Feasibility and Optimization of Delay Guarantees for Non-homogeneous Flows in IEEE 802.11 WLANs

Yan Gao*, Chee Wei Tan†, Ying Huang‡, Zheng Zeng*, P. R. Kumar*
*Department of Computer Science & CSL, University of Illinois at Urbana-Champaign
† Department of Computer Science, City University of Hong Kong
‡Department of Computer Science, University of Illinois at Urbana-Champaign

***Abstract*—Due to the rapid growth of real-time applications and the ubiquity of IEEE 802.11 MAC as a layer-2 protocol for wireless local area networks (WLANs), it is of increasing interest to support quality of service (QoS) in such WLANs. In this paper, we develop a simple but accurate enough analytical model for predicting queueing delay in non-homogeneous random access based WLANs. This leads to tractable solutions for meeting queueing delay specifications of a number of flows. Using this model, we address the *feasibility problem* of whether the mean delays required by a set of inelastic flows can be guaranteed in WLANs. Based on the model and feasibility analysis, we further develop an optimization technique to minimize the delays for inelastic flows. We present extensive simulation results to demonstrate the accuracy of our model and the performance of the algorithms.**

I. INTRODUCTION

The recent rapid growth of real time applications has led to a strong need to provide quality of service (QoS) for mobile computers and portable devices in wireless local area networks (WLANs). This has to be supported over the IEEE 802.11 since it has gained widespread popularity and become the de facto WLAN standard. However, the mechanisms employed in the IEEE 802.11 MAC, namely random access and the distributed coordination function (DCF), render it substantially more difficult to ensure delay guarantees because of the channel contention and the random back-off mechanism. Therefore, as the first task confronting researchers in this field, it is necessary to characterize the delays in such networks. Second, it is important to devise solutions that provide the required delay performance. We address both issues in this paper.

Existing studies on the performance analysis of the IEEE 802.11 MAC have focused on its throughput capacity in networks with saturated traffic; see Bianchi [1], Cali, Conti, and Gregori [2]. In [3], a discrete-time M/Geo/1 queueing analysis under network saturation is studied. Models for unsaturated homogeneous networks have also been reported. For example, Medepalli and Tobagi [4] present a unified model for multihop networks that approximates each queue by an independent M/M/1 queue. However, this approximation is not accurate for detailed delay analysis in WLAN. Tikoo and Sikdar [5] present a G/G/1 queueing model for delay analysis in homogeneous networks. Their focus is on performance analysis of the standard IEEE 802.11 DCF. Various studies have also been conducted on providing QoS support in WLANs, and most use centralized polling techniques based on the point coordination function (PCF). For example, Coutras, Gupta and Shroff [6] analyze the performance of PCF in support of voice services. However, they do not address the fact that both best-effort traffic and real-time traffic coexist in WLANs, and IEEE 802.11 DCF is the de facto setting used in most WLANs.

Providing QoS requires networks to support service differentiation under non-homogeneous traffic dynamics. Networks should also reallocate limited resources from the over-provisioned flows to the under-provisioned flows. IEEE 802.11e has been proposed to enhance the original standard to support QoS. However, IEEE 802.11e classifies flows only by their applications (e.g., voice, video, etc.) and provides the same service to flows that fall in the same class. Moreover, it only differentiates priority among flows, but does not actually provide delay guarantees. A non-homogeneous and adaptive WLAN is highly preferred over one that operates in a fixed homogeneous manner. However, an accurate model of non-homogeneous flows in random access WLANs, especially their delay characterization, is still elusive.

We develop a simple but sufficiently accurate analytical model based on an M/G/1 queue for non-homogeneous unsaturated IEEE 802.11 networks. We characterize the channel access delay with respect to the contention window and the probability that the queue is nonempty. The latter in turn depends on the channel access delay. Both this probability of being non-empty and the access delay can be jointly obtained by solving a coupled system of nonlinear equations through a fixed point iteration with a carefully chosen initial point so that it converges to a fixed point. Moreover, we show that in random access networks, the second moment of the access delay is determined only by its first moment if the packet size is sufficiently large. This approximation simplifies the formula of the queueing delay. Thereby, we analytically determine whether the network can provide the delay guarantees required by the QoS flows.

The contributions of the paper are summarized as follows:

This material is based upon work partially supported by NSF under Contracts CNS-1035378, CCF-0939370, CNS-0905397, CNS-1035340, USARO under Contract Nos. W911NF-08-1-0238 and W911NF-07-10287, AFOSR under Contract FA9550-09-0121, and CityU Project 7008087.

1) We derive a simple but accurate model for queueing delay in non-homogeneous IEEE 802.11 MAC based networks. We use it to determine the feasibility of using a random-access based WLAN to serve a set of real-time flows with mean delay requirements.
2) We provide characterization of the average delay and access rate, and propose fixed point algorithms to compute them. A linear system approximation is derived to complement the analysis.
3) We provide an algorithm to minimize the delays for a set of inelastic flows while meeting mean delay requirements.
4) We validate our algorithm to provide performance guarantees through extensive NS-2 simulations.

We motivate the non-homogeneous IEEE 802.11 flows problem in Section II. In Section III, we characterize the channel access delay and queueing delay. In Section IV, we study fixed point iterations related to the queueing model. We show how to optimize the delay performance in Section V. Numerical results are given in Section VI.

II. Problem Statement

A. Non-homogeneous IEEE 802.11 network

In IEEE 802.11 DCF networks, each node with a packet to transmit randomly selects a back-off timer BC from $[1, CW-1]$, where CW denotes the *contention window*. If the channel is sensed idle, these nodes decrement their timers until one of them expires. Then that node attempts to access the channel and the remaining nodes pause their timers. The count-down resumes when the channel is idle again. If more than one node attempts in the same slot, a collision occurs. A collided transmission is retried up to a retransmission limit before it is discarded. In the standard IEEE 802.11 network, the contention window of each node is set to be the same. This homogeneous or uniform setting works well for best-effort traffic where fairness is to be taken into account. However, the increasing need to support heterogeneous QoS requirements of different flows requires networks to have the ability to provide service differentiation to real-time flows.

We consider a WLAN with nodes that are capable of changing the backoff parameters by tuning only the contention window CW. Thus, our scheme is IEEE 802.11 standards-compatible. We analytically show that CW alone can effectively be used for resource allocation and performance differentiation.

B. Soft Deadline

We are interested in the soft deadline that is the mean delay of a flow. Soft-deadline guarantees are important for several real-time applications such as voice over IP, online games and IP-TV, since they often require a fixed bit rate but are sensitive to mean delays.

We now formulate the problem. Consider a WLAN where N nodes are active and each has a QoS flow to the access point (AP). These flows differ in rate and delay requirements. Assume that for each node i the packet arrivals form a Poisson process and the inter-arrival time is exponentially distributed with mean $1/\lambda_i$. Note that the Poisson arrival model has been used in the literature, e.g. [3]. Let $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_N]^T$ be the arrival rate vector. Additionally, each flow has a soft deadline D_i to meet. The average queueing delay of the packet for flow i is required to be less than D_i.

Let $\boldsymbol{D} = [D_1, D_2, ..., D_N]^T$ denote the target delay vector. Given both $\boldsymbol{\lambda}$ and $\boldsymbol{D}$, the question we address is the following: Does there exist an assignment of contention windows $\boldsymbol{CW} = [CW_1, CW_2, ..., CW_n]^T$ such that all deadlines are met? This question is especially important for admission control, which needs to decide whether it is feasible to accommodate a new flow in the network without hurting the performance of high priority flows (i.e., the already existing flows). Furthermore, if it is indeed feasible, to achieve all these mean delays, then how should one assign CW_i for each node i? We provide analysis to answer these two key questions in the following.

III. Analytical Model of Non-homogeneous IEEE 802.11 Network

A. Media Access Delay

We now analytically address the media access delay of a *non-homogenous* WLAN. We do not employ the exponential back-off algorithm implemented in the standard protocol because our standalone scheme consists of choosing a fix contention window for each flow so as to meet its mean delay. Imposing the redundant CW adjustment mechanism, e.g., exponential back-off algorithm, is an unnecessary layer of adaptation that is not needed in our scheme. We note that similar approaches have been adopted in the literature. The schemes proposed in [7], [8] disable the exponential backoff, and directly adjust the contention window. However, their goal is to maximize the throughput, while ours is to provide the delay guarantees for heterogeneous flows.

We consider an "access rate" for node i that is equal to $2/CW_i$. This corresponds to IEEE 802.11 DCF with BC chosen randomly from $[1, CW-1]$ [2], [1], [7], [9]. Since our flows are not "saturated," queues may be empty, i.e., they do not transmit any packet. Let **NE** denote the event "queue not empty" and **E** denote the event "queue empty". Then, the unconditional channel access (**CA**) probability of node i is

$$\mathrm{P}[\mathbf{CA}] = \mathrm{P}[\mathbf{CA}|\mathbf{E}]\mathrm{P}[\mathbf{E}] + \mathrm{P}[\mathbf{CA}|\mathbf{NE}]\mathrm{P}[\mathbf{NE}].$$

It is obvious that $\mathrm{P}[\mathbf{CA}|\mathbf{E}]$ is equal to zero because the node has no packet to transmit when empty. We next approximate $\mathrm{P}[\mathbf{CA}|\mathbf{NE}]$ by $\frac{2}{CW}$, which is only an approximation when the random backoff is chosen uniformly within $[1, CW-1]$, but is not an approximation if it is attempted after an exponentially distributed interval. Denoting $p_i = 2/CW_i$, and ρ_i as the probability that the queue is not empty, we have

$$\mathrm{P}[\mathbf{CA}] = \frac{2\rho_i}{CW_i} = \rho_i p_i. \tag{1}$$

Let $\boldsymbol{p}$ be the vector $[p_1, p_2, ..., p_N]^T$, noting that $0 \le p_i \le 1$. Likewise, $\boldsymbol{\rho} := [\rho_1, \rho_2, \ldots, \rho_N]^T$.

Now, we compute the probability P_I^i that the channel is idle when node i has a packet to send, the probability P_S^i

that the channel is successfully carrying a packet of node i, and the probability P_O^i that node i sees the channel as busy though node i does not transmit a packet successfully. Note that $P_I^i + P_S^i + P_O^i = 1$ for all i. All these quantities are to be computed as a function of the vector $\boldsymbol{p}$.

Note that node i competes for channel access only when it has a packet to transmit. Under this condition, node i finds the channel idle in a time slot if it does not attempt and no other node attempts at the beginning of this slot. Hence,

$$P_I^i = (1 - p_i) \prod_{j \neq i}^{N} (1 - \rho_j p_j). \tag{2}$$

Node i successfully transmits a packet if it attempts, and no other node attempts in the same slot. This probability is

$$P_S^i = p_i \prod_{j \neq i}^{N} (1 - \rho_j p_j). \tag{3}$$

Otherwise, node i sees the channel occupied by other activities, consisting of successful transmissions of other nodes or collided transmissions. Note that the collided transmissions consist of both the transmissions involving node i as well as those not involving node i. Thus, we have

$$P_O^i = 1 - P_I^i - P_S^i = 1 - \prod_{j \neq i}^{N} (1 - \rho_j p_j). \tag{4}$$

Define the *service time* x of a packet (also referred to as *channel access delay* here) as the time from the instant the packet reaches the head of the queue in the node till the instant it successfully departs from the queue. This service time includes two parts, namely the channel contention delay and the packet transmission time. For simplicity, we assume that all packets are of the same size and all nodes adopt the same bit rate to transmit. Thus, they have the same packet transmission airtime denoted by T. More precisely, in the IEEE 802.11 network, a *packet transmission airtime* is given by

$$T := \text{DIFS} + \text{PACKET} + \text{SIFS} + \text{ACK}, \tag{5}$$

where DIFS denotes the duration of the distributed interframe space, PACKET denotes the transmission time of a data packet, SIFS denotes the duration of the short interframe space, and ACK denotes the transmission time of an acknowledgement. There are two access modes used in IEEE 802.11 DCF, namely the basic access mode and the RTS/CTS access mode. In this paper, we model the system only for the basic access mode. The RTS/CTS access mode is usually disabled in practice due to its large overhead. In the basic access mode, a collision is detected when a node does not receive an ACK within an ACK-timeout period. The ACK-timeout is defined to be the time to transmit an ACK frame plus SIFS. Thus, we assume that the airtime spent on a collided transmission is the same in duration as that of a successful transmission.

We denote a slot-time duration by τ. Let t_k denote the time instant when the k-th idle slot begins, i.e., the instant that the channel is idle at the beginning of the corresponding slot. There are two possible events following this instant: a) the channel continues to be idle for a duration of τ until the next idle slot begins; b) at least one of the nodes attempts to transmit in this slot, which results in a T time unit channel-busy period. We assume that the intervals $S_i(k) = t_{k+1} - t_k$ are independent and identically distributed random variables and refer to these intervals as *virtual slots*.

Assume that the time interval from the time the packet reaches the head of the queue at node i to the time it starts to depart from the queue consists of K_i virtual slots, where K_i is a random variable independent of S_i. Its distribution is given by

$$\text{P}[K_i = n] = (1 - P_S^i)^n P_S^i \quad \text{for } n = 0, 1, 2, \ldots. \tag{6}$$

It follows that

$$E[K_i] = P_S^i \sum_{n=0}^{\infty} n(1 - P_S^i)^n = \frac{1 - P_S^i}{P_S^i}. \tag{7}$$

For node i, its *service time* is therefore

$$x_i = \sum_{k=1}^{K_i} S_i(k) + T, \tag{8}$$

where the $S_i(k)$ are Bernoulli random variables that are either equal to τ if the channel is idle or equal to T if a transmission of a node other than i occurs:

$$S_i(k) = \begin{cases} \tau & \text{with probability } \frac{P_I^i}{1 - P_S^i} \\ T & \text{with probability } \frac{P_O^i}{1 - P_S^i}. \end{cases} \tag{9}$$

Then, we have

$$E[S_i] = \frac{P_I^i \tau + P_O^i T}{1 - P_S^i}. \tag{10}$$

One can see that both $E[S_i] < \infty$ and $E[K_i] < \infty$. From the independence of S_i and K_i, we can apply Wald's equation [10] to obtain

$$X_i := E[x_i] \quad = \quad E[K_i] E[S_i] + T. \tag{11}$$

Substituting (7) and (10) into (11) gives

$$X_i \quad = \quad \frac{P_I^i \tau + P_O^i T}{P_S^i} + T. \tag{12}$$

Note that (12) captures an interesting relationship between the expected service time and the access rate in CSMA-based random access WLAN. Since the network is unsaturated, we need to determine the probability ρ_i that the queue is non-empty. However, since each node is an M/G/1 queue, we have

$$\rho_i = \lambda_i X_i. \tag{13}$$

Substituting (12) into (13), we have N equations with N unknowns $[x_1, x_2, \cdots, x_N]$. Solving this N dimensional vector fixed point problem gives us the service times for the non-homogeneous random access flows.

We summarize this relationship that allows us *to compute the mean service times for non-homogeneous random access WLANs*: Given the contention windows CW_i, the mean service times are given by (12), where P_I^i, P_S^i and P_O^i are given by (2,3,4), with p_i defined by (1). The quantities ρ_i's in (1) satisfy (13).

B. Queueing Delay

In the previous section, we have derived an analytical model that can be used to compute the service time if the access rates of all nodes are given via their contention windows. Since many real-time applications such as online games, VoIP and IPTV require strict limits on jitter and mean delay, in this section, we study how the non-homogeneous contention window settings and the non-homogeneous throughput requirements jointly affect the average queueing delay.

Define the *queueing delay* of a packet to be the time from the instant that the packet arrives at the queue to the instant that the packet successfully departs from the queue. The average queue size of the $M/G/1$ queue is given by [11]:

$$E[Q_i] = \lambda_i X_i + \frac{\lambda_i^2 E[x_i^2]}{2(1-\lambda_i X_i)}, \tag{14}$$

where Q_i denotes the queue size and $E[x_i^2]$ is the second moment of the service time.

Using Little's law, the average queueing delay Y_i is

$$Y_i = \frac{E[Q_i]}{\lambda_i} = X_i + \frac{\lambda_i E[x_i^2]}{2(1-\lambda_i X_i)}. \tag{15}$$

To determine the average queueing delay (15), we need to determine the second moment of the service times. In (8), we have characterized the service time x by a sequence of virtual slots S plus a transmission airtime T. Taking squares on both sides of (8), we have

$$\begin{aligned} x_i^2 &= (\sum_{k=1}^{K_i} S_i[k] + T)^2 \\ &= \sum_{k=1}^{K_i} S_i^2[k] + 2\sum_{k=2}^{K_i}\sum_{l=1}^{k-1} S_i[k]S_i[l] + 2T\sum_{k=1}^{K_i} S_i[k] + T^2. \end{aligned} \tag{16}$$

Applying Wald's equation again, we get

$$\begin{aligned} E[x_i^2] = E[K_i]E[S_i^2] + E[K_i^2 - K_i]E^2[S_i] \\ + 2TE[K_i]E[S_i] + T^2. \end{aligned} \tag{17}$$

Using the distribution of S_i in (9), we compute

$$E[S_i^2] = \frac{\tau^2 P_I^i + T^2 P_O^i}{1 - P_S^i}. \tag{18}$$

To determine $E[K_i^2 - K_i]$, we first obtain the moment generating function of K_i from (6) as follows:

$$M_{K_i}(B) = \sum_{n=0}^{\infty} B^n (1-P_S^i)^n P_S^i = \frac{P_S^i}{1-(1-P_S^i)B}. \tag{19}$$

It is easy to verify that

$$\begin{aligned} \frac{\mathrm{d}^2 M_{K_i}(B)}{\mathrm{d}\, B^2}|_{B=1} &= \sum_{n=0}^{\infty} n(n-1)B^{n-2}(1-P_S^i)^n P_S^i|_{B=1} \\ &= \sum_{n=0}^{\infty} n(n-1)(1-P_S^i)^n P_S^i = E[K_i^2 - K_i]. \end{aligned} \tag{20}$$

Hence, from (19) and (20), we get

$$E[K_i^2 - K_i] = \frac{2(1-P_S^i)^2}{(P_S^i)^2}. \tag{21}$$

Thus, substituting (7), (10), (21) and (18) into (17), we get the second moment of the service time for node i as follows:

$$\begin{aligned} E[x_i^2] = \frac{\tau^2 P_I^i + T^2 P_O^i}{P_S^i} + \frac{2(\tau P_I^i + T P_O^i)^2}{(P_S^i)^2} \\ + 2T\frac{\tau P_I^i + T P_O^i}{P_S^i} + T^2. \end{aligned} \tag{22}$$

Substituting (22) and (12) into (15), we therefore obtain the average queueing delay with respect to vector $\boldsymbol{p}$.

Queueing delays as a function of contention windows $Y_i(\boldsymbol{p})$: Consider a non-homogeneous random access WLAN with contention windows CW_i and packet transmission time T. Then the average queueing delay is given by (15), where $E[x_i]$ is given by (12), $E[x_i^2]$ is given by (22), P_I^i, P_S^i and P_O^i are given by (2,3,4), and $\boldsymbol{\rho} = [\rho_1, \rho_2, \cdots, \rho_N]^T$ is a fixed point of (13).

C. Queueing Delay and Channel Access Delay

Substituting (12) into (22), we have

$$\begin{aligned} E[x_i^2] &= 2(X_i - T)^2 + 2T(X_i - T) + T^2 \\ &\quad + \frac{P_I^i \tau^2 + P_O^i T^2}{P_I^i \tau + P_O^i T}(X_i - T). \end{aligned} \tag{23}$$

Since $\lim_{\tau\to 0} E[x_i^2] = 2(X_i - T)^2 + 2T(X_i - T) + T^2 + T(X_i - T)$, if we assume that the packet transmission airtime T is sufficiently large compared to slot-time τ, then we get a simplified formula for $E[x_i^2]$ as follows:

$$E[x_i^2] = (2X_i - T)X_i. \tag{24}$$

Note that (24) implies that the second moment of x_i is determined by its first moment only. This demonstrates an interesting property inherent in the random access mechanism. Therefore, the average delay is

$$Y_i = \frac{(2-\lambda_i T)X_i}{2(1-\lambda_i X_i)}. \tag{25}$$

Now, (25) is equivalent to

$$X_i = \frac{2Y_i}{2 - \lambda_i T + 2\lambda_i Y_i}. \tag{26}$$

Note that (26) captures an important property: *The mean queueing delay in a random access network is determined only by the mean channel access delay.*

IV. Analysis of Fixed-Point Problems

A. Nonlinear Characterization of Delay and Access Rate

We have shown that when the transmission airtime T is sufficiently large compared to the slot-time τ, the queueing delay $\boldsymbol{Y}$ is determined by $\boldsymbol{X}$. Thus, we need to characterize X_i, i.e., to analyze (12). Based on (12), we derive a set of fixed point equations:

$$p_i X_i + (1-p_i)(T-\tau) = \frac{T}{\prod_{j\neq i}(1-\lambda_j X_j p_j)} \quad \forall\, i. \tag{27}$$

There are two problems of interest, namely *analysis* and *design*. We now express the fixed point problems of both problems in the form of (27). The analysis or performance

evaluation (PE) problem consists of determining the delay, given the access rates. The design or access rate assignment (ARA) problem is the reverse; it consists of determining the access rates for the flows so as to meet all the delay constraints.
Performance Evaluation (**PE**): We fix the access rate $\boldsymbol{p}$, and evaluate the channel access delay $\boldsymbol{X}$. For node i, its delay can be written as

$$\begin{aligned} X_i &= I_i^{PE}(\boldsymbol{X}) \\ &:= \frac{T}{p_i \prod_{j\neq i}(1-\lambda_j X_j p_j)} - \frac{(1-p_i)(T-\tau)}{p_i}. \end{aligned} \tag{28}$$

We denote by $\boldsymbol{X}^*$ a fixed point of (28), assuming one exists. We consider the following fixed point iteration to solve (28):

$$\boldsymbol{X}(k+1) = \boldsymbol{I}^{PE}(\boldsymbol{X}(k)). \tag{29}$$

From a protocol designer's viewpoint, it is instead more interesting to compute the access rate assignment such that all the flows meet their required delays:
Access Rate Assignment (**ARA**): We want to adjust the access rate $\boldsymbol{p}$ such that all the delays $\boldsymbol{X}$ are fulfilled. For node i, the access rate is given by

$$\begin{aligned} p_i &= I_i^{ARA}(\boldsymbol{p}) \\ &:= \frac{T}{(X_i - T + \tau)\prod_{j\neq i}(1-\lambda_j X_j p_j)} - \frac{T-\tau}{X_i - T + \tau}. \end{aligned} \tag{30}$$

We denote by $\boldsymbol{p}^*$ a fixed point of (30), assuming one exists. We consider the following fixed point iteration to solve (30):

$$\boldsymbol{p}(k+1) = \boldsymbol{I}^{ARA}(\boldsymbol{p}(k)). \tag{31}$$

B. Linear System Approximation

Note that $\lambda_i X_i p_i < 1$ for all i if the system is stable. Now, using the fact that $1/(1-z) \geq 1+z$ for nonnegative $z<1$, we can lower bound the RHS of (27) by an affine expression. In particular, we have

$$p_i X_i + (1-p_i)(T-\tau) \geq T(1+\sum_{j\neq i}\lambda_j X_j p_j) \ \ \forall\, i. \tag{32}$$

Note that we can approximate the inequality in (32) by an equality if we assume small $p_i X_i$ for all i, and apply Taylor's expansion theorem to the RHS of (27). This leads us to consider the following fixed point equation:

$$p_i X_i - \sum_{j\neq i}\lambda_j T p_j X_j = p_i T + (1-p_i)\tau \ \ \forall\, i. \tag{33}$$

Now, we can consider two different *linear fixed point equations* in the form of (33): One in terms of $\boldsymbol{X}$ for performance evaluation assuming fixed $\boldsymbol{p}$, and the other in terms of $\boldsymbol{p}$ for access rate assignment assuming fixed $\boldsymbol{X}$.

The following results show that each of these two linear fixed point iterations has a unique solution.

Theorem 4.1: If $[p_1, p_2, ..., p_N]^T$ are given, (33) has a unique solution for $[\tilde{X}_1, \tilde{X}_2, ..., \tilde{X}_N]^T$.

Proof: Let $\boldsymbol{y}$ denote the vector $[p_1\tilde{X}_1, p_2\tilde{X}_2, \cdots, p_N\tilde{X}_N]^T$ and $\boldsymbol{b}$ denote the vector $[p_1T + (1-p_1)\tau, p_2T + (1-p_2)\tau, \cdots, p_NT + (1-p_N)\tau]^T$. Then we represent (33) by

$$\boldsymbol{y} = F\boldsymbol{y} + \boldsymbol{b}, \tag{34}$$

where F is an irreducible nonnegative matrix with entries:

$$F_{lj} = \begin{cases} 0, & \text{if } l = j \\ \lambda_i T, & \text{if } l \neq j. \end{cases} \tag{35}$$

We now apply nonnegative matrix theory to characterize the solution to (34). Let Λ_A denotes the spectral radius of a nonnegative matrix A. By the Collatz-Wielandt theorem (see, e.g., [12]),

$$\Lambda_F \leq \max_i \sum_{j\neq i}\lambda_j T < \sum_i \lambda_i T < 1, \tag{36}$$

where the last inequality follows from the necessary condition that the $M/G/1$ system is stable only if the workload is strictly less than 1, i.e., $\sum_i \lambda_i T < 1$. Next, we state the following result from [12].

Lemma 4.2: A necessary and sufficient condition for a solution $\mathbf{z} \geq \mathbf{0}, \mathbf{z} \neq \mathbf{0}$ to exist to the equations $(I-A)\mathbf{z} = \mathbf{c}$, for any $\mathbf{c} \geq \mathbf{0}, \mathbf{c} \neq \mathbf{0}$ is that $\Lambda_A < 1$. In this case there is only one solution $\mathbf{z}$, which is strictly positive and given by $\mathbf{z} = (I-A)^{-1}\mathbf{c}$.
Applying Lemma 4.2 to (34), this implies $(I-F)^{-1}\boldsymbol{b}$ has a unique positive solution. This proves the theorem. ■

Lemma 4.3: Assume $\boldsymbol{X}$ is given. If $\tilde{\boldsymbol{p}}$ is the fixed point of (33), and $\boldsymbol{p}^*$ is the fixed point of (27), then we have, component wise, $\tilde{\boldsymbol{p}} < \boldsymbol{p}^*$.

Proof: Suppose the following holds:

$$\tilde{p}_i X_i + (1-\tilde{p}_i)(T-\tau) = T(1+\sum_{j\neq i}\lambda_j X_j \tilde{p}_j) \ \ \forall\, i, \tag{37}$$

$$p_i^* X_i + (1-p_i^*)(T-\tau) = \frac{T}{\prod_{j\neq i}(1-\lambda_j X_j p_j^*)} \ \ \forall\, i. \tag{38}$$

For each i, we subtract (37) from (38) to obtain

$$\begin{aligned} &(X_i - T + \tau)(p_i^* - \tilde{p}_i) \\ &= \frac{T}{\prod_{j\neq i}(1-\lambda_j X_j p_j^*)} - T(1+\sum_{j\neq i}\lambda_j X_j \tilde{p}_j) \\ &> T(1+\sum_{j\neq i}\lambda_j X_j p_j^*) - T(1+\sum_{j\neq i}\lambda_j X_j \tilde{p}_j) \\ &= \sum_{j\neq i}\lambda_j T X_j (p_j^* - \tilde{p}_j). \end{aligned} \tag{39}$$

Let $\boldsymbol{u}$ denote a vector with the ith entry as $X_i(p_i^* - \tilde{p}_i)$. Now, (39) for all i can be written in matrix form as

$$(I-C)\boldsymbol{u} = \boldsymbol{v} > 0, \tag{40}$$

where $\boldsymbol{v}$ denotes some positive vector (with the positive slack of inequality (39) as its ith entry), and C is a positive matrix with entries

$$C_{lj} = \begin{cases} (T-\tau)/X_l, & \text{if } l = j \\ \lambda_i T, & \text{if } l \neq j. \end{cases} \tag{41}$$

Since C is a positive matrix, using the Perron-Frobenius theorem, Λ_C is strictly positive. Now, Λ_C satisfies

$$\begin{aligned} \Lambda_C &\overset{(a)}{\leq} \max_i(\textstyle\sum_{j\neq i}\lambda_j T + \frac{T-\tau}{X_i}) \\ &\overset{(b)}{<} \max_i(\textstyle\sum_{j\neq i}\frac{T}{X_j} + \frac{T-\tau}{X_i}) \overset{(c)}{<} \textstyle\sum_i \frac{T}{X_i} \overset{(d)}{<} 1, \end{aligned} \tag{42}$$

where inequality (a) is due to the Collatz-Wielandt theorem, inequality (b) is due to the service rate $1/X_i$ being strictly larger than the arrival rate λ_i (as (26) enforces this constraint), inequality (c) is obvious, and inequality (d) is due to the necessary stability condition for a $M/G/1$ queue. Applying Lemma 4.2 to (40), shows that $\boldsymbol{u}$ is strictly positive. This proves the lemma. ■

C. Convergence

Theorem 4.4: If $\boldsymbol{p}^*$ exists, then starting from $\tilde{\boldsymbol{p}}$, the **ARA** algorithm produces a monotone increasing sequence of vectors $\boldsymbol{p}(k)$ that converges to a fixed point.

Proof: By *Lemma* 4.3, we know $\tilde{\boldsymbol{p}} < \boldsymbol{p}^*$. Note that $\boldsymbol{I}^{ARA}(\boldsymbol{p})$ is a monotone non-decreasing function. Thus, starting from $\tilde{\boldsymbol{p}}$, we have $\boldsymbol{p}(1) = \boldsymbol{I}^{ARA}(\tilde{\boldsymbol{p}}) < \boldsymbol{I}^{ARA}(\boldsymbol{p}^*)$ and $\boldsymbol{p}(1) = \boldsymbol{I}^{ARA}(\tilde{\boldsymbol{p}}) \geq \tilde{\boldsymbol{p}}$. Suppose $\boldsymbol{p}(1) \leq \boldsymbol{p}(2) \leq \cdots \leq \boldsymbol{p}(n) \leq \boldsymbol{p}^*$. Then monotonicity implies

$$\begin{aligned} \boldsymbol{p}^* &= \boldsymbol{I}^{ARA}(\boldsymbol{p}^*) \geq \boldsymbol{I}^{ARA}(\boldsymbol{p}(n)) \\ &= \boldsymbol{p}(n+1) \geq \boldsymbol{I}^{ARA}(\boldsymbol{p}(n-1)) = \boldsymbol{p}(n). \end{aligned} \tag{43}$$

That is, $\boldsymbol{p}^* \geq \boldsymbol{p}(n+1) \geq \boldsymbol{p}(n)$. Hence, the sequence $\boldsymbol{p}(n)$ is nondecreasing and bounded above by $\boldsymbol{p}^*$. It implies that $\boldsymbol{p}(n)$ converges to a fixed point. ■

One can use a similar approach to prove the convergence of the **PE** algorithm (29), and the proof is omitted.

V. Applications

A. Feasibility Problem

To demonstrate the utility of the proposed model, we use the above algorithm to address the following important ARA question: *In an IEEE 802.11 network, if the arrival rates $\boldsymbol{\lambda}$ and the required delays $\boldsymbol{D} = [D_1, D_2, ..., D_N]^T$ are given, does there exist a set of access rates $[p_1, p_2, \cdots, p_N]^T$ such that the resulting delay for each node i is guaranteed to be smaller than D_i*? We refer to this problem as the *average delay feasibility problem*.

More formally, we say that $\{(\lambda_1, D_1), (\lambda_2, D_2), \cdots, (\lambda_N, D_N)\}$ is feasible if there exist $[p_1, p_2, \cdots, p_N]^T$ such that

$$Y_i(\boldsymbol{p}) \leq D_i \; \forall \, i. \tag{44}$$

We argue that if there exists a $\boldsymbol{p}$ such that the equality holds (i.e., $Y_i \equiv D_i$, for $i = 1, 2, ..., N$), then $\{(\lambda_1, D_1), (\lambda_2, D_2), ..., (\lambda_N, D_N)\}$ is feasible. We implicitly assume in the following that if a vector of delays is feasible, then any set of component-wise larger set of delays is also feasible. Equivalently, we have the expected channel access delay as

$$X_i = \frac{2D_i}{2 - \lambda_i T + 2\lambda_i D_i}, \tag{45}$$

where we substitute $Y_i = D_i$. Note that both D_i and λ_i are inputs, and hence X_i is completely determined by them. Consequently, $\rho_i = \lambda_i X_i$ is also determined. Substituting ρ_i into (12) yields a fixed point problem to determine contention windows $\boldsymbol{p}$. One can use the **ARA** algorithm proposed in the previous section to solve this fixed point problem. After obtaining the fixed point $\boldsymbol{p}^*$, if $0 < p_i^* < 1$ for all i, then we can assert and conclude that the flows are feasible, and a feasible contention window CW_i is then the maximum integer that is smaller than $2/p_i^*$. Otherwise, we conclude that the flows are not feasible because if the fixed point had existed, the **ARA** algorithm is guaranteed to converge. We will provide examples in simulations.

B. Minimization of Delay

We now consider a scheme for the delay minimization problem that is solved by a central controller, e.g., access point in a WLAN, which collects the QoS requirements $\{(\lambda_1, D_1), (\lambda_2, D_2), ..., (\lambda_N, D_N)\}$ from all nodes.[1] Based on this information, the WLAN first solves the feasibility problem in Section V-A, and then optimizes the delay performance. Assume that the ith node has a cost function $\bar{f}_i(Y_i)$ that is differentiable, non-decreasing and strictly convex. Now, from (25), Y_i is convex in X_i. We substitute (25) into $\bar{f}_i(Y_i)$ to yield a convex function in X_i, which we denote as $f_i(X_i)$. This leads us to consider the following optimization problem:

$$\min \; \sum_{i=1}^{N} f_i(X_i) \tag{46}$$

$$\text{s.t.} \quad 0 \leq X_i \leq \hat{X}_i := \frac{2D_i}{2 - \lambda_i T + 2\lambda_i D_i} \;\; \forall \, i, \tag{47}$$

$$X_i = I_i^{PE}(\boldsymbol{X}(\boldsymbol{p})) \;\; \forall \, i, \tag{48}$$

$$0 < p_i \leq 1 \;\; \forall \, i, \tag{49}$$

$$\text{Variables:} \quad X_i, \; p_i \; \forall \, i. \tag{50}$$

Above, constraint (47) guarantees that the average delay is less than the required delay. However, the constraint (48) that relates $\boldsymbol{p}$ to $\boldsymbol{X}$ is nonconvex. Hence, the optimization problem (46) is nonconvex. In this paper, we use the barrier method [13] to compute a local optimal solution. The barrier method is an interior-point method which, when started from a feasible point, yields a solution in the interior of the feasible region. This property is useful for finding a feasible solution as it is critical to meet the delay requirements, i.e., the delay constraints (47) as they are satisfied at all times.

Based on (47) and (49), we consider the barrier function:

$$B_i(\boldsymbol{p}) := \frac{1}{\hat{X}_i - X_i(\boldsymbol{p})} + \frac{1}{1 - p_i} + \frac{1}{p_i} \; \forall \, i. \tag{51}$$

Note that the barrier function increases to $+\infty$ as any of the constraints approaches its boundary. Let ϵ_i be a positive weight associated with $B_i(\boldsymbol{p})$ for all i. Then we consider the optimization problem:

$$\max \; J(\boldsymbol{p}) := \sum^{N} f_i(X_i(\boldsymbol{p})) + \sum^{N} \epsilon_i B_i(\boldsymbol{p}). \tag{52}$$

We present the following algorithm based on the gradient method to solve (52) [13].

Gradient Algorithm

1) Obtain an initial point $\boldsymbol{p}^0$ by solving the feasibility problem as discussed in Section V-A.

[1] We assume that each node has only one QoS flow for the AP.

2) For a fixed $\boldsymbol{p}^k$ (output of the feasibility problem), run the **PE** algorithm till convergence to some tolerance to obtain $\boldsymbol{X}^k$.
3) For fixed $\boldsymbol{p}^k$ and $\boldsymbol{X}^k$, obtain $\frac{\mathrm{d}\, J(\boldsymbol{p}^k)}{\mathrm{d}\, p_i^k}$ from (52) and (48).
4) Update $\boldsymbol{p}$ by

$$p_i^{k+1} = p_i^k - \beta_i \frac{\mathrm{d}\, J(\boldsymbol{p}^k)}{\mathrm{d}\, p_i^k} \ \ \forall\, i.$$

5) Repeat from Step 2) until convergence to some small tolerance.

We let β_i be a diminishing stepsize [13].

Due to the nonconvexity, our gradient algorithm in general yields a feasible solution that is not the global optimal solution of (46). However, by exploiting the linear system approximation in Section IV-B, we obtain a relaxation to (46) that yields a lower bound to the global optimal value of (46):

$$\min \ \sum_{i=1}^{N} f_i(X_i) \tag{53}$$

$$\text{s.t.} \quad 0 \le X_i \le \hat{X}_i \ \ \forall\, i, \tag{54}$$

$$X_i \ge ((I-F)^{-1}\boldsymbol{b}(\boldsymbol{p}))_i / p_i \ \ \forall\, i, \tag{55}$$

$$0 < p_i \le 1, \ \ \forall\, i, \tag{56}$$

where $(A\boldsymbol{x})_i$ denotes the ith element of the vector $A\boldsymbol{x}$, and $\boldsymbol{b}(\boldsymbol{p}) = [(T-\tau)p_1 + \tau, (T-\tau)p_2 + \tau, \cdots, (T-\tau)p_N + \tau]^T$. Note that (53) is obtained by relaxing the constraint (55) in (46) using (32) and some rearrangement. Now, (53) is still nonconvex. However, by making the change of variable $\tilde{p}_i = \log p_i$ for all i, we obtain the following convex problem that is equivalent to solving (53):

$$\min \ \sum_{i=1}^{N} f_i(X_i) \tag{57}$$

$$\text{s.t.} \quad 0 \le X_i \le \hat{X}_i \ \ \forall\, i, \tag{58}$$

$$X_i \ge ((I-F)^{-1}\boldsymbol{b}(e^{\tilde{\boldsymbol{p}}}))_i / e^{\tilde{p}_i} \ \ \forall\, i, \tag{59}$$

$$\tilde{p}_i \le 0, \ \ \forall\, i, \tag{60}$$

where $e^{\boldsymbol{a}} = [e^{a_1}, e^{a_2}, \cdots, e^{a_n}]^T$. Our simulations show that (57) yields a value that is slightly smaller than the feasible solution obtained from the gradient algorithm, showing that our gradient algorithm can compute a near-optimal solution.

VI. Simulation Results

A. Simulation Setup

The simulation environment is created using the NS-2 network simulator (version ns2.31) [14]. Table I summarizes the system parameters used in the simulation. Throughout the simulation, the only parameters that are changed are CW_{min} and CW_{max}. Note that the model does not employ exponential back-off. Thus, after obtaining a CW from the analytical model, we just set $CW_{min} = CW_{max} = CW$ to disable the exponential back-off. These values of CW_{min} and CW_{max} shown in Table I are referred to as the default settings for comparison. Collocated topologies were created in which all nodes can carrier-sense each other. Each sender node is attached to a Poisson traffic generation agent in which packet inter-arrival times can be customized. The interface queues at each node used a Droptail policy and the queue size is set at 5000 packets. Each simulation was run for 400 seconds in simulation time. Two metrics, namely the channel access delays and the queueing delays, are measured for each flow. For the channel access delay, we measure the time interval from the instant that the packet arrives at the head of the queue to the instant that the packet successfully departs from the queue. For the queueing delay, the time interval from the instant that a packet is sent by the application layer (labeled by "AGT" in trace files) to the instant that the packet is successfully received is measured.

Packet payload	1024 bytes
UDP header	20 bytes
MAC header	28 bytes
PHY header	24 bytes
ACK frame	38 bytes
Channel bit rate	11 Mbps
PHY header bit rate	1 Mbps
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
CW_{min}	31
CW_{max}	1023
Retransmission limit	7

TABLE I
System Parameters

B. Accuracy of the Analytical Model

The accuracy of the model is measured through three scenarios: channel access delays under saturated conditions, channel access delays under unsaturated conditions, and queueing delays. For each simulation, both the simulation results (denoted by "simulation") and the theoretical results obtained from our model (denoted by "theoretical") are plotted for comparison.

1) Channel access delays under saturated conditions: Since we claim our model is generic, it should apply to saturated conditions as well. In these simulations, three links were examined. The sender of each link sends a saturated traffic to the receiver. The theoretical results are obtained by applying (12), where $\rho_i = 1$ due to saturated conditions.

Two scenarios are studied. In the first scenario, CW_1 of link 1 is varied between 4 and 116 while the contention windows of link 2 and link 3 are fixed with $CW_2 = CW_3 = 32$. Figure 1 plots the simulation results as well as the theoretical results. One can observe that as CW_1 is increased, link 1's access delays increase. Even though CW_2 and CW_3 are not changed, their corresponding access delays decrease because CW_1 is increased. In the second scenario, CW_1 was changed, while holding the fixed ratio $CW_1 : CW_2 : CW_3 = 1 : 2 : 3$. The results are shown in Figure 2. One can observe that except for the nonlinear part when CW_1 is very small, the channel access delays agree with the theoretical values. The nonlinear initial part of the curves is due to the fact that the collision probability is extremely high when every node has a small CW for contention resolution.

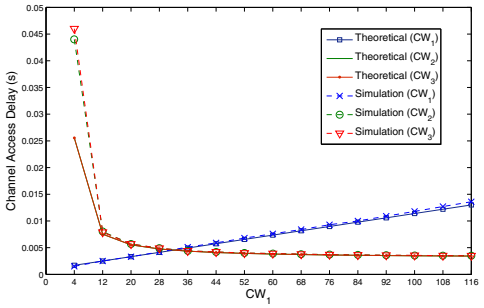

Fig. 1. Saturated conditions: Channel access delays v.s. CW_1, where $CW_2 = CW_3 = 32$.

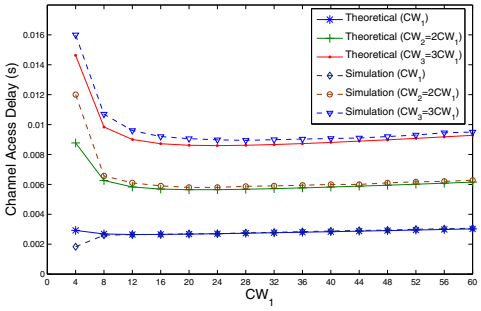

Fig. 2. Saturated conditions: Channel access delays v.s. CW_1, fix the ratio $CW_1 : CW_2 : CW_3 = 1 : 2 : 3$.

2) Channel access delays and queueing delays under unsaturated conditions: For the unsaturated conditions, three scenarios are examined. The first scenario is intended to study how traffic arrival rates affect channel access delays. The inter-arrival time of flow 3 is varied, while keeping the other two links' arrival rates fixed. The fixed packet inter-arrival times are $\frac{1}{\lambda_1} = 0.03$ and $\frac{1}{\lambda_2} = 0.005$. For the contention window, $CW_1 = CW_2 = CW_3 = 32$ is set. Figure 3 plots the results. The theoretical results are obtained by solving (12) using the **PE** algorithm.

In the second scenario, it is examined how CW affects channel access delays. Traffic arrival rates, and CW_1 and CW_2 are fixed. Only CW_3 is changed from 12 to 44. The results are shown in Figure 4. One can observe that as CW_3 is increased, the delays of flow 3 increase. As a side effect, the delays of flow 1 and flow 2 drop.

The third scenario is used to demonstrate how channel access delays change in response to the number of nodes. Each link has the same traffic rate and the same CW. In particular, $\frac{1}{\lambda_i} = 0.03$ and $CW_i = 32$ for all i. Only the number of links is changed. Figure 5 plots the results. As expected, the access delays increase as the number of links grow.

3) Queueing delays: We repeat the same three scenarios for queueing delays. One can observe similar trends in these figures to their counterparts for the channel access delays. From the three scenarios, one can see that the theoretical results do accurately match the simulation results. The accuracy is not only reflected in the trend but also in the quantitative values.

C. Performance Evaluation

In the following simulations, two case studies are examined to demonstrate the applicability of the model and evaluate the performance of the proposed algorithm. Each point in the figures is a time-average of the queueing delay over every 50 simulation seconds.

1) Feasibility: In the first case, when the capacity is insufficient, the default 802.11 setting cannot meet the delay guarantees of all the QoS flows. But, with the proposed scheme, one can find an appropriate setting at which all delay requirements are met. The three required delays are assumed to be 0.02 seconds. Note that this delay requirement is realistic according to [15]. The data rates of the inelastic flows are fixed as follows: $\frac{1}{\lambda_1} = 0.025$, $\frac{1}{\lambda_2} = 0.004$ and $\frac{1}{\lambda_3} = 0.003$. The **ARA** algorithm was run to obtain a set of feasible contention windows: $CW_1 = 66$, $CW_2 = 23$ and $CW_3 = 18$. Figure 9 plots the simulation results. One can see that the default IEEE 802.11 can guarantee the delays only for flows 1 and 2, whereas the delay of flow 3 is much larger than the allowed delay. However, the network can actually guarantee all the delays if contention windows are appropriately adjusted. In fact, one does see that the delays of all flows are met when the network uses the contention windows that are computed as described in this paper.

2) Minimizing delays: In this case study, the performance of the scheme that minimizes the average delays for inelastic flows, while preserving their delay guarantees is evaluated. The particular cost function

$$\hat{f}_i(Y_i) = \frac{Y_i^2}{\lambda_i} \tag{61}$$

is used. Inelastic flows have fixed arrival rates $\frac{1}{\lambda_1} = 0.04$, $\frac{1}{\lambda_2} = 0.004$ and $\frac{1}{\lambda_3} = 0.003$. The delay requirements are still 0.02 seconds. Compared to the input of the first case, one can observe that the network capacity is sufficient for this input. Thus, there should be room for the flows to improve the performance (i.e., queueing delays in this case). Using the gradient algorithm presented in Section V-B, the optimal CWs are computed to be $CW_1 = 19$, $CW_2 = 23$, and $CW_3 = 19$. The comparisons are plotted in Figure 10. One can observe that when configured with the CWs suggested in this paper, the network does achieve the optimized delays and provide a certain level of fairness. In contrast, in the default IEEE 802.11, flow 3 suffers from bad delay performance and experiences serious unfairness.

VII. Conclusions

We have presented a simple but apparently accurate model for analyzing queueing delay in non-homogeneous IEEE 802.11 MAC based WLANs. The model allows us to analytically study the feasibility problem of whether the network can provide the mean delay guarantees required by several non-homogeneous QoS flows. Future in order to optimize the performance of QoS flows, we have proposed an optimization algorithm to minimize the mean delays, as measured by

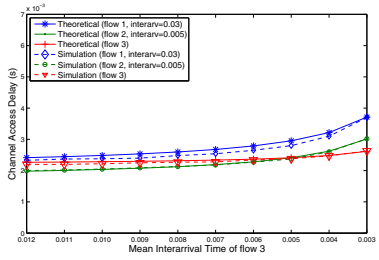

Fig. 3. Unsaturated conditions: Channel access delays v.s. $\frac{1}{\lambda_3}$, where $\frac{1}{\lambda_1} = 0.03$, $\frac{1}{\lambda_2} = 0.005$, and $CW_1 = CW_2 = CW_3 = 32$.

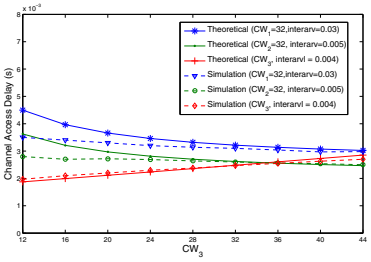

Fig. 4. Unsaturated conditions: Channel access delays v.s. CW_3, where $\frac{1}{\lambda_1} = 0.03$, $\frac{1}{\lambda_2} = 0.005$, $\frac{1}{\lambda_3} = 0.004$ and $CW_1 = CW_2 = CW_3 = 32$.

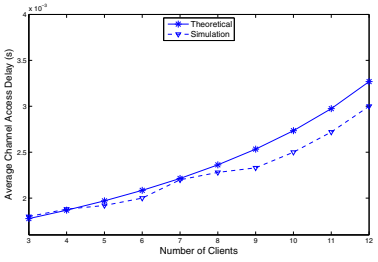

Fig. 5. Unsaturated conditions: Channel access delays v.s. the number of links, where all $\frac{1}{\lambda_i} = 0.03$ and all $CW_i = 32$.

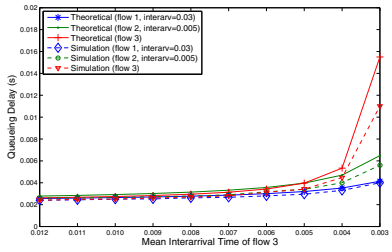

Fig. 6. Unsaturated conditions: Queueing delays v.s. $\frac{1}{\lambda_3}$, where $\frac{1}{\lambda_1} = 0.03$, $\frac{1}{\lambda_2} = 0.005$, and $CW_1 = CW_2 = CW_3 = 32$.

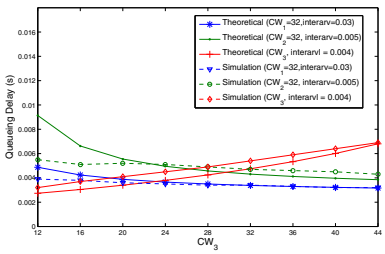

Fig. 7. Unsaturated conditions: Queueing delays v.s. CW_3, where $\frac{1}{\lambda_1} = 0.03$, $\frac{1}{\lambda_2} = 0.005$, $\frac{1}{\lambda_3} = 0.004$ and $CW_1 = CW_2 = CW_3 = 32$.

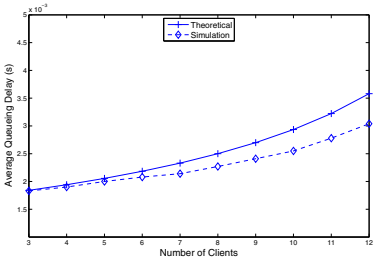

Fig. 8. Unsaturated conditions: Queueing delays v.s. the number of links, where all $\frac{1}{\lambda_i} = 0.03$ and all $CW_i = 32$.

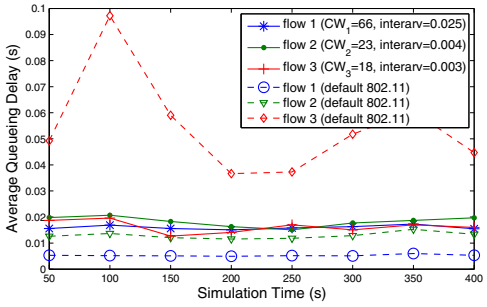

Fig. 9. A feasible solution: Illustration of the queueing delay dynamics.

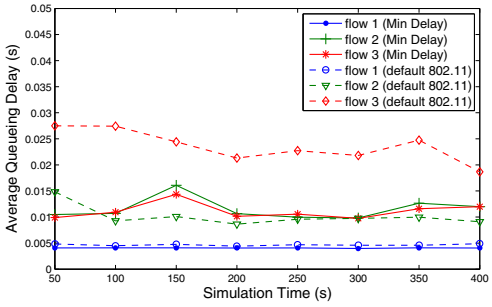

Fig. 10. Minimizing delays: $\frac{1}{\lambda_1} = 0.04$, $\frac{1}{\lambda_2} = 0.004$ and $\frac{1}{\lambda_3} = 0.003$.

a certain cost function, for a set of inelastic flows while preserving mean delay guarantees. Extensive NS-2 simulations have been conducted to verify the accuracy of the model and to evaluate the performance of the algorithms.

REFERENCES

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[2] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. on Networking*, December 2000.

[3] A. Abdrabou and W. Zhuang, "Service time approximation in IEEE 802.11 single-hop ad hoc networks," *IEEE Trans. on Wireless Communications*, vol. 7, no. 1, pp. 305–313, 2008.

[4] K. Medepalli and F. A. Tobagi, "Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications," in *Proc. IEEE INFOCOM*, 2006.

[5] O. Tickoo and B. Sikdar, "Queueing analysis and delay mitigation in IEEE 802.11 random access MAC based wireless networks," in *Proc. IEEE INFOCOM*. IEEE, 2004, pp. 1404–1413.

[6] C. Coutras, S. Gupta, and N. Shroff, "Scheduling of real-time traffic in IEEE 802.11 wireless LANs," *Wireless Networks*, vol. 6, no. 6, pp. 457–466, 2000.

[7] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," *Proc. ACM SIGCOMM*, 2005.

[8] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Allerton Conference on Communication, Control, and Computing*, 2008.

[9] Y. Gao, D.-M. Chiu, and J. C. Lui, "Determining the end-to-end throughput capacity in multi-hop networks: Methodology and applications," *Proc. ACM SIGMETRICS Perform. Eval. Rev.*, 2006.

[10] A. Wald, "Sequential tests of statistical hypotheses," *Ann. Math. Stat*, vol. 16, pp. 117–186, 1945.

[11] L. Kleinrock, *Queueing Systems*. John Wiley & Sons, 1975.

[12] E. Seneta, "Non-negative matrices and Markov chains," Springer, 2006.

[13] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[14] "The network simulator-ns2," http://www.isi.edu/nsnam/ns.

[15] Cisco, "Understanding delay in packet voice networks," http://www.cisco.com/, 2006.