

Achievable Bandwidth Estimation for Stations in Multi-Rate IEEE 802.11 WLAN Cells

Eduard Garcia, David Viamonte, Rafael Vidal and Josep Paradells
Wireless Networks Group - Technical University of Catalonia (UPC)
{eduardg, viamonte, rvidal, teljpa}@entel.upc.edu

Abstract

This paper analyzes the effect of multi-rate transmissions in a CSMA wireless LAN environment. Observations in a real testbed showed that bandwidth resources (in Bytes/s) are shared fairly among all stations even though transmissions carried out at lower rates capture the medium for longer periods, which drastically reduces the overall throughput. The intrinsic concept of fairness in a CSMA scheme with multiple rates is quantified by means of a new formulation which is validated through simulations and practical measurements. The algorithm presented provides the maximum achievable bandwidth that can be offered to a given IEEE 802.11 station. Having this information has evident applications in real-time multimedia transmissions over WLANs. The algorithm was also run in commercial APs as a proof of concept, after analyzing its implementation issues.

The model presented in section 4 of this paper has been revised as a result of a posterior analysis. Please refer to (1) for an updated and extended version of section 4 only.

(1): <http://hdl.handle.net/2117/2045>

1. Introduction

The present popularity of WLANs, especially those based on IEEE 802.11 standards, requires special attention with the aim of explaining all observable phenomena which affect their efficiency and their provision of quality. IEEE 802.11 standards define several sets of modulations and coding rates for the different physical layers. Each different scheme provides a different transmission rate, but the higher the chosen rate, the worse it performs in presence of noise and interference. As the quality of the signal gets worse, the physical rate must be lowered in order to achieve an acceptable packet error ratio (PER). Furthermore, CSMA-type access is used: a station wishing to transmit first probes the medium and transmits only if the medium is sensed idle, ensuring long-term channel access fairness among all active stations. In a multi-rate environment, this concept of fairness involves a considerable loss of efficiency for the whole cell, since “slow” stations will capture the common medium for longer periods. In this way all stations obtain an equal bandwidth (expressed in Bytes/s). In consequence fairness is achieved at the cost of penalizing higher rate stations, which

leads to a loss of efficiency. In other words, on the one hand fair access in a multi-rate CSMA/CA network is translated into an equitable bandwidth allocation (in Bytes/s), but on the other hand it entails an unfair sharing of time, which reduces the throughput that fast stations can achieve. The problem was first described as a “performance anomaly” in [1]. These statements are verified in the example of fig. 1: connected to one IEEE 802.11b AP, two hosts (*A* and *B*) are in saturation state as they sent 1500-byte UDP datagrams. Different rates were used in *B* (represented on the x-axis in Fig. 1) for a fixed rate in *A*. No matter what rate they use, the individual throughput is almost the same for both stations. As we decreased the rate of one of the stations, the overall throughput decreased significantly.

In this paper we discuss the effect of a multi-rate environment in WLANs and provide a formulation that allows an accurate estimation of the achievable throughput, motivated by its application in multimedia transmissions over WLANs. Most of the related previous work in the field was focused on finding the maximum capacity for real-time flows based on saturation conditions and without considering background traffic [2]. In [3] a special interest was devoted to the effect of multi-rate stations.

We provide an analytical model that is validated through practical measurements and simulations. From this formulation we derived an algorithm that was developed to run in commercial IEEE 802.11 Access Points. The algorithm performs an estimation of the maximum bandwidth that the Access Point (AP) can offer to each of its associated stations. The results of the mechanism we present can also be used to compute a load metric based on time share, which describes the actual load of a cell more precisely than the number of active users or carried traffic. The use of our analytical method to provide the capacity that a network is able to offer to a given station avoids the undesired effects of existing invasive mechanisms (e.g. see [4]), which consist of injecting real data over the air interface, thus consuming valuable resources and possibly producing a negative impact on ongoing real-time transmissions [5][6].

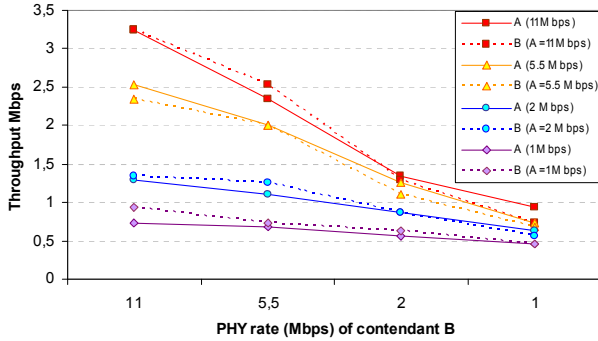


Fig. 1. Average throughput of stations A and B for different 802.11b physical rates

Our contribution is thus the design and implementation of a mechanism that accurately performs an estimation of the cell throughput and the achievable individual bandwidth for a set of stations transmitting within a WLAN cell, using potentially different modulations due to varying wireless environments.

Being able to perform accurate throughput estimations is of capital importance in several scenarios, and in particular in the case of real-time multimedia applications. This estimation can be used as a basis to perform a number of decisions, such as:

- Audio / Video CODEC selection and parameterization for multimedia sessions.
- Admission Control and Policing based on network conditions.

The rest of the paper is organized as follows: First, section 2 presents the motivation of the proposed mechanism through a discussion of its applications. Section 3 briefly introduces the basic concepts of IEEE 802.11 standards. The formulation and the algorithm used to compute the available bandwidth is detailed through Section 4. In Section 5, implementation details are discussed. Section 6 presents the evaluation results, and finally our concluding remarks are given in Section 7.

2. Applicability statements

2.1. Multimedia CODEC adaptation

A first scope where this kind of mechanism can be used relates to the ability to adapt audio/video CODEC settings to the actual network conditions. For example, a station under bad radio coverage conditions wishing to use a high quality video stream may actually affect performance of all users connected to the same AP (possibly with better radio conditions). An AP implementing the proposed algorithm would be able to periodically provide information about network conditions to all stations, so that they are able to perform the best decision at any given time (e.g.: CODEC

selection). In [7] authors propose an architecture where the CODEC of VoIP calls in a multi-rate WLAN is selected according to both MAC layer information and RTCP (Real-Time Control Protocol) statistics. In a similar way and also in the context of IEEE 802.11 WLANs, in [8] frame size and rate of an MPEG-2 stream are varied depending on detected channel conditions. Both solutions propose small incremental adaptations based on measurements carried out in the terminal. But, as we will explain in the next sections, the actual capacity for a given station also depends on the channel conditions seen from neighboring stations. This information would allow streams to be adapted rapidly with the optimal parameters.

Alternatively, the AP could provide network information to a network element operating at the application layer, so that it is able to control the CODEC settings of endpoints wishing to initiate multimedia sessions. This model is similar to the approach followed by 3GPP for IP Multimedia sessions based on the Session Initiation Protocol (SIP) [9]. SIP proxies may modify media stream offers so that they adapt to operator policies and capabilities. By connecting the information provided by the AP (running the proposed algorithm) to the policy functions implemented by the 3GPP IP Multimedia Subsystem (IMS) a decision can be made at any given time to determine the optimum set of encoding settings to be used, based on properly calculated throughput.

2.2. Admission Control adaptation

In addition to CODEC selection, the proposed algorithm can be used to enhance the Admission Control strategy implemented by the AP. Effectively, the AP may use the proposed algorithm to estimate the effect that a new user would cause in the overall cell throughput. Consider for example the case in which several multimedia sessions are running smoothly under a given AP, and suddenly a new node tries to connect using the lowest transmission rate. As we have explained above, the station with lower modulation severely affects the overall cell throughput. Different AP capacity estimations were used in [10] and [11] as a metric for admission control in WLANs.

Being able to perform proper Admission Control mechanisms is of interest in certain cases, and of crucial importance in others. Consider for example the case of a GSM user that roams into a WLAN cell and wishes to make use of the Generic Access Network (GAN) functionality adopted by 3GPP. This service makes use of the WLAN infrastructure to replicate the GSM user experience, thus alleviating operators from having to deploy costly indoor cellular coverage in complex environments. In this case, the GSM user expects the same quality of experience (QoE) from the GSM service, regardless of it being supported by a pure GSM infrastructure or from a GAN-WLAN AP. In such

case, the access network should support the following Admission Control capabilities:

- When a GSM user wishes to connect to its service provider using a dual GSM-GAN device over the WLAN infrastructure, the AP must be able to provide reasonable certainty that the GSM call will be successful. Otherwise, the user should be rejected in case it is still feasible to keep the call under pure cellular coverage (this is the case if WLAN and GSM coverage overlap).
- When a new data station wishes to attach to the AP, the network must be able to ensure that already ongoing GSM/GAN sessions will not experience any degradation due to the upcoming user. Otherwise, the new request may be rejected.

3. IEEE 802.11 protocols

Since the definition of the first IEEE 802.11 standard for WLANs [12], several variants have appeared that increase the bit rate. However, the MAC working procedure has remained the same. The IEEE 802.11 MAC procedure provides two operating modes: *Distributed Coordination Function (DCF)* and *Point Coordination Function (PCF)*. The *DCF* uses the contention MAC algorithm *CSMA/CA*, whereas the *PCF* offers contention free access. The two modes are used alternately in time.

The *DCF* works as follows: before initiating a transmission, a station senses the channel to determine whether it is busy. If the medium is sensed idle during a period of time called *Distributed Interframe Space (DIFS)*, the station is allowed to transmit. If the medium is sensed busy, the transmission is delayed until the channel is idle again. A slotted binary exponential backoff interval (*BO*) is uniformly chosen in $[0, CW-1]$, where *CW* is the contention window. The backoff timer is decreased as long as the channel is sensed idle, stopped when a transmission is in progress, and reactivated when the channel is sensed idle again for more than *DIFS*. When the backoff timer expires, the station starts transmitting. After each successfully received data frame, the receiver waits for a *Short Interframe Space (SIFS)* period and transmits an acknowledgment frame (*ACK*). The value of *CW* is set to its minimum value, CW_{min} , in the first transmission attempt, and increases integer powers of two at each retransmission, up to a pre-determined value CW_{max} .

The protocol described above is called *basic* or *two-way handshaking* mechanism. In addition, the specification also contains a *four-way* frame exchange protocol called *RTS/CTS* mechanism, which works as follows: a station gains channel access through the contention process described previously, and sends a special frame called *Request to Send (RTS)*, instead of the actual data frame. In response to that, the receiver sends a *Clear to Send (CTS)* frame after a *SIFS* interval. Subsequently, the requesting station is allowed to start the data frame transmission after a *SIFS* period. The

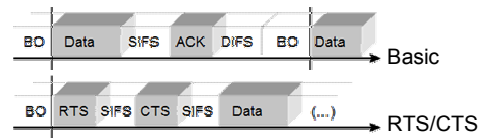


Fig. 2: Message sequence for basic and RTS/CTS access

main objective of *RTS/CTS* handshake is the resolution of the *hidden terminal* problem. The mechanism is also employed to minimize the lost periods caused by collisions – the *RTS* frame is much shorter than data frames. See fig. 2 for the complete message exchange sequence for basic access and *RTS/CTS*.

Finally, the IEEE 802.11 DCF MAC protocol supports two kinds of *Basic Service Set (BSS)*: the independent *BSS*, known as ad-hoc networks, which have no connection to wired networks, and the infrastructure *BSS*, which contains an AP connected to the wired network. The infrastructure *BSS* can be used to provide wireless access to IP networks and services in a similar way to that of cellular systems.

The *PCF* mode is only allowed in the infrastructure *BSS*. In this case, the AP polls its associated stations one after another by sending polling messages. Moreover, if the access point has data ready to send to a station being polled, it can be included in the polling message. If the polled station has data to send to the AP, it is transmitted in the response message after a *SIFS* period.

Moreover, as of late 2005, the IEEE 802.11 task group “E” released a new standard [13] that defines a set of Quality of Service enhancements for the IEEE WLANs; it defines procedures for managing network QoS using classes of service. The extensions introduced consider the two access mechanisms: *DCF* and *PCF*. The new MAC protocol is called *Hybrid Coordination Function (HCF)*. It combines a contention channel access mechanism, the *Enhanced Distributed Channel Access (EDCA)*, and a polling-based channel access, the *HCF Controlled Channel Access (HCCA)*. *EDCA* is designed to manage 8 different traffic priorities. Packets belonging to the different traffic priorities are mapped into 4 access categories (*ACs*); each of them represents a different priority level. The *HCCA* (which is not mandatory) works like the *PCF* with traffic classes. *HCCA* is the most complex coordination function. With *HCCA*, QoS can be parameterized with great precision. However, IEEE 802.11e enabled devices usually implement *EDCA* only, which is not intended to offer any form of QoS guarantee to the users in terms of bandwidth allocation, bounded delay, etc. Nonetheless, this service prioritization is translated into an increment of the probability to win the contention for the access to the common channel.

4. Estimation for a new station

Deriving the throughput that is currently devoted to a station with ongoing data transmissions can be achieved by means of straightforward measurements. On the other hand,

predicting the throughput that a new station will get before it actually starts transmitting, or the maximum bandwidth that can be allocated to a station if it increases its offered traffic in a multi-rate cell requires a deeper study. In this section we first analyze an ideal scenario in the simplest case: all stations are in saturation. Dispensing with the assumption of saturation, we then study a more general scenario where stations can have different bandwidth requirements. This additional complexity brings the need to develop an algorithm to derive the achievable throughput. Finally, the algorithm is adapted to the particularities of IEEE 802.11 WLANs.

4.1. The Saturation case

Let N be a set of n nodes attached to the base station of a cell. All stations from N share bandwidth resources by means of a MAC protocol which allows the assignment of a specific priority p_i to the i -th station. Every station transmits at rate r_i . We define the time period T_{cycle} as the average time between two consecutive transmissions of the station with lowest priority, taking into account the fact that there are frames waiting to be sent in all the stations' queues at any time (i.e. in saturation):

$$T_{cycle} = \sum_{j=1}^n p_j \frac{1}{r_j} \quad (1)$$

Considering that the packet size distribution function is the same for all stations after a long period, we used in (1) normalized length frames to simplify subsequent formulation.

Each node adds an individual load L_i defined as the portion of T_{cycle} that is used by node i :

$$L_i = \frac{p_i}{r_i \cdot T_{cycle}} \quad (2)$$

Thus, it is trivial to obtain the individual throughput S_i and the total throughput S as:

$$S_i = r_i L_i = \frac{p_i}{T_{cycle}} \quad (3)$$

$$S = \sum_{i=1}^n S_i = \frac{1}{T_{cycle}} \sum_{i=1}^n p_i \quad (4)$$

4.2. Non-Saturated Stations

In the previous sub-section we have studied how the bandwidth of a cell is shared among multi-rate stations, which are in saturation state. In a more general scenario, in which the stations may have different bandwidth requirements without reaching saturation, IEEE 802.11 MAC keeps assuring fairness in terms of access probability. All

stations contributing with a load equal to or smaller than the bandwidth that should be allocated under saturation conditions will be able to carry all their offered traffic. Saturated stations, in addition to their corresponding bandwidth, will fairly share the excess time that is not used by the non-saturated nodes.

Based on these considerations, we propose a new algorithm for the load calculation in a multi-rate 802.11 WLAN cell, taking into account the traffic offered by each station (TO_i) and the values of T_{cycle} , L_i and S_i computed in saturation conditions as depicted in formulations (1), (2) and (3). The algorithm not only provides each station's load; it also allows bandwidth prediction for greedy applications (e.g. FTP, e-mail, etc.), which try to get all the bandwidth available to them. This definition is also applied to UDP applications that require more bandwidth than that already allocated to them. We define the parameter δ_i as the portion of the maximum throughput achievable by station i (if in saturation: S_i) that is actually used: $\delta_i = TO_i/S_i$. The proposed algorithm is as follows:

Algorithm 1: AAC Algorithm

```

Tcycle' ← Tcycle
OrderIncr(N, δi)
for all i ∈ N do
  if δi ≤ 1
    Li ← TOi/ri
    Texc ← (1-δi)/ri
    Tcycle' ← Tcycle' - 1/ri
  for all j ∈ N and δj > 1 do
    Lj ← Lj(1+Texc/Tcycle')
    Sj ← Ljrj
    δj ← TOj/Sj
  end for
end if
end for

```

Stations are first ordered on the basis of the increasing value of parameter δ . Stations with $\delta \leq 1$ use fewer (or equal) resources than those that would be allocated in saturation conditions and their offered traffic will therefore be carried. The portion of time not used by non-saturated stations (T_{exc}) will be divided fairly between the remaining stations, according to a new time T_{cycle}' in which all the stations that have already been served are not counted. Note that greedy applications are modeled with $\delta_j > 1$ regardless of S_j .

4.3. The Case of IEEE 802.11 WLANs

One particular case occurs as a result of not using priorities, i.e. the priority $p_i = 1/n \forall i \in N$. In this case, as we can see in (3) for saturation conditions, all stations will obtain an equal throughput, independently of their own rate r_i , and $S_i = S/n \forall i \in N$. This is the behavior that can be observed in an IEEE 802.11 WLAN (QoS option not implemented), since long-term channel access probability is

guaranteed to be equal for all stations due to CSMA/CA.

In order to know the actual bandwidth available to applications we must take Layer 2 overheads into account. Efficiency is also affected by the number of competing stations and the channel quality. We must expand expression (1) originally based on normalized length frame transmissions, to include overheads and the effect of channel errors and collisions for every node i (using basic access):

$$t_i = DIFS + \overline{T_{BO}}(RET_i) + T_{data}(i) + SIFS + T_{ACK} \quad (5)$$

$$T_{cycle} = \sum_{i=1}^n t_i \quad (6)$$

Note that t_i is defined for a basic CSMA/CA access; if RTS/CTS handshake is used, $2 \cdot SIFS$, T_{RTS} and T_{CTS} must be added to t_i . The duration of the data frame is T_{data} . $\overline{T_{BO}}$ represents the average time a station waits for the *backoff* timer to expire before attempting to transmit. $\overline{T_{BO}}$ depends on the number of previous transmission attempts. The average value of the *backoff* interval after j consecutive transmissions is given by:

$$\overline{T_{BO}}(j) = \begin{cases} \frac{2^j(CW_{min} + 1) - 1}{2} T_{slot} & 0 \leq j \leq 6 \\ \frac{CW_{max}}{2} T_{slot} & j \geq 6 \end{cases} \quad (7)$$

As previously done in [14], we consider that the number of packet retransmissions necessary to successfully transmit a single packet is a geometrically distributed random variable. If $P_f(i)$ is defined as the probability that an i 's frame has to be retransmitted (including both the effect of collisions and channel errors), the average number of transmissions for a frame is:

$$RET_i = 1 + \frac{P_f(i)}{1 - P_f(i)} \quad (8)$$

T_{data} can be further decomposed into¹:

$$T_{data}(i) = T_{preamble} + \frac{8 \cdot (H_{MAC} + MSDU_i)}{r_i} \quad (9)$$

where r_i is the physical bitrate at which node i sends data frames with a payload of $MSDU$ Bytes. The rest of the values vary depending on the standard and the modulation. Standard values for these parameters are summarized in table I.

We define the overhead produced in a node i 's layers 1 and 2 as $OH_i = r_i \cdot t_i / (8 \cdot MSDU_i)$. For the computation of the actual TO (offered traffic) we must take into account this overhead and the possible retransmissions knowing the traffic offered by upper layers (TO_{app}):

$$TO_i = \frac{OH_i}{1 - P_f(i)} TO_{app}(i) \quad (10)$$

Applying these changes to the formulation and algorithm presented in 4, we can do capacity estimations from real measurements. The AP is the only node that is able to compute in real time the available bandwidth for a given user since it maintains statistics from which all required parameters can be derived, including the physical rate used by each station, $MSDU$ size and P_f .

Summarizing, we have presented a new algorithm that provides an estimation of the throughput that the stations attached to a given AP can obtain. Our formulation and algorithm is easily applicable in EDCA enabled WLANs, selecting the proper values for p_i (out of the scope of this work). The core of the complexity for running this algorithm is discussed in the next section.

5. Implementation Issues

A key issue to solve when it comes time to implement the proposed algorithm lies in the acquisition of the required parameters. Most of them can be easily derived from statistics provided directly by driver/firmware functions. Nevertheless, obtaining P_f is not trivial since the actual number of frames sent (and lost) by its associated stations is not known by the AP without any kind of information exchange. But the AP knows the SNR at which it receives frames from all its client nodes. From SNR values and the modulation m used by the station for its transmissions, the AP is able to derive the BER (Bit Error Ratio) for a certain client, which can be obtained theoretically using the formulas given in [16]. However, we have used tabulated values from the empirical curves provided with the Intersil Prism HFA3863 transceiver data sheet [17]. Thus we avoid unnecessary complex calculations in the code. As in [18] [19], we use the following approximation to compute PER (Packet Error Ratio), knowing BER^m and the length of the packet (L bits):

$$PER_i^m = 1 - (1 - BER^m)^L \quad (11)$$

Note that we have omitted the effect of losing an ACK

Table I: IEEE 802.11 PHY Parameters

r (Mbps)	DSSS - CCK				OFDM			
	1	2	5.5	11	6	12	24	54
T_{ACK} (μ s)	304	248	212	202	44	32	28	24
$SIFS$ (μ s)	10				9			
$DIFS$ (μ s)	50				34			
T_{slot} (μ s)	20				9			
$T_{preamble}$ (μ s)	192				20			
CW_{min}	31				15			
CW_{max}	1023				1023			
H_{MAC} (Bytes)	28				28			

¹ case of DSSS-CCK modulations, for more details on OFDM see [15].

frame since its probability is negligible due to its short size and the fact that it is usually transmitted at the slowest (i.e. more reliable) bitrate.

On the other hand, the probability that a packet “sees” a collision (P_{col}) is the same for all stations (including the AP). Knowing its own P_f (# of packets not acknowledged / # of packets sent), the AP can do an estimation of P_{col} and an estimation of any $P_f(i)$ following:

$$P_f(i) = PER_i + P_{col} - PER_i P_{col} \quad (12)$$

5.1. Implementation details

The algorithm is fast and its implementation requires little resources. Therefore it is suitable for running on devices with limited features, such as any commercial AP. In order to prove the viability of our proposal, we implemented the algorithm in a commercial AP running a Linux-based OS: the 4G Systems AccessCube². The algorithm presented in section 4.2 with the modifications mentioned in 4.3 was programmed in standard C for Linux and cross-compiled to run on the AccessCube’s MIPS architecture. Some of this AP’s features include:

- 400MHz MIPS processor
- 32MB flash
- 64MB RAM
- 2 Prism 2.5 based WLAN IEEE 802.11b interfaces
- Linux-based OS: NyLon (kernel $\geq 2.4.27$)

The AccessCube’s OS includes the HostAP³ driver for the Intersil Prism 2.5 based 802.11b devices on board. This driver provides a helpful collection of statistics which are accessible to applications via the *proc* filesystem. For each associated station, the driver regularly updates information including the number of packets and the number of Bytes sent/received by that station, the number of packets sent at a given rate, and the SNR of the last received packet. The driver also provides information about the performance of the AP itself, e.g. packet discards, retransmission attempts, etc. All these parameters allow an application to fill in the variables needed to process the algorithm in a timely manner, so the effects of varying flows and channel conditions can be taken into account.

The resources employed by a single run of the algorithm are insignificant in most of the cases. This way, the performance of the AP is not affected by the extra load even though the AP is serving a large amount of clients, since most of the AP functionality relies on the wireless card firmware, which does not share resources with user space applications. In the extreme that the AP has to inspect statistics of 50 associated stations, the AccessCube’s CPU

time usage needed to compute the capacity available for a single station ranges from 20 to 30 ms and the memory usage reported by the OS is below 1.1%.

6. Evaluation

In order to validate our mechanism, first of all, several Opnet simulations were run using the 802.11g standard set of parameters. Without losing generality, the simulated scenario consisted of a single IEEE 802.11g cell with an AP and three stations (A , B and C): A used a physical rate of 48Mbps to carry the requirements of a greedy application (FTP), B offered traffic described by a 7Mbps UDP CBR source while transmitting at 24Mbps, and C increased its traffic demands linearly from 0 to 5Mbps using a physical rate of 12Mbps. Figure 3 shows the carried throughput measured at the application layer for all three stations. Solid lines are drawn for values obtained by simulation and the dotted lines are the values provided by our algorithm. In saturation conditions and following (3), the throughput that any of these three stations will obtain is about 5Mbps, and this is the value to which each station converges, as seen in figure 3. As C increases its offered traffic, not only it is clear that less capacity is available for greedy stations, but also since C is the slowest station, the global throughput is decreased as well. By applying the same concept to 802.11b, as shown in Fig. 1, we can compare our results with practical measurements. For example, if A transmits at a rate of 5.5Mbps, B transmits at 1Mbps and $P_f(A) = 0.03$, $P_f(B) = 0.04$ (from measurements at the AP, as explained in 5), following (5) and (6), one cycle is $T_{cycle} = 16.194$ ms. The individual loads are $L_A = 0.19$ and $L_B = 0.81$, i.e. 81% of the time, the channel is captured by station B . However, both stations are getting similar bandwidth, $S_A = 719$ kbps and $S_B = 711$ kbps. These numbers are close to the values shown in Fig. 1 ($S_A = 725 \pm 51$ kbps; $S_B = 698 \pm 44$ kbps). Note that the difference between the two stations’ bandwidth is due to the existence of packet losses: if $P_f(A) = P_f(B) = 0$, both stations would get an identical bandwidth. The cell is fully loaded ($L = 100\%$), but note that a load metric based on throughput measurements will provide an erroneous notion of the actual

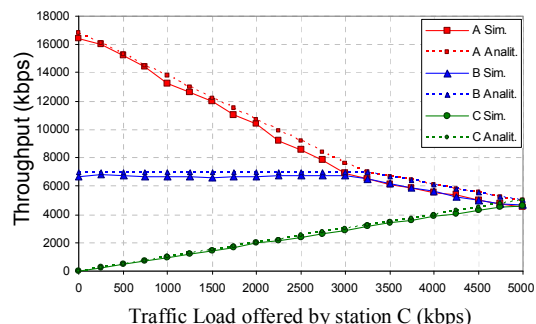


Fig. 3. Throughput of A (48M), B (24M) and C (12M) for different traffic demands in C : A in saturation and 7Mbps CBR in B .

² <http://www.meshcube.org>

³ <http://hostap.epitest.fi>

load since the carried throughput measured (1.43Mbps) seems a low load in comparison to the 6.50Mbps that can be achieved in an IEEE 802.11b WLAN.

6.1. Real-Time estimations

A real testbed was also set up in order to evaluate the fidelity of the measurements with the algorithm running on a commercial IEEE 802.11b AP (as detailed in 5.1). The testbed is completed with three stations. Before detailing the configuration of our testbed, it is worth having a look at the next figure: fig. 4, obtained analytically with our model, provides a helpful image illustrating how the available bandwidth per station varies depending on the modulations used in a cell with three transmitters in saturation, and sending maximum size frames; i.e. the curves in fig. 4 are an upper bound for saturation. Observe that the presence of a single node using the slowest modulation lowers drastically the global throughput, independently of the physical rates used by the rest of the stations (see dotted surface in fig. 4 is nearly flat). Once these upper bounds are known, we can choose the appropriate offered traffic and physical rates for our scenario so that the undesired effects of multi-rate contenders are clearly visible.

Back to the testbed description (see fig. 5); there are three stations *A*, *B* and *C*, associated with the same IEEE 802.11b AP. All three stations have different traffic profiles: *A* is the source of an MPEG-2 video stream. The MPEG-2 video codec can be formatted as constant length packets of 188 bytes (this is called Transport Stream) that can build payloads of $k \times 188$ bytes. The transport of TS packets over IP/UDP/RTP usually includes 7 TS packets = 1316+40 bytes, in order to approach the Ethernet MTU and maximize efficiency. *A*'s packets are spaced out so that a 2Mbps CBR stream is obtained.

In *B*, a greedy application is always trying to send as many 1500 byte UDP packets as possible, while *C* follows a bursty pattern: the average time between consecutive bursts is 20s, the average duration of a burst is 8s; bursts consist of 1000 byte UDP packets in such a way that STA *C* reaches

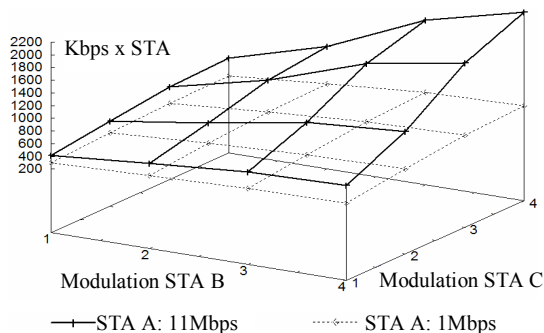


Fig. 4. Maximum achievable throughput for 3 stations in saturation different modulations (1, 2, 5.5 and 4=11Mbps)

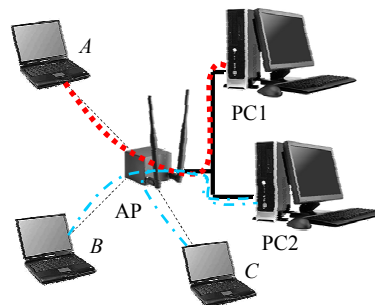


Fig. 5. Testbed for real time: AP runs algorithm and serves three STAs. Connected to PC1 and PC2 with 100M Ethernet

saturation.

The testbed topology is shown in figure 5. The stream originated in *A* is sent to *PC1* while *PC2* is the destination of data flows originated in *B* and *C*. *PC1* and *PC2* are directly connected to the AP by means of a 100Mbps Ethernet segment, so we can guarantee that the bottleneck resides in the air interface. *A* sends frames at 11Mbps, *C* uses 1Mbps and *B* decreases its bitrate one level (modulation) every 20s, starting with 11Mbps. For this experiment, the AP runs the algorithm once per second to provide an estimation of the maximum capacity available for station *A*.

Measurements are shown in fig. 6: during the first 10s station *A* has to compete with only one element, station *B*, which is sending frames at 11Mbps. The measured capacity is greater than *A*'s requirements so *A* can therefore carry all its offered traffic. From 10 to 20s, a burst of *C*'s packets makes the capacity measurements for *A* fall below 2Mbps, which corresponds to the real throughput obtained by *A*'s flow. During the period 20-28s, *B* sends frames at 5.5Mbps, and the capacity for *A* measured at the AP is again greater than. As it can be seen in the figure, the subsequent measurements are representative of the actual throughput obtained in *A*'s transmissions. This experiment has shown that the implementation in a commercial AP of the algorithm presented in previous sections is able to detect significant capacity fluctuations that affect a given station. We argue that this real-time knowledge in the AP will outperform any end-to-end estimation, in view of the fact that the AP is usually the system's bottleneck and is thus the best place to measure the actual capacity; but we leave a detailed comparison for a future work. Note that the algorithm is slightly overestimating the actual capacity; the values provided are, in average, less than a 10% above the actual throughput measured at the same time instant, while the error measured in the previous simulations were smaller than 5%.

7. Conclusions and future work

We examined the effect of multi-rate transmissions in an IEEE 802.11 WLAN cell. We verified that a fair bandwidth distribution is provided by CSMA/CA at the cost of

penalizing stations that use higher physical rates. These observations led us to develop a mechanism for computing throughput in a multi-rate 802.11 WLAN, which was validated against simulations and practical measurements. We also stated that the computation in terms of time provides an evaluation which is more representative of the actual cell load than the use of traffic (in Bytes/s). The availability of this metric has evident applications in the transmission of multimedia streams. These applications were widely discussed in the paper.

As a proof of concept, an implementation of the studied algorithm was developed to run in Linux-based commercial APs, showing the accuracy that can be provided in a real scenario. Once the algorithm has been validated, in the near future this testbed will be extended to evaluate the benefits of applying our metric in the scenarios discussed: admission control and dynamic CODEC adaptation.

Observe that the considerations taken in the initial applicability statements may require some extensions to the basic AC capabilities of deployed WLAN AP's. In particular, AC in a WLAN-GAN or WLAN-IMS environment may be shared across a set of network elements, including the AP itself, Policy Decision and Policy Enforcement Points, Application Layer nodes (e.g.: SIP proxies) and operator policies. Alternatively, if a station wishes to perform proper CODEC selection based on estimated throughput, a mechanism to transmit such information from the AP towards the station should be developed (at present, only the AP is able to implement the algorithm, while the stations remain unaware of its results).

Acknowledgment

This research work was funded by the ERDF, the Spanish Government through TEC2006-04504, and the i2CAT foundation.

References

[1] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proceedings of the 22nd IEEE INFOCOM'03*, vol. 2, pp. 836-843, March 2003.

[2] S. Garg and M. Kappes, "Can I add a VoIP call?," in *IEEE International Conference on Communications, ICC'03*, vol. 2, pp. 779-783, May 2003.

[3] N. Smavatkul, Y. Chen and S. Emeott, "Voice Capacity of IEEE 802.11a with Automatic Rate Selection," in *Proc. of IEEE GLOBECOM '03*, vol. 1, pp. 518-522, December, 2003.

[4] A.J. Nicholson, Y. Chawathe, M.Y. Chen, B.D. Noble and D. Wetheral, "Improved Access Point Selection," in *Proc. of MobiSys 2006*. June, 2006.

[5] N. Cranley and M. Davis, "The Effects of Background Traffic on the end-to-end Delay for Video Streaming Applications over IEEE 802.11b WLAN Networks," in *17th IEEE Personal, Indoor Mobile Radio Communications PIMRC*, Sept. 2006.

[6] M. Narbutt and M. Davis, "Gauging VoIP call quality from 802.11 WLAN resource usage," in *Proc. of the 2006 International*

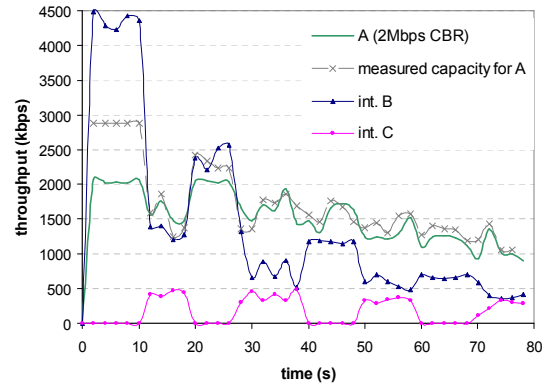


Fig. 6. Real-time capacity measurements for *A* in the presence of interference from *B* (greedy) and *C* (bursty)

Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM'06, June, 2006.

[7] A. Sfairopoulou, C. Macián and B. Bellalta, "QoS Adaptation in SIP-based VoIP calls in Multi-rate IEEE 802.11 Environments," *3rd IEEE Int. Symposium on Wireless Communication Systems, ISWCS'06*, September 2006.

[8] G. Convertino et. al., "Wireless Adaptive Video Enhanced Streaming: WAVES," in *ST Journal of Research*, vol. 2, num. 1, pp: 158-167, 2005.

[9] 3GPP TS 23.203 v7.0.0, "Policy and Charging control architecture (Release-7)", September 2006.

[10] S. Garg and M. Kappes, "Admission Control for VoIP Traffic in IEEE 802.11 Networks," in *Proc. of IEEE Global Telecommunications Conference, 2003 GLOBECOM '03*, vol. 6, pp. 3514 - 3518, December, 2003.

[11] A. Bazzi, M. Diolaiti, and G. Pasolini, "Measurement based Call Admission Control Strategies in Infrastructured IEEE 802.11," *16th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'05.*, September 2005.

[12] IEEE 802.11 WG. IEEE Standard for Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications. New York, USA: The IEEE, Inc., 1999.

[13] IEEE 802.11 WG. IEEE Standard for Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service Enhancements, IEEE 802.11e. New York, USA: The IEEE, Inc., November 2005.

[14] G. Bianchi and I. Tinnirello, "Improving Load Balancing Mechanisms in Wireless Packet Networks," in *IEEE International Conference on Communications 2002, ICC 2002*, vol. 2, pp. 891-895, April 2002.

[15] D. Qiao, S. Choi, and K. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs," *IEEE transactions on Mobile Computing*, vol. 1, pp. 278-292, December 2002.

[16] J.G. Proakis. *Digital Communications*. McGraw-Hill, 4th ed. 2001.

[17] HFA3863: *Direct Sequence Spread Spectrum Base-band Processor with Rake Receiver and Equalizer*. Data Sheet. Intersil Inc., Dec. 2001.

[18] T. Nadeem and A. Ashok, "IEEE 802.11 DCF Enhancements for Noisy Environments", in *The 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'04*, September 2004.

[19] P. Chatzimisios, A. Boucouvalas, and V. Vistas, "Influence of channel BER on IEEE 802.11 DCF," *Electronics Letters*, vol. 39, pp. 1687-1689, November 2003.