A Theory of QoS for Wireless

I-Hong Hou
Department of Computer Science
University of Illinois
Urbana, IL 61801, USA
ihou2@illinois.edu

Vivek Borkar
Tata Institute of Fundamental Research
Mumbai 400 005, India
borkar@tifr.res.in

P. R. Kumar
CSL and Department of ECE
University of Illinois
Urbana, IL 61801, USA
prkumar@illinois.edu

***Abstract*—Wireless networks are increasingly used to carry applications with QoS constraints. Two problems arise when dealing with traffic with QoS constraints. One is *admission control*, which consists of determining whether it is possible to fulfill the demands of a set of clients. The other is finding an *optimal scheduling policy* to meet the demands of all clients. In this paper, we propose a framework for jointly addressing three QoS criteria: *delay*, *delivery ratio*, and *channel reliability*.**

We analytically prove the necessary and sufficient condition for a set of clients to be feasible with respect to the above three criteria. We then establish an efficient algorithm for admission control to decide whether a set of clients is feasible. We further propose two scheduling policies and prove that they are *feasibility optimal* in the sense that they can meet the demands of every feasible set of clients. In addition, we show that these policies are easily implementable on the IEEE 802.11 mechanisms. We also present the results of simulation studies that appear to confirm the theoretical studies and suggest that the proposed policies outperform others tested under a variety of settings.

I. Introduction

Wireless networks have been widely deployed for a variety of purposes. Among the many applications that benefit from wireless networks, those with quality of service (QoS) constraints are increasingly of interest. They include video streaming, VoIP, real-time monitoring, networked control, etc. One common characteristic of these applications is that they have some requirements on throughput, delay, and delivery ratio. Hence, most current network mechanisms, which only provide "best-effort" service, are not adequate for these applications.

While there has been much research interest in providing QoS, there is a dearth of analytical studies and theoretical guarantees on the service that can be provided. A fundamental difficulty is that it is important to specifically take into account a most important feature of wireless network, that is, the lossy channel. As more and more devices, such as cordless phones, Bluetooth and Zigbee devices, are accessing the same unlicensed channel as wireless networks, packet loss can no longer be neglected.

We provide an analytical framework for addressing QoS constraints in wireless networks that allows the incorporation of three criteria with each flow: delay, delivery ratio, and channel reliability. We first identify a necessary condition for a set of flows to be feasible with respect to the above three QoS criteria. Next, two dynamic scheduling policies for these applications are proposed. We prove that the proposed policies can meet the demand of every set of flows that satisfies the identified necessary condition. Thus, we not only show that the necessary condition is indeed sufficient, but we also prove our proposed policies are optimal. Finally, while the necessary and sufficient condition involves exponentially many inequalities to be checked, we show that those criteria can be simplified into linearly many tests. We thus obtain an efficient admission control algorithm for flows with QoS.

Our contribution is therefore threefold. First we propose a mathematical framework for QoS for handling deadlines, delivery ratios and channel unreliability. Second, the linear time algorithm makes admission control computationally efficient. Third, the simple nature of the policies proved to be feasibility optimal shows that scheduling for QoS is tractable and feasible at run time.

In addition to the theoretical results, we also evaluate the proposed policies by simulation. We implement the two policies by widely-used IEEE 802.11 mechanisms. We compare the two policies against the IEEE 802.11 Distributed Coordination Function (DCF) and a server-centric scheduler that gives equal transmission opportunities to all clients. Simulation results suggest that the proposed policies offer much better service than the two compared mechanisms.

The rest of the paper is organized as follows: Section II summarizes some existing work on providing QoS. Section III formally models the wireless channels and formulates a framework for addressing QoS constraints. Based on the formulation, Section IV develops some preliminary results that provide insights into designing scheduling policies and employing admission control. In Section V, we propose two scheduling policies. We prove they are both feasibility optimal in Section VI. In addition, in Section VII, we propose an efficient admission control algorithm. In Section VIII, we show how to implement the

This material is based upon work partially supported by USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, NSF under Contract Nos. ECCS-0701604, CNS-07-21992, CNS-0626584, and CNS-05-19535, and a grant from General Motors India Lab. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the above agencies.

proposed scheduling policies on IEEE 802.11. Simulation results are described in Section IX. Section X concludes the paper.

II. Related Work

Scheduling policies for QoS support on error-prone wireless channels have been increasingly of interest in recent years. Tassiulas and Ephremides [17] have proposed a max weight scheduling policy and proved that it is throughput optimal. Neely [10] have further evaluated this policy and have shown that the policy achieves a constant average delay. Shakkottai and Stolyar [14] have evaluated various scheduling policies to support a mixture of real-time and non-real-time traffic. Johnsson and Cox [7] have proposed a policy that aims to achieve both small packet delay and high user throughput. Dua and Bambos [4] have focused on the trade-off between user fairness and system performance and designed a policy for this purpose. However, all these works lack a thorough theoretical study with provable performance guarantees. Raghunathan et al [12] and Shakkottai and Srikant [13] have developed analytical results on scheduling. However, the goal of their works is to minimize the total number of expired packets among all users, which will inevitably be unfair to clients with poor channel qualities. Stolyar and Ramanan [16] aim at offering QoS guarantees on a per-client basis. Their approach offers asymptotic optimality only when the period is large. Kawata et al [8] have focused more on implementation issues and enhancing QoS for the IEEE 802.11 mechanisms. Their simulations have been conducted in a controlled environment where packet losses are infrequent. Other works [1] [2] have considered different performance metrics and modeling assumptions.

Compared to scheduling policies, there are fewer analytical studies on admission control. Xiao et al [18] and Pong el al [11] have proposed admission control algorithms to guarantee a certain amount of bandwidth for each user but do not take into account any latency constraints. Garg et al [5], Zhai et al [19], and Shin and Schulzrinne [15] have used various metrics to predict system performance statistically but lack a theoretical study.

III. A Model For QoS

We consider a system with N wireless clients and one access point (AP). Each client wishes to transmit packets to the AP with some QoS constraints. It is assumed that time is slotted. At the beginning of a time slot, the AP broadcasts a control message, indicating which client can transmit in the time slot. The assigned client then sends out a packet if it has a packet waiting to be transmitted. The size of a time slot is the time required for the AP to send the control packet plus the time for a client to transmit a data packet. While this model appears to assume that there is no traffic from the server to clients, we show in Section VIII-B that all the results are applicable to the case where clients require QoS-constrained traffic from the server.

The QoS constraints for a client are described as follows: At the beginning of every period of length τ, where the length is measured in time-slots, each client $n \in \{1, 2, \cdots, N\}$ generates one data packet. The packets of all clients are to be delivered to the AP within the next τ time slots before the end of the period. If a packet is not delivered by the end of the period, it is marked as expired and removed from the system. Thus, we can guarantee that the delay of each delivered packet is less than τ. Further, client n requires a delivery ratio of at least q_n. That is, the proportion of expired packets cannot exceed $1 - q_n$. Finally, reflecting the nature of the unreliable wireless channels, client n has a channel reliability of p_n; that is, the proportion of transmissions of client n that are successfully delivered to the AP is p_n. This channel reliability reflects qualities of both uplink and downlink since a successful transmission includes the delivery of both the control message by the server and the data packet by the designated client. The value of p_n can be obtained by probing messages before the client is admitted into the system and updated as long as the client stays in the system. The different values of p_n for different clients also reflect the fact that wireless links are not homogeneous and vary in quality from user to user. The decision on which client is chosen to transmit on a slot is specified by a scheduling policy which makes the decision causally based on the entire past history of events up to that time slot.

We wish to provide a service for clients with QoS constraints as described above.

Definition 1: A set of clients with the above QoS constraints is said to be *fulfilled* by a particular scheduling policy η if the time averaged delivery ratio of each client n is at least q_n with probability 1. [1]

Due to the limited wireless resource, the requested QoS demands of the set of clients may exceed the capacity of the wireless network. In this case, a service that aims to fulfill all clients may end up providing poor performance. Therefore, a desired service must incorporate some admission control mechanism. To perform admission control, it is vital to verify whether a set of clients is *feasible*:

Definition 2: A set of clients is *feasible* if there exists some scheduling policy that fulfills it.

In addition to an efficient admission control mechanism,we also aim to design a *feasibility optimal* scheduling policy:

Definition 3: A scheduling policy is said to be *feasibility optimal* if it fulfills every feasible set of clients. [2]

[1] More formally, $\liminf_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} 1(\text{A packet of client } n \text{ is delivered successfully in period } k) \geq q_n$, with probability one, for each client $n = 1, 2, \cdots, N$, where $1(\cdot)$ is the indicator function of the event.

[2] This is analogous to the notion of "throughput optimality" in queuing systems.

IV. Necessary Condition For Feasibility Of QoS

It is quite evident that the more time slots we allocate to a particular client, the more likely it is that we can meet the demand of that client. We further observe that whether the demand of a client is met is only related to the proportion of time slots in which the client is transmitting.

Lemma 1: The delivery ratio of client n is at least q_n with probability 1 if and only if the long-term time average of the proportion of time slots that client n is transmitting is at least $w_n = \frac{q_n}{p_n \tau}$.

Proof: Define:

$$u_n(t) = \begin{cases} 1, & \text{if client } n \text{ makes a transmission at time } t, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$d_n(t) = \begin{cases} 1, & \text{if client } n \text{ delivers a packet at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathfrak{F}_t$ be the σ-algebra generated by $\{(u_n(k), d_n(k-1)), \text{for } 1 \le k \le t \text{ and } 1 \le n \le N\}$. (We set $d_n(0) = 0$ for all n.)

Then $E[d_n(t)|\mathfrak{F}_t] = p_n u_n(t)$. Hence, by the martingale stability theorem of Loeve [9],

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} [d_n(t) - p_n u_n(t)] = 0, \text{ a.s.} \tag{1}$$

Since the delivery ratio of client n must be at least q_n,

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} d_n(t) \ge \frac{q_n}{\tau}, \text{ a.s.}$$

Hence, $\liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} u_n(t) \ge \frac{q_n}{p_n \tau}$ from (1). ∎

We will hereafter refer to w_n as the *implied work load* for client n. Determining whether client n is fulfilled is therefore equivalent to deciding whether the share of time that client n gets is at least as large as its implied work load. This helps to partially decouple the clients.

We next study whether it is possible to fulfill a set of clients. Since there is at most one client that can transmit in any time slot, we immediately obtain the following necessary condition:

Lemma 2: A set of N clients can be feasible only if $\sum_{n=1}^{N} w_n \le 1$.

This necessary condition, however, is not sufficient. Since each client only generates one packet in each period, it might be the case that at some slot in a period, all packets in the system are delivered and the system is forced to stay idle. (Recall that expired packets are deleted from the system at the end of a period, and so only new packets are available in a system at the beginning of each period). While the amount of time that the system is idling may be influenced by the scheduling policy, we show it is the same for some particular policies:

Definition 4: A scheduling policy is *work conserving* if it never idles whenever there is an undelivered packet in the system.

Lemma 3: The probability distribution of the amount of time that the system is idling in any period is identical for every work conserving policy.

Proof: Let the random variable γ_n denote the number of slots a packet of client n is transmitted before it is delivered. The distribution of γ_n is $Prob(\gamma_n = t) = p_n(1-p_n)^{t-1}$. Under any work conserving policy, the number of idle time slots in a period, L, is the number of time slots left after all packets in the system are delivered:

$$L = \begin{cases} \tau - \sum_{n=1}^{N} \gamma_n, & \text{if } \sum_{n=1}^{N} \gamma_n < \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Hence its probability distribution is the same under every work conserving policy. ∎

The following observation allows us to focus only on work conserving policies when designing a feasibility optimal policy:

Lemma 4: Let η be a scheduling policy that can meet the demands of a particular set of clients. Then there exists a work conserving policy η' that can also meet the requirements of the same set of clients.

Proof: Policy η can be extended to be a work-conserving policy η' by simply filling slots that η would leave idle by transmitting any undelivered packets in arbitrary order. This cannot reduce the number of undelivered packets in any period. ∎

Note that for at least $E[L]$ number of slots of each period, on average, the server must be idle. The fraction of idle time is therefore at least $\frac{E[L]}{\tau}$. Hence we can improve the necessary condition in Lemma 2 to

$$\sum_{n=1}^{N} w_n \le 1 - \frac{E[L]}{\tau}. \tag{2}$$

However, we can go even further by considering subsets of the set of all clients $\{1, 2, \cdots, N\}$. For any subset $S \subseteq \{1, 2, \cdots, N\}$, let

$$I_S := \frac{E[\max\{0, \tau - \sum_{n\in S} \gamma_n\}]}{\tau}.$$

It is a lower bound on the fraction of time spent idling, if S were the set of all clients. Clearly if $\{1, 2, \cdots, N\}$ is feasible, then every subset S must be feasible. Hence we can tighten the condition (2):

Lemma 5: A set of clients is feasible only if $\sum_{n\in S} w_n \le 1 - I_S$ holds for every subset S.

The reason why the condition for a subset S is not subsumed by the condition for all clients is that $\sum_{n\in S} w_n$ is monotone increasing in S, while I_S is monotone decreasing in S. Surprisingly, we will show that the above necessary condition is actually sufficient in Section VI.

V. Scheduling Policies

In this section, we propose two scheduling policies on providing QoS. Both policies are what we call *largest debt*

first policies. The idea of a largest debt first policy is to compute the *debt* owed to each client at the beginning of every period. The server then determines the priorities of clients according to their debts, a clients with larger debt getting a higher priority. Ties are broken by lexicographic order. In each time slot of the period, the client with the highest priority among those who have not yet succeeded in a transmission is scheduled to transmit. The only difference between the two policies lies in the definitions of debts.

The first policy, which we call the *largest time-based debt first* policy, tries to make every client get a share of time at least as large as its implied work load. To see how much a client lags behind its implied work load, we define debt as follows:

Definition 5: The *time-based debt* of client n at time t is defined as $t \times w_n$ minus the actual number of time slots that client n has transmitted by time slot t. The policy which assigns priorities accordingly is the *largest time-based debt first policy*.

The next policy, which we call the *largest weighted-delivery debt first* policy, approaches the QoS requirements more directly. It seeks to make every client have a success rate higher than the desired delivery ratio, that is, q_n.

Definition 6: Let $c_n(t)$ be the number of successful transmissions of client n up to time t. The *weighted-delivery debt* of client n at time t is defined as $(\frac{t}{\tau} \times q_n - c_n(t))/p_n$. The policy which assigns priorities accordingly is called the *largest weighted-delivery debt first policy*.

VI. Proofs of Optimality

We prove that the two largest debt first policies are feasibility optimal policies. The proof is based on Blackwell's approachability theorem [3]. We first describe the content of this theorem.

Consider a single-player repeated game. The payoff, instead of being a single number, is an N-dimensional vector, $v \in \mathbb{R}^N$. In each stage i, the player picks an action, $a(i)$, according to some history dependent policy, based on past actions $a(1), \cdots, a(i-1)$, and past payoffs in the previous stages, $v(1), \cdots, v(i-1)$. The payoff of this stage, $v(i)$, is a random vector whose distribution is a given function, which we call the *payoff function*, of the action $a(i)$ taken. We are interested in the distribution of the long-term average payoff, $\lim_{j\to\infty} \sum_{i=1}^{j} v(i)/j$. To study the problem, Blackwell introduced the concept of *approachability*:

Definition 7: Let $A \subseteq \mathbb{R}^N$ be any set in N-dimensional space. We shall say A is approachable under policy η, if for every $\delta > 0$ and $\varepsilon > 0$ there is a j_0 such that,

$$Prob\{\rho(j) \geq \delta \text{ for some } j \geq j_0\} \leq \varepsilon,$$

where $\rho(j)$ denotes the distance of the point $\sum_{i=1}^{j} v(i)/j$ from A. In other words, this means the long-term average payoff will approach A with probability 1.

Let $R(a)$ be the expected payoff of action a, i.e., the expected value corresponding to the probability distribution of the payoff function. Blackwell's sufficient condition for approachability is the following:

Theorem 1: Let $A \subseteq \mathbb{R}^N$ be any closed set. If for every $x \notin A$ there is an action $a\ (= a(x))$ such that the hyperplane through y, the closest point in A to x, perpendicular to the line segment xy, separates x from $R(a)$, then A is approachable with the policy η which uses action $a(x_j)$ if $x_j = (\sum_{i=1}^{j} v(i)/j) \notin A$, and an arbitrary action otherwise.

Since every feasible set of clients has to satisfy the necessary condition stated in Lemma 5, it suffices to show that both policies fulfill every set of clients that satisfies the necessary condition.

Theorem 2: The largest time-based debt first policy is feasibility optimal.

Proof: We first translate the largest time-based debt first policy into a policy for the single player game. A stage in the game corresponds to a period in our model. The action that the player, which is the server in our system, can take, is to decide the priorities of the clients, with the interpretation that a client with a certain priority is transmitted only after the client with the immediately higher priority is successful. The payoff a player gets in the stage is the net change of the time-based debt of each client. To be more precise, the payoff is a vector whose n^{th} element equals τw_n minus the actual number of time slots that client n is transmitting in this period.

Lemma 1 shows that a group of clients is fulfilled if every of them gets an average share of time at least w_n. In terms of approachability, this means a group of clients is fulfilled if the set $A := \{z = [z_1, z_2, \cdots, z_N] | z_n \leq 0, \forall n\}$ is approachable.

Now we apply Theorem 1. Suppose at the beginning of some period the average payoff $x = [x_1, x_2, \cdots, x_N]$ does not lie in A. We can reorder the clients so that $x_1 \geq x_2 \geq \cdots \geq x_m > 0 \geq x_{m+1} \geq \cdots \geq x_N$. The closest point in A to x is $y = [0, 0, \cdots, 0, x_{m+1}, x_{m+2}, \cdots, x_N]$. The hyperplane H through y and perpendicular to the line segment xy is comprised of points z satisfying $h(z) := \sum_{n=1}^{m} x_n z_n = 0$.

Let $\bar{x}$ be the expected payoff in this round according to the largest time-based debt first policy. Also, let $\bar{w}_n$ be the expected portion of time slots that client n is transmitting during this period. We can now express

$$\begin{aligned}\bar{x} &= [\tau w_1 - \tau \bar{w}_1, \tau w_2 - \tau \bar{w}_2, \cdots] \\ &= \tau [w_1 - \bar{w}_1, w_2 - \bar{w}_2, \cdots].\end{aligned}$$

Since $h(x) = \sum_{n=1}^{m} x_n^2 > 0$, in order to show that H separates x and $\bar{x}$, it suffices to show $h(\bar{x}) \leq 0$. We have

$$\begin{aligned}h(\bar{x}) &= \tau \sum_{n=1}^{m} x_n (w_n - \bar{w}_n) \\ &= \tau \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} w_k - \sum_{k=1}^{n} \bar{w}_k)] \\ &\quad + \tau x_m (\sum_{k=1}^{m} w_k - \sum_{k=1}^{m} \bar{w}_k).\end{aligned}$$

By the largest time-based debt first policy, the server will give priorities according to the ordering $1, 2, \cdots, N$. Hence, $\sum_{k=1}^{n} \bar{w}_k$ will be the average portion of time the system spends working if only clients 1 through n are present in the system. In other words, $\sum_{k=1}^{n} \bar{w}_k = 1 - I_{\{1,2,\cdots,n\}}$. Now, according to the necessary condition stated in Lemma 5, we have $\sum_{k=1}^{n} w_k \leq 1 - I_{\{1,2,\cdots,n\}} = \sum_{k=1}^{n} \bar{w}_k$. Further, $x_1 \geq x_2 \geq \cdots \geq x_m > 0$. Hence, $h(\bar{x}) \leq 0$ and the largest time-based debt first policy is therefore feasibility optimal. ■

Theorem 3: The largest weighted-delivery debt first policy is feasibility optimal.

Proof: Again, we translate the largest weighted-delivery debt first policy into one for the single player game. As in the previous proof, a stage in the game corresponds to a period in the system, and the action of a player is to decide the priorities of the clients at each stage. However, in this proof, the payoff corresponds to the net change of the weighted-delivery debt of each client. In other words, the payoff is an N-dimensional vector, whose n^{th} element is $(q_n - 1)/p_n$ if the packet of client n is delivered in the period, and q_n/p_n if not. A set of clients is fulfilled if each client has a negative weighted-delivery debt. Hence, we need to show that the set $A := \{z = [z_1, z_2, \cdots, z_N] | z_n \leq 0, \forall n\}$ is approachable.

Suppose at the beginning of some period, the average payoff $x = [x_1, x_2, \cdots, x_N]$ does not lie in A. We can reorder the clients so that $x_1 \geq x_2 \geq \cdots \geq x_m > 0 \geq x_{m+1} \geq \cdots \geq x_N$. The closest point in A to x is $y = [0, 0, \cdots, 0, x_{m+1}, x_{m+2}, \cdots, x_N]$. The hyperplane H through y and perpendicular to the line segment xy is comprised of points z with $h(z) := \sum_{n=1}^{m} x_n z_n = 0$.

Let π_n be the probability that client n delivers its packet in the period. The expected payoff of this period is $\bar{x} = [(q_1 - \pi_1)/p_1, (q_2 - \pi_2)/p_2, \cdots, (q_N - \pi_N)/p_N]$. Since $h(x) = \sum_{n=1}^{m} x_n^2 > 0$, we only need to show $h(\bar{x}) \leq 0$ to establish approachability of A. We have

$$\begin{aligned}
h(\bar{x}) &= \sum_{n=1}^{m} x_n \frac{q_n - \pi_n}{p_n} \\
&= \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} \frac{q_k}{p_k} - \sum_{k=1}^{n} \frac{\pi_k}{p_k})] \\
&\quad + x_m (\sum_{k=1}^{m} \frac{q_k}{p_k} - \sum_{k=1}^{m} \frac{\pi_k}{p_k}) \\
&= \tau \sum_{n=1}^{m-1} [(x_n - x_{n+1})(\sum_{k=1}^{n} w_k - \sum_{k=1}^{n} \frac{\pi_k}{\tau p_k})] \\
&\quad + \tau x_m (\sum_{k=1}^{m} w_k - \sum_{k=1}^{m} \frac{\pi_k}{\tau p_k}) \text{ (since } w_n = \tfrac{q_n}{\tau p_n}).
\end{aligned}$$

Since $x_1 \geq x_2 \geq \cdots \geq x_m > 0$, it suffices to show $\sum_{k=1}^{n} w_k \leq \sum_{k=1}^{n} \frac{\pi_k}{\tau p_k}$, for every n. Notice now that the necessary condition in Lemma 5 requires $\sum_{k=1}^{n} w_k \leq 1 - I_{\{1,2,\cdots,n\}}$. The proof is therefore complete if we can show $\sum_{k=1}^{n} \frac{\pi_k}{\tau p_k} = 1 - I_{\{1,2,\cdots,n\}}$. This is done in Lemma 6 below. ■

Lemma 6: Under the priority order $\{1, 2, \cdots, N\}$, $\sum_{k=1}^{n} \frac{\pi_k}{p_k} = \tau(1 - I_{\{1,2,\cdots,n\}})$, for $n = 1, 2, \cdots, N$.

Proof: We prove this by induction. First consider the case where $n = 1$. Since client 1 has the highest priority, it fails to deliver its packet only when every transmission during this period fails. Thus, $\pi_1 = 1 - (1 - p_1)^\tau$. Next we compute the value of $I_{\{1\}}$. The probability that client 1 delivers its packet on the γ^{th} transmission is $p_1(1-p_1)^{\gamma-1}$, which will result in $\tau - \gamma$ idle time slots. Hence, $I_{\{1\}} = \frac{1}{\tau} \sum_{\gamma=1}^{\tau} (\tau - \gamma) p_1 (1 - p_1)^{\gamma-1} = 1 - \frac{1-(1-p_1)^\tau}{p_1 \tau}$. This yields

$$\tau - \tau I_{\{1\}} = [1 - (1 - p_1)^\tau]/p_1 = \frac{\pi_1}{p_1}, \text{ for every period.} \quad (3)$$

Assume $\sum_{k=1}^{n} \frac{\pi_k}{p_k} = \tau(1 - I_{\{1,\cdots,n\}})$ for all $n \leq m$. We wish to show $\sum_{k=1}^{m+1} \frac{\pi_k}{p_k} = \tau(1 - I_{\{1,\cdots,m+1\}})$. Let $\Delta_p := \sum_{k=1}^{m+1} \frac{\pi_k}{p_k} - \sum_{k=1}^{m} \frac{\pi_k}{p_k} = \frac{\pi_{m+1}}{p_{m+1}}$ and $\Delta_I := \tau(1 - I_{\{1,\cdots,m+1\}}) - \tau(1 - I_{\{1,\cdots,m\}}) = \tau I_{\{1,\cdots,m\}} - \tau I_{\{1,\cdots,m+1\}}$. Since client $m+1$ is the one with least priority among clients 1 through $m+1$, it can transmit only after all packets from clients 1 through m are delivered. Suppose there are σ time slots left when client $m+1$ can transmit, with σ a random variable. Let $\pi_{m+1}(t)$ be the probability that the packet of client $m+1$ is delivered, and let $\zeta_{m+1}(t)$ be the number of time slots left when client $m+1$ succeeds, given that $\sigma = t$. As in the case of $n = 1$, we have $\frac{\pi_{m+1}(t)}{p_{m+1}} = t - \zeta_{m+1}(t)$. Δ_p and Δ_I can be obtained as

$$\begin{aligned}
\Delta_p &= \sum_t \frac{\pi_{m+1}(t)}{p_{m+1}} Prob(\sigma = t) \\
&= \sum_t (t - \zeta_{m+1}(t)) Prob(\sigma = t) \\
&= \sum_t t Prob(\sigma = t) - \sum_t \zeta_{m+1}(t) Prob(\sigma = t) \\
&= \tau I_{\{1,\cdots,m\}} - \tau I_{\{1,\cdots,m+1\}} = \Delta_I.
\end{aligned}$$

Finally, since $\Delta_p = \Delta_I$, $\sum_{k=1}^{m+1} \frac{\pi_k}{p_k} = \tau(1 - I_{\{1,\cdots,m+1\}})$. By induction, $\sum_{k=1}^{n} \frac{\pi_k}{p_k} = \tau(1 - I_{\{1,2,\cdots,n\}})$ holds for all n. ■

Since both policies fulfill every set of clients that is consistent with the necessary condition in Lemma 5, the necessary condition is also sufficient.

Theorem 4: A set of clients is feasible if and only if $\sum_{n \in S} w_n \leq 1 - I_S$ for every subset S of the clients, where I_S is the expected proportion of idle time for S under any work conserving policy.

VII. A Efficient Algorithm For Admission Control

Performing admission control essentially consists of deciding whether a set of clients is feasible. While Theorem 4 states a necessary and sufficient condition for feasibility, it requires testing every subset of N clients, and thus results in exponentially many tests in N. In this section, we show that we only need to evaluate a total number of N conditions to determine feasibility. The following theorem therefore makes admission control computationally efficient, and feasible to implement.

Theorem 5: Order the clients so that $q_1 \geq q_2 \geq \cdots \geq q_N$. Let S_k be the subset of clients $\{1, 2, \cdots, k\}$. The set of all clients is feasible if and only if $\sum_{n\in S_k} w_n \leq 1 - I_{S_k}$ for all k.

Proof: It is quite obvious that the condition is necessary for feasibility. Hence, we only need to prove that it is also sufficient. We prove the converse statement: If a set of clients is not feasible, then for at least one k, $\sum_{n\in S_k} w_n > 1 - I_{S_k}$.

Define a *minimal infeasible subset* as a smallest subset S with $\sum_{n\in S} w_n + I_S > 1$. Within every infeasible set there is always at least one such subset. Let m be the client with the largest index in S, i.e., the client latest in the order in S. We claim that $\sum_{n\in S_m} w_n + I_{S_m} > 1$. If S is S_m itself, then we are done. Otherwise, pick a client m' latest in the order from $S_m \backslash S$ and let $S' := S \cup \{m'\}$. Let $D_w(S, m') := \sum_{n\in S\cup\{m'\}} w_n - \sum_{n\in S} w_n$ and $D_I(S, m') := I_{S\cup\{m'\}} - I_S$. Since $\sum_{n\in S} w_n + I_S > 1$, we can show $\sum_{n\in S'} w_n + I_{S'} > 1$ by establishing $D_w(S, m') + D_I(S, m') \geq 0$.

It is obvious that $D_w(S, m') = w_{m'}$. The expression for $D_I(S, m')$ is more complicated. Note that $-\tau D_I(S, m')$ is the number of time slots that client m' will be transmitting if it is given the least priority when the set of clients is S'. Suppose it takes t time slots for every client in S to deliver its packet, then (3) shows the expected number of time slots client m' spends transmitting is $[1-(1-p_{m'})^{\tau-t}]/p_{m'}$. Hence, $-\tau D_I(S, m') = \sum_{t=1}^{\tau} f_S(t)[1 - (1 - p_{m'})^{\tau-t}]/p_{m'}$, where $f_S(t)$ is the probability that all clients in S deliver their packets in exactly t time slots . Now we have

$$\begin{aligned} D_w(S, m') + D_I(S, m') &= w_{m'} - \frac{1}{\tau}\sum_{t=1}^{\tau} f_S(t)\frac{1 - (1 - p_{m'})^{\tau-t}}{p_{m'}} \\ &> \frac{q_{m'}}{\tau p_{m'}} - \frac{1}{\tau}\sum_{t=1}^{\tau} f_S(t)\frac{1}{p_{m'}} \\ &= \frac{1}{\tau p_{m'}}(q_{m'} - \sum_{t=1}^{\tau} f_S(t)). \end{aligned}$$

What remains is to determine the value of $\sum_{t=1}^{\tau} f_S(t)$. Since S is a minimal infeasible subset, the set $S \setminus \{m\}$ is feasible. Let η be any policy that fulfills $S \setminus \{m\}$. Consider the following policy η' for S: Whenever there is an undelivered packet for clients among $S\backslash\{m\}$, schedule clients as η does. Client m is scheduled to transmit only after packets for all other clients are delivered. Now, every client in $S \setminus \{m\}$ is fulfilled under this policy. However, since S is not feasible, at least one client in S is not fulfilled under any policy. Hence, client m is not fulfilled under policy η', and so we have $Prob$(client m succeeds) $< q_m$. Further, since client m is given the least priority under policy η', we have $Prob$(client m succeeds) $= Prob$(all packets in S are delivered) $= \sum_t f_S(t)$. This implies $\sum_t f_S(t) < q_m$. Inserting this in the inequality in the previous paragraph yields

$$\begin{aligned} D_w(S, m') + D_I(S, m') &> \frac{1}{\tau p_{m'}}(q_{m'} - \sum_t f_S(t)) \\ &> \frac{1}{\tau p_{m'}}(q_{m'} - q_m) \geq 0, \end{aligned}$$

since client m occurs later in the order than client m', and we have sorted clients so that $q_1 \geq q_2 \geq \cdots \geq q_N$.

Now we have established $\sum_{n\in S'} w_n + I_{S'} > 1$, where $S' = S\cup\{m'\}$. If $S' = S_m$, we are done. Otherwise, we can choose another client m'' with the highest order in $S_m \backslash S'$ and repeat the above argument to show $\sum_{n\in S'\cup\{m''\}} w_n + I_{S'\cup\{m''\}} > 1$. By induction, $\sum_{n\in S_m} w_n + I_{S_m} > 1$. ■

In addition to reducing the number of needed tests to N, this theorem also helps improve the efficiency of evaluating each test. In each test, we need to obtain the values of $\sum_{n\in S_m} w_n$ and I_{S_m}, both of which require more than a constant computation time. However, by using the fact that $S_m = S_{m-1} \cup \{m\}$, we can incrementally obtain these values and improve complexity.

Obtaining $\sum_{n\in S_m} w_n$ is easy since it equals $\sum_{n\in S_{m-1}} w_n + w_m$, and requires only one addition operation given the value of $\sum_{n\in S_{m-1}} w_n$. The computation of I_{S_m} is more complicated. Let $g_{S_m}(t)$ be the probability that all packets in S_m are delivered at or before time t, resulting in at least $\tau - t$ idle time slots. I_{S_m}, being the expected proportion of idle time slots, is $\frac{1}{\tau}\sum_{i\geq 1} Prob$(number of idle time slots$\geq i$) $= \frac{1}{\tau}\sum_{i=1}^{\tau-1} g_{S_m}(t)$. Consider the value of $g_{S_m}(t)$:

$$\begin{aligned} g_{S_m}(t) =& Prob(\text{all packets in } S_m \text{ are delivered by } t) \\ =& \sum_{i=1}^{\tau} Prob(\text{all packets in } S_{m-1} \text{ are delivered by} \\ & \text{time } t-i, \text{ and clients } m \text{ takes } i \text{ time slots} \\ & \text{to succeed}) \\ =& \sum_{i=1}^{\tau} g_{S_{m-1}}(t-i)[p_m(1-p_m)^{i-1}]. \end{aligned}$$

Thus, the vector of $[g_{S_m}(i)]$ is indeed the convolution of the vectors of $[g_{S_{m-1}}(i)]$ and $[p_m(1-p_m)^{i-1}]$, which can be computed in $O(\tau^2)$ time by brute force, or $O(\tau \log \tau)$ by using the Fast Fourier Transform algorithm.

A complete algorithm for deciding whether a set of clients is feasible is given in Algorithm 1. The complexity of the algorithm is $O(N\tau^2)$ or $O(N\tau \log \tau)$, depending on the implementation of convolution.

It should be noted that Algorithm 1 assumes the clients are sorted already. While sorting can require $\Theta(N \log N)$ time in general, this operation can also be made more efficient. In practice, clients often come into the system one by one rather than in a group. Admission control can hence be simplified as one of deciding whether the system can accommodate a new client. In this case, the server can keep a sorted list of the existing clients. When a new client requests admission, the server inserts the new client into the proper position in the list, and runs Algorithm

1. Since inserting a new client into the list takes $O(N)$ time, it remains $O(N\tau^2)$ or $O(N\tau \log \tau)$ time to determine whether to admit a new client.

Algorithm 1 IsFeasible

Assume clients are sorted so that $q_1 \geq q_2 \geq \cdots \geq q_N$
$totalW \leftarrow 0$
$[g_{S_0}(i)] \leftarrow [0, 0, \cdots, 0]$
for $m = 1$ to N **do**
  $totalW \leftarrow totalW + \frac{q_m}{\tau p_m}$
  $[g_{S_m}(i)] \leftarrow [g_{S_{m-1}}(i)] * [p_m(1-p_m)^{i-1}]$
  $totalI \leftarrow \frac{1}{\tau} \sum_{i=1}^{\tau-1} g_{S_m}(i)$
  if $totalW + totalI > 1$ **then**
    return Infeasible
return Feasible

VIII. Implementation Issues

In this section, we describe how to implement the largest debt first policies under the IEEE 802.11 mechanisms. While our earlier problem formulation appears to assume there is only traffic from clients to the server, we explain how to modify the policies to deal with duplex traffic.

A. Implementation on IEEE 802.11

The IEEE 802.11 standard provides two different channel access mechanisms, the mandatory Distributed Coordination Function (DCF), and the optional Point Coordination Function (PCF) [6]. The PCF is a centralized access mechanism proposed to support time-bounded services. We will implement the largest debt first policies based on the PCF mechanism.

In PCF, the transmissions of prioritized traffic are coordinated by a station called the Point Coordinator (PC), which is the server in our scheme. The PC grants channel access to clients by sending a CF-POLL packet. The client indicated by the CF-POLL packet then replies with a Data packet if it has data to send, or a Null packet if it doesn't. To ensure that the PCF has a higher priority over the DCF, the PC and the client indicated by the CF-POLL transmit packets after detecting the channel as being idle for a period of PCF Interframe Space (PIFS) and Short Interframe Space (SIFS), respectively. The DCF, on the other hand, starts the backoff process only after the channel idles for a period of DCF Interframe Space (DIFS). The values of PIFS and SIFS are 30 μs and 10 μs, respectively, while the value of DIFS is 40 μs in IEEE 802.11 b/g. Since PIFS and SIFS are smaller than DIFS, PCF is guaranteed higher priority for accessing the channel.

In our implementation, the server works as the PC in PCF. It indicates which client should transmit according to the chosen policy. Further, to eliminate the time needed for the server to send an ACK, all clients send their packets by broadcasting. Finally, since PCF can coexist with DCF, the system can still accommodate traffic with no QoS constraints.

B. Dealing with Duplex Traffic

In Section III, we have assumed that there is only uplink traffic, that is, traffic from clients to the server, and no downlink traffic, which goes from the server to clients. In many applications, such as VoIP, this assumption doesn't hold. However, it is easy to modify our solution to deal with duplex traffic.

When a client requests downlink traffic with QoS constraints, the server creates a pseudoclient. The scheduling policies compute the debt owed to the pseudoclient and its priority in each period, just as though it were a normal client. When the pseudoclient is scheduled to transmit, the server sends out a packet to the requesting client. The packet is considered delivered if the server receives an ACK from the client. In this case, the length of the slot time is determined so that it is long enough for both uplink traffic, which includes the CF-POLL packet and the Data packet, and downlink traffic, which includes the Data packet and the ACK. While the different times required by uplink traffic and downlink traffic may cause additional idle time in each time slot, this cost is insignificant. The only difference between uplink traffic and downlink traffic is that uplink traffic needs to transmit the CF-POLL packet while the downlink traffic requires an ACK packet. The CF-POLL packet only needs to specify a receiver address to indicate which client is allowed to transmit. Thus, the format of CF-POLL is essentially the same as the format of ACK, and it takes approximately equal time to transmit both packets. Hence, the time required by uplink traffic is almost the same as that required by downlink traffic.

IX. Simulation Results

A. Simulation Setup

We follow the G.711 codec, which is a ITU-T standard for audio compression, in deciding parameters for traffic with QoS constraints. G.711 generates data at 64 kbps. With a 20 ms packetization interval, this results in a 160 Bytes VoIP packet. We use IEEE 802.11b as the underlying MAC protocol, whose transmission rate can be as high as 11 Mb/s. Some details of parameters are given in Table I. Under this setting, the total transmission time for a CF-POLL packet and a Data packet is slightly smaller than 610 μs, allowing 32 time slots in a 20 ms period. All the results in the following sections are the average over 100 runs.

We have implemented the two largest debt first policies, the largest time-based debt first policy and the largest weighted-delivery debt first policy, on ns-2, and compared them against the naive approach using IEEE 802.11 DCF. To evaluate the performance of different mechanisms, we define a *deadline miss ratio* (*DMR*) function. Let d_n be the delivery ratio achieved by the client n at some given time. The *deadline miss ratio of client* n is defined as

$$DMR_n = \begin{cases} q_n - d_n, & \text{if } q_n > d_n, \\ 0, & \text{otherwise,} \end{cases}$$

and the *deadline miss ratio of the system* is defined as the sum of the deadline miss ratios of all clients.

One may argue that the comparison between largest debt policies and IEEE 802.11 DCF is not fair. By using IEEE 802.11 PCF, our policies avoid additional overheads, including the time spent on backing off and the risk of packet collision, that are inevitable in DCF. To make the comparison fair, we therefore also implement a *random policy*, which is also based on PCF, that assigns priorities randomly at the beginning of each period.

TABLE I: Simulation Setup

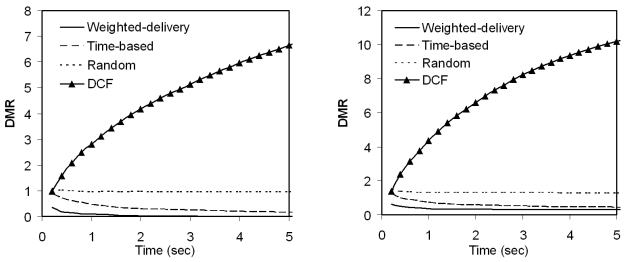
Packetization interval	20 ms
Payload size per packet	160 Bytes
Transmission data rate	11 Mb/s
SIFS	10 μs
PIFS	30 μs
DIFS	40 μs

B. Simplex Traffic

We first consider the case where there is only uplink traffic. We consider two groups of clients, group A and group B. Clients in group A carry more important messages and require a 99% delivery ratio, while clients in group B require a 80% delivery ratio. The channel reliability of the n^{th} client in both groups is assumed to be $(60+n)\%$. Using Algorithm 1, it can be shown that a set of 11 group A clients and 12 group B clients is feasible, but a set of 12 group A clients and 12 group B clients is not.

We first run simulations of the four different policies, namely, the two largest debt first policies, the random policy based on PCF, and the DCF mechanism, on a set of 12 group A clients and 11 group B clients. Figure 1a presents the deadline miss ratios of different policies. The deadline miss ratios of the two largest debt first policies converge to zero over time, showing that both policies fulfill the set of clients. The largest weighted-delivery debt first policy has a better performance over the largest time-based debt first policy since it converges faster. This is because the largest weighted-delivery debt first policy uses the feedback information of MAC to count the actual number of packets delivered for each client. This gives a better estimate on whether a client requires more transmission opportunities. The largest time-based debt first policy, on the other hand, uses the more indirect approach by counting the number of times a client transmits, which gives a slower convergence rate. While the largest weighted-delivery debt first policy may require a more complicated implementation, it is suitable for applications where the short-term performance is also important.

Both the other two policies do not satisfy all clients. The deadline miss ratio of the random policy remains approximately 1. The random policy cannot be feasibility optimal since it gives each client equal priorities in the long term, regardless of the required delivery ratio of each client. However, clients with more important data should be granted more transmission opportunities than others. Failing to take this factor into account makes the random policy not feasibility optimal. Meanwhile, the DCF mechanism has a much higher deadline miss ratio. This is due to the lack of awareness of delay constraints in DCF. When a client puts a packet in the transmission queue, the packet cannot be removed until it is transmitted or dropped by the MAC. Thus, when the packet generation rate exceeds the packet outgoing rate, the queuing delay gets even larger, resulting in a large deadline miss ratio.



(a) Performance of a feasible set (b) Performance of an infeasible set

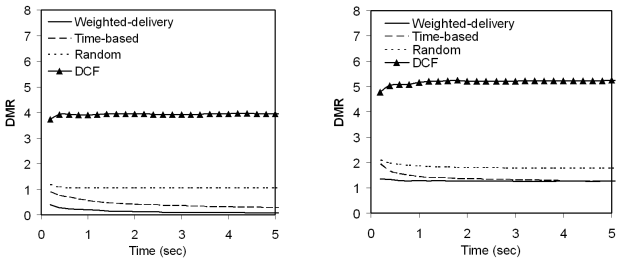
Fig. 1: Deadline miss ratios for simplex traffic

To evaluate the accuracy of Algorithm 1, we run simulations on a set of 12 group A clients and 12 group B clients, which is reported infeasible by Algorithm 1. Figure 1b shows the results. It can be seen that all the four policies offer non-zero deadline miss ratios, confirming the infeasibility of this set. It can also be seen that, like the case of a feasible set, the two largest debt first policies result in the least deadline miss ratios. This result suggests that our proposed policies still work well even when link qualities are not high enough to satisfy the clients. Since the link quality does vary with time in wireless networks, it is likely that the network may suffer from temporary link quality downgrades from time to time. The ability to provide good service under these downgrades is essential to the robustness of a system. In this case, our policies appear to be robust.

C. Duplex Traffic

We next consider the case where clients generate duplex traffic. As in the previous section, we assume there are two groups, A and B, of clients, who require 99% and 80% delivery ratio for both uplink and downlink traffic, respectively. The channel reliability of the n^{th} client in both groups is $(60+n)\%$. Using Algorithm 1, we find that a set of 6 group A clients and 5 group B clients is feasible, while a set of 6 group A clients and 6 group B clients is not.

We run simulations on the feasible set of 6 group A clients and 5 group B clients. Figure 2a presents the deadline miss ratios of the four policies. The results are similar to those in the previous section, with the two largest debt first policies having near-zero deadline miss ratios, while the other two policies result in non-optimal



(a) Performance of a feasible set (b) Performance of an infeasible set

Fig. 2: Deadline miss ratios for duplex traffic

performance. These results suggest that our proposed policies remain feasibility optimal even when they are applied to duplex traffic.

A somewhat surprising result is that the deadline miss ratio of DCF, unlike that in the last section, levels off quickly. This is caused by the different delays experienced by uplink traffic and downlink traffic. When clients request duplex traffic, each client only generates one packet per period. The server, however, needs to generate one packet for each client per period. Thus, the server is much more heavily-loaded than the clients. However, the DCF gives each wireless node equal transmission opportunities, causing each client to get a more than enough share of transmission time, with the server being given an inadequately small transmission opportunity. A closer look at the simulation data shows that the downlink traffic contributes 3.82 to the deadline miss ratio while the deadline miss ratio of uplink traffic is only 0.15, confirming our reasoning.

As in the previous section, we also evaluate the performance of the four tested policies on an infeasible set of 6 group A clients and 6 group B clients. As shown in Figure 2b, all policies fail to satisfy this set of clients, suggesting that Algorithm 1 is also accurate for duplex traffic. Further, the two proposed policies result in the least deadline miss ratios, showing that they are robust against temporary downgrades of link qualities.

A final remark is that the random policy is always better than DCF in all simulation settings. This shows that the additional overheads of DCF, including the time spent on backing off and the imbalance of uplink and downlink delay, can hurt the performance of a system greatly. Since PCF can coexist with DCF, it seems preferable to use PCF for traffic with QoS constraints while other traffic can still be served by DCF.

X. Concluding Remarks

We have studied the problem of providing QoS on unreliable wireless networks. The QoS constraints are expressed in terms of a delay bound, which is the same as the period of packet arrivals, and a user-defined delivery ratio lower bound. We have analytically proved a necessary and sufficient condition for a set of clients to be feasible. Based on this condition, we have proposed a linear time algorithm for admission control. We have also designed two scheduling policies for traffic with QoS constraints. We have proved that both policies are feasibility optimal. We have shown that these policies are easily implementable on the widely-used IEEE 802.11 mechanisms. Finally, simulation results for various scenarios have been provided. Our theoretical studies are confirmed since the two proposed policies are indeed feasible for all feasible sets of clients tested. Even when dealing with infeasible sets of clients, our proposed policies still outperformed others.

The model introduced in this paper is restrictive in some aspects. For example, it is assumed that all clients generate packets at the same rate and the delay requirements are the same for all packets. How to extend the model for more complicated and realistic scenarios is an interesting topic for future research.

References

[1] M. Agarwal and A. Puri. Base station scheduling of requests with fixed deadlines. In *Proc. of IEEE INFOCOM 2002*.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting. CDMA data QoS scheduling on the forward link with variable channel conditions. *Bell Laboratories Technical Report*, April 2000.

[3] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific J. Math*, 6(1), 1956.

[4] A. Dua and N. Bambos. Deadline constrained packet scheduling for wireless networks. In *62nd IEEE VTC Fall 2005*.

[5] S. Garg and M. Kappes. Admission control for VoIP traffic in IEEE 802.11 networks. In *Proc. of GLOBECOM 2003*.

[6] IEEE. Wireless LAN medium access control (MAC) and physical (PHY) specifications. 1999.

[7] K. B. Johnsson and D. C. Cox. An adaptive cross-layer scheduler for improved QoS support of multiclass data services on wireless systems. *IEEE J. on Selected Areas in Communications*, 23(2), 2005.

[8] T. Kawata, S. Shin, A. G. Forte, and H. Schulzrinne. Using dynamic PCF to improve the capacity for VoIP traffic in IEEE 802.11 networks. In *Proc. of IEEE WCNC 2005*.

[9] M. Loeve. *Probability Theory II*. Springer-Verlag, 1978.

[10] M. J. Neely. Delay analysis for max weight opportunistic scheduling in wireless systems. In *Proc. of Allerton Conf.*, 2008.

[11] D. Pong and T. Moors. Call admission control for IEEE 802.11 contention access mechanism. In *Proc. of GLOBECOM 2003*.

[12] V. Raghunathan, V. Borkar, M. Cao, and P.R. Kumar. Index policies for real-time multicast scheduling for wireless broadcast systems. In *Proc. of IEEE INFOCOM 2008*.

[13] S. Shakkottai and R. Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *Wireless Networks*, 8(1), Jan. 2002.

[14] S. Shakkottai and A. L. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in HDR. In *Proc. of 17th International Teletraffic Congress (ITC-17)*, 2001.

[15] S. Shin and H. Schulzrinne. Call admission control in IEEE 802.11 WLANs using QP-CAT. In *Proc. of INFOCOM 2008*.

[16] A. L. Stolyar and K. Ramanan. Largest weighted delay first scheduling: Large deviations and optimality. *Ann. Appl. Probab.*, 11(1), 2001.

[17] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. on Information Theory*, 39, March 1993.

[18] Y. Xiao and H. Li. Evaluation of distributed admission control for the IEEE 802.11e EDCA. *Communications Magazine, IEEE*, 42(9), Sept. 2004.

[19] H. Zhai, X. Chen, and Y. Fang. A call admission and rate control scheme for multimedia support over IEEE 802.11 wireless LANs. *Wireless Networks*, 12(4), August 2006.