Experimental results on the support of TCP over 802.11b: an insight into fairness issues

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*— Great attention has been dedicated, in the recent years, to the WLAN standards that are 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. The aim of this paper is to deeply investigate traffic issues in 802.11b networks by emphasizing the interaction between WLAN link layer parameters or Access Point buffer provisioning with uplink/downlink TCP fairness. The novel aspect is that this investigation is fully made in an experimental environment. A great portion of flows that are exchanged in a WLAN are TCP-based (e.g. FTP flows). We prove, with real experiments, 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.**

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 a 1:1 ratio is maintained, resulting in fair allocation bandwidth. Two problems however exist: 1) the solution is not tailored to TCP flows with different round trip

This work has been partially supported by the PRIN TWELVE project founded by the Italian Research Ministry.

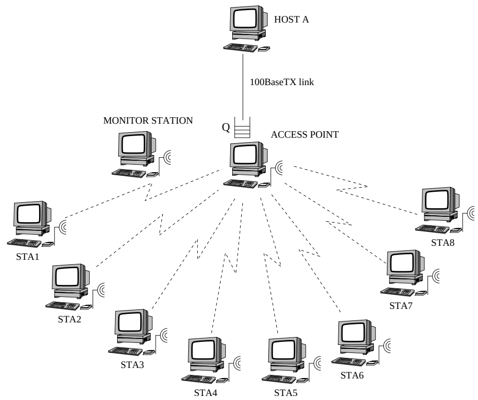


Fig. 1. Testbed layout.

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. 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 which shows 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 where it is easier to analyze the TCP evolution.

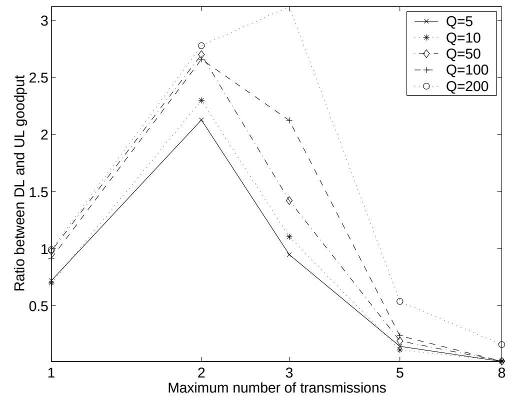
The adopted wireless LAN cards are 3COM 3CRDW696 802.11b driven by the Intersil Prism 2.5 chip-set [13]. All cards utilize the same firmware version (including the AP and the monitor station). The choice of this chip-set has been motivated by the high reconfigurability of the relevant options and by the possibility to use the HOSTAP driver [14] to implement a AP system on a linux PC. In fact, the Prism 2.5 chip-set (that is used both in WLAN host cards and in commercial APs) can be used to drive an AP in two modes: *Firmware-based* and *Host-based* AP mode. In the first mode, used in commercial APs, the chip-set utilizes a *tertiary* firmware for the AP functionalities such as authentication, association and forwarding of MAC frames. In the second mode, used in our testbed, the most time-critical actions are performed by the firmware (i.e. frame transmission, frame reception, beacon and probe frame handling), whereas other functionalities (such as authentication and association) are demanded to the host driver (the HOSTAP driver in our testbed). Moreover, the Prism MAC firmware implements a monitor mode that enables a 802.11b card to receive and pass to the host driver all frames with a PLCP header correctly received, irrespective of MAC frame check sequence errors, along with baseband layer information such as signal and noise levels.

The key MAC parameters (e.g., DIFS, SIFS, MAC header, etc.) are set according to the IEEE 802.11b standard. 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.

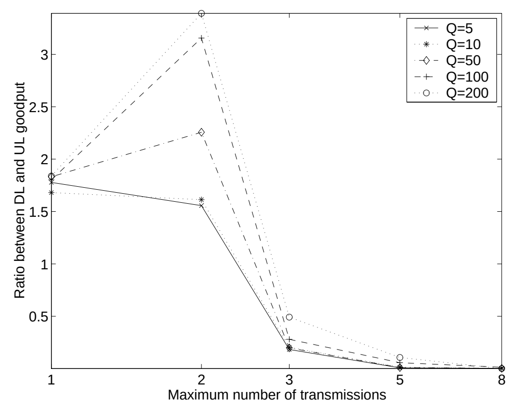
The TCP version used in the experiments is the SACK-TCP [16] with window scaling and timestamp enabled [17]; TCP ACKs are sent according to the "delayed ACK" algorithm [18] (see [19] for a detailed insight into the linux TCP congestion control implementation). The TCP buffer sizes have been increased 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.

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 metric 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 and a custom scheduling discipline proposed to alleviate the TCP uplink/downlink unfairness problem). 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.



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 2. Downlink/uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

III. Analysis of the Experimental Results

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.

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 paper we refer to the first and the second scenario as dl2-ul2 and dl4-ul4 respectively. Given the symmetric characteristic of the scenarios, the ideal downlink/uplink goodput ratio is 1:1. We choose the symmetric scenario since it allows a comprehensive understanding of the balancing between uplink and downlink. In general, in typical WLAN scenarios, the symmetric assumption is not respected and the most of the traffic is in the downlink direction.

Figures 2(a) and 2(b) depict 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, with a FIFO scheduling discipline, 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 AP buffer size. As far as the 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

[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.

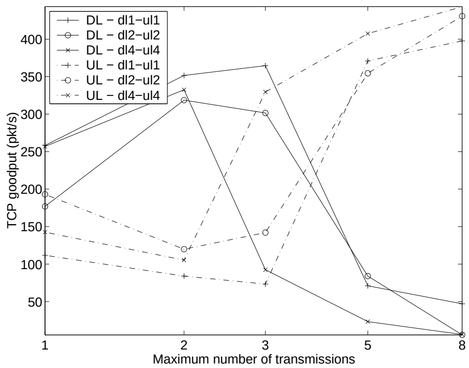
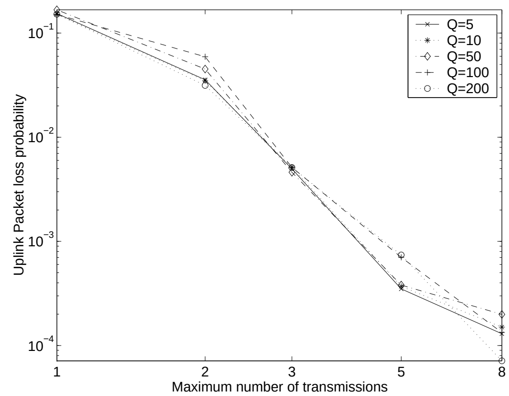
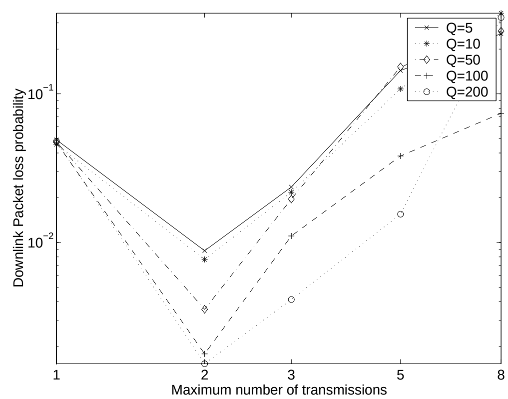


Fig. 3. Uplink and downlink goodputs vs. M_t, for the FIFO scheduling discipline (Q=100)



(a)



(b)

Fig. 4. Uplink (a) and downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.

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.

Figure 3 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 for $M_t > 3$, 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 not in the access link. A better insight into these phenomena is given in Figure 4 where TCP DATA packet loss probability (estimated exploiting TCP packet retransmissions) is reported. Figures 4(a) and 4(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 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 and several factors influence the equilibrium between uplink and downlink connection goodput. We can summarize these factors as follows:

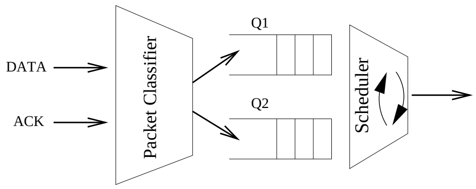


Fig. 5. Scheduling discipline.

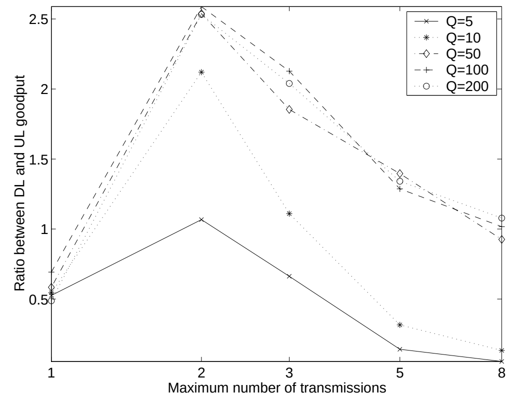
- Small AP buffer sizes favor uplink flows by increasing the downlink DATA packet loss probability;
- Large AP buffer sizes alleviate the throttling of downlink;
- A high number of transmission attempts favors the uplink;
- The downlink/uplink goodput ratio is unbalanced even in the dl1-ul1 scenario.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

IV. TCP UPLINK-DOWNLINK SEPARATION VIA TRAFFIC CONTROL

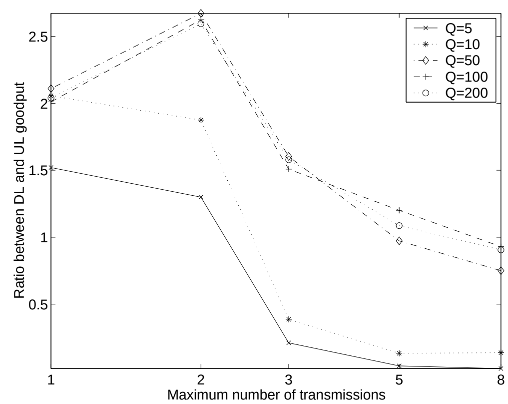
To increase the fairness between uplink and downlink in case of TCP flows we propose a simple traffic control mechanism. As well known, TCP sending rate is controlled by the rate the ACKs are received by the sender entity. The idea is to control the aggressiveness of the uplink flows by reducing the 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 5. 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 6(a) and 6(b) depict the ratio between the overall uplink goodput and the uplink one in the dl2-ul2 and dl4-ul4 scenarios respectively. Comparing Figure 2(a) with Figure 6(a) and Figure 2(b) with Figure 6(b), it is evident that 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



(a) Scenario dl2-ul2



(b) Scenario dl4-ul4

Fig. 6. Downlink/uplink goodput ratio vs. M_t, for different values of AP buffer sizes.

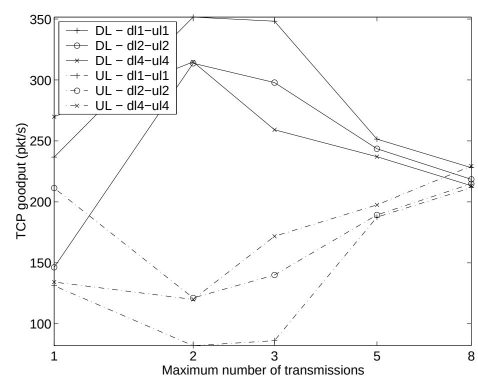
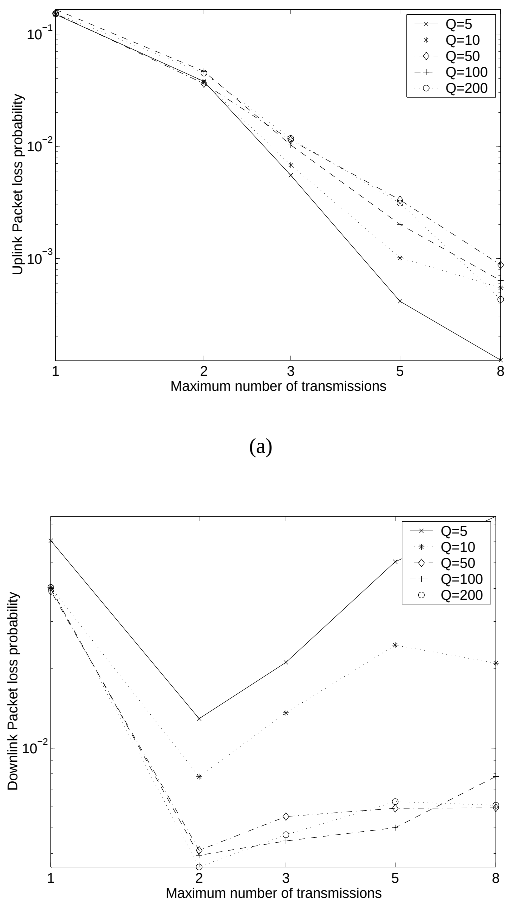


Fig. 7. Uplink and downlink goodputs vs. M_t, for the custom scheduling discipline (Q=100).

at Figure 7: 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



(a)

(b)

Fig. 8. Uplink (a) and downlink (b) packet loss probability vs. M_t, for different values of AP buffer sizes.

the uplink does not change with the customized scheduling discipline (comparison between Figure 4(a) with Figure 8(a)), whereas 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

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.