A mechanism to enforce TCP Fairness in 802.11 wireless LANs and its performance evaluation in a real test-bed

Nicola Blefari-Melazzi, Andrea Detti, Alessandro Ordine, Stefano Salsano

DIE, University of Rome “Tor Vergata”
{blefari, andrea.detti, alessandro.ordine, stefano.salsano}@uniroma2.it

Abstract

In this paper, we study the problem of maintaining fairness for TCP connections in wireless local area networks (WLANs), based upon the IEEE 802.11 standard and operating in DCF mode. Although this mode of operation ensures fair access to the medium at the MAC level, it does not provide fairness among TCP connections. TCP unfairness may result in significant degradation of performance, leading to users perceiving unsatisfactory quality of service. We propose and analyze a solution capable of enabling TCP fairness with minimal additional complexity. The proposed solution is based on utilizing an IP rate-limiter with an adaptive rate control mechanism and does not require modifying existing standards at the MAC or network layers. The solution is fully compatible with existing devices and can be integrated either within the access point or within and external device (e.g., the network gateway). The performance analysis is performed in a real test-bed and proves the feasibility and effectiveness of our mechanism. To allow fellow researchers to reproduce our work we published on the WEB all the implementation code.

1. Introduction

Wireless LANs based on the IEEE 802.11 standard and working in the so-called “infra structured” mode provide mobile stations (STAs) with access to the wired network by means of Access Points (AP). Most existing STAs operate according to the so-called DCF mode. This is the scenario that we consider in this paper.

It is important for such infra-structured WLANs, especially when offering public access to the Internet, that any TCP engine would be capable of starting a connection with negligible delay, as well as achieving and maintaining a reasonable throughput. On the contrary, it is well know that the interaction of TCP with 802.11 protocols results in: i) unfairness between upstream and downstream TCP connections [1][2]; ii) unfairness among upstream TCP connections [2].

Both these phenomena show up when there are packet losses in the AP downlink buffer. These losses may reach significant values also because the AP does not enjoy a privileged access to WLAN capacity with respect to STAs, even if it has to handle more traffic than a single STA.

As regards the first phenomenon, downstream TCP connections (i.e., connections having the source in the fixed network) may be penalized with respect to the upstream ones (i.e., connections having the source in STAs). This is due to the fact that when packet losses occur in the AP downlink buffer [7], downstream TCP connections lose segments and reduce their send-rate [12]; on the other side, upstream TCP connections lose TCP ACKs, which does not affects the growth of their send-rate. This implies that downstream TCP connections reduce their requests of radio resource while upstream TCP connections exploit radio resource left unused.

As regards the second phenomenon, some upstream TCP connections may enjoy a greater throughput than other upstream TCP connections. This “intra-upstream” unfairness is due to heavy losses of TCP ACKs that may occur in the AP downlink buffer; such losses affect differently the TCP send-rate, as a function of the current value of the congestion window [5].

In both cases, it may happen that the unfairness reaches “critical” levels, characterized by a complete starvation of TCP connections.

Several papers proposed solutions to alleviate unfairness phenomena. The proposed solutions can be classified according to the layer at which they operate (MAC [11], IP [2][10], TCP [1][8]) or according to the level of fairness that they achieve (per-connection fairness [8][9][10], aggregate upstream/downstream fairness [2], per-station fairness [1][11], etc.).

In this paper, we pursue an aggregate upstream/downstream fairness and our mechanism operates at the IP layer. This paper is a follow-up of [2], where we propose an IP rate-limiter that controls the uplink traffic, i.e., the traffic going from STAs to the AP. The aim of this controller is to reduce the aggressiveness of upstream TCP connections and to provide an equal share of radio resources to the aggregate upstream/downstream traffics. To the best of our knowledge, the approach that we propose in [2] is the only one that can be implemented not only in the AP but also in a nearby router, without requiring changes to the 802.11 standard, nor any enhancement to mobile stations. This is an important advantage, as it adds flexibility and reduces time to market.

With respect to [2], the novelty of this paper is two-fold: i) the rate of the IP rate-limiter is not anymore a fixed parameter, as in [2], but it can be varied adaptively, thus avoiding potential wastes of precious radio resources; ii) we evaluate the system performance by means of a real test-bed, instead of resorting to simulations. Again, to the best of our knowledge, this is the first example of a mechanism to provide fairness in a WLAN that it is implemented and tested in the field.

More details on our solution, together with a complete performance evaluation based on simulations can be found in [4].

As for the organization of the paper, in section 2 we introduce the IP rate-limiter with adaptive rate control; in section 3 we describe our test-bed and present its performance; finally, in section 4 we draw our conclusions.

2. IP rate-limiter with adaptive rate control

Fig. 1 depicts our reference scenario. It includes a local area network and an external fixed network (e.g., the Internet). The local area network has both a wireless and a wired part. The wireless part is connected to the wired part through an access point; in turn, the wired part is connected to the external fixed network by means of a gateway router. The network bottleneck is the wireless part: the wired capacity is much greater than the wireless one.

In order to reduce the aggressiveness of upstream TCP connections, we control the overall wireless uplink traffic by means of a rate-limiter. The rate-limiter can be located not only within the AP, but also on any device crossed by the wired uplink traffic. As a matter of fact, the wired uplink traffic is equal to the wireless one, as the wireless part is the network bottleneck.

The rate-limiter operates at IP level and is implemented with a Token Bucket Filter (TBF) [3] characterized by two parameters: R and B_{bucket}. The parameter R is the rate at which tokens are generated (expressed in bit/s); the parameter B_{bucket} is the bucket size, (expressed in bytes). The TBF operates only as a policer, i.e., if not enough tokens are contained in the bucket when a packet arrives, then the packet is dropped. The dropping of packets trigger the TCP congestion control, which reduces the rate of upstream connections; thus, the TBF indirectly controls the send-rate of upstream TCP connections.

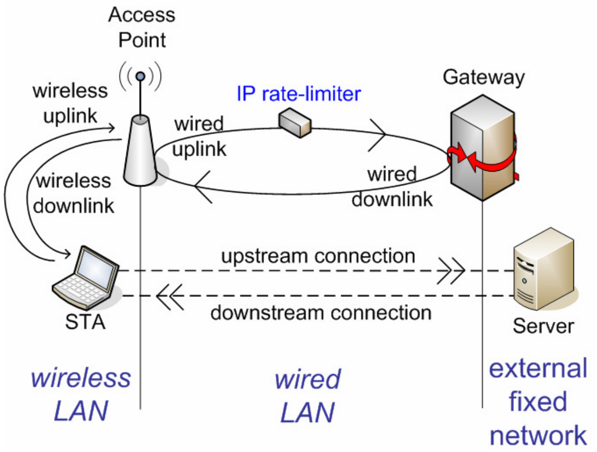

Fig. 1 – Reference scenario

In [2] the values of the TBF parameters are chosen once and for all as R=C/2 and B_{bucket} = 300 kbytes, where C is the maximum TCP throughput achievable in the WLAN, measured at the IP-MAC wireless. The rationale of this choice is discussed in [4]. The quantity C is evaluated as 4.6 Mbit/s for 802.11b WLANs. However, a TBF with a constant value of R has two major drawbacks: 1) if downstream connections use less than half of the WLAN capacity, the capacity of upstream connections is un-necessarily limited to the rate C/2, leading to a waste of radio resource; 2) this approach implicitly assumes that the overall capacity C is known, which is not always the case; in fact, the STAs can operate at different physical rates (e.g., from 1Mbit/s to 11Mbit/s in 802.11b).

To solve the first problem, we introduce an algorithm to dynamically adapt the rate R in the range from C/2 to C. The resulting device is named *IP rate-limiter with adaptive rate control*. The rate R is chosen as the greatest value for which no packet loss occurs in the AP downlink buffer. In fact, the absence of packet losses in the AP downlink buffer avoids the unfairness, while maximizing the value of R, under the no loss constraint, avoids to un-necessarily reduce uplink traffic. However, we note that, in presence of greedy downstream TCP connections, R can not be less than $C/2$.

To solve the second problem, the WLAN capacity C is estimated at run-time and not anymore assigned a fixed value as in [2].

As regards the parameter B_{bucket}, we note that the sensitiveness of our mechanism with respect to it is rather limited (as we verified by means of both test-bed and simulations results). Thus, we did not judged as worthwhile the introduction of a run time control for this parameter. As a matter of fact, the choice of a proper value for this parameter can be seen more as a fine tuning aspect of our approach rather than as an issue with important performance consequences.

The implementation of the adaptive rate control algorithm is reported in Tab. 1 and can be explained as follows. Each T_p milliseconds a new control round starts. At the end of control round *#k*, the rate R is increased or decreased depending on if packet loss in the downlink AP buffer occurred or not, during the control round.

If no packet is lost, we infer that the WLAN capacity is not saturated; thus, the rate R is increased by R_{step} (expressed in bit/s). In any case, the rate R is upper bounded by *C_max*, which is the maximum expected capacity of the WLAN measured at the IP-MAC interface (e.g., about 6 Mbit/s in case of 802.11b).

On the contrary, if a loss occurred then it means that during the #*k* round the WLAN capacity is saturated; in such case it is possible to estimate the WLAN capacity as the sum of the uplink (R_{up}) and downlink (R_{down}) throughput, i.e., $C= R_{up}+ R_{down}$. Here, the throughput is defined as the average bit rate crossing the IP-MAC wireless interface in the round *#k*.

Once that the value of C is updated, R is reduced of R_{step}, until it reaches the lower bound $R=C/2$.

It can be that R becomes greater than the actual value of R_{up}; this happens after a long sequence of lossy control rounds; in such case a decrease of R_{step} is not enough to have an effect on the send-rate of upstream TCP connections. For this reason, if the previous control round is lossless (`first_loss=true`), then R is reduced to min ($R-R_{step}$, $R_{up}-R_{step}$).

Another consideration regards the parameter R_{step} that controls the maximum speed of rate increase and decrease (equal to R_{step} / T_p). Too small values of R_{step} may make difficult the startup of new downstream TCP connection (that need a certain amount of free capacity) and may reduce the efficiency when the bandwidth needs to be increased after a sudden reduction of the capacity required by downlink connection. On the contrary, too large values of R_{step} may give rise to significant throughput oscillations due to interactions with the underlying TCP congestion control mechanisms.

Tab. 1 - Adaptive rate control algorithm

```
Each time k*Tp

if (None_AP_packet_loss) {
  first_loss = true;
  R = min (R+Rstep,C_max);
} else {
  C = Rdown + Rup;
  if (first_loss) {
     first_loss = false
     R = max(min(R-Rstep ,Rup-Rstep),C/2)
  } else {
     R = max (R-Rstep, target_rate);
  }
}
```

Finally, it is worth to note that the estimation of the throughputs R_{up} and R_{down} can also be done outside the AP, in a router of the access network (we did this by using the IPchains firewalling code, in a Linux gateway). Also, the estimation of the packet loss in the AP downlink buffer can be done outside the AP by means of an SNMP query to the MIB of the AP. This means that *IP rate-limiter with adaptive rate control* does not have to be necessarily implemented within the AP.

3. Experimental results

Fig. 2 depicts the test-bed layout, while Fig. 3 shows a picture of the test-bed.

The test-bed includes ten Personal Computers (PCs) and an access point Cisco Aironet 1200. A PC plays the role of the server residing in the fixed network. Another PC is the network gateway. The remaining 8 PCs are the STAs, equipped with 802.11b cards at 11Mbit/s. All the PCs mount the Microsoft Windows XP operating system, with the exception of the network gateway, running Linux, kernel version 2.6.17.13, Kubuntu distribution.

The gateway has two Fast Ethernet interfaces, named *eth0* and *eth2*. The *eth0* interface connects the gateway to the fixed-network server, while *eth2* connects the gateway to the AP. As regards the addressing space, the gateway routes between two different networks: 192.168.100.0/24 on the *eth0* side; and 10.0.0.0/8 on the *eth2* side.

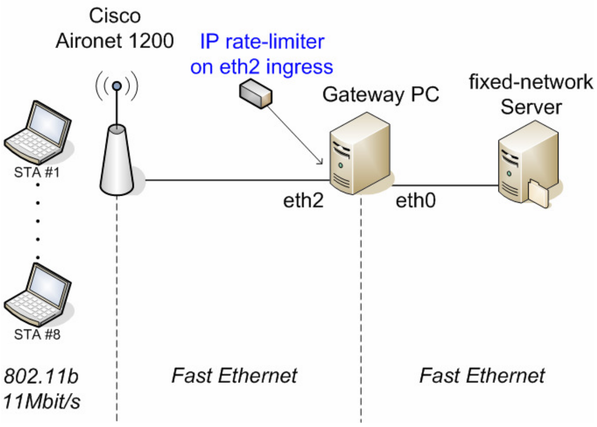

Fig. 2 – Test-bed layout

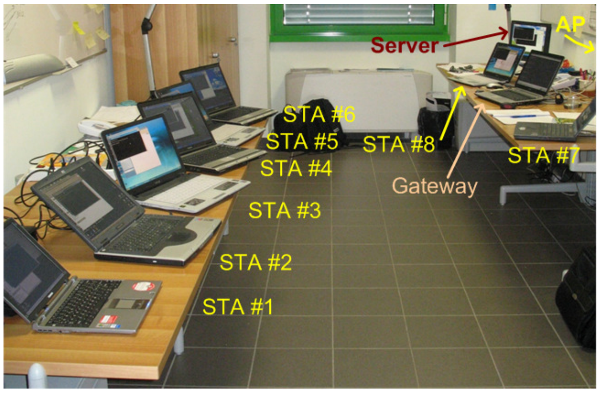

Fig. 3 – The test-bed

The IP rate-limiter with adaptive rate control is located within the gateway and operates at the ingress of the *eth2* interface. It has two components: i) the TBF policer built-in in the Linux kernel that acts as the IP rate-limiter; ii) the user-space application *rl* that computes the dynamic value of *R* as specified in Tab. 1. The original TBF policer of the Linux kernel has been modified to allow a user-space application to vary at run time the TBF rate. The *rl* application has been developed from scratch. All the software code is available in [6].

We evaluated the system performance with two different TCP traffic models: static and dynamic. The first one allows to assess the performance in steady state, while the dynamic traffic model allows to highlight what happens when traffic conditions change.

In the case of *static traffic model*, we assume that a single STA supports one downstream and one upstream TCP connection with the fixed-network server. As a consequence, if we let N the number of STAs, then the number N_{up} of upstream connections and the number N_{down} of downstream connections are equal to N. Each TCP connection is active for the entire duration of the test, that is 5 minutes.

In the case of *dynamic traffic model*, we use 6 STAs and the number N_{up} of upstream and N_{down} of downstream active connections versus the time is reported in Tab. 2.

Tab. 2 – Number of connections versus time in case of dynamic traffic model

N_{up}	6	6	6	0	6
N_{down}	6	0	6	6	6
Time (s)	20÷100	100÷200	200÷300	300÷400	400÷500

The main TCP parameters are: segment payload=1460 bytes; max congestion window=65536 bytes. We repeat the tests two times: with and without IP rate-limiter with adaptive rate control.

The merit figures that we consider are the upstream, downstream and total (i.e., the sum of these two components) TCP goodput. The TCP goodput is defined as the number of useful bits per unit of time received at the connection sink.

3.1. Test-bed results without rate-limiter

Fig. 4 reports the average upstream, downstream and total goodput as a function of the number N of STAs in the case of static traffic model.

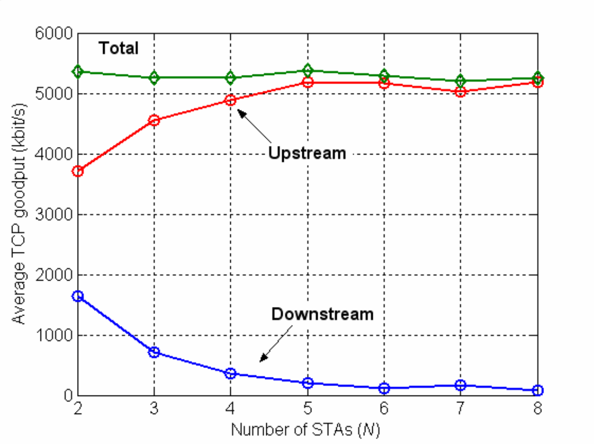

Fig. 4 - Average upstream, downstream and total TCP goodput as a function of *N* (static traffic, without rate-limiter)

The total goodput that we measure is in line with the theoretical value that one could expect, considering the fact that the TCP implementation of Microsoft Windows uses the delayed ACK policy and sends a TCP ACK each two received TCP segments.

For all the values of N reported in the figure, the downstream goodput is significantly lower than the upstream goodput. This is an evidence of the upstream/downstream unfairness. A critical unfairness occurs for N greater than 4 STAs.

Fig. 5 shows the average upstream and downstream goodput for single STAs with N=6 and a static traffic model. We observe also an intra-upstream unfairness, even though not a critical one. For instance, STA #4 perceives an upstream goodput equal to about two times the upstream goodput of STA #1. In [4] we show by means of simulations that also the intra-upstream unfairness may become critical, in the sense that some upstream connections are unable to start. This phenomenon occurs when a large number of STAs are present, thus, our test bed does not reproduce this behavior.

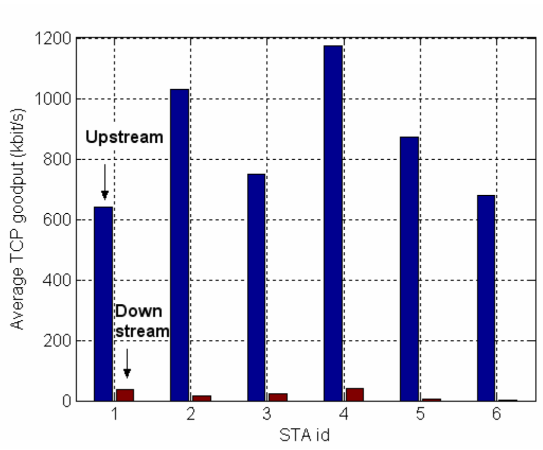

Fig. 5 - Average upstream and downstream TCP goodput for single STA (*N*=6, static traffic, without rate-limiter)

Fig. 6 reports the instantaneous upstream, downstream and total TCP goodput for N=6 and with a static traffic model. Most of the total goodput is made up of upstream goodput, while the downstream connections are almost starved.

Finally, Fig. 7 shows the instantaneous upstream, downstream and total TCP goodput with a dynamic traffic model. We can see that the downstream traffic survives only when there are not upstream connections, i.e., between time interval 300-400 s.

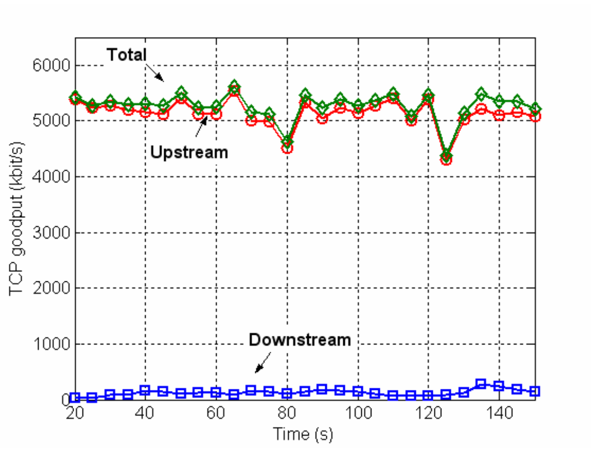

Fig. 6 - Instantaneous upstream, downstream and total TCP goodput (*N*=6, static traffic, without rate-limiter)

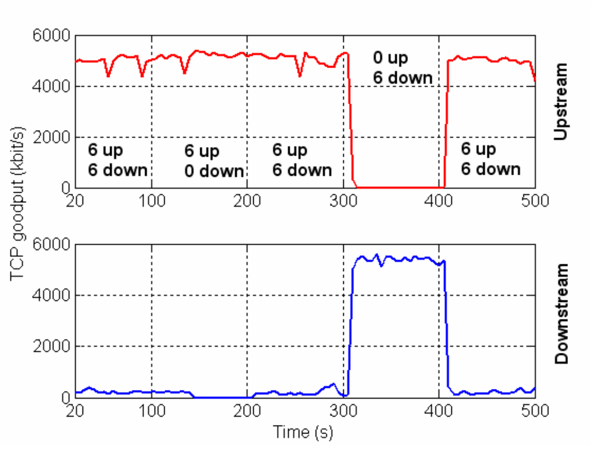

Fig. 7 - Instantaneous upstream, downstream and total TCP goodput (dynamic traffic, without rate-limiter)

3.2. Test-bed results with rate-limiter

In these tests we assume that: i) the adaptive rate control refresh period T_p is equal to 1 s; ii) the rate step R_{step} is equal to 200 kbit/s; iii) the bucket size B_{bucket} is equal to 250 kbytes. The latter value has been chosen by means of test trials, maximizing the fairness for the numbers of STAs considered in this paper. However, as noted above, different choices in the range of values that we examined (50-500 kbytes) would have had a limited impact on performance.

Fig. 8 reports the average upstream, downstream and total TCP goodput as a function of the number N of STAs with a static traffic model. The total goodput is very similar to that plotted in Fig. 4, i.e., the one obtained without rate-limiter. Thus, the IP rate-limiter does not waste radio resources. Moreover, the upstream and downstream goodputs are almost equal to each other, confirming the effectiveness of our approach.

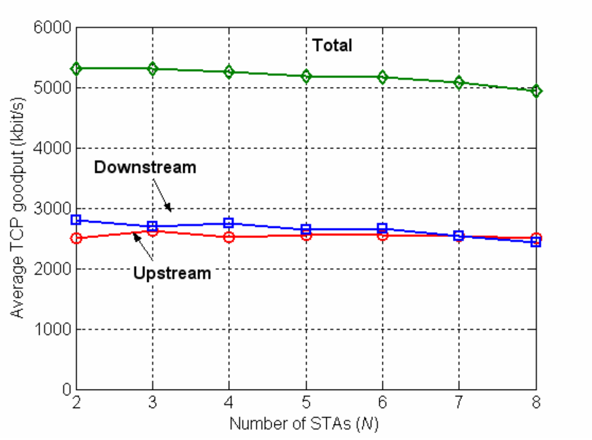

Fig. 8 - Average upstream, downstream and total TCP goodput as a function of *N* (static traffic, with rate-limiter)

Fig. 9 shows the average upstream and downstream TCP goodput for single STAs with *N*=6 and a static traffic model. It is interesting to note that we obtain also an intra-upstream and intra-downstream fairness.

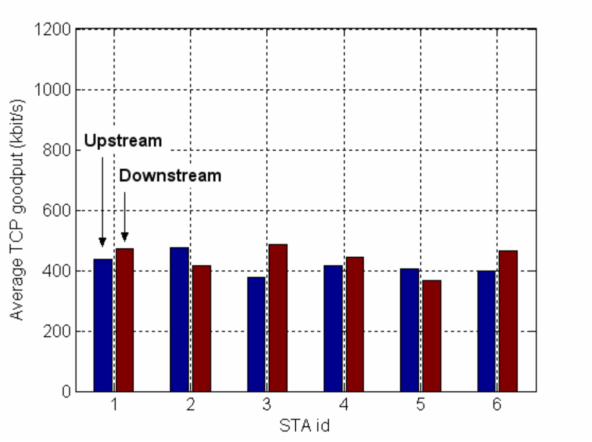

Fig. 9 - Average upstream and downstream TCP goodput for single STA (*N*=6, static traffic, with rate-limiter)

Fig. 10 reports the instantaneous upstream, downstream and total TCP goodput with *N*=6 and a static traffic model. We note that upstream, downstream goodput are almost equal to each other not only in average, as shown in Fig. 8, but also in the short term.

Finally, Fig. 11 shows the instantaneous upstream, downstream and total TCP goodput with a dynamic traffic model. We can see that when both the upstream and downstream connections are active, the radio resource are fairly shared, since the upstream and downstream goodput are almost equal to each other. Moreover, when the upstream or the downstream connections are switched off, the remaining active connections use all the radio bandwidth, without wasting radio resource.

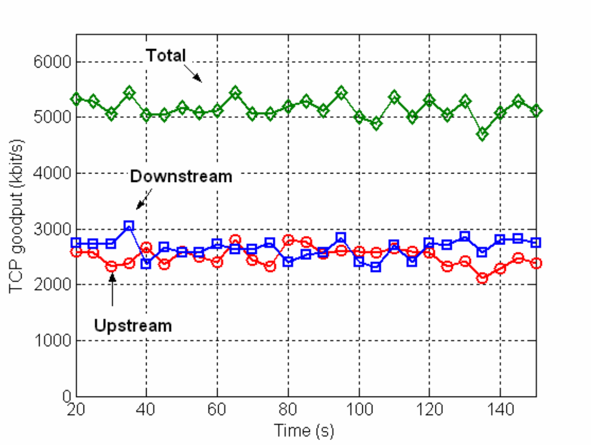

Fig. 10 - Instantaneous upstream, downstream and total TCP goodput (*N*=6, static traffic, with rate-limiter)

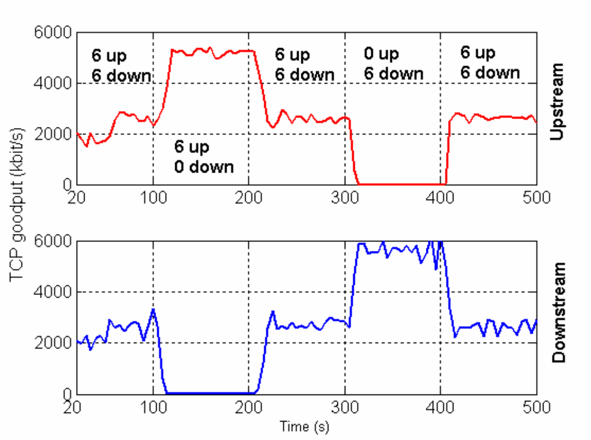

Fig. 11 - Instantaneous upstream, downstream and total TCP goodput (dynamic traffic, without rate-limiter)

4. Conclusions

In this paper, we addressed some TCP fairness issues in a wireless access network based on the IEEE 802.11 standard.

We proposed a solution based on an IP rate-limiter operating on the uplink traffic, that can be implemented within or outside the access point. The rate of the rate-limiter is dynamically selected as a function of the network traffic conditions.

The test-bed results show that our proposed mechanism provide a fair TCP access to radio resources and avoids critical starvation in all considered network and traffic scenarios.

Coming to possible extensions of this work, a straightforward one is the support of a mix of TCP and UDP traffic (supporting real time services). We are addressing this issue by taking into account the capacity consumed by UDP flows in the adaptive rate control setting, either on a pre-reservation basis or on the basis of a dynamic estimation. The solution will also consider a separate queuing of UDP and TCP packets in the access point.

References

[1] S. Pilosof, R. Ramjee, Y. Shavitt, P. Sinha, “Understanding TCP fairness over Wireless LAN”, IEEE INFOCOM 2003, March 30-April 3, 2003, San Francisco, USA.

[2] Blefari-Melazzi, N.; Detti, A.; Ordine, A.; Salsano, S., "Controlling TCP Fairness in WLAN access networks using a Rate Limiter approach," IEEE 2nd International Symposium on Wireless Communication Systems, pp. 375- 379, 5-7 Sept. 2005.

[3] S. Shenker, J. Wroclawski, “Network Element Service Specification Template”, RFC 2216, September 1997.

[4] 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”, University of Roma Tor Vergata, technical report. Available in http://netgroup.uniroma2.it/docs/TR-tcp-fairness.pdf.

[5] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, “The Effects of Asymmetry on TCP Performance”, ACM Mobile Networks and Applications (MONET) Journal, Vol. 4, No. 3, 1999.

[6] http://netgroup.uniroma2.it/download/wlan_arc.tar.gz

[7] Liang, F.; Wang, X.; Xu, L.; Fan, J., "On AP buffer effect upon TCP fairness over WLAN," IEEE International Conference on Mobile Technology, Applications and Systems, vol., no.pp. 4 pp.-, 15-17 Nov. 2005.

[8] Wang, X.; Liang, F., "TFBR: A New Queue Management Guaranteeing TCP Fairness Based on TCP Advertised Receiving Windows Over WLAN," IEEE 2nd International Conference on Mobile Technology, Applications and Systems vol., no.pp. 1-7, 15-17 Nov. 2005.

[9] Yi Wu; Zhisheng Niu; Janfeng Zhu, "Upstream/downstream unfairness issue of TCP over wireless LANs with per-flow queueing," IEEE International Conference on Communications, vol.5, no.pp. 3543- 3547 Vol. 5, 16-20 May 2005.

[10] Xiaoyang Lin; Xiaolin Chang; Muppala, J.K., "VQ-RED: An efficient virtual queue management approach to improve fairness in infrastructure WLAN," The IEEE Conference on Local Computer Networks, vol., no.pp. 7 pp.-, 15-17 Nov. 2005.

[11] KIM et al., “Distributed Access Time Control for Per-Station Fairness in Infrastructure WLANs”, KIM et al. IEICE Trans Commun.2006; E89-B: 2572-2579.

[12] Padhye, J.; Firoiu, V.; Towsley, D.F.; Kurose, J.F., "Modeling TCP Reno performance: a simple model and its empirical validation," IEEE/ACM Transactions on Networking, vol.8, no.2pp.133-145, Apr 2000.