Timestepped Stochastic Simulation of 802.11 WLANs*

Arunchandar Vasan and A. Udaya Shankar

Department of Computer Science and UMIACS

University of Maryland, College Park, MD 20742 USA

Email: {arun,shankar}@cs.umd.edu

Abstract—We present Timestepped Stochastic Simulation (TSS) for 802.11 WLANs. TSS overcomes scalability problems of packet-level simulation by generating a sample path of the system state $\mathbf{S}(t)$ at time $t = \delta, 2\delta, \cdots$, rather than at each packet transmission. In each timestep $[t, t + \delta]$, the distribution $\Pr(\mathbf{S}(t + \delta) \mid \mathbf{S}(t))$ is obtained analytically, and $\mathbf{S}(t + \delta)$ is sampled from it.

Our method computes sample paths of instantaneous goodput $N_i(t)$ for all stations i in a WLAN over timesteps of length δ. For accurate modeling of higher layer protocols, δ should be lesser than their control timescales (e.g., TCP's RTT). At typical values of δ (e.g, 50ms), $N_i(t)$'s are correlated across both timesteps (e.g., a station with high contention window has low goodput for several timesteps) and stations (since they share the same media). To model these correlations, we obtain, jointly with the $N_i(t)$'s, sample paths of the WLAN's DCF state, which consists of a contention window and a backoff counter at each station.

Comparisons with packet level simulations show that TSS for WLANs is accurate and yields an improvement in simulation runtime of up to two orders of magnitude. Our transient analysis of 802.11 complements prior literature and also yields: (1) the distribution of the instantaneous aggregate goodput; (2) the distribution of instantaneous goodput of a tagged station conditioned on its MAC state; and (3) quantification of short-term goodput unfairness.

I. INTRODUCTION

Packet-level simulation (PLS) of computer networks becomes prohibitively expensive as link speeds, workloads, and network size increase. Timestepped Stochastic Simulation (TSS) is a method to achieve the modeling accuracy of PLS at a fraction of the computational cost. TSS generates **sample paths** of the network state, just as in packet-level simulation, but only at increments of discrete timesteps rather than at every packet transmission. If $\mathbf{S}(t)$ represents the network state at time t, TSS generates an evolution of $\mathbf{S}(t)$ at $t = \delta, 2\delta, \cdots$, given $\mathbf{S}(0)$. In each timestep $[t, t+\delta]$, the distribution $\Pr(\mathbf{S}(t+\delta) \mid \mathbf{S}(t))$ is obtained analytically assuming that all stochastic inputs are time-invariant in $[t, t + \delta]$, and $\mathbf{S}(t + \delta)$ is sampled from it. This time-invariance assumption holds for δ less than the feedback time-scale of the end-to-end control mechanisms (i.e., TCP roundtrip time), in which case TSS generates sample paths which are very good approximations of those generated by packet-level simulation. The time-invariance requirement means that δ cannot be too large; we use $\delta = 50$ms.

TSS for networks of point-to-point links was presented in reference [12]. This paper extends TSS to 802.11 WLANs [13]. Consider an 802.11 WLAN with α stations, where each station i is either active or inactive over time, with transitions occurring only at timestep boundaries. A station that is active (inactive) at time t has (no) packets to send in its output queue throughout $[t, t + \delta]$. (The output queue is fed, in general, by state-dependent data sources, e.g., TCP.) Let $\mathbf{M}(t)$ be the vector of length α whose ith entry indicates whether station i is active, and $M(t)$ be the number of active stations. Let $N_i(t)$ be the *goodput* of station i in timestep $[t, t + \delta]$, defined as the number of packets successfully transmitted by station i in the timestep. $N_i(t)$ is zero for a station i inactive at t. For a station i active at t, $N_i(t)$ depends on all the active stations at t, as determined by the 802.11 MAC protocol. The MAC protocol we consider here is the basic 802.11 Distributed Coordination Function (DCF) protocol.

Given $\mathbf{M}(t)$, our method computes evolutions of the goodput vector $\langle N_i(t) : 1 \leq i \leq \alpha \rangle$ for $t = 0, \delta, 2\delta, \cdots$. For the timestep size δ of interest (i.e., $\delta = 50$ms), the DCF protocol introduces strong dependencies in the $N_i(t)$'s, specifically, positive correlation in $N_i(t)$ across timesteps and the negative correlation between $N_i(t)$'s across stations i in the same timestep. It is essential to capture these dependencies, otherwise the evolutions of the $N_i(t)$'s would not be an adequate foundation for simulating upper-level protocols (e.g., TCP) in a timestepped manner. Thus the key issue is the *short-term* behavior of the DCF protocol.

Our method computes evolutions of the goodput vector and DCF state jointly: at each timestep, the goodput and DCF state at the end of the timestep is obtained in terms of the goodput and DCF state at the previous timestep. We validate against PLS by comparing the resulting marginal distributions, the crosscorrelations (across stations), and the autocorrelations (across timesteps) of the per-station instantaneous goodput and DCF state. We find that TSS is quite accurate and yields runtime speed-up of up to two orders of magnitude.

Detailed comparisons are in Section VII, but a quick comparison against PLS is shown in Figures 1(a) through 1(c) for a 100s simulation where $M(t)$ is 32 at $t = 0$s and drops to 16, 8, 4 at $t = 25$s, 50s, and 75s respectively. Figure 1(a) shows two sample paths of $N_i(t)$, (one from TSS and one from PLS). Note the variation in $N_i(t)$. Figure 1(b) compares the evolutions of the $E[N_i(t)]$ and $Dev[N_i(t)]$. TSS matches PLS
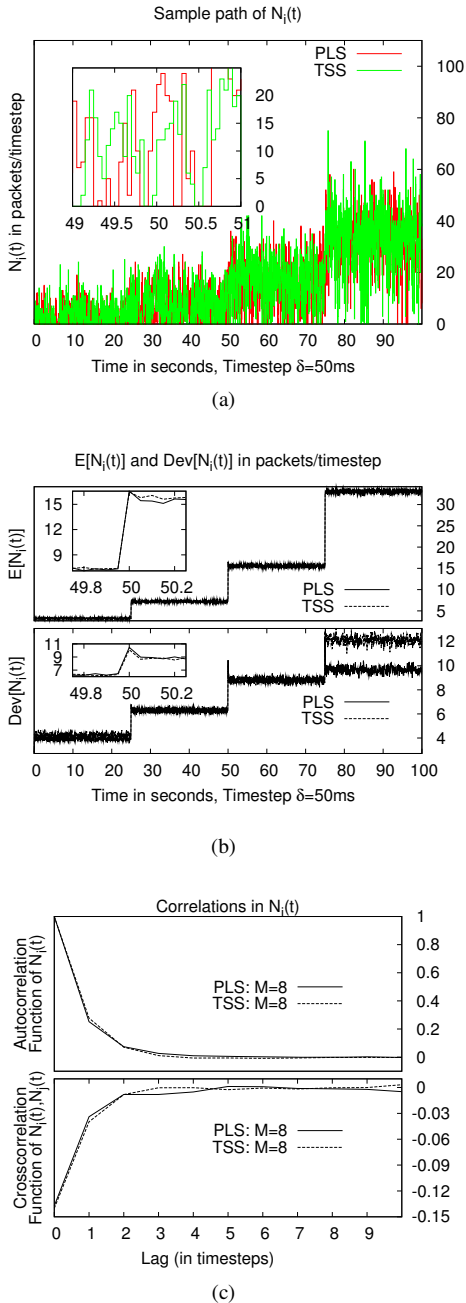
Fig. 1. Sample path of $N_i(t)$; $E[N_i(t)]$ and $Dev[N_i(t)]$; and autocorrelation function of $N_i(t)$ and crosscorrelation function for $N_i(t)$, $N_j(t)$ between $t = 50s$ and $t = 75s$ where $M(t) = 8$. The number of active stations $M(t)$ is initially 32 and is halved at $t = 25s, 50s,$ and $75s$.

well, especially at the transition points, except for $Dev[N_i(t)]$ between $t = 75$ and $t = 100$ when there are only two active stations, which we explain later. Figure 1(c) compares the autocorrelation of $N_i(t)$ and the crosscorrelation between $N_i(t)$ and $N_j(t)$ between $t = 50s$ and $t = 75s$ with 8 active stations; again, TSS matches PLS well.

A. The Timestepped Simulator

The DCF state of station i at time t consists of its **contention window** $C_i(t)$ and its **backoff counter** $B_i(t)$. It

turns out, as we explain later, that $B_i(t)$ can be generated from $C_i(t)$. Thus in each timestep $[t, t + \delta]$, TSS obtains $\langle N_i(t), C_i(t + \delta) : 1 \leq i \leq \alpha \rangle$ given $\langle N_i(t - \delta), C_i(t) : 1 \leq i \leq \alpha \rangle$ in the following steps:

- Obtain the per-station collision probability $p(t)$ as a function of the number of active stations $M(t)$ at time t (available from $\mathbf{M}(t)$).
- Obtain the distribution $\Pr(N_A(t))$ of the instantaneous aggregate goodput $N_A(t) \triangleq \sum N_i(t)$ given $p(t)$ and $M(t)$. Sample $N_A(t)$ from it.
- For each station i, obtain $\Pr(N_i(t)|C_i(t))$ by abstracting the interaction with the rest of the stations by a collision in each attempt with probability $p(t)$.
- Sample $N_i(t)$ from $\Pr(N_i(t)|C_i(t))$ for all i dependently such that the sampled $N_i(t)$'s are correlated and add up to $N_A(t)$.
- For each station i, obtain $\Pr(C_i(t+\delta)|N_i(t), C_i(t))$ and sample $C_i(t + \delta)$ from it independently.

Because all involved probability distributions can be parametrized in terms of $M(t)$ and δ, they can be precomputed or cached across TSS runs. Thus the time taken to sample from these distributions is agnostic to both transmission bitrates involved and δ. In sum, each timestep of TSS takes $O(M)$ time, which is a huge improvement over the PLS runtime of $O(M \times \delta \times bit\text{-}rate)$.

B. Our contributions and related work

To the best of our knowledge, there has been no prior work on timestepped stochastic simulation of WLANs or on the transient analysis needed to achieve this. Specific contributions include the following:

- Transient analysis of 802.11 performance, yielding a method to generate sample paths of instantaneous metrics. Prior performance analyses (e.g., [3], [6], [9], [19], [16]) obtain the average aggregate goodput.
- Distribution of the instantaneous aggregate goodput $N_A(t)$ by obtaining both the mean and variance of the aggregate goodput renewal period.
- Distribution of the instantaneous goodput of a tagged station conditioned on its MAC state. This explains and quantifies the short-term unfairness in instantaneous goodputs due to the 802.11 DCF backoff mechanism.

Prior analytical models (e.g., [3], [6], [9], [19], [16], [2]) for 802.11 performance evaluation consider long-term averages for a constant number of active stations. The general approach is to observe a tagged station between two successful transmissions and estimate the average time and number of attempts taken for the same, from which one obtains the long-run average per-station and aggregate goodput and collision probabilities.

There has been much work on timestepped simulation of point-to-point networks (reference [12] provides a list). For WLANs, reference [10] develops a fluid timestepped deterministic simulator in which the *average* aggregate goodput is the input to the fluid model, and obtains long-run metrics. It
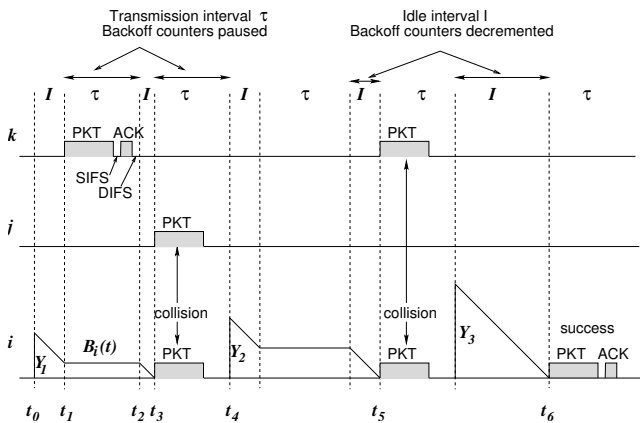
Fig. 2. Aggregate evolution of a WLAN, with evolution of tagged station i shown in more detail.

is not clear if this model can capture the short-term effects of DCF on TCP workloads.

In sum, our work complements prior work by obtaining *distributions of instantaneous metrics* and *sample-paths* that approximate packet-level simulation well.

II. 802.11 DCF OVERVIEW AND MODELING

We briefly overview the 802.11 DCF protocol using standard terminology [13]. Time is slotted. As shown in Figure 2, when station i gets a new packet to transmit (at time t_0), it sets its contention window $C_i(t)$ to the initial size γ, and sets its backoff counter $B_i(t)$ to a random value from $Uniform[0..C_i(t)-1]$. It continuously decrements $B_i(t)$ at the rate of one unit per slot while the media is idle, pauses while the media is busy, and transmits when $B_i(t)$ reaches zero. The **transmission interval** τ equals PKT+ACK+SIFS+DIFS. If the transmission is unsuccessful (i.e., ACK not received), $C_i(t)$ is doubled (binary exponential backoff). If successful, $C_i(t)$ is reset to the initial size γ. In both cases, a new value of $B_i(t)$ is chosen from $Uniform[0..C_i(t)-1]$ for the next transmission attempt. If successful transmission does not occur within β attempts, then the packet is **aborted** and the MAC state is reset.

The WLAN's aggregate evolution, obtained by superposing the per-station evolutions, is a sequence of transmission intervals alternating with variable **idle intervals I**. Note that each idle interval equals the smallest of the backoff counters at the end of the previous transmission interval. Also, a transmission interval is of size τ whether or not it is successful (as in references [4], [9], [11], [10], [13]).

A. Model assumptions and notation

Table I shows some notation used in TSS. We assume the following:

- Within any timestep $[t, t+\delta]$, each idle interval is treated as an IID copy of a stationary random variable I.
- There are no aborts, which is reasonable because the probability of abort, $p(t)^\beta$, is negligible for standard values of protocol parameters (e.g., less than 0.007 for $p(t) < 0.5$ and $\beta = 7$).

$M(t)$	number of active stations
$p(t)$	per-station (conditional) collision probability
$\lambda(t)$	per-station attempt rate
$p_A(t)$	aggregate collision probability
$N_A(t)$	aggregate goodput (random variable)
For a tagged packet:	
K	number of transmission attempts (rv)
Y_j	initial value of backoff counter for attempt j (rv)
X	total backoff duration: $Y_1 + \cdots + Y_K$ (rv)

TABLE I
NOTATION FOR TSS QUANTITIES DEFINED IN TIME INTERVAL $[t, t+\delta]$.

- RTS/CTS is not used.
- We assume that every successful transmission is received at all stations (i.e., no hidden or exposed terminals) and that a collision results in checksum errors at all receivers (i.e., no physical layer capture).
- We assume that the transmission bitrate and packet size are constant.

Define the **backoff timeline** to be the sequence of all idle intervals ordered by their occurrence time, i.e., the transmission intervals in the real timeline are collapsed to points. Note that an interval of δ slots in the real timeline would on average have $\delta E[I]/(E[I]+\tau)$ idle interval slots. We show (in Section III-D) that deviations from the average are rare. So we assume the following to simplify the analysis:

- Any interval of length δ slots in the real timeline contracts (corresponds) to an interval of length $\eta\delta$ slots in the backoff timeline, where $\eta \triangleq E[I]/(E[I]+\tau)$.

III. THE DISTRIBUTION $\Pr(N_A(t))$

The distribution of the instantaneous aggregate goodput $N_A(t)$ is obtained by extending the analysis in reference [2] with second moments and the central limit theorem for renewal processes. Following [2], we get the per-station collision probability and attempt rate, from which we get the aggregate attempt rate and collision probability, from which we get the distributions of the idle interval, the transmission renewal period, and the goodput renewal period. Applying the central limit theorem yields the desired goodput and throughput distributions. From the latter distribution, we justify the real-to-backoff-timeline contraction approximation.

A. Analysis of per-station attempt process in backoff timeline

The attempt process of a station i in the backoff timeline is a sequence of "blocks", where each block corresponds to the lifetime of a tagged packet. Specifically, each block is a sequence of intervals of the form $\langle Y_1, Y_2 \cdots Y_K \rangle$, where K is the number of transmission attempts for the packet, Y_j is the initial backoff counter for attempt j, there is a transmission after each Y_j, and the transmission after Y_K alone is successful. The **total backoff duration X** for the tagged packet is given by $Y_1 + Y_2 + \cdots + Y_K$.

For each timestep $[t, t+\delta]$, the per-station collision probability $p(t)$ can be obtained given $M(t)$, either by a fixed point iteration [2] or empirically from packet-level simulation

[18]. K is a truncated geometric random variable with failure probability $p(t)$. Each Y_j is uniformly distributed over $[0, \cdots, \gamma 2^{j-1} - 1]$. Thus $E[K]$ and $E[X]$ can be obtained.

In the backoff timeline, the per-station attempt process is a markovian renewal process with average overall cycle (renewal) period $E[X]$ and average number of attempts in a cycle $E[K]$. By the renewal reward theorem [15], the attempt rate $\lambda(t)$ is given by $E[K]/E[X]$, which is thus available given $M(t)$.

B. Analysis of aggregate attempt process in backoff timeline

The aggregate attempt process is the superposition of the per-station attempt processes. In the backoff timeline, the aggregate attempt process is a sequence of blocks, where each block is a sequence of intervals of the form $\langle I_1, I_2 \cdots I_L \rangle$, where each I_i is an IID copy of the idle interval I, a single station transmits successfully after I_L, and two or more stations transmit unsuccessfully after each I_i for $i \neq L$.

The per-station attempt processes are assumed to evolve independently in the backoff timeline, with each being subject to the per-station collision probability $p(t)$. Then the probability that there is at least one transmission in a slot of the aggregate attempt process is $1 - (1 - \lambda(t))^{M(t)}$, where $\lambda(t)$ is the per-station attempt rate obtained above. So the idle interval I is a geometric random variable with success probability $1 - (1 - \lambda(t))^{M(t)}$. The aggregate collision probability $p_A(t)$ is the probability of two or more transmissions in a slot given at least one transmission. Assuming that collisions are independent, L is geometric with success probability $1 - p_A$.

C. Analysis of aggregate attempt process in real timeline

The aggregate attempt process in the real timeline is a sequence of blocks, where each block is a sequence of intervals of the form $\langle T_1, T_2, \cdots, T_L \rangle$, where each T_i is an IID copy of $T = I + \tau$ and only T_L is successful (as before). Thus the aggregate attempt process is a renewal process with period T. With $E[T] = E[I] + \tau$ and $Var[T] = Var[I]$, by the central limit theorem for renewal processes [15], we have:

The aggregate throughput (i.e., number of T intervals) in $[t, t + \delta]$ is normally distributed with mean $\delta/E[T]$ and variance $\delta Var[T]/E[T]^3$.

The aggregate goodput process is a renewal process with period $G = T_1 + T_2 + \cdots + T_L$. We have $E[G] = E[E[G|L]] = E[T].E[L]$, which yields the mean goodput in $[t, t + \delta]$ as $\delta/E[G]$ (all prior works (e.g., [2], [3], [9]) stop here). Also $Var[G]$ can be shown to be $E[L]Var[T] + Var[L]E[T]^2$, where $E[L]$ and $Var[L]$ can be obtained from the distribution of L. Thus we have:

The aggregate goodput $N_A(t)$ is normally distributed with mean $\delta/E[G]$ and variance $\delta Var[G]/E[G]^3$.
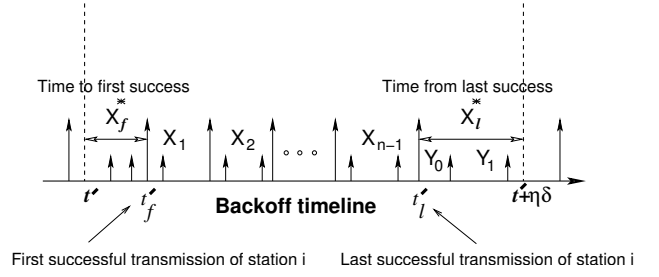

Fig. 3. Transmissions of a tagged station in the backoff timeline interval $[t', t' + \eta\delta]$ corresponding to the real timeline interval $[t, t + \delta]$. A longer arrows indicates a successful transmission, a shorter arrow failure.

D. Real-to-backoff-timeline contraction approximation

Because each throughput renewal T equals $I + \tau$ and I is geometrically distributed, the total idle interval in $[t, t + \delta]$ is the sum of a (normally distributed) random number of geometric random variables. One can compute the mean and variance of this compound random variable and approximate this by a normal distribution. However, its deviation is very small relative to its mean (less than 8% for $\delta = 50$ms). Thus it can be approximated by a constant equal to its mean.

IV. THE DISTRIBUTION $\Pr(N_i(t)|C_i(t))$

To obtain the distribution of $\Pr(N_i(t)|C_i(t))$, we consider the per-station attempt process for station i in the backoff timeline. We first analyze the case that the station begins its first attempt for some packet exactly at t.

A. Obtaining $\Pr(N_i(t)|C_i(t) = \gamma, B_i(t) = Uniform[0..\gamma-1])$

Recall that X is the total backoff duration for one successful transmission. There are n successful transmissions in $[t, t + \delta]$ the sum of n IID copies of X is less than the aggregate idle interval $\eta\delta$ and the $n+1$th successful transmission occurs outside the interval. Let E_1 denote the event $X_1 + \cdots + X_n \leq \eta\delta$ and E_2, the event $X_1 + \cdots + X_{n+1} \leq \eta\delta$. Clearly, $E_1 \subset E_2$. Thus $Pr(N_i(t) = n) = Pr(E_1 \bigwedge \overline{E_2}) = Pr(E_1) - Pr(E_2)$. Further $Pr(E_1) = F_X^n(\eta\delta)$, where the pdf f_X^n of the distribution $\Pr(X_1 + X_2 + \cdots + X_n)$ is the n-fold convolution of f_X with itself, and F_X^n denotes the corresponding cdf. We have developed an algorithm to compute this convolution efficiently; details are available in reference [18].

B. Obtaining $\Pr(N_i(t)|C_i(t) = \gamma 2^{c-1}, B_i(t) = b)$

We now analyze the case where at time t, the station is in the middle (as determined by $B_i(t)$) of an arbitrary attempt (as determined by $C_i(t)$) for the transmission of the first packet in $[t, t + \delta]$. Let X_f^* denote the **first time to success** in the backoff timeline. Conditioned on $X_f^* = x$, $Pr(N_i(t) = n)$ is given by the probability of fitting $n - 1$ copies of X in the backoff timeline interval $\eta\delta - x$. Because the first attempt for the next packet begins at $t' + X_f^*$, this probability can be obtained exactly as in the previous case. So we want to obtain the pdf of $X_f^*|C_i(t), B_i(t)$.

Given $C_i(t) = \gamma 2^{c-1}$ and $B_i(t) = b$, the first transmission occurs at $t' + b$ in the backoff timeline interval as seen in Figure 3. The number of further attempts K (which can be zero) is a

truncated geometric distribution with failure probability $p(t)$. Recall that the Y_i's denote the backoff chosen for each of these attempts. Now $X_f = x$ iff $b + Y_{c+1} + Y_{c+2} + \cdots Y_{c+K} = x$. Each of the Y_i's is uniformly distributed in increasing intervals so $Pr(X_f|B_i(t), C_i(t))$ can be obtained. Unconditioning $Pr(N_i(t)|X_f)$ using $Pr(X_f|B_i(t), C_i(t))$ yields $Pr(N_i(t)|B_i(t), C_i(t))$.

C. Obtaining $Pr(N_i(t)|C_i(t) = \gamma 2^{c-1})$

We have so far obtained $Pr(N_i(t)|C_i(t), B_i(t))$ and we want to obtain $Pr(N_i(t)|C_i(t))$. If $C_i(t) = \gamma 2^{c-1}$, then $B_i(t)$ was chosen from the *Uniform*$[0..C_i(t)-1]$ when it was renewed. Therefore at a given t, the distribution of $B_i(t)$ is distributed according to the forward recurrence time (remaining time) distribution [15] corresponding to the distribution *Uniform*$[0..C_i(t)-1]$. Thus we have $Pr(B_i(t) = b|C_i(t) = \gamma 2^{c-1}) = \frac{2(C_i(t)-b-1)}{C_i(t)(C_i(t)-1)}$ for $b \in [0, C_i(t)-1]$. Unconditioning on $B_i(t)$ gives $Pr(N_i(t)|C_i(t))$.

D. Short-term unfairness in 802.11

Short-term unfairness in 802.11 has been the subject of much research [7], [8], [17], [20]. Reference [20] examines short-term unfairness for hidden terminals while references [8], [7] claim 802.11 is fair over intervals that are defined in terms of the number of inter-transmissions that other hosts may perform between two transmissions of a given station. Our analysis naturally yields a quantification of the short-term unfairness over arbitrary fixed intervals (δ here) even with no hidden terminals.

Consider a pair of tagged stations i and j among M active stations. Note that a difference between $C_i(t)$ and $C_j(t)$ automatically results in a difference in the means of $N_i(t), N_j(t)$. To quantify the extent of short-term unfairness in goodputs, we use Jain's fairness index J_F [14]. For two stations, $J_F(N_i, N_j)$ is defined to be $\frac{(N_i + N_j)^2}{2(N_i^2 + N_j^2)}$ and ranges in $[1/2, 1]$, where $1/2$ corresponds to lowest fairness (one station gets all the goodput while the other gets nothing) and 1 corresponds to highest fairness (both get equal goodput). Specifically, we compute $E[J_F(N_i, N_j)]$ in two different ways: 1) by approximating the jdf of $\langle N_i(t), N_j(t) \rangle$ the product of the pdf's of $N_i(t)$ and $N_j(t)$, which are identical when unconditioned; and 2) by packet level simulations (PLS) described later in Section VII. Note that the pdf of $N_i(t)$ can be obtained by unconditioning that of $N_i(t)|C_i(t)$ on $C_i(t)$. Values of $E[J_F(N_i, N_j)]$ are shown for varying M in Table II. The value predicted by the analysis matches that obtained from PLS for the unconditioned Jain's index almost exactly confirming that the short-term unfairness in 802.11 can be quantified by the distribution of the instantaneous goodput.

V. DEPENDENT SAMPLING OF $N_i(t)$'S

$N_i(t)$ is determined by two factors: 1) the initial state $C_i(t)$; and 2) interaction between stations in $[t, t+\delta]$. Factor 1, which influences the autocorrelation in a station's goodput across timesteps, is modeled by Pr($N_i|C_i$). However, Pr($N_i|C_i$)

| M | $E[J_F(N_i, N_j)]$ | |
	PLS	Analysis
4	0.94	0.95
8	0.83	0.84
16	0.73	0.74

TABLE II
SHORT-TERM UNFAIRNESS ILLUSTRATED BY $E[J_F(N_i, N_j)]$ AS OBTAINED BY PLS AND ANALYSIS FOR VARIOUS VALUES OF M.

does not capture the crosscorrelation between stations' goodputs completely because it abstracts the interaction with other stations by the average behavior through the per-station collision probability $p(t)$. So to account for factor 2, we first obtain a sample of $N_A(t)$ and obtain samples $N_i(t)$ such that $\sum N_i(t) = N_A(t)$ in a negatively correlated manner.

A. Algorithm for sampling $N_i(t)$

The algorithm needs to sample from marginal distributions of $N_i(t)|C_i(t)$ while accounting for correlation. Our basic approach is to sample from different parts of the marginal distribution depending on other stations' goodputs. The main steps in our algorithm (preliminary approaches and details are in reference [18]) are:

- Obtain a sample $N_A(t)$ of aggregate goodput.
- Obtain a random permutation of the stations' indices by a Knuth shuffle [5] and allocate goodputs to stations in order of this permutation as long per-station goodputs are lesser than $N_A(t)$ in sum.
- If sum of goodputs allocated so far is greater (lesser) than the expected goodput for all stations above (below) some tolerance, then sample goodput of next station from the lower (upper) tail of the pdf Pr($N_i|C_i$).
- If sum of goodputs allocated so far is close to the mean subject to both the upper and lower tolerances, then sample goodput of next station from the full pdf Pr($N_i|C_i$).

Let n^* be a goodput such that $Pr(N_i \leq n^*|C_i) = 1/2$. By sampling the lower (upper) tail of Pr($N_i|C_i$), we mean sampling from the distribution Pr($N_i|C_i, N_i \leq n^*$) (distribution Pr($N_i|C_i, N_i > n^*$)). The Knuth shuffle and iterative allocation of goodput to each station take $O(M)$ deterministic time. Because the pdf's can be precomputed or cached, the overall runtime is $O(M)$ and independent of the transmission bitrate; any PLS would take $O(M\delta \times \text{bit-rate})$.

VI. THE DISTRIBUTION $Pr(C_i(t+\delta)|N_i(t), C_i(t))$

We first analyze the case $N_i(t) = n, n \neq 0$. Recall that X_f^* is the time to the first success, and let X_l^* denote the time in the backoff timeline from the last success in the interval to the end of the interval. Figure 3 illustrates this.

Our goal is to obtain the distribution $Pr(X_l^*|C_i(t), N_i(t))$. Once this is done, we can obtain the distribution of $C_i(t+\delta)$ given that X_l^* slots have been spent in backing off since the last successful transmission. Now $Pr(N_i(t) = n|C_i(t))$ can be expressed in terms of X_f^* and $X_i's$. That is, $Pr(N_i(t) = n)$ is the probability that $X_f^* + X_1 + \cdots + X_{n-1} \leq \eta\delta$ and $X_f^* + X_1 + \cdots + X_n > \eta\delta$. We have already seen how to

obtain $\mathsf{Pr}(\,X_f^*|C_i(t)\,)$ in Section IV. Now X_l^* is given by $\eta\delta - (X_f^* + X_1 + \cdots X_{n-1})$. Thus by Bayes' theorem, we obtain $\mathsf{Pr}(\,X_l^*|C_i(t), N_i(t) = n\,)$.

Suppose $X_l^* = x$. This means the total backoff X_n of the $n+1$-th successful transmission is greater than x. For $C_i(t + \delta) = 2^{c-1}\gamma$ to occur after spending a backoff duration x from the last successful transmission, we want $c-1$ unsuccessful transmissions, $Y_1 + \cdots + Y_c$ to just exceed x, and $Y_1 + \cdots + Y_{c-1}$ should be less than x. Thus we can obtain the distribution of $C_i(t + \delta)|X_l^*$. Unconditioning on X_l^* yields $\mathsf{Pr}(\,C_i(t + \delta)|C_i(t), N_i(t)\,)$.

When $N_i(t) = 0$, as in Section IV, we approximate $B_i(t)$ as the forward recurrence time. Using this and a similar application of Y_i's, we obtain the distribution of $C_i(t + \delta)$ conditioned on X_f^*. Again, unconditioning on X_f^* yields the required distribution.

VII. RESULTS

Our main results are broadly along two directions: quantifying speedup and validating accuracy. Because the pdf's required for TSS are precomputed using the transient analysis, we first quantify the cost for precomputation in (memory) space and time. Then we compare the runtime improvement offered by TSS over PLS. Next, we validate 1) the transient analysis of 802.11; and 2) the overall TSS technique for WLANs.

For validation of the transient analysis of 802.11, we compare the conditional pdf's, namely, $\mathsf{Pr}(\,N_i(t)|C_i(t)\,)$ and $\mathsf{Pr}(\,C_i(t + \delta)|N_i(t), C_i(t)\,)$. For validation of TSS, we compare an "internal" (to the method) metric, namely, $C_i(t)$ and an "external" metric, namely, $N_i(t)$. Specifically, we consider:

1) the pdf of $N_i(t)$;
2) the autocorrelation function of the timeseries $N_i(0), N_i(\delta), \cdots$ that captures correlations across time;
3) the crosscorrelation function between the series $N_i(0), N_i(\delta), \cdots$ and $N_j(0), N_j(\delta), \cdots$ that captures correlations across stations.

The same three points of comparison are considered for the metric $C_i(t)$ as well. In addition, we also consider the pdf of the aggregate goodput $N_A(t)$.

A. Simulation setup

Because TSS models only the MAC layer, to insure a fair comparison of the time taken for a simulation, we have implemented a simple 802.11 MAC layer packet level simulator (PLS) instead of resorting to a full blown simulator such as ns-2 [1]. This avoids the overheads of upper layer (routing, transport) as well as lower layer (physical) events in ns-2 which TSS for 802.11 does not model. As an illustration of ns-2 overheads, a simulation run of 1000 seconds for a scenario of two constant bit rate (CBR) flows sharing one 802.11 channel takes about 4.5 seconds in our custom simulator with logging enabled, while ns-2 takes about 70 seconds with all logging disabled.
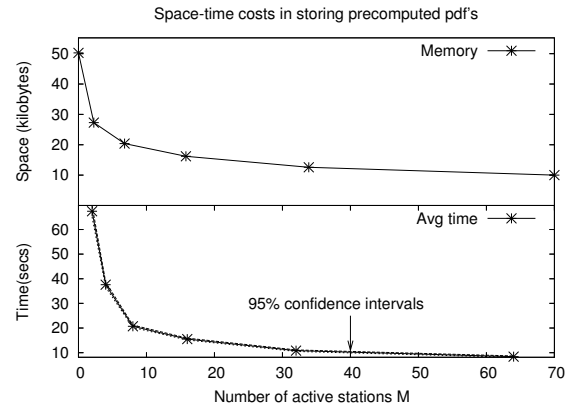

Fig. 4. Space and time costs for precomputation of $\mathsf{Pr}(\,N_i(t)|C_i(t)\,)$ and $\mathsf{Pr}(\,C_i(t + \delta)|C_i(t), N_i(t)\,)$ for $\delta = 50$ms with M varying in $[2, 4, 8, \cdots, 64]$.

All simulations were carried on a machine with a 3.2GHz Pentium-4 processor and 1.5Gb RAM running Red Hat Enterprise Linux release 3. We use a fixed packet size of 1500 bytes including the MAC-layer overhead and the 802.11a parameters: slot size of $9\mu s$, SIFS of $16\mu s$, data bitrate of 54Mbps, ACK bitrate 6Mbps, PHY-layer overhead of $20\mu s$, and contention window ranging over the 7 values $[16, 32, \cdots, 1024]$ with 7 maximum attempts. Unless otherwise mentioned, all stations always have packets to transmit, i.e., are active, during the entire run of the simulation.

B. Precomputation costs in space and time

Using the transient analysis, for each tuple $\langle M, C_i(t)\rangle$, a table of tuples of the form $\langle N_i(t), pdfval\rangle$ is obtained for $\mathsf{Pr}(\,N_i(t)|C_i(t)\,)$. Likewise, for each tuple $\langle M, C_i(t), N_i(t)\rangle$, a table of tuples of the form $\langle C_i(t+\delta), pdfval\rangle$ is obtained for $\mathsf{Pr}(\,C_i(t + \delta)|N_i(t), C_i(t)\,)$. Because $C_i(t)$ ranges over the standard seven values $[16, \cdots, 1024]$, for a fixed M, the sizes of all tables are determined by the maximum value $N_i(t)$ can take.

For a fixed M, let n_m denote the maximum value of $N_i(t)$ for which entries of tables are computed. So the tables for $\mathsf{Pr}(\,N_i(t)|C_i(t)\,)$ have $7n_m$ entries in all. Likewise, the tables for $\mathsf{Pr}(\,C_i(t + \delta)|C_i(t), N_i(t)\,)$ have $7 \times n_m \times 7 = 49n_m$ entries in all. Each entry in the table is stored as a `double` of size eight bytes. So the space required is $400n_m$ bytes. For $M = 2$, n_m is about 130, and this yields a space requirement of about 52000 bytes (in uncompressed form).

Figure 4 shows the space and time requirement for precomputing pdf's for $\delta = 50$ms with M varying in $[2, 4, 8, \cdots, 64]$. Because n_m decreases with increasing M, both the space and time required decrease with increasing M. The space requirement is almost negligible compared to memory consumed in typical packet level simulators, and the time requirement is a one-time cost shared across all runs of a simulation scenario. Nevertheless, these costs can be reduced by interpolating the pdf's among the parameters M and N_i (C_i is likely not a suitable candidate for interpolation for large M).
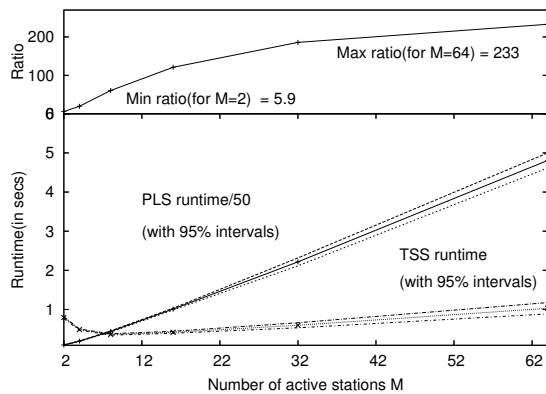
Fig. 5. Runtimes of TSS and PLS and their ratio for a 1000s simulation with M varying in $[2, 4, 8, \cdots, 64]$. Each point is an average of 100 runs. For PLS, the runtime has been scaled down by a factor of 50 to enable visual comparison with TSS. For TSS, the runtime includes the time taken to load precomputed pdf's from disk, and the time taken for precomputation is amortized over 100 runs.

Fig. 6. Comparison between empirically obtained pdf of $N_A(t)$ for $t = 5$s and $\delta = 50$ms for varying M. The deviations of $N_A(t)$ predicted by the analysis are overestimates for $M > 2$ while for $M = 2$, it is an underestimate.

Fig. 7. PDF of $N_i(t)|C_i(t)$ for various values of $C_i(t)$ and varying M.

C. Runtime comparison

TSS for WLANs provides an improvement up to two orders of magnitude in the runtime over PLS. Figure 5 shows the average time taken by both PLS and TSS and their ratio, for a 1000s simulation run with M in $[2, 4, 8, \cdots, 64]$. Each point plotted in Figure 5 and its associated 95% confidence interval is obtained from 100 runs. For PLS, the curve is shown scaled down by a factor of 50 to enable visual comparison with TSS. For TSS, the runtime includes the time taken to load precomputed pdf's from disk, and the time taken for precomputation is amortized over 100 runs. The PLS curve shows a linear increase in the runtime as expected. The TSS curve shows a dip and then an increase. This is because the amortized time to calculate precomputed pdf's is significant compared to the actual simulation loading time and runtime for smaller M; once a threshold has been crossed in M, the computational costs predominate. The trend in the TSS runtime curve for smaller M is similar to the precomputation cost curves in Figure 4.

D. The distribution $\Pr(\,N_A(t)\,)$

We obtain the distribution of the instantaneous aggregate goodput for $\delta = 50$ms through simulations and analysis. In each run of the simulation, the system is "warmed up" for 5s from a "cold start" and then a sample of the instantaneous aggregate goodput is obtained. We obtain the pdf of the instantaneous aggregate goodput from the samples of 10000 such runs and the results comparing it with analytically predicted distribution are shown in Figure 6.

We make the following observations:

- The distribution of $N_A(t)$ can be well approximated by a gaussian as predicted by the analysis.
- The means of the distributions obtained by simulation coincide almost exactly with those obtained by analysis.
- The peaks (deviations) of the normal distributions obtained by simulations are higher (lower) than those obtained by analysis; for $M = 2$ the scenario is reversed.
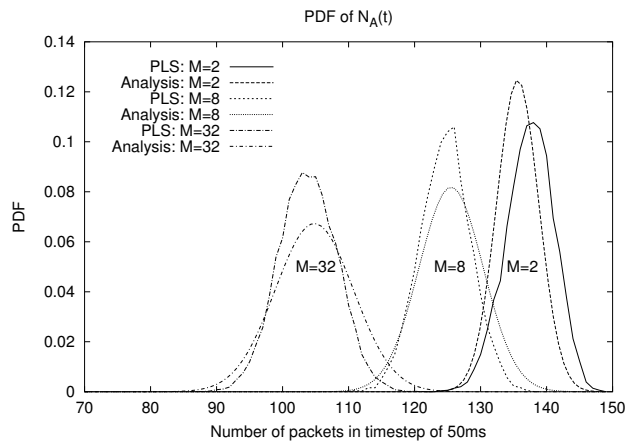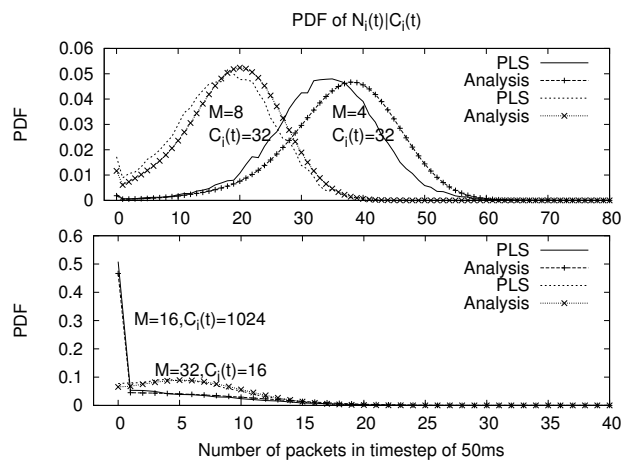
The last observation can be explained as follows. For the analysis, we had assumed that each throughput renewal in the global timeline is a failure with a fixed probability independent of the past. In reality, there are two factors that affect the variance. Specifically, the size of an idle interval is positively correlated with the event that the preceding throughput renewal(s) is a collision, and the event that a throughput renewal is a failure is negatively correlated with the event that previous throughput renewal(s) is a failure. Reference [18] provides more details.

E. $\Pr(\,N_i(t)|C_i(t)\,)$ and $\Pr(\,C_i(t + \delta)|N_i(t), C_i(t)\,)$

For each value of M, we do 100000 simulation runs with M constant throughout the simulation runs. In each simulation run, at $t = 5$s and $\delta = 50$ms, a sample of $N_i(t), C_i(t)$, and $C_i(t+\delta)$ is obtained. From 100000 samples from 100000 such runs, a frequency distribution of N_i is obtained for each fixed C_i as an estimate of the conditional probability distribution $\Pr(\,N_i|C_i\,)$. From this same set of samples, a conditional distribution of $C_i(t + \delta)$ given $C_i(t), N_i(t)$ is also obtained. This entire exercise is repeated for varying values of M.
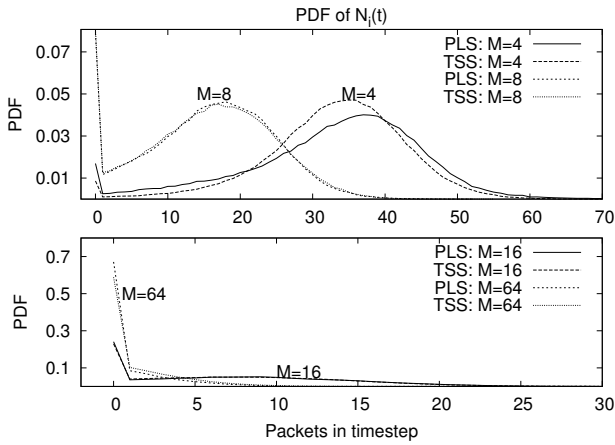
Fig. 8. Distribution of unconditional $N_i(t)$

Figure 7 shows the PDF $\Pr(N_i|C_i)$ for for varying M. The pdf's do not match exactly for small M because of our approximation in obtaining the random total backoff in an interval $[t, t+\delta]$ by a constant $\eta\delta$. However, the accuracy improves with increasing M. As M and contention window size increase, the goodput starts deviating from normal significantly with an increased probability of zero instantaneous goodput. The analysis captures this trend as can be seen in Figure 7. We also note that $Pr(N_i(t) = 0|C_i(t) = 1024)$ for $M = 16$ is much higher (around 0.45) than $Pr(N_i(t) = 0|C_i(t) = 16)$ for $M = 32$ (around 0.075) illustrating the short-term unfairness; even though the number of active stations is doubled (i.e. $M = 32$), the probability of zero goodput is much lower (than for $M = 16$) because of a favorable contention window (in this case 16). Validation of the distribution of $C_i(t+\delta)|N_i(t), C_i(t)$ can be found in reference [18].

F. The distributions $\Pr(N_i(t))$ and $\Pr(C_i(t))$

We now compare the unconditional distributions of $N_i(t)$ and $C_i(t)$. As can be seen in Figure 8, the distribution of $N_i(t)$ as obtained from TSS is very close to that obtained from PLS except for a large M (e.g., 64) where it overestimates the time with zero goodput (and underestimates the others). This is because TSS overestimates the probability of $C_i(t)$ being high for large M. A more detailed explanation of this, a comparison of the distribution of $C_i(t)$, and validation of correlations can be found in reference [18].

VIII. Conclusions and future work

We presented a transient analysis of 802.11 and used it to perform TSS for WLANs. The transient analysis of 802.11 looks at a tagged station within a timestep and obtains conditional pdf of $N_i(t)$ conditioned on the MAC state $C_i(t)$ and that of new MAC state $C_i(t + \delta)$ conditioned on $C_i(t)$ and $N_i(t)$. The TSS technique uses this transient analysis to obtain the instantaneous goodputs of all stations such that they 1) add up to the aggregate instantaneous goodput and 2) have the required correlation structure. In sum, the method obtains the sample path evolutions of the contention windows and instantaneous goodputs of all stations with time. We validate

the transient analysis and TSS technique against PLS. TSS scales well with increasing number of stations and is agnostic to bit-rate. Proposed work includes integration of the technique with traffic sources with state dependent control (e.g. TCP) and modeling the PHY layer (with arbitrary location of stations).

REFERENCES

[1] NS-2 network simulator. http://www.isi.edu/nsnam/ns.
[2] A. Kumar, E. Altman, D. Miorandi and M. Goyal. New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs. In *Proceedings of IEEE INFOCOM*, 2006.
[3] G. Bianchi. Performance analysis of the IEEE 802.11 Distributed Coordination Function. In *IEEE Journal On Selected Areas in Communications*, March 2000.
[4] D. Malone, K. Duffy, and D. Leith. Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions. *IEEE/ACM Transactions on Networking.*, 15, February 2007.
[5] D.E.Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley Publishing Company, 1981.
[6] F.Cali, M. Conti, and E. Gregori. IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement. In *Proceedings of INFOCOM*, 1998.
[7] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau. Fairness and its Impact on Delay in 802.11 Networks. In *Proceedings of IEEE GLOBECOM*, 2004.
[8] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau. Short-Term Fairness of 802.11 Networks with Several Hosts. In *Proceedings of the Sixth IFIP IEEE International Conference on Mobile and Wireless Communication Networks, MWCN*, 2004.
[9] H.Kim and J.Hou. Improving protocol capacity with model-based frame scheduling in ieee 802.11-operated wlans. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom)*, 2003.
[10] H.Kim and J.Hou. A fast simulation framework for IEEE 802.11-operated wireless LANs. In *Proceedings of the joint international conference on Measurement and modeling of computer systems ACM SIGMETRICS/PERFORMANCE*, 2004.
[11] K. Medepalli and F. A. Tobagi. Throughput analysis of ieee 802.11 wireless lans using an average cycle time approach. In *Proceedings of IEEE GLOBECOM*, 2005.
[12] A. Kochut and A.U. Shankar. Timestep Stochastic Simulation of Computer Networks using Diffusion Approximation. In *Proceedings of IEEE/ACM MASCOTS 2006, 14th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Monterey, California*, 2006.
[13] The Institute of Electrical and Electronic Engineers Inc. IEEE 802.11, 1999 edition (ISO/IEC 8802-11:1999). *http://standards.ieee.org/getieee802/802.11.html*.
[14] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. *DEC Research Report TR-301*, 1984.
[15] S. M. Ross. *Introduction to Probability Models*. Academic Press, Inc.
[16] O. Tickoo and B. Sikdar. ''On the Impact of IEEE 802.11 MAC on Traffic Characteristics''. *IEEE Journal on Selected Areas in Communications*, 21(2):189–203, February 2003.
[17] V. Ramaiyan, A. Kumar, and E. Altman. Fixed Point Analysis of Single Cell IEEE 802.11e WLANs: Uniqueness, Multistability and Throughput Differentiation. In *Proceedings of ACM SIGMETRICS*, 2006.
[18] A. Vasan and A. U. Shankar. Timestepped Stochastic Simulation of 802.11 WLANs. In *Technical Report, University of Maryland, CS-TR-4866, UMIACS-TR-2007-19*. URL: http://drum.umd.edu, 2007.
[19] Y.C.Tay and K.C. Chua. A Capacity analysis for the IEEE 802.11 MAC protocol. In *Wireless Networks*, January 2001.
[20] Z. Li, S. Nandi, and A. K. Gupta. Modeling the Short-Term Unfairness of IEEE 802.11 in Presence of Hidden Terminals. In *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, 2004.