A comparison of HTB based Channel-Aware Schedulers for 802.11 systems

R. G. Garroppo, S. Giordano, S. Lucetti and G. Risi
Department of Information Engineering, University of Pisa
Address: Via Diotisalvi 2, 56126 Pisa, Italy
Email: {r.garroppo, s.giordano, s.lucetti, g.risi}@iet.unipi.it

May 4, 2005

Abstract

The 802.11 commercial devices adopt a simple FIFO scheduler, which does not allow to isolate flows addressed to different destinations. This characteristic leads to the well-known performance anomaly of 802.11: when one or more STAs experiment poor radio channel conditions, they increase the time spent to transmit a single packet leading to an inefficient use of the shared medium. These phenomena have as a consequence the worsening of the performance of all the STAs sharing the wireless medium independently of their radio channel conditions. In this scenario, the paper proposes an architecture for a scheduling algorithm to implement in the AP for the downlink traffic, based on Hierarchical Token Bucket (HTB). The peculiarity of the proposed architecture is its ability to take into account, besides the transport service class required by the destination user, the channel quality experimented by the destination mobile STAtion (STA). Starting from this architecture two scheduling schemes are then proposed: the Wireless HTB (WHTB) and the Time-based WHTB (TWHTB). Both schemes are implemented in a prototype of AP. Hence, the performance of the proposed scheduling algorithms are experimentally evaluated and compared with those obtained with standard scheduling algorithm, which do not take into account information on channel quality. Finally, the paper presents a comparison in terms of implementation issues highlighted by the two proposed schemes, during the developing phase and the experimental analysis.

1 Introduction

IEEE 802.11 [1] networks offer simplicity of operation and configuration in a number of operative scenario. However, the MAC design, associated to the intrinsic unreliability of the wireless channel, produces severe degradation of the throughput achievable by all the stations in a single Basic Service Set (BSS) if just one of them is characterized by unfavourable radio propagation conditions. Indeed, automatic data rate adaptation and multiple retransmissions cause such station(s) to occupy the channel for longer time, reducing the radio resources left available to the other stations. In particular, [2] has highlighted as, even in the case where all but one stations use the highest nominal data rate, the effective throughput of all the stations in a BSS is degraded below the lowest adopted data rate. The main cause of such behavior is the long term fairness of 802.11 MAC, based on CSMA/CA protocol, which guarantees an equal long term channel access probability to all stations.

Additionally, when working in infrastructured mode, the Access Point (AP) has to handle the traffic directed to all the associated stations. However, an AP contends for transmission opportunities with the same priority (CSMA/CA protocol parameters) of the other stations. Therefore, it is clear that the AP has the major bottleneck role of the system [3] [4]. Commercial APs manage the traffic towards the associated STAtions (STAs) according to a First-In-First-Out (FIFO) queueing discipline. As a consequence, when considering downlink direction, mutual throughput degradation occurs also within the same device, worsening the overall system performance.

The main contribution of the paper is the definition of an architecture based on the Hiearchical Token Bucket (HTB) able to differentiate the transport service offered by an IEEE 802.11 AP taking into account the radio link quality experimented by the different mobile stations. Using this architecture, two different scheduling schemes, differing in the procedure for the estimation of the goodput towards a destination STA, are presented: the Wireless HTB (WHTB) and the Time-based WHTB (TWHTB). The proposed schemes differ on the procedure used to estimated the achievable goodput towards a particular STA. In particular, in the WHTB, the estimation procedure is based on the measurements of the SNR at the AP, and in the translation of this value in an estimated goodput. The translation is done by means of a curve SNR vs. goodput obtained in a calibration phase. In the TWHTB, the goodput is estimated dividing the frame size by the measured time needed for the successful transmission of the frame. A detailed analysis of the WHTB is presented in [5]. A review of channel-aware schedulers proposed in literature can be found in [6] and reference therein. In this work, different Channel State Dependent (CSD) schedulers are discussed. The CSD schedulers differ from the proposed algorithms since CSD require the modeling of the channel, e.g. Gilbert-Elliot model. In the proposed WHTB and TWHTB, the scheduling process is made on-line using parameters that the AP is able to measure while operating. In this class of schedulers belong the scheme presented in [7]. However, this scheme does not take into account the downlink transport service differentiation issue.

2 Architecture of a Channel-Aware Scheduler

The aim of a channel-aware scheduler is to exploit information on the radio link quality experienced by the different users associated to the same AP in order to optimize the global goodput of the considered WLAN cell. The goodput is defined as the throughput perceived by the IP level; it takes into account the reduced efficiency due to MAC and PHY protocols overhead, data link layer retransmissions, current data rate, and so on. However, the actions taken to optimize the global goodput must consider also the required quality of service at IP level chosen by the users. To take into account both aspects, a possible architecture can be obtained modifying a scheduler used in wired networks in order to take advantage of the information on the radio link quality experienced by the users.

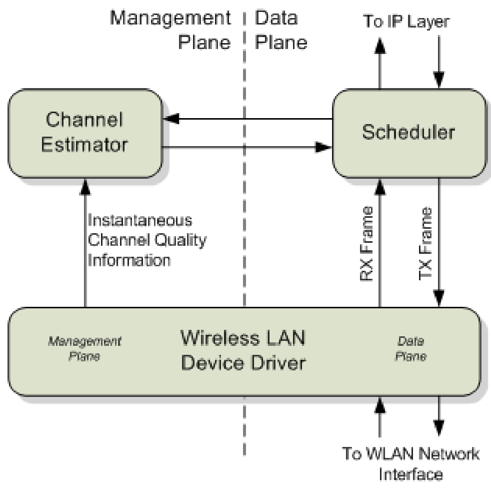

Figure 1: Architecture of a channel-aware scheduler

The reference channel-aware scheduler architecture is represented in Fig. 1. On the right hand side, the Data Plane is devoted to the transmission and reception of the frames; it is composed of the WLAN device driver, which performs the actual transmission and reception, and the scheduler, which is driven by link quality information provided by the Management Plane. The Management Plane is composed of the part of the WLAN device driver deputed to export the information on the channel quality towards each receiver station, and of Channel Estimator module, which translates this information into values suitable to feed the scheduler. As far as the scheduler itself is concerned, it can be either an algorithm designed specifically for wireless network or a scheduler developed for wired networks opportunely modified to take into account the information on wireless channel.

3 Introducing channel-awareness in the HTB

This Section briefly describes the main features of the HTB scheduler, and the extensions needed to introduce the support for channel-awareness in the scheduling process; for an extensive description of the HTB inner operations, the reader can refer to [8].

In short, HTB traffic classes are organized in a tree structure; each class is configured with an average rate to be guaranteed (*rate* parameter) and a maximum rate which cannot be exceeded (*ceil* parameter). To fulfill such requirements, each class is controlled by two internal token buckets, after which the scheduler is named. The HTB scheduler grants the right to transmit to classes which have not exceeded their allowed *ceil*, according to a Deficit Round Robin (DRR) algorithm. Classes which have not exceeded their *rate* can unconditionally transmit; classes which have exceeded their allowed *rate* but not their *ceil* can transmit only borrowing unused bandwidth, if available, from other classes.

The basic concept behind the introduction of channel-awareness in the HTB scheduler is to utilize an estimate of the actual effective goodput achievable towards each station to opportunely modify the scheduling algorithm.

Basically, once the goodput is available, the evaluation of the radio resources occupied by each transmitted packet (of length Pkt_{length}) is performed taking into account an *effective* packet length, called *stretched*, which is evaluated according to the following expression:

$$stretched = \min(Pkt_{length} \cdot \frac{R_{MAX}}{\hat{R}(t_0)}, bound)$$

where $\hat{R}(t_0)$ is the estimated average goodput towards the specific destination at the time of packet's dequeue, and R_{MAX} is the maximum attainable goodput towards the same destination when channel quality is maximum. The *stretched* value is upper bounded by the value *bound* to reduce starvation issues when very poor channel conditions are experimented. Based on these observations two different schedulers have been developed and tested: the Wireless Hierarchical Token Bucket (WHTB) and Time-based Wireless HTB (TWHTB).

4 The Wireless Hierarchical Token Bucket (WHTB)

In this architecture, a key role is played by the module that relates the channel quality towards a particular STA to an estimated capacity of the link towards such destination. This module is called the Wireless Channel Monitor (WChMon), and feeds the WHTB scheduler with indications about the effective exploitation of the nominal capacity towards each associated STA.

The WHTB then calculates the *stretched* packet length dividing the actual packet length by a corrective parameter, indicated as the *relative throughput*, being defined as the ratio between the estimated effective throughput and the maximum (nominal) one.

4.1 The Wireless Channel Monitor

The Wireless Channel Monitor (WChMon) used in the WHTB has been developed at the Washington University in St. Louis and is described in [9].

In most currently available wireless chipsets an Automatic Gain Control (AGC) unit monitors the signal condition and adapts the RF circuit of the card. This information is also used to compute three values indicating the signal level, the noise level and the signal quality (other designs only provide one value which indicates the signal level). These values are stored in the Parameter Storage Area (PSA) of the wireless card and can be read by a device driver which export them in the kernel memory. The reported values are related to the signal and noise levels in dB (allowing to estimate a SNR indicator), and are computed for each received packet.

The driver, after computing the SNR associated to the received frame transmitted by a STA, transfers this information to the WChMon which, by means of a table, estimates the goodput the AP expects to reach towards that mobile station. Each time the scheduler needs to be informed about the channel available capacity towards a certain destination, it invokes the WChMon specifying the STA address. The table which relates the SNR to the available goodput towards a specific STA is obtained by means of an initial *calibration* process of WChMon. The *WChMon Signal-to-Goodput Mapper* (*WChMonSigMap*) [9] is used for calibration: it is a Java application for Linux whose task is to build a table on the basis of experimental data.

4.2 The developed WHTB system

The WHTB has been realized integrating the WChMon in the architecture in Fig. 1, and modifying the HTB accordingly. The HTB code modification can be synthesized into two fundamental operations:

1. Each packet's length is converted into a stretched length:

$$stretched = \frac{length}{\hat{g}(t_0)} = \frac{length}{\frac{\hat{R}(t_0)}{R_{MAX}}} = length \cdot \frac{R_{MAX}}{\hat{R}(t_0)}$$

   where $\hat{R}(t_0)$ is the goodput, estimated by the WChMon at time instant t_0 of the scheduler choice of packet dequeue, towards the proper destination, while R_{MAX} is the attainable goodput towards the same destination when channel quality is maximum.

2. The value *stretched* is upper bounded, to prevent that very low values of relative goodput produce an artificial length so large that *deficit* goes under zero (which is a nonsense). The upper bound to *stretched* is set to the *quantum*.

Summarizing, the two operations can be expressed by means of the single following expression:

$$stretched = \min(length \cdot \frac{R_{MAX}}{\hat{R}(t_0)}, quantum)$$

5 The Time-based Wireless HTB (TWHTB)

The TWHTB differs from the WHTB in the channel estimation procedure. Indeed, differently from the approach which led to the WHTB scheduler (based on SNR measurements), an analysis based on time needed to successfully transmit a frame (including retransmissions and reception of the ACK message) has been pursued.

Fig. 2 shows the various components of the time T_{SUCC} needed to complete a successful frame transmission. Note that, although the *DIFS* and *Backoff* periods

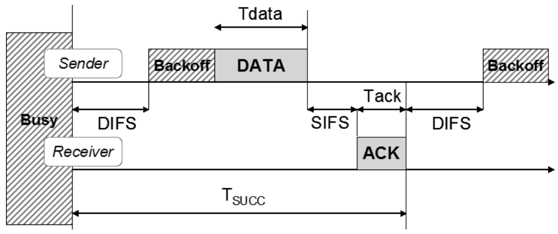

Figure 2: Time budget of the transmission of a data frame

are not strictly related to frame transmission (as far as the transmitter is concerned, they are idle periods), they have been included in the T_{SUCC} evaluation, since they limit the maximum frame rate, and have to be respected independently of the system conditions. Several works (see [10][11]) have used such time analysis to derive the maximum achievable goodput in saturation conditions of a single UDP source. The estimated effective goodput is obtained by the straightforward relation:

$$\hat{G} = \frac{Pkt_{length}}{T_{SUCC}}$$

optionally subtracting the $Pkt_{headers}$ size of the overhead bits which constitute IP and higher layers protocol headers if application level goodput is desired.

With similar considerations, we extend the analysis to the case of frame transmission requiring one or more retransmissions to define the Cumulative Frame Transmission Time (CFTT). The CFTT is defined as the amount of time needed to successfully transmit a frame (including retransmissions and possible rate reduction), and is measured from the instant of the transmission of the first bit of the frame, to the reception of the positive ACK which confirms its reception.

As a reference result, it is possible to analitically evaluate the CFTT, in saturation conditions, as a function of the number of frame transmission attempts before reception of the corresponding ACK message. Neglecting the propagation delays, a single transmission attempt duration can be splitted in several components [10], easily identifyable in Fig. 2. In the case of unsuccessful transmission of the frame, the sender recognizes such event if it does not receive back the ACK message within a defined $T_{ACK\_Timeout}$. We indicate with $T_{FAIL}(k)$ the amount

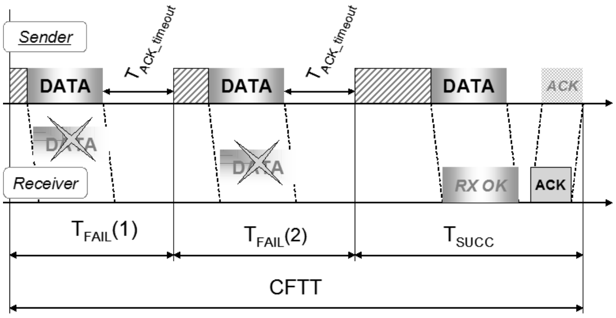

Figure 3: Example of CFTT in the case of 2 retransmissions

# of transmission attempts			
1	2	3	4
min-**avg**-max CFTT @ 11 Mbps			
1.29-**1.60**-1.91	2.58-**3.52**-4.46	3.88-**6.09**-8.30	5.17-**9.93**-14.7
min-**avg**-max CFTT @ 5.5 Mbps			
2.08-**2.39**-2.70	4.17-**5.11**-6.05	6.25-**8.46**-10.7	8.33-**13.1**-17.9
min-**avg**-max CFTT @ 2 Mbps			
4.85-**5.16**-5.47	9.70-**10.6**-11.7	14.6-**16.8**-19.0	19.4-**24.2**-28.9
min-**avg**-max CFTT @ 1 Mbps			
9.26-**9.57**-9.88	18.5-**19.5**-20.4	27.8-**30.0**-32.2	37.0-**41.8**-46.6

Table 1: CFTT vs Number of Transmission Attempts before successful reception of ACK frame (milliseconds)

of time associated to the k-th failed transmission attempt. The index k takes into account the different values of the $T_{Backoff}$ for each retransmission, due to the exponential Backoff update algorithm of 802.11; furthermore, since $T_{Backoff}$ is a random variable, these durations are random variables too.

Basing on these considerations, and as illustrated by Fig. 3 the overall CFTT can be expressed as $CFTT(N) = \sum_{k=0}^{N-1} T_{FAIL}(k) + T_{SUCC}$, where N is the number of transmission attempts needed to successfully transmit the frame. Table 1 collects the minimum, mean and maximum values of the CFTT (in the cases of 1 to 4 transmission attempts), for the transmission of a 1024 Byte UDP packet using the four nominal data rates available in 802.11b. The issue of rate adaptation between consecutive attempts is not covered in the present tractation for sake of simplicity. However, Table 1 gives an indication on the expectable range of values of CFTT which can be measured in actual networks.

5.1 TWChMon, a channel estimator based on CFTT

As indicated above, the knowledge of the CFTT for a given frame allows to estimate the achievable goodput towards the corresponding destination. We have integrated a real-time CFTT measurement procedure in the architecture depicted in Fig. 1; the resulting goodput estimates are used to feed the HTB scheduler to compute the *stretched* values of the Pkt_{length}. With respect to WHTB, the only difference in architecture is the channel estimation module. In the present approach, with reference to Figure 1, it is indicated with Time-based Wireless Channel Monitor (TWChMon)and operates as detailed below.

For each transmitted frame, both the instant of transmission begin and the instant of the reception of the corresponding ACK frame are detected. The difference between these two time values represents the frame CFTT. On the basis of such value, the TWChMon evaluates the estimated goodput (useful for the scheduling procedure) as

$$\hat{G} = \frac{Pkt_{length}}{CFTT}$$

To smooth the quickly varying channel dynamics (and to average the random values assumed by T_{BO}), a first order IIR low pass filter is applied to the result before letting it available for the modified HTB:

$$\hat{G}_{AVGD}(k) = 0.75 \cdot \hat{G}_{AVGD}(k-1) + 0.25 \cdot \hat{G}$$

As far as implementation issues are concerned, in the first phase of development of a channel-aware scheduler prototype we have used two network devices at the AP. One of them behaves normally, and is used to perform current transmission and reception of the frames, whereas the other is put in *monitor* mode. The *monitor* mode is a special operating state allowed by some wireless devices by means of proper kernel support [12], which allows the user to access (in read mode) all the frames on-air received by the interface; these include Management Frames (such as Beacons) as well as Control Frames (which include RTS, CTS and ACK frames). In order to be able to export ACK reception events to the TWChMon, the HostAP [13] driver used in the prototype has been modified accordingly.

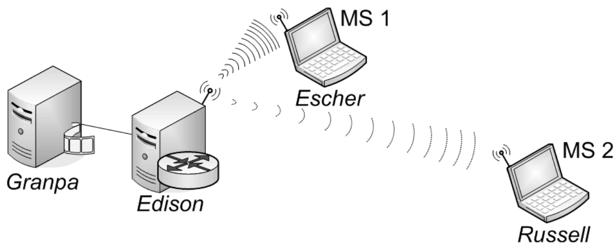

Figure 4: Experimental testbed

6 Parameters of experimental analysis

The experimental testbed is depicted in Fig. 4; all the devices are PC based linux boxes (2.4 Linux kernel). Both the MSs and the PC, which acts as AP and channel-aware scheduler (Edison), run a modified version of the kernel that includes the HTB patches for the Linux Traffic Control [14]. The HTB, and its modification (WHTB e TWHTB), have then been used to define some bandwidth shares between the two flows. In particular, 1 to 4 and 1 to 1 bandwidth shares have been adopted as test cases in the measurement campaigns. In order to create two data flows towards the two MSs, the BRUTE [15] traffic generation tool has been activated on Granpa. In this set of experiments, BRUTE has been configured to produce two CBR flows, characterized by constant inter-departure times and packet size. Most of the experiments that have been carried out have been performed slowly moving away one of the MSs (MS2) from tha AP, and jointly measuring the channel quality perceived and the effective goodput received by each MS.

The parameters of the traffic generation process at Granpa are as follows; the generated flows are CBR, with costant both inter-departure times and packet size. In detail, inter-departure time has been set to obtain 750 packets per second, whereas the packet size has been set to 1024 byte. The resulting rate is about 6 Mbps for each flow; hence, the overall rate offered to Edison is about 12 Mbps, well above the about 5 Mbps which can be handled in saturation conditions for this packet size, as highlighted by both analytical, simulative and experimental analysis in [10].

The duration of each measurement run has been set to 81.92 seconds. This unusual value (not a round number) has been chosen in order to simplify the tasks, at the end of each measurement session, of the post processing procedure necessary to evaluate the achieved goodput. Indeed, in such period of time, the number of packets received by each MS is numerically equal to ten times the average goodput expressed in Kbps. The task to evaluate the average goodput is then converted in simply counting the received packets and multiplying such value by one hundred.

In addition to the mean goodput received by each MS, another important performance parameter has been considered in our measurement. Such parameter is the percentage of the radio resources occupied in the data transmission attempts towards each STA. The evaluation of such percentage has been performed according to the following considerations. Since MS1 always experiments good channel quality, it is possible to suppose that each transmission attempt towards MS1 is successful. The percentage of the channel occupancy for the transmissions addressed to MS1 is then equal to the ratio between the mean goodput received by MS1 and the saturation goodput of the system in ideal channel conditions (both MS1 and MS2 with good channel quality). On the other hand, the difference between the saturation goodput and the goodput received by MS1 represents the transmission capacity used to deliver packets to MS2. Note that this value is typically different from the achieved gooput because of retransmissions which occur as soon as MS2 moves away from the AP.

The value of the percentage of the effective utilized resources indicates whether the scheduler is efficient in avoiding MS2 to occupy the medium with many unsuccessful retransmission attempts, at the expenses of MS1. Further interpretations of the resource occupancy percentage will be given in the next Section, where some measurement results will be presented and analyzed.

MS2 Position	MS	CBQ	HTB	DRR	WHTB	TWHTB
Good	1	2545	2534	2546	2511	2562
	2	2545	2512	2543	2514	2552
Medium	1	2000	1925	2038	2061	2125
	2	2000	1855	2037	2028	1636
Bad	1	1462	1492	1471	2107	2020
	2	1460	1153	1469	956	795
Very Bad	1	714	778	573	1825	1676
	2	676	369	562	313	350

Table 2: Performance comparison among the different schedulers: Mean Received Goodput in Kbps with 1:1 bandwidth share

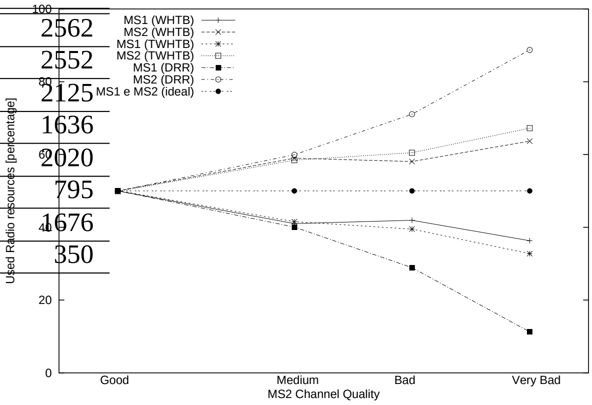

Figure 5: Comparison between the radio resources used to transmit to MS1 and MS2 with DRR, WHTB and TWHTB schedulers

7 Performance analysis

In order to evaluate the benefit produced by the adoption of the proposed WHTB and TWHTB schedulers, a number of experimental measurements have been carried out using the experimental testbed depicted in Fig. 4. In this Section, the results related to two different choices of the nominal bandwidth shares assigned to each MS are presented. The first presented case is constituted by an even assignment of the nominal transmission capacity. The second one refers to an uneven share of the resources between the two stations, which privileges MS1 in a 4 to 1 ratio.

Each set of measurements has been performed keeping MS1 close to AP with good channel quality, and moving MS2 away from the AP, stopping in four different positions, characterized by different values of channel quality. The four positions have been chosen on a heuristic basis, using the link quality indicator available with the device driver, ranging from Good (on which no retransmissions occur) to intermediate states Medium, Bad and Very Bad. The same testbed has been configured using classical schedulers for wired networks, in order to compare the WHTB and TWHTB performance and verify the effectiveness of the approach. The reference schedulers are the Class Based Queueing (CBQ) discipline [16], the DRR and the plain HTB.

7.1 Bandwidth share 1:1

Measurement results obtained according to the procedure described in the previous subsection are reported in Table 2 and Figure 5. Table 2 collects the effective received goodput by the two stations with the adoption of three standard scheduling algorithms (CBQ, HTB and DRR) as well as the proposed WHTB and TWHTB. The results highlight the different behavior of the five scheduling algorithms as far as the isolation of the two flows is concerned. In particular, the three classical schedulers fail to fulfill the bandwidth reservation associated to MS1 when MS2 starts to experiment bad channel conditions not managing to deliver the requested 2.5 Mbps to MS1 in any configuration except the ideal one (both MS1 and MS2 in the Good channel quality state). Indeed, the CBQ, DRR and HTB operate to try and maintain the imposed ratio between the two assigned bandwidth. Consequently, the goodput received by MS1 degrades as long as MS2 suffers an increasing number of losses and retransmissions. On the other hand, the WHTB (or TWHTB) reduces the influence of MS2 position on the goodput received by MS1, limiting the goodput reduction to about 35% versus the over 75% of the other schedulers. Slight differences can be observed between the WHTB and TWHTB behavior.

When adopting the WHTB (or TWHTB) scheduler, although the received goodput values are below the requested 2.5 Mbps, they are still much higher than those achieved by the other schedulers, and represent a considerable fraction (from 65 to over 80%) of the nominal assignment, highlighting the goodness of the proposed approaches.

Fig. 5 highlights the different consumption of transmission resources between the WHTB, TWHTB and DRR schedulers (behavior of HTB and CBQ is analogous to DRR one). The latter allows the transmission towards MS2 to almost monopolize (up to 90% share) the use of the radio resources; indeed, the AP uses only 10% of the available bandwidth to transmit packets to MS1. On the contrary, the WHTB and TWHTB schedulers allow to limit the unfairness between the resources used by each class, with a maximum unevenness of about 65% to 35%, regardless of the position of MS2 among the Medium, Bad or Very Bad conditions. Although it does not guarantee a complete decoupling between the two flows, the WHTB and TWHTB represent a noticeable enhancement with respect to the standard schedulers.

At the current stage of development, the performance of TWHTB are degraded, when some destinations are in Very Bad position, by the limited number of positive ACK messages that can be used to estimate the channel quality. Ongoing work is devoted to optimize interaction of the device driver with the wireless interface registers, aiming at an increased knowledge of transmission attempt outcome.

7.2 Bandwidth share 1:4

The results obtained considering this scenario as summarized in Table 3.

The analysis of the Table emphasizes that relevant differences among the different scheduling schemes can be observed when the MS2 experiments Bad or Very-Bad conditions. In these scenarios, the WHTB and TWHTB schedulers outperform the other three in all these working conditions. The CBQ and DRR schedulers behave almost the same, the MS2 position severely influencing the gooput towards MS1; also, the HTB performs only slightly better than these two. On the contrary, the WHTB keeps the goodput delivered to MS1 always over 3.2 Mbps, irrespective of the position of MS2. The sensible increase of the network utilization adopting WHTB is particularly relevant when considering the case of MS2 in the Very Bad. Indeed, the three reference schedulers severely degrade the aggregate goodput of the cell to values well below 1 Mbps, whereas WHTB keeps its value to about 3.4 Mbps, allowing MS1 to experiment transmission conditions only slightly worse than in the ideal case. The TWHTB performs slightly worse than WHTB; indeed, the TWHTB permits to achieve only about 3.0 Mbps.

MS2 Position	MS	CBQ	HTB	DRR	WHTB	TWHTB
Good	1	4037	4061	4054	3937	4011
	2	1017	1011	1025	991	1081
Medium	1	3404	3966	3276	3344	3579
	2	859	609	828	788	721
Bad	1	2621	3544	2161	3960	3669
	2	662	503	546	469	327
Very Bad	1	1996	2105	1449	3401	3047
	2	497	255	342	160	190

Table 3: Performance comparison among the different schedulers: Mean Received Goodput in Kbps with 1:4 bandwidth share

8 Comparison WHTB vs. TWHTB

In the performance analysis presented in the previous section, the WHTB has shows to outperform the TWHTB. However, the implementation work and the experimental analysis have highlighted some limits of WHTB, difficult to overcome. On the contrary, the implementation of TWHTB can be improved. As a consequence, this second proposed scheme seems to be the most interesting.

The main open issue of WHTB is channel estimation, which is not very accurate. In particular, it is based on the assumption that the channel is symmetric, i.e. the SNR seen by the AP is analogous to the SNR seen by the receiver located on the STA. Therefore the AP, which only knows the SNR associated to the received frames from a given STA, can estimate, with fairly good approximation, the SNR perceived at the STA, and consequently the expected goodput towards that STA.

Because of asymmetries in the wireless transmission channel, when packets are sent from the AP to a MS, calibration gives results that are slightly different from those obtained when packets are sent in the opposite direction.

Indeed, it should be taken into account that a successful transmission of a packet is made up of two phases: transmitting a frame and receiving the ACK. When the channel is not symmetric, it may happen that the data frame experiences a fairly good SNR, while the ACK has a higher probability of being corrupted due to a higher noise level at the frame transmitter, or viceversa.

To verify channel simmetry, and to mitigate possible differences, calibration in both directions was done and the table to be used for scheduling was obtained by a final extrapolation and smoothing.

During normal operations, WChMon applies a first order IIR low pass filter to the SNR values reported by the driver. Indeed, instant SNR reported by the driver manifests a high degree of variability, since it refers to a single received frame and is consequently affected by short-term signal power fluctuations.

Problems of WChMon that can be assumed also for the TWChMon are discussed in the following.

8.0.1 WChMon with medium channel quality

When one of the two mobile stations is located in a medium-quality area, WChMon overestimates quality and assigns to this station a good value in the throughput mapping; therefore, as for traditional schedulers, the average goodput of the MS in good position decreases as much as the one in medium position. This is mainly related to a WChMon calibration issue, due to the asymmetry of transmission channel. Alternatives methods for actual goodput estimation, based on reception of ACK frames are currently being investigated to reduce this problem.

8.0.2 WChMon with very bad channel quality

When one of the two mobile stations is located near the border of the AP coverage area, a problem affect the correct behavior of the wireless network. Some packets are lost because the number of retries exceeds *Retry_Limit*; WChMon is not designed to handle this case, and hence produces a less accurate goodput estimation in such conditions, worsening the overall performance.

Summarizing, the two proposed schemes show similar performance. From an implementation point of view, the WHTB presents a big problems: it is based on the assumption that the channel is symmetric. Furthermore, it requires the calibration of the WChMon before starts its operations. On the contrary, the TWHTB permits to overcome these two big problems. Hence, although it presents some drawbacks, its implementation can be improved.

9 Conclusions

To overcome the performance anomaly of the 802.11 system, evidenced firstly in [2], and to achieve a fixed bandwidth share between the users receiving downlink traffic, two scheduling algorithms have been designed and implemented in a Linux based AP prototype. Both schemes are based on the HTB scheduler. In particular, the HTB has been extended in order to produce scheduling algorithms able to consider both the information on radio channel condition of the different stations and the transport service class required by the destination user. The two proposed scheduling schemes differ from the procedure used to estimated. The performance of the proposed schedulers have been experimentally analyzed and compared with those obtained by means of classical scheduling algorithms, such as DRR, CBQ and HTB, defined for wired network scenario. The performance analysis has highlighted the efficiency of the proposed schemes, i.e. WHTB and TWHTB; the experimental results show that classical scheduling algorithms are unable to maintain the assigned bandwidth shares to the stations. On the contrary, the proposed schedulers permit not to penalize the STAs experimenting good channel condition and, as a consequence, their experimented goodput, independently of the channel conditions of other STAs.

Such result is due to the capacity of the proposed scheduling algorithms to limit the difference between the channel occupancy time of each STA with respect to the assigned bandwidth shares, independently of its channel condition. On the contrary, when classical schedulers are used, stations in bad channel conditions increase their channel occupancy time to the detriment of the STAs in good channel conditions.

The comparison in term of implementation issues has highlighted that the WHTB cannot be improved and is limited by the assumption on the asymmetry of the radio channel and by the calibration procedure of WChMon.

Acknowledgment

This work has been sponsored by the Italian COFIN research program TWELVE.

References

[1] http://grouper.ieee.org/groups/802/11/

[2] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, *Performance Anomaly of 802.11b*, Proc. of IEEE Infocom 2003, San Francisco, April 2003

[3] D. P. Hole, F. A. Tobagi, *Capacity of an IEEE 802.11b Wireless LAN supporting VoIP*, Proc. of IEEE ICC 2004, Paris, June 2004

[4] R. G. Garroppo, S. Giordano, S. Lucetti, *Admission Region for Multimedia Services in IEEE 802.11e Systems*, Proc. Networks 2004, vol. 1, pp. 411-416, Wien 2004

[5] R. G. Garroppo, S. Giordano, S. Lucetti, E. Valori, *The Wireless Hierarchical Token Bucket: a Channel Aware Scheduler for 802.11 Networks,* Tech. Report TR-2004-12-01.pdf, http://netgroup-serv.iet.unipi.it/reports/TR-2004-12-01.pdf

[6] R. Rom and Hwee Pink Tan, *Framework for Performance Analysis of Channel-aware Wireless Schedulers* Submitted to Journal of Communication Networks, October 2004.

[7] M. Portoles, Z. Zhong, S. Choi, *IEEE 802.11 Downlink Traffic Shaping Scheme For Multi-User Service Enhancement* Proc. IEEE PIMRC'03, Beijing, China, Sept. 7-10, 2003

[8] M. Devera, *Hierarchical token bucket theory*, May 2002, http://luxik.cdi.cz/~devik/qos/htb/manual/theory.htm

[9] Lars Wischhof, John W. Lockwood, *Packet Scheduling for Link-Sharing and Quality of Service Support in Wireless Local Area Networks*, Technical Report WUCS-01-35, Applied Research Laboratory, Washington University in St. Louis, November 2001

[10] R. G. Garroppo, S. Giordano, S. Lucetti, F. Russo, *IEEE 802.11b Performance Evaluation: Convergence of Theoretical, Simulation and Experimental Results*, Proc. of Networks 2004, vol. 1, pp. 405-410, Wien 2004

[11] G. Anastasi, E. Borgia, M. Conti, E. Gregori, *IEEE 802.11 ad hoc networks: performance measurements*, Proc. 23rd Int'l Conf. on Distributed Computing Systems Workshops, Providence, Rhode Island, USA 2003

[12] J. Tourrilhes, *Wireless LAN resources for Linux*, 1996-2003, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/

[13] J. Malinen, *Host AP driver for Intersil Prism2/2.5/3*, http://hostap.epitest.fi

[14] M. Devera, *HTB Linux queuing discipline manual - user guide*, May 2002, http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm

[15] BRUTE: Brawny and Rough Udp Traffic Engine, http://netgroup-serv.iet.unipi.it/brute

[16] S. Floyd, V. Jacobson, *Link-sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, August 1995