Scheduling Heterogeneous Real-Time Traffic over Fading Wireless Channels

I-Hong Hou

***Abstract*—We develop a general approach for designing scheduling policies for real-time traffic over wireless channels. We extend prior work, which characterizes a real-time flow by its traffic pattern, delay bound, timely-throughput requirement, and channel reliability, to allow clients to have different deadlines, and allow a variety of channel models. In particular, our extended model consider scenarios where channel qualities are time-varying, the access point may not have explicit information on channel qualities, and the access point may or may not employ rate adaptation. Thus, our model allow the treatment of more realistic fading channels as well as scenarios with mobile nodes, and the usage of more general transmission strategies.**

We derive a sufficient condition for a scheduling policy to be feasibility optimal, and thereby establish a class of feasibility optimal policies. We demonstrate the utility of the identified class by deriving a feasibility optimal policy for the scenario with rate adaptation, time-varying channels, and heterogeneous delay bounds. When rate adaptation is not available, we also derive a feasibility optimal policies for both scenarios where the access point may or may not have explicit knowledge on channel qualities. For the scenario where rate adaptation is not available but clients have different delay bounds, we describe a heuristic. Simulation results are also presented which indicate the usefulness of the scheduling policies for more realistic and complex scenarios.

***Index Terms*—Wireless networks; real-time communications; delay constraints; fading channels; rate adaptation.**

I. Introduction

With the wide deployment of Wireless Local Area Networks (WLANs) and advances in multimedia technology, wireless networks are increasingly being used to carry real-time traffic, such as VoIP and video streaming. These applications usually specify throughput requirement while meeting specified delay bounds. We study the problem of designing scheduling policies for such applications.

While there has been much research on scheduling real-time traffic over wireline networks, the results are not directly applicable to wireless networks where channels are unreliable, with qualities that may be time-varying either due to fading or node mobility. Also, individual clients may impose differing delay requirements. These features present new challenges to the scheduling problems.

We consider the scenario where an Access Point (AP) is required to serve real-time traffic for a set of clients. A previous work [1] solves the scheduling problem in a restrictive environment and proposes two feasibility optimal policies. In particular, it assumes a fixed transmission rate, a static channel model, and that all clients in the system require the same delay bound. We extend this model to relax these limitations. The extended model considers a variety of scenarios where wireless channel qualities are time-varying. These scenarios include those where the AP may and may not have explicit knowledge on channel qualities, and those where the AP may and may not employ rate adaptation. We establish a sufficient condition for a scheduling policy to be feasibility optimal. Based on this we describe a class of policies and prove that they are all feasibility optimal.

To demonstrate the utility of the class of policies, we study four particular scenarios of interest. Three of these scenarios consider systems without rate adaptation. They address the challenges of time-varying channels, lack of explicit knowledge on channel qualities, and heterogeneous delay bound of clients, respectively. The last scenario employs rate adaptation and treats time-varying channels, as well as allowing different delay bounds for different clients. We derive on-line scheduling policies for each of them.

We have also tested the derived policies using the IEEE 802.11 standard in a simulation environment. The results suggest that the four policies outperform others, including the policies in [1], and a server-centric policy that schedules packets randomly. In particular, since the policies introduced in the previous work fail to provide satisfactory performance in the environments studied here, this suggests that neglecting the facts that the system can apply rate adaptation, that wireless channels are time-varying, and the possibility that clients may require different delay bounds, can result in malperformance of the derived policies.

Section II reviews some of the related work. Section III describes the extension of the model in [1]. In Section IV, we derive a general class of policies that are feasibility optimal. Based on this class, we obtain scheduling policies in Sections V, VI, and VIII, and a heuristic in Section VII, for different scenarios. In Section IX, we discuss implementation issues and simulation results. Section X concludes the paper.

II. Related Work

The problem of providing QoS over unreliable wireless channels has received growing interest in recent years. Tassiulas and Ephremides [2] have considered the problem in a single-hop network by assuming ON/OFF channels and derived a throughput-optimal policy. Though the policy is unaware of packet delay, Neely [3] has shown that average packet delay is constant regardless of the

I-Hong Hou is with Department of ECE, Texas A&M University, College Station, Texas 77843-3128, USA. email: ihou@tamu.edu

This material is based upon work partially supported by USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, AFOSR under Contract FA9550-09-0121, and NSF under Contract Nos. CNS-07-21992, ECCS-0701604, CNS-0626584, and CNS-05-19535.

Part of this work has been presented at the 2010 IEEE INFOCOM.

network size. Andrews et al [4] have proposed another policy that aims to improve packet delay. Liu, Wang, and Giannakis [5] have used a cross-layer approach to provide differentiated service for a variety of classes of clients. Grilo, Macedo, and Nunes [6] have proposed a resource-allocation algorithm based on the expected transmission time of each packet. These studies have focused on improving the average delay of all packets. In contrast, we aim to provide a strict delay guarantee for each packet in this paper. Raghunathan et al [7] and Shakkottai and Srikant [8] have both approached this problem by analytically demonstrating algorithms to minimize the total number of expired packets in the system. Soldati et al [12] have studied meeting the deadline of a packet by jointly considering routing and scheduling when there is only one packet in the system. Li and Eryilmaz [13] have proposed scheduling policies for real-time flows in multihop networks. Hou, Borkar, and Kumar [9] have studied the problem of providing QoS based on delay bounds and delivery ratio requirements, and proposed two optimal policies under some restrictive assumptions. Their work has been further extended to deal with variable-bit-rate traffic [1]. Jaramillo and Srikant [10] have considered serving systems where real-time flows and non-real-time flows coexist under two types of channel models. Jaramillo, Srikant, and Ying [11] have proposed a framework for developing scheduling policies in ad hoc networks with real-time flows, and provided a tractable online scheduling policies for a special case. In this paper, we extend the previous work to more realistic scenarios, including rate adaptation, time-varying channels and heterogeneous delay bounds among clients. We also show that there exist simple and optimal online scheduling policies in many different cases. Fattah and Leung [14] and Cao and Li [15] have surveyed other existing scheduling policies for providing QoS.

III. System Model

We begin by extending the model proposed in [1], which only considers a static channel condition and fixed delay bounds for all clients, to account for network behavior and application requirements for serving real-time flows in wireless systems.

Consider a wireless system with N clients, $\{1, 2, \ldots, N\}$, and one access point (AP). Time is slotted with slots $t \in \{1, 2, 3, \ldots\}$. Time slots are further grouped into *intervals* where each interval consists of T consecutive time slots in $(kT, (k+1)T]$. Each client is associated with a real-time flow. At the beginning of each interval, i.e., at time slots $\{1, T+1, 2T+1, \ldots\}$, each flow may generate a finite number of packets probabilistically. We model the packet arrivals as a stationary, irreducible Markov process with finite states. Packet arrivals can be dependent between clients, and packet arrivals in an interval can depend on other intervals.

We assume that the AP schedules all transmissions for all clients, and the AP has instant feedback information on whether a transmission is successful or not. This applies to scenarios where all clients generate downlink traffic, as the AP can choose which client's packet to be transmitted and request the chosen client to reply an ACK if the client receives the packet successful. In this case, a transmission is considered successful if the AP receives the ACK after the transmission. In systems where clients may generate uplink traffic, this can still apply to a wide range of server-centric wireless communication technologies, such as IEEE 802.11 Point Coordination Function (PCF), WiMax, and Bluetooth. For example, when the AP schedules a client with uplink traffic in PCF, it first sends a POLL packet to the client. The chosen client then replies with a data packet upon receiving the POLL packet. In this case, the AP does not need to reply an ACK to the client, and the transmission is considered successful if the AP receives the data packet from the client.

Each client n specifies a strict per-packet delay bound τ_n time slots, with $\tau_n \leq T$. If the packet for client n is not delivered by the ${\tau_n}^{th}$ time slot of the interval, the packet expires and is discarded. By discarding expired packets, we guarantee that the delays of all successfully delivered packets for client n is at most τ_n time slots. The strict per-packet delay bound is similar to the delay guarantees in real-time system literatures [16].

In this paper, we develop a framework that can be applied to a wide range of channel models. We start with a general and abstract description of the concept of channel model, and then provide detailed definitions of several channel models used in this paper. A channel model involves the description of the behaviors of wireless transmissions. Under a channel model, we use *channel state* to denote all the parameters related to the channel model. The parameters may, for example, include signal-to-noise ratio (SNR), link reliability, transmission rate, etc, depending on the channel model. We assume that the channel state can change over time, and model it as a stationary, irreducible Markov process with a finite set of states $\mathcal{C}$. Under the stationary distribution, the probability that channel state c occurs is f_c and the channel state remains constant within each interval. The channel state and the packet arrivals in an interval are assumed to be independent of each other, and are assumed to be independent from the scheduling decisions of the AP. We assume that the AP knows the channel state c in each interval. VoIP traffic typically has delay requirements about tens of msec, and, as shown in Bhagwat et al [17], the average time that a channel stays in one state ranges from 50 msec to 10 sec, which can be much larger than the delay requirements of VoIP. Further, obtaining channel state information for each client may incur significant overhead, making it infeasible for the AP to update channel state information too frequently. Thus, in many practical settings, it is reasonable to assume that the channel state does not change within an interval. The reason that we model both channel state and packet arrivals by irreducible Markov processes with finite states is to ensure that both of them have some steady-state distributions. In Section IV, we will show that there exists a framework where the AP makes scheduling decisions only based on the channel state and packet arrivals in the current interval, and the AP does not need to know the underlying model for channel state and packet arrivals.

We consider three channel models in this paper: one where the AP uses a fixed transmission rate and has instant knowledge on channel states, one where the AP also uses a fixed transmission rate but does not have full instant knowledge on channel states, and one where rate

adaptation is employed.

For the first channel model, the AP uses a fixed transmission rate for all clients. We take the length of a time slot to be the time needed for one transmission, including all overheads, and hence the AP can make exactly one transmission in each time slot. We consider the unreliable nature of wireless transmissions. There are many studies, such as [18], that analyze the relationship between SNR and link reliability. Therefore, by defining the channel state c to be the value of SNR for each link, the AP can determine the probability that a transmission for client n is successful, which we denote by $p_{c,n}$.

The second channel model is similar to the first one. However, we assume that the AP does not know explicitly the value of SNR for each link. Instead, it only has some initial information about the distribution of SNR, and hence the distribution of link reliability. We define the channel state c as the AP's initial information. Under channel state c, the link reliability of client n can be among a finite number of values, and the AP only knows the probability distribution of the link reliability of client n. We assume that the link reliability of client n is independent of that of any other client. Further, the AP can calculate the posterior probability distribution of link reliabilities any time during an interval, based on events in the interval.

The third channel model considers the usage of rate adaptation. When rate adaptation is employed, the AP chooses different transmissions rates for different clients based on their respective channel qualities so as to ensure all transmissions are error free. The transmission rate for a client effectively decides the amount of time needed to deliver a packet for the client. Under channel state c, it takes $s_{c,n}$ time slots to make an error free transmission to client n.

Each client n requires a timely-throughput of at least q_n packets per interval.This timely-throughput bound can also be interpreted as a delivery ratio requirement of $q_n/\{$average number of packets of n per interval$\}$.

Definition 1: Let H_t be the set of all possible histories of the system up to time slot t. H_t includes information on all past packet arrivals, channel states, packet transmissions, packet deliveries, etc. Let $h_t \in H_t$ be the actual history that the system has experienced. Based on h_t, a *scheduling policy* chooses, possibly at random, a packet to transmit at time t or idles, subject to limitations of the channel model.

Definition 2: A set of clients, $\{1, 2, \ldots, N\}$ is *fulfilled* under a scheduling policy η, if for every $\epsilon > 0$,

$$Prob\{\frac{d_n(t)}{t/T} > q_n - \epsilon, \text{ for every } n\} \to 1, \text{ as } t \to \infty,$$

where $d_n(t)$ is the number of packets delivered to client n up to time t.

In this paper, we propose a framework for designing *feasibility optimal* scheduling policies that fulfills every *strictly feasible* systems. In the following, we use $[q_n]$ to denote the vector containing timely-throughput requirements of clients $\{q_1, q_2, \ldots, q_N\}$.

Definition 3: A system with timely-throughput requirements $[q_n]$ is *feasible* if there exists some scheduling policy that fulfills it. The *feasible region* of a system is then defined to be the set of all feasible $[q_n]$.

Definition 4: A system with timely-throughput requirements $[q_n]$ is *strictly feasible* if $q_n > 0$ for all n and there exists some $0 < \alpha < 1$ such that the same system with timely-throughput requirements $[q_n/\alpha]$ is also feasible. [1]

Definition 5: A scheduling policy is *feasibility optimal* if it fulfills all strictly feasible systems.

IV. A Framework for Feasibility Optimal Policies

A. Preliminaries

In this section, we propose a framework for designing feasibility optimal policies. We first introduce several preliminary results that will be used to establish the framework.

The overall system can be viewed as a controlled Markov chain. In the controlled Markov chain, we represent the state of the system in an interval by its channel state and packet arrivals. The control action of the Markov chain is the scheduling decisions of the AP. Based on the scheduling decisions and the state of the system, the AP obtains a random vector of payoffs which consists of the number of packet deliveries for each client. We have:

Lemma 1: [19] For any set of clients that can be fulfilled, there exists a *stationary randomized policy* that fulfills the clients, which uses a probability distribution based only on the channel state, the set of undelivered packets, and the number of time slots remaining in the system (and not any events depending on past intervals), according to which it randomly chooses an undelivered packet to transmit, or stays idle.

We then show that the feasible region is convex.

Lemma 2: The feasible region of any system is convex.

Proof: Let $[q_n]$ and $[q'_n]$ be two vectors in the feasible region of some system. Let η and η' be stationary randomized policies that fulfill the two vectors, respectively. Then, the policy that randomly picks one of the two policies, with η being chosen with probability α, at the beginning of each interval, fulfills the vector $[\alpha q_n + (1-\alpha)q'_n]$. Thus, the vector $[\alpha q_n + (1-\alpha)q'_n]$ also falls in the feasible region of the system. Therefore, the feasible region of the system is convex. ■

B. Scheduling Policies for Static Unreliable Channels

In previous work [1], the problem of feasibility optimal scheduling has been addressed for a special case of the first channel model where the channel state is static, and all clients require the same delay bounds, i.e. $|\mathcal{C}| = 1$ and $\tau_n \equiv \tau$. In the special case, we will use p_n instead of $p_{c,n}$ since the channel state is static, and τ instead of τ_n. The previous work also assumes that each client may generate at most one packet in each interval.

Two *largest debt first* scheduling polices were proved to be feasibility optimal, where the AP, based on the past history, calculates a debt for each client. In each interval, the AP sorts all clients according to their debts, and schedules a packet for client n only after all packets for clients with larger debts have been delivered. The first policy, the *largest time-based debt first policy*, uses the *time-based debt* for client n at time slot t, defined as $\frac{t}{T}(q_n/p_n)$

[1] Equivalently, $[q_n]$ is strictly feasible if it is an interior point of the feasible region.

minus the number of time slots that the AP has spent on transmitting packets for client n up to time slot t. The other policy, the *largest weighted-delivery debt first policy*, uses the *weighted-delivery debt* for client n at time slot t, defined as $\frac{\frac{t}{T}q_n - d_n(t)}{p_n}$, where $d_n(t)$ is the number of delivered packets for client n up to time slot t.

C. A Sufficient Condition for Feasibility Optimality

We now describe a more general class of policies that is feasibility optimal. We start by extending the concept of "debt".

Definition 6: A variable $r_n(k)$, whose value is determined by the past history of the client n up to the k^{th} interval, or time slot kT, is called a *pseudo-debt* if:

1) $r_n(0) = 0$, for all n.
2) At the beginning of each interval, $r_n(k)$ increases by a constant strictly positive number $z_n(q_n)$, which is an increasing linear function of q_n.
3) $r_n(k+1) = r_n(k) + z_n(q_n) - \mu_n(k)$, where $\mu_n(k)$ is a non-negative and bounded random variable whose value is determined by the behavior of client n. Further, $\mu_n(k) = 0$ if the AP does not transmit any packet for client n.
4) The set of clients is fulfilled if and only if $Prob\{\frac{r_n(k)}{k} < \varepsilon\} \to 1$, as $k \to \infty$, for all n and all $\varepsilon > 0$.

In the following example, we illustrate that both the time-based debt and the weighted-delivery debt are pseudo-debts under a static channel model.

Example 1: At the beginning of each interval, the time-based debt $r_n^{(1)}(k)$ increases by $w_n = \frac{q_n}{p_n}$, and decreases by the number of time slots that the AP has transmitted packets for client n during the interval. [1] has shown that condition (4) is also satisfied.

Similarly, $r_n^{(2)}(k)$, the weighted-delivery debt is also a special case, where $z_n(q_n) = \frac{q_n}{p_n}$, and $\mu_n(k)$ is {number of packets delivered for n in interval k}$/p_n$. It satisfies condition (4) by definition.

We can also consider the *delivery debt* $r_n^{(3)}(k) := q_n k - d_n(kT)$, where $d_n(t)$ is the number of delivered packets for client n up to time slot t. For delivery debt, $z_n(q_n) := q_n$, and $\mu_n(k)$ is the number of packets delivered for n in interval k. Thus, delivery debt is also a pseudo-debt. □

We can also define the *feasible region for debt* as the set of $[z_n(q_n)]$ such that the corresponding $[q_n]$ is feasible. Since $z_n(q_n)$ is a linear function of q_n and the feasible region for $[q_n]$ is a convex set (Lemma 2), the feasible region for $[z_n(q_n)]$ is also a convex set.

Using the concept of pseudo-debt, we prove a sufficient condition for feasibility optimality. The proof resembles one used by Neely [3], though in a different context, and is based on:

Theorem 1 (Lyapunov Drift Theorem [3]): Let $L(t)$ be a non-negative Lyapunov function. Suppose there exists some constant $B > 0$ and non-negative function $f(t)$ adapted to the past history of the system such that:

$$E\{L(t+1) - L(t)|H_t\} \le B - \epsilon f(t),$$

for all t, where H_t is the system history up to time t as discussed in Definition 1. Then: $\limsup_{t\to\infty} \frac{1}{t}\sum_{i=0}^{t} E\{f(i)\} \le B/\epsilon$. □

Theorem 2: Let $r_n(k)$ be a pseudo-debt.A policy that maximizes the *payoff function*

$$\sum_{n=1}^{N} E\{r_n(k)^+ \mu_n(k)|c_k, S_k, [r_m(k)]\} \tag{1}$$

in each interval is feasibility optimal, where c_k denotes the channel state in the k^{th} interval, S_k is the set of packets that arrive at the AP at the beginning of the k^{th} interval, and $x^+ := \max\{x, 0\}$.

Proof: Define $L(k) = \frac{1}{2}\sum_{n=1}^{N} r_n(k)^2$. Since $r_n(k+1) = r_n(k) + z_n(q_n) - \mu_n(k)$,

$$\begin{aligned}
&\Delta(L(k)) := E\{L(k+1) - L(k)|c_k, S_k, [r_m(k)]\} \\
=&E\{\frac{1}{2}\sum_{n=1}^{N} r_n(k+1)^2 - \frac{1}{2}\sum_{n=1}^{N} r_n(k)^2|c_k, S_k, [r_m(k)]\} \\
=&E\{\sum_{n=1}^{N} r_n(k)[z_n(q_n) - \mu_n(k)]|c_k, S_k, [r_m(k)]\} + B(k),
\end{aligned}$$

where $B(k) := E\{\frac{1}{2}\sum_{n=1}^{N}[z_n(q_n) - \mu_n(k)]^2|c_k, S_k, [r_m(k)]\}$. Then $B(k) \le B$, for all k, for some B. Hence for any policy:

$$\Delta(L(k)) \le E\{\sum_{n=1}^{N} r_n(k)[z_n(q_n) - \mu_n(k)]|c_k, S_k, [r_m(k)]\} + B. \tag{2}$$

Suppose $[q_n]$ is strictly feasible. The vector $[z_n(q_n)]$ is thus an interior point of the feasible region (for debt) , and there therefore exists some $\alpha \in (0, 1)$ such that $[z_n(q_n)/\alpha]$ is also in the feasible region. Let $z_{min} = \min\{z_1(q_1), z_2(q_2), \ldots, z_N(q_N)\}$. The N-dimensional vector $[z_{min}]$ whose elements are all z_{min} falls in the feasible region. Since the feasible region is convex, the vector $\alpha[z_n(q_n)/\alpha] + (1-\alpha)[z_{min}] = [z_n(q_n) + (1-\alpha)z_{min}]$ is also in the feasible region.

By Lemma 1, there exists a stationary randomized policy η' that fulfills the set of clients with timely-throughput bounds for the vector $[z_n(q_n) + (1-\alpha)z_{min}]$. Since both channel states and packet arrivals can be modeled by a finite-state Markov chain, there exists some constant M such that the expected timely-throughput of client n under η' is at least q'_n in any M consecutive intervals regardless of the system history prior to the M consecutive intervals, where q'_n is chosen so that $z_n(q'_n) \ge z_n(q_n) + (1-\alpha)z_{min} - \frac{(1-\alpha)z_{min}}{2}$. In other words, let $\mu'_n(k)$ be the decrease in the pseudo-debt for client n under η' during interval k, then there exists M such that

$$\begin{aligned}
&E\{\frac{\sum_{i=k}^{k+M-1}\mu'_n(i)}{M}|c_k, S_k, [r_m(k)]\} \\
\ge&z_n(q_n) + (1-\alpha)z_{min} - \frac{(1-\alpha)z_{min}}{2} \\
=&z_n(q_n) + \frac{(1-\alpha)z_{min}}{2},
\end{aligned}$$

for all k.

Let η be a policy that maximizes the payoff function (1), for all k. Then defining $\mu_n(k)$ and $r_n(k)$ as the decrease

resulting from policy η and the pseudo-debt, we have:

$$\sum_{n=1}^{N} E\{r_n(k)^+\mu_n(k)|c_k, S_k, [r_m(k)]\} \geq \sum_{n=1}^{N} E\{r_n(k)^+\mu'_n(k)|c_k, S_k, [r_m(k)]\}.$$

We can assume without loss of generality that the policy does not work on any client n with $r_n(k) \leq 0$, that is, $\mu_n(k) = 0$ if $r_n(k) \leq 0$.[2] From (2), we obtain that, under η:

$$\begin{aligned}
&E\{\frac{L(k+M)}{M} - \frac{L(k)}{M}|c_k, S_k, [r_m(k)]\}\\
=&E\{\frac{\sum_{i=k}^{k+M-1}\Delta(L(i))}{M}|c_k, S_k, [r_m(k)]\}\\
\leq&E\{\sum_{i=k}^{k+M-1}\sum_{n=1}^{N} r_n(i)^+[z_n(q_n) - \mu_n(i)]|c_k, S_k, [r_m(k)]\}/M\\
&+B\\
\leq&E\{\sum_{i=k}^{k+M-1}\sum_{n=1}^{N} r_n(i)^+[z_n(q_n) - \mu'_n(i)]|c_k, S_k, [r_m(k)]\}/M\\
&+B\\
\leq&E\{\sum_{n=1}^{N} r_n(k)^+[z_n(q_n) - \sum_{i=k}^{k+M-1}\frac{\mu'_n(i)}{M}]|c_k, S_k, [r_m(k)]\}\\
&+B+A\\
\leq&-\sum_{n=1}^{N} r_n(k)^+\frac{(1-\alpha)z_{min}}{2} + B + A,
\end{aligned}$$

where A is a constant, as $z_n(q_n)$, $\mu'_n(i)$, and $r_n(i) - r_n(k)$ are all bounded for all n and $i \in [k, k+M-1]$.

Let $\epsilon := (1-\alpha)z_{min}/2$. Let $\hat{L}(k) := L(kM)/M$. By the above discussion, we have $E\{\hat{L}(k+1) - \hat{L}(k)|\text{system history up to the } (kM-1)^{th} \text{ interval}\} \leq A + B - \epsilon\sum_{n=1}^{N} r_n(kM)^+$. By Theorem 1,

$$\limsup_{k\to\infty}\frac{1}{k}\sum_{i=0}^{k} E\{\sum_{n=1}^{N} r_n(kM)^+\} \leq (A+B)/\epsilon. \quad (3)$$

Finally, since $|\sum_{n=1}^{N} r_n(kM+M)^+ - \sum_{n=1}^{N} r_n(kM)^+|$ is bounded for all k, (3) implies that $\frac{1}{k}E\{\sum_{n=1}^{N} r_n(kM)^+\} \to 0$ as $k \to \infty$, as shown in Lemma 3 below. This shows that $\frac{r_n(kM)^+}{k}$ converges to 0 in probability for all n. As $|r_n(kM)^+ - r_n(i)^+|$ is bounded for all $i \in [kM, kM+M]$, and M is also bounded, we have $\frac{r_n(k)^+}{k}$ converges to 0 in probability for all n. Hence, η is feasibility optimal. ■

Lemma 3: Let $f(t)$ be a non-negative function such that $|f(t+1) - f(t)| \leq M$, for some $M > 0$, for all t. If $\limsup_{t\to\infty}\frac{1}{t}\sum_{i=0}^{t} f(i) \leq B$, for some constant B, then $\lim_{t\to\infty}\frac{1}{t}f(t) = 0$.

Proof: We prove by contradiction. Suppose $\limsup_{t\to\infty}\frac{1}{t}f(t) > \delta$, for some $\delta > 0$. Thus, $f(t) > t\delta$ infinitely often. Suppose $f(t) > t\delta$ for some t. Since $|f(t) - f(t-1)| < M$, we have $f(t-1) > t\delta - M$. Similarly, $f(t-2) > t\delta - 2M$, $f(t-3) > t\delta - 3M, \ldots, f(t-\lfloor t\delta/M\rfloor) > t\delta - \lfloor t\delta/M\rfloor M \geq 0$. Summing over these terms gives: $\sum_{i=t-\lfloor t\delta/M\rfloor}^{t} f(i) > \frac{t\delta\lfloor t\delta/M\rfloor}{2}$, and thus, $\sum_{i=0}^{t}\frac{1}{t}f(i) > \frac{\delta\lfloor t\delta/M\rfloor}{2}$. Since $f(t) > t\delta$ infinitely often, $\limsup_{t\to\infty}\sum_{i=0}^{t}\frac{1}{t}f(i) = \infty$, which is a contradiction. ■

Theorem 2 suggests a more general procedure to design feasibility optimal scheduling policies. To design a scheduling policy in a particular scenario, we need to choose an appropriate pseudo-debt and obtain a policy to maximize the payoff function. Maximizing the payoff function is, however, in general, difficult. Nevertheless, in some special cases, evaluating the payoff function gives us simple feasibility optimal policies, or, at least, some insights into designing a reasonable heuristic, as long as we choose the correct pseudo-debt. In the following sections, we demonstrate the utility of this approach.

Finally, in many systems, real-time flows and non-real-time flows coexist. In addition to fulfill the timely-throughput requirements of real-time flows, it is also important not to starve non-real-time flows. Theorem 2 suggests that a feasibility optimal policy does not need to schedule transmissions for clients whose pseudo-debts are negative. By not scheduling these clients, the AP can allocate more time slots for clients with non-real-time traffic and improve their performance.

V. Computationally Tractable Scheduling for Unreliable Time-Varying Channels

We now consider the case when all clients require the same delay bound, $\tau_n \equiv T$, and the AP uses a fixed transmission rate and has instant knowledge of link reliabilities, which are time-varying. In this case, the AP can make exactly one transmission in each time slot. We propose an on-line scheduling policy and prove that it is feasibility optimal.

We use the delivery debt, $r_n^{(3)}(k)$, of Example 1. Thus, $\mu_n(k)$ is the number of packet deliveries for client n in the k^{th} interval.

We call our proposed policy the *joint debt-channel policy*. Suppose at the beginning of an interval, the delivery debt vector is $[r_n^{(3)}(k)]$, the channel state is c, and the set of arrived packets is S. The joint debt-channel policy prioritizes all clients who has packet arrivals and positive $r_n^{(3)}(k)$ according to $r_n^{(3)}(k)p_{c,n}$, where clients with larger $r_n^{(3)}(k)p_{c,n}$ get higher priorities. Algorithm 1 formally describe this policy. The computational complexity of this policy is $O(N\log N)$ per interval.

Algorithm 1 Joint Debt-Channel Policy

1: **for** $n = 1$ to N **do**
2: $\quad r_n^{(3)}(k) \leftarrow q_n k - d_n(kT)$, for all n
3: Sort clients with packet arrivals such that $r_1^{(3)}(k)p_{c,1} \geq r_2^{(3)}(k)p_{c,2} \geq \cdots \geq r_{N_0}^{(3)}(k)p_{c,N_0} > 0 \geq r_{N_0+1}^{(3)}(k)p_{c,N_0+1} \geq \ldots$
4: Transmit packets for clients 1 through N_0 by the ordering

[2]Since a policy cannot lose its feasibility optimality by doing more work, this assumption is not restrictive.

Theorem 3: The joint debt-channel policy is feasibility optimal.

Proof: We first note that the joint debt-channel policy schedules an undelivered packet of client n that maximizes $r_n^{(3)}(k)p_{c,n}$ in every time slot.

Let $\hat{S}(t)$ be the collection of undelivered packets at the t^{th} time slot of an interval, and $\hat{\mu}_n(\hat{S}(t),t)$ be the random variable that indicates the number of packets delivered for client n between the t^{th} time slot and the end of the interval. To simplify notations, we represent each packet by the client that it is associated with. Let $V(\hat{S}(t),t)$ be the value of $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\}$ under the joint debt-channel policy. We show that, given $\hat{S}(t)$ and t, $V(\hat{S}(t),t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\}$ among all policies, for all $t \in [1,T]$.

We prove this by induction on T. When $t=T$, the AP can make exactly one more transmission. If the AP schedules a packet for client m, $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\} = r_m^{(3)}(k)^+p_{c,m}$, which is maximized by the joint debt-channel policy.

Assume that $V(\hat{S}(t),t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\}$ among all policies, for all $t \in [\hat{t}+1,T]$. Now we consider the case when $t=\hat{t}$. Let m be the client that has an undelivered packet and maximizes $r_m^{(3)}(k)^+p_{c,m}$, which is to be scheduled by the joint debt-channel policy. Consider some policy η and let $V^\eta(\hat{S}(t),t)$ be the value of $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\}$ under η.

Suppose η also schedules client m in the t^{th} time slot. We then have $\hat{S}(t+1)=\hat{S}(t)\backslash\{m\}$ with probability $p_{c,m}$, and $\hat{S}(t+1)=\hat{S}(t)$ with probability $1-p_{c,m}$, under both η and the joint debt-channel policy. Thus, we have

$$\begin{aligned}V^\eta(\hat{S}(t),t) =& r_m^{(3)}(k)^+p_{c,m} + p_{c,m}V^\eta(\hat{S}(t)\backslash\{m\},t+1)\\ &+(1-p_{c,m})V^\eta(\hat{S}(t),t+1),\end{aligned}$$

and

$$\begin{aligned}V(\hat{S}(t),t) =& r_m^{(3)}(k)^+p_{c,m} + p_{c,m}V(\hat{S}(t)\backslash\{m\},t+1)\\ &+(1-p_{c,m})V(\hat{S}(t),t+1).\end{aligned}$$

By the induction hypothesis, $V^\eta(\hat{S}(t)\backslash\{m\},t+1) \le V(\hat{S}(t)\backslash\{m\},t+1)$ and $V^\eta(\hat{S}(t),t+1) \le V(\hat{S}(t),t+1)$. Therefore, $V^\eta(\hat{S}(t),t) \le V(\hat{S}(t),t)$.

On the other hand, suppose η schedules some client $l \neq m$ in the t^{th} time slot. We can modify η so that it schedules l in the t^{th} time slot, and then uses the joint debt-channel policy starting from the $(t+1)^{th}$ time slot. By the induction hypothesis, this modification does not decrease $V^\eta(\hat{S}(t),t)$. Also, as client m maximizes $p_{c,m}r_m^{(3)}(k)$, the modified η schedules client m for transmission in the $(t+1)^{th}$ time slot. The resulting policy is equivalent to one that schedules m in the t^{th} time slot, l in the $(t+1)^{th}$ time slot, and then uses the joint debt-channel policy starting from the $(t+2)^{th}$ time slot. Hence, by the arguments in previous paragraphs, $V^\eta(\hat{S}(t),t) \le V(\hat{S}(t),t)$.

In sum, given $\hat{S}(t)$ and t, $V(\hat{S}(t),t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+\hat{\mu}_n(\hat{S}(t),t)\}$ among all policies, for $t \in [\hat{t},T]$ and all $\hat{S}(t)$. Further, at the beginning of an interval, we have $t=1$ and $\hat{S}(1)$ being the set of all packets. Hence, the joint debt-channel policy maximizes $\sum_{n=1}^N E\{r_n(k)^+\mu_n(k)|c_k,S_k,[r_m(k)]\}$. By Theorem 2, the joint debt-channel policy is feasibility optimal. ■

VI. A Scheduling Policy without Instant Channel Knowledge

In this section, we address the case where all clients require the same delay bound, i.e., $\tau_n \equiv T$, the AP does not employ rate adaptation, i.e., all transmissions take one time slot, and the AP does not have full instant knowledge on channel qualities. In particular, we assume that, under channel state c, the link reliability between client n and the AP can be among a finite number of values, namely, $\hat{p}_{c,n}(1),\hat{p}_{c,n}(2),\ldots$, and the probability that this link reliability equals $\hat{p}_{c,n}(i)$ is denoted by $\phi_{c,n}(i)$. Thus, at the beginning of the interval, the expected value of link reliability between client n and the AP is $\sum_i \hat{p}_{c,n}(i)\phi_{c,n}(i)$. Finally, we assume that each client can generate at most one packet in each interval. We will discuss why this assumption is necessary in the end of this section.

While the AP does not know the exact value of link reliability for each client, it can estimate the link reliability and update its estimation by calculating the posterior probability distribution of the link reliability based on events in the interval. Since link reliabilities for different clients are independent from each other, the AP's estimation for the link reliability of client n is only influenced by events regarding transmissions for client n. Moreover, if the packet for client n is successfully delivered in a transmission, the AP no longer needs to estimate the link reliability of client n, as it does not have any more packets for client n, and hence cannot schedule any more transmissions for client n, in the interval. Thus, the AP only needs to estimate the link reliability for client n by the number of failed transmissions for client n in the interval. We hereby denote by $E[p_{c,n}|\xi_n]$ as the conditional expected value of client n under channel c, given that there are ξ_n failed transmissions for client n in the interval. Note that $E[p_{c,n}|\xi_n]$ is not only the AP's estimate of link reliability, but also the conditional probability that the next transmission of n is successful, given the system history.

We use the delivery debt, $r_n^{(3)}(k)$, of Example 1. Thus, $\mu_n(k)=1$ if there is a packet delivery for client n in the k^{th} interval, and $\mu_n(k)=0$, otherwise.

We propose a policy, which is called the *largest expectation policy*, for this scenario. Suppose that, at some time slot in an interval, the packet for client n has been transmitted ξ_n times in this interval. The AP then schedules the client n who maximizes $E[p_{c,n}|\xi_n]r_n^{(3)}(k)$ among those whose packets are not yet delivered for transmission in this time slot. The largest expectation policy is illustrated in Algorithm 2.

The largest expectation policy turns out to be feasibility optimal. Before proving its optimality, we first show that $E[p_{c,n}|\xi_n]$ is non-increasing in ξ_n. That is, as the number of failed transmissions increases, the AP's estimation for link reliability decreases.

Lemma 4: For any c,n, $E[p_{c,n}|0] \ge E[p_{c,n}|1] \ge E[p_{c,n}|2] \ge \ldots$.

Proof: Suppose the link reliability for client n is $\hat{p}_{c,n}(i)$, the probability that transmissions for client n fail

Algorithm 2 Largest Expectation Policy

1: **for** $n = 1$ to N **do**
2: $r_n^{(3)}(k) \leftarrow q_n k - d_n(kT)$, for all n
3: $\xi_n \leftarrow 0$
4: **for** each time slot in the interval **do**
5: Transmit packet for the client n who maximizes $E[p_{c,n}|\xi_n]r_n^{(3)}(k)$
6: $\xi_n \leftarrow \xi_n + 1$

ξ times without succeeding is $(1-\hat{p}_{c,n}(i))^{\xi}$. Thus, using Bayes' Theorem, we have

$$E[p_{c,n}|\xi] = \frac{\sum_i \hat{p}_{c,n}(i)\phi_{c,n}(i)(1-\hat{p}_{c,n}(i))^{\xi}}{\sum_j \phi_{c,n}(j)(1-\hat{p}_{c,n}(j))^{\xi}},$$

and therefore

$$\begin{aligned}
&E[p_{c,n}|\xi] - E[p_{c,n}|\xi+1] \\
=&\frac{\sum_i \hat{p}_{c,n}(i)\phi_{c,n}(i)(1-\hat{p}_{c,n}(i))^{\xi}}{\sum_j \phi_{c,n}(j)(1-\hat{p}_{c,n}(j))^{\xi}} \\
&- \frac{\sum_i \hat{p}_{c,n}(i)\phi_{c,n}(i)(1-\hat{p}_{c,n}(i))^{\xi+1}}{\sum_j \phi_{c,n}(j)(1-\hat{p}_{c,n}(j))^{\xi+1}}.
\end{aligned}$$

Let $B := \sum_i \phi_{c,n}(i)(1-\hat{p}_{c,n}(i))^{\xi} \sum_j \phi_{c,n}(j)(1-\hat{p}_{c,n}(j))^{\xi+1}$ and $A_{i,j} := \phi_{c,n}(i)\phi_{c,n}(j)(1-\hat{p}_{c,n}(i))^{\xi}(1-\hat{p}_{c,n}(j))^{\xi} = A_{j,i}$, we can further derive

$$\begin{aligned}
&E[p_{c,n}|\xi] - E[p_{c,n}|\xi+1] \\
=&\frac{\sum_{i\neq j} A_{i,j}[\hat{p}_{c,n}(i)(\hat{p}_{c,n}(i) - \hat{p}_{c,n}(j))]}{B} \\
=&\{\sum_{i<j} A_{i,j}[\hat{p}_{c,n}(i)(\hat{p}_{c,n}(i) - \hat{p}_{c,n}(j)) \\
&+ \hat{p}_{c,n}(j)(\hat{p}_{c,n}(j) - \hat{p}_{c,n}(i))]\}/B \\
=&\frac{\sum_{i<j} A_{i,j}(\hat{p}_{c,n}(i) - \hat{p}_{c,n}(j))^2}{B} \\
\geq&0.
\end{aligned}$$

Hence, $E[p_{c,n}|0] \geq E[p_{c,n}|1] \geq E[p_{c,n}|2] \geq \ldots$, for any c, n. ■

We now show that the largest expectation policy is feasibility optimal.

Theorem 4: The largest expectation Policy is feasibility optimal.

Proof: The proof is similar to the proof of Theorem 3. Let $\hat{S}(t)$ be the set of clients that have an undelivered packet at the t^{th} time slot of an interval ,$\xi_n(t)$ be the number of failed transmissions that the AP has scheduled for client n before the t^{th} time slot, $\xi(t)$ be the vector containing $\xi_n(t)$, for all n, and $\hat{\mu}_n(\hat{S}(t), \xi(t), t)$ be the random variable that indicates there is a successful packet delivery for client n between the t^{th} time slot and the end of the interval. Let $V(\hat{S}(t), \xi(t), t)$ be the value of $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), t)\}$ under the largest expectation policy, given channel state c. We show that $V(\hat{S}(t), t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), \xi(t), t)\}$ among all policies, for all $t \in [1, T]$.

We prove this by induction on T. When $t = T$, the AP can make exactly one more transmission. If the AP schedules client m, $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), \xi(t), t)\} = r_m^{(3)}(k)^+ E[p_{c,m}|\xi_m(t)]$, which is maximized by the largest expectation policy.

Assume that $V(\hat{S}(t), \xi(t), t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), \xi(t), t)\}$ among all policies, for all $t \in [\hat{t}+1, T]$. Now we consider the case when $t = \hat{t}$. Let m be the client that has an undelivered packet and maximizes $r_m^{(3)}(k)^+ E[p_{c,m}|\xi_m(t)]$, which is to be scheduled by the largest expectation policy. Consider some policy η and let $V^{\eta}(\hat{S}(t), \xi(t), t)$ be the value of $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), \xi(t), t)\}$ under η.

Suppose η also schedules client m in the t^{th} time slot. We then have $\xi_m(t+1) = \xi_m(t) + 1$, $\xi_n(t+1) = \xi_n$, for all $n \neq m$, $\hat{S}(t+1) = \hat{S}(t) \backslash \{m\}$ with probability $E[p_{c,m}|\xi_m(t)]$, and $\hat{S}(t+1) = \hat{S}(t)$ with probability $1 - E[p_{c,m}|\xi_m(t)]$, under both η and the largest expectation policy. Thus, we have

$$\begin{aligned}
&V^{\eta}(\hat{S}(t), \xi(t), t) = \\
&r_m^{(3)}(k)^+ E[p_{c,m}|\xi_m(t)] \\
&+ E[p_{c,m}|\xi_m(t)] V^{\eta}(\hat{S}(t)\backslash\{m\}, \xi(t+1), t+1) \\
&+ (1 - E[p_{c,m}|\xi_m(t)]) V^{\eta}(\hat{S}(t), \xi(t+1), t+1),
\end{aligned}$$

and

$$\begin{aligned}
&V(\hat{S}(t), \xi(t), t) \\
=&r_m^{(3)}(k)^+ E[p_{c,m}|\xi_m(t)] \\
&+ E[p_{c,m}|\xi_m(t)] V(\hat{S}(t)\backslash\{m\}, \xi(t+1), t+1) \\
&+ (1 - E[p_{c,m}|\xi_m(t)]) V(\hat{S}(t), \xi(t+1), t+1).
\end{aligned}$$

By the induction hypothesis, $V^{\eta}(\hat{S}(t)\backslash\{m\}, \xi(t+1), t+1) \leq V(\hat{S}(t)\backslash\{m\}, \xi(t+1), t+1)$ and $V^{\eta}(\hat{S}(t), \xi(t+1), t+1) \leq V(\hat{S}(t), \xi(t+1), t+1)$. Therefore, $V^{\eta}(\hat{S}(t), \xi(t), t) \leq V(\hat{S}(t), \xi(t), t)$.

On the other hand, suppose η schedules some client $l \neq m$ in the t^{th} time slot. We can modify η so that it schedules l in the t^{th} time slot, and then uses the largest expectation policy starting from the $(t+1)^{th}$ time slot. By the induction hypothesis, this modification does not decrease $V^{\eta}(\hat{S}(t), \xi(t), t)$. Also, by scheduling l in the t^{th} time slot, we have $\xi_l(t+1) = \xi_l(t) + 1$, $\xi_m(t+1) = \xi_m(t)$, and $\xi_n(t+1) = \xi_n(t)$, for all n other than l and m. Therefore, by Lemma 4, $E[p_{c,l}|\xi_l(t+1)] \leq E[p_{c,l}|\xi_l(t)]$, $E[p_{c,m}|\xi_m(t+1)] = E[p_{c,m}|\xi_m(t)]$, and $E[p_{c,n}|\xi_n(t+1)] = E[p_{c,n}|\xi_n(t)]$, for all n other than l and m. As client m maximizes $E[p_{c,m}|\xi_m(t)]r_m^{(3)}(k)$ in the t^{th} time slot, it also maximizes $E[p_{c,m}|\xi_m(t+1)]r_m^{(3)}(k)$ in the $(t+1)^{th}$ time slot, and hence the modified η schedules client m for transmission in the $(t+1)^{th}$ time slot. The resulting policy is equivalent to one that schedules m in the t^{th} time slot, l in the $(t+1)^{th}$ time slot, and then uses the largest expectation policy starting from the $(t+2)^{th}$ time slot. Hence, by the arguments in previous paragraphs, $V^{\eta}(\hat{S}(t), \xi(t), t) \leq V(\hat{S}(t), \xi(t), t)$.

In sum, $V(\hat{S}(t), \xi(t), t)$ is the maximum of $E\{\sum_n r_n^{(3)}(k)^+ \hat{\mu}_n(\hat{S}(t), \xi(t), t)\}$ among all policies, for $t \in [\hat{t}, T]$. By induction and Theorem 2, the largest expectation policy is feasibility optimal. ■

We have assumed that each client can generate at most one packet in each interval in this section. One may wonder whether the largest expectation policy is

still feasibility optimal when clients may generate more than one packet in an interval. The following example shows that the largest expectation policy does not always maximize $E\{\sum_n r_n^{(3)}(k)^+\mu_n(k)\}$, and hence may not be feasibility optimal, when clients may generate more than one packet in an interval.

Example 2: Consider a system with two clients and two time slots in an interval. At the beginning of the k^{th} interval, each client generates two packets. The link reliability of client 1 can either be 100% or 0%, each with probability 0.5. Hence, the expected value of link reliability of client 1 is 0.5. The link reliability of client 2 is 100%. Finally, we assume $r_1^{(3)}(k) = 10$ and $r_2^{(3)}(k) = 6$.

The largest expectation policy schedules transmissions for client 2 in both time slots in the interval, and hence $E\{\sum_n r_n^{(3)}(k)^+\mu_n(k)\} = 12$ under the largest expectation policy. Consider another policy, η, that schedules client 1 in the first time slot. If the transmission succeeds, which means that the link reliability of client 1 is indeed 100%, η schedules client 1 in the second time slot. On the other hand, if the transmission fails, η schedules the transmission for client 2 in the second time slot. With some calculations, it can be shown that $E\{\sum_n r_n^{(3)}(k)^+\mu_n(k)\} = 13$ under η. Thus, the largest expectation policy does not maximize $E\{\sum_n r_n^{(3)}(k)^+\mu_n(k)\}$. □

The reason that the largest expectation policy does not maximize $E\{\sum_n r_n^{(3)}(k)^+\mu_n(k)\}$ when clients can generate more than one packet in an interval is because Lemma 4 no longer holds. Lemma 4 states that the expected value of link reliability of a client decreases with the number of transmissions scheduled for it. This only holds when all previously scheduled transmissions fail, and the expected value of link reliability is no longer useful when a transmission for the client succeeds. However, when the client generates more than one packet, it can still be scheduled after a successful transmission, and hence the expected value of link reliability is still needed after a successful transmission.

VII. A Heuristic for Heterogeneous Delay Bounds

We now describe a heuristic for packet scheduling, for the case where each channel state is static and transmission rate is fixed, but clients require different delay bounds. We use p_n to represent channel reliability.

We will use the time-based debt, $r_n^{(1)}(k)$, as discussed in Example 1. The payoff function is $E\{\sum_{n=1}^N r_n^{(1)}(k)^+\mu_n(k)\}$.

We assume that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$. Let $\gamma_{n,i}$ be the number of transmissions the AP needs to make for client n in order to deliver the i^{th} packet generated by n. While $\gamma_{n,i}$ is a random variable that cannot be foretold, we examine how to maximize $\sum_{n=1}^N r_n^{(1)}(k)^+\mu_n(k)$ if we *knew* $\gamma_{n,i}$.

We solve this by proceeding backwards in time. During time slots $[\tau_{N-1}+1, \tau_N]$, all packets except the one for client N have expired, and we can only make transmissions for client N during these time slots. Thus, it does not make sense to schedule client N for more than $\gamma_N^{N-1} := \sum_i \gamma_{N,i} - (\tau_N - \tau_{N-1})$ transmissions before time slot τ_{N-1}. Next, in the time slots between $[\tau_{N-2}+1, \tau_{N-1}]$, only clients $N-1$ and N can be scheduled. An obvious choice is to schedule the client with larger debt first, with the restriction that it is not scheduled for more than γ_n^{N-1} time slots, and to then schedule the other client. (For simplicity, we let $\gamma_{N-1}^{N-1} := \sum_i \gamma_{N-1,i}$.) We can further obtain the remaining transmissions allowed for client n before time slot τ_{N-2}, which we call γ_n^{N-2}, as γ_n^{N-1} minus the number of transmissions scheduled for client n during time slots $[\tau_{N-2}+1, \tau_{N-1}]$. Transmissions of the remaining time slots are scheduled similarly.

While it is impossible to know the exact value of $\gamma_{n,i}$ in advance, we can estimate it. One estimate is its expected value, $\frac{1}{p_n}$. However, this estimate does not consider the timely-throughput requirements. If a client has significantly larger debt than others, a reasonably good policy would allocate enough time slots so that the probability of delivery for each packet generated by the client in this interval is at least its delivery ratio bound. So we estimate $\gamma_{n,i}$ by the number of transmissions that we need to allocate for client n so that it can achieve its delivery ratio bound. This estimate $\gamma_{n,i}$ is $\lceil \log_{1-p_n}(1 - \{\text{delivery ratio bound of } n\})\rceil$. The total number of time slots allocated for client n can then be expressed as

$$\gamma_n = \lceil \log_{1-p_n}(1 - \{\text{delivery ratio bound of } n\})\rceil \times \{\text{number of packets generated by } n\}. \quad (4)$$

We thus derive the Adaptive-Allocation Policy shown in Algorithm 3.

Algorithm 3 Adaptive-Allocation Policy

for $n = 1$ to N **do**
  $r_n^{(1)}(k) \leftarrow$ time-based debt
  Set γ_n by Eq. (4)
Sort clients so $r_1^{(1)}(k) \geq r_2^{(1)}(k) \geq \cdots \geq r_N^{(1)}(k)$
$alloc \leftarrow n \times 1-$vector
for $t = T$ to 1 **do**
  $n \leftarrow 1$
  while ($\tau_n < t$ or $\gamma_n \leq 0$) and $n \leq N$ **do**
    $n \leftarrow n + 1$
  if $r_n^{(1)}(k) > 0$ **then**
    $alloc[t] \leftarrow n$
  else
    $alloc[t] \leftarrow N + 1$
  if $n \leq N$ **then**
    $\gamma_n \leftarrow \gamma_n - 1$
for each time slot t **do**
  if $alloc[t] \leq N$ and the packet for client $alloc[t]$ has not been delivered **then**
    transmit the packet for client $alloc[t]$
  else
    transmit the packet with the largest positive time-based debt

VIII. Scheduling Policy with Rate Adaptation

We now propose a feasibility optimal scheduling policy when rate adaptation is employed. Channel qualities can be time-varying and clients may have different deadlines.

To derive the scheduling policy, we use the delivery debt $r_n^{(3)}(k)$. Thus, $z_n(q_n) := q_n$, while $\mu_n(k)$ is the number of packets delivered for client n.

Suppose at the beginning of interval k, the delivery debt vector is $[r_n^{(3)}(k)]$, the channel state is c, and the collection of arrived packets is S. If a client generates more than one packet, say , i packets, in the interval, we instead assume that there are i clients that correspond to this client in the system, each of them corresponds to one packet that is generated by the original client. The i clients have the same delivery debt and channel condition as that of the original client. Hence, we can assume that each client generates at most one packet in an interval for the remaining of this section. This assumption only serves to simplify notations and does not have any meaningful impact on theoretical analysis.

We represent each packet in S by the client that generates it. The transmission time for client n is $s_{c,n}$ time slots, and client n stipulates a delay bound of τ_n. Since transmissions are assumed to be error-free when rate adaptation is applied, the scheduling policy consists of finding an ordered subset $S' = \{m_1, m_2, \ldots, m_{N'}\}$ of S such that $\sum_{n=1}^{l} s_{c,m_n} \leq \tau_{m_l}$, for all $1 \leq l \leq m_{N'}$. That is, when clients are scheduled according to the ordering, no packets for clients in S' would miss their respective delay bounds. By Theorem 2, a policy using an ordered set S' that maximizes $\sum_{n \in S'} r_n^{(3)}(k)$ with the above constraint is feasibility optimal. This is a variation of the knapsack problem. When S' is selected, reordering clients in S' in an earliest-deadline-first fashion also allows all packets to meet their respective delay bounds. Based on this observation, we derive the feasibility optimal scheduling algorithm, the Modified Knapsack Algorithm, as shown in Algorithm 4.

The intuition of Algorithm 4 is as follows: Let $M[n,t]$ be the maximum debt a policy can collect if only clients 1 through n can be scheduled and all transmissions need to complete before time slot t. Thus, $\max_{S'} \sum_{n \in S'} r_n^{(3)}(k) = M[N,T]$. Also, iteratively:

$$M[n,t] = \begin{cases} M[n,t-1] & \text{if } t > \tau_n, \\ \max\{M[n-1,t], & \\ r_n^{(3)}(k) + M[n-1,t-s_{c,n}]\} & \text{otherwise,} \end{cases}$$

where $M[n-1,t]$ is the maximum debt can be collected when client n is not scheduled, and $r_n^{(3)}(k) + M[n-1,t-s_{c,n}]$ is that when client n is scheduled. Algorithm 4 then applies dynamic programming to compute $M[n,t]$ according to the above recursion. Interested readers may refer to Section 17.2 of [20] for more details on the knapsack problem. The complexity of this algorithm is $O(NT)$, and it is thus reasonably efficient.

Algorithm 4 Modified Knapsack Policy

for $n = 1$ to N **do**
  $r_n^{(3)}(k) \leftarrow q_n k - d_n(kT)$
Sort clients such that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$
for $n = 1$ to N **do**
  $M[n,0] \leftarrow 0$
  $S'[n,0] \leftarrow \phi$
for $n = 1$ to N **do**
  for $t = 1$ to T **do**
    if $t > \tau_n$ **then**
      $M[n,t] \leftarrow M[n,t-1]$
      $S'[n,t] \leftarrow S'[n,t-1]$
    else if client n has a packet AND $r_n^{(3)}(k) + M[n-1,t-s_{c,n}] > M[n-1,t]$ **then**
      $M[n,t] \leftarrow r_n^{(3)}(k) + M[n-1,t-s_{c,n}]$
      $S'[n,t] \leftarrow S'[n-1,t-s_{c,n}] + \{n\}$
    else
      $M[n,t] \leftarrow M[n-1,t]$
      $S'[n,t] \leftarrow S'[n-1,t]$
schedule according to $S'[N,T]$

IX. Simulation Results

We have implemented the scheduling policies discussed in previous sections by using the IEEE 802.11 PCF standard in the *ns-2* simulator. We present the simulation results for the scenario with time-varying channels, and with clients requiring different delay bounds. In each scenario, we compare our policies against the two largest debt first policies of [1], and a policy that assigns priorities to clients randomly, *random*. IEEE 802.11e, an enhancement to 802.11 for QoS, allows clients with real-time traffic to use smaller contention window and inter frame space to obtain priorities over clients with non-real-time traffic. However, clients with real-time traffic have to compete with each other in a random access manner with equal channel access probabilities, without any QoS based preference or discrimination. Further, the inter frame space and contention window size are smaller in PCF than in 802.11e. Thus, the random policy can be viewed as an improved version of 802.11e. Similar to the previous work, we conduct two sets of simulations for each scenario, one with clients carrying VoIP traffic, and one with clients carrying video streaming traffic. The major difference between the two settings lies in their traffic patterns. Many VoIP codecs generate packets intervalically. Thus, future packet arrivals can be easily predicted and may be dependent among different clients. For example, if two clients generate packets at the same rate, then either all or none of their packets arrive simultaneously. On the other hand, video streaming technology, such as MPEG, may generate traffic with variable-bit-rate (VBR). Thus, packets arrive at the AP probabilistically, with probability depending on the context of the current frame, and arrivals are independent among different clients.

For the VoIP traffic, we follow the standards of the ITU-T G.729.1 [21] and G.711 [22] codecs. Both codecs generate traffic intervalically. G.729.1 generates traffic with bit rates 8 – 32 kbits/s, while G.711 generates traffic at a higher rate of 64 kbits/s. We assume the interval length, T, is 20 ms, and the payload size of a packet is 160 Bytes. The codecs generate one packet every several intervals; with the duration between packet arrivals depending on the bit rate used.

We use MPEG for the video streaming setting. MPEG VBR traffic is usually modeled as a Markov chain consisting of three activity states [23] [24]. Each state generates traffic probabilistically at different mean rates, with the state being determined by the current frame of the video. The statistical mean rates in each state are those obtained in an experimental study [24]. We use them in setting the traffic patterns of MPEG traffic. We assume the interval

TABLE I: MPEG Traffic Pattern

Activity	Great	High	Regular
Data rate	501597	392237	366587
Arrival probability	1	0.8	0.75

length to be 6 ms and the payload size of a packet to be 1500 Bytes. Table I shows the statistical results of the experimental study [24], where we also present them in terms of the packet arrival probability of our setting. In Table I, "Data rate" is measured in bits/GoP, where 1 GoP= 240 ms.

We simulate 20 runs for each setting, each run lasting one minute in simulated time. All results shown are averaged over the 20 runs. We use the *total timely-throughput deficiency*, defined as the sum of $(q_n -$ actual timely-throughput of $n)^+$ over all clients, as the performance metric. In addition to evaluating how well the tested policies serve clients with real-time traffic, we also wish to know whether the policies starve those with non-real-time traffic. Hence we add a client with saturated non-real-time traffic in all simulations. Packets for the non-real-time client are scheduled in all time slots that are left idle otherwise. We measure the throughput of the client with non-real-time traffic by the average number of packets delivered.

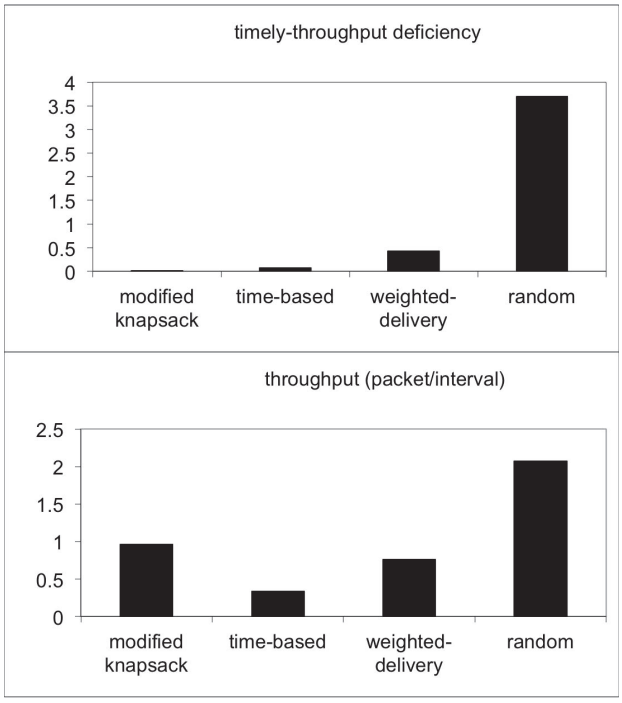

Fig. 1: Performance for VoIP traffic with rate adaptation.

A. Rate Adaptation

We present the simulation results under the scenario where rate adaptation is applied, channels are time-varying, and clients may require different delay bounds.

We first show the results for VoIP traffic. We use IEEE 802.11b as the MAC protocol, which can provide a maximum data rate of 11 Mb/s. We assume that the channel capacity of each client alternates between 11 Mb/s and 5.5 Mb/s. Simulation results suggest that the times needed for a transmission, including all MAC overheads such as the time for waiting an ACK, are around 480 μs and 610 μs for the two transmission rates, respectively. Ideally, the length of a time slot should be a common divisor of the transmission times needed under the two used data rates. We approximate this value by 160 μs. Thus, transmitting a packet requires 3 time slots when using 11 Mb/s and 4 time slots when using 5.5 Mb/s. Further, an interval consists of 125 time slots.

There are two groups of clients, A and B. Clients in group A generate one packet every three intervals, or at rate 21.3 kbits/s, and require 90% of each of the clients' packets to be delivered, or a timely-throughput requirement of 19.2 kbits/s. Clients in group B generate one packet every two intervals at rate 32 kbits/s, and require 70% of each of the clients' packets to be delivered, corresponding to a timely-throughput requirement of 22.4 kbits/s. The two groups can be further divided into subgroups, A_1, A_2, A_3, B_1, and B_2, each with 22 clients. Clients in subgroup A_i generate packets at intervals $[i, i+3, i+6, \dots]$, and clients in subgroup B_i generate packets at intervals $[i, i+2, i+4, \dots]$. Finally, clients in group A require a delay bound equal to the interval length, or 125 time slots, while clients in group B require a delay bound equal to two-third of the interval length, or 83 time slots.

Simulation results are shown in Figure 1. The modified knapsack policy incurs the least total timely-throughput deficiency, which is less than 0.003 packet/interval, among all evaluated policies. This is because all the other three policies neglect the time-varying channels with different data rates and the heterogeneous delay bounds. Further, by only scheduling those clients with positive delivery debts, the modified knapsack policy achieves higher throughput for the non-real-time client than both the policies proposed in [1]. The random policy results in the highest throughput for the non-real-time client. However, this is because it sacrifices the real-time clients. In fact, its total timely-throughput deficiency is as large as 3.7 packet/interval. This huge difference suggests that the random policy, and therefore also 802.11e, are not adequate for providing QoS when multiple clients with real-time traffic are present.

Next we consider the scenario with MPEG traffic. Since video streaming requires much higher bandwidth than VoIP, we use 802.11a as the underlying MAC, which can support up to 54 Mb/s. We assume that channel capacity for each client alternates between 54 Mb/s and 24 Mb/s. The transmission times for a data-ACK handshake require 660 μs with 54 Mb/s data rate, and 940 μs with 24 Mb/s. The length of a time slot is 60 μs. Thus, the transmission times for the two data rates are 11 time slots and 16 time slots, respectively. Further, an interval consists of 100 time slots.

We again assume there are two groups of clients. Clients in group A generate packets according to Table I, and clients in group B are assumed to offer only lower quality video by generating packets only 80% as often as those in group A, in each of the three states. We assume clients in group A require 90% delivery ratios, and clients in group B require 60% delivery ratios. Since the length of

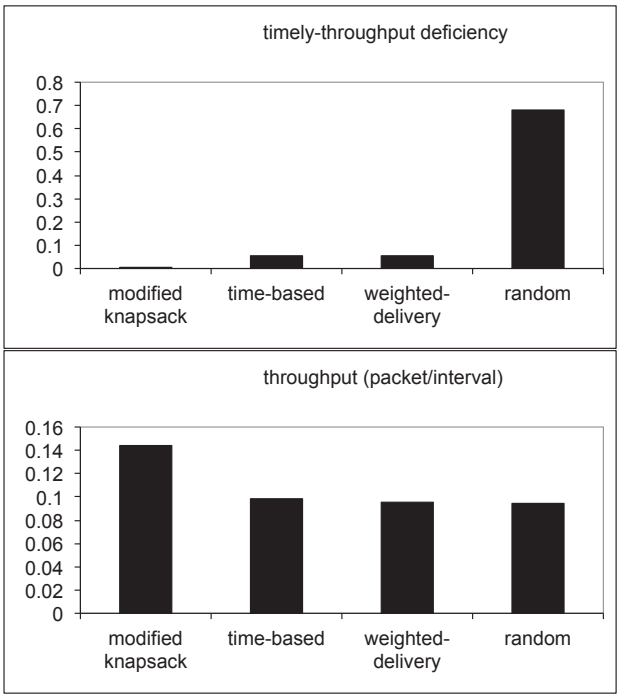

Fig. 2: Performance for MPEG traffic with rate adaptation.

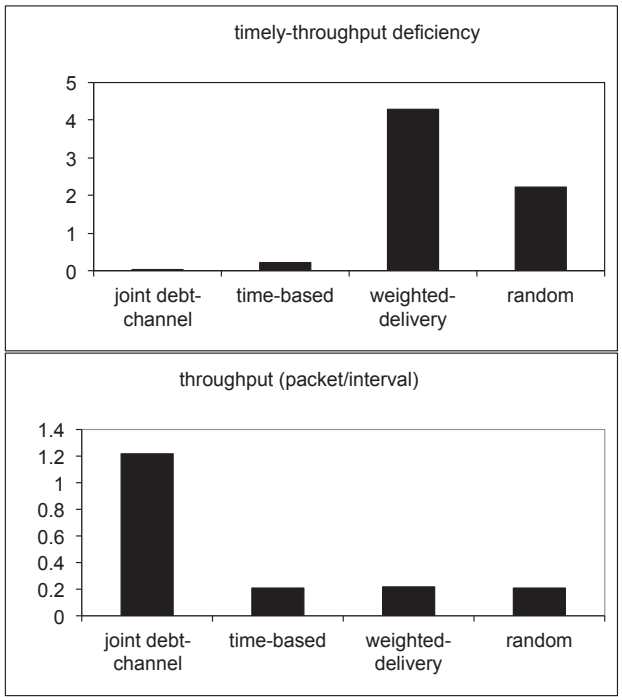

Fig. 3: Performance for VoIP traffic under time-varying channels.

an interval for MPEG is very small, it is less meaningful to discuss heterogeneous delay bounds. Thus, we assume all clients require a delay bound equal to the length of an interval. We further assume that there are 6 clients in both groups.

Simulation results are shown in Figure 2. As in the case of VoIP traffic, the modified knapsack policy achieves the smallest total timely-throughput deficiency among all the four policies. Also, by not scheduling clients with non-positive debts, the modified knapsack policy also achieves the highest throughput for the non-real-time client.

B. Time-varying Channels

We now consider the scenario with time-varying channels, with all clients requiring delay bounds equal to interval length. We model the wireless channel by the widely used Gilbert-Elliot model [25] [26] [27], with the wireless channel considered as a two-state Markov chain, with "good" state and "bad" states. A simulation study by Bhagwat et al [17] shows that the link reliability can be modeled as 100% when the channel is in the good state, and 20% when the channel is in the bad state. The duration that the channel stays in one state is exponentially distributed with mean 1 – 10 sec for the good state, and 50 – 500 msec for the bad state.

As the two largest debt first policies are not directly applicable to time-varying channels, we consider some simple modifications for them. For the largest time-based debt first policy, we modify it so that it treats the channel as a static one, with link reliability equal to the time-averaged link reliability. For the largest weighted-delivery debt first policy, the weighted-delivery debt for client n at time slot t is defined as $\frac{t}{T}q_n - d_n(t)$ divided by the current link reliability.

For the case of VoIP traffic, we use 802.11b as the underlying MAC and use a fixed transmission rate of 11 Mb/s. We consider the same two groups of clients as in the previous section. We assume that the mean duration of the bad state is 500 msec for all clients, and the mean duration of the good state is $1+0.5n$ sec for the n^{th} client in each subgroup. The time-average link reliability of the n^{th} client in each subgroup can be computed as $\frac{2.2+n}{3+n}$. There are 19 clients in each of the subgroups.

Simulations results are shown in Figure 3. The joint debt-channel policy incurs near zero total timely-throughput deficiency, while all the other policies have much larger total timely-throughput deficiency. The fact that the largest time-based debt first policy fails to fulfill the set of clients suggests that only considering the average channel reliability, without taking channel dynamics into account, is not satisfactory. A somewhat surprising result is that the total timely-throughput deficiency for the largest weighted-delivery debt first policy is even larger than that for the random policy. This is because the policy favors those clients with poor channels. When the channel state is time-varying, it may make more sense to postpone the transmissions for a client with a poor channel until its channel condition turns better. Thus, using weighted-delivery debt for time-varying channels is not only inaccurate, but even harmful in some settings. It can also be shown that the throughput for the client with saturated non-real-time traffic is the highest with the joint debt-channel policy. By only scheduling those real-time clients with positive delivery debts, the policy prevents putting too much effort into any real-time client, and thus reserves enough resources for clients with non-realtime traffic.

For MPEG traffic, we assume there are two groups

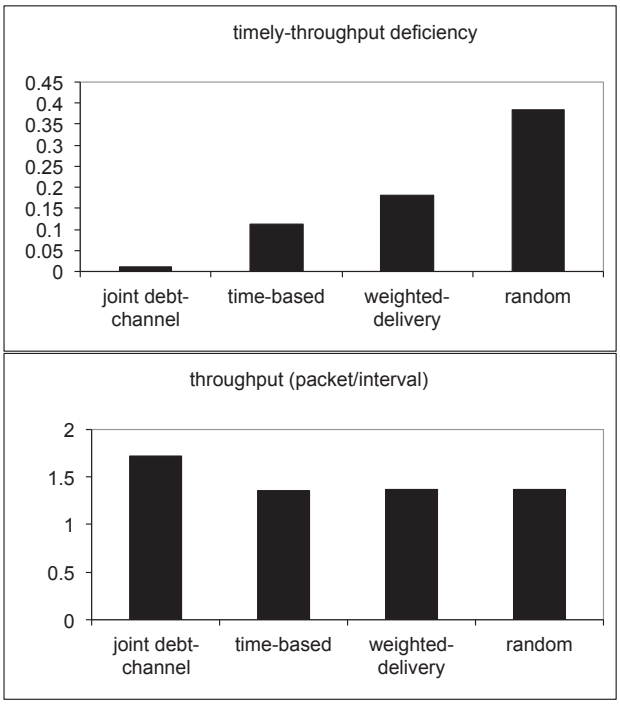

Fig. 4: Performance for MPEG traffic under time-varying channels

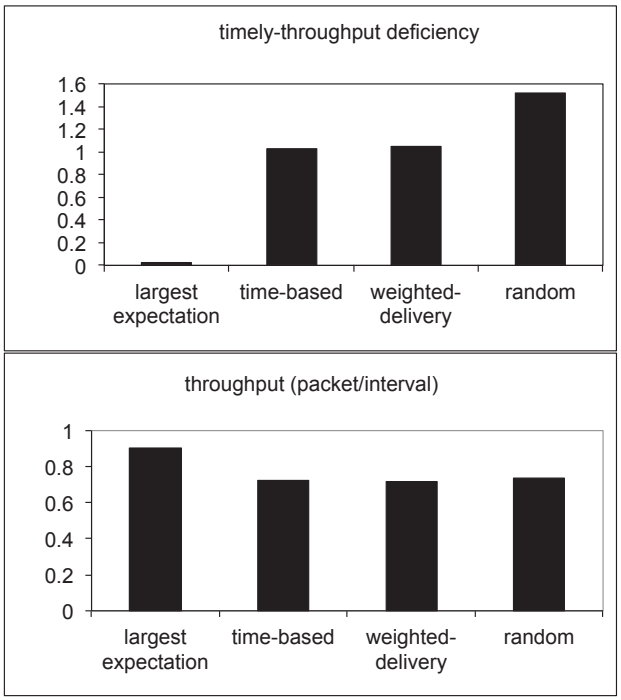

Fig. 5: Performance for VoIP traffic when AP does not have instant channel information

of clients, with the same traffic patterns and delivery ratio requirements as those in the previous section. We use 802.11a with a fixed data rate of 54 Mb/s as the underlying MAC. The mean duration when the channel is in the bad state is 500 msec for all clients, and the mean duration in the good state is assumed to be $1 + 0.5n$ sec for the n^{th} client in each group. There are 4 clients in both groups.

Simulation results are shown in Figure 4. As in the case of VoIP traffic, the joint debt-channel policy incurs very small total timely-throughput deficiency while all the other policies have significantly higher total timely-throughput deficiency. This result suggests that the simple modifications of the two largest debt first policies do not work under time-varying channels. Also, by only scheduling real-time clients with positive delivery debts, the joint debt-channel policy achieves higher throughput for the client with non-real-time traffic.

C. Time-Varying Channels without Instant Channel Knowledge

In this section, we present simulation results when the AP does not have instant knowledge on channel qualities. We assume that, in each interval, the link reliability between the AP and a client is either 20%, 50%, or 80%, each with probability 1/3. Thus, the expected link reliability for a client is 50%. As in the simulations of the previous section, both the largest time-based debt first policy and the largest weighted-delivery debt first policy compute their respective debts by assuming the link reliability of a client equal to the expected link reliability. Table II shows the posterior expected link reliability of a client, given the number of failed transmissions for the client in an interval.

For VoIP traffic, we use 802.11b with 11 Mb/s as the underlying MAC. We also assume that there are two groups, A and B, of clients. Clients in group A generate one packet every three intervals and require 90% of their packets to be delivered on time, while clients in group B generate one packet every two intervals and require 70% of their packets to be delivered on time. There are a total number of 30 group A clients and 20 group B clients.

Simulation results are shown in Figure 5. Our proposed policy, the largest expectation policy, incurs very small total timely-throughput deficiency, while all the other policies incur much higher total timely-throughput deficiency. This is because the largest expectation policy can identify clients with poor link reliabilities by events occurred in the interval, and thus avoiding waste too much time on those clients. As the largest expectation policy uses its time slots more efficiently, it also saves more time slots for the non-real-time flow.

For MPEG traffic, we use 802.11a with 54 Mb/s as the underlying MAC. We assume that there are two groups, A and B, of clients. Clients in group A generate packets according to Table I and require 90% of their packets to be delivered on time. Clients in group B generate packets only 80% as often as those in group A and require 60% of their packets to be delivered on time. We assume that there are 2 clients in group A and 5 clients in group B.

Simulation results are shown in Figure 6. As in the VoIP traffic, the largest expectation policy achieves the smallest total timely-throughput deficiency among the four policies.

D. Heterogeneous Delay Bounds

Now, we study the scenario where the channel state is static but clients require different delay bounds. Since

TABLE II: Posterior expected link reliability

Number of failed transmissions	0	1	2	3	4	5	6	7	8	9
Posterior expected link reliability	50%	38%	30.6%	26.6%	24.2%	22.7%	21.7%	21.1%	20.7%	20.4%

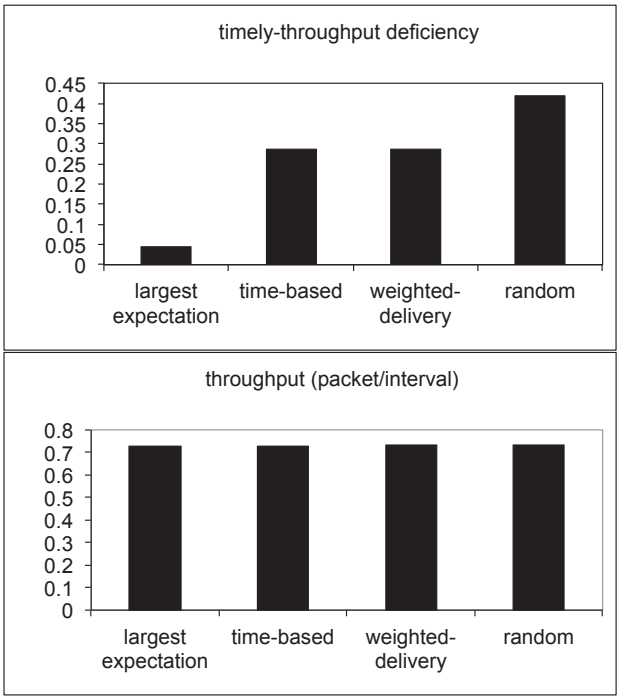

Fig. 6: Performance for MPEG traffic when AP does not have instant channel information

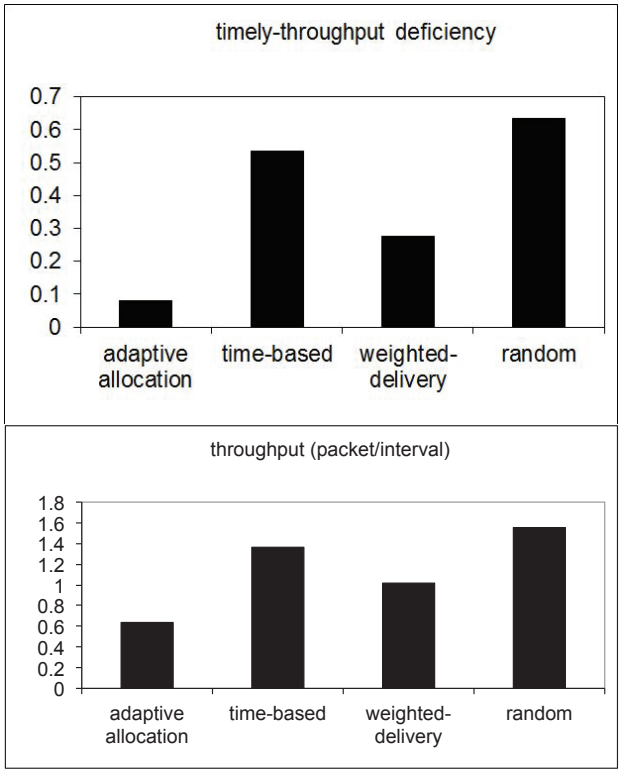

Fig. 7: Performance for VoIP traffic under heterogeneous delay bounds

the length of an interval for MPEG traffic is too small, we only simulate VoIP. There are two groups of clients. All clients generate traffic at rate 64 kbits/sec, and thus each of them has a packet in each interval. Clients in group A require 90% delivery ratio, with delay bounds equal to the interval length. Clients in group B require 50% delivery ratio, with delay bounds equal to two-thirds of the interval length, or 22 time slots. The channel reliability for the n^{th} client in group A is $(84+n)\%$, and that for the n^{th} client in group B is $(29+n)\%$.

Simulation results are shown in Figure 7. The adaptive allocation policy has the smallest total timely-throughput deficiency. This is because the other policies, especially the two largest debt first policies, do not consider heterogeneous delay bounds at all. It is not difficult to see that, to maximize the capacity of the system, a policy should, in some sense, work in an "earliest deadline first" fashion. Without considering heterogeneous delay bounds, the largest debt first policies may unwisely schedule clients with longer delay bounds before those with shorter delay bounds, and thus result in poor channel utilization. On the other hand, such poor channel utilization will result in a large number of idle time slots. Thus, the throughputs for the non-real-time traffic under these policies are higher than those for the adaptive allocation policy.

X. Conclusion

We have analytically studied the problem of scheduling real-time traffic over wireless channels. We have extended the model used in [1], which considers unreliable wireless channels and real-time application requirements, including traffic patterns, delay bounds, and timely-throughput bounds, to a much wider range of channel models. We have developed a general class of polices that are feasibility optimal. This class can serve as a guideline for designing computationally tractable feasibility optimal policies. We have demonstrated the utility of the class by deriving scheduling policies for a general case when rate adaptation is employed and three special cases when it is not, including scenarios where channel qualities are time-varying, the AP does not have instant knowledge on channel qualities, and different clients require different delay bounds. Simulation results show that the policies outperform policies described in [1]. Thus we have shown not only that the policy class is useful in designing scheduling policies, but also that neglecting some realistic and complicated settings can result in unsatisfactory policies.

References

[1] I.-H. Hou and P. Kumar, "Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels," in *Proc. of ACM MobiHoc*, pp. 175–184, 2009.

[2] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. on Information Theory*, vol. 39, no. 2, pp. 89–103, 1993.

[3] M. Neely, "Delay analysis for max weight opportunistic scheduling in wireless systems," in *Proc. of Allerton Conf.*, 2008.
[4] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, 2001.
[5] Q. Liu, X. Wang, and G. Giannakis, "A cross-layer scheduling algorithm with QoS support in wireless networks," *IEEE Trans. on Vehicular Technology*, vol. 55, no. 3, pp. 839–847, 2006.
[6] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE802.11 networks," *IEEE Wireless Communications*, vol. 10, no. 3, pp. 36–43, 2003.
[7] V. Raghunathan, V. Borkar, M. Cao, and P. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *Proc. of IEEE INFOCOM*, pp. 1570–1578, 2008.
[8] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *Wireless Networks*, vol. 8, no. 1, pp. 13–26, 2002.
[9] I.-H. Hou, V. Borkar, and P. Kumar, "A theory of QoS for wireless," in *Proc. of IEEE INFOCOM*, 2009.
[10] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," *IEEE/ACM Trans. Netw.*, vol. 19, pp. 1125–1136, Aug. 2011.
[11] J. Jaramillo, R. Srikant, and L. Ying, "Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 979 –987, may 2011.
[12] P. Soldati, H. Zhang, Z. Zou, and M. Johansson, "Optimal routing and scheduling of deadline-constrained traffic over lossy networks," in *Proc. of IEEE Globecom*, pp. 1 –6, dec. 2010.
[13] R. Li and A. Eryilmaz, "Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multi-hop networks," in *Proc. of IEEE INFOCOM*, pp. 3065 –3073, april 2011.
[14] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, 2002.
[15] Y. Cao and V. Li, "Scheduling algorithms in broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 76–87, 2001.
[16] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, pp. 46–61, Jan. 1973.
[17] P. Bhagwat, P. Bhattacharya, A. Krishma, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," *Wireless Networks*, vol. 3, no. 1, pp. 91–102, 1997.
[18] R. Khalili and K. Salamatian, "Evaluation of packet error rate in wireless networks," in *Proceeding of MSWIM*, vol. 4, 2004.
[19] F. J. Beutler and K. W. Ross, "Optimal policies for controlled markov chains with a constraint," *Journal of Mathematical Analysis and Applications*, vol. 112, no. 1, pp. 236 – 252, 1985.
[20] B. B. H. Korte and J. Vygen, *Combinatorial optimization – Fourth Edition*. Springer.
[21] ITU-T, "G.729 based Embedded Variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729," *ITU-T Recommendations*, 2006.
[22] ITU-T, "Pulse Code Modulation (PCM) of voice frequencies," *ITU-T Recommendations*, 1988.
[23] L. D. la Cruz and J. Mata, "Performance of dynamic resources allocation with QoS guarantees for MPEG VBR video traffic transmission over ATM networks," in *Proc. of GLOBECOM*, pp. 1483–1489, 1999.
[24] I. V. M. F., J. Alins-Delgado, M. Aguilar-Igartua, and J. Mata-Diaz, "Modelling an adaptive-rate video-streaming service using Markov-rewards models," in *Proc. of QSHINE*, pp. 92–99, 2004.
[25] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, 1963.
[26] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, 1960.
[27] H. Wang and N. Moayeri, "Finite-state Markov channel – a useful model for radio communication channels," *IEEE Trans. on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

I-Hong Hou (S10-M12) received the B.S. in Electrical Engineering from National Taiwan University in 2004, and his M.S. and Ph.D. in Computer Science from University of Illinois, Urbana-Champaign in 2008 and 2011, respectively.

In 2012, he joined the department of Electrical and Computer Engineering at the Texas A&M University, where he is currently an assistant professor. His research interests include wireless networks, wireless sensor networks, real-time systems, distributed systems, and vehicular ad hoc networks.

Dr. Hou received the C.W. Gear Outstanding Graduate Student Award from the University of Illinois at Urbana-Champaign, and the Silver Prize in the Asian Pacific Mathematics Olympiad.