Performance Analysis under Finite Load and Improvements for Multirate 802.11

Gion Reto Cantieni, Qiang Ni *, Chadi Barakat, Thierry Turletti

Planète Group - INRIA Sophia Antipolis, France

Abstract

Automatic rate adaptation in CSMA/CA wireless networks may cause drastic throughput degradation for high speed bit rate stations (STAs). The CSMA/CA medium access method guarantees equal long-term channel access probability to all hosts when they are saturated. In previous work it has been shown that the saturation throughput of any STA is limited by the saturation throughput of the STA with the lowest bit rate in the same infrastructure. In order to overcome this problem, we first introduce in this paper a new model for finite load sources with multirate capabilities. We use our model to investigate the throughput degradation outside and inside the saturation regime. We define a new fairness index based on the channel occupation time to have more suitable definition of fairness in multirate environments. Further, we propose two simple but powerful mechanisms to partly bypass the observed decline in performance and meet the proposed fairness. Finally, we use our model for finite load sources to evaluate our proposed mechanisms in terms of total throughput and MAC layer delay for various network configurations.

Key words: IEEE 802.11b, wireless LAN, Stochastic processes, Queueing theory, Network measurements

1 Introduction

In recent years, the IEEE 802.11b protocol for wireless LAN (WLAN) has become very popular as an access scheme for wireless and mobile Internet users. Access Points (APs) can be deployed wherever service customers need fast and mobile access to information. Such environments can be an airport, a campus, or a business building. The IEEE 802.11b standard specifies the Medium Access Control (MAC) layer, as well as the physical (PHY) layer. Currently, for the MAC layer, the standard defines two medium access coordination functions: the contention-based Distributed Coordination Function (DCF) and the contention-free based Point Coordination Function (PCF) [1]. In this paper we only consider the DCF

* Corresponding author. This work has been done while he was with INRIA. Present address: Hamilton Institute, National University of Ireland Maynooth (NUIM), Co. Kildare, Ireland. Fax: +353-17086269.

Email address: `Qiang.Ni@ieee.org` (Qiang Ni).

Preprint submitted to Elsevier Science *9 June 2004*

access method. The PCF access method is not mandatory and, therefore, is rarely implemented in current 802.11b products.

The DCF access method is based on the *Carrier Sense Multiple Access* with *Collision Avoidance* (CSMA/CA) principle. Each STA has the same priority when competing for an empty slot time, which guarantees long-term fairness in access probability. Before an STA attempts a *first* packet transmission, it has to sense the medium. If the medium is found idle for a minimum time equal to the Distributed Inter Frame Space (DIFS), the packet will be transmitted directly. Otherwise, the STA enters into backoff and randomly sets its backoff timer within the range of the Contention Window (CW). The backoff timer is decremented by one every slot time the medium is sensed to be idle and it is frozen when medium is sensed busy. When it reaches zero, the STA starts the next transmission. Upon the correct receipt of a packet, the receiver has to send an acknowledgment (ACK) after a time equal to the Short Inter Frame Space (SIFS). If no ACK is received, the sending STA assumes a collision, doubles its current CW, randomly resets its backoff timer, and retransmits the packet when the timer reaches again 0.

The IEEE 802.11b specifications for the PHY layer support multi-rate adaptation and allow channel bit rates up to 11Mb/s. As in any wireless communication system, bit errors due to noise and interference from the Industrial-Science-Medical (ISM) band are of fundamental concern. High bit error rates in wireless environments require not only sophisticated channel coding but also control over the channel modulation rate. It is well known that a decrease in the symbol period increases the probability of an incorrect detection. The 802.11b standard tackles this problem by offering four different modulation rates. The mechanism, which is implemented in current 802.11b products, counts the number of unsuccessful frame transmissions and reduces its channel bit rate accordingly from 11Mb/s to either 5.5Mb/s, 2Mb/s, or 1Mb/s. However, the standard does not consider the fact that packet transmission at 1Mb/s might take up to eleven times longer than an equal packet size transmission at 11Mb/s! The standard still guarantees all STAs the same long-term medium access probability. As a result, the medium underlies a completely unfair time allocation for STAs with different rates. This unfairness is especially reflected in the throughput of the STA with the highest bit rate, namely 11Mb/s. It has been proven in [3] that, if there are two different bit rates in the same environment, the saturation throughput of any STA will be equal to the saturation throughput of the STA with the lowest channel bit rate. For instance, this phenomenon can be seen in Figure 1 where we measure the saturation throughputs of two STAs, one fixed STA close to the AP and transmitting all of the time at its maximum rate 11 Mb/s, and another STA moving around the AP whose transmission rate varies as indicated in the top figure. Both STAs are saturated with UDP packets of payload size equal to 1470 bytes. The bottom figure shows the saturation throughput of both STAs averaged over 1 second time intervals. We notice how the saturation throughput of the fixed STA follows that of the moving STA even though the close STA has an excellent wireless connection to the AP and always has data frames to transmit.

This *performance anomaly* of IEEE 802.11b has been analyzed in [3] using a simplified model and assuming saturated sources, further no solutions are proposed in [3]. We define a saturated source as an STA always having packets to send in its queue. In [6] the complex behavior of 802.11b protocol is analyzed with Markov chains, assuming one single modulation rate and saturated sources.

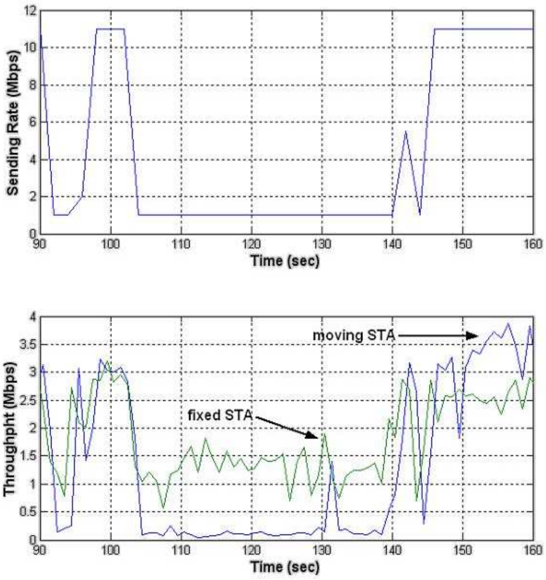

Fig. 1. The throughput of one fixed STA and one moving STA.

In our real 802.11b testbed we conduct experiments which show that the throughput degradation faced by high-rate STAs strongly depends on how loaded the low-rate STAs are. This explains the need for a model considering non-saturated as well as saturated sources. An analytical model for non-saturated sources is proposed in [4] based on [6], however, the assumptions only hold for very low traffic load. Although an infinite MAC buffer is considered, the model in [4] discards all packets in the buffer after the first packet has been taken by the DCF. In [5], a different approach is taken to analyze the performance under statistical traffic. The on-off characteristics of the STAs are modelled with a state-dependent single server queue where the service time for the different states are estimated from the saturation throughput obtained in [6]. This model assumes equal service time and equal packet sending rates for all participating STAs and, therefore, cannot be applied to multirate environments. Moreover, the model in [5] is not very accurate in modelling finite-load scenarios since it supposes that an STA reaches directly its saturation throughput, which is not acceptable if the active time of an STA is comparable to the transitory regime duration. All this motivates us to develop a model for finite load sources with a MAC buffer for multirate environments. We explain in this paper the model for the case of two bit rates. Its extension to more than two bit rates is a straightforward exercise that we omit for lack of space. Note that we present at the end of the paper some analytical results for the case of three bit rates. Clearly our model for non saturated STAs in multirate environments can be specified to the case of one bit rate, which in itself is an important finding.

We analyze the fairness problem of 802.11b in multirate environments using our model and real experiments. The observed performance anomaly drives the need for a different fairness metric. Thus, we propose a new fairness index giving equal channel occupation time to all STAs. We provide two solutions (optimal minimum CW and optimal payload size) to meet our fairness index and to improve the total throughput in multirate environments.

The rest of this paper is organized as follows. In the following section, we introduce and derive our proposed model for finite load sources in the case of two bits rates. In Section 3, we validate our model based on realistic experiments in our 802.11b testbed. In Section 4, we define a new fairness index, then we present and evaluate our two proposed mechanisms. In Section 5 we present a validation by analytical modelling of our proposed mechanisms in the case of three bit rates. Finally in Section 6 we conclude our work.

2 Model for finite load sources

In our model, for any source load and for multirate environments, we take a novel approach by actually modeling the MAC buffer with an $M/G/1$ queue. The consideration of other queues is possible (for instance finite buffer queues) but this requires the distribution of the time a packet spends at the MAC layer before being correctly transmitted, which is complex to obtain without any particular approximation, so we leave this extension of our model to a future research. With the $M/G/1$ queue we only need the average of this time, whose computation can be accurately done as we will later see. In addition to the $M/G/1$ assumption to model the buffers in STAs, our model is generalized to support n_S STAs at a physical rate of S (indexed $k = 1, 2, \ldots n_S$) and n_F STAs at a physical rate of F (indexed $k = n_S + 1, n_S + 2, \ldots n_F + n_S$). Let $n = n_S + n_F$ be the total number of STAs. In our experiments we mostly use 1Mb/s for the rate S (S for slow) and 11Mb/s for the rate F (F for fast). For the rest of this paper we call STAs with rate S *slow* whereas STAs with rate F *fast*. We choose a Poisson process with rate λ_k packets/second to model the arrivals of packets at the MAC buffer of STA k. Even though the Poisson assumption may not be realistic, it provides insightful results and allows our model to be tractable. Our model is based on the following assumptions:

1) The effects of bit errors due to noise are ignored. Consequently, packets are lost only when they encounter collisions due to other simultaneous transmissions.
2) Propagation delays and hidden terminals are not considered.
3) The collision probability is independent of the number of retransmissions.
4) Each STA is assumed to have an infinite buffer and new packets are assumed to arrive according to a Poisson process.

2.1 Our approach

Our proposed model, depicted in Figure 2 and which is in part based on the one proposed in [6] for the case of saturated sources, consists of an aggregation of states in which an STA can reside. As defined in the standard, an STA has to run at least one backoff between two successive transmissions [1]. Therefore, after each successful transmission, the CW is reset to its minimum value W_0 and the STA enters into a backoff even if there is no packet in the queue. In our model, we check the queue after each successful transmission or after having reached the maximum number of retransmissions m. If there is a packet, we enter into backoff directly, otherwise, if the queue is empty, we enter into a separate backoff which we call *post-backoff*. The backoff states have been already proposed in [6], our model proposed in addition the post-backoff states (on the top left of the Markov chain in Figure 2), which are necessary for the study of the non-saturated regime. In Figure 2, q denotes the probability of having an empty queue after a packet has been successfully transmitted or after having reached the maximum number of retransmissions. The MAC queue is checked again after the post-backoff has expired. If there is a packet, it will be transmitted directly in next slot time, otherwise the STA will reside in a vacation state $notx$ until the next packet arrives. $p_{pb\to notx}$ denotes the probability of having no packet in the queue at the end of the post-backoff. If in the $notx$ state the medium is sensed idle at the occurrence of the first packet in the MAC queue, the STA sends the packet immediately from state $frtx$. For this case the medium is sensed busy at the first arrival, the STA enters into backoff and the packet is transmitted when the backoff timer reaches 0. With probability $p_{notx\to frtx}$ the medium is sensed idle at the first packet arrival and with probability $p_{notx\to bo}$, it is sensed busy.

For a general STA k, $k \in [1, n]$, we use the tuple (s, r) to represent the different states in the backoff stages, with s being the backoff stage number $s = 0', 0, 1, \ldots, m', \ldots, m$, and r being the value of the backoff timer in the range $[0, W_s - 1]$. W_s is the size of the CW at stage s and is computed by $W_s = 2^s W_0$ if $s \leq m'$. Otherwise, if $s \in [m', m]$, W_s is kept at its maximum value $W_{max} = 2^{m'} W_0$. With m we denote the maximum number of packet retransmissions before the packet is dropped. According to [1], the default value for m' is 5 and it is 7 for m. We use $s = 0'$ to account for the post-backoff stage. $\pi_{s,r}$ will denote the probability to be in state (s, r). For the remaining two states, $notx$ and $frtx$, we denote with π_{notx} and π_{frtx} their respective state probabilities. With p we denote the probability that the packet transmitted by STA k collides, which is equal to the probability that at least one other STA transmits a packet at the same time. We assume that the packet collision process is Bernoulli. This assumption has been made in [6] and has shown good performance in computing the throughput of 802.11, especially when the number of STAs is high. The Bernoulli assumption allows us to describe the state of an STA with the discrete time Markov chain depicted in Figure 2. The state transitions appear at the beginning of each slot time, where a transition may be executed after a transmission or an empty slot time. Therefore, the interval between the beginnings of two consecutive slot times may have either the length of an empty slot time σ_0 or of a packet transmission (successful or not).

It can be seen that our Markov chain is ergodic, therefore, a unique stationary distribution exists. Also note that only the oval forms represent states where actually time is spent. The hexagons in Figure 2 represent circuit points which we name C_0 and C_1. We use lower case c_0 (resp. c_1) to denote the probability to cross C_0 (resp. C_1). In Appendix I we derive expressions for c_0 and c_1 as a function of $\pi_{0,0}$.

2.2 Transmission probability

We now derive the expression for the state probabilities in steady state of a general STA k, $k \in [1, n]$. In a first step we express all state probabilities in terms of $\pi_{0,0}$. Later, we use the normalization condition to obtain $\pi_{0,0}$ itself. From the balance equation in the steady state we obtain the following relations:

$$\pi_{0',r} = \frac{W_0 - r}{W_0} q\, c_0, \qquad r \geq 1, \tag{1}$$

$$\pi_{0,r} = \frac{W_0 - r}{W_0} c_1, \qquad r \geq 0, \tag{2}$$

$$\pi_{s,r} = \frac{W_s - r}{W_s} p^s\, \pi_{0,0}, \qquad 1 \leq s \leq m, \tag{3}$$

where W_s is the length of the CW in stage s and is equal to $2^s W_0$ for $0 \leq s \leq m'$ and to $2^{m'} W_0$ for $m' \leq s \leq m$. The probability π_{notx} to be in state $notx$ is equal to:

$$\pi_{notx} = \frac{c_0\; q\; p_{pb\to notx}}{p_{notx\to frtx} + p_{notx\to bo}}. \tag{4}$$

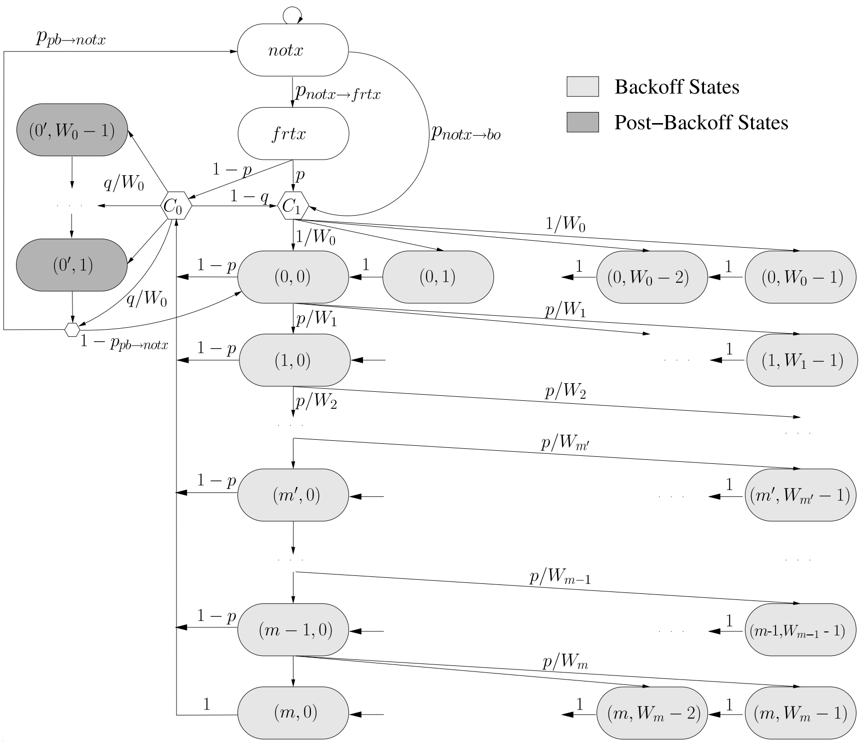

Fig. 2. Finite load source model for DCF

Using (4) the state probability π_{frtx} becomes:

$$\pi_{frtx} = p_{notx\to frtx}\ \pi_{notx}. \qquad (5)$$

The expressions for the transition probabilities $p_{pb\to notx}$, $p_{notx\to frtx}$, $p_{notx\to bo}$, and q are derived in the next subsections. Equations (1), (2), (3), (4), and (5) express all state probabilities as a function of $\pi_{0,0}$. $\pi_{0,0}$ is obtained by using the normalization condition:

$$1 = \sum_{s=0}^{m}\sum_{r=0}^{W_s-1}\pi_{s,r} + \sum_{r=1}^{W_0-1}\pi_{0',r} + \pi_{notx} + \pi_{frtx} \quad (6)$$

From (6) we obtain $\pi_{0,0}$ as a function of p. The collision probability (p) is equal to the probability $p_{k,otr}$ that at least one of the other $n-1$ STAs transmits a packet. Therefore, p for STA k can be written as:

$$p = p_{k,otr} = 1 - \prod_{\substack{i=1\\ i\neq k}}^{n}(1-\tau_i), \qquad (7)$$

where τ_i denotes the probability that an STA i transmits a packet in a randomly chosen slot time. Such a transmission occurs either if the backoff timer (r) of an STA reaches zero or if an STA, after some idle period in state $notx$, conducts a first transmission from state $frtx$. The transmission probability for STA k can be obtained from:

$$\tau_k = \sum_{s=0}^{m}\pi_{s,0} + \pi_{frtx}, \qquad k\in[1,n]. \qquad (8)$$

Using (8) we can setup a non-linear system of equations where the τ_k are the unknowns. Indeed, τ_k can be expressed using (7), (8) and the state probabilities of the Markov chain of the STA k as a function of p, and hence as a function of the transmission probabilities of the other $n-1$ STAs: $\tau_k = f\left(\tau_1,\tau_2,\ldots,\tau_{k-1},\tau_{k+1},\ldots,\tau_{n_S+n_F}\right)$. Thus, we obtain n_S+n_F equations with the same number of unknowns, which allows us to compute all τ_k. We solve this system using the method $fsolve$ implemented in the MATLAB optimization toolbox.

2.3 Transition probabilities

We now derive the expressions for the transition probabilities. These probabilities are not equal for all STAs, so we use the index k to refer to STA for which we are computing the probabilities. Note that we use for the derivation both variables p and $p_{k,otr}$, although they have the same meaning. We use p when we know that STA k attempts a transmission. p is the probability that this transmission fails due to collisions. We use $p_{k,otr}$ for the cases where STA k is not involved in a medium access for the current slot. $p_{k,otr}$ is the probability that one or more stations other than k attempt to transmit a packet in the current slot time.

The transition probability from the post-backoff state to the *notx* state is denoted by $p_{pb\rightarrow notx}$ and is equal to the probability of not receiving any packet during the time spent in the post backoff. The timer for the post-backoff is a Random Variable (RV) B, which is uniformly distributed over the interval $[0, W_0 - 1]$. Note that if the timer is chosen to be 0, then the STA will enter directly into the *notx* state with probability 1. Further, we introduce a random vector $\overline{\sigma} = \{\sigma_1, \ldots, \sigma_B\}$ of length B representing the sequence of slot lengths observed by STA k. The time STA k resides in the post-backoff state is equal to the sum over all σ_i. Knowing B and $\overline{\sigma}$ one can write $p_{pb\rightarrow notx|B,\overline{\sigma}} = e^{-\lambda_k \sum_{i=1}^{B} \sigma_i}$. Now using the uniform distribution of B and the assumption that all σ_i are independent and identically distributed, we show in Appendix II that $p_{pb\rightarrow notx}$ can be written as follows:

$$p_{pb\rightarrow notx} = \frac{1}{W_0} \sum_{b=0}^{W_0-1} E\left[e^{-\lambda_k \sigma}\right]^b. \qquad (9)$$

To compute (9) we need the distribution of the RV σ, which is the length of a random slot time observed by an STA k. The RV σ may take different values in six cases depending on the transmission events of the other active STAs. With σ_0, we refer to the length of an idle slot time and is in our case equal to 20μs. T_s is the time the medium is sensed busy if a successful transmission occurs, while T_c represents the time the medium is busy when a collision occurs. When T_s and T_c are indexed with S, they are parameters at physical rate S, and when indexed with F, the rate is F. We will give later the explicit expressions for T_c and T_s. To derive the distribution of σ, we introduce four new probabilities whose explicit equations are given in Appendix III. The probabilities that at least one of the n_S slow STAs and one of the n_F fast STAs transmit a packet are denoted respectively by $p^S_{k,otr}$ and $p^F_{k,otr}$. With $p^S_{k,os}$ we indicate the probability of having a successful transmission by one of the n_S slow STAs knowing that at least one slow STA transmits a packet. We give an equivalent meaning for $p^F_{k,os}$. Using the nomination from above, the distribution of σ is equal to:

1) If no other STA transmits a packet, σ is equal to the length of an idle slot time σ_0 with probability:

$$P\{\sigma = \sigma_0\} = 1 - p_{k,otr}. \qquad (10)$$

2) Further, if one of the n_S slow STAs successfully transmits a packet, the slot length is equal to T^S_s. Such a slot length can be observed with probability:

$$P\{\sigma = T^S_s\} = p^S_{k,otr}\ p^S_{k,os}. \qquad (11)$$

3) Similarly, if one of the n_F fast STAs successfully transmits a packet, the slot length becomes T^F_s and has probability:

$$P\{\sigma = T^F_s\} = p^F_{k,otr}\ p^F_{k,os}. \qquad (12)$$

4) A slot length of T^S_c can be observed if at least two STAs out of the n_S slow STAs transmit packets simultaneously and so cause a collision. In addition, the condition that none of

the other fast n_F STAs transmits a packet has to be imposed, to be sure that the collision happens explicitly between the n_S slow STAs. Therefore, such a slot length can be observed with probability:

$$P\{\sigma = T_c^S\} = p_{k,otr}^S(1 - p_{k,os}^S - p_{k,otr}^F). \quad (13)$$

We deduce $p_{k,otr}^F$ since $1 - p_{k,os}^S$ as defined corresponds to the probability that the packet transmitted by a slow STA collides, but does not impose that the other STAs causing the collision should be of S type, they can be of S or F type.

5) Similarly, a collision explicitly within at least two STAs out of the n_F fast STAs implies a slot length of T_c^F and has probability:

$$P\{\sigma = T_c^F\} = p_{k,otr}^F(1 - p_{k,os}^F - p_{k,otr}^S). \quad (14)$$

6) The last case is that at least one out of the n_S slow STAs and one out of the n_F fast STAs are involved in a collision. Then, the length of a slot time will be the maximum of either T_c^S or T_c^F with probability:

$$P\{\sigma = \max\left(T_c^S, T_c^F\right)\} = p_{k,otr}^S \, p_{k,otr}^F. \quad (15)$$

The transition probability from state *notx* to state *frtx* is denoted by $p_{notx\to frtx}$ and is given by the probability that during an empty slot σ_0, at least one packet arrival occurs:

$$p_{notx\to frtx} = P\{\sigma = \sigma_0\}(1 - e^{-\lambda_k \sigma_0}). \quad (16)$$

The probability to transit from state *notx* to backoff stage 0 considers all complementary events from (16) and so its probability equals: $p_{notx\to bo} = E\left[1 - e^{-\lambda_k \sigma}\right] - p_{notx\to frtx}$.

2.4 Computation of the probability q

The properties of the $M/G/1$ MAC buffer intervene in the computation of q, the probability of having no packet in the buffer upon packet departures from the MAC layer of an STA. In an $M/G/1$ queue, q is simply equal to

$$q = \max\left(0, 1 - \lambda_k E[S_{T,k}]\right), \quad (17)$$

where $E[S_{T,k}]$ is the first moment of the service time for packets from STA k. The $M/G/1$ queue has the property that the distribution of queue length is the same at packet arrivals, packet departures, and at random time [7]. In our model, $S_{T,k}$ is the time that a packet spends in the MAC layer from the point of leaving the MAC buffer until its successful transmission (or until the abortion of its transmission when the maximum number of backoff stages m is reached). This time has an unknown complex distribution, from here comes the need for the M/G/1 queue. Note that to compute q for more complex queues than $M/G/1$ (for instance finite buffer queues), we need the complete distribution of $S_{T,k}$, which is not easy to obtain, the computation of the average service time is already a difficult problem.

A packet may have different average service times depending on the state of the queue upon its arrival. We consider the cases of having an empty or nonempty queue, therefore, $E[S_{T,k}]$ has to be conditioned on q and is equal to:

$$E[S_{T,k}] = (1 - q)E[T_{bo,k}] + qE[T_{pb,k}]. \quad (18)$$

We first derive an expression for $E[T_{bo,k}]$, which is the average service time of a packet that finds the queue non empty when it arrives. We have:

$$\begin{aligned} E[T_{bo,k}] =& \sum_{s=0}^{m} p^s \frac{W_s - 1}{2} E[\sigma] + (1 - p^{m+1})T_{s,k} \\ &+ \sum_{s=1}^{m+1} p^s(1 - p)sE[T_{col}]. \quad (19) \end{aligned}$$

The first term in (19) accounts for the total time needed to attain a transmission state, which is called $(s, 0)$ in Figure 2, where

$s = 0, \ldots, m$. A transmission state is the state which represents the value 0 of the backoff timer, and therefore triggers directly a packet transmission. The second term is the expected value of the time needed to actually accomplish the physical transmission and the receipt of the ACK. If $k \in [1, n_S]$, $T_{s,k}$ is equal to T_s^S (32), otherwise $T_{s,k}$ is equal to T_s^F (33). The third term accounts for the expected number of collisions that the STA k might enter. $E[T_{col}]$ is the average time that STA k spends in a collision. For anyone of the n_S slow STAs, k is within the interval $[1, n_S]$, $E[T_{col}]$ becomes:

$$E[T_{col}] = \frac{(1 - p^F_{k,otr})p^S_{k,otr}}{p_{k,otr}} T_c^S + \frac{p^F_{k,otr}}{p_{k,otr}} \max\left(T_c^F, T_c^S\right). \tag{20}$$

For anyone of the n_F fast STAs, k is within $[n_S + 1, n_S + n_F]$, $E[T_{col}]$ is equal to:

$$E[T_{col}] = \frac{(1 - p^S_{k,otr})p^F_{k,otr}}{p_{k,otr}} T_c^F + \frac{p^S_{k,otr}}{p_{k,otr}} \max\left(T_c^F, T_c^S\right). \tag{21}$$

We now seek for an expression of the second term $E[T_{pb,k}]$ in (18). With $E[T_{pb,k}]$ we denote the average service time of a packet that at its arrival finds the MAC queue empty. As it is shown in Figure 2, a packet may arrive either while the node resides in the post-backoff, or it may arrive after the post-backoff has already expired and so finds the node in state *notx*. We introduce a Bernoulli RV V to condition on whether the STA enters after the post-backoff into the vacation state *notx* or enters directly into state $(0, 0)$. We define V as follows: $V = 1$ with probability $p_{pb\to notx}$ and $V = 0$ with probability $(1 - p_{pb\to notx})$. Therefore, if V is equal to 0, the node conducts a transmission attempt directly from state $(0, 0)$, otherwise, if V is equal to 1, the node will reside in *notx* and will wait for the next packet arrival. Therefore, we can condition the value of $E[T_{pb}]$ upon V and write:

$$E[T_{pb,k}] = (1 - p_{pb\to notx})(E[T_{inpb}] + E[T_{pb\to bo} \mid V = 0]) + p_{pb\to notx} E[T_{pb\to bo} \mid V = 1]. \tag{22}$$

With $E[T_{inpb}]$ we express the expected time the packet waits since its arrival before being transmitted, knowing that it arrives during the post-backoff. This time is strictly positive if the packet arrives before the post-backoff timer reaches 0. In order to find $E[T_{inpb}]$, we generalize the problem to find the average residual time $R(X)$ for a packet that arrives at rate λ, given the observation interval $[0, X]$. Suppose that the packet arrives at instant $x_0 < X$, then $R(X) = X - x_0$, else if the packet arrives at an instant $x_0 > X$, $R(X)$ becomes zero. We propose the following function for $R(X)$ and prove it in Appendix IV.

$$R(X) = X + \frac{e^{-\lambda X}}{\lambda} - \frac{1}{\lambda} \tag{23}$$

$E[T_{inpb}]$ can now be found by setting X to the length of the post-backoff (sum of all σ_i) and conditioning it on the fact that we know that at least one packet arrived in the desired interval. As X consists of a random vector $\overline{\sigma} = \{\sigma_1, \ldots, \sigma_B\}$ and a RV B, $R(X)$ has to be computed, similarly to (9), by taking its expected value with respect to $\overline{\sigma}$ and B. Therefore, we can write the following relation: $(1 - p_{pb\to notx}) \cdot E[T_{inpb}] = E[R\left(\sum_{i=1}^{B} \sigma_i\right)]$.

Making the same assumption about independence of σ_i as for computing (9), one can obtain the following expression for $E[T_{inpb}]$:

$$E[T_{inpb}] = \frac{\frac{1}{W_0}\sum_{b=0}^{W_0-1}\left(bE[\sigma] + \frac{1}{\lambda_k} E\left[e^{-\lambda_k \sigma}\right]^b\right) - \frac{1}{\lambda_k}}{1 - p_{pb\to notx}} \tag{24}$$

Further from (22), assuming that at least one packet arrives during the post-backoff (V=0), $E[T_{pb\to bo} \mid V=0]$ accounts for the time needed to send successfully the packet (or to abort its transmission) starting in the transmission state $(0,0)$:

$$E[T_{pb\to bo} \mid V=0] = (1-p)T_{s,k} + pE\left[T_{col} + T_{bo,k}^{(i=1)}\right]. \quad (25)$$

Index $(i=1)$ denotes that (19) is computed starting from stage one instead of stage zero.

If V turns out to be equal to 1, then we start counting the time for a successful transmission upon the packet arrival in the state $notx$. Two cases exist: One is that the packet arrives when the medium is busy, another is that the packet arrives during an empty slot time.

$$E[T_{pb\to bo} \mid \ V=1] = (1-p_{k,otr})E[T_{ftx}] + p_{k,otr}E[T_{noftx}], \quad (26)$$

where $E[T_{ftx}]$ considers that the medium is idle upon the arrival of the packet:

$$E[T_{ftx}] = \frac{P\{\sigma=\sigma_0\}R(\sigma_0)}{p_{notx\to frtx}} + pE[T_{col} + T_{bo,k}] +(1-p)T_{s,k}, \quad (27)$$

and $E[T_{noftx}]$ treats the case where the medium is sensed busy upon the packet arrival, and so the STA enters directly into backoff without conducting a first transmission attempt from state $frtx$:

$$E[T_{noftx}] = E[T_{bo,k}] + \frac{E[R(\sigma)] - (1-p_{k,otr})R(\sigma_0)}{p_{notx\to bo}}. \quad (28)$$

Finally, combining (17) and (18) the following expression for q can be obtained:

$$q = 1-\min\left(1, \frac{\lambda_k E[T_{pb,k}]}{1-\lambda_k E[T_{bo,k} - T_{pb,k}]}\right). \quad (29)$$

The second term is strictly larger than 0 and therefore has only to be upper bounded by 1.

2.5 Implementation of the model

Our model consists mainly in a non-linear system of n equations where the unknowns are the transmission probabilities τ_i of STAs. To build this system we proceed as follows:

1) Take an STA, say k, and use (8) to write its transmission probability τ_k as a function of its collision probability p, the stationary probability if its state (0,0) $\pi_{0,0}$, and the probability to find its queue empty q.
2) Find the expression of $\pi_{0,0}$ as a function of p of q using the normalization equation (6).
3) Express q as a function of p and the τ_i of the other STAs using (29).
4) Finally, express p as a function of the τ_i of the other STAs using (7).

We implement this system in MATLAB and we solve it numerically for the τ_i of all STAs. Then, we proceed from bottom to top in the above points to compute the other parameters of the model as p, q, $\pi_{0,0}$, and other stationary probabilities.

2.6 Throughput analysis

We derive now the throughput of each individual STA. Analogous to the RV σ, we introduce the RV σ_G which gives the length of a general slot time accounting for all n STAs. The distribution of σ_G is equivalent to that given by equations (10)-(15) except that now the k^{th} STA should also be considered as a transmitting STA.

The throughput Z_k for any STA k is by definition the volume of data STA k successfully transmits in a slot time divided by the average slot length $E[\sigma_G]$:

$$Z_k = \frac{1}{E[\sigma_G]} \tau_k (1 - p_{k,otr}) P_k. \qquad (30)$$

P_k is the payload size of STA k. It is equal to P_S for a slow STA and to P_F for a fast STA.

In our analysis we do not consider the RequestToSend / ClearToSend (RTS/CTS) access method and the throughput is computed at the application layer. The packet header from the transport, network, and data link control layer [8] is equal to:

$$H = MAC_{hdr} + IP_{hdr} + TRANSPORT_{hdr}. \qquad (31)$$

Upon a successful receipt of a packet, an ACK is transmitted at the physical rate of the received packet. The duration of an ACK is t^S_{ACK} for a slow STA and t^F_{ACK} for a fast one. In addition, the PHY layer adds to each transmission a constant Physical Layer Convergence Protocol (PLCP) preamble and header of total duration t_{PLCP}. Similar to [6], the slot time duration T_s and T_c become:

$$\begin{aligned}
T^S_s &= 2t_{PLCP} + DIFS + t^S_D + SIFS \\
&\quad + t^S_{ACK}, && (32)\\
T^F_s &= 2t_{PLCP} + DIFS + t^F_D + SIFS \\
&\quad + t^F_{ACK}, && (33)\\
T^S_c &= t_{PLCP} + DIFS + t^S_D, && (34)\\
T^F_c &= t_{PLCP} + DIFS + t^F_D, && (35)
\end{aligned}$$

where the index t^S_D (resp. t^F_D) denotes the time needed to transmit a packet of length $H + P_S$ (resp. $P_F + H$) at rate S (resp. F).

3 Measurements

We set up the platform depicted in Figure 3 with three notebooks (*Duke*, *Viking*, *Soleil*) sharing a 802.11b wireless infrastructure. The three notebooks are running Linux RedHat 8.0 (Kernel 2.4.18) with Netgear MA401 wireless cards based on the Intersil Prism II chipset. Since Poisson arrival model is not a good model for TCP sources, we use the UDP traffic generator *mgen* [12] while varying payload size as well as the data sending rate. Each measurement is done over 40 seconds and repeated five times. The traffic is always directed from the mobile host towards our server *Spoutnik*. In order to get a better insight into the *performance anomaly* studied in our paper and described in [3], we conduct the experiments with STA *Duke* running at 1 Mb/s and the others at 11Mb/s. We actually change the linux-wlan driver for *Duke* such that only the physical transmission rate 1Mb/s is supported. The two other STAs, *Viking* and *Soleil*, are forced to have a physical rate of 11 Mb/s. All three notebooks are placed within two meters from the AP and are not in movement. The system parameters of the IEEE 802.11b protocol are summarized in Table 1. Further, we do not use the RTS/CTS option in our testbed. We validate our model based on two experiments, which we discuss in the next subsections.

Table 1
802.11b Protocol parameters and header definitions

t_{PLCP}	194 μs	ACK	14 bytes
DIFS	50 μs	MAC_{hdr}	34 bytes
SIFS	10 μs	IP_{hdr}	20 bytes
σ_0	20 μs	$TRANSPORT_{hdr}$	8 bytes

3.1 The slow STA with different data rates

For this experiment both fast STAs are saturated with a high rate UDP traffic. Further, all three STAs use a payload size of 1470 bytes. *Duke*, which has a physical rate of 1Mb/s, changes its data sending rate from 50 to 750Kb/s. As the two fast STAs should perform

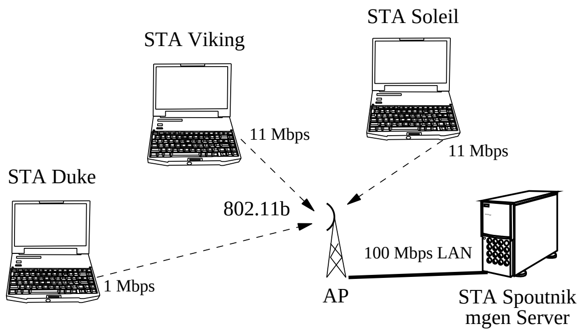

Fig. 3. Experiment setup in our 802.11b testbed

equally, we only measure the throughput of *Viking* and *Duke*. The model and experimental results are compared in Figure 4. We also plot in Figure 4 the 95 percent confidence intervals of the experimental results. As the data sending rate of the slow STA goes above 670 Kb/s, all three STAs have the same throughput. We call this regime *saturated* because all three STAs generate saturated traffic. In saturation, the 802.11 access method guarantees equal access probability for all STAs. Therefore, all τ_k are equal and consequently the throughput of each STA computed with (30) returns the same value. This phenomenon has also been observed in [3].

3.2 The slow STA with different payload sizes

Similar to the previous experiment, *Viking* and *Soleil* generate saturated UDP traffic with a bit rate of 11Mb/s. This time we limit the data rate of STA *Duke*. With the *mgen* client running on *Duke*, we generate Poisson traffic with an average rate of 320Kb/s. The payload size of the two fast STAs is fixed to 1470 bytes. We change the payload size for slow STA *Duke* and plot the resulting throughput for *Duke* and the fast STA *Viking*. Figure 5 shows the good match between experiment and model as well as the 95 percent confidence intervals of the experimental results. It can be seen that *Duke* does not attain the throughput of 320Kb/s until its payload size becomes larger than 300 bytes. We call the regime below this value *saturated*. In this regime, the number of packets to send is so high that the sending queue of *Duke* is always full, and so *Duke* attains its saturation throughput for these particular payload sizes. In the saturated regime, increasing the payload size of the slow STA increases the degradation of the throughput of the fast STA, because the channel occupation time of the slow STA increases too. Above 300 bytes, the slow STA is not saturated anymore and the fast STA can continuously improve its throughput. This is because above 300 bytes the number of packets transmitted by the slow STA decreases, consequently the fast STAs have more chances to access the channel.

4 Solutions for 802.11 performance anomaly

Motivated by the results of our previous experiments, we propose and evaluate in this section two different mechanisms to gain control over the throughput degradation which occurs in a multirate environment. In Figure 4 it can be seen that avoiding data rates of the slow STAs above a certain rate can help a lot to improve the throughput of the fast STAs running at 11Mb/s. In our first mechanism we propose to change the minimum CW of the slow STAs in order to lower their saturation throughput.

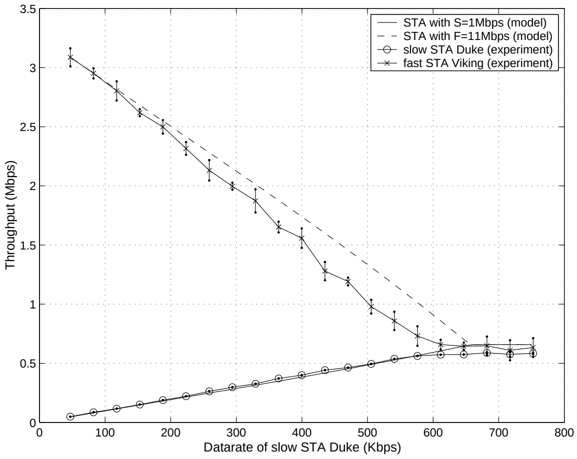

Fig. 4. Throughput comparison between model and experiment when we vary the data rates. The saturation throughput of *Viking* is slightly overestimated in our model, which is mainly caused by the perfect medium assumption.

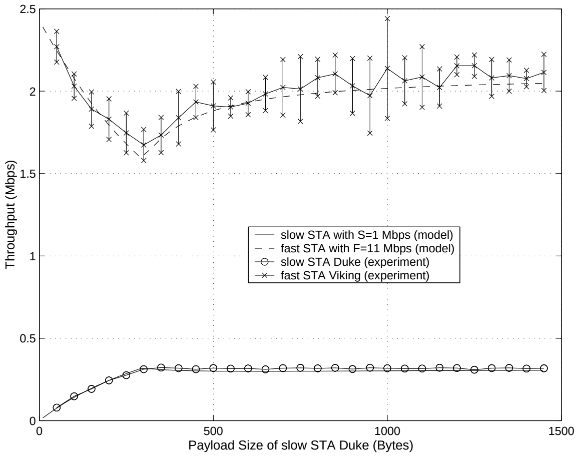

Fig. 5. Throughput for a fast and a slow STA. The fast STAs are saturated with a fixed payload size of 1470 bytes. The data rate for the slow STA Duke is limited to 320Kb/s.

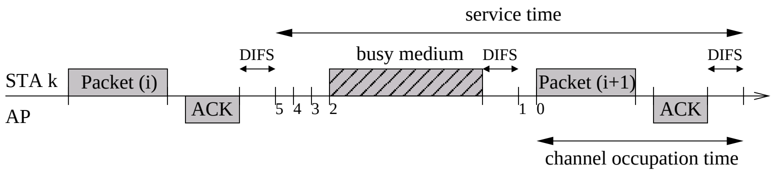

Fig. 6. Service time and channel occupation time.

In the second mechanism, we propose to reduce the packet size of the slow STAs, which also increases the throughput of the fast STAs as shown in Figure 5. Our mechanisms are not supposed to maximize the total throughput, which could actually be done by turning off the slow STAs. Our objective is to attain fairness between slow and fast STAs, which is achieved when all STAs use the medium equally long in the time domain. Since we are interested in providing short term rather than long term fairness, we only consider the saturated regime for both types of STAs. To do so, for each STA k we define the ratio of time it is actually using the medium as the ratio between its channel occupation time $T_{s,k}$ and its average service time $E[S_{T,k}]$ (see Figure 6). With the vector $\overline{x}$ we denote the time allocation, where the k^{th} element of $\overline{x}$ is:

$$x_k = \frac{T_{s,k}}{E[S_{T,k}]} = \frac{T_{s,k}}{E[T_{bo,k}]}. \tag{36}$$

The second equality in (36) is true because in the saturated regime, the probability q of having no packet in the MAC buffer becomes 0, hence (18) equals (19). Now, we use Jain's fairness index [9] to evaluate how fair a particular allocation $\overline{x}$ is. We call this fairness index F_J and it is defined as:

$$F_J(\overline{x}) = \frac{\left[\sum_{k=1}^{n} x_k\right]^2}{n \sum_{k=1}^{n} x_k^2}, \tag{37}$$

where n is the total number of STAs. In [9] it is shown that $F_J \leq 1$, and that the equality holds if, and only if, all n x_k have the same value. Therefore, our mechanisms aim at finding the value of the minimum CW (subsection 4.1) and the packet size (subsection 4.2) that maximizes the fairness index F_J.

4.1 Mechanism 1: Fair value for CW_{min}

We can limit the throughput of an STA by lowering its priority for the medium access. This can be accomplished by either increasing the minimum CW (W_0) or by increasing the $DIFS$ time, which is similar to what is proposed in the new standard 802.11e [2] for the maintenance of Quality of Service (QoS). In our work, we are interested in finding the value of W_0 that maximizes the fairness index defined in (37). We investigate the fairness in ($n_S = a, n_F = b$) networks, with the rate of the slow STAs fixed at 1Mb/s, that of the fast STAs is fixed at 11Mb/s. The parameters a and b are the scale factors for the number of slow and fast STAs.

Table 2
Optimal Values for W_0 (W_{opt})

PHY Rate (Mb/s)	W_{opt} **Value**
S=1, F=11	242
S=2, F=11	120
S=5.5, F=11	51

In Figure 7, we fix n_S to 1 and we take two values for n_F: 1 and 10. We apply our analytical model to plot the throughput of the slow STA and of one fast STA versus the W_0 value of the slow STA. Note that we only change W_0 of the slow STA and do not limit its maximum CW (W_{max}). Therefore, W_{max} changes dynamically and its value is computed as $W_0 \cdot 2^m$. The minimum CW of the other n_F fast STAs is kept at its default value of 32. It can be seen that the throughput of the fast STAs considerably improves by increasing the value of W_0 of the slow STA.

In Figure 8 we plot the fairness index defined in (37) versus the value of W_0 of the slow STA. We show the curves of F_J for different sizes of networks. Fairness is achieved by setting W_0 to 242, independently from the number of slow and fast STAs! This value of 242 for the optimal minimum CW (W_{opt}) does not apply in cases where the rate S is equal to 2 or 5.5Mb/s or F is not equal to 11Mb/s. The values of W_{opt} for different physical rates are summarized in Table 2. For instance these values hold when two physical rates exist. Later we will consider some scenarios where three physical rates exist and show that these values of W_0 in Table 2 remain the optimal to be used.

To support this mechanism, we propose that the optimal minimum CW for each STA is computed by the AP, which is aware of the modulation rates of the individual STAs. These optimal values can then be broadcasted with a beacon frame, where each STA can find its mapping for the correct minimum CW to use.

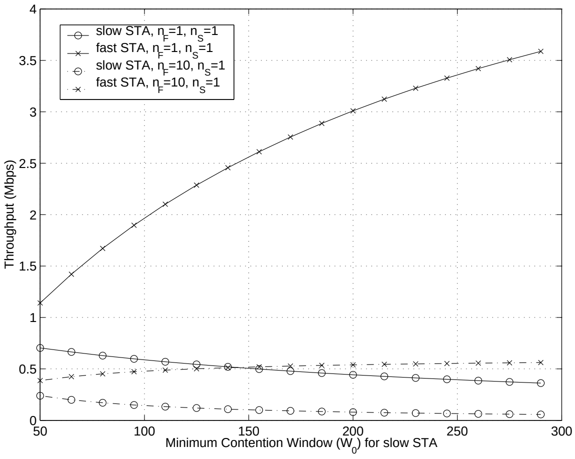

Fig. 7. Saturation throughput for the fast and slow STAs with different W_0 of the slow STA. All STAs have a packet size of 1470 bytes. W_0 of the fast STA is fixed to 32.

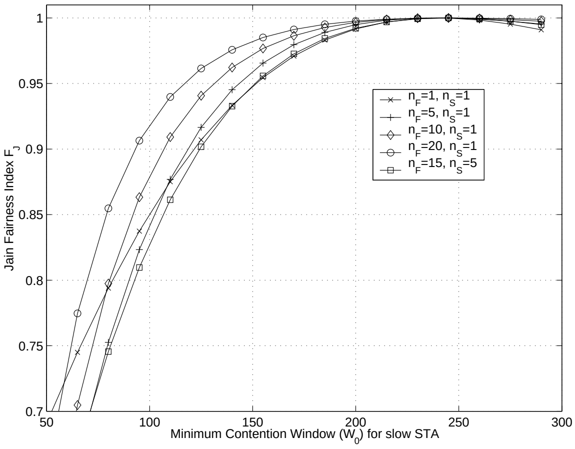

Fig. 8. Fairness of time allocation for different W_0 of the slow STA. W_{max} of the slow STA is dynamic. W_0 of the fast STAs is fixed to 32. All STAs have a payload size of 1470 bytes and are saturated.

The same idea could have been implemented by using a fixed W_{max}. We repeat the same computation, limiting the maximum CW for all STAs to 1024. Figure 9 shows the results. As expected W_{opt} is no longer a single value, it is now sensitive to the number of STAs. An implementation of this case requires continuous adaptation of W_0 to the current network size and, therefore, increases the complexity. So, we give the preference to the implementation using a dynamic W_{max} and do not study the second implementation further.

4.2 Mechanism 2: Fair value for P_S

A more intuitive way to attain fairness is to change the payload size of an STA according to its current physical transmission rate. We study it here in our framework for the fairness in time allocation. In saturation, the service time for all STAs is the same, assuming that all STAs use the same MAC parameters (not including the packet size). From (36) it can be seen that the problem of finding the fair allocation is then reduced to the problem of making all $T_{s,k}$ equal. In this work we take a reference physical rate of 11Mb/s, consequently the ref-

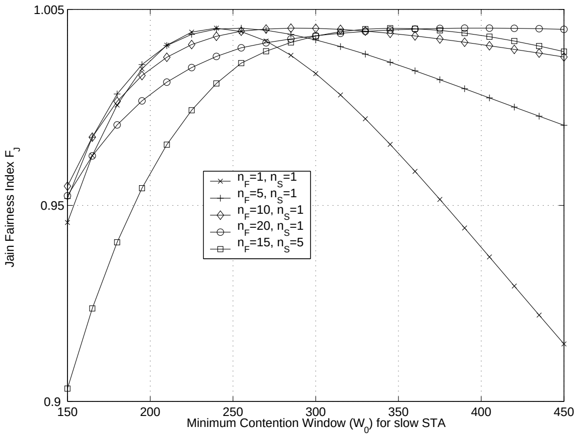

Fig. 9. Fairness of time allocation for different W_0 of the slow STA. W_{max} is static fixed to 1024 for all STAs.

erence transmission time becomes T_s^F. Fairness is achieved if all STAs have the same transmission time T_s^F. Therefore, the slow STAs have to reduce their payload such that T_s^S equals T_s^F. As we compute the throughput at the application level, we also define the parameters to obtain fairness at this level. The optimal payload size can be obtained by setting (32) and (33) equal and determining P_S.

We denote with P_{opt} the payload size for which the maximum fairness ($F_J = 1$) is achieved. P_{opt}, expressed in payload bits, is equal to:

$$P_{opt} = \frac{S \cdot P_F - (F - S)(H + ACK)}{F}. \quad (38)$$

H denotes the packet header and is defined in (31). ACK denotes the length in bits of the acknowledgment sent by the AP. Note that the optimal payload size P_{opt} for the slow STAs yields fairness for any configuration of the network, as long as at least one STA operates at the maximum rate F. This mechanism could be implemented in the MAC layer of each node. If a node's physical transmission rate drops to either 5.5, 2, or 1Mb/s, it adapts its MTU size according to its rate: $P_{opt}^{MTU} = P_{opt} + IP_{hdr} + TRANSPORT_{hdr}$.

Table 3
Values for P_{opt} and P_{opt}^{MTU} in bytes

PHY Rates (Mb/s)	P_{opt}	P_{opt}^{MTU}
S=1, F=11	65	93
S=2, F=11	205	233
S=5.5, F=11	697	725

Table 3 shows the optimal MTU values for different configurations. The disadvantage of this method is that it causes strong fragmentation at the higher layers, thereby increasing the overhead even further. Fragmentation overhead could be avoided if one could inform the application layer to generate packets respecting the optimal MTU size. We propose to use the Path MTU (PMTU) discovery technique as described in [10]. However, in our particular case, the MAC layer has to act like a router. For example, if TCP is used as end-to-end protocol, when the MAC receives a datagram that exceeds the optimal MTU size, it can return an *ICMP Destination Unreachable* message to the TCP source, with the code indicating "*fragmentation needed and DF set*" [11], and with the optimal MTU size to use. If UDP is used, it is still possible to use such a technique but only if the application can be modified to respond to such an ICMP packet.

4.3 Discussion

We analyze mechanisms 1 and 2 in terms of their total throughput. We also present a simple comparison based on packet delay. We consider a network where we have one slow STA at 1 Mb/s ($n_S = 1$) and a varying number of fast STAs at 11 Mb/s. In Figure 10, the total throughput of mechanisms 1 and 2 is compared to the basic configuration. We also compare the performance of the two mechanisms to the total throughput obtained by the basic configuration when all STAs are fast. With basic configuration we mean that all STAs have the same packet size and the same value of W_0. Both mechanisms clearly outperform the basic configuration in the multirate environment and they are close to the ideal case where all STAs are fast. This clear improvement of the total throughput is achieved because the slow STA has been *punished* by either lowering its transmission probability or by lowering its payload size. Further, it can be seen that the two proposed mechanisms have almost the same total throughput. We notice that once the number of participating STAs is larger than ten, collisions occur more often and the total throughput starts declining. This phenomenon could be avoided by using RTS/CTS instead of the basic access mechanism, which would help to attain an almost constant total throughput over the number of STAs.

Next, we present a simple analysis to evaluate the increase in packet delay that the slow STA might face when using W_{opt} and P_{opt} as adaptive parameters. This analysis is not thorough, we include it to give an idea on how the two mechanisms perform in terms of packet delay [1] . To do that, we fit the service time of packets at the MAC layer to an exponential RV shifted by the minimum service time. Note that for a slow STA the minimum service time is deterministic and equal to T_s^S. We denote the service time with the RV X, which is equal to the sum of an exponential RV with expected value $1/\mu$ and T_s^S. We obtain μ from the following relation, where $E[S_T]$ is given in (18):

$$E[X] = T_s^S + \frac{1}{\mu} = E[S_T]. \qquad (39)$$

We can now compute the second moment of X and use it in the Pollaczek-Khinchin formula [8] to get the expected waiting time $E[W]$ in the $M/G/1$ buffer. The average packet delay for our system is then the sum of average service time $E[X]$ and the obtained waiting time $E[W]$. We evaluate the packet delay versus throughput of a slow STA (running at 1Mb/s) in the two different network sizes: $n_S = 1, n_F = 1$ and $n_S = 1, n_F = 5$. The trade-off of mechanism 1 (W_{opt}) against mechanism 2 (P_{opt}) is shown in Figure 11. In each case, the fast STAs generate fully saturated traffic. Setting W_0 to W_{opt} rather than using the optimal payload size mechanism, can help to achieve a much higher throughput for a slow STA without exploding the packet delay. This can be partly explained by the fact that a packet with a payload size of 65 bytes has a huge relative overhead. On the other hand, a small packet size and an equal W_0 value for all STAs lead to less delay, if the throughput of the slow STA is low. The minimum CW method is based on the idea of introducing a certain delay to the slow STAs, thereby increasing the service time. This delay is fixed for any network configuration and cannot be bypassed. Therefore, a CW of 242 introduces long waiting times even for a low loaded network. Giving preference to one of the two mechanisms is very difficult, because their performance strongly depends on how much delay can be accepted by end-to-end

[1] Under our $M/G/1$ assumption, an accurate analysis of the average packet delay requires the knowledge of the second moment of the service time of packets at the MAC layer. For instance this quantity is not available.

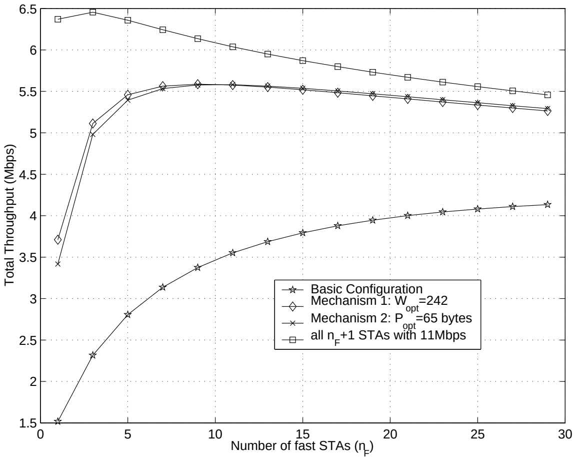

Fig. 10. Total throughput versus number of fast STAs. All STAs are saturated.

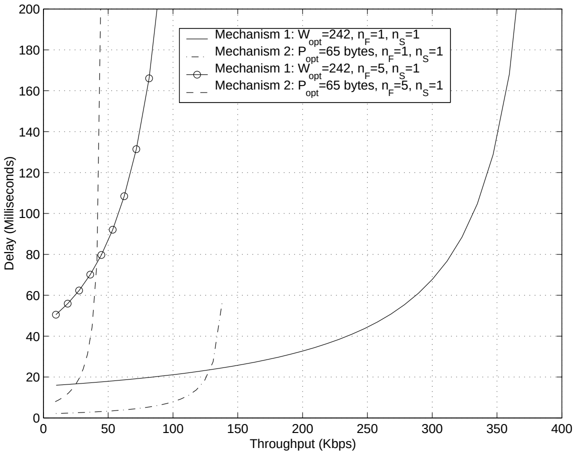

Fig. 11. MAC layer delay for the slow STA using either Mechanisms 1 or 2. The fast STAs are saturated and use a payload size of 1470 bytes.

protocols.

5 Extension to three modes

In this section we use an extended version of our model to investigate fairness in the case of having three different bit-rates. For lack of space we only present the numerical results, extending our model to three bit rates or more is a simple exercise. Our main purpose in this section is to prove that the optimal values we found for the minimum CW in case of two bit rates (Table 2) still hold in the three bit rates environment. We do not address the optimal payload size since by definition it is optimal for any number of bit rates and for any number of STAs, at a condition that the reference point remains always the fast STA at 11 Mb/s, otherwise the values in Table 3 have to be recomputed.

Equivalent to n_S and n_F we introduce n_M to count the number of STAs using a physical rate of M Mb/s. The total number of STAs is still denoted by n, where now $n = n_S + n_F + n_M$. We fix for our simulations S at 1Mb/s, M at 2Mb/s and F at 11Mb/s. Similar to Section 4.1, we keep the minimum CW (W_0) of the fast STAs fixed at 32 and we change W_0 of the STAs

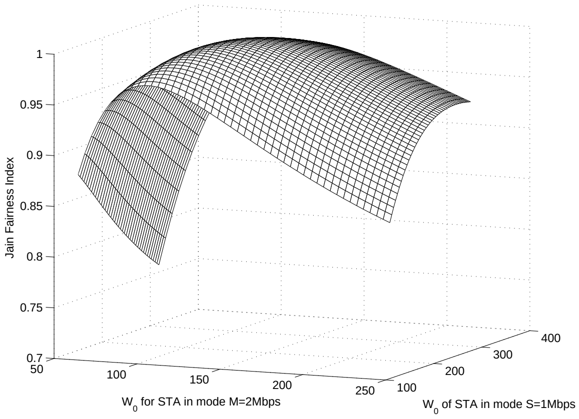

Fig. 12. Fairness of time allocation for different W_0 of STAs in modes S and M. W_0 of the fast STAs is fixed to 32. All STAs have a payload size of 1470 bytes and are saturated.

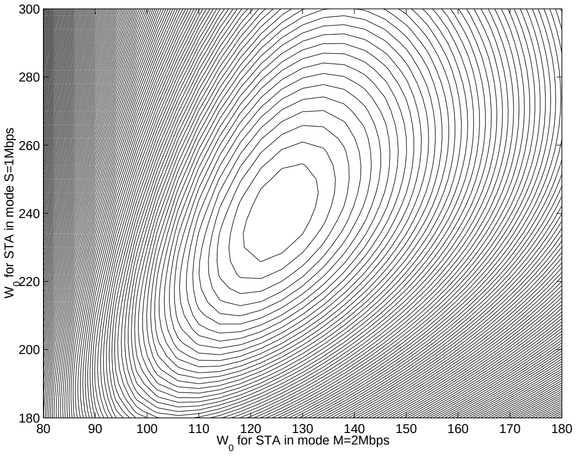

Fig. 13. Contour plot for the fairness of time allocation for different W_0 of STAs in modes S and M. W_0 of the fast STA is fixed to 32. All STAs have a payload size of 1470 bytes and are saturated ($n_S = n_M = n_F = 1$).

in mode S and M. Again, the maximum CWs are dynamically adapted to their corresponding W_0. For every pair of values for W_0 of STAs in mode S and M, we compute the fairness index defined by (37). This gives rise to the three dimensional plot shown in Figure 12. In this first simulation we take for each mode only one single STA. On the x-axis we put the minimum CW of the STA in mode M and on the y-axis the minimum CW of the STA in mode S. Observe from Figure 12 that the peak of the fairness index indicating surface is very flat. This makes it very hard to point out only one single pair of optimal minimum CWs. We relax the constraint on the optimal minimum CW and seek for minimum CW pairs returning at least a fairness index of $F_J = 0.999$. Such minimum CW pairs can easily be found by displaying equal values for F_J in a contour plot. The most inner circle in the contour plot in Figure 13 borders the set of W_0 pairs (W_0^M, W_0^S) that yield a minimum fairness index of 0.999. The spacing between two equal-potential lines is about 0.001. It can be observed that the W_0 pair (120,242) from Table 2 lies within this set satisfying our constraint on a fair channel time

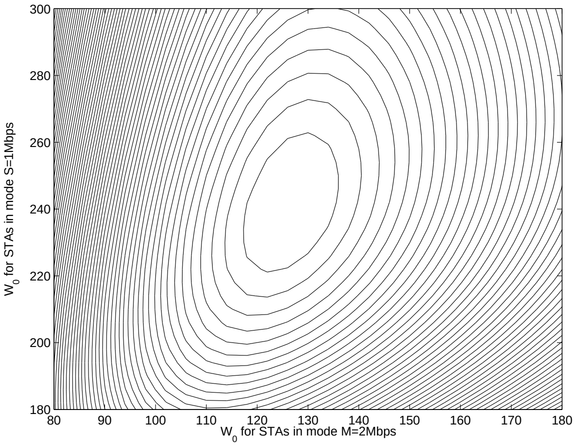

Fig. 14. Contour plot for the fairness of time allocation for different W_0 of STAs in modes S and M. W_0 of the fast STAs (mode F) is fixed to 32 ($n_S = 4$, $n_M = 6$ and $n_F = 8$).

allocation.

In a next step we want to verify the values from Table 2 for a random number of STAs. For this purpose we choose (without any particular reason) $n_S = 4$, $n_M = 6$ and $n_F = 8$. In Figure 14 we give the resulting fairness index in a contour plot. Equivalent to the contour plot from above, the most inner circle defines the set of CW pairs which yield a minimum fairness index of 0.999. Again, the W_0 pair (120,242) from Table 2 lies within this set. We notice that the decline in fairness index is less steeper that when operating only with one STA per mode. This results in a larger set of minimum CW pairs satisfying a fair channel time allocation. The same results can be observed in Figure 8 and 9 from Section 4.1.

6 Conclusions

In this paper, we presented an analytical model for finite load sources using the IEEE 802.11b DCF protocol. We used a novel approach by modelling the MAC buffer with an $M/G/1$ queue. In addition, the model is generalized such that different modulation rates can be supported simultaneously. Our model shows good performance when comparing it against real measurements from our 802.11b testbed.

Motivated by the current unsatisfying performance and fairness in a 802.11b multirate infrastructure, we invent a new fairness metric for general CSMA/CA multirate networks. This metric is used to propose two different mechanisms that meet our fairness objective and that provide a considerably better total throughput. Our proposed mechanisms are only executed on slow STAs and change either W_0 or the packet size to some pre-computed values. This procedure shows a very low complexity and therefore makes our mechanisms suitable for an implementation in the MAC protocol stack.

Appendix I

We derive here the probabilities to enter into the circuit points C_0 and C_1. Using the balance equations, c_0 can be obtained from:

$$
\begin{aligned}
c_0 = &\sum_{i=0}^{m-1} \pi_{i,0}(1-p) + \pi_{m,0} + (1-p)\pi_{frtx} \\
= &\pi_{0,0} + \frac{c_0\ (1-p)q\ p_{notx\to frtx}\ p_{pb\to notx}}{p_{notx\to bo} + p_{notx\to frtx}}. \quad (40)
\end{aligned}
$$

Knowing c_0, c_1 becomes:

$$
\begin{aligned}
c_1 = &c_0(1-q) + \\
&\frac{c_0\ q\ p_{pb\to notx}(p\ p_{notx\to frtx} + p_{notx\to bo})}{p_{notx\to bo} + p_{notx\to frtx}}. \quad (41)
\end{aligned}
$$

Appendix II

Suppose that the value for the backoff counter and the length of each slot time is known. Then, the probability that no packet arrives during the post-backoff is equal to:

$$p_{pb\to notx|B,\overline{\sigma}} = e^{-\lambda \sum_{i=1}^{B} \sigma_i}. \quad (42)$$

B is uniformly distributed. If we condition on every possible value of B, still knowing the length of each slot time, we can write:

$$
\begin{aligned}
p_{pb\to notx|\overline{\sigma}} = &\sum_{b=0}^{W_0-1} \frac{1}{W_0} e^{-\lambda \sum_{i=1}^{b} \sigma_i} \\
= &\frac{1}{W_0} \sum_{b=0}^{W_0-1} \prod_{i=1}^{b} e^{-\lambda\sigma_i}. \quad (43)
\end{aligned}
$$

Finally, we assume that σ_i are independent and identically distributed, therefore $p_{pb\to notx}$ becomes:

$$p_{pb\to notx} = \frac{1}{W_0} \sum_{b=0}^{W_0-1} E\left[e^{-\lambda\sigma}\right]^b. \quad (44)$$

Appendix III

With the index k, we refer to the STA which is actually observing the medium without accessing it. The probability that at least one of the n_S STAs transmits a packet holds:

$$p^S_{k,otr} = 1 - \prod_{\substack{i=1 \\ i\neq k}}^{n_S} (1-\tau_i). \quad (45)$$

The probability that at least one of the n_F STAs transmits a packet holds:

$$p^F_{k,otr} = 1 - \prod_{\substack{i=n_S+1 \\ i\neq k}}^{n} (1-\tau_i). \quad (46)$$

The probability to observe a successful transmission by one of the n_S STAs, knowing that there is a transmission by one of the n_S STAs, is equal to:

$$p^S_{k,os} = \frac{1}{p^S_{k,otr}} \sum_{\substack{i=1 \\ i\neq k}}^{n_S} \tau_i \prod_{\substack{y=1 \\ y\neq i,k}}^{n} (1-\tau_y). \quad (47)$$

The probability to observe a successful transmission by one of the n_F STAs, knowing that there is a transmission by one of the n_F STAs, is equal to:

$$p^F_{k,os} = \frac{1}{p^F_{k,otr}} \sum_{\substack{i=n_S+1 \\ i\neq k}}^{n} \tau_i \prod_{\substack{y=1 \\ y\neq i,k}}^{n} (1-\tau_y). \quad (48)$$

Appendix IV

We know that we have at instant 0 no packet in the queue. Therefore, the time of the next arrival is distributed according to an exponential RV. With t we denote the time of the packet arrival, and so the remaining time in the interval of interest $[0, X]$ is equal to $X - t$ and denoted by *residual time*. The average residual time has to account for every possible arrival time and therefore $R(X)$ can be derived as follows:

$$R(X) = \int_0^X \lambda e^{-\lambda t} (X-t)\, dt = X + \frac{e^{-\lambda X}}{\lambda} - \frac{1}{\lambda}. \quad (49)$$

References

[1] IEEE Std 802.11-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 1999.

[2] IEEE 802.11 WG, "Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Quality of Service (QoS)", *IEEE 802.11e/Draft 4.1,* February 2003.

[3] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "Performance Anomaly of 802.11b", *IEEE Infocom 2003*, March 2003.

[4] F. A.-Shabdiz, S. Subramaniam, "A Finite Load Analytical Model for the IEEE 802.11 Distributed Coordination Function MAC", *WiOpt'03 workshop*, INRIA Sophia Antipolis, March 2003.

[5] C. H. Foh, M. Zukerman, "Performance Analysis of the IEEE 802.11 MAC Protocol", *Proceedings of the EW 2002 Conference*, Florence, Italy pp. 184-190, February 2002.

[6] Giuseppe Bianchi, "Performance Anaylsis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communications*, Vol. 18, Number 3, March 2000.

[7] L. Kleinrock, "Queueing Systems, Vol. I: Theroy", *John Wiley*, 1975.

[8] D. Bertsekas, R. Gallager, "Data Networks, second edition", *Prentice-Hall International Editions*, 1987.

[9] R. Jain, D. Chiu, W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", *DEC Report*, DEC-TR-301, September 1984.

[10] J. Mogul, S. Deering, "Path MTU Discovery", *RFC 1191,* November 1990.

[11] J. Postel, "Internet Control Message Protocol", *RFC 792,* SRI Network Information Center, September 1981.

[12] The MGEN (Multi-Generator Toolset),

http://manimac.itd.nrl.navy.mil/MGEN/.

Gion Reto Cantieni is working towards his masters and engineering degree at EPFL Lausanne, Switzerland. He will get his degree in communications systems by March 2005. During the academic year 2001-2002, he was visiting Carnegie Mellon in Pittsburgh in the cadre of an exchange program, where his emphasis was on Wireless and Optical Networks. In his masters at EPFL, he is mainly focusing on network security, distributed (information) systems and different QoS aspects in IP Backbone Networks. In summer 2003, he visited the Plante group at INRIA Sophia Antipolis, France, as an intern and in summer 2004, he is joining Intel research Cambridge for his masters thesis.

E-mail: Gionreto.Cantieni@epfl.ch

Qiang Ni received the B.S., M.S. and Ph.D. degrees from Hua Zhong University of Science and Technology (HUST), Wuhan City, China in 1993, 1996 and 1999 respectively. From 1999 to 2001, he was a postdoctoral research fellow in the multimedia and wireless communication laboratory, HUST. He did his research internship at wireless and networking group of Microsoft Research Asia during the year of 2000. In 2001, he joined INRIA, where he was an expert research engineer at Plante group, France. He is currently a senior research scientist at Hamilton Institute, National University of Ireland Maynooth. Since 2002 he has been active as a voting member for the IEEE 802.11 wireless LAN Standard Working Group. He has served as Technical Program Committee member for the IEEE Vehicular Technology Conference, Integrated Heterogeneous Wireless Networks Symposium, 2003. His current research interests include communication protocol design and analysis for wireless LAN/PAN/MAN and 4G networks, vertical handover and mobility management in mobile networks, and multimedia transmission over hybrid wired/wireless networks.

E-mail: Qiang.Ni@ieee.org

Chadi Barakat is a permanent research scientist in the Plante research group at INRIA - Sophia Antipolis since March 2002. He got his Electrical and Electronics engineering degree from the Lebanese University of Beirut in 1997, and his master and Ph.D. degrees in Networking from the University of Nice - Sophia Antipolis in 1998 and 2001. His Ph.D. has been done in the Mistral group at INRIA - Sophia Antipolis. From April 2001 to March 2002, he was with the LCA department at EPFL-Lausanne for a post-doctoral position, and from March to August 2004 he was a visiting faculty member at Intel Research Cambridge. Chadi Barakat was the general chair of PAM 2004 and serves in the program committees of many international conferences as Infocom, PAM, WONS, ASWN and Globecom. His main research interests are congestion and error control in computer networks, the TCP protocol, voice over IP, wireless LANs, Internet measurement and traffic analysis, and performance evaluation of communication protocols.

E-mail: Chadi.Barakat@sophia.inria.fr

Thierry Turletti received the M.S. (1990) and the Ph.D. (1995) degrees in computer science both from the University of Nice - Sophia Antipolis, France. He has done his PhD studies in the RODEO group at INRIA Sophia Antipolis. During the year 1995-96, he was a postdoctoral fellow in the Telemedia, Networks and Systems group at LCS, MIT. He is currently a research scientist at the Plante group at INRIA Sophia Antipolis. His research interests include multimedia applications, congestion control and wireless networking. Dr. Turletti currently serves on the Editorial Board of *Wireless Communications and Mobile Computing.*

E-mail: Thierry.Turletti@sophia.inria.fr