

A Hierarchical Token Bucket Algorithm to Enhance QoS in IEEE 802.11: Proposal, Implementation and Evaluation

J.L. Valenzuela, A. Monleon, I. San Esteban, M. Portoles, O. Sallent
Dept. of Signal Theory and Communications - Polytechnic University of Catalonia (UPC)
Barcelona, Spain
valens@tsc.upc.es

Abstract—In this paper the performance of the IEEE 802.11 wireless local area networks in combination with Hierarchical Token Bucket traffic shaper is assessed. The analysis allows to show the basic advantages of the proposed scheduler. Results are obtained on a real IEEE 802.11b testbed to gain insight into HTB practical implementation issues. The HTB concept extended to a wireless scenario in this paper can also be applicable to IEEE 802.11e standard.

Keywords: WLAN, IEEE 802.11, QoS, throughput.

I. INTRODUCTION

Wireless LANs have experienced an impressive growth in the last years mainly by means of IEEE 802.11 family of standards, which has received the widest market acceptance. Besides, more complex services are envisaged to be supported in this competitive technology, such as multimedia communications, video on demand, VoIP, etc. create these components, incorporating the applicable criteria that follow.

Despite the important role of wireless LANs based in particular on 802.11b, it cannot be considered a well mature standard because of the performance, the lack of quality of service (QoS), security, etc. Nevertheless, the Task Group E within the IEEE 802.11 Working Group is developing a project (802.11e) whose purpose is to enhance the current 802.11 MAC to expand support for applications with Quality of Service (QoS) requirements [2]. In the meantime, and because of the plausible concerns about the time needed to stabilize and agree the standard in a first step and the time to market for 802.11e products in a second step, there have been proposals to provide QoS in 802.11b at MAC level, such as Distributed fair Scheduling [6], Blackburst [7], Aard's Differentiation Scheme [8], etc. These proposals modify parameters that define how a station (STA) access the wireless medium, i.e. they modify parameters of either the fundamental access method of the IEEE 802.11 MAC called Distributed Coordination Function (DCF), or the optional access method, which is the Point Coordination Function (PCF) [1]. In fact, some of these proposals are being included in the future Hybrid Coordination Function (HCF) in 802.11e [2].

The purpose of this paper is to provide appropriate quality of service mechanisms in WLAN with the novelty that, instead of focusing on MAC layer, the proposed solution is set at IP level with Hierarchical Token Bucket (HTB), which exercises

control over the transmissions, queuing and dequeuing packets in a determined and configurable way. HTB is a very complete and useful traffic shaper that has been successfully tested on wired environments [9] and [10]. This paper extends the use of this algorithm in a WLAN environment, and the proposed solution at IP layer could be incorporated into MAC layer as an enhancement to IEEE 802.11e.

One of the characteristics of current IEEE 802.11 products is the link adaptation, which consists on downgrade the bit rate transmission to a lower value when repeated unsuccessful frames transmissions are detected. This behaviour is shown to be very efficient for a standalone host. Nevertheless, link adaptation may seriously degrade the WLAN global performance, which penalizes fast hosts and privileges slow stations [3]. The HTB mechanism proposed in this paper can be a solution to the mentioned problem of stations transmitting at different rates. Furthermore, using Hierarchical Token Bucket less aggressive medium access behaviour can be achieved, with the corresponding positive influence on the throughput standard deviation.

Results for Hierarchical Token Bucket proposal in this paper have been obtained in a real WLAN testbed with IEEE 802.11b products. The testbed approach has been preferred in a first step in order to assess many practical issues regarding HTB in a real wireless environment. Since we found that HTB is a promising solution for future developments in combination with IEEE 802.11e enhancements, the next step would be to simulate this solution under IEEE 802.11e framework.

II. BASICS ON HIERARCHICAL TOKEN BUCKET

Hierarchical Token Bucket (HTB) is a class based queue discipline. A queue discipline (qdisc) can be seen as a black box which is able to queue and dequeue packets in order and at times determined by the algorithm hidden in it. It is located between IP layer and the layer 2 (MAC), as it is shown in figure 1.

HTB is based on hierarchical classes where three class types exist: root, inner and leaf. Root classes are suited on the top of the hierarchy and all traffic goes out through them. Inner classes have father and daughter classes. Finally, leaf classes are terminal classes, so they have father classes but not daughter classes. These three types of classes are described in

figure 2. In leaf classes, traffic from upper layers is injected following a classification which must be performed using filters, so it is possible to difference kinds of traffic and priorities, which should have different treatment. In this way, before traffic enters in a leaf class, it needs to be classified through filters with different rules, which can filter by kinds of services, IP addresses or even network addresses. This process is known as classifying process. Furthermore, when traffic has been classified, it is scheduled and shaped. In order to perform these tasks, HTB uses the concept of tokens and buckets to control the bandwidth use in a link. To adjust the throughput, HTB generates tokens at necessary cadence and dequeues packets from the bucket only if tokens are available. The main idea is shown in figure 3.

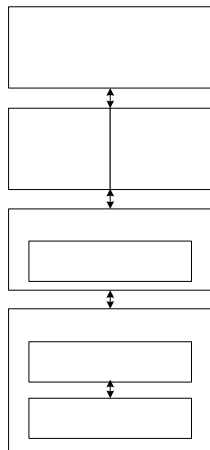


Figure 1. Layers detail.

The main advantage claimed for HTB traffic shaper is the bandwidth sharing. In this way, every class has associated an assured rate (AR), ceil rate (CR), actual rate (R), priority level (P) and Quantum (Q). This excess bandwidth is shared depending on the priorities that we have assigned to the classes. So, high priority classes can borrow more excess bandwidth than low priority classes. Thus, when R of one class has reached AR, it borrows tokens from its parent class. When this class has reached CR, it queues packets until new tokens/ctokens are available. This process is known as policing process. The complete behavior is summarized in figure 4.

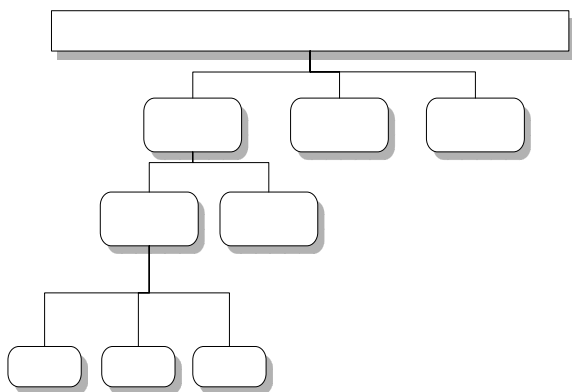


Figure 2. Hierarchical classes in HTB

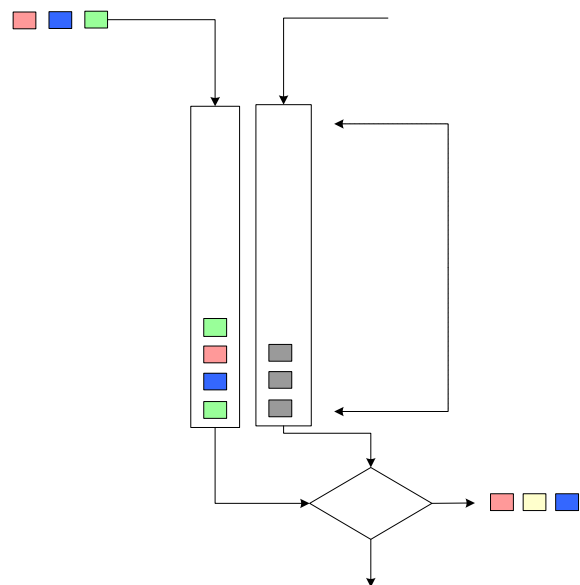


Figure 3. Token Bucket Filter performance.

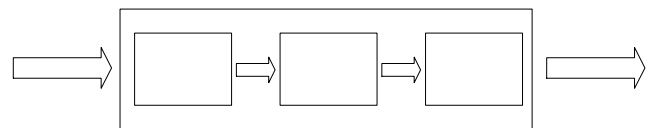


Figure 4. HTB summary behaviour

III. TESTBED IMPLEMENTATION

A common scenario as depicted in figure 5 have been considered. The server is connected to the Access Point via a 100 Mb/s switched Ethernet, so that the wireless link is the bottleneck. Furthermore, the application generated traffic for the test is only in the uplink, i.e. it goes from hosts to Access Point, because uplink traffic is more critical than downlink traffic. In addition, the wireless local network is configured in infrastructure mode, because of a relatively higher quality of service in this mode than in ad-hoc mode.

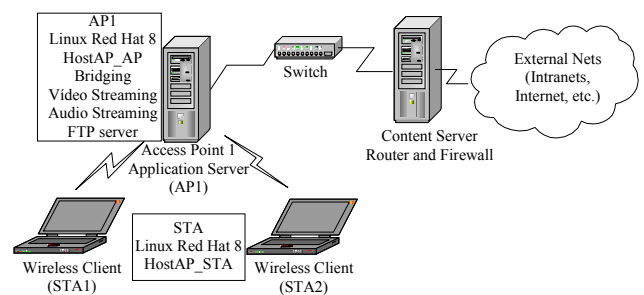


Figure 5. Scenario of tests: two hosts generate TCP and UDP traffic to a host behind the Access Point.

Current 802.11b products can degrade the bit rate from 11 Mbps to 5.5, 2 or 1 Mbps when a host detects repeated unsuccessful frame transmissions, in this scenario. To characterise these situation on the testbed, hosts have been

TCP

UDP

configured to send their packets at different fixed rates: 11Mbps, 11Mbps and 5.5Mbps, 2Mbps and 1Mbps respectively.

To generate the traffic we have used Iperf tool [12], which generates both TCP and UDP traffic, and ethereal tool [13], to capture all frames from client stations to content server and vice versa. HTB is included in Linux Kernel 2.4.20 and above.

IV. SAMPLE RESULTS

One very well known effect over IEEE 802.11 is that, in case of several host transmitting at different rates, the throughput of all hosts with higher rates are degraded below the level of the lower rate hosts as it is shown in figure 6. The testbed in this case has been configured with STA1 transmitting at 11Mbps, while STA2 transmits at 2Mbps. It can be observed that, instead of reaching the throughput of 5Mbps, it falls to a total throughput of 1.7Mbps approximately.

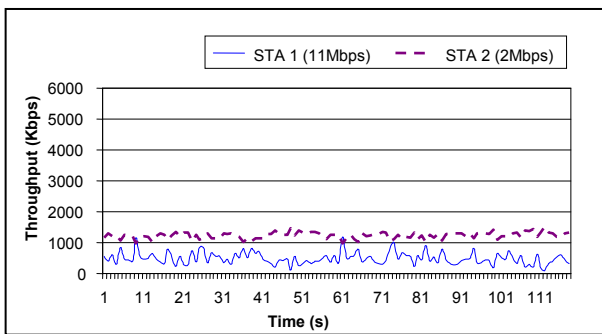


Figure 6. Two STA's transmitting at different rates.

The above result shows two basic problems: the first one is the average throughput of all hosts, imposed by the lowest bit rate host. The reason for this anomaly is the CSMA/CA channel access method of DCF, because it guarantees that the channel access probability is equal for all hosts. So, when a low rate host captures the channel, this station penalizes other hosts transmitting a longer time than fast hosts do [3]. For this reason, we need to find a solution that can deal traffic from different hosts independently, to guarantee a sustained throughput to all hosts. The second problem is the standard deviation, which reaches high values.

To cope with these situations, the traffic should be constrained by HTB in hosts in combination with the Access Point to obtain desired QoS effects. Performance using both UDP and TCP traffic are considered. In each experiment, the average throughput and standard deviation of both hosts are measured. The STA Bit rate column in the tables below shows the physical transmission rate, while HTB Limit column presents the assured rate configuration parameter of HTB algorithm. We group UDP and TCP results in terms of bit rate transmission due to the similar behaviour of the global system. The main difference between UDP and TCP tests is the total available throughput to share. In this way, TCP has less available bandwidth than UDP traffic, because TCP traffic needs ACK frames, and the access point competes for the channel too. As it is shown in [4], the proportion of useful

bandwidth depends on the number of hosts, and consequently, the useful bandwidth is lower for same conditions.

Following results are shown in two different parts, depending on the bit rate combination of both stations.

A. Two hosts transmitting at 11Mbps

The average throughput of each station in these tests is obtained. Then, HTB limits are configured with these values to analyse the new behaviour. As it can be seen, the average throughput results are very similar to the experiment without HTB, but the standard deviation has been significantly reduced. In fact, standard deviation is reduced from 156Kbps to 16Kbps for STA1 for UDP traffic and from 163Kbps to 13Kbps for STA2 for UDP. The standard deviation figures in Table I and Table II are complemented with figure 7 and figure 8, where it is shown the achieved throughput along time when HTB is not used and when HTB is included. The throughput stability obtained with HTB is clearly observed. Also notice that, in this case, where the two stations are transmitting at the same bit rate competing for the channel, throughput is shared in a linear way with 1:1 ratio and, consequently, STA1 and STA2 are obtaining the same throughput.

TABLE I.

HTB Status	STA Bit rate (Mbps)		HTB Limit (Kbps)		Average Throughput (Kbps)		Std Deviation σ (Kbps)	
	STA1	STA2	STA1	STA2	STA1	STA2	STA1	STA2
Without HTB	11	11	-	-	3018	3046	156	163
With HTB	11	11	3000	3000	3003	3000	16	13

Measured Throughput and Standard Deviation for 11 Mbps rates and UDP Traffic.

TABLE II.

HTB Status	STA Bit rate (Mbps)		HTB Limit (Kbps)		Average Throughput (Kbps)		Std Deviation σ (Kbps)	
	STA1	STA2	STA1	STA2	STA1	STA2	STA1	STA2
Without HTB	11	11	-	-	2546	2449	204	198
With HTB	11	11	2400	2400	2388	2397	50	35

Measured Throughput and Standard Deviation for 11 Mbps rates and TCP Traffic.

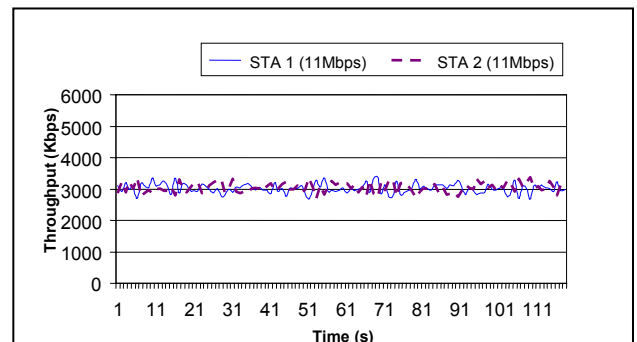


Figure 7. Throughput of two stations at 11Mbps without HTB.

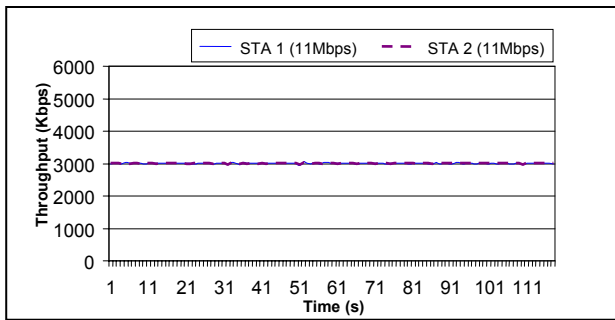


Figure 8. Throughput of two stations at 11Mbps with HTB.

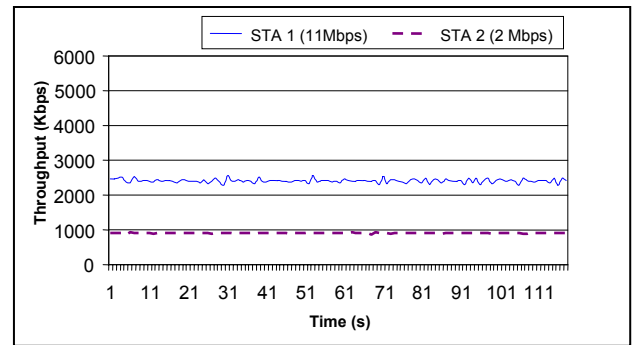


Figure 10. Throughput of two stations at 11 and 2Mbps with HTB

B. One host transmitting at 11Mbps and another at 2Mbps

In this test, different bit rate transmissions are considered. STA1 is 11 Mbps bit rate and STA2 is 2Mbps. When STA2 transmits UDP traffic at 2 Mbps, it uses five longer time than STA1. For this reason, if HTB constricts STA2, available bandwidth to share is almost four times larger the bandwidth HTB has constricted. For example, when HTB limit of STA2 is reduced from 1300 Kbps to 900 Kbps, STA1 HTB limit is upgraded from 900 Kbps to 2400Kbps, so a 50 % gain on available throughput follows. Additionally, the standard deviation reduces significantly (Figure 9 and figure 10). In TCP traffic the gain is almost 53 %.

TABLE III.

HTB Status	STA Bit rate (Mbps)		HTB Limit (Kbps)		Average Throughput (Kbps)		Std Deviation σ (Kbps)	
	STA1	STA2	STA1	STA2	STA1	STA2	STA1	STA2
Without HTB	11	2	-	-	965	1303	164	62
With HTB	11	2	900	1300	900	1301	42	27
	11	2	2400	900	2404	901	56	11

Measured Throughput and Standard Deviation for 11 and 2 Mbps rates and UDP Traffic.

TABLE IV.

HTB Status	STA Bit rate (Mbps)		HTB Limit (Kbps)		Average Throughput (Kbps)		Std Deviation σ (Kbps)	
	STA1	STA2	STA1	STA2	STA1	STA2	STA1	STA2
Without HTB	11	2	-	-	491	1233	201	103
With HTB	11	2	500	1200	477	1107	59	57
	11	2	1900	700	1873	719	56	19

Measured Throughput and Standard Deviation for 11 and 2 Mbps rates and TCP Traffic.

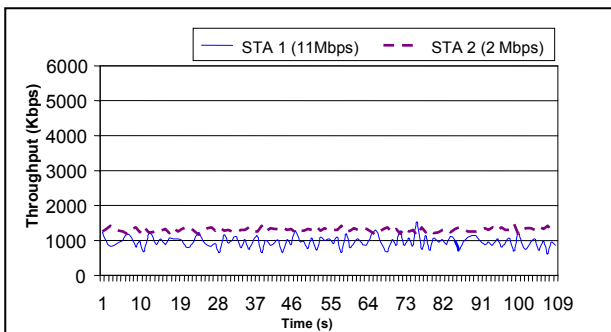


Figure 9. Throughput of two stations at 11 and 2Mbps without HTB

V. CONCLUSIONS

This paper has presented the performance of the IEEE 802.11b and it has described the well known problems of this technology, but specially the lack of quality of service. We propose the use of Hierarchical Token Bucket algorithm, located in the IP layer. This solution controls how the packets are delivered to MAC layer. This method is independent of the wireless standard used and it can be used jointly with any actual of future devices.

We can offer quality of service in wireless local area networks constraining stations transmitting at a lower bit rate. Applying this concept we can achieve a plain and sustained throughput with low standard deviation to stations or even to differentiated services.

REFERENCES

- [1] ANSI/IEEE, 802.11-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE.
- [2] IEEE 802.11 WG, "Draft Supplement to International Standard for Information Technology-Telecommunications and Information Exchange between Systems LAN/MAN Specific Requirements." IEEE 802.11e/D4.0, 2002
- [3] Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, Andrzej Duda, "Performance anomaly of 802.11b", INFOCOM 2003.
- [4] J. Antonio Garcia-Macias, Franck Rousseau, Gilles Berger-Sabbatel, Leyla Toumi, Andrzej Duda, "Quality of Service and Mobility for the Wireless Internet", Wireless Networks 9, 341-352, 2003.
- [5] Yasir Drabu, "A Survey of QoS Techniques in 802.11", 2002. Available in <http://www.cs.kent.edu/~ydrabu/>
- [6] Nitin H. Vaidya, Paramvir Bahl, Seema Gupta, "Distributed fair scheduling in a wireless LAN," Proceedings of the sixth annual international conference on Mobile computing and networking, p.167-178, August 06-11, 2000, Boston, Massachusetts, United States.
- [7] J.L. Sobrinho and A.S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," Bell Labs Technical Journal, pages 172-187, Autumn 1996.
- [8] I. Aad, C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," Proceedings. of IEEE INFOCOM 2001, vol. 1, pp. 209 -218, April 2001.
- [9] Martin Devera, "Hierarchical Token Bucket", <http://luxik.cdi.cz/~devik/qos/htb/>
- [10] Bert Hubert, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul B Schroeder, Jasper Spaans, Pedro Larroy, "Linux Advanced Routing & Traffic Control HOWTO", <http://lartc.org/howto/>
- [11] Martin A. Brown, "Traffic Control HOWTO", <http://www.linuxsecurity.com/docs/LDP/Traffic-Control HOWTO/>
- [12] Ajay Tirumala, Feng Qin, Jon Dugan, Jim Ferguson, Kevin Gibbs, "Iperf", <http://dast.nlanr.net/Projects/Iperf/>
- [13] Gerald Combs, "Ethereal", <http://www.ethereal.com/>