-compute the checksum; this could become time/resource consuming and results in a non-scalable solution.

The work in [10] considers unfairness at the TCP level due to different channel behaviors at the physical 802.11 layer.

This work is supported by the Italian Ministry for University and Research (MIUR) under the PRIN project TWELVE (http://twelve.unitn.it).

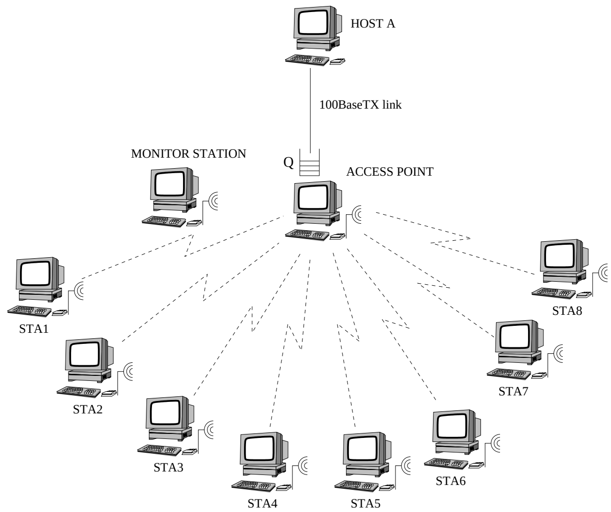


Fig. 1. Testbed layout.

Authors propose an algorithm which improves the fairness among STAs that experience short channel failures.
Finally, several works suggest to exploit the upcoming standard IEEE 802.11e to solve the unfairness by differentiating the MAC access methods in uplink and downlink. The paper in [7] investigates the use of the 802.11e MAC EDCF to address transport layer unfairness in WLANs. A simple solution is developed that uses the 802.11e AIFS and CW_{min} parameters to ensure fairness between competing TCP uploads. Authors in [11] present measurements made using 802.11e wireless test-bed and show how this new standard can be used to mitigate damaging cross-layer interactions between MAC and TCP. TCP ACK are prioritized by using suitable 802.11e MAC parameters in both the AP and the wireless STAs. This partially restores the fairness between uplink and downlink.

The aim of this paper is to deeply investigate the flow fairness in 802.11b by stressing the interaction between WLAN link layer parameters (e.g., ARQ retransmission persistence degree) and transport protocols. The novel aspect is that this investigation is fully made in an experimental environment constituted by an AP and up to 8 wireless STAs. Thanks to an extensive experimental analysis we are able to show the main effects of the 802.11b MAC on the TCP behavior and to propose a simple solution to alleviate the uplink/downlink unfairness.

The rest of the paper is organized as follows: Section II describes the testbed architecture and components. In Section III we report results of the measurement campaign focusing on the goodput behavior of downlink and uplink. In Section IV, we propose a simple mechanism, implemented in the AP, that mitigates the uplink/downlink unfairness. Main conclusions are drawn in Section V.

II. Description of the test-bed scenario

The test-bed reproduces a typical wired-cum-wireless scenario (Figure 1). It is composed of a 802.11b infrastructured WLAN and a dedicated 100BaseTX Ethernet link between the 802.11b AP and a PC (HOST A) that is the starting and/or the terminating point of all TCP connections. One PC acts as Access Point, 8 PCs as 802.11b client stations (from STA1 to STA8 in the figure), one PC equipped with a 802.11b network card monitors all the traffic on the air. The AP acts as a bridge between the wireless LAN and the 100BaseTX link exploiting the standard linux bridging functionalities [12]. In the test-bed topology, each STA is within the transmission range of all other STAs. All STAs are located in the same room and are motionless. The traffic is captured at:

- the monitor station, thus allowing the analysis of all the 802.11b frames exchanged on the air interface (this PC is equipped with a 802.11 wireless card that reads MAC headers and other 802.11b control information);
- the HOST A to analyze the TCP evolution.

The adopted wireless LAN cards are 3COM 3CRDW696 802.11b driven by the Intersil Prism 2.5 chip-set [13]. All cards utilize the same firmware version (including the AP and the monitor station). The choice of this chip-set has been motivated by the high reconfigurability of the relevant options and by the possibility to use the HOSTAP driver [14] to implement a AP system on a linux PC.

TABLE I
802.11B CONFIGURATION VALUES.

SIFS	$10\mu s$
DIFS	$50\mu s$
Slot Time	$20\mu s$
CW_{min}	32 Slot Time
CW_{max}	1024 Slot Time
MAC header	36 bytes
PHY header	24 bytes
MAC ack header	22 bytes
Bit Rate	11 Mbit/s

The key MAC parameters (e.g., DIFS, SIFS, MAC header, etc.) are set according to the IEEE 802.11b standard as reported in Table I. We set the MTU at 1500 bytes (fragmentation has been disabled) and the rate in the WLAN at 11Mbps. We disabled the RTS/CTS mechanism. Specific manufacturer features (out of 802.11b standard) have been disabled by default (e.g. power control, fallback rate control, etc.).
To study the effect of the AP buffer size on the uplink/downlink fairness, we exploit the standard linux traffic control tools [15] that enable us to modify the network interface card buffer size and queuing discipline.

The TCP version used in the experiments is the SACK-TCP [16] with window scaling and timestamp enabled [17]; TCP ACKs are sent according to the "delayed ACK" algorithm [18] (see [19] for a detailed insight into the linux TCP congestion control implementation).
Different software packages have been used in the test-bed: TCP traffic is generated to emulate bulk data transfer with a modified version of *ttcp* [20], changed to allow the *ttcp* server to accept multiple TCP connections simultaneously. The packet capturing tool is *tethereal* [21]. The traffic analysis and

the performance metrics computation have been performed with several *awk* [22] scripts.

Experiments have been performed varying the maximum number of transmission attempts M_t at the 802.11b link layer, varying the AP buffer size Q and the scheduling discipline, i.e. FIFO scheduling discipline in Section III and a custom scheduling discipline proposed to alleviate the TCP uplink/downlink unfairness problem in Section IV.

All the PCs are equipped with an additional 100BaseTX Ethernet card that it is used for control purposes. Experiments are configured and controlled by HOST A through *ssh* commands [23].

Each experiment lasts 500 seconds and all the metrics have been computed on the last 450 seconds of the experiments to remove the transient phase of TCP connections. A script runs at the end of every experiment to check consistency of the collected data and test-bed set-up: in particular, the number of active STAs and connections, the rate of all transmitted packets on the air and the absence of RTS/CTS packets are controlled[1]. The purpose of these checks is to enhance the test reliability.

III. Uplink/Downlink scenario

The aim of this section is to deeply investigate the flow fairness in 802.11b by stressing the interaction between WLAN link layer parameters (e.g., ARQ retransmission persistence degree), the AP buffer size configuration and transport protocols. The innovative aspect is that this investigation is fully made in an experimental environment constituted by an AP and up to 8 wireless STAs. Thanks to an extensive experimental analysis we are able to show the main effects of the 802.11b MAC on the TCP behavior and to propose a simple solution to alleviate the Uplink/Downlink unfairness.

In order to understand the TCP Uplink/Downlink fairness issue in the 802.11b scenario, we focus our analysis on two main metrics:

1) the ratio between the overall TCP Downlink goodput and overall TCP Uplink goodput[2];
2) the packet loss probability estimated through the analysis of packet traces captured at the HOST_A.

We considered two different settings, in two different sections. In the first one Uplink connection flow competes for the channel with one Downlink flow. In the second one 2 (or 4) Uplink connections share the medium with 2 (or 4) Downlink flows.

A. Single TCP connection scenarios

As a first step, we investigate the Downlink/Uplink fairness in case of 1 Uplink TCP flow and 1 Downlink TCP flow dependently on the number of retransmission attempts at link layer and on the AP buffer size. The scenario is of great interest because it shows that TCP congestion control mechanisms greatly impact on the performance of the system and the unfairness between flows arise even in a scenario typically considered fair from a 802.11 point of view.

We first consider the case of 1 TCP Uplink connection (STA1-HOST_A) and 1 Downlink connection (HOST_A-STA2)[3]. The scenario is depicted in Figure 2(a). It is to notice that the sender of the Uplink connection is different from the receiver of the Downlink connection. In the remainder of this work, we refer to this as "scenario 1".

The TCP buffer sizes have been increased in order to avoid that the bottleneck of the TCP mechanism is the receiver advertised window. In this way we are able to capture all the effects of the congestion control interacting with 802.11 MAC access mechanisms.

Figure 3 shows the ratio between Downlink and Uplink goodputs (a) and the global goodput (b) versus the maximum number of transmission attempts for different values of the AP buffer size, whereas Figure 4 shows the Uplink (a) and Downlink (b) goodput in the same conditions. From both figures, it is clear that the fairness between Uplink and Downlink is never reached. The ratio between the goodputs varies drastically with the number of transmission attempts and the AP buffer size. When the number of transmission attempts is zero - no retransmissions allowed - Downlink goodput is around 250 packets/s whereas Uplink one is about 125 packets/s independently of the AP buffer size. When M_t increases, the gap between Downlink and Uplink performance increases. When $M_t = 3$, the ratio between Downlink and Uplink goodput reaches values between 2 and 6. In this case the AP buffer size influences the unfairness, i.e. large AP buffers favor the Downlink connection, whereas a small buffer size smothers the unfairness problem. When M_t further increases, the situation reverses and the Uplink connection reaches a higher goodput than the Downlink one. In the M_t = 8 case, for example, the Downlink goodput is lower than 100 packets/s and the Uplink one is between 400 and 350 packets per seconds. Even in this case, a large AP buffer alleviates the throttling of Downlink performance. An insight into the packet loss probability of the Uplink and Downlink connections (Figures 5(a) and 5(a) respectively), shows that:

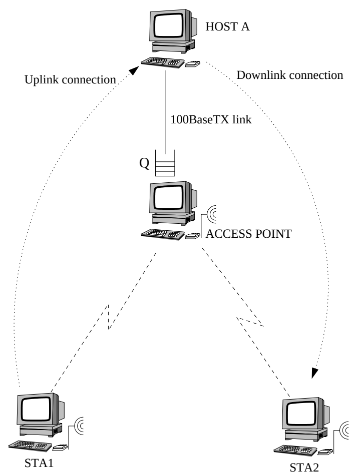
- when M_t=1 the Uplink packet loss probability is higher than Downlink packet loss probability, allowing Downlink connection to outperform the Uplink one;
- the Uplink packet loss probability is a decreasing function of M_t;
- the Downlink packet loss probability starts decreasing with M_t, but then increases again.

The upper and the lower parts of Figure 6(a) show the evolution of the estimated number of Downlink and Uplink outstanding packets respectively in the scenario 1 when $M_t =$ 1 and the AP buffer size is 100 packets. As it is possible to

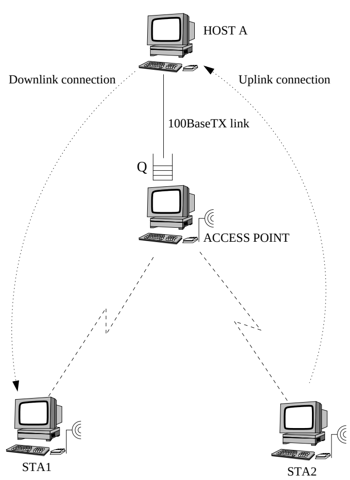
[1]These checks are important in a scenario where it is difficult to distinguish between standard features and features developed by 802.11b card producers; e.g. by default the Intersil Prism 2.5 chipset decreases the transmission rate after a pre-defined number of unsuccessful transmission attempts.

[2]The goodput is defined as the throughput at TCP layer, excluding retransmitted packets.

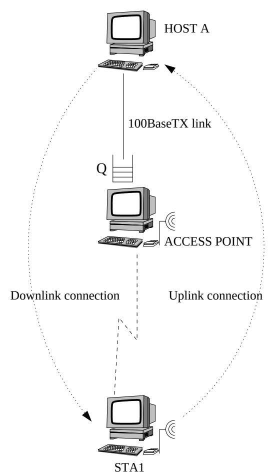
[3]The inverse case, STA2-HOST_A and HOST_A-STA1 connections, presents the same results.



(a) scenario 1

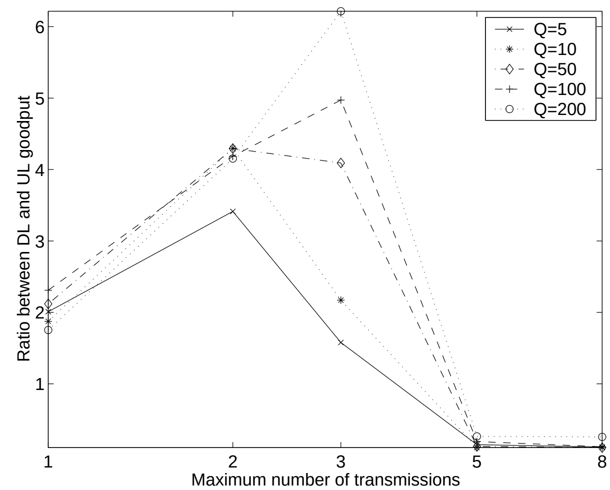


(b) scenario 2

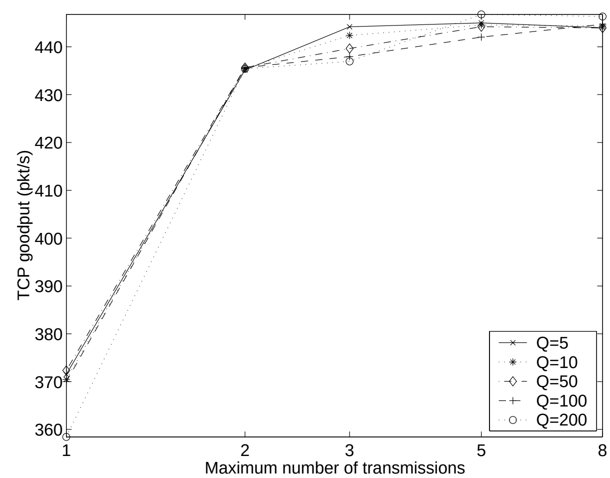


(c) scenario 3

Fig. 2. Single connection reference scenarios.



(a)



(b)

Fig. 3. Downlink/Uplink goodputs (ratio and overall) vs. M_t, for different values of AP buffer sizes in the dl1-ul1 scenario.

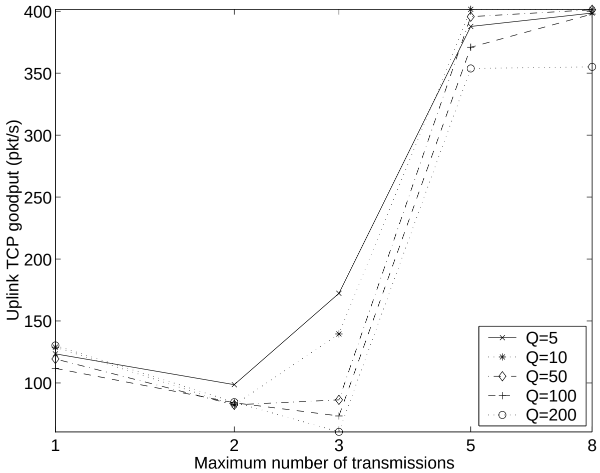
notice, on average the number of outstanding packets is higher in the Downlink case. To a lower packet loss probability it corresponds in fact a higher number of outstanding packets, thanks to the TCP congestion control.

When M_t increases the situation reverses and the Uplink connection obtains a higher goodput. Figure 6(b) shows the number of outstanding packets in the dl1-ul1 scenario when M_t is 8 and the AP buffer size is 100 packets. In this case the number of Downlink outstanding packets follows the typical saw-tooth behaviour of the TCP congestion window, whereas in the number of Uplink outstanding packets is never halved provoking the greedy behaviour of TCP flows. We recognized three different working regions: i) when M_t is 1, ii) when M_t is between 2 and 5 transmission attempts and iii) when M_t is great enough to cover the most of the packet losses on the wireless channel.

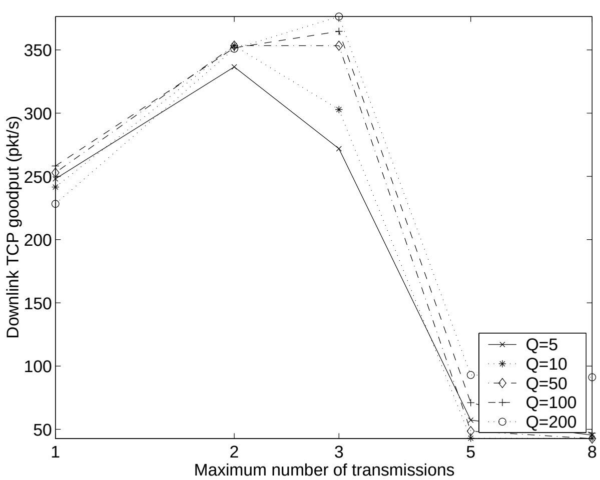
To analyze the source of these phenomenon, we create some ad-hoc network scenarios to focus on every single issue. First of all we verified in the network scenario 2 (inverting TCP sender and receiver hosts, Figure 2(b)) that results are not due to a misconfigurated and/or underprovisioned hardware. Experiments, not reported here, show the same values depicted in the previous figures, confirming that previous results do not depend on the specific hardware and on the position of STA1 with respect to STA2.

Verified that the performance results are not due to the particular configuration, the first question to solve is why the achieved goodput of Downlink is greater than the Uplink one for small values of M_t.

The phenomenon can be analyzed considering that when the packet loss probability is high, TCP congestion control

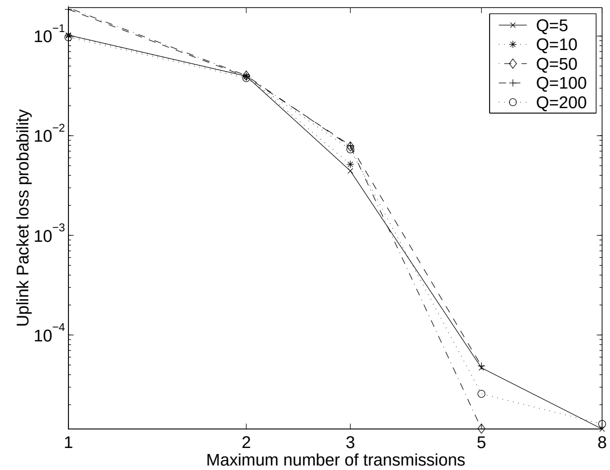


(a)

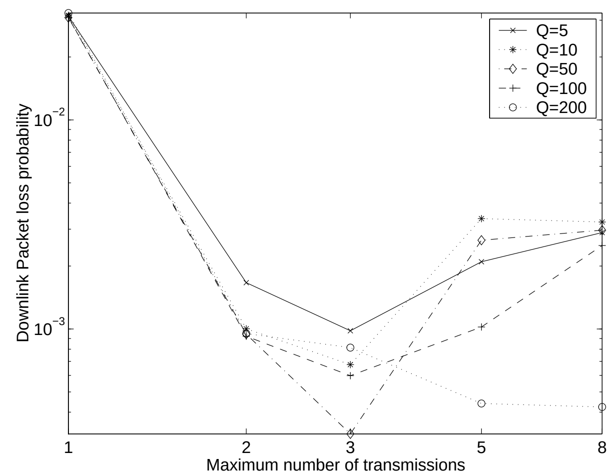


(b)

Fig. 4. Uplink (a) and Downlink (b) goodput vs. M_t, for different values of AP buffer sizes in the dl1-ul1 scenario.



(a)



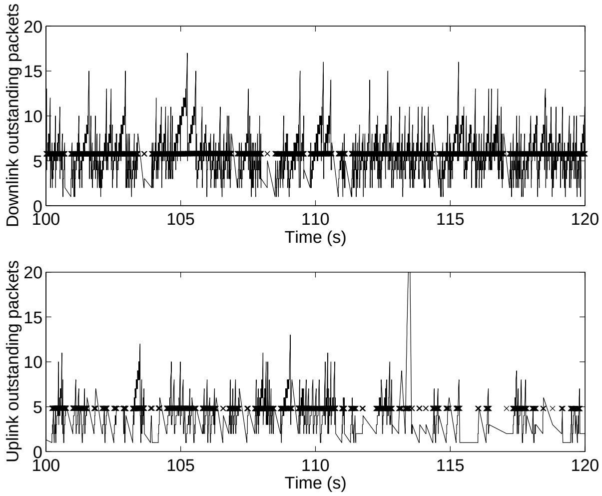
(b)

Fig. 5. Uplink (a) and Downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes in the dl1-ul1 scenario.

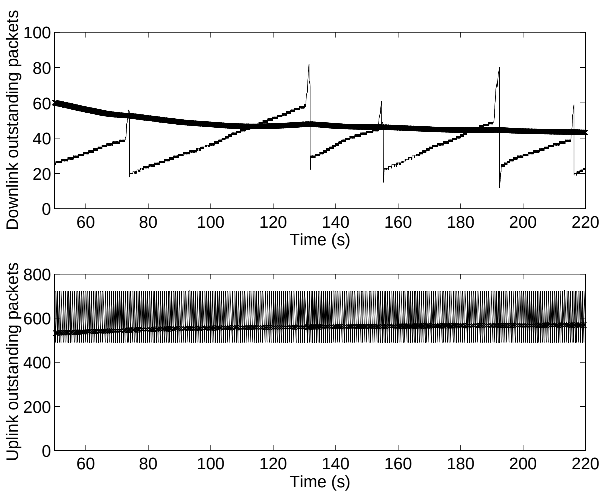
algorithms are not able to fully exploit the available bandwidth and the obtained rate does not saturate the buffer. In this condition, supposing that the Uplink TCP connection achieves a rate of x packets per seconds and that the Downlink achieves a rate of y packets/s, the sending rate of the involved stations at the MAC layer is x for STA1, $y/2$ for STA2 and $y+x/2$ for the AP, because to a data sending rate of x corresponds a rate in the opposite direction of $x/2$ TCP ACK packets per second. Supposing that at the beginning of the experiment x is equal to y, it implies that the AP has more transmission opportunities and hence a lower packet loss probability of STA1. Moreover TCP congestion control reacts to losses reducing the sending rate, amplifying the effect of the scenario asymmetry.

As a second configuration scenario, we changed the receiver buffers' settings to focus the analysis on the AP buffer size and the interaction with the congestion window of Uplink and Downlink connections.

Figures 7 show the ratio between goodputs, when the TCP receiver buffer is limited to 64 kbytes, i.e. about 44 packets (Figure 7(a)) and when it is limited to 32 kbytes, i.e. 22 packets (Figure 7(b)). It is worth remembering that the number of outstanding packets, i.e. the effective window, is the minimum between the congestion window $cwnd$ and the receiver's advertised window $rwnd$ that reflects the status of the receiver buffer. Two main effects can be noticed: i) the AP buffer size has a smaller influence on the unfairness characteristics than in the case of unlimited receiver buffer (except in the case of very small buffers, i.e. Q=5 and Q=10); ii) when the number of transmission attempts increases the 802.11 bandwidth is shared fairly and the goodput ratio reaches 1.

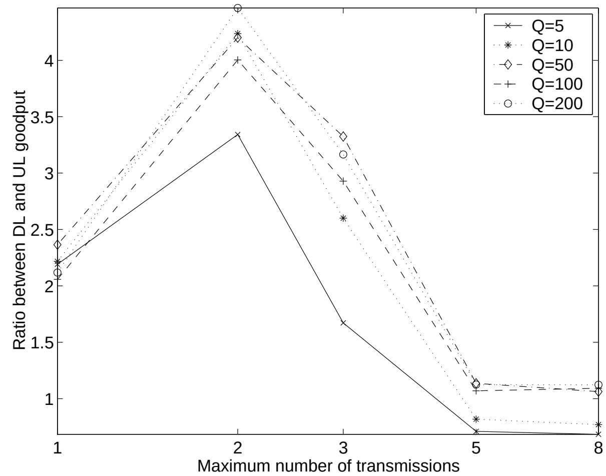


(a) $M_t = 1$

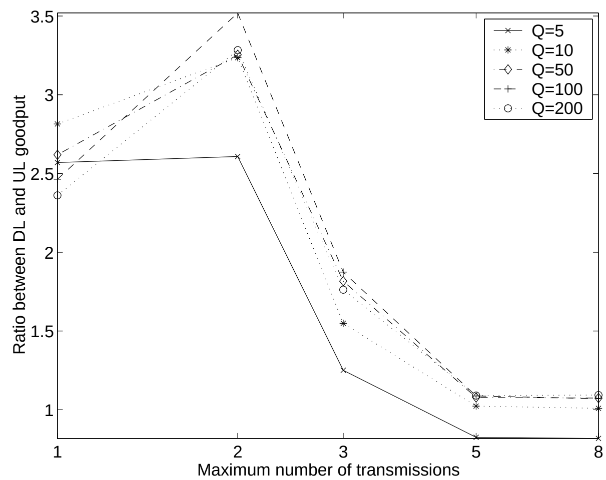


(b) $M_t = 8$

Fig. 6. Downlink and Uplink outstanding packets in the dl1-ul1 scenario with Q=100.



(a)

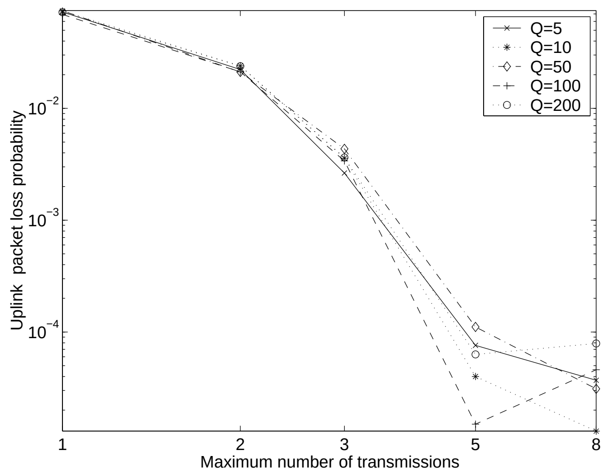


(b)

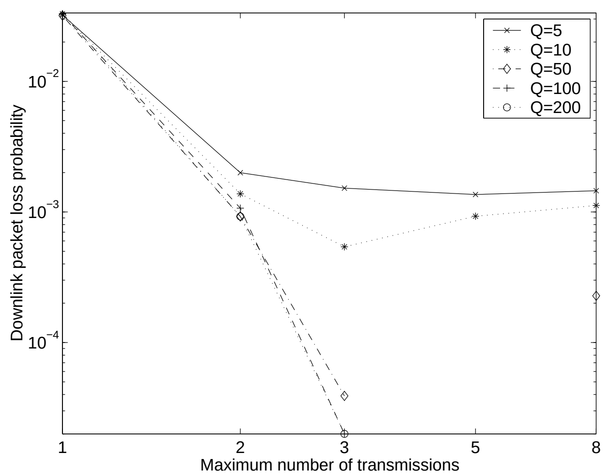
Fig. 7. Downlink/Uplink goodput ratio vs. M_t, for different values of AP buffer sizes, with limited receiver buffer, 64K (a) and 32K (b).

This particular scenario shows that limiting the receiver buffer (and hence the advertised window) the number of outstanding packets decreases. Consequently the number of losses due to congestion in the AP buffer decreases and the Uplink sending rate is less aggressive. On the other hand, it pinpoints that the Uplink and Downlink connections can be differently influenced by the AP buffer size. In the Downlink case, when the number of outstanding packets is not limited by the receiver buffer, packets are lost in the AP buffer and TCP congestion control algorithms react to this loss by diminishing the number of packets in flight; this favors the Uplink connection that it is not affected by packet losses. In fact, the AP buffer does not impact directly on Uplink connection because only ACK packets are lost in the AP buffer and no data packets are lost in the sender buffer. As reported in Figures 8(a) and 8(b) where the Uplink and Downlink packet loss probability is shown (with receiver buffer size of 64 kbytes), the Downlink packet loss probability decreases with M_t when the AP buffer size is large leading to a fair sharing of the bandwidth. When the AP buffer size is quite small, e.g. Q equals to 5 or 10 packets, the advertised window is larger than the maximum congestion window allowed by the bandwidth-delay product and the buffer size, and packet losses occur in the AP buffer.

Then we create the scenario 3 in which the station involved in the transmission and the one involved in reception is the same: Uplink connection, STA1-HOST_A and Downlink connection HOST_A-STA1. This scenario, depicted in Figure 2(c), is introduced to point out the effect of a perfectly symmetric scenario. Figure 9 depicts the goodput ratio of scenario 3 in

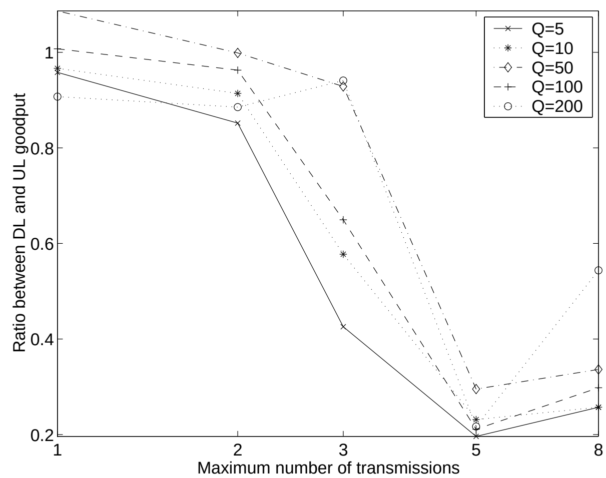


(a)

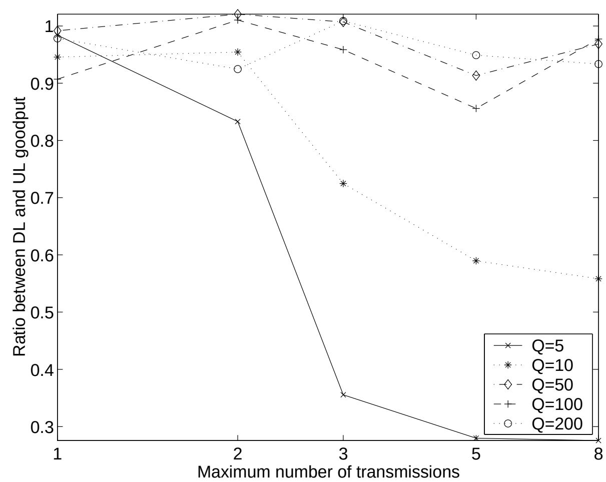


(b)

Fig. 8. Uplink (a) and Downlink (b) goodput ratio vs. M_t, for different values of AP buffer sizes in the scenario 1 with receiver buffer of 64 kbytes.



(a) unlimited receiver buffer



(b) 64 kbytes receiver buffer

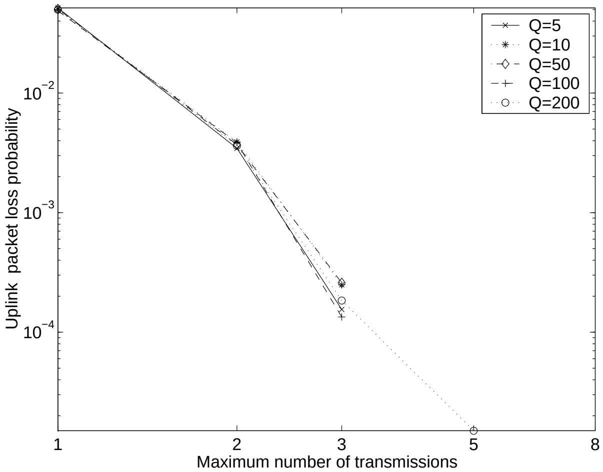
Fig. 9. Downlink/Uplink goodput ratio in the scenario 3 vs. M_t, for different values of AP buffer sizes.

the case of unlimited receiver buffer size (a) and 64 kbytes receiver buffer size (b). In this scenario, the behaviour of the ratio between Downlink and Uplink goodput is very different from scenario 1. First of all, when M_t is 1, the ratio is about 1 indicating that the fairness between Uplink and Downlink is optimal. The fairness is maintained when M_t increases, the peak presents in the scenario 1 does not appear any more. When M_t increases and the TCP receiver buffer is large, data packet losses in the AP buffer size favors Uplink flow and the ratio between Downlink and Uplink goodputs decreases. By comparing Figure 9(a) and 9(b), we note that

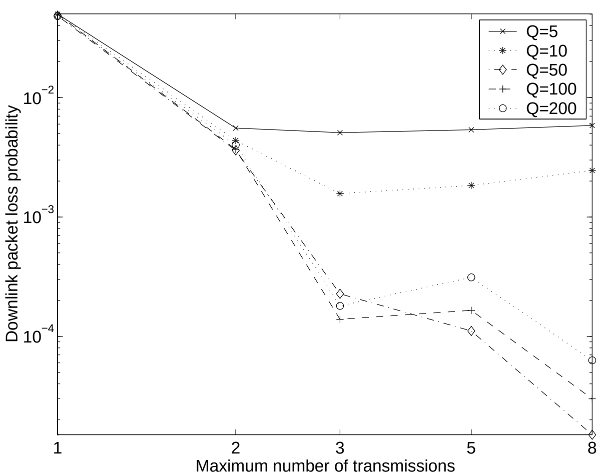
- The unfairness between Uplink and Downlink disappears when the network scenario is symmetric with small M_t.
- The unfairness due to Downlink losses in the AP buffer is alleviated when the congestion window of the Downlink sender is limited by the receiver advertised window and the access point buffer size is large.

In scenario 3, limiting the receiver buffer size leads to the optimal working point where the fairness between Uplink and Downlink is independent of the number of transmissions at link layer and of the AP buffer. However, limiting the advertised window to obtain the fairness (as proposed in [6]) is not an optimal choice when several connections share the same link while connections' round trip times are not the same. In the latter case in fact, the achieved goodput per flow would be inversely proportional to the round trip time experienced.

Figure 10 shows the packet loss probability in the scenario 3 (both Uplink and Downlink) when the receiver buffers are limited to 44 packets. It is to be observed that the packet loss probability when $M_t = 1$, is the same, leading to a fair

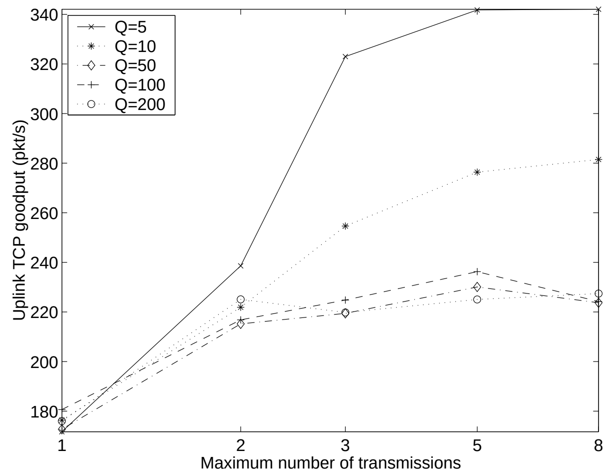


(a)

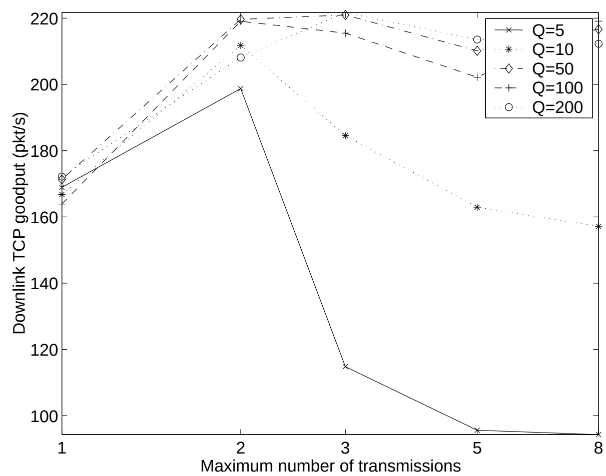


(b)

Fig. 10. Uplink (a) and Downlink (b) packet loss probability in the scenario 3 with 64 kbytes receiver buffers vs. M_t, for different values of AP buffer sizes.



(a)



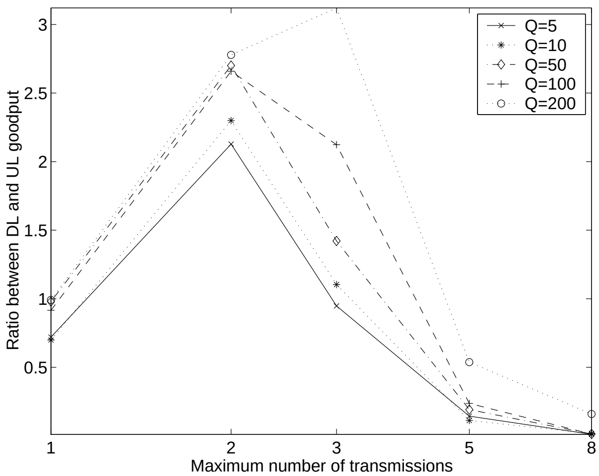
(b)

Fig. 11. Uplink (a) and Downlink (b) goodput in the scenario 3 with 64 kbytes receiver buffers vs. M_t, for different values of AP buffer sizes.

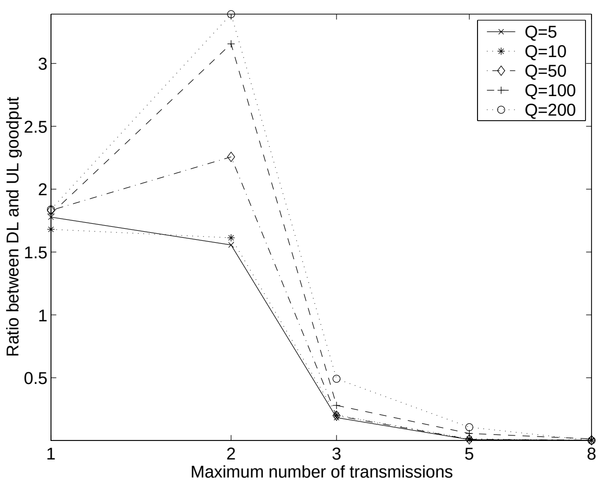
share of the bandwidth. When M_t increases both packet loss probabilities decrease and the fairness is maintained. Only when the AP buffer size is very small, the Downlink and Uplink packet loss probabilities split leading to unfairness due to data packet losses in the AP buffer.
Figure 11 completes previous results.

B. Multiple TCP connections scenarios

As far as regards multiple TCP connection configuration, two scenarios are considered: in the first one there are 2 TCP Uplink connections (STA1-HOST_A and STA2-HOST_A) and 2 Downlink connections (HOST_A-STA3 and HOST_A-STA4). In the second scenario there are 4 TCP Uplink connections (between STA1-4 and HOST_A) and 4 Downlink connections (between HOST_A and STA5-7). In the remainder of this work, we refer to the first and the second scenario as dl2-ul2 and dl4-ul4 respectively. Given the symmetric traffic characteristic, the ideal Downlink/Uplink goodput ratio is 1:1. In general, in typical WLAN scenarios, the symmetric assumption is not respected and the most of the traffic is in the Downlink direction. The asymmetric traffic scenario is not considered in this work because beyond the purpose of the present analysis.
Figures 12 depicts the Downlink/Uplink goodput ratio versus the maximum number of transmission attempts (M_t) at 802.11b link layer. The metric is measured for different values of the AP buffer size in the dl2-ul2 and dl4-ul4 scenarios respectively. It is worth noticing that the goodput ratio is influenced significantly by both the maximum number of retransmission attempts and the AP buffer size. As far as the



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 12. Downlink/Uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

behavior as a function of M_t is concerned, it can be noted that:

- when M_t=1, in case of dl2-ul2 scenario, the Downlink/Uplink goodput ratio is about 1:1 when the AP buffer size is large (between 50 and 200 packets), whereas the Uplink is favored when the buffer size is smaller. In the dl4-ul4 case, Downlink connections achieve a higher goodput with respect to the Uplink connections;
- when the number of link layer retransmission increases (M_t=2) the Downlink goodput increases in spite of Uplink performance. The larger is the AP buffer, the more the Downlink goodput is higher than Uplink;
- by increasing M_t, the Uplink connections seize the available bandwidth and the Downlink connections starve. A larger buffer slightly alleviates the phenomenon.

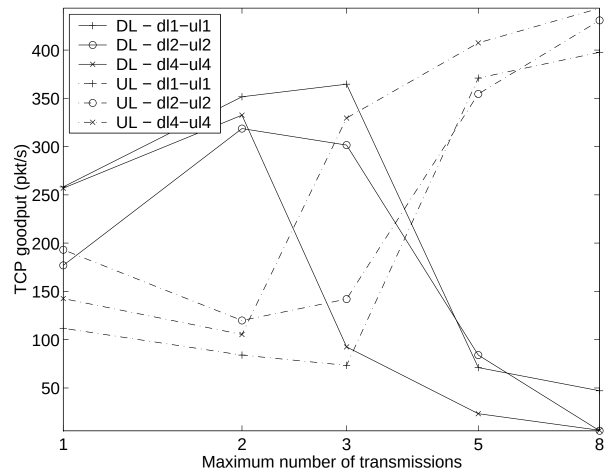
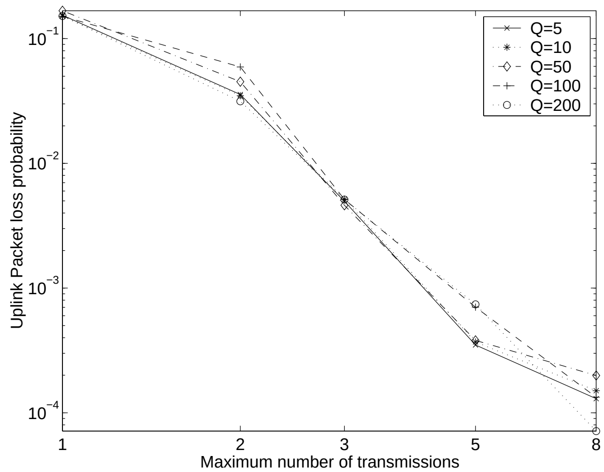


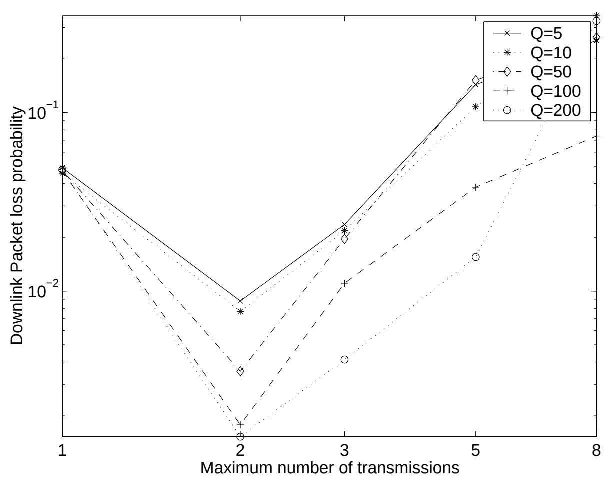
Fig. 13. Uplink and Downlink goodputs vs. M_t, with Q=100.

Figure 13 depicts the Downlink goodput (solid line) and Uplink goodput (dot and dashed line) varying M_t, in dl1-ul1 (one TCP session in Downlink and one TCP session in Uplink), dl2-ul2 and dl4-ul4, when the AP buffer size is 100 packets.

Let us concentrate on the simple case dl1-ul1: for M_t=1, the Downlink behaviour is satisfactory in terms of Downlink/Uplink goodput ratio. However, it could be noticed that the overall goodput is about 370 packets/s (250 packets/s in Downlink and 120 packets/s in Uplink). An useful expedient to improve the overall goodput in 802.11b is to increase the number of allowed transmission attempts at link layer to overcome the collision problem and minimize packet losses. The resulting performance anomaly is that, the overall goodput increases as expected, however, the Downlink/Uplink unfairness reverses, favoring Uplink connections in spite of Downlink. This is mainly due to the different behaviour of TCP sender entities: TCP senders of Uplink connections are directly connected to the bottleneck link, whereas in the Downlink case the bottleneck is in the wireless portion of the network and the TCP sender is not directly connected to it. A better insight into these phenomena is given in Figure 14 where TCP data packet loss probability (estimated exploiting TCP packet retransmissions) is reported. Figures 14(a) and 14(b) depicts respectively the Uplink and Downlink packet loss probability in the dl4-ul4 scenario. We notice that, when M_t=1, the Uplink packet loss probability is higher than Downlink one, allowing Downlink goodput to outperform Uplink goodput. When the number of retransmission attempts increases, the Uplink packet loss probability decreases monotonically, whereas the Downlink packet loss probability decreases till M_t=2 and then it increases again. While the Uplink packet loss probability is mainly due to packet collisions on the wireless channel, the Downlink packet loss probability is the combination of two phenomena. On the one hand there are packet losses due to collisions that decrease when M_t becomes larger. On the other hand, there are losses due to congestion in the AP



(a)



(b)

Fig. 14. Uplink (a) and Downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.

buffer. These losses increase when M_t increases, because the congestion windows of TCP connections are able to inflate and fill the AP buffer. While Uplink connections contribute to congestion the AP buffer with ACK packets, their performance is not influenced because data packets are not lost. It is to be considered that in traffic saturation conditions [1], every device achieves an equal portion of bandwidth (including the AP), leading to a ratio 1:(n+1) between Downlink and Uplink, where n is the number of STAs. With TCP as transport protocol, Downlink flows are greatly influenced by congestion in the AP buffer and the ratio between Downlink and Uplink goodput decreases below the 1:(n+1) ratio. Our experimental results confirm the influence of AP buffer size on the TCP Uplink/Downlink fairness problem (as also shown in [6]). With respect to [6], we show that the phenomenon is more complex

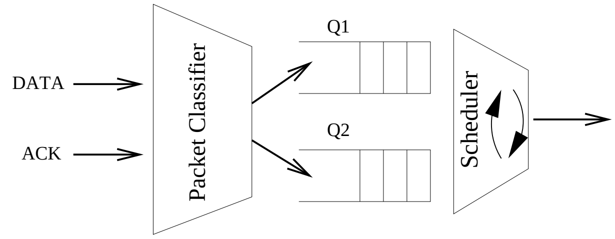


Fig. 15. Scheduling discipline.

and several factors influence the equilibrium between Uplink and Downlink connection goodput. We can summarize these factors as follows:

- Small AP buffer sizes favor Uplink flows by increasing the Downlink data packet loss probability;
- Large AP buffer sizes alleviate the throttling of Downlink;
- A high number of transmission attempts favors the Uplink;
- The Downlink/Uplink goodput ratio is unbalanced even in the dl1-ul1 scenario. The increasing number of supported TCP flows worsens the unbalancing phenomenon.

IV. TCP Uplink-Downlink separation via traffic control

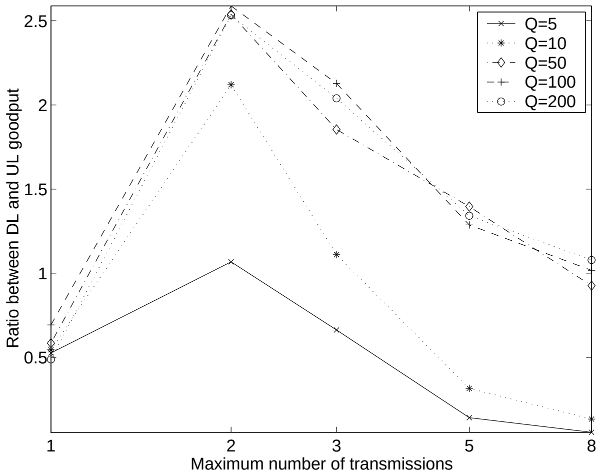
To increase the fairness between Uplink and Downlink in case of TCP flows we propose a simple traffic control mechanism. As well known, TCP sending rate is controlled by the rate the ACKs are received by the sender entity. The idea is to control the aggressiveness of the Uplink flows by reducing the ACK rate issued by the AP. In this way the AP is able to control the throughput of the Downlink versus the Uplink one.

In the most of TCP implementations (see [19] and [24]), an ACK is generated at the TCP receiver side, every two data packets (according to the delayed ACK algorithm [18]).

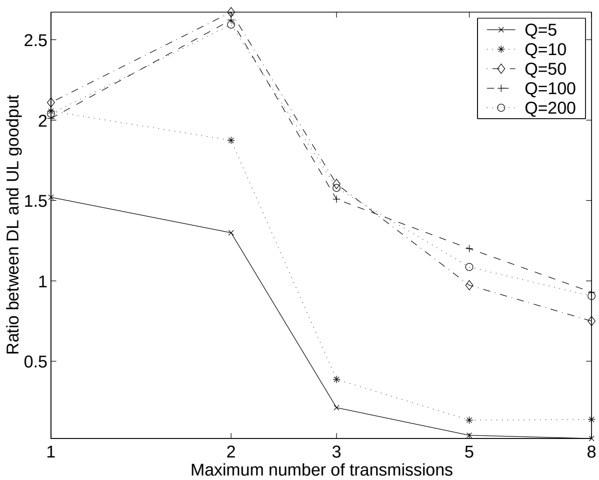
We implemented a simple scheduling discipline, at the AP buffer, that forces to 1:3 the ratio between ACK and DATA packets flowing through the AP towards the STAs. Since the reception of one ACK allows the transmission of two new packets, the uplink rate is forced to be the same of the downlink one because the ACK rate generates a doubled uplink data rate identical to the downlink data rate.

The scheduling scheme is represented in Figure 15. It is composed of a packet classifier that inspects packet characteristics (in our case TCP header fields) and forwards packets respecting user-defined rules to different queues. In case of our schedule, the classifier distinguishes between TCP data packets and TCP ACKs and enqueues them in Q1 and Q2 respectively. The scheduler is the entity that serves Q1 and Q2 in a weighted round robin fashion with a ratio of 2:3 for TCP data queue (Q1) and 1:3 for the ACK queue (Q2).

Figures 16(a) and 16(b) depict the ratio between the overall Uplink goodput and the Uplink one in the dl2-ul2 and dl4-ul4 scenarios respectively. Comparing Figure 12(a) with Figure 16(a) and Figure 12(b) with Figure 16(b), it is evident that



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 16. Downlink/Uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

in the region where Downlink goodput is higher than Uplink goodput the scheduler is not able to increase Uplink/Downlink fairness. When M_t increases and the AP buffer size is not too small (larger or equal to 50 packets), the scheduler is able to keep the goodput ratio about 1:1 indicating that Uplink and Downlink connections are experiencing the same goodput. Benefits of the proposed scheduler can be pinpointed by having a look at Figure 17: in all the considered scenarios, the Downlink and Uplink goodputs converge to same value when M_t increases. It is to be noticed that already for $M_t > 5$ performance target is reached.

An insight into the packet loss probability experienced by TCP connections shows that the packet loss probability of the Uplink does not change with the customized scheduling discipline (comparison between Figure 14(a) with Figure 18(a)), whereas

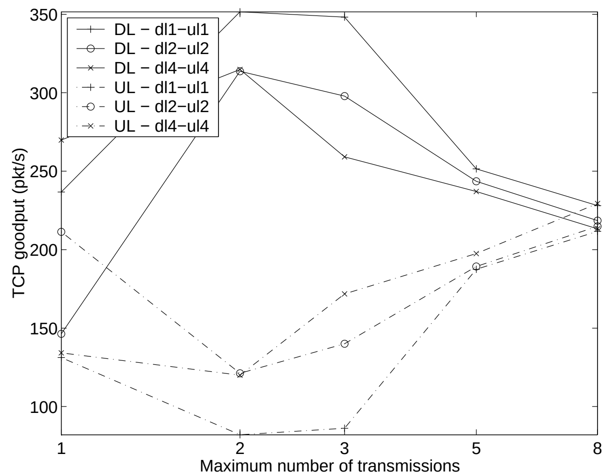


Fig. 17. Uplink and Downlink goodputs vs. M_t, for the custom scheduling discipline (Q=100).

the Downlink packet loss probability trend changes and the packet loss probability is reduced. The separation of the ACK scheduling in the AP presents two benefits: i) the TCP sender rate in the STAs decreases, ii) the ACK packet pressure in the AP MAC queue is reduced, diminishing Downlink data packet losses.

It is worth noticing that our proposal is designed to equally share the 802.11b bandwidth between Uplink and Downlink not considering the number of active Uplink and Downlink connections. A more complex mechanism that estimates the number of active Uplink and Downlink connections and dynamically adapts the weights of the scheduler is needed to maintain the Downlink/Uplink goodput ratio proportional to the ratio between active Downlink and Uplink connections.

V. Conclusions and Further Work

In this work, we deeply investigate the flow fairness in 802.11b in an experimental test-bed constituted by an AP and up to 8 wireless STAs. The aim is to highlight the interaction between WLAN link layer parameters and transport protocols. Thanks to an extensive experimental analysis we were able to show the main effects of the 802.11b MAC on the TCP behavior stressing in particular the effect of the AP buffer size, the number of link layer retransmission attempts and of concurring flows. The novelty of this contribution is a circumstantial report on TCP performance in a real 802.11b test-bed. To solve some performance anomalies we implemented in the AP a simple packet scheduling policy that succeeds in alleviating the uplink/downlink unfairness. The scheduling is a simple software module that can be included in the AP and that acts only by reading the TCP header fields of the exchanged TCP segments and by storing them in different queues handled with different priorities. It results easy to implement, scalable and not time consuming. Future work will be dedicated to adapt the proposed scheduling discipline to dynamic traffic conditions and asymmetric scenarios.

REFERENCES

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Function", IEEE Journal of Selected Areas in Telecommunications, Vol. 18, No. 3, pp. 535-547, March 2000.

[2] M. Methfessel et al., "Vertical Optimization of Data Transmission for Mobile Wireless Terminals," IEEE Wireless Communications Magazine, December 2002.

[3] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "Performance Anomaly of 802.11b", Proc. of IEEE INFOCOM, 2003.

[4] G. Xylomenos, G. C. Polyzos, "TCP and UDP Performance over a Wireless LAN," Proc. of IEEE INFOCOM, 1999.

[5] A. Vasan and A. U. Shankar, "An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs," technical report, available at http://www.cs.umd.edu/ shankar/selected-publications.html

[6] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," Proc. of IEEE INFOCOM 2003, San Francisco, CA, USA, March 2003.

[7] D.J. Leith, P. Clifford, "Using the 802.11 e EDCF to Achieve TCP Upload Fairness over WLAN Links," Proc. IEEE WiOpt'05, pp. 109-118, April 2005.

[8] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, M. Meo, "Smart Traffic Scheduling in 802.11 WLANs with Access Point," Proc. IEEE VTC 2003 Orlando, USA, October 2003.

[9] S.W. Kim, B.S. Kim, Y. Fang, "Downlink and Uplink Resource Allocation in IEEE 802.11 Wireless LANs," IEEE Transactions on Vehicular Technology, Vol. 54, No.1, pp. 320-327, January 2005.

[10] M. Bottigliengo, C. Casetti, C. F. Chiasserini, M. Meo, "Short-term Fairness for TCP Flows in 802.11b WLANs", Proc. of IEEE INFOCOM 2004.

[11] A.C.H. Ng, D. Malone and D. J. Leith, "Experimental evaluation of TCP performance and fairness in an 802.11e test-bed," Proc. ACM SIGCOMM Workshop on Experimental Approaches To Wireless Network Design and Analysis, Philadelphia, USA, August 2005.

[12] "Linux Bridging HOWTO," http://bridge.sourceforge.net/howto.html

[13] "RM025 - Prism Driver Programmers Manual," June 2002, Intersil.

[14] hostap, software available at http://hostap.epitest.fi .

[15] B. Hubert and alii, "Linux Advanced Routing & Traffic Control HOWTO," http://lartc.org/howto

[16] J. Mahdavi et al., "TCP Selective Acknowledgment Options", RFC2018, IETF, October 1996.

[17] V. Jacobson, R. Braden and D. Borman, "RFC 1323 - TCP Extensions for High Performance," IETF May 1992.

[18] M. Allman, V. Paxson, W. Stevens, "RFC 2581 - TCP Congestion Control," IETF, April 1999.

[19] P. Sarolahti, A. Kuznetsov, "Congestion Control in Linux TCP," In Proc. of USENIX'02, June 2002.

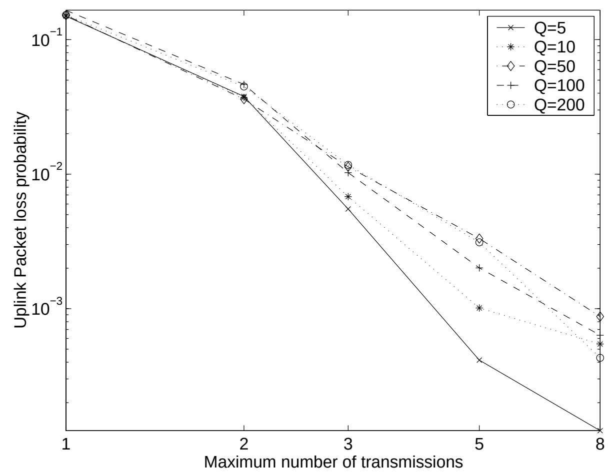
[20] ttcp, software available at http://ftp.arl.mil/ mike/ttcp.html.

[21] ethereal 0.10.4 release, available at http://www.ethereal.com.

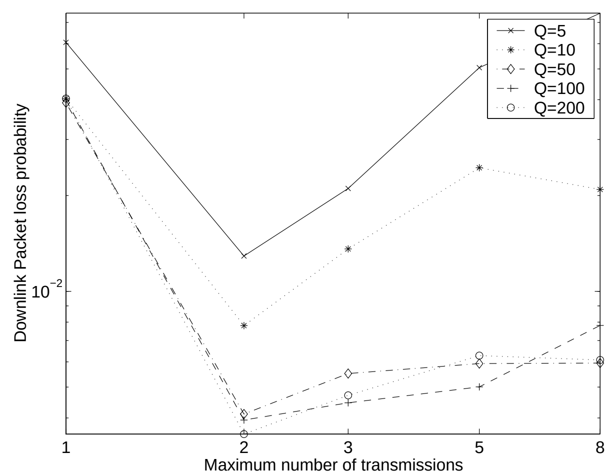
[22] www.gnu.org/software/gawk/gawk.html.

[23] http://www.openssh.com/.

[24] D. MacDonald, W.Barkley, "Microsoft Windows 2000 TCP/IP Implementation Details," available at http://www.microsoft.com.



(a)



(b)

Fig. 18. Uplink (a) and Downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.