Characterization and Optimization of Delay Guarantees for Real-Time Multimedia Traffic Flows in IEEE 802.11 WLANs

Yan Gao, *Member, IEEE*, Chee Wei Tan, *Senior Member, IEEE*, Ying Huang, *Member, IEEE*, Zheng Zeng, *Member, IEEE*, and P. R. Kumar, *Fellow, IEEE*

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 becomes increasingly important to support delay-based quality of service (QoS) in such WLANs. In this paper, we develop a simple and accurate enough analytical model for predicting the queueing delay of real-time multimedia traffic flows in non-homogeneous random access based WLANs. This leads to tractable analysis for meeting queueing delay specifications of a number of flows. In particular, we address the *feasibility problem* of whether the mean delays required by a set of User Datagram Protocol (UDP) flows supporting real-time multimedia traffic can be guaranteed in WLANs. Based on the model and feasibility analysis, we further develop an optimization technique to minimize the delays for the traffic flows. Moreover, we present a decentralized algorithm and report its implementation and present extensive simulation and experimental trace-based results to demonstrate the accuracy of our model and the performance of the algorithms.

Index Terms—Queueing theory, IEEE 802.11 WLAN, multimedia resource allocation, delay guarantees, quality of service

1 Introduction

The recent rapid growth of real-time applications has led to a strong need to provide delay-based 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 medium access control (MAC), namely random access and the distributed coordination function (DCF), render it more difficult to ensure delay guarantees due to the channel contention and the use of random back-off mechanism. Delay-based performance characterization of the IEEE 802.11 MAC is thus very challenging in general. In addition, it is important to control and optimize random access to meet the required delay performance. We study these two issues in this paper.

The existing studies on the performance analysis of the IEEE 802.11 MAC have largely focused on its throughput capacity in networks with saturated traffic, see Bianchi [2], Cali et al. [3]. In [4], a M/G/1 queue is analyzed under network saturation. Models for unsaturated homogeneous networks have also been reported in the literature. For example, Medepalli and Tobagi [5] present a unified queueing model for multi-hop networks that approximates each queue by an independent M/M/1 queue. However, this approximation may not be accurate for delay analysis in WLANs. Tikoo and Sikdar [6] present a G/G/1 queueing model for delay analysis in homogeneous networks, they focus only on the performance analysis of the standard IEEE 802.11 DCF. There has also been various studies using M/G/1 queue to analyze the performance in WLANs, see for example [7], [8]. Also, most work consider centralized polling techniques based on the point coordination function (PCF). For example, Coutras et al. [9] analyze the performance of PCF to support voice service applications. However, in practice, both best-effort traffic and real-time traffic can coexist in WLANs, and since IEEE 802.11 DCF is the de facto setting used in most WLANs, it is important to know how these affect the delay performance.

Providing delay-based QoS requires WLAN networks to support service differentiation under non-homogeneous traffic. The networks should also allocate the 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 within the same class. Moreover, it only differentiates priority among flows, and does not provide delay guarantees, i.e., best effort service. A non-homogeneous and adaptive WLAN is preferred over one that

- *Y. Gao is with the Accenture Technology Labs, Beijing, China. E-mail: gaoyan.hrb@gmail.com.*
- *Y. Huang is with Google Inc., Mountain View, CA 94043. E-mail: huang23@illinois.edu.*
- *Z. Zeng is with Apple Inc., Cupertino, CA 95014. E-mail: cedarzeng@gmail.com.*
- *C. W. Tan is with the College of Science and Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong. E-mail: cheewtan@gmail.com.*
- *P.R. Kumar is with the Department of Electrical and Computer Engineering at Texas A&M University, College Station, TX 77843. E-mail: prk.tamu@gmail.com.*

Manuscript received 28 Feb. 2013; revised 8 Oct. 2014; accepted 29 Dec. 2014; date of current version 31 Mar. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2407395

operates in a fixed homogeneous manner. However, an accurate model of non-homogeneous flows in random access WLANs, especially with respect to their delay characterization, is still elusive.

In this paper, we develop a simple and sufficiently accurate analytical model based on an $M/G/1$ queueing model for non-homogeneous unsaturated IEEE 802.11 networks. We characterize the mean service time with respect to the contention window and the probability that the queue is nonempty. Also we show that 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. The analysis can help determine whether we can meet the mean delay requirements of the QoS flows.

The contributions of the paper are summarized as follows:

1) We propose a simple and accurate model to analyze the mean queueing delay of non-homogeneous flows in IEEE 802.11 random access networks. This can be used to determine the feasibility of a set of real-time flows with mean delay requirements.
2) We characterize the relationship between the average delay and the access rate and propose a fixed point algorithm to compute them based on a linear system approximation.
3) We formulate the minimization of the total mean delay of a set of User Datagram Protocol (UDP) traffic flows with mean delay requirements.
4) We propose algorithms based on the proposed queueing model and show that a distributed algorithm performs well as compared to a centralized algorithm.
5) We validate our algorithms through performance evaluation (PE) using NS-2 simulations and video trace-based experiments.

We motivate the non-homogeneous IEEE 802.11 flow queueing delay problem in Section 2. In Section 3, we study a queueing model to analyze the mean service time and the mean queueing delay. In Section 4, we study fixed point iterations related to this queueing model. We show how to optimize the mean delay performance in Section 5. The simulation results can be found in Section 6. We propose a distributed algorithm and address several design issues in Section 7. Video trace-based experiments are given in Section 8. Finally, we conclude the paper in Section 9.

2 Problem Statement

2.1 IEEE 802.11 Networks and QoS

In IEEE 802.11 DCF random access networks, each node with a packet to transmit selects randomly a back-off timer counter 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. This particular node attempts to access the channel to transmit while the remaining nodes pause their timers. The decrementing mechanism resumes when the channel is idle once again. If more than one node attempt to transmit in the same slot, a collision occurs. A collided transmission is repeated until a retransmission limit is reached. In a standard IEEE 802.11 network, the contention window of each node is set to be the same. This homogeneous setting works well in practice for best-effort traffic and ensures fairness to all the nodes.

However, the increasing need to support the QoS requirements of different flows requires service differentiation especially delay guarantees for real-time flows. The standard IEEE 802.11e has been proposed to enhance the original standard to support QoS. However, the IEEE 802.11e differentiates flows by their applications (e.g., voice, video, etc.) and provides the same service to all the flows within the same class. This differentiation also does not provide delay guarantees. To provide mean delay guarantees in random access WLANs, we consider nodes that are capable of changing their back-off parameters, e.g., adapting the CW. In particular, we show that appropriately adapting CW can provide delay guarantees to real-time flows, e.g., video traffic.

2.2 Soft Deadline Guarantees

In this paper, we focus on soft deadline based on the mean delay of a flow. Soft-deadline guarantees are important to real-time applications such as voice over Internet protocol, online games, and Internet protocol television that require fixed bit rates and are sensitive to average delays. Consider a WLAN where N nodes are active and each node has a QoS flow transmitting to the access point (AP). These flows differ in their rate and mean delay requirement. Assume that for each node i the packet arrival is a Poisson process and the inter-arrival time is exponentially distributed with mean $1/\lambda_i$. Let $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$ be the arrival rate vector. Also, each flow has a delay deadline D_i to meet. It is required that the average queueing delay of a packet for flow i is less than or equal to D_i. Let $\boldsymbol{D} = [D_1, D_2, \ldots, D_N]^T$ denote the delay specification vector.

Now, suppose both $\boldsymbol{\lambda}$ and $\boldsymbol{D}$ are given. Does there exist an assignment of contention windows $\boldsymbol{CW} = [CW_1, CW_2, \ldots, CW_N]^T$ for the N flows such that all the deadlines of all the nodes are met? This question is important to admission control in determining the feasibility of a flow among existing higher priority flows. Furthermore, if it is feasible, how to achieve all these mean delays? This means, how to assign CW_i to each node i? We study these two key questions in the following.

3 Analytical Model of Non-Homogeneous IEEE 802.11 Network

3.1 Random Access Delay

We first analyze the *access delays* of *non-homogenous* flows in the WLAN. We do not consider the exponentially increasing back-off mechanism implemented in the IEEE 802.11 protocol because our scheme explicitly determines the contention window for each flow to meet the delay requirements for all the flows. Imposing an additional CW adjustment mechanism, e.g., exponential back-off algorithm, may complicate the analysis and is left for future work. In the literature, the schemes proposed in [10], [11] disable the exponential back-off mechanism, and directly adjust the contention window to maximize throughput. Our focus is on mean delay guarantees for non-homogeneous flows.

We will consider an "access rate" for node i that is equal to $2/CW_i$. This corresponds to IEEE 802.11 DCF with back-off timer counter chosen randomly from $[1, CW-1]$ [2], [3], [10], [12]. When the queues are empty, the nodes do not transmit. Let **NE** denote the event that "queue is not empty" and **E** denote the event that "queue is empty." Then, the probability of channel access (**CA**) by node i is

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

Now, the conditional probability $\mathrm{P}[\mathbf{CA}|\mathbf{E}]$ is equal to zero because a node has no packet to transmit when the queue is empty. In addition, we approximate the conditional probability $\mathrm{P}[\mathbf{CA}|\mathbf{NE}]$ by $2/CW$. Note that this is an approximation only when the back-off mechanism is enabled to choose uniformly within $[1, CW-1]$, but otherwise this is exact (and not an approximation) if transmission is attempted after an exponentially distributed interval. Let $p_i = 2/CW_i$. Let the probability that the queue being not empty, i.e., $\mathrm{P}[\mathbf{NE}]$, be ρ_i. Then, we have

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

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

Next, 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 successfully carries a packet of node i, and the probability P_O^i that node i sees the channel as busy though node i itself has not successfully transmitted a packet. In particular, we also have $P_I^i + P_S^i + P_O^i = 1$ for all i. All these quantities are in fact functions of the vector $\boldsymbol{p}$.

Next, we examine the dependence of $\boldsymbol{\rho}$ on $\boldsymbol{p}$. Since node i competes for channel access only when it has a packet to transmit, node i finds the channel idle in a time slot if no other node attempts to transmit 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 the packet with a probability given by

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

Otherwise, node i sees the channel being occupied by other activities that are either successful transmissions by other nodes or collisions. Since the collisions are due to both the transmissions from node i and the other nodes, we have

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

In the following, we study the *service time* of a packet as the time from the instant the packet reaches the head of the queue in the node to the instant that it successfully departs from the queue. This service time is a random variable that includes two parts: the channel contention delay and the packet transmission air time. For analysis tractability, we assume that all the packets have the same size and all nodes use the same transmission bit rate with the same packet transmission airtime given by T. In particular, in the IEEE 802.11 network, this *packet transmission airtime* T is given by

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

where DIFS denotes the duration of the distributed inter-frame space, PACKET denotes the transmission time of a data packet, SIFS denotes the duration of the short inter-frame space, and ACK denotes the transmission time of an acknowledgement. There are two access modes in the IEEE 802.11 DCF, namely the basic access mode and the request-to-send/clear-to-send (RTS/CTS) access mode. In this paper, we focus only on the basic access mode. In the basic access mode, a collision is detected whenever a node does not receive an ACK within an ACK-timeout that is defined to be the time to transmit an ACK frame plus the short inter-frame space duration. Thus, we assume that the airtime spent on a collision 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 kth idle slot of node i 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 that results in a T time unit channel-busy period. Define the interval for node i as

$$S_i(k) = t_{k+1} - t_k.$$

We assume that $S_i(k)$ are independent and identically distributed random variables for node i and we refer to these intervals as *virtual slots*.

Assume that the time interval from the instant the packet reaches the head of the queue at node i to the instant it starts to depart from the queue has K_i virtual slots, where K_i is a geometrically distributed random variable independent of $S_i(k)$:

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

and we have

$$E[K_i] = \frac{1-P_S^i}{P_S^i}. \tag{7}$$

We can therefore characterize the *service time* of node i denoted by x_i as given by

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

where $S_i(k)$ are the Bernoulli random variables that are either equal to τ if the channel is idle or else equal to T if a node other than i transmits, i.e.,

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

Therefore, we have

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

It is easy to see that $E[S_i(k)] < \infty$ and $E[K_i] < \infty$.

Now, from the independence of $S_i(k)$ and K_i, we can apply Wald's equation (see, e.g., [13]) to obtain

$$\begin{aligned} X_i :&= E[x_i] \\ &= E[K_i]E[S_i(k)] + T. \end{aligned} \tag{11}$$

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

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

In particular, (12) relates the expected service time to the access rate of the IEEE 802.11 DCF.

Since the network may be unsaturated, the probability that the queue is non-empty, denoted by ρ_i, is given by (assuming the M/G/1 queueing analysis):

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

Substituting (12) into (13), we obtain N equations with N unknowns $[X_1, X_2, \ldots, X_N]^T$ whose solutions provide the service times for the flows in the network.

In summary, we have obtained the fundamental relationship that allows us *to compute the mean service times for non-homogeneous flows in random access WLAN*: 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), and (4) respectively, with p_i defined by (1). In addition, the quantities ρ_i's in (1) satisfy (13).

3.2 Queueing Delay

In the previous section, we have analyzed the service time if the access rates are controlled by adapting the contention windows. Since real-time applications such as online games and voice over Internet protocol have specific requirements on the jitter and delay, we next study how the non-homogeneous contention window settings and the non-homogeneous throughput requirements affect the average queueing delay.

Let us 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 [14]:

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

where Q_i denotes the queue size and $E[x_i^2]$ is the second moment of the service time. Using Little's law [15], the average queueing delay Y_i is

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

To determine the average queueing delay (15), we need to also determine the second moment of the service times. In (8), we have characterized the service time as a sequence of the virtual slots $S_i(k)$ plus a transmission airtime T. Next, taking squares on both sides of (8), we have

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

Applying Wald's equation [14], we get

$$\begin{aligned} E\left[x_i^2\right] =& E[K_i] E\left[S_i^2(k)\right] + E\left[K_i^2 - K_i\right] E^2[S_i(k)] \\ &+ 2TE[K_i]E[S_i(k)] + T^2. \end{aligned} \tag{17}$$

Using the distribution of $S_i(k)$ in (9), we have

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

To determine $E[K_i^2 - K_i]$, we apply the moment generating function of K_i from (6) as follows (see, e.g., [15] on the method of moment generating function):

$$\begin{aligned} M_{K_i}(z) &= \sum_{n=0}^{\infty} z^n \left(1 - P_S^i\right)^n P_S^i \\ &= \frac{P_S^i}{1 - \left(1 - P_S^i\right) z}. \end{aligned} \tag{19}$$

It can be shown that

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

Hence, from both (19) and (20), we have

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

Finally, substituting (7), (10), (18) and (21) into (17), we obtain the second moment of the service time for node i:

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

After substituting (12) and (22) into (15), we obtain the average queueing delay Y_i as a function with respect to $\boldsymbol{p}$.

Queueing delay as a function of contention windows $Y_i(\boldsymbol{p})$. In summary, for non-homogeneous flows in WLAN with contention windows CW_i and packet transmission time T, the mean queueing delay is given by (15), where X_i is given by (12), $E[x_i^2]$ is given by (22) with P_I^i, P_S^i and P_O^i given by (2), (3) and (4) respectively.

3.3 Queueing Delay and Service Time for Small Slot Times

We further simplify our above analysis by using some practical assumptions. Substituting (12) into (22), we have

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

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 the slot-time τ, then we get a simplified formula for $E[x_i^2]$ given by:

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

Note that (24) implies that the second moment of the service time x_i of node i can be determined only by its first moment. Therefore, the average delay reduces to

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

Now, (25) is equivalent to

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

Interestingly, (26) illustrates a unique feature under the small slot-time assumption: *The queueing delay in a random access network is determined only by the service time.*

4 Analysis of Fixed-Point Problems

4.1 Nonlinear Characterization of Delay and Access Rate

In the previous section, 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 $\boldsymbol{X}$. Recall that X_i is given by (12). We can thus derive a set of fixed point equations given by

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

There are two parts to (27): analysis or design. The analysis part (or the performance analysis problem analysis below) consists of determining the delay, given the access rates. The design part (or the access rate assignment (ARA) problem below) consists of determining the access rates for the flows so as to meet all the delay constraints. Both parts can be solved using the fixed point equation in (27).

Performance evaluation. We fix the access rate $\boldsymbol{p}$, and evaluate the service time $\boldsymbol{X}$. 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 let X^* be a fixed point of (28) that is assumed to exist. Thus, we consider the following fixed point iteration to solve (28):

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

Access rate assignment. From a protocol designer's viewpoint, it can be interesting to compute the access rate assignment such that all the flows meet their required delays. In other word, we adapt 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 let $\boldsymbol{p}^*$ be a fixed point of (30) that is assumed to exist. Thus, we consider the following fixed point iteration to solve (30):

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

4.2 Linear System Approximation

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

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

Furthermore, let us approximate the inequality in (32) by an equality if we assume sufficiently small[1] $\lambda_i X_i p_i$ for all i, and apply a Taylor series expansion for the right-hand side of (27) and ignore higher order nonlinear terms. This leads us to consider the following fixed point equation:

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

We let $\tilde{\boldsymbol{p}}$ be a fixed point of (33). Now, we can consider two different *linear fixed point equations* in the form of (33): The first one in terms of $\boldsymbol{X}$ for the performance evaluation part assuming a fixed $\boldsymbol{p}$, and the second one in terms of $\boldsymbol{p}$ for the access rate assignment part assuming a fixed $\boldsymbol{X}$.

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

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

Proof. Let $\boldsymbol{y}$ denote the vector $[p_1\tilde{X}_1, p_2\tilde{X}_2, \ldots, p_N\tilde{X}_N]^T$ and $\boldsymbol{b}$ denote the vector $[p_1 T + (1-p_1)\tau, p_2 T + (1-p_2)\tau, \ldots, p_N T + (1-p_N)\tau]^T$. Using these notations, we represent (33) in matrix form as

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

where $\boldsymbol{F}$ is an irreducible nonnegative matrix with entries:

[1] A sufficiently small condition on $\lambda_i X_i p_i$ in order for (33) to hold is given by $\lambda_i X_i p_i < \frac{1}{2(N-1)}\ \forall i$. The proof can be found in the appendix.

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

We now apply nonnegative matrix theory to characterize the solution to (34). Let $\Lambda(\boldsymbol{A})$ denote the Perron-Frobenius eigenvalue[2] of a nonnegative matrix $\boldsymbol{A}$. By the Collatz-Wielandt theorem (see, e.g., [16]),

$$\begin{aligned} \Lambda(\boldsymbol{F}) &\leq \max_{i=1,\ldots,N} \sum_{j\neq i} \lambda_j T \\ &< \sum_{i=1}^{N} \lambda_i T \\ &< 1, \end{aligned} \tag{36}$$

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., $\sum_{i=1}^{N} \lambda_i T < 1$. Next, we state the following result from [16]. □

Lemma 4.2. *A necessary and sufficient condition for a solution $\boldsymbol{z} \geq \boldsymbol{0}, \boldsymbol{z} \neq \boldsymbol{0}$ to exist to the equations $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{z} = \boldsymbol{c}$, for any $\boldsymbol{c} \geq \boldsymbol{0}, \boldsymbol{c} \neq \boldsymbol{0}$ is that $\Lambda(\boldsymbol{A}) < 1$. In this case there is only one solution $\boldsymbol{z}$, which is strictly positive, i.e., $z \neq 0$ and $z \geq 0$, and given by $\boldsymbol{z} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{c}$.*

Applying Lemma 4.2 to (34), this implies that $(\boldsymbol{I} - \boldsymbol{F})^{-1}\boldsymbol{b}$ has a unique positive solution. This proves the theorem.

Lemma 4.3. *Assume that $\boldsymbol{X}$ 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, $\tilde{\boldsymbol{p}} \leq \boldsymbol{p}^*$ and $\tilde{p} \neq p^*$.*

Proof. Suppose the following holds:

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

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

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

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

Let us define the vector

$$\boldsymbol{u} = [X_1(p_1^* - \tilde{p}_1), X_2(p_2^* - \tilde{p}_2), \ldots, X_N(p_N^* - \tilde{p}_N)]^T.$$

Now, (39) for all i can be written in a compact form as

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

where $\boldsymbol{v}$ denotes some positive vector (with the positive slack of the inequality (39) as its ith entry), and $\boldsymbol{C}$ 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 $\boldsymbol{C}$ is a positive matrix, using the Perron-Frobenius theorem [16], $\Lambda(\boldsymbol{C})$ is strictly positive. Now, $\Lambda(\boldsymbol{C})$ satisfies

$$\begin{aligned} \Lambda(\boldsymbol{C}) &\overset{(a)}{\leq} \max_{i=1,\ldots,N} \left(\sum_{j\neq i} \lambda_j T + \frac{T - \tau}{X_i}\right) \\ &\overset{(b)}{<} \max_{i=1,\ldots,N} \left(\sum_{j\neq i} \frac{T}{X_j} + \frac{T - \tau}{X_i}\right) \\ &< \sum_{i=1}^{N} \frac{T}{X_i} \\ &\overset{(c)}{<} 1, \end{aligned} \tag{42}$$

where inequality (a) is due to the Collatz-Wielandt theorem [16], inequality (b) is due to the service rate $\frac{1}{X_i}$ being strictly larger than the arrival rate λ_i (as (26) enforces this constraint), and inequality (c) is due to the necessary condition for stability of a M/G/1 queue. Applying Lemma 4.2 to (40), $\boldsymbol{u}$ is strictly positive. This proves the lemma. □

We remark that the linear approximation to (27) only holds under certain regimes of the operating points in IEEE 802.11 MAC. It can be useful for tractable analysis, but there are limitations to this approximation. For example, increasing the number of nodes will increase service time. As future work, it is important to study other (nonlinear) approximations under which (27) can be further simplified.

4.3 Convergence

We address below the convergence result related to the fixed-point algorithms in (29) and (31) for the **ARA** and the **PE**, respectively.

Theorem 4.4. *Suppose that $\boldsymbol{p}^*$ 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 that $\tilde{\boldsymbol{p}} < \boldsymbol{p}^*$. Observe 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 that $\boldsymbol{p}(1) \leq \boldsymbol{p}(2) \leq \cdots \leq \boldsymbol{p}(n) \leq \boldsymbol{p}^*$. Then, monotonicity implies that

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

[2] The Perron-Frobenius eigenvalue of an irreducible nonnegative matrix $\boldsymbol{A}$ is the spectral radius (eigenvalue with the largest absolute value) of A denoted by $\Lambda(A)$ and furthermore, $\Lambda(A)$ is simple and positive, see, e.g., [16].

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 $\boldsymbol{p}^*$. Thus, $\boldsymbol{p}(n)$ converges to a fixed point. □

Similarly, one can prove the convergence of the **PE** algorithm (29), and the proof is omitted.

5 Applications

5.1 Feasibility Problem

As an application of the queueing model in the previous section, we address the following **ARA** question: *In an IEEE 802.11 WLAN, suppose that the arrival rates* $\boldsymbol{\lambda}$ *and the required delays* $\boldsymbol{D} = [D_1, D_2, \ldots, D_N]^T$ *are given. Does there exist a set of access rates* $[p_1, p_2, \ldots, p_N]^T$ *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*. In other words, we say that $\{(\lambda_1, D_1), (\lambda_2, D_2), \ldots, (\lambda_N, D_N)\}$ is feasible if there exist $[p_1, p_2, \ldots, p_N]^T$ 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 in the above (i.e., $Y_i \equiv D_i$, for $i = 1, 2, \ldots, N$), then $\{(\lambda_1, D_1), (\lambda_2, D_2), \ldots, (\lambda_N, D_N)\}$ is feasible. Let us assume that if a vector of delays is feasible, then any set of component-wise larger set of delays is also feasible. This means that we have the expected channel access delay of node i as given by

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

where we substitute $Y_i = D_i$. Note that both D_i and λ_i are inputs that determine X_i. Consequently, $\rho_i = \lambda_i X_i$ is also determined. Substituting ρ_i into (12) yields a fixed point problem to solve for the contention windows $\boldsymbol{p}$.

In particular, one can use the **ARA** algorithm proposed in the previous section. After obtaining the fixed point $\boldsymbol{p}^*$, if $0 < p_i^* < 1$ for all i, then we can deduce that the flows are feasible, and a feasible contention window CW_i is then given by the maximum integer that is smaller than $2/p_i^*$ for all i. Otherwise, the flows are not feasible, because if the fixed point were to exist, the **ARA** algorithm is guaranteed to converge.

5.2 Delay Minimization

We now consider a scheme for the delay minimization problem that is solved by a central controller, e.g., an access point in a WLAN, which collects the QoS requirements $\{(\lambda_1, D_1), (\lambda_2, D_2), \ldots, (\lambda_N, D_N)\}$ from all the nodes.[3] The WLAN access point first solves the feasibility problem in Section 5.1, and then optimizes the delay performance. The ith node has a cost function $\bar{f}_i(Y_i)$ that is assumed to be 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 $f_i(X_i)$. Consider the following optimization problem:

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

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

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

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

In the above, the 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, thus (46) is generally hard to solve. To numerically solve (46), we use the barrier method (interior-point method) in optimization theory [17] to compute a local optimal solution. Since the barrier method yields a solution in the interior of the feasible set, this can be useful to find a feasible solution that meets the delay requirements, i.e., the delay constraints (47) as they are satisfied at all the intermediate solution iterates.

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

$$B_i(\boldsymbol{p}) := \frac{1}{\hat{X}_i - X_i(\boldsymbol{p})} + \frac{1}{1 - p_i} + \frac{1}{p_i} \; \forall\, i. \tag{50}$$

Note that (50) goes to $+\infty$ when any ith constraint approaches its boundary. Let ϵ_i be a positive weight associated with $B_i(\boldsymbol{p})$ for all i. Consider the problem:

$$\max J(\boldsymbol{p}) := \sum_{i=1}^{N} f_i(X_i(\boldsymbol{p})) + \sum_{i=1}^{N} \epsilon_i B_i(\boldsymbol{p}). \tag{51}$$

The solution to (51) yields a suboptimal solution that is feasible to (46). We present the following algorithm based on the gradient method to solve (51) [17].

Gradient Algorithm

1) Obtain an initial point $\boldsymbol{p}^0$ by solving the feasibility problem in Section 5.1.
2) For a fixed $\boldsymbol{p}^k$ (solution of the feasibility problem), run the **PE** algorithm till convergence to obtain $\boldsymbol{X}^k$.
3) Update $\boldsymbol{p}$ by

$$p_i^{k+1} = p_i^k - \beta_i \frac{d\, J(\boldsymbol{p}^k)}{d\, p_i^k} \;\; \forall\, i, \tag{52}$$

where β_i is a positive diminishing stepsize [17].

4) Go to Step 2 until convergence to a small tolerance.

Due to the nonconvexity in (46), this gradient algorithm yields a solution that in general is not the global optimal solution of (46). However, by exploiting the linear system approximation in Section 4.2, we can obtain a relaxation to (46) that yields a lower bound to the (unknown) global optimal value of (46):

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

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

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

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

[3] We assume that each node has only a single QoS flow for the AP.

TABLE 1
System Parameters of IEEE 802.11 DCF

Packet payload	1,024 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}	1,023
Retransmission limit	7

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

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

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

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

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

where $e^{\boldsymbol{p}} = [e^{p_1}, e^{p_2}, \ldots, e^{p_n}]^T$. In practice, it is observed through our simulations in the following section that (57) often yields an optimal value that is only slightly smaller than the objective value evaluated at the suboptimal solution obtained by the gradient algorithm. This demonstrates that the gradient algorithm can compute a near-optimal solution.

6 Simulation Results

6.1 Simulation Setup

We perform our simulation using the NS-2 network simulator (version ns2.31) [18]. Table 1 summarizes the system parameters used in the simulation. We do not employ the exponential back-off mechanism and directly set $CW_{min} = CW_{max} = CW$ where CW is obtained from our algorithms. These values of CW_{min} and CW_{max} shown in Table 1 are the default baseline settings for comparison purpose. Collocated topologies are created in which all the nodes can carrier-sense one another. 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 use a droptail policy and the queue size is set at 5,000 packets. Each simulation runs for 400 seconds in simulation time. Two metrics, namely the service time and the queueing delays, are measured for each flow. For the service time, 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 is measured. 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.

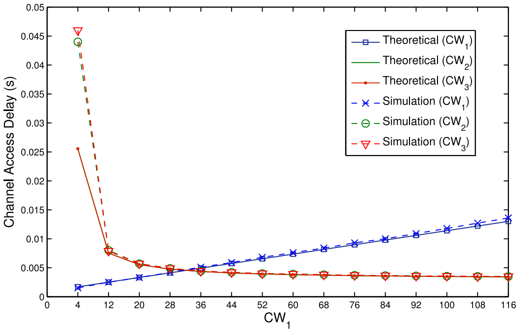


Fig. 1. Saturated conditions: service time versus CW_1, where $CW_2 = CW_3 = 32$.

6.2 Accuracy of the Analytical Model

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

6.2.1 Service Time under Saturated Conditions

In these simulations, three links are examined. The sender of each link sends a saturated flow to the receiver. The theoretical results are obtained by applying (12), where $\rho_i = 1$ due to the saturated condition. Two scenarios are studied in the following. 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 at $CW_2 = CW_3 = 32$. Fig. 1 compares the simulation results with the analysis. We 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 is changed, while keeping a fixed ratio of $CW_1 : CW_2 : CW_3 = 1:2:3$. From Fig. 2, we observe that except for the nonlinear part when CW_1 is very small, the channel access delays largely agree with the theoretical values. The nonlinear part of the curves is due to the fact that

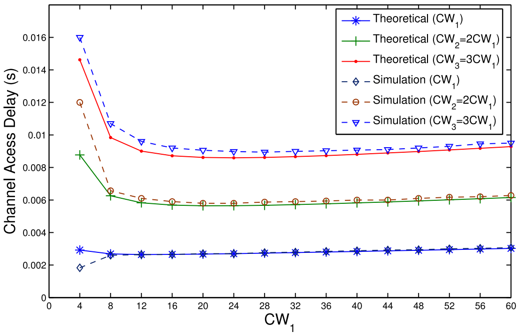


Fig. 2. Saturated conditions: service time versus CW_1, with a fixed ratio $CW_1 : CW_2 : CW_3 = 1:2:3$.

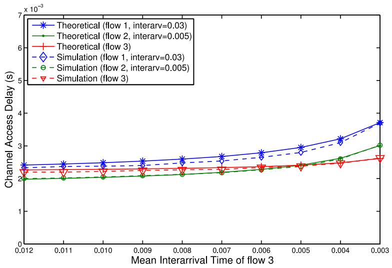


Fig. 3. Unsaturated conditions: service time versus $\frac{1}{\lambda_3}$, where $\frac{1}{\lambda_1} = 0.03$, $\frac{1}{\lambda_2} = 0.005$, and $CW_1 = CW_2 = CW_3 = 32$.

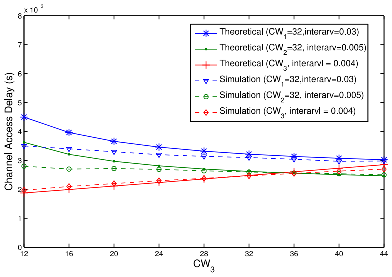


Fig. 4. Unsaturated conditions: service time versus CW_3, where $\frac{1}{\lambda_1} = 0.03, \frac{1}{\lambda_2} = 0.005, \frac{1}{\lambda_3} = 0.004$, and $CW_1 = CW_2 = CW_3 = 32$.

the collision probability becomes larger when every node has a small CW for contention resolution.

6.2.2 Service Time and Queueing Delays under Unsaturated Conditions

Three scenarios are examined. The first scenario studies how traffic arrival rates affect service times. The inter-arrival time of flow 3 is varied, while keeping the other arrival rates fixed with packet inter-arrival times $\frac{1}{\lambda_1} = 0.03$ and $\frac{1}{\lambda_2} = 0.005$. For the contention window, we set $CW_1 = CW_2 = CW_3 = 32$. Fig. 3 compares the numerical performance with the analysis by solving (12) using the **PE** algorithm.

In the second scenario, it is examined how CW affects the channel access delays. We fix the arrival rates of CW_1 and CW_2. Only CW_3 is changed from 12 to 44. From Fig. 4, we observe that, as CW_3 is increased, the delay of flow 3 increases, and the delays of flow 1 and flow 2 decrease.

The third scenario is used to demonstrate how service time changes 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. From Fig. 5, we observe that the access delays increase as the number of links grows.

6.2.3 Queueing Delays

We repeat the above three scenarios for the queueing delays in Fig. 6, 7, and 8 and we observe that the theoretical analysis are verified by the simulation results. The accuracy is not only reflected in the trend but also in the quantitative values.

6.3 Performance Evaluation

In the following, three case studies are examined to evaluate the performance of our proposed algorithms. Each point in the figures is a time-average of the queueing delay over every 50 simulation seconds.

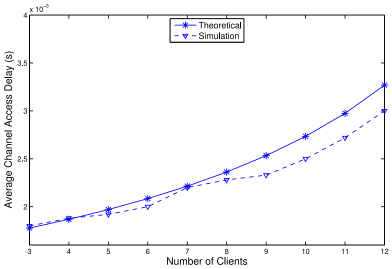


Fig. 5. Unsaturated conditions: service time versus the number of links, where $\frac{1}{\lambda_i} = 0.03$ and $CW_i = 32$ for all i.

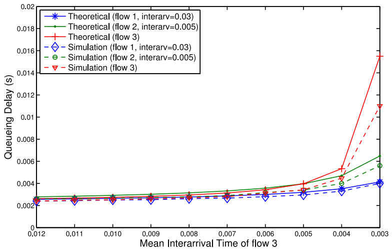


Fig. 6. Unsaturated conditions: Queueing delays versus $\frac{1}{\lambda_3}$, where $\frac{1}{\lambda_1} = 0.03, \frac{1}{\lambda_2} = 0.005$, and $CW_1 = CW_2 = CW_3 = 32$.

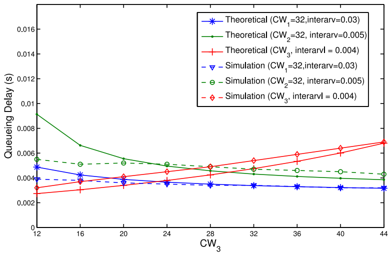


Fig. 7. Unsaturated conditions: Queueing delays versus CW_3, where $\frac{1}{\lambda_1} = 0.03, \frac{1}{\lambda_2} = 0.005, \frac{1}{\lambda_3} = 0.004$, and $CW_1 = CW_2 = CW_3 = 32$.

6.3.1 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. However, by optimizing the CW, one can find an appropriate setting in which all the delay requirements are met. The three required delays are assumed to be 0.02 seconds. This delay requirement is realistic in practice according to [19]. The data rates of the UDP traffic 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 is run to obtain a set of feasible contention windows: $CW_1 = 66$, $CW_2 = 23$, and $CW_3 = 18$. From Fig. 9, we see that the baseline default IEEE 802.11 setting only guarantees the mean delay requirements for flows 1 and 2, whereas the mean delay of flow 3 is much larger than the required mean delay. However, the WLAN can guarantee all the mean delays if the contention windows are appropriately adjusted. In fact, the mean delay specifications of all the flows are met when the contention windows are computed using our algorithms.

6.3.2 Minimizing Delays

In this case, the performance of our scheme to minimize the average delays for UDP traffic flows, while meeting the delay guarantees, is evaluated. We consider the following cost function

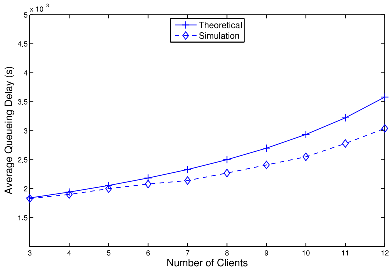


Fig. 8. Unsaturated conditions: Queueing delays versus the number of links, where $\frac{1}{\lambda_i} = 0.03$ and $CW_i = 32$ for all i.

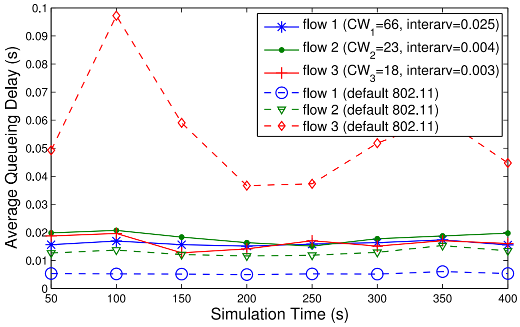


Fig. 9. Illustration of the queueing delay dynamics.

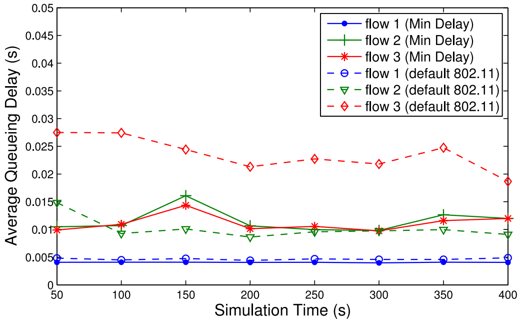


Fig. 10. Minimizing delays with the parameters: $\frac{1}{\lambda_1} = 0.04$, $\frac{1}{\lambda_2} = 0.004$, and $\frac{1}{\lambda_3} = 0.003$.

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

The UDP traffic 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 fixed at 0.02 seconds. As compared to the input in the first case, the network capacity is sufficient for this particular input. Thus, there is room for the flows to improve their performance (i.e., queueing delays in this case). Using the gradient algorithm presented in Section 5.2, the optimal CWs are computed to be $CW_1 = 19$, $CW_2 = 23$, and $CW_3 = 19$. The comparisons are plotted in Fig. 10. We observe that when configured with the CWs computed by our algorithms, the WLAN achieves the optimized mean delays and also provides a certain level of fairness. In contrast, using the baseline setting of IEEE 802.11, flow 3 suffers from bad delay performance and unfairness.

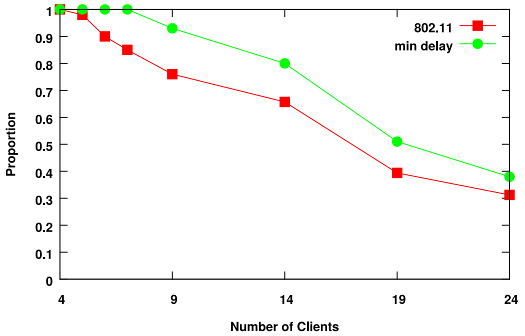


Fig. 11. Proportion of nodes satisfying the delay requirements (0.02 s) using the gradient algorithm with respect to the number of nodes in the network.

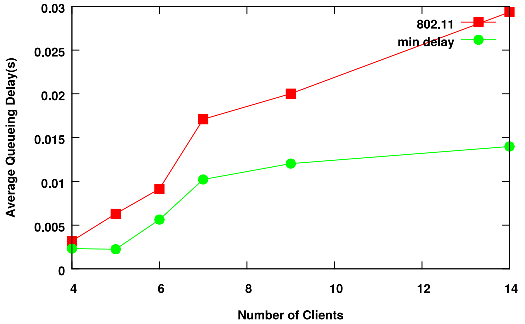


Fig. 12. Average queueing delay using the gradient algorithm with respect to the number of nodes in the network.

6.3.3 *Scaling Up with Nodes*

We study how the number of nodes affects the performance of the proposed algorithm. The delay requirements are still fixed at 0.02 seconds. The data rates of the UDP traffic, i.e., $1/\lambda_i, i = 1, \ldots, N$, are randomly generated between 0.05 and 0.1. Each simulation runs for 100 seconds. For a fixed number of nodes, we repeat each simulation for five times and compare the averages. If the number of the contention window is not feasible (smaller than 1), CW_{min} is used instead.

Fig. 11 shows the proportion of the nodes that meet the delay requirement (0.02 s), and Fig. 12 shows the average queueing delay. From both figures, we observe that the proposed algorithm demonstrates the ability to decrease the average queueing delay as comparing to the traditional 802.11 exponential back-off mechanism. Furthermore, we observe that the proposed algorithm can allow the nodes to meet the delay requirements when the number of nodes increases. This demonstrates that the proposed algorithm is useful even when the number of nodes increases beyond tens or twenties of nodes (a typical number in existing IEEE 802.11 WiFi networks).

7 Distributed Algorithm Design

In practice, it can be useful to find the appropriate contention window allocations in a distributed manner. In the following, we derive a distributed algorithm to adapt the service rates by adjusting the contention window of each node.

7.1 Design

We have defined three probabilities P_I^i, P_S^i and P_O^i in (2), (3), and (4), respectively. Let us assume that there is an *observer* who is monitoring the channel activities. The observer sees one of two possible states in a virtual slot: the channel is either idle during the virtual slot or busy due to the other nodes' transmission. Denote $\tilde{P}_I^i$ as the probability that the observer sees an idle slot. Then, we have

$$\tilde{P}_I^i = \prod_{j \neq i}^{N} (1 - \rho_j p_j). \tag{62}$$

Denote the number of consecutive idle slots between any two transmissions of node i as n_i. Since n_i follows a geometric distribution, the probability of the event that $n_i = k$ is

$$P[n_i = k] = \left(\tilde{P}_I^i\right)^k \left(1 - \tilde{P}_I^i\right), \tag{63}$$

and the expectation of n_i is

$$E[n_i] = \frac{\tilde{P}_I^i}{1 - \tilde{P}_I^i}. \tag{64}$$

Rearranging (64), we thus have

$$\tilde{P}_I^i = \frac{E[n_i]}{1 + E[n_i]}. \tag{65}$$

Now, substituting (62) into (12) yields

$$X_i = \frac{(1 - p_i)\tilde{P}_I^i \tau + (1 - \tilde{P}_I^i)T}{p_i \tilde{P}_I^i} + T. \tag{66}$$

By replacing $\tilde{P}_I^i$ in (66) by (65), we thus have

$$p_i(X_i - T + \tau) = \frac{T}{E[n_i]} + \tau. \tag{67}$$

Now, an unbiased estimation of $E[n_i]$ is the average of its samples. In particular, node i can count $n_i[k]$ and estimate $E[n_i]$ by

$$\bar{n}_i = \frac{\sum_{k=1}^{K} n_i[k]}{K}, \tag{68}$$

for a sufficiently large enough positive integer K. This suggests that one can use (67) to design a distributed algorithm since all the parameters, p_i, X_i and $\bar{n}_i$, are locally available at node i. Rearranging (67) and substituting $CW_i = 2/p_i$ yields

$$X_i = \frac{1}{2}\left(\frac{T}{\bar{n}_i} + \tau\right) CW_i + T - \tau. \tag{69}$$

If we consider CW_i as the variable, (69) defines a mapping from CW_i to X_i. Assume that the target access delay is X_i^*. We consider how to adapt CW_i so that $X_i(CW_i)$ meets X_i^* ideally. In this regard, we consider the following problem:

$$\min \sum_{i=1}^{N} (X_i^* - X_i(CW_i))^2 \tag{70}$$

$$\text{s.t.} \quad CW_i \geq 2 \;\; \forall i. \tag{71}$$

The target access delay is met when $X_i(CW_i)$ is equal to X_i^* for all i. As the objective is a quadratic function of CW_i, (70) is convex in the feasible region of $CW_i > 2$. Therefore, a gradient algorithm can be used to solve (70).

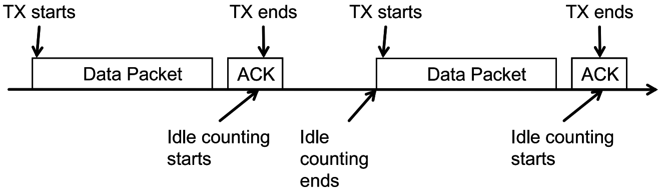


Fig. 13. Illustration of a node transmission (TX) start and end with the working of the idle counter.

The derivative of the objective function in (70) with respect to CW_i is

$$\frac{d}{d\,CW_i}\left(X_i^* - X_i(CW_i)\right)^2 = -(X_i^* - X_i(CW_i))\left(\frac{T}{\bar{n}_i} + \tau\right). \tag{72}$$

Thus, we propose the following algorithm to update CW_i:

$$\begin{aligned} &CW_i(t+1) \\ &= \max\left\{CW_i(t) + \alpha(X_i^* - X_i(CW_i(t)))\left(\frac{T}{\bar{n}_i} + \tau\right), 2\right\}, \end{aligned} \tag{73}$$

where α is an appropriately chosen stepsize [17].

In summary, CW_i is gradually driven to the optimal point such that the difference between X_i and X_i^* is minimized. The convergence proof of this gradient descent method to solve (70) optimally is standard, e.g., see [17].

7.2 Implementation Issues

In the following, we describe the implementation of (73) in the NS-2 simulator. First, we introduce a counting mechanism to count the number of idle slots between two consecutive transmissions of node i that we denote by $\bar{n}_i$. The contention window is then updated from (73) based on $\bar{n}_i$.

7.2.1 *Idle Counter*

The channel state changes when a packet transmission begins or ends, and each node can carrier-sense this change. When the channel state changes, the *idle counter* is triggered to count the number of idle slots between two consecutive transmissions. We illustrate how the idle counter works in Fig. 13. First, a wireless node senses the channel, and detects that a transmission has ended. The idle counter is reset and increments at every slot time (e.g., $20\,\mu s$); when it detects a new transmission, the idle counter stops and computes the average number of idle slots $\bar{n}_i$. Denote the counter at the jth count as $c[j]$; then we can update $\bar{n}$ by an exponential-moving average update:

$$\bar{n}[j+1] = (1 - \eta)\bar{n}[j] + \eta c[j], \tag{74}$$

where the parameter η is a constant between 0 and 1.

7.2.2 *Contention Window Adaptation*

Using the delay specifications (or deadlines) provided by upper layer applications, we compute X_i by (26), and node i then adapts its contention window CW_i using (73), where $\bar{n}$ is the value of the idle counter.

7.2.3 *Performance of Distributed Algorithm*

We evaluate the performance using NS-2 simulation. In our simulation, the data rates of the UDP traffic flows are fixed

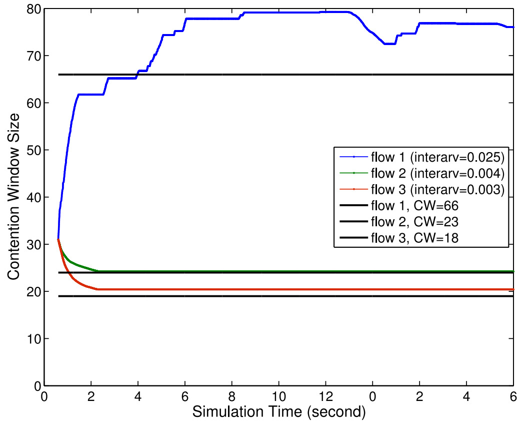


Fig. 14. Convergence of the distributed algorithm for contention window adaptation.

as: $\frac{1}{\lambda_1} = 0.025, \frac{1}{\lambda_2} = 0.004$ and $\frac{1}{\lambda_3} = 0.003$. Assume that the delay specifications of these flows are 0.02 second. The contention windows are computed by the centralized algorithm **ARA** for comparison: $CW_1 = 66$, $CW_2 = 23$, and $CW_3 = 18$. These contention windows are shown by the solid lines in Fig. 14. The evolution of the contention windows is also plotted. We observe that, starting from an initial value 31, each contention window gradually moves towards its equilibrium value. The contention windows of flow 2 and flow 3 converge close to the equilibrium value within a few seconds. However, the convergence of flow 1 is not as obvious as the other two. Since the contention window of flow 1 is much larger than the other two, it fluctuates within a larger range. The other two contention windows fluctuate within a smaller range and eventually converge. This fluctuation is an artifact of the gradient algorithm as we use a constant small stepsize α in our implementation. We have also studied the performance of the algorithm when the number of nodes increases. We observe that the delay performance of the distributed algorithm is comparable to that of the centralized algorithm except that the distributed algorithm converges much slower (after a few more seconds).

8 Video-Based Experimental Evaluation

In this section, we use a video transmission simulator EvalVid [20] to evaluate the performance of our algorithms when MPEG video is transmitted.

8.1 Overview of EvalVid

The structure of the EvalVid framework [20], [21] is shown in Fig. 15 that illustrates how EvalVid measures the video quality-of-service metrics. For more information, we refer the readers to [20], [21].

8.2 Video Distortion

We focus on three key metrics associated with video distortion for three types of video frames, namely, I-frames, P-frames, and B-frames.

- Packet loss: For each type of data, we compute the packet loss rate as follows. Let V denote the type of data in the packet (one of I, P, B frames), PR_V denote the number of type V packets received, and PS_V denote the number of type V packets sent, then the packet loss rate PL_V is given by

$$PL_V = \frac{PR_V}{PS_V} \times 100\%. \tag{75}$$

- Frame loss: The frame loss calculation is introduced because we cannot easily deduce the frame loss rate from packet loss rate. Let FR_V denote the number of type V frames received and FS_V denote the number of type V frames sent, then the frame loss rate is

$$FL_V = \frac{FR_V}{FS_V} \times 100\%. \tag{76}$$

- Delay and jitter: Frames in digital videos have to be displayed at a constant rate. Displaying a frame before or after the required time specification leads to a phenomenon called "jerkiness" [22]. This issue is addressed by play-out buffers that can be used to absorb the jitter introduced by network delays, and the buffer size is predefined based on the playback time in our experiments.

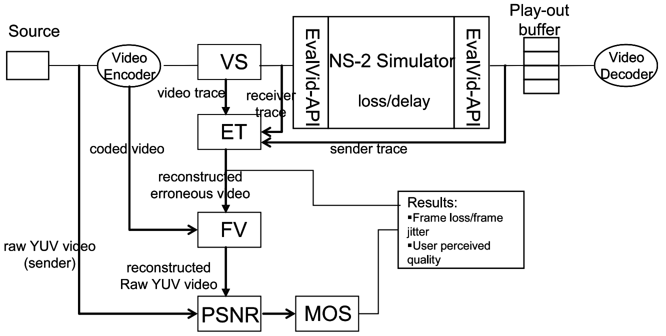


Fig. 15. Overview of the video transmission simulation using the EvalVid system.

8.3 Experiment Design

Our experiments use real video data downloaded from the Video Trace Library [23]. The video traces are provided in two formats, namely YUV QCIF (176 × 144) and YUV CIF (352 × 288). Since these two formats have different resolutions, the amount of data that needs to be transmitted per unit time are different for the same video sequence. For example, the size of 300-frame-long Foreman video in YUV CIF is 44M, whereas the size of the same video in YUV QCIF is 11M. For comparison purpose, we conduct two experiments using different algorithms in EvalVid. We use the default IEEE 802.11 DCF for the first experiment and use the proposed algorithm for the second experiment. In both experiments, we use the two formats of video sources. Furthermore, we set MTU to 1,000 bytes. The packet arrivals of these two video formats are shown in Figs. 16 and 17. The video frame is sent at every 33.33 ms in 30 frames/sec video traffic.

The first experiment is described as follows: we assume there are two groups of users. Users in Group 1 download while playing a video in YUV CIF format. Users in Group 2 download while playing the same video in YUV QCIF format. Each group consists of two users. All the users adopt

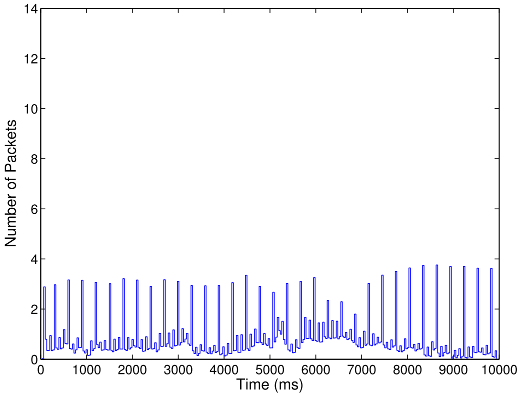


Fig. 16. Packet Arrivals versus Time: Foreman video in YUV QCIF.

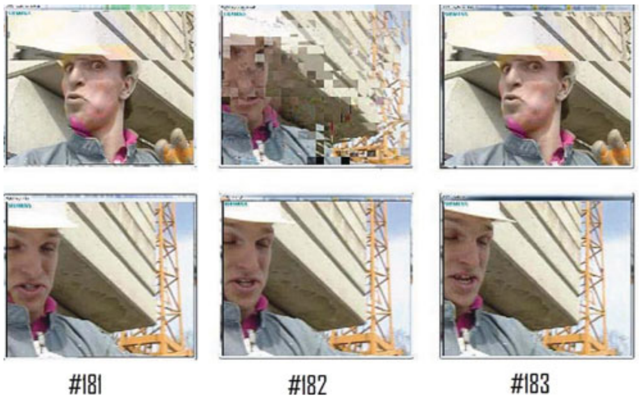


Fig. 18. The illustration of video quality for different contention windows. The number below indicates the video frame number.

the default IEEE 802.11 DCF. Each video provider first encodes the YUV file to obtain the compressed MPEG-4 file. Then, users use MP4.exe (from Evalvid) to record the tracefiles for a sender. The tracefiles are linked to a NS-2 UDP agent attached to the sender node. Simulated packets are generated based on the tracefiles and then sent to the receiver. The sender records the time instant when these packets are sent out and the receiver records the time instant when these packets are received. At the end of the simulation, these two records produce the received compressed MPEG-4 file. Finally, the decoder decodes the file and reconstructs the received video.

The second experiment differs from the first one only in using the proposed algorithm. In comparison, the improvement can be observed from the quality of the received video as shown in Fig. 18.

Three consecutive frames are selected from the received video in group 1. The upper top three frames are taken from the video transmitted using the default IEEE 802.11 DCF setting, and the bottom three frames are taken from the video transmitted using parameters derived from our scheme. Observe that the bottom three frames have negligible distortion, because the contention windows have been adjusted to guarantee the delay requirement of the video in CIF format. In contrast, we see that the upper top three frames are corrupted, and thus some frames are displayed incorrectly. If a frame is lost during the transmission, the decoder fills the gap with the most recent successfully-decoded frame. Thus, an old frame appears repetitively in some instances.

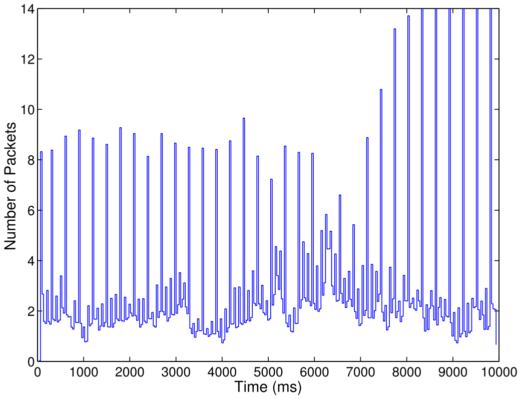


Fig. 17. Packet Arrivals versus Time: Foreman video in YUV CIF.

We also compare the (peak signal to noise ratio (PSNR) results for the two experiments as shown in Fig. 19. These experiments demonstrate that our scheme can improve the quality of service even if the packet arrivals do not strictly follow a Poisson distribution and that it works relatively well in practice for realistic applications such as live video players.

9 Conclusions

We have presented a simple and accurate model to analyze queueing delays in non-homogeneous IEEE 802.11 MAC based WLANs. Our queueing analysis allows us to study the feasibility problem of whether the network can provide the mean delay guarantees required by non-homogeneous QoS flows in transmission. Furthermore, in order to optimize the performance of QoS flows, we have proposed centralized and distributed algorithms to minimize the mean queueing delays for a set of UDP traffic flows while meeting mean delay guarantees. Extensive NS-2 simulations and video trace-based experiments have been conducted to verify the accuracy of the queueing model and to evaluate the performance of the proposed algorithms.

Appendix

A Sufficient Condition for the Approximation in (33) The right-hand side of (27) is given by

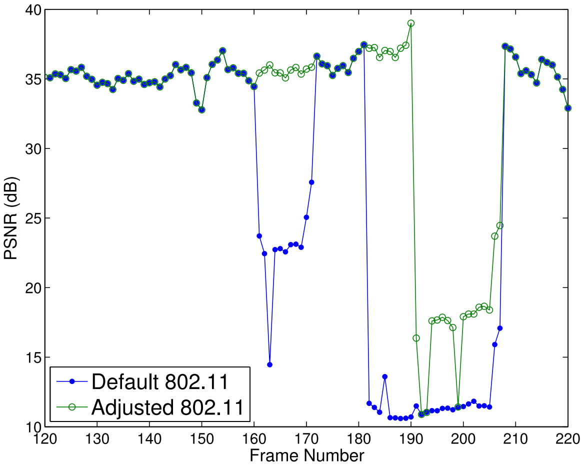


Fig. 19. The evolution of PSNR for different video frames.

$$\begin{aligned}
&\frac{T}{\prod_{j\neq i}(1-\lambda_j X_j p_j)}\\
&\overset{(a)}{\leq} \frac{T}{1-\sum_{j\neq i}\lambda_j X_j p_j}\\
&\overset{(b)}{=} T\left(1+\sum_{j\neq i}\lambda_j X_j p_j+\left(\sum_{j\neq i}\lambda_j X_j p_j\right)^2+\cdots\right)\\
&\overset{(c)}{=} T\left(1+\sum_{j\neq i}\lambda_j X_j p_j+\frac{\left(\sum_{j\neq i}\lambda_j X_j p_j\right)^2}{1-\sum_{j\neq i}\lambda_j X_j p_j}\right)\ \forall i,
\end{aligned}\tag{77}$$

where inequality (a) is true when $\sum_i \lambda_i X_i p_i < 1$ and $\lambda_i X_i p_i > 0$ for all i, equality (b) is the Taylor series expansion for (a), and equality (c) is due to the sum from the third term to infinity in the outer bracket of equality (b). Suppose that we want $\frac{(\sum_{j\neq i}\lambda_j X_j p_j)^2}{1-\sum_{j\neq i}\lambda_j X_j p_j} < \sum_{j\neq i}\lambda_j X_j p_j$. Then, this is equivalent to $\sum_{j\neq i}\lambda_j X_j p_j < \frac{1}{2}$ for $i=1,\ldots,N$. Now, adding up these N equations yield a single equation $\sum_{i=1}^{N}\lambda_i X_i p_i < \frac{N}{2(N-1)}$ that can be satisfied by a sufficient condition $\lambda_i X_i p_i < \frac{1}{2(N-1)}$ $\forall i$.

ACKNOWLEDGMENTS

This material is based upon work partially supported by the US National Science Foundation (NSF) Contract CNS-1302182, AFOSR Contract FA9550-13-1-0008, NSF Science & Technology Center Grant CCF-0939370, the Research Grants Council of Hong Kong, Project No. RGC CityU 11212114, and the National Research Foundation Fellowship NRF-NRFF2015-07. Part of this work was presented in [1] at the IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, April 10-15, 2011.

REFERENCES

[1] Y. Gao, C. W. Tan, Y. Huang, Z. Zeng, and P. Kumar, "Feasibility and optimization of delay guarantees for non-homogeneous flows in IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, 2011, pp. 2660–2668.

[2] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[3] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

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

[5] 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, pp. 1–12.

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

[7] O. Tickoo and B. Sikdar, "Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 878–891, Aug. 2008.

[8] L. Feng, J. Q. Li, and X. D. Lin, "A new delay analysis for IEEE 802.11 PCF," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 4064–4069, Oct. 2013.

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

[10] 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," in *Proc. ACM SIGCOMM*, 2005, pp. 121–132.

[11] L. B. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[12] Y. Gao, D. M. Chiu, and J. C. S. Lui, "Determining the end-to-end throughput capacity in multi-hop networks: Methodology and applications," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 39–50, 2006.

[13] A. Wald, "Sequential tests of statistical hypotheses," *Ann. Math. Statist.*, vol. 16, no. 2, pp. 117–186, 1945.

[14] L. Kleinrock, *Queueing Systems*. New York, NY, USA: Wiley, 1975.

[15] H. Kobayashi, B. L. Mark, and W. Turin, *Probability, Random Processes, and Statistical Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[16] E. Seneta, *Non-Negative Matrices and Markov Chains*. New York, NY, USA: Springer, 2006.

[17] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[18] The network simulator-ns2 [Online]. Available: http://www.isi.edu/nsnam/ns, 2015.

[19] Understanding delay in packet voice networks [Online]. Available: http://www.cisco.com/, 2006.

[20] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid—A framework for video transmission and quality evaluation," in *Proc. 13th Int. Conf. Comput. Perform. Eval. Model. Techn. Tools*, 2003, pp. 255–272.

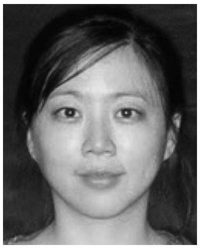
[21] C. H. Ke, C. K. Shieh, W. S. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video transmission," *J. Inf. Sci. Eng.*, vol. 24, no. 2, pp. 425–440, 2008.

[22] S. Wolf and M. Pinson, "Video quality measurement techniques," U.S. Dept. Commerce, NTIA, Washington, D.C., USA, Tech. Rep. TR-02-392, 2002.

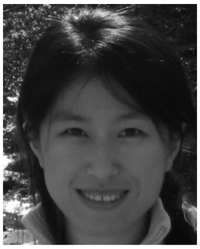
[23] Video trace library for network performance evaluation [Online]. Available: http://trace.eas.asu.edu/index.html, 1999.

Yan Gao received the BE degree in EE from the University of Science and Technology of China in 2003, the MPhil degree in information engineering from the Chinese University of Hong Kong in 2006, and the PhD degree in computer science from the University of Illinois at Urbana Champaign in 2011. He joined Accenture Technology Labs in November 2011 as an associate R&D manager. He is leading the research effort on intelligent transportation systems, smart city, and IIoT analytics. His recent research is focused on exploiting cellular network infrastructure data to track people/vehicles' movement and exploring both business and technology innovations through understanding the movements. He is a member of the IEEE.

Chee Wei Tan (M'08-SM'12) received the MA and PhD degrees in electrical engineering from Princeton University, Princeton, NJ, in 2006 and 2008, respectively. He is an associate professor at the City University of Hong Kong. Previously, he was a postdoctoral scholar at the California Institute of Technology (Caltech), Pasadena, CA. He was a visiting faculty at Qualcomm R&D, San Diego, CA, in 2011. His research interests are in networks, inference in online large data analytics, and optimization theory and its applications. Dr. Tan was the recipient of the 2008 Princeton University Wu Prize for Excellence and was awarded the 2011 IEEE Communications Society AP Outstanding Young Researcher Award and the 2015 National Research Foundation Fellowship. He was the chair of the IEEE Information Theory Society Hong Kong Chapter in 2014-2015. He was twice selected to participate at the US National Academy of Engineering China-America Frontiers of Engineering Symposium in 2013 and 2015. Dr. Tan currently serves as an Editor for the *IEEE Transactions on Communications*. He is a senior member of the IEEE.



Ying Huang received the bachelor's degree in computer science from Peking University in 2004, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2010. She is a software engineer at Google. She has been working on many products across search, social, display ads, and youtube. Her research interests are data management and content sharing for delay tolerant networks, topology control, mobility models, security, and QoS. She is a member of the IEEE.



Zheng Zeng received the BE degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2005; and the PhD degree from the Department of Computer Science, University of Illinois at Urbana-Champaign in 2011. She is currently working in Apple Inc. She is a member of the IEEE.

P.R. Kumar is at Texas A&M University. His current research is focused on energy systems, wireless networks, secure networking, automated transportation, and cyber-physical systems. Dr. Kumar is a member of the National Academy of Engineering of the USA, and a fellow of the World Academy of Sciences. He received an honorary doctorate by the ETH, Zurich. He received the Outstanding Contribution Award of ACM SIGMOBILE, the IEEE Field Award for Control Systems, the Donald P. Eckman Award of the American Automatic Control Council, and the Fred W. Ellersick Prize of the IEEE Communications Society. He is an honorary professor at IIT Hyderabad, and a D.J. Gandhi distinguished visiting professor at IIT Bombay. He is fellow of the ACM and IEEE.