Multiuser Scheduling in a Markov-Modeled Downlink Using Randomly Delayed ARQ Feedback

Sugumar Murugesan, *Member, IEEE*, Philip Schniter, *Senior Member, IEEE*, and Ness B. Shroff, *Fellow, IEEE*

***Abstract*—This paper focuses on the downlink of a cellular system and studies opportunistic multiuser scheduling under imperfect channel state information, by exploiting the memory inherent in the channel. The channel between the base station and each user is modeled by a two-state Markov chain and the scheduled user sends back an ARQ feedback that arrives at the scheduler with a random delay, *i.i.d.* across users and time. The scheduler indirectly estimates the channel via accumulated delayed-ARQ feedback and uses this information to make scheduling decisions. The throughput maximization problem is formulated as a partially observable Markov decision process (POMDP). For the case of two users in the system, it is shown that a greedy policy is sum throughput optimal for any distribution on the ARQ feedback delay. For the case of more than two users, the greedy policy is suboptimal and numerical studies demonstrate that it has near optimal performance. Also, the greedy policy can be implemented by a simple algorithm that does not require the statistics of the underlying Markov channel or the ARQ feedback delay, thus making it robust against errors in system parameter estimation. Establishing an equivalence between the two-user system and a genie-aided system, a simple closed form expression for the sum capacity of the downlink is obtained. Further, inner and outer bounds on the capacity region of the downlink are obtained.**

***Index Terms*—ARQ feedback, capacity region, cellular downlink, delay, greedy policy, Markov channel, Markov decision processes, opportunistic multiuser scheduling, sum capacity.**

I. Introduction

With the ever-increasing demand for high data rates, opportunistic multiuser scheduling, introduced by Knopp and Humblet in [1], and defined as *allocating the resources to the user experiencing the most favorable channel conditions*, has gained immense popularity among wireless network designers. Opportunistic multiuser scheduling essentially exploits the multiuser diversity in the system and has motivated several researchers (e.g., [2]–[6]) to study the performance gains obtained by opportunistic scheduling under various scenarios. While the *i.i.d.* flat fading model is used in these works to model time varying channels (for a general treatment on opportunistic scheduling with minimal assumptions on the channel, see [7]), it fails to capture the memory in the channel observed in realistic scenarios. Hence, more recently, opportunistic scheduling has also been investigated by modeling the channels by Markov chains (e.g., [8]–[13]). However, in these works, the channel state information that is crucial for the success of any opportunistic scheduling scheme is assumed to be readily available at the scheduler. This is a simplifying assumption that does not hold in reality, where a non-trivial amount of resource must be spent in gathering the information on the channel state. Another line of work (e.g., [14], [15]) attempts to exploit the memory in the Markov-modeled channels to gather this information. Specifically, Automatic Repeat reQuest (ARQ) feedback, that is traditionally used for error control (e.g., [16]–[19]) at the data link layer, is used to estimate the state of the Markov-modeled channels.

These two lines of work can be combined to create a new design paradigm: exploit multiuser diversity in Markov-modeled channels (e.g., [8]–[13]) and use the already existing ARQ feedback mechanism to estimate the state of these Markov-modeled channels (e.g., [14], [15]). Assuming instantaneous ARQ feedback (i.e., it arrives at the end of the slot) and ON-OFF Markov channel model (the Gilbert-Elliott model [20]), this problem was addressed in independent works [21], [22]. In [21], the authors studied opportunistic spectrum access in a cognitive radio setting—a setup mathematically equivalent to the instantaneous ARQ based opportunistic scheduling in a Markov-modeled downlink—and showed that a simple greedy scheduling policy is optimal. In [22], we directly addressed the instantaneous ARQ based downlink scheduling problem. By identifying a special mathematical structure in the problem, we derived a closed form expression for the two-user sum capacity of the downlink and obtained bounds on the system capacity region.

In this paper, we model the downlink channels by two state (ON-OFF) Markov chains and study the ARQ based joint channel learning-scheduling problem when the ARQ feedback arrives at the scheduler with a *random delay* that is *i.i.d.* across users and time. The delay in the feedback channel is an important consideration that cannot be overlooked in realistic scenarios. The effect of feedback delay on channel resource allocation has been studied under various settings in the past (e.g., [23]–[26]). While these works assume deterministic delay, we consider random, *i.i.d.* feedback delay. An instance when

Manuscript received February 16, 2010; revised September 10, 2011; accepted September 14, 2011. Date of current version February 08, 2012. This work was supported by the NSF CAREER Grant 237037, the Office of Naval Research Grant N00014-07-1-0209, and NSF grants CNS-0721236, CNS-0626703, ARO W911NF-08-1-0238, CNS-1065136, and CNS-1012700.

S. Murugesan was with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA, and is currently with the Department of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: sugumar.murugesan@asu.edu).

P. Schniter is with the Department of Electrical and Computer Engineering, Ohio State University (e-mail: schniter@ece.osu.edu).

N. B. Shroff is with the Department of Electrical and Computer Engineering and the Department of Computer Science and Engineering, Ohio State University (e-mail: shroff@ece.osu.edu).

Communicated by A. Nosratinia, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2173717

the feedback delay can be *i.i.d.* and non-negligible is when the delay is of the order of the scheduling slot length, results from channel propagation time of the feedback signal, and when the feedback channel environment changes drastically due to high mobility of users—a possibility in reality. In essence, by modeling the feedback delay to be random, we attempt to capture the effect of the non-idealities of the feedback channel on the joint channel learning-scheduling problem, in a more general framework.

When compared to the instantaneous ARQ case, the randomly delayed ARQ case adds additional layers of complexity to the scheduling problem, making it different and far more challenging than the former. However, we show that, when there are two users in the system, for any ARQ delay distribution, the greedy policy that was optimal in the instantaneous ARQ case [21] is also optimal in the delayed ARQ case. For more than two users, however, using a counterexample, we show that the greedy policy is not, in general, optimal. Despite the suboptimality, extensive numerical experiments suggest that the greedy policy has near optimal performance. Encouraged by this insight, we study the structure of the greedy policy and show that it can be implemented via a simple algorithm that is immune to errors in the estimates of the Markov channel parameters and the ARQ delay statistics. We also study the fundamental limits of the Markov-modeled downlink with randomly delayed ARQ feedback. By establishing an equivalence between the two-user downlink and a genie-aided system, we derive a simple closed form expression for the sum capacity of the two-user downlink, while obtaining bounds on the sum capacity for larger number of users. We further derive inner and outer bounds on the capacity region of the downlink and tighten these bounds for special cases of the system parameters.

The rest of the paper is organized as follows. The problem setup is described in Section II, followed by a study of the optimality properties of the greedy policy in Section III-A. Section III-B contains a numerical performance analysis of the greedy policy. In Section III-C, we discuss the implementation structure of the greedy policy. We then study the sum capacity and the capacity region of the Markov-modeled downlink in Section IV, followed by concluding remarks in Section V.

II. Problem Setup

A. Channel Model

We consider downlink transmissions with N users. For each user, there is an associated queue at the base station that accumulates packets intended for that user. We assume that each queue is infinitely backlogged. The channel between the base station and each user is modeled by an *i.i.d.* two-state Markov chain. Each state corresponds to the degree of decodability of the data sent through the channel. State 1 (ON) corresponds to full decodability, while state 0 (OFF) corresponds to zero decodability. Note also that the states can be interpreted as a quantized representation of the underlying channel strength, which lies on a continuum. Time is slotted and the channel of each user remains fixed for a slot and moves into another state in the next slot following the state transition probability of the Markov chain. The time slots of all users are synchronized. The two-state Markov channel is characterized by a 2×2 probability transition matrix

$$P = \begin{bmatrix} p & 1-p \\ r & 1-r \end{bmatrix}, \tag{1}$$

where

$$\begin{aligned} p :=& \text{Prob}\{\text{channel is in ON state in the current slot} | \\ & \quad \text{channel was in ON state in the previous slot}\} \\ r :=& \text{Prob}\{\text{channel is in ON state in the current slot} | \\ & \quad \text{channel was in OFF state in the previous slot}\}. \end{aligned}$$

We focus on the case when the crossover from ON to OFF state (respectively, OFF to ON) is less likely to occur than staying in ON state (respectively, OFF state). This is positive correlation, i.e., $p > r$. We restrict our attention to $p > r$ throughout this paper and include discussions on the $p < r$ case, where possible.

B. Scheduling Problem

The base station (henceforth known as the scheduler) is the central controller that controls the transmission to the users in each slot. In any time slot, the scheduler does not know the *exact* channel state of the users and it must schedule the transmission of the head-of-line packet of exactly one user. Thus, a TDMA styled scheduling is performed here. The power spent in each transmission is fixed. At the beginning of a time slot, the head-of-line packet of the scheduled user is transmitted. The scheduled user attempts to decode the received packet and based on the decodability of the packet sends back ACK(bit 1)/NACK(bit 0) feedback signals to the scheduler at the end of the time slot, over an error-free feedback channel. The feedback channel is assumed to suffer from a random delay that is *i.i.d.* across users and time. This delayed feedback information, along with the label of the time slot from which it is acquired, will be used by the scheduler in scheduling decisions. The scheduler aims to maximize the sum of the rate of successful transmission of packets to all the users in the system. We formally define the problem below.

C. Formal Problem Definition

Since the scheduler must make scheduling decisions based only on a partial observation[1] of the underlying Markov chain, the scheduling problem can be represented by a *partially observable Markov decision process (POMDP)*. See [29] for an overview of POMDPs. We now formulate our problem in the language of POMDPs. The key quantities used throughout this paper are summarized in Appendix E.

Horizon: The number of consecutive slots over which scheduling is performed is the horizon. We index the time slots in decreasing order with slot 1 corresponding to the end of the horizon. Throughout this paper, the horizon is denoted by m, i.e., the scheduling process begins at slot m.

Feedback arriving at slot t: For some slot $t, t \leq m$, let $n(t)$ be the number of ARQ feedback bits ($\{0, 1\}$) arriving at the end of slot t from the users scheduled in the previous slots. Due to the random nature of the feedback delay, $n(t)$ can take values in the

[1] In this case, the set of time-stamped binary delayed feedback on the channels.

set $\{0, \ldots, m-t+1\}$. Let F_t represent all the ARQ feedback arriving at the end of slot t. Thus $F_t \in \{0,1\}^{n(t)}$, if $n(t) > 0$ and $F_t = \emptyset$, if $n(t) = 0$. The ARQ feedback is time-stamped and thus, since the scheduler has a record on which users were scheduled in the past slots, it can map the feedback bits F_t to the users and slots they originated from. Let f_k be the feedback that originated during slot k, where $k \leq m$. Note that since in each slot one and only one user is scheduled, f_k is neither empty nor has multiple values, i.e., $f_k \in \{0,1\}$ with bit 0 mapped to NACK and bit 1 to ACK feedback.

Delay of feedback from user i in slot t: Let $D(i,t)$ be the random variable corresponding to the delay, in number of slots, experienced by the feedback sent by user i in slot t. Let $D(i,t) = 0$ correspond to the case when the ARQ feedback originating from user i in slot t arrives at the scheduler at the end of the same slot t. We assume the distribution of $D(i,t)$ to be i.i.d. across users i and time t throughout this work, and let $P_D(d)$, $d \in \{0,1,\ldots\}$ denote the probability mass function of D.

Belief value of user i in slot t – $\pi_t(i)$: This represents the probability that the channel of user $i \in \{1 \ldots N\}$, in slot t, is in the ON state, given all the past feedback about the channel. Define $T^u(.)$, for $u \in \{0,1,\ldots\}$, as the u-step belief evolution operator given by $T^u(x) = T(T^{(u-1)}(x)) = T^{(u-1)}(T(x))$ with $T(x) = xp + (1-x)r$ and $T^0(x) = x$ for $x \in [0,1]$. Now if, at the end of slot $t+1$, the arriving feedback F_{t+1} contains the ARQ feedback from user i from slot $k \in \{m, m-1, \ldots, t+1\}$, i.e., f_k, then, if k is the latest slot from which an ARQ feedback from user i has arrived, then $\pi_t(i)$ is obtained by applying the 1-step belief evolution operator repeatedly over all the time slots between 'now' (slot t) and slot k, i.e.,

$$\pi_t(i) = \begin{cases} T^{k-t}(1) = T^{(k-t-1)}(p), & \text{if } f_k = 1 \\ T^{k-t}(0) = T^{(k-t-1)}(r), & \text{if } f_k = 0, \end{cases} \tag{2}$$

where we have used $T^u(x) = T^{u-1}(T(x))$. If k is not the latest slot from which an ARQ feedback from user i has arrived (possible since the random nature of the feedback delay can result in out-of-turn arrival of ARQ feedback), then due to the first-order Markovian nature of the channels, this ARQ feedback does not have any new information to affect the belief value, and so $\pi_t(i) = T(\pi_{t+1}(i))$. Similarly, if F_{t+1} does not contain any feedback from user i, then $\pi_t(i) = T(\pi_{t+1}(i))$.

Reward structure: In any slot t, a reward of 1 is accrued at the scheduler when the channel of the scheduled user is found to be in the ON state, else 0 is accrued.

Scheduling Policy $\mathfrak{A}_k$: A scheduling policy $\mathfrak{A}_k$ in slot k is a mapping from all the information available at the scheduler in slot k along with the slot index k to a scheduling decision a_k. Formally,

$$\begin{aligned} \mathfrak{A}_k &: ([\pi_m, \pi_{m-1}, \ldots, \pi_k]^k, \{a_m, a_{m-1}, \ldots, a_{k+1}\}) \longrightarrow a_k \\ &\quad \forall k \in [1,m], \pi_k \in [0,1]^N. \end{aligned} \tag{3}$$

where $\{a_m, a_{m-1}, \ldots, a_{k+1}\}$ are the past scheduling decisions and $[\pi_m, \pi_{m-1}, \ldots, \pi_k]^k$ are the belief values of the channels of all users, corresponding to slots $\{m, m-1, \ldots, k\}$, held by the scheduler at the moment (slot k).

Net expected reward in slot t, V_t: With the scheduling policy, $\{\mathfrak{A}_k\}_{k=1}^t$, fixed, the net expected reward in slot t, i.e., V_t, is the sum of the reward expected in the current slot t and the net reward expected in all the future slots $k < t$. Formally, with a_k denoting the scheduling decision in slot k,

$$\begin{aligned} &V_t([\pi_m, \pi_{m-1}, \ldots, \pi_t]^t, \{a_m, a_{m-1}, \ldots, a_{t+1}\}, \{\mathfrak{A}_k\}_{k=1}^t) \\ &\quad = R_t(\pi_t, a_t) + \mathrm{E}\big[V_{t-1}([\pi_m, \pi_{m-1}, \ldots, \pi_t, \pi_{t-1}]^{t-1}, \\ &\qquad\qquad \{a_m, a_{m-1}, \ldots, a_{t+1}, a_t\}, \{\mathfrak{A}_k\}_{k=1}^{t-1})\big], \end{aligned} \tag{4}$$

where $R_t(\pi_t, a_t)$ is the expected immediate reward and the expectation in the future reward is over the feedback received in slot t, i.e., F_t, along with the originating slot indices. Note that the belief vector $[\pi_m, \pi_{m-1}, \ldots, \pi_t]^t$ is up-to-date based on all previous scheduling decisions and the ARQ feedback received before slot t. With the reward structure defined earlier, the expected immediate reward can be written as

$$R_t(\pi_t, a_t) = \pi_t(a_t).$$

Performance Metric: For a given scheduling policy $\{\mathfrak{A}_k\}_{k=1}^m$, the performance metric is given by the sum throughput (sum rate of successful transmission) over a finite horizon, m:

$$\eta_{\text{sum}}(m, \{\mathfrak{A}_k\}_{k=1}^m) = \frac{V_m(\pi_m, \{\mathfrak{A}_k\}_{k=1}^m)}{m}, \tag{5}$$

where π_m is the initial belief values of the channels.

III. Greedy Policy—Optimality, Performance Evaluation and the Implementation Structure

A. On the Optimality of the Greedy Policy

Consider the following policy:

$$\begin{aligned} \widehat{\mathfrak{A}}_k : \pi_k \longrightarrow a_k &= \arg\max_i R_k(\pi_k, a_k = i) \\ &= \arg\max_i \pi_k(i) \; \forall k \geq 1, \pi_k \in [0,1]^N. \end{aligned} \tag{6}$$

Since the above given policy attempts to maximize the expected immediate reward, without any regard to the expected future reward, it follows an approach that is fundamentally *greedy* in nature. We henceforth call $\{\widehat{\mathfrak{A}}_k\}_{k=1}^m$ the greedy policy and let $\hat{a}_k$ denote the scheduling decision in slot k under the greedy policy. We now proceed to establish the optimality of the greedy policy when $N = 2$. We first introduce the following lemma.

Lemma 1: For any $u, v \in \{0,1,2,\ldots\}$ and any $x, y \in [0,1]$ with $x \geq y$,

$$\begin{aligned} T^u(p) &\geq T^{u+1}(x) \\ T^u(r) &\leq T^{u+1}(x) \\ T^u(x) &\geq T^u(y) \\ T^u(p) &\geq T^v(r). \end{aligned} \tag{7}$$

A formal proof of the lemma can be found in Appendix A. The results of Lemma 1 can be explained intuitively. Note that

$T^u(x)$ is the belief value of the channel (probability that the channel is in the ON-state) in the current slot given the belief value, u slots earlier, was x. Also note that $T^u(p)$ (similarly $T^u(r)$) gives the belief value in the current slot given the channel was in the ON state (similarly OFF state) $u+1$ slots earlier. Now, since the Markov channel is positively correlated $(p > r)$, the probability that the channel is in the ON state in the current slot given it was in the ON state $u+1$ slots earlier $(T^u(p))$ is at least as high as the probability that the channel is ON in the current slot given it was *ON with probability* $x \in [0,1]$, $u+1$ slots earlier $(T^{(u+1)}(x))$. This explains the first inequality in Lemma 1. The second and third inequalities can be explained along similar lines. Regarding the last inequality, consider slots t, k such that $t > k$. Due to the Markovian nature of the channel, the closer slot k is to t, the stronger is the memory, i.e., the dependency of the channel state in k with that of t. Now, since the channel is positively correlated, if the channel was in the ON state in slot t, the closer k is to t, the higher is the probability that the channel is ON in slot k. By definition, this probability is given by $T^u(p)$ with $u = t-k-1$. Thus $T^u(p)$ monotonically decreases with u. Using a similar explanation, $T^u(r)$ monotonically *increases* with u. The limiting value of both these functions, as $u \longrightarrow \infty$, is the probability that the channel is ON when no information on the past channel states is available. This is given by the steady state probability[2]. This explains $T^u(p) \geq T^v(r)$ for any $u, v \in \{0,1,\ldots\}$.

Proposition 2: For $N = 2$, the sum throughput, $\eta_{\text{sum}}(m, \{\mathfrak{A}_k\}_{k=1}^m)$, of the system is maximized by the greedy policy $\{\widehat{\mathfrak{A}}_k\}_{k=1}^m$ for any ARQ delay distribution.

Proof: The following is a condensed version of the proof. Detailed proof can be found in [31]. Consider a slot $t < m$. Fix a sequence of scheduling decisions $\vec{a}_{t+1} := \{a_m, a_{m-1}, \ldots, a_{t+1}\}$. Recall the definition of F_{t+1}, the feedback arriving at the end of slot $t+1$, from Section II-C. Let τ_{t+1} denote the originating slots corresponding to feedback F_{t+1}, i.e., if the feedback from users a_u and a_v, for $m \geq u > v \geq t+1$, both arrive at slot $t+1$, then $F_{t+1} = [f_u \; f_v]$ and $\tau_{t+1} = [u \; v]$. Also define $k_1 \in \{\emptyset, m, m-1, \ldots, t+1\}$ as the latest slot from which the ARQ feedback of user 1 is available at the scheduler by (the beginning of) slot t. Formally, if at least one ARQ feedback from user 1 has arrived at the scheduler by slot t, then

$$k_1 = \min_{k \in \{m, m-1, \ldots, t+1\} \text{ s.t } a_k = 1, \; f_k \text{ has arrived by slot } t} k. \quad (8)$$

If no ARQ feedback from user 1 has arrived by slot t, i.e., if $\nexists$ a k such that '$k \in \{m, m-1, \ldots, t+1\}$ s.t $a_k = 1$, f_k has arrived by slot t,' then $k_1 = \emptyset$. Let $l_1 = k_1 - t - 1$, when $k_1 \neq \emptyset$, be a measure of 'freshness' of the latest feedback from user 1. Let $l_1 = \emptyset$ when $k_1 = \emptyset$. Similarly define k_2, l_2 for user 2. With these definitions, the proof proceeds in two steps: In step 1, we show that the greedy decision in slot t, given the ARQ feedback and the scheduling decision from slot $\min(k_1, k_2)$, is independent of the feedback and scheduling decision corresponding to slot $\max(k_1, k_2)$. In step 2, we show that, if the greedy policy is implemented in slot t, then the expected immediate reward in slot t is independent of the scheduling decisions $\vec{a}_{t+1}$. We then provide induction based arguments to establish the proposition.

Step 1: Let $\vec{F}_{t+1} := \{F_m, F_{m-1}, \ldots, F_{t+1}\}$ and $\vec{\tau}_{t+1} := \{\tau_m, \tau_{m-1}, \ldots, \tau_{t+1}\}$. The greedy decision in slot t, conditioned on the past feedback and scheduling decisions is given by

$$\hat{a}_t|_{\vec{F}_{t+1}, \vec{\tau}_{t+1}, \vec{a}_{t+1}, \pi_m} = \hat{a}_t|_{f_{k_1}, f_{k_2}, l_1, l_2, \vec{a}_{t+1}, \pi_m}. \quad (9)$$

The preceding equation comes directly from the first order Markovian property of the underlying channels. Consider the case when $k_1 < k_2 \leq m$ $(\Rightarrow l_1 < l_2)$ or $k_1 = k_2 = \emptyset$ $(\Rightarrow l_1 = l_2 = \emptyset)$. By examining the belief values in the current slot t for all possible values of feedback f_{k_1} and f_{k_2}, along with Lemma 1, the greedy decision can be written as

$$\hat{a}_t|_{f_{k_1}, f_{k_2}, l_1, l_2, \vec{a}_{t+1}, \pi_m} = \begin{cases} 1, & \text{if } f_{k_1} = 1 \\ 2, & \text{if } f_{k_1} = 0 \\ \arg\max_{i \in \{1,2\}}(\pi_m(i)), & \text{if } k_1 = \emptyset, \; k_2 = \emptyset. \end{cases} \quad (10)$$

Thus the greedy decision is independent of feedback f_{k_2} if $k_1 < k_2$. We now proceed to generalize (10). Let k^* denote the latest slot for which an ARQ feedback is available from one of the users by slot t and let $l = k^* - t - 1$ for $k^* \neq \emptyset$ and $l = \emptyset$ for $k^* = \emptyset$ be a measure of freshness of the latest ARQ feedback. Thus, using the preceding discussion, we have

$$\hat{a}_t|_{f_{k_1}, f_{k_2}, l_1, l_2, \vec{a}_{t+1}, \pi_m} = \hat{a}_t|_{f_{k^*}, l, \vec{a}_{t+1}, \pi_m} = \begin{cases} a_{k^*}, & \text{if } k^* \neq \emptyset, f_{k^*} = 1 \\ \bar{a}_{k^*}, & \text{if } k^* \neq \emptyset, \; f_{k^*} = 0 \\ \arg\max_{i \in \{1,2\}}(\pi_m(i)), & \text{if } k^* = \emptyset \end{cases} \quad (11)$$

where $\bar{a}_{k^*}$ is the user not scheduled in slot k^*. This completes step 1 of the proof.

Step 2: If the greedy policy is implemented in slot t, the immediate reward expected in slot t, conditioned on scheduling decisions $\vec{a}_{t+1}$ and initial belief π_m can be rewritten as

$$\begin{aligned} &\mathrm{E}_{\pi_t|\vec{a}_{t+1}, \pi_m} R_t(\pi_t, \hat{a}_t) \\ &\quad = \mathrm{E}_{\pi_t|l=\emptyset, \vec{a}_{t+1}, \pi_m}(R_t(\pi_t, \hat{a}_t)) P(l = \emptyset | \vec{a}_{t+1}, \pi_m) \\ &\qquad + \mathrm{E}_{l, l\neq\emptyset|\vec{a}_{t+1}, \pi_m} \mathrm{E}_{\pi_t|l, l\neq\emptyset, \vec{a}_{t+1}, \pi_m}(R_t(\pi_t, \hat{a}_t)), \end{aligned} \quad (12)$$

where l is defined after (10). Note that

$$\mathrm{E}_{\pi_t|l=\emptyset, \vec{a}_{t+1}, \pi_m}(R_t(\pi_t, \hat{a}_t)) = \max_i T^{(m-t)}(\pi_m(i)) \quad (13)$$

since, with $l = \emptyset$, i.e., no past feedback at the scheduler, the belief values at slot t is independent of the past scheduling decisions and is simply given by $\pi_t = T^{(m-t)}(\pi_m)$. Now rewriting the second part of (12),

$$\begin{aligned} &\mathrm{E}_{l, l\neq\emptyset|\vec{a}_{t+1}, \pi_m} \mathrm{E}_{\pi_t|l, l\neq\emptyset, \vec{a}_{t+1}, \pi_m}(R_t(\pi_t, \hat{a}_t)) \\ &\quad = \mathrm{E}_{l, l\neq\emptyset|\vec{a}_{t+1}, \pi_m} \mathrm{E}_{\pi_{l+t+1}|l, l\neq\emptyset, \vec{a}_{t+1}, \pi_m} \\ &\qquad \mathrm{E}_{\pi_t|\pi_{l+t+1}, l, l\neq\emptyset, \vec{a}_{t+1}, \pi_m}(R_t(\pi_t, \hat{a}_t)). \end{aligned} \quad (14)$$

[2] We will discuss the steady state probability in Section IV.

Consider $\mathrm{E}_{\pi_t|\pi_{l+t+1},l,l\neq\emptyset,\vec{a}_{t+1},\pi_m}(R_t(\pi_t,\hat{a}_t))$. From the first step of the proof, the greedy decision in slot t can be made solely based on the latest feedback, i.e., $f_{k^*=l+t+1}$. This was recorded in (11). Thus, averaging over $f_{k^*=l+t+1}$, we have

$$\begin{aligned}
&\mathrm{E}_{\pi_t|\pi_{l+t+1},l,l\neq\emptyset,\vec{a}_{t+1},\pi_m}(R_t(\pi_t,\hat{a}_t))\\
&\quad=\pi_{l+t+1}(a_{l+t+1})T^l(p)+(1-\pi_{l+t+1}(a_{l+t+1}))\times\\
&\qquad\big(\pi_{l+t+1}(\bar{a}_{l+t+1})T^l(p)+(1-\pi_{l+t+1}(\bar{a}_{l+t+1}))T^l(r)\big)\\
&\quad=P\big(\{S_{l+t+1}(1)=1\cup S_{l+t+1}(2)=1\}|\\
&\qquad\qquad\pi_{l+t+1},l,l\neq\emptyset,\vec{a}_{t+1},\pi_m\big)T^l(p)\\
&\qquad+P\big(\{S_{l+t+1}(1)=0\cap S_{l+t+1}(2)=0\}|\\
&\qquad\qquad\pi_{l+t+1},l,l\neq\emptyset,\vec{a}_{t+1},\pi_m\big)T^l(r)
\end{aligned}\tag{15}$$

where $S_k(i)$ is the $1/0$ state of the channel of user i in slot k. We can now simplify (14) as

$$\begin{aligned}
&\mathrm{E}_{l,l\neq\emptyset|\vec{a}_{t+1},\pi_m}\mathrm{E}_{\pi_t|l,l\neq\emptyset,\vec{a}_{t+1},\pi_m}(R_t(\pi_t,\hat{a}_t))\\
&\quad=\mathrm{E}_{l,l\neq\emptyset|\vec{a}_{t+1},\pi_m}\\
&\qquad\Big(P\big(\{S_{l+t+1}(1)=1\cup S_{l+t+1}(2)=1\}|\pi_m\big)T^l(p)\\
&\qquad+P\big(\{S_{l+t+1}(1)=0\cap S_{l+t+1}(2)=0\}|\pi_m\big)T^l(r)\Big).
\end{aligned}\tag{16}$$

We have used the following argument in the last equality: the event $(\{S_{l+t+1}(1)=1\cup S_{l+t+1}(2)=1\})$ is controlled by the underlying Markov dynamics and is independent of the scheduling decisions $\vec{a}_{t+1}$. Likewise, this event is independent of the value of l since we have assumed that the feedback channel and the forward channel are independent.

By explicitly including the probability mass function of the quantity l as a function of the ARQ delay statistics, we can rewrite (12) as

$$\begin{aligned}
&\mathrm{E}_{\pi_t|\vec{a}_{t+1},\pi_m}R_t(\pi_t,\hat{a}_t)\\
&\quad=\prod_{k=m}^{t+1}P(D(a_k,k)>k-t-1)\max_i T^{(m-t)}(\pi_m(i))\\
&\quad+\sum_{l=0}^{m-t-1}P(D(1,l+t+1)\leq l)\prod_{k=t+l}^{t+1}P(D(1,k)>k-t-1)\\
&\qquad\Big(P\big(\{S_{l+t+1}(1)=1\cup S_{l+t+1}(2)=1\}|\pi_m\big)T^l(p)\\
&\qquad+P\big(\{S_{l+t+1}(1)=0\cap S_{l+t+1}(2)=0\}|\pi_m\big)T^l(r)\Big).
\end{aligned}\tag{17}$$

The expected reward in slot t is thus independent of the sequence of actions $\{a_m,a_{m-1}\ldots a_{t+1}\}$ if the greedy policy is implemented in slot t. By extension, the total reward expected from slot t until the horizon is independent of the scheduling vector $\vec{a}_{t+1}$ if the greedy policy is implemented in slots $\{t,t-1,\ldots,1\}$, i.e.,

$$\sum_{k=t}^{1}\mathrm{E}_{\pi_k|\vec{a}_{t+1},\pi_m}R_k(\pi_k,\hat{a}_k)=\sum_{k=t}^{1}\mathrm{E}_{\pi_k|\pi_m}R_k(\pi_k,\hat{a}_k).\tag{18}$$

Thus, if the greedy policy is optimal in slots $\{t,t-1,\ldots,1\}$, then, it is also optimal in slot $t+1$. Since t is arbitrary and since the greedy policy is optimal at the horizon, by induction, the greedy policy is optimal in every slot $\{m,m-1,\ldots,1\}$. This establishes the proposition. ■

Remarks: When the Markov channels are negatively correlated, i.e., $p<r$, using arguments similar to those in the preceding proof, we have shown [31] that the greedy policy is optimal when $N=2$, for any ARQ delay distribution. Returning to the original positive correlation setup, the arguments in the proof of Proposition 2 hold true even when the ARQ delay is not identically distributed across time. Thus, the greedy policy is optimal for $N=2$ even when the ARQ delay distribution is time-variant. Also, since m is arbitrary, the greedy policy maximizes the sum throughput over an infinite horizon. We record this below.

Corollary 3: For $N=2$, the greedy policy is optimal when the performance metric is the sum throughput over an infinite horizon, i.e.,

$$\{\hat{\mathfrak{A}}_k\}_{k\geq1}=\arg\max_{\{\mathfrak{A}_k\}_{k\geq1}}\lim_{m\longrightarrow\infty}\frac{V_m(\pi,\{\mathfrak{A}_k\}_{k\geq1})}{m}\tag{19}$$

for any initial belief π.

The optimality of the greedy policy does not extend to the case $N>2$. We record this in the following proposition.

Proposition 4: The greedy policy is not, in general, optimal when there are more than two users in the downlink.

Proof Outline: We establish the proposition using a counterexample with deterministic ARQ delay of $D=1$, i.e., $P_D(d=1)=1$, and arbitrary values of $N;N>2$ and $m;m>3$. We construct a variant of the greedy policy that schedules a non-greedy user in a specific time slot under a specific sample path of the past channel states observable by the scheduler. In the rest of the slots and under other realizations, the constructed policy performs greedy scheduling. We explicitly evaluate the difference in the rewards corresponding to the constructed policy and the greedy policy and show that, there exists system parameters such that the constructed policy has a reward strictly larger than the greedy policy. Thus the greedy policy is, in general, not optimal when $N>2$. A formal proof can be found in Appendix B.

Remarks: Note that, in contrast, it has been shown in [21] that the greedy policy is optimal for any number of users when the ARQ feedback is *instantaneous*, i.e., $D=0$. To summarize, the optimality of the greedy policy vanishes

- when the ARQ delay is increased from zero to higher values, with the number of users unconstrained, or
- when the number of users is increased from two to higher values, with the ARQ delay being random and unconstrained.

These observations point to the volatile nature of the underlying dynamics of the scheduling problem, with respect to the greedy policy optimality.

It would be interesting to see how the optimality properties of the greedy policy extend to more general channel models. Considering the multi-rate channels, i.e., when the number

TABLE I
COMPARISON OF THE PERFORMANCE OF THE GREEDY POLICY WITH THE OPTIMAL REWARD

N	Delay $= [P_D(0) \ldots P_D(d_{\max})]$	p	r	V_{opt}	V_{greedy}	%subopt
3	[0.8822 0.1178]	0.9172	0.2858	6.0707	6.0696	0.0182 %
4	[0.5387 0.4613]	0.9464	0.1666	5.9700	5.9586	0.1910 %
3	[0.5908 0.3959 0.0132]	0.6619	0.2389	3.9933	3.9914	0.0476 %
4	[0.6647 0.1844 0.1510]	0.9281	0.2824	5.8934	5.8854	0.1364 %

TABLE II
COMPARISON OF THE PERFORMANCE OF THE GREEDY POLICY WITH THE OPTIMAL REWARD IN THE GENIE-AIDED SYSTEM. MAXIMUM ARQ DELAY,$d_{\max} = 1$

	Delay	$p = 0.5848,\ r = 0.3509$			$p = 0.6392,\ r = 0.2328$		
		V_{genie}	V_{greedy}	%genie	V_{genie}	V_{greedy}	%genie
N=10	[0 1]	5.3908	5.2912	1.8470 %	5.5279	5.2067	5.8109 %
	$[\frac{1}{3}\ \frac{2}{3}]$	5.6547	5.4281	4.0072 %	5.9195	5.4119	8.5741 %
	$[\frac{1}{2}\ \frac{1}{2}]$	5.7867	5.4987	4.9771 %	6.1152	5.5208	9.7203 %
	$[\frac{2}{3}\ \frac{1}{3}]$	5.9187	5.5712	5.8703 %	6.3110	5.6353	10.7070 %
	Delay	$p = 0.9148,\ r = 0.4309$			$p = 0.3079,\ r = 0.2517$		
		V_{genie}	V_{greedy}	%genie	V_{genie}	V_{greedy}	%genie
N=20	[0 1]	8.8565	8.8254	0.3504 %	3.4487	3.4371	0.3368 %
	$[\frac{1}{3}\ \frac{2}{3}]$	8.9715	8.9291	0.4723 %	3.5525	3.4661	2.4315 %
	$[\frac{1}{2}\ \frac{1}{2}]$	9.0290	8.9820	0.5203 %	3.6043	3.4807	3.4300 %
	$[\frac{2}{3}\ \frac{1}{3}]$	9.0865	9.0357	0.5593 %	3.6562	3.4955	4.3967 %

of states is greater than two, the special ‘toggle’ structure that led to the optimality of the greedy policy in the ON-OFF channel vanishes. In fact, we have shown [30] that, even when the number of states is increased by 1, the general greedy policy optimality vanishes and the optimality can be shown to hold only under very restrictive conditions on the Markov channel statistics. Now, consider the case when the two-state Markov channels are non-identical across users. In this setup, using a counterexample, we have shown [31] that the greedy policy is not, in general, optimal, even when $N = 2$ and the ARQ delay is instantaneous. In summary, the optimality of the greedy policy vanishes even under minimal deviations from the original setup. These observations further indicate the volatile nature of the underlying scheduling problem dynamics.

Returning to the original setup at hand, numerical results suggest that the greedy policy, despite being not optimal in general, has near optimal performance. We discuss this next.

B. Performance Evaluation of the Greedy Policy

Table I provides a sample of the net expected reward under the greedy policy (V_{greedy}) in comparison with that of the optimal policy (V_{opt}) when $N = 3$, and when $N = 4$, for horizon length $m = 7$. The ARQ delay probability mass function is generated (uniform) randomly with the maximum delay $d_{\max}$ fixed first. The low values of the quantity $\%\text{subopt} = \frac{V_{\text{opt}} - V_{\text{greedy}}}{V_{\text{opt}}} \times 100\%$ illustrates the near optimal performance of the greedy policy for the system parameters considered. Note that, the optimal reward, V_{opt}, is evaluated by a brute-force search over the scheduling decisions in every slot $t \in \{m, m-1, \ldots, 1\}$, that is prohibitively complex for larger values of N and m. We, therefore, perform an indirect study of the greedy policy performance in Tables II–IV, that allows us to consider wider range of system parameters. We first define the genie-aided system as follows: for any slot k, the feedback f_k includes the channel state information, corresponding to slot k, of not only the scheduled user a_k but also that of all the users in the system. Thus the optimal reward in the genie-aided system, V_{genie}, is an upper bound to the optimal reward in the original system, V_{opt}. Also, V_{genie} can be evaluated using closed-form expressions, with complexity much lower than that of V_{opt}. We will discuss the evaluation of V_{genie} in the context of the genie-aided system sum capacity in Section IV-A.

In Table II, with the maximum ARQ delay $d_{\max} = 1$, the net expected reward under the greedy policy is compared with V_{genie} when $N = 10$ and when $N = 20$, for randomly generated values of p and r. The length of the horizon is fixed at $m = 10$. The probability mass function of the ARQ delay, denoted by ‘Delay’ in the table, is controlled to have a weakening ‘tail’ from [0 1] to $[\frac{2}{3}\ \frac{1}{3}]$. The quantity $\%\text{genie} = \frac{V_{\text{genie}} - V_{\text{greedy}}}{V_{\text{genie}}} \times 100\%$ is an upper bound to the quantity %subopt introduced earlier. Table III is similarly constructed with the maximum ARQ delay $d_{\max} = 2$. In both Tables II and III, we see that $\%\text{genie}$ is predominantly low-valued, suggesting that the greedy policy has near optimal performance. Also, note that, as the tail of the ARQ

TABLE III
COMPARISON OF THE PERFORMANCE OF THE GREEDY POLICY WITH THE OPTIMAL REWARD IN THE GENIE-AIDED SYSTEM. MAXIMUM ARQ DELAY, $d_{\max} = 2$

	Delay	$p = 0.2148,\ r = 0.1100$			$p = 0.6863,\ r = 0.4136$		
		V_{genie}	V_{greedy}	%genie	V_{genie}	V_{greedy}	%genie
N=10	[0 0 1]	2.0196	2.0162	0.1716 %	6.2768	6.2571	0.3131 %
	$[\frac{1}{6}\ \frac{1}{3}\ \frac{1}{2}]$	2.1261	2.0384	4.1241 %	6.4895	6.3813	1.6663 %
	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$	2.2152	2.0577	7.1089 %	6.6375	6.4743	2.4587 %
	$[\frac{1}{2}\ \frac{1}{3}\ \frac{1}{6}]$	2.3018	2.0772	9.7568 %	6.7764	6.5677	3.0792 %
	Delay	$p = 0.8822,\ r = 0.2816$			$p = 0.7120,\ r = 0.5713$		
		V_{genie}	V_{greedy}	%genie	V_{genie}	V_{greedy}	%genie
N=20	[0 0 1]	8.0485	7.9811	0.8376 %	7.0084	7.0066	0.0251 %
	$[\frac{1}{6}\ \frac{1}{3}\ \frac{1}{2}]$	8.3208	8.1880	1.5952 %	7.0868	7.0585	0.3989 %
	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$	8.4754	8.3186	1.8493 %	7.1495	7.1017	0.6675 %
	$[\frac{1}{2}\ \frac{1}{3}\ \frac{1}{6}]$	8.6131	8.4490	1.9057 %	7.2099	7.1448	0.9018 %

TABLE IV
ILLUSTRATION OF THE EFFECT OF THE MARKOV CHANNEL MEMORY, $(p - r)$ ON THE REWARD FUNCTIONS. MAXIMUM ARQ DELAY, $d_{\max} = 2$

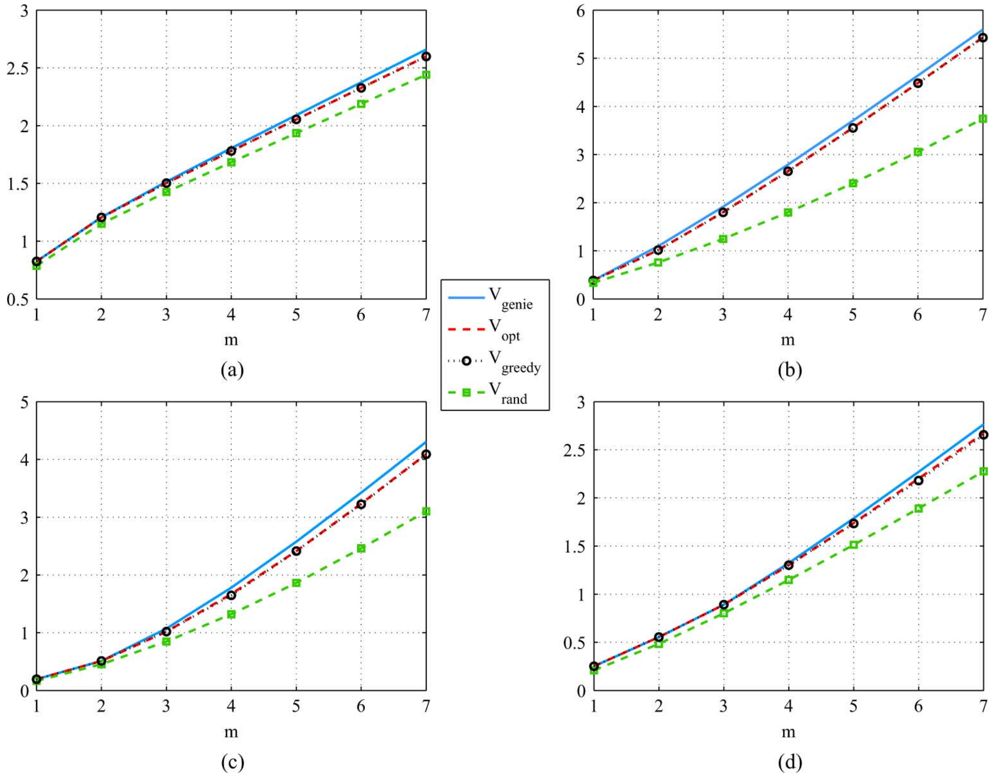
$(p-r)$	Delay	V_{genie}	V_{greedy}	%genie	$(p-r)$	Delay	V_{genie}	V_{greedy}	%genie
0.2	[0 0 1]	5.6342	5.6232	0.1953 %	0.8	[0 0 1]	7.9848	7.7252	3.2520 %
	$[\frac{1}{6}\ \frac{1}{3}\ \frac{1}{2}]$	5.8068	5.7105	1.6592 %		$[\frac{1}{6}\ \frac{1}{3}\ \frac{1}{2}]$	8.3585	8.0181	4.0726 %
	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$	5.9357	5.7797	2.6283 %		$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$	8.5551	8.1843	4.3347 %
	$[\frac{1}{2}\ \frac{1}{3}\ \frac{1}{6}]$	6.0584	5.8494	3.4499 %		$[\frac{1}{2}\ \frac{1}{3}\ \frac{1}{6}]$	8.7265	8.3522	4.2890 %

delay mass function weakens, both V_{genie} and V_{greedy} increase. This is expected since, with a weakening tail, the ARQ feedback is stochastically more ‘fresh,’ thereby facilitating better informed scheduling decisions and higher rewards in both genie-aided and original systems. Also note that, as the tail weakens, the gap between the *optimal* rewards in the genie-aided system and the original system can be expected to increase, since the gap between the information content of the full feedback (genie-aided system) and the ARQ feedback increases with a weakening tail. Thus, the relatively high values of $\%\text{genie}$ corresponding to weaker delay tails, could be due to an inherent system level gap between the genie-aided and the original systems, and need not necessarily be a pointer to the greedy policy performance. The last statement is further strengthened by the fact that the greedy policy is optimal when the ARQ delay tail is at the weakest, i.e., when the feedback is instantaneous [21].

In Table IV, we study the effect of the Markov channel memory, defined as $(p - r)$, on the reward functions. With $d_{\max} = 2$, $m = 10$ and $N = 20$, we consider two extreme values of the channel memory, i.e., $(p - r) = 0.2$ and $(p - r) = 0.8$. In both cases of channel memory, we have fixed the steady state probability of the ON state to be $\pi_{ss} = 0.5$, by fixing $p + r = 1$. This essentially provides a degree of fairness when comparing these two cases. Note that, for a fixed delay statistic, the rewards V_{genie} and V_{greedy} increase with increase in the channel memory. This is due to an increase in the value of the feedback, as the channel memory increases. Also, we see an increase in the value of $\%\text{genie}$ as the memory increases. This points to two underlying phenomena: 1) An increase in the inherent sub-optimality associated with greedy scheduling as the channel memory increases 2) Similar to the case of weakening delay tail, an increase in the channel memory results in an increase in the system level gap between the genie-aided and the original systems, by way of an increase in the gap between the information content of full feedback (genie-aided system) and the ARQ feedback.

In Fig. 1, we compare the performance of the greedy policy with that of a random policy (V_{rand}) that ignores the ARQ feedback and schedules purely based on initial beliefs. For randomly generated system parameters, V_{opt} and V_{greedy} are plotted alongside V_{genie} and V_{rand}, for increasing horizon lengths. From the figure, we observe that the greedy policy achieves a performance comparable to that of the optimal policy in both the original and the genie-aided systems, while V_{rand} is significantly lower than V_{greedy}. Also, understandably, the performance gap between the greedy and the random policies appears to be large for large system memory, i.e., $(p - r)$, the case with Figs. 1(b) and 1(c).

Summarizing, Tables I–IV suggest that the greedy policy has near optimal performance for a wide range of system parameters and that the ARQ delay profile and the channel memory affect the reward values in ways that can be explained intuitively. In addition, note that $\%\text{genie}$ is also an upper bound to the quantity $\frac{V_{\text{genie}} - V_{\text{opt}}}{V_{\text{genie}}} \times 100\%$. Thus the low values of $\%\text{genie}$ provide the following larger message: using only the 1-bit ARQ feedback for opportunistic scheduling is associated with system level per-



(a) (b)

(c) (d)

Fig. 1. Net expected reward under the greedy policy in comparison with system-level performance limits. System parameters used: plot (a) $N = 3, p = 0.4070$, $r = 0.1999, P_D(0) = 0.3379, P_D(1) = 0.5666, P_D(2) = 0.0954, \pi_m = [0.7487\ 0.8256\ 0.7900]$ (b) $N = 3, p = 0.9930, r = 0.1267, P_D(0) = 0.8855$, $P_D(1) = 0.1145, \pi_m = [0.3631\ 0.2662\ 0.3857]$ (c) $N = 3, p = 0.9694, r = 0.1556, P_D(0) = 0, P_D(1) = 1, \pi_m = [0.1207\ 0.1962\ 0.1791]$ (d) $N = 3$, $p = 0.7965, r = 0.1365, P_D(0) = 0, P_D(1) = 0, P_D(2) = 1, \pi_m = [0.1351\ 0.2523\ 0.2410]$.

formance comparable to the case when feedback is available from *all* the users.

C. Structure of the Greedy Policy

Motivated by the near optimal performance of the greedy policy, we proceed to study its structure, which turns out to be very amenable for practical implementation. We begin by defining the following quantity:

Schedule order vector, O_t, in slot t: The user indices in decreasing order of $\pi_t(i)$, i.e.,

$$\begin{aligned} O_t(1) &= \arg\max_i \pi_t(i) \\ &\vdots \\ O_t(N) &= \arg\min_i \pi_t(i). \end{aligned}$$

Thus, the greedy decision in slot t is $\hat{a}_t = O_t(1)$.

Now, in any slot $t \leq m$, any user i falls under one of the following two cases:

1) The scheduler has received at least one ARQ feedback from user i by the beginning of slot t. Let k_i, for $m \geq k_i > t$, be the latest slot for which the ARQ feedback from user i is available at the scheduler. Since the channel is first-order Markovian, the belief value of the channel of user i in the current slot t is dependent only on the feedback f_{k_i} and k_i. The belief value is given by

$$\pi_t(i) = \begin{cases} T^{k_i-t-1}(p) & \text{if } f_{k_i} = 1 \\ T^{k_i-t-1}(r) & \text{if } f_{k_i} = 0. \end{cases} \tag{20}$$

2) The scheduler does not have any ARQ feedback from user i by the beginning of slot t. In this case

$$\pi_t(i) = T^{(m-t)}\pi_m(i). \tag{21}$$

Recall that $\pi_m(i)$ is the initial belief value of the channel of user i when the scheduling process started at slot m.

At slot t, let $\mathcal{A}_t$ denote the set of users, i, whose latest feedback, f_{k_i}, is an ACK. Let $\mathcal{N}_t$ denote the set of users, j, whose latest feedback, f_{k_j}, is a NACK. Let the users from whom the scheduler has not yet received any feedback constitute set $\mathcal{X}_t$. From (20) and (21), using Lemma 1, the greedy decision in slot t can be written as

$$\hat{a}_t = \begin{cases} \arg\min_{i\in\mathcal{A}_t} k_i & \text{if } \mathcal{A}_t \neq \emptyset \\ \arg\max_{i\in\mathcal{X}_t} \pi_m(i) & \text{if } \mathcal{A}_t = \emptyset \text{ and } \mathcal{X}_t \neq \emptyset \\ \arg\max_{i\in\mathcal{N}_t} k_i & \text{if } \mathcal{A}_t = \emptyset \text{ and } \mathcal{X}_t = \emptyset. \end{cases} \tag{22}$$

Now, for ease of implementation, we visualize the sets $\mathcal{A}_t$, $\mathcal{X}_t$ and $\mathcal{N}_t$ as queues with elements ordered in the following specific ways: Let $\mathcal{A}_t(i)$ denote the i^{th} element of queue $\mathcal{A}_t$ and the elements be ordered such that $k_{\mathcal{A}_t(1)} < k_{\mathcal{A}_t(2)} \cdots < k_{\mathcal{A}_t(n(\mathcal{A}_t))}$, where $n(A)$ denotes the cardinality of set A. Note that the user that gave an ACK from the most recent slot lies at the head of queue $\mathcal{A}_t$. The elements of $\mathcal{X}_t$ are ordered such that $\pi_m(\mathcal{X}_t(1)) \geq \pi_m(\mathcal{X}_t(2)) \cdots \geq \pi_m(\mathcal{X}_t(n(\mathcal{X}_t)))$. The elements of $\mathcal{N}_t$ satisfy $k_{\mathcal{N}_t(1)} > k_{\mathcal{N}_t(2)} \cdots > k_{\mathcal{N}_t(n(\mathcal{N}_t))}$, i.e., the user with the oldest NACK feedback lies on top of queue $\mathcal{N}_t$. Define a combined queue constructed by concatenating the queues $\mathcal{A}_t$, $\mathcal{X}_t$ and $\mathcal{N}_t$ in that order. From (20) and (21), using Lemma 1,

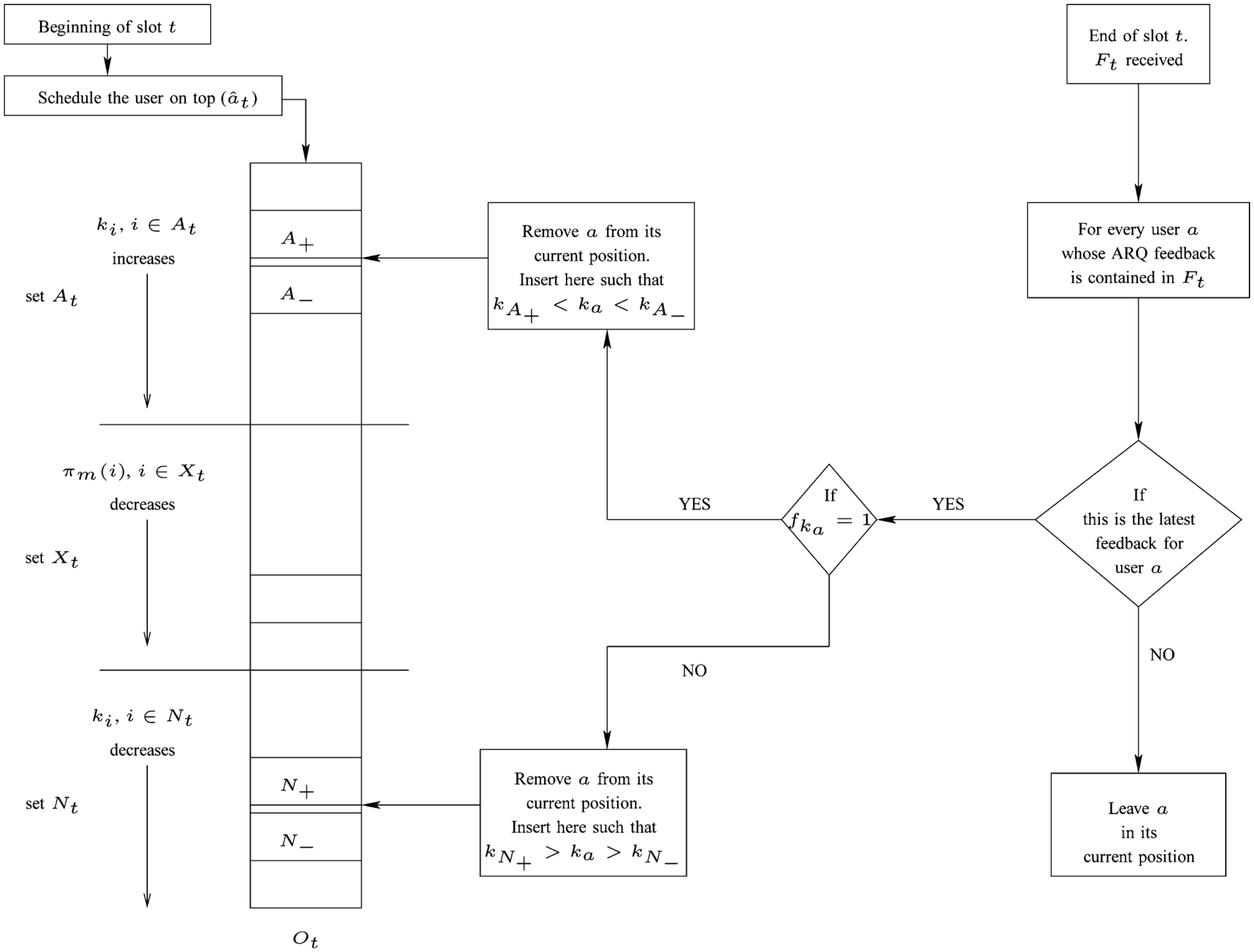

Beginning of slot t
Schedule the user on top ($\hat{a}_t$)
k_i, $i \in A_t$
increases
set A_t
A_+
A_-
$\pi_m(i)$, $i \in X_t$
decreases
set X_t
k_i, $i \in N_t$
decreases
set N_t
N_+
N_-
O_t
Remove a from its current position. Insert here such that $k_{A_+} < k_a < k_{A_-}$
YES
If $f_{k_a} = 1$
YES
NO
Remove a from its current position. Insert here such that $k_{N_+} > k_a > k_{N_-}$
End of slot t. F_t received
For every user a whose ARQ feedback is contained in F_t
If this is the latest feedback for user a
NO
Leave a in its current position

Fig. 2. Greedy policy implementation under random ARQ delay.

we see that the users in the combined queue are arranged in decreasing order (top-down) of belief values with the top-most user being the greedy decision in slot t. Thus the combined queue is, in fact, the schedule order vector O_t.

We now discuss the evolution of the schedule order vector. For every user a whose ARQ feedback is contained in F_t, implement the following procedure: Let t_a indicate the originating slot for the ARQ feedback from user a contained in F_t. Now, if t_a is the latest slot from which the ARQ feedback of user a is available at the scheduler, then $k_a = t_a$. The new schedule order vector O_{t-1} is formed by removing user a from its current position (in O_t) and placing it in the sub-queue $\mathcal{A}_{t-1}$ (if $f_{k_a} = 1$) or in the sub-queue $\mathcal{N}_{t-1}$ (if $f_{k_a} = 0$) at an appropriate location (so that the ordering based on k_i is not violated). If $t_a \neq k_a$, i.e., t_a is not the latest slot, then user a is not moved. Similarly, users whose ARQ feedback are not contained in F_t are not moved. The last two statements are direct consequences of the following facts:

- For an user a whose ARQ feedback is contained in F_t but is not the latest feedback from that user, the belief value evolves as $\pi_{t-1}(a) = T(\pi_t(a))$. Similarly, for an user b whose ARQ feedback is *not* contained in F_t, the belief value evolves as $\pi_{t-1}(b) = T(\pi_t(b))$. Both these cases were discussed in Section II-C.
- From Lemma 1, if $x \geq y$, then $T(x) \geq T(y)$.

Now, at slot $t-1$, the user on top of O_{t-1} is the greedy decision. Thus the greedy decision in any slot is determined by the latest ARQ feedback and the corresponding originating slot index of all the users in the system. Note that this implementation does not require the Markov channel statistics (other than the knowledge that $p > r$) and the statistics of the ARQ feedback delay. An illustration of the greedy policy implementation is provided in Fig. 2. For the special case of deterministic ARQ feedback delay $D = d$, the evolution from O_t to O_{t-1} is greatly simplified [31] leading to a simpler implementation structure. This is illustrated in Fig. 3.

IV. On Downlink Sum Capacity and Capacity Region

We now proceed to study the fundamental limits on the downlink system performance—the sum capacity and the capacity region.

A. Sum Capacity of the Downlink

The sum capacity of the downlink is defined as the maximum sum throughput over an infinite horizon with steady state initial conditions. Formally, with N users in the system,

$$C_{\text{sum}}(N) = \max_{\{\mathfrak{A}_k\}_{k\geq 1}} \lim_{m \longrightarrow \infty} \frac{V_m(\pi_{ss}, \{\mathfrak{A}_k\}_{k\geq 1})}{m}, \qquad (23)$$

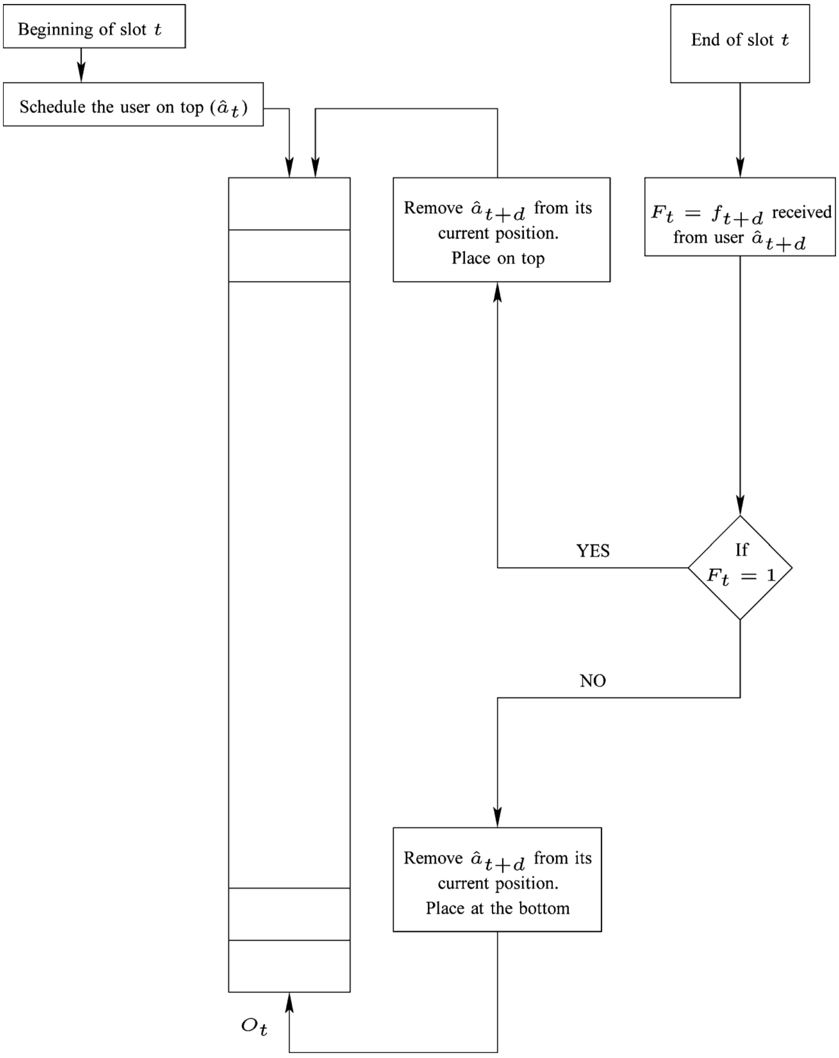

Beginning of slot t
Schedule the user on top ($\hat{a}_t$)
End of slot t
Remove $\hat{a}_{t+d}$ from its current position. Place on top
$F_t = f_{t+d}$ received from user $\hat{a}_{t+d}$
YES
If $F_t = 1$
NO
Remove $\hat{a}_{t+d}$ from its current position. Place at the bottom
O_t

Fig. 3. Greedy policy implementation under deterministically delayed ARQ, i.e., $D = d$.

where $\forall i \in \{1, \ldots, N\}$, $\pi_{ss}(i) = p_s$, the steady state probability of the Markov channel. We now proceed to derive p_s. The Markov chain transition matrix $P = \begin{bmatrix} p & 1-p \\ r & 1-r \end{bmatrix}$ can be expressed as $P = U\Lambda V$, where

$$U = \begin{bmatrix} 1 & 1 \\ 1 & \frac{-r}{1-p} \end{bmatrix}$$
$$\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & p-r \end{bmatrix}$$
$$V = \frac{1}{1+\frac{r}{1-p}} \begin{bmatrix} \frac{r}{1-p} & 1 \\ 1 & -1 \end{bmatrix},$$

with $VU = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Assuming[3] $p + (1-r) < 2$,

$$\lim_{n \longrightarrow \infty} P^n = \begin{bmatrix} \frac{r}{1-(p-r)} & 1 - \frac{r}{1-(p-r)} \\ \frac{r}{1-(p-r)} & 1 - \frac{r}{1-(p-r)} \end{bmatrix}$$
$$\Rightarrow p_s = \frac{r}{1-(p-r)}.$$

Recall, from Section III-B, the definition of the genie-aided system: In any slot k, the feedback f_k contains the channel state information, corresponding to slot k, of not only the scheduled user but also that of all the users in the system. Also, the delay profile from the original system is retained in the genie-aided system, i.e., the *cumulative* feedback f_k arrive at the scheduler with delay $D(a_k, k)$ that is *i.i.d.* across scheduling choice a_k and originating slot k with the probability mass function $P_D(d)$. Thus, thanks to the cumulative nature of the feedback, the scheduling decision in the current slot does not affect the information available for scheduling in future slots. Hence, the greedy policy is optimal in the genie-aided system. With this insight, we now report our result on the sum capacity of the original downlink with two users.

Proposition 5: When $N = 2$, the sum capacity of the Markov-modeled downlink with randomly delayed ARQ equals that of the genie-aided system. This sum capacity equals

$$C_{\text{sum}}(N = 2) = \sum_{l=0}^{\infty} \Big[p_s T^l(p) + (1-p_s)p_s \Big] P(D \leq l) \prod_{d=0}^{l-1} P(D > d). \tag{24}$$

[3] $p + (1-r) = 2$ leads to $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, a trivial case with no steady state.

Furthermore, the greedy policy achieves this sum capacity.

Proof: We first focus on the sum capacity of the genie-aided system, i.e., the sum throughput of the greedy policy in the genie-aided system. Recall, from Section III-A, the quantity L—the measure of freshness of the latest ARQ feedback. We defined L such that $L = l \Rightarrow$ the latest feedback is $l+1$ slots old. We extend the meaning of L to the genie-aided system. Due to the first order Markovian nature of the channels, in the genie-aided system, conditioned on the latest feedback, f_{t+l+1} (with t denoting the current slot), the belief values (and hence the greedy scheduling decision) in the current slot are independent of the feedback from previous slots, i.e., $f_{k,k>t+l+1}$. Thus, with $R_{\text{genie}}^{\text{greedy}}(l,N)$ denoting the conditional (conditioned on $L=l$) immediate reward corresponding to the greedy policy, in the N-user genie-aided system with steady state initial conditions, the sum capacity of the genie-aided system can be written as

$$C_{\text{sum}}^{\text{genie}}(N) = \mathrm{E}_l\left[R_{\text{genie}}^{\text{greedy}}(l,N)\right]. \tag{25}$$

We now evaluate $R_{\text{genie}}^{\text{greedy}}(l,N)$. From Lemma 1, the belief value (in the current slot) of an user with an ON channel $l+1$ slots earlier, i.e., $T^l(p)$, is higher than the belief value of an user with an OFF channel $l+1$ slots earlier, i.e., $T^l(r)$. Thus, in steady state,

$$\begin{aligned} R_{\text{genie}}^{\text{greedy}}(l,N) &\\ = P(&\text{at least one of the } N \text{ users has an ON channel in} \\ &\qquad\qquad \text{steady state})T^l(p) \\ &+ P(\text{all users have OFF channels in steady state})T^l(r) \\ = (1&-(1-p_s)^N)T^l(p) + (1-p_s)^N T^l(r). \end{aligned} \tag{26}$$

By explicitly including the probability mass function of the quantity l as a function of the ARQ delay statistics, from (25) and (26), we have

$$\begin{aligned} C_{\text{sum}}^{\text{genie}}(N) &= \mathrm{E}_l R_{\text{genie}}^{\text{greedy}}(l,N) \\ &= \sum_{l=0}^{\infty}\Big[(1-(1-p_s)^N)T^l(p) + (1-p_s)^N T^l(r)\Big]\times \\ &\qquad\qquad P(D\le l)\prod_{d=0}^{l-1}P(D>d). \end{aligned} \tag{27}$$

When $N=2$, with minor algebraic manipulations, we have

$$\begin{aligned} C_{\text{sum}}^{\text{genie}}&(2) \\ &= \sum_{l=0}^{\infty}\Big[p_s T^l(p) + (1-p_s)p_s\Big]P(D\le l)\prod_{d=0}^{l-1}P(D>d). \end{aligned} \tag{28}$$

We now proceed to prove that the sum throughput of the greedy policy in the original system equals that of the greedy policy in the genie-aided system when $N=2$. We established in the course of the proof of Proposition 2 that, in the original system with $N=2$, conditioned on $L=l$, the greedy decision in the current slot t is solely determined by the ARQ feedback from slot $t+l+1$ with the following decision rule: When the user scheduled in slot $t+l+1$, i.e., a_{t+l+1}, sends back an ACK, that user is scheduled in the current slot t, i.e., $\hat{a}_t = a_{t+l+1}$. Otherwise, the other user is scheduled in slot t. We can interpret this decision logic of the greedy policy as below: When at least one of the users had an ON channel in slot $t+l+1$, that user[4] is identified for scheduling in the current slot t, leading to an expected current reward of $T^l(p)$. Reward $T^l(r)$ is accrued only when both the channels were in the OFF state in slot $t+l+1$.

Note that the decision rule and the accrued immediate rewards corresponding to the greedy policy in the original system are the same as that of the greedy policy in the genie-aided system. Thus, in the original system, under the greedy policy, no improvement in the immediate reward can be achieved even if the channel states of both the users in slot $t+l+1$ are available at the scheduler in slot t. This, along with the fact that both the systems have the same delay profile, establishes the equivalence between the original and the genie-aided systems, when $N=2$, in terms of the sum throughput achieved by the greedy policy. We have already proved the sum throughput optimality of the greedy policy in the original system when $N=2$ (Proposition 2) and in the genie-aided system for a general value of N. Thus the sum capacity of the original system for $N=2$ is given by $C_{\text{sum}}^{\text{genie}}(2)$ in (28). The proposition thus follows. ■

Remarks: Insights on the result in Proposition 5 can be obtained by examining the fundamental trade-off when scheduling in the Markov-modeled downlink. In particular, scheduling must take into account

1) data transmission in the current slot, which influences the immediate reward, and
2) probing of the channel for future scheduling decisions, which influences the reward expected in future slots.

The optimal schedule strikes a balance between these two objectives (that need not contradict each other). From the discussion in the proof of Proposition 5, we see that, in the original system, when $N=2$, the choice of the user whose channel is probed becomes irrelevant as far as the optimal future reward is concerned. Similarly, in the genie-aided system, since the channel state information of all the users (general N system) is sent to the scheduler (with equal delay that is *i.i.d.* across the scheduling choice) irrespective of which user was scheduled, the optimal future reward is independent of the current scheduling decision. This results in the optimality of the greedy policy in the original and the genie-aided systems and creates a sum capacity equivalence between these two systems, when $N=2$.

The equivalence with the genie-aided system vanishes when $N>2$, since observing only one user is not enough to capture an 'ON-user,' if one exists. This was possible when $N=2$. Thus, when $N>2$, there is room for throughput improvement when the channel state information of all the users is available at the scheduler even if there is a delay (the genie-aided system). The genie-aided system sum capacity is thus an upper bound to the sum capacity of the original system. We record this next.

Corollary 6: When $N>2$, the sum capacity, $C_{\text{sum}}(N)$, of the downlink can be bounded as

$$C_{\text{sum}}(2) \le C_{\text{sum}}(N) \le C_{\text{sum}}^{\text{genie}}(N) \tag{29}$$

[4] User a_{t+l+1} is given higher priority if both channels were ON.

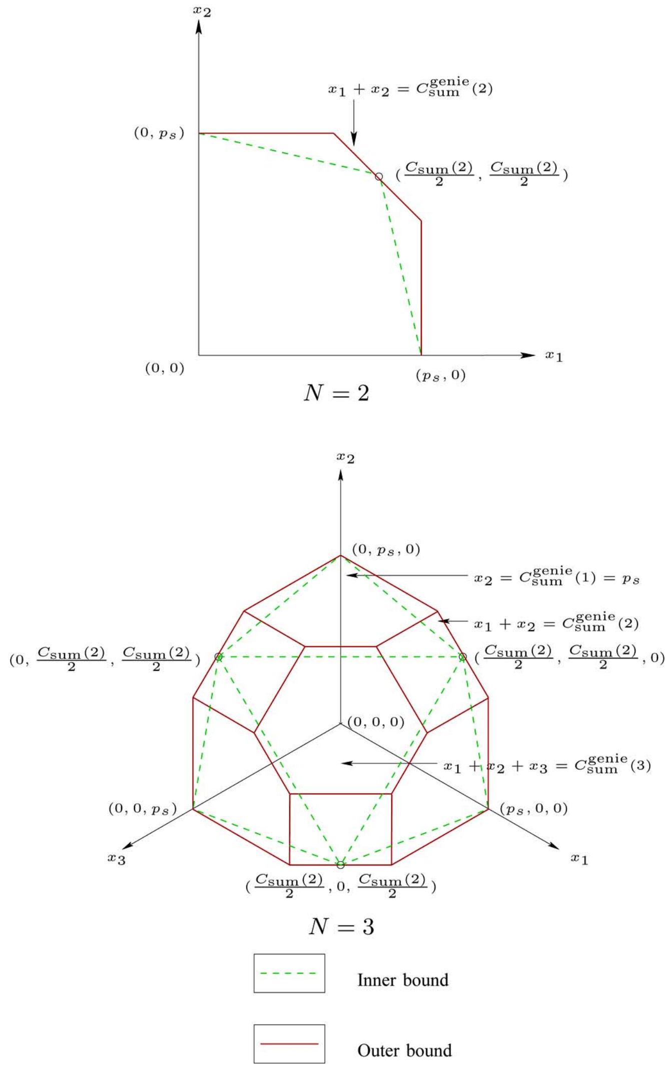

x_2
$x_1 + x_2 = C_{\text{sum}}^{\text{genie}}(2)$
$(0, p_s)$
$(\frac{C_{\text{sum}}(2)}{2}, \frac{C_{\text{sum}}(2)}{2})$
$(0,0)$
$(p_s, 0)$
x_1
$N = 2$
x_2
$(0, p_s, 0)$
$x_2 = C_{\text{sum}}^{\text{genie}}(1) = p_s$
$x_1 + x_2 = C_{\text{sum}}^{\text{genie}}(2)$
$(0, \frac{C_{\text{sum}}(2)}{2}, \frac{C_{\text{sum}}(2)}{2})$
$(\frac{C_{\text{sum}}(2)}{2}, \frac{C_{\text{sum}}(2)}{2}, 0)$
$(0,0,0)$
$x_1 + x_2 + x_3 = C_{\text{sum}}^{\text{genie}}(3)$
$(0, 0, p_s)$
$(p_s, 0, 0)$
x_3
x_1
$(\frac{C_{\text{sum}}(2)}{2}, 0, \frac{C_{\text{sum}}(2)}{2})$
$N = 3$
Inner bound
Outer bound

Fig. 4. Illustration of bounds on the capacity region of the downlink with randomly delayed ARQ when $N = 2$ and when $N = 3$.

Proof: The lower bound $C_{\text{sum}}(2)$, given in (24), is achieved by the scheduler when, in each slot, it considers only two users (fixed set) for scheduling and ignores the rest, effectively emulating a two-user downlink. The upper bound is the sum capacity of the genie-aided system with N users, as given in (27). ∎

B. Bounds on the Capacity Region of the Downlink

Define the capacity region of the downlink as the *exhaustive* set of achievable throughput vectors. Formally, let $\mu_i^{\mathfrak{A}}$ denote the throughput of user i under policy $\mathfrak{A}$. Let $I_k(i)$ be the indicator function on whether user i was scheduled in slot k, i.e.,

$$I_k(i) = \begin{cases} 1 & \text{if } i = a_k \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

Thus

$$\mu_i^{\mathfrak{A}} = \lim_{m \longrightarrow \infty} \frac{\mathrm{E}\left[\sum_{k=1}^{m} R_k^{\mathfrak{A}}(\pi_k, a_k) I_k(i)\right]}{m}, \tag{31}$$

where $R_k^{\mathfrak{A}}(\pi_k, a_k)$ is the immediate reward accrued by the scheduler in slot k under policy $\mathfrak{A}$. The expectation is over the belief vector π_k with steady state initial conditions. Now, the capacity region of the downlink, $\mathcal{C}$, is defined as the union of the throughput vectors, $(\mu_1^{\mathfrak{A}}, \dots, \mu_N^{\mathfrak{A}})$, over all scheduling policies, i.e.,

$$\mathcal{C} = \cup_{\mathfrak{A}} \{(\mu_1^{\mathfrak{A}}, \dots, \mu_N^{\mathfrak{A}})\}. \tag{32}$$

Let $H_{\text{convex}}(X)$ be the convex hull of the set of points X, defined as

$$\begin{aligned} &H_{\text{convex}}(X) \\ &\quad = \left\{ \sum_{i=1}^{n(X)} \beta_i x_i \;\middle|\; x_i \in X, \beta_i \in \mathbb{R}, \beta_i \geq 0, \sum_{i=1}^{n(X)} \beta_i = 1 \right\}. \end{aligned}$$

where $n(X)$ is the cardinality of set X. With these definitions we now state our results on the downlink capacity region.

Proposition 7: An outer bound on the capacity region of the Markov-modeled downlink with randomly delayed ARQ is given by the complement of the N-dimensional polyhedron $\mathcal{P}$ represented by

$$\begin{aligned} \mathcal{P} = \Big\{ &(x_1 \geq 0, x_2 \geq 0 \dots x_N \geq 0) : \\ &\sum_{i \in S} x_i \leq C_{\text{sum}}^{\text{genie}}(n(S)), \forall S \subseteq \{1, \dots N\} \Big\}, \end{aligned} \tag{33}$$

where

$$\begin{aligned} C_{\text{sum}}^{\text{genie}}(N) = \sum_{l=0}^{\infty} \Big[(1-(1-p_s)^N) T^l(p) + (1-p_s)^N T^l(r) \Big] \times \\ P(D \leq l) \prod_{d=0}^{l-1} P(D > d). \end{aligned}$$

An inner bound on the capacity region is given by the set of points $(x_1, \dots, x_N)$ such that

$$\begin{aligned} &(x_1, \dots, x_N) \\ &\quad \in H_{\text{convex}}(O, \{X_i\}_{\forall i \in \{1,\dots,N\}}, \{Y_{j,k}\}_{\forall j,k \in \{1,\dots,N\}, j \neq k}) \end{aligned} \tag{34}$$

where $O, X_i, Y_{j,k} \in \mathbb{R}^N$. O is the origin $(0, \dots, 0)$. $X_i = (0, \dots, 0, p_s, 0, \dots, 0)$ with p_s at the i^{th} location. $Y_{j,k,j \neq k} = (0, \dots, 0, \frac{C_{\text{sum}}(2)}{2}, 0, \dots, 0, \frac{C_{\text{sum}}(2)}{2}, 0, \dots, 0)$ with $\frac{C_{\text{sum}}(2)}{2}$ at locations j and k, where

$$\begin{aligned} C_{\text{sum}}(2) = \sum_{l=0}^{\infty} \Big[p_s T^l(p) + (1-p_s) p_s \Big] \times \\ P(D \leq l) \prod_{d=0}^{l-1} P(D > d). \end{aligned}$$

Fig. 4 illustrates the capacity region bounds from Proposition 7 when $N = 2$ and when $N = 3$. A detailed proof of the proposition is available in Appendix C.

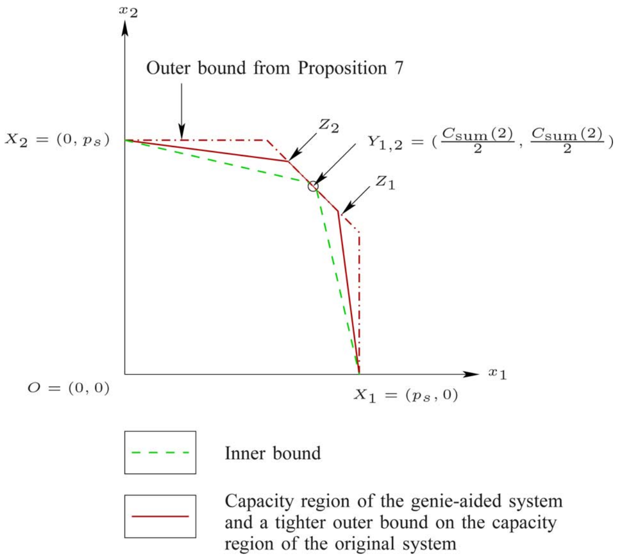


Fig. 5. Illustration of the capacity region of the genie-aided system and tighter bounds on the capacity region of the original system when $N = 2$, with deterministic ARQ delay.

For the special case of $N = 2$ users and deterministic ARQ feedback delay, $D = d$, we obtain the exact capacity region of the genie-aided system and hence tighter bounds to the capacity region of the original system.

Proposition 8: For $N = 2$ users, with a deterministic ARQ delay of $D = d$, $d \geq 0$ slots, the capacity region of the genie-aided system is given by the set of points (x_1, x_2) such that

$$\begin{aligned} &(x_1, x_2) \in H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2) \\ \text{where} \quad & O = (0,0) \\ & X_1 = (p_s, 0) \\ & X_2 = (0, p_s) \\ & Z_1 = \left(p_s T^d(p) + (1-p_s)^2 T^d(r), (1-p_s) p_s T^d(p)\right) \\ & Z_2 = \left((1-p_s) p_s T^d(p), p_s T^d(p) + (1-p_s)^2 T^d(r)\right). \end{aligned} \tag{35}$$

The relative positions of the points X_1, X_2, Z_1, Z_2 and O are illustrated in Fig. 5. A detailed proof of the proposition is available in Appendix D.

We now report tighter bounds on the capacity region of the original system, when $N = 2$ and the ARQ delay is deterministic.

Corollary 9: For $N = 2$ users, with a deterministic ARQ delay of $D = d$, $d \geq 0$ slots, an outer bound on the capacity region of the original system is given by the set of points (x_1, x_2) such that

$$\begin{aligned} &(x_1, x_2) \notin H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2) \\ \text{where} \quad & O = (0,0) \\ & X_1 = (p_s, 0) \\ & X_2 = (0, p_s) \\ & Z_1 = \left(p_s T^d(p) + (1-p_s)^2 T^d(r), (1-p_s) p_s T^d(p)\right) \\ & Z_2 = \left((1-p_s) p_s T^d(p), p_s T^d(p) + (1-p_s)^2 T^d(r)\right) \end{aligned} \tag{36}$$

and an inner bound is given by the set of points (x_1, x_2) such that

$$\begin{aligned} &(x_1, x_2) \in H_{\text{convex}}(O, X_1, Y_{1,2}, X_2) \\ &\text{where } Y_{1,2} = \left(\frac{C_{\text{sum}}(2)}{2}, \frac{C_{\text{sum}}(2)}{2}\right) \end{aligned}$$

with $C_{\text{sum}}(2) = p_s T^d(p) + (1-p_s)p_s$, the sum capacity of the system.

Proof: The outer bound is the region complementary to the capacity region of the genie-aided system reported in Proposition 8. The inner bound was obtained in Proposition 7 with $C_{\text{sum}}(2)$ from (24) re-derived using $P(D = d) = 1$. ■

Fig. 5 illustrates the improved outer bound from Corollary 9 along with the bounds derived in Proposition 7.

V. Conclusion

We addressed the problem of opportunistic multiuser scheduling for a system consisting of a base station or access point transmitting to users within its domain. We model the downlink channels by two-state Markov chains, with ON and OFF states, and assume that the data destined for each user is infinitely backlogged. We allow for the ARQ feedback from each user to the base station to be randomly (*i.i.d.* over all users) delayed. For the case of two users in the system, we showed that the greedy policy is sum throughput optimal for any distribution of the ARQ feedback delay. However, for more than two users, there exists scenarios for which the greedy policy is not optimal. Nevertheless, extensive numerical experiments suggest that the greedy policy has near optimal performance. Encouraged by this, we studied the structure of the greedy policy and showed that it can be implemented by a simple algorithm that does not require the statistics of the underlying Markov channel nor the ARQ feedback delay, thus making it robust against errors in estimation of these statistics. Focusing on the fundamental limits of the downlink system, we obtained an elegant closed form expression for the sum capacity of the two-user downlink and derived inner and outer bounds on the capacity region of the Markov-modeled downlink with randomly delayed ARQ feedback.

In summary, we addressed opportunistic multiuser scheduling based on existing ARQ feedback mechanisms, while taking into account an important non-ideality in the feedback channel—the random delay. We studied this scheduling problem by examining various aspects of the 'easy to implement' greedy policy and by establishing fundamental limits on the downlink system performance. We believe that the work we have initiated here, along with the proof techniques we have developed, could be the first steps towards studying the joint channel learning—scheduling problem under more general scenarios: such as, when the users have heterogeneous demands or when the queues are non-backlogged with random packet arrivals.

TABLE V
BELIEF VALUES, SCHEDULING DECISIONS, IMMEDIATE REWARDS IN SLOTS 2 AND 1 FOR VARIOUS REALIZATIONS OF ARQ FEEDBACK UNDER THE GREEDY POLICY

$\{f_4, f_3\}$	π_2 at slot 2	$\hat{a}_2$	$\hat{R}_2$	π_1 at slot 1	$\hat{a}_1$	$\hat{R}_1$
{1,1}	$\begin{bmatrix} T(p) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	1	$T(p)$	$\begin{bmatrix} T(p) \\ T^3(\pi_4(2)) \\ T^3(\pi_4(3)) \end{bmatrix}$	1	$T(p)$
{1,0}	$\begin{bmatrix} T(p) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	1	$T(p)$	$\begin{bmatrix} T(r) \\ T^3(\pi_4(2)) \\ T^3(\pi_4(3)) \end{bmatrix}$	2	$T^3(\pi_4(2))$
{0,1}	$\begin{bmatrix} T(r) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	2	$T^2(\pi_4(2))$	$\begin{bmatrix} T(p) \\ T^3(\pi_4(2)) \\ T^3(\pi_4(3)) \end{bmatrix}$	1	$T(p)$
{0,0}	$\begin{bmatrix} T(r) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	2	$T^2(\pi_4(2))$	$\begin{bmatrix} T(r) \\ T^3(\pi_4(2)) \\ T^3(\pi_4(3)) \end{bmatrix}$	2	$T^3(\pi_4(2))$

APPENDIX A
PROOF OF LEMMA 1

Recall the definition of the u-step belief evolution operator: $T^u(x) = T(T^{(u-1)}(x)) = T^{(u-1)}(T(x))$ with $T(x) = xp + (1-x)r = x(p-r) + r$ and $T^0(x) = x$ for $x \in [0,1]$ and $u \in \{0,1,2,\ldots\}$. For $u \in \{1,2,\ldots\}$, $x \in [0,1]$,

$$\begin{aligned} T^u(p) &= T^{(u-1)}(p)p + (1 - T^{(u-1)}(p))r \\ T^{(u+1)}(x) &= T^u(x)p + (1 - T^u(x))r \\ T^u(p) - T^{(u+1)}(x) &= (p-r)(T^{(u-1)}(p) - T^u(x)). \end{aligned} \tag{37}$$

Thus if, for $u \in \{1,2,\ldots\}$, $T^{(u-1)}(p) - T^u(x) \geq 0$, then, since $p > r$, we have $T^u(p) - T^{(u+1)}(x) \geq 0$. By induction, using $p \geq T(x) = xp + (1-x)r$ for any $x \in [0,1]$, we have $T^u(p) \geq T^{u+1}(x)$ for any $u \in \{0,1,2,\ldots\}$ and $x \in [0,1]$. The second inequality in the lemma can be proved along the same lines using $r \leq T(x) = xp + (1-x)r$.

Consider the third inequality. By definition, for any $x, y \in [0,1], T^u(x) - T^u(y) = (p-r)(T^{(u-1)}(x) - T^{(u-1)}(y))$. Thus, if $T^{(u-1)}(x) - T^{(u-1)}(y)$, then $T^u(x) - T^u(y) \geq 0$. When $x \geq y$, by induction, $T^u(x) - T^u(y) \geq 0$ for any $u \in \{0,1,2,\ldots\}$. This establishes the third inequality.

Considering the last inequality, the belief evolution operator can be expressed as

$$\begin{aligned} T^u(x) &= T(T^{(u-1)}(x)) = T(T(T^{(u-2)}(x))) \\ &= x(p-r)^u + r\left(\frac{1-(p-r)^u}{1-(p-r)}\right) \\ &= \frac{r}{1-(p-r)} + (p-r)^u\left(x - \frac{r}{1-(p-r)}\right) \end{aligned} \tag{38}$$

for $u \in \{0,1,2,\ldots\}$ and $x \in [0,1]$. Thus $T^u(p) = \frac{r}{1-(p-r)} + (p-r)^u\left[\frac{(p-r)(1-p)}{1-(p-r)}\right]$. Note that, since $p > r, T^u(p) \geq \frac{r}{1-(p-r)}$. Also, $T^u(r) = \frac{r}{1-(p-r)} - (p-r)^u\left[\frac{(p-r)r}{1-(p-r)}\right] \leq \frac{r}{1-(p-r)}$. This establishes the last inequality in the lemma.

APPENDIX B
PROOF OF PROPOSITION 4

The proof proceeds in two steps: (1) We first construct a counterexample to the optimality of the greedy policy when the horizon, $m = 4$ and arbitraty number of users N; $N > 2$, (2) Based on this counterexample, we then construct a more general counterexample, with arbitrary m; $m > 3$ and N; $N > 2$. We proceed with the first step next.

Assume an arbitrary number of users, N; $N > 2$. Let the horizon $m = 4$. Assume a deterministic ARQ delay of one time slot, i.e., $P_D(d = 1) = 1$ and $P_D(d \neq 1) = 0$. Let the users be indexed in decreasing order of their initial beliefs, i.e., $\pi_m(1) \geq \pi_m(2) \geq \cdots \pi_m(N)$. The net expected reward corresponding to the greedy policy is given by

$$\begin{aligned} &V_4(\pi_4, \{\hat{\mathfrak{A}}_k\}_{k=1}^4) \\ &\quad = \pi_4(1) + T(\pi_4(1)) \\ &\qquad + \mathrm{E}_{f_4|\pi_4,a_4=1}[\hat{R}_2] + \mathrm{E}_{f_3,f_4|\pi_4,a_4=1,a_3=1}[\hat{R}_1]. \end{aligned} \tag{39}$$

Note that since the delay is one slot, the first ARQ feedback comes at the end of slot 3. Thus, the greedy decision in both slots 4 and 3 is user 1. Also, the greedy scheduler has access to feedback f_4 only, at the beginning of slot 2 and both feedback f_4 and f_3, at the beginning of slot 1. Therefore, $\hat{R}_2$ is averaged over f_4 and $\hat{R}_1$ is averaged over f_4 *and* f_3. The average total reward under greedy policy can thus be evaluated by averaging over all realizations of f_4 and f_3. Table V lists the belief values of the three users in slots 2 and 1 for various values of $\{f_4, f_3\}$ along with the greedy decisions and immediate rewards in slots 2 and 1. Note from the table that the belief value π_2 at slot 2 is a function of f_4 only, while π_1 at slot 1 is a function of both f_4 and f_3, consistent with the preceding discussion.

The probabilities of occurrence of the various realizations of $\{f_4, f_3\}$ are summarized below

$$P(f_4, f_3) = \begin{cases} \pi_4(1)p, & \text{if } \{f_4, f_3\} = \{1,1\} \\ \pi_4(1)(1-p), & \text{if } \{f_4, f_3\} = \{1,0\} \\ (1-\pi_4(1))r, & \text{if } \{f_4, f_3\} = \{0,1\} \\ (1-\pi_4(1))(1-r), & \text{if } \{f_4, f_3\} = \{0,0\}. \end{cases} \tag{40}$$

TABLE VI
BELIEF VALUES, SCHEDULING DECISIONS, IMMEDIATE REWARDS IN SLOTS 2 AND 1 FOR VARIOUS REALIZATIONS OF ARQ FEEDBACK UNDER POLICY $\tilde{\boldsymbol{\mathfrak{A}}}_k$

$\{f_4, f_3\}$	π_2 at slot 2	$\tilde{a}_2$	$\tilde{R}_2$	π_1 at slot 1	$\tilde{a}_1$	$\tilde{R}_1$
{1,1}	$\begin{bmatrix} T(p) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	1	$T(p)$	$\begin{bmatrix} T^2(p) \\ T(p) \\ T^3(\pi_4(3)) \end{bmatrix}$	2	$T(p)$
{1,0}	$\begin{bmatrix} T(p) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	1	$T(p)$	$\begin{bmatrix} T^2(p) \\ T(r) \\ T^3(\pi_4(3)) \end{bmatrix}$	1	$T^2(p)$
{0,1}	$\begin{bmatrix} T(r) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	2	$T^2(\pi_4(2))$	$\begin{bmatrix} T^2(r) \\ T(p) \\ T^3(\pi_4(3)) \end{bmatrix}$	2	$T(p)$
{0,0}	$\begin{bmatrix} T(r) \\ T^2(\pi_4(2)) \\ T^2(\pi_4(3)) \end{bmatrix}$	2	$T^2(\pi_4(2))$	$\begin{bmatrix} T^2(r) \\ T(r) \\ T^3(\pi_4(3)) \end{bmatrix}$	3	$T^3(\pi_4(3))$

Thus the net expected reward under the greedy policy is given by

$$\begin{aligned} &V_4(\pi_4, \{\hat{\boldsymbol{\mathfrak{A}}}_k\}_{k=1}^4) \\ &\quad = \pi_4(1) + T(\pi_4(1)) + \pi_4(1)p\big(2T(p)\big) \\ &\qquad + \pi_4(1)(1-p)\big(T(p) + T^3(\pi_4(2))\big) \\ &\qquad + (1-\pi_4(1))r\big(T^2(\pi_4(2)) + T(p)\big) \\ &\qquad + (1-\pi_4(1))(1-r)\big(T^2(\pi_4(2)) + T^3(\pi_4(2))\big). \end{aligned} \tag{41}$$

Now, with a_k^* indicating the optimal decision in slot k, consider the following policy $\tilde{\boldsymbol{\mathfrak{A}}}_k$ such that $\tilde{a}_4 = 1, \tilde{a}_3 = 2, \tilde{a}_2 = a_2^*, \tilde{a}_1 = a_1^*$. Since the ARQ delay is deterministic and equals one slot, the decision in slot 2 does not affect the reward in slot 1. Thus the greedy policy is optimal in slot 2. Trivially, greedy policy is optimal in slot 1, as well. Thus $a_2^* = \hat{a}_2, a_1^* = \hat{a}_1$. The average total reward under $\tilde{\boldsymbol{\mathfrak{A}}}_k$ is given by

$$\begin{aligned} V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}_k\}_{k=1}^4) &= \pi_4(1) + T(\pi_4(2)) + \mathrm{E}_{f_4|\pi_4, a_4=1}[\tilde{R}_2] \\ &\quad + \mathrm{E}_{f_3, f_4|\pi_4, a_4=1, a_3=2}[\tilde{R}_1] \\ &= \pi_4(1) + T(\pi_4(2)) + \mathrm{E}_{f_4|\pi_4, a_4=1}[\hat{R}_2] \\ &\quad + \mathrm{E}_{f_3, f_4|\pi_4, a_4=1, a_3=2}[\hat{R}_1]. \end{aligned} \tag{42}$$

We evaluate $V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}_k\}_{k=1}^4)$ along the lines of the greedy net expected reward evaluation. Table VI summarizes the beliefs, scheduling decision $\tilde{a}_k$ and immediate rewards in slots 2 and 1 for all the realizations of $\{f_4, f_3\}$ when $\{\tilde{a}_4, \tilde{a}_3\} = \{1, 2\}$. Users are once again ordered according to their initial belief values, i.e., $\pi_4(1) \geq \pi_4(2) \geq \pi_4(3)$. Note from the table that the belief value π_2 at slot 2 is a function of f_4 only, while π_1 at slot 1 is a function of both f_4 and f_3, consistent with the ARQ delay profile.

The probabilities of occurrence of the various realizations of $\{f_4, f_3\}$ when $a_4 = 1, a_3 = 2$, are summarized below.

$$\begin{aligned} &P(f_4, f_3) \\ &= \begin{cases} \pi_4(1)T(\pi_4(2)), & \text{if } \{f_4, f_3\} = \{1,1\} \\ \pi_4(1)(1 - T(\pi_4(2))), & \text{if } \{f_4, f_3\} = \{1,0\} \\ (1-\pi_4(1))T(\pi_4(2)), & \text{if } \{f_4, f_3\} = \{0,1\} \\ (1-\pi_4(1))(1-T(\pi_4(2))), & \text{if } \{f_4, f_3\} = \{0,0\}. \end{cases} \end{aligned} \tag{43}$$

Thus, the net expected reward under policy $\tilde{\boldsymbol{\mathfrak{A}}}_k$ is given by

$$\begin{aligned} &V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) \\ &\quad = \pi_4(1) + T(\pi_4(2)) + \pi_4(1)T(\pi_4(2))\big(2T(p)\big) \\ &\qquad + \pi_4(1)(1 - T(\pi_4(2)))\big(T(p) + T^2(p)\big) \\ &\qquad + (1-\pi_4(1))T(\pi_4(2))\big(T^2(\pi_4(2)) + T(p)\big) \\ &\qquad + (1-\pi_4(1))(1-T(\pi_4(2)))\big(T^2(\pi_4(2)) + T^3(\pi_4(3))\big). \end{aligned} \tag{44}$$

We now proceed to show that, for $N = 3$, deterministic ARQ delay $D = 1$ and horizon $m = 4$, $\exists\ p, r, \pi_4$ such that the net expected reward corresponding to policy $\tilde{\boldsymbol{\mathfrak{A}}}_k$ is strictly higher than that of the greedy policy. The difference in reward, after algebraic manipulations is given by

$$\begin{aligned} &V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) - V_4(\pi_4, \{\hat{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) \\ &\quad = (p-r)\Big(\pi_4(2) - \pi_4(1) \\ &\qquad + (p-r)^2(1-\pi_4(1))\pi_4(3)\big(1 - r - (p-r)\pi_4(2)\big)\Big). \end{aligned} \tag{45}$$

For the special case $\pi_4(1) = \pi_4(2) = \frac{1}{2}$, we have

$$\begin{aligned} &V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) - V_4(\pi_4, \{\hat{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) \\ &\quad = \frac{(p-r)^3}{2}\Big(1 - \frac{p+r}{2}\Big)\pi_4(3). \end{aligned} \tag{46}$$

For any $p < 1$, since $p > r$, $V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4) > V_4(\pi_4, \{\hat{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4)\ \forall\ \pi_4(3) > 0$. With the net expected reward of the optimal policy being no less than $V_4(\pi_4, \{\tilde{\boldsymbol{\mathfrak{A}}}\}_{k=1}^4)$, we see that the greedy policy is not in general optimal. Table VII lists a few other values of p, r, π_4 for which the greedy policy is suboptimal. This establishes a counterexample for the optimality of the greedy policy when $N > 2$.

A more general counterexample for an arbitrary horizon length m can be constructed based on the one thus established. We proceed with this construction in the sequel. As before, assume $\pi_m(1) \geq \pi_m(2) \geq \cdots \pi_m(N)$ and a deterministic ARQ delay of one time slot, i.e., $P_D(d = 1) = 1$. With $S_k(i)$ indicating the underlying state of the channel of user

TABLE VII
SAMPLE SYSTEM PARAMETERS WHEN THE GREEDY POLICY IS SUBOPTIMAL. NUMBER OF USERS $N=3$, DETERMINISTIC DELAY $D=1$, HORIZON $m=4$ IS USED.

p	r	π_4	$V_4(\pi_4, \{\tilde{\mathfrak{A}}\}_{k=1}^4)$	$V_4(\pi_4, \{\hat{\mathfrak{A}}\}_{k=1}^4)$	$V_4(\pi_4, \{\tilde{\mathfrak{A}}\}_{k=1}^4)$ $-V_4(\pi_4, \{\hat{\mathfrak{A}}\}_{k=1}^4)$
0.9308	0.1797	$\begin{bmatrix}0.5216\\0.5130\\0.3305\end{bmatrix}$	2.6368	2.6141	0.0227
0.8875	0.0186	$\begin{bmatrix}0.3416\\0.3310\\0.2648\end{bmatrix}$	1.6155	1.5454	0.0701

i in slot k, consider the following realization of the channel: $\mathcal{R}_1 = \{S_m(1) = 1, S_{m-1}(1) = 1, \dots S_5(1) = 1;\ S_m(2) = 1, S_{m-1}(2) = 1, \dots S_5(2) = 1\}$. Recall the policy $\tilde{\mathfrak{A}}$ from above. We define a variant of this policy, $\tilde{\mathfrak{B}}$, as follows. Policy $\tilde{\mathfrak{B}}$ performs greedy scheduling in slots $\{m \dots 5\}$. Under realization $\mathcal{R}_1$, policy $\tilde{\mathfrak{B}}$, being a greedy scheduler, schedules user 1 in slots $\{m \dots 5\}$. Thus the realization $\{S_m(1) = 1, S_{m-1}(1) = 1, \dots S_6(1) = 1\}$ is observable by policy $\tilde{\mathfrak{B}}$ by the beginning of slot 4. From slot 4, under realization $\mathcal{R}_1$, define policy $\tilde{\mathfrak{B}}$ such that it behaves along the lines of policy $\tilde{\mathfrak{A}}$ defined earlier. Also, policy $\tilde{\mathfrak{B}}$ performs greedy scheduling in all slots $\{m \dots 1\}$ under all channel realizations other than $\mathcal{R}_1$. Thus the reward difference between policy $\tilde{\mathfrak{B}}$ and the greedy scheduler $\hat{\mathfrak{A}}$ is given by the difference in slots $\{4 \dots 1\}$, under realization $\mathcal{R}_1$, weighted by the probability of this realization. Formally,

$$\begin{aligned} &V_m(\pi_m, \{\tilde{\mathfrak{B}}\}_{k=1}^m) - V_m(\pi_m, \{\hat{\mathfrak{A}}\}_{k=1}^m) \\ &\quad = \mathrm{Prob}\{\mathcal{R}_1\} \\ &\qquad \left(V_4(\pi_4, \{\tilde{\mathfrak{A}}\}_{k=1}^4) - V_4(\pi_4, \{\hat{\mathfrak{A}}\}_{k=1}^4)\right) \end{aligned} \tag{47}$$

with $\pi_4 = \{p, p, T^{m-4}(\pi_m(3)), \dots T^{m-4}(\pi_m(N))\}$. Note that the belief values π_4 reflect the realization $\mathcal{R}_1$ and the greedy nature of both policies until slot 5. Now, since policy $\tilde{\mathfrak{B}}$ is defined to behave like policy $\tilde{\mathfrak{A}}$ from slot 4, we can use the reward difference expression in (45) to simplify (47), as below.

$$\begin{aligned} &V_m(\pi_m, \{\tilde{\mathfrak{B}}\}_{k=1}^m) - V_m(\pi_m, \{\hat{\mathfrak{A}}\}_{k=1}^m) \\ &\quad = \mathrm{Prob}\{\mathcal{R}_1\}\Big((p-r)\Big(\pi_4(2) - \pi_4(1) \\ &\qquad + (p-r)^2(1-\pi_4(1))\pi_4(3) \\ &\qquad \times \left(1 - r - (p-r)\pi_4(2)\right)\Big)\Big)\big|_{\pi_4(1)=\pi_4(2)=p} \\ &\quad = \mathrm{Prob}\{\mathcal{R}_1\}\Big((p-r)^3(1-p)((1-r) - (p-r)p)\Big). \end{aligned} \tag{48}$$

Note that the last equality is always positive rendering the greedy policy $\hat{\mathfrak{A}}$ suboptimal. This establishes a more general counterexample to the optimality of the greedy policy when $N > 2$. The proposition is thus proved.

APPENDIX C
PROOF OF PROPOSITION 7

Considering the genie-aided system, for any policy $\mathfrak{A}$, let the throughput vector be denoted by $(\mu_1^{\mathfrak{A},\text{genie}}, \dots, \mu_N^{\mathfrak{A},\text{genie}})$. For a subset of users $S \subseteq \{1 \dots N\}$, by the definition of sum capacity, we have

$$\sum_{i \in S} \mu_i^{\mathfrak{A},\text{genie}} \le C_{\text{sum}}^{\text{genie}}(n(S)). \tag{49}$$

This establishes the complement of the polyhedron $\mathcal{P}$ as an outer bound on the capacity region of the genie-aided system, and by extension, an outer bound on the capacity region of the original system.

Now, consider the inner bound $H_{\text{convex}}(O, \{X_i\}_{\forall i \in \{1,\dots,N\}}, \{Y_{j,k}\}_{\forall j,k \in \{1,\dots,N\}, j \neq k})$. In the original system, throughput vector $X_i = (0, \dots, 0, p_s, 0, \dots, 0)$ can be achieved by scheduling to user i at all times. Recall that the greedy policy achieves the sum capacity when $N = 2$. Also the sum throughput $C_{\text{sum}}(2)$ is split equally between the two users thanks to the inherent symmetry between users. Thus throughput vector $Y_{j,k,j\neq k} = (0, \dots, 0, \frac{C_{\text{sum}}(2)}{2}, 0, \dots, 0, \frac{C_{\text{sum}}(2)}{2}, 0, \dots, 0)$ can be achieved by greedy scheduling over the users j and k alone at all slots. Throughput vector O corresponds to idling in every slot. Therefore, any throughput vector in the convex hull $H_{\text{convex}}(O, \{X_i\}_{\forall i \in \{1,\dots,N\}}, \{Y_{j,k}\}_{\forall j,k \in \{1,\dots,N\}, j \neq k})$ can be achieved by time sharing between the policies that achieve throughput vectors $\in \{O, X_i, Y_{j,k,j\neq k}\}$. This establishes the result on the inner bound.

APPENDIX D
PROOF OF PROPOSITION 8

We proceed by first showing that the region complementary to $H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2)$ is an outer bound on the capacity region of the genie-aided downlink. Consider a broad class of schedulers in the genie-aided system, with each member identified by the parameters $\alpha_i \in [0, 1]$, $i \in \{1, \dots, 4\}$. A member of this class obeys the following decision logic at slot t:

- If $\begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, then schedule user 1 with probability α_1 and user 2 w.p. $1 - \alpha_1$.
- If $\begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, then $a_t = \begin{cases} 1 \ w.p. \ \alpha_2 \\ 2 \ w.p. \ 1 - \alpha_2 \end{cases}$
- If $\begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, then $a_t = \begin{cases} 1 \ w.p. \ \alpha_3 \\ 2 \ w.p. \ 1 - \alpha_3 \end{cases}$
- If $\begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, then $a_t = \begin{cases} 1 \ w.p. \ \alpha_4 \\ 2 \ w.p. \ 1 - \alpha_4 \end{cases}$

Note that, thanks to the first order Markovian nature of the underlying channels, any scheduling policy in the genie-aided system falls under the above class of schedulers or will have a member of this class achieving the same throughput vector as itself. We now proceed to show that the throughput vector achieved by any member of this class belongs to $H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2)$.

With $\vec{\alpha} = \{\alpha_1, \ldots, \alpha_4\} \in [0,1]^4$ fixed, the throughput for user 1 is given by

$$\begin{aligned}
&\mu_1^{\vec{\alpha},\text{genie}} \\
&= \sum_{i,j\in\{0,1\}} P\left(\begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} i \\ j \end{bmatrix}\right) \times \\
&P\left(a_t = 1 \middle| \begin{bmatrix} S_{t+d+1}(1) \\ S_{t+d+1}(2) \end{bmatrix} = \begin{bmatrix} i \\ j \end{bmatrix}\right) P(S_t(1) = 1 | S_{t+d+1}(1) = i) \\
&= (1-p_s)^2 \alpha_1 T^d(r) + (1-p_s) p_s \alpha_2 T^d(r) \\
&\quad + p_s(1-p_s)\alpha_3 T^d(p) + p_s^2 \alpha_4 T^d(p),
\end{aligned} \tag{50}$$

with $p_s = \frac{r}{1-(p-r)}$. Similarly,

$$\begin{aligned}
&\mu_2^{\vec{\alpha},\text{genie}} \\
&= (1-p_s)^2(1-\alpha_1)T^d(r) + (1-p_s)p_s(1-\alpha_2)T^d(p) \\
&\quad + p_s(1-p_s)(1-\alpha_3)T^d(r) + p_s^2(1-\alpha_4)T^d(p).
\end{aligned} \tag{51}$$

For notational simplicity, we will henceforth denote the throughputs simply by μ_1 and μ_2. The sum throughput is now given by

$$\mu_1 + \mu_2 = p_s + (1-p_s)p_s(T^d(p) - T^d(r))(\alpha_3 - \alpha_2). \tag{52}$$

Note that the values of α_1 and α_4 are irrelevant from the sum throughput point of view. From here, the proof proceeds by considering two cases: when $\alpha_3 \leq \alpha_2$ and when $\alpha_3 > \alpha_2$. When $\alpha_3 \leq \alpha_2$, we show that $(\mu_1, \mu_2) \in H_{\text{convex}}(O, X_1, X_2)$. When $\alpha_3 > \alpha_2$, we show that $(\mu_1, \mu_2) \in H_{\text{convex}}(X_1, Z_1, Z_2, X_2)$. Detailed proofs for these statements are available in [31]. Combining these two cases, we establish that the region complementary to $H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2)$ is an outer bound on the capacity region of the genie-aided system.

Revisiting the class of schedulers identified by $\vec{\alpha}$, it can be shown from (50) and (51) that a scheduler with $\vec{\alpha} = \{1, 0, 1, 1\}$ achieves a throughput vector $(\mu_1, \mu_2) = Z_1 = \big(p_s T^d(p) + (1-p_s)^2 T^d(r), (1-p_s)p_s T^d(p)\big)$. Similarly, a scheduler with $\vec{\alpha} = \{0, 0, 1, 0\}$ achieves a throughput vector $(\mu_1, \mu_2) = Z_2 = \big((1-p_s)p_s T^d(p), p_s T^d(p) + (1-p_s)^2 T^d(r)\big)$. Throughput vectors X_1 or X_2 can be achieved by scheduling to only user 1 or 2, respectively, at all times. Thus any throughput vector within the region $H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2)$ can be supported by time sharing between the schedulers that achieve throughput vector $\in \{O, X_1, Z_1, Z_2, X_2\}$. This establishes $H_{\text{convex}}(O, X_1, Z_1, Z_2, X_2)$ as an inner bound on the capacity region of the genie-aided system.

Combining the outer and inner bound results establishes the proposition.

Appendix E
Key Quantities

N	Number of users in the downlink.
p	$P(\text{channel is ON in the current slot} \mid \text{channel was ON in the previous slot})$.
r	$P(\text{channel is ON in the current slot} \mid \text{channel was OFF in the previous slot})$.
m	Horizon.
$\pi_t(i)$	Belief value of user i in slot t.
$T^u(.)$	u-step belief evolution operator.
a_k	Index of the user scheduled in slot k.
$\mathfrak{A}_k$	Scheduling policy applied in slot k.
$\widehat{\mathfrak{A}}_k$	Greedy scheduling policy applied in slot k.
f_k	Feedback originating from slot k.
F_t	Feedback arriving at slot t.
$D(i,t)$	Delay of feedback from user i in slot t.
$P_D(.)$	Probability mass function of *i.i.d.* delay D.
V_t	Net expected reward in slot t.
C_{sum}	Sum capacity of the downlink.
$C_{\text{sum}}^{\text{genie}}$	Sum capacity of the genie-aided downlink.
$\mu_i^{\mathfrak{A}}$	Throughput of user i under scheduling policy $\mathfrak{A}$.

References

[1] R. Knopp and P. A. Humblet, "Information capacity and power control in single cell multiuser communications," in *Proc. IEEE Int. Conf. Communications*, Seattle, WA, Jun. 1995, pp. 331–335.

[2] R. W. Heath, M. Airy, and A. J. Paulraj, "Multiuser diversity for MIMO wireless systems with linear receivers," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2001, pp. 1194–1199.

[3] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277–1294, Jun. 2002.

[4] J. Chung, C. S. Hwang, K. Kim, and Y. K. Kim, "A random beamforming technique in MIMO systems exploiting multiuser diversity," *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 848–855, Jun. 2003.

[5] A. Gyasi-Agyei, "Multiuser diversity based opportunistic scheduling for wireless data networks," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 670–672, Jul. 2005.

[6] J. Huang, R. A. Berry, and M. L. Honig, "Wireless scheduling with hybrid ARQ," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 2801–2005.

[7] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 2053–2064, Oct. 2001.

[8] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 473–489, Aug. 1999.

[9] T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair queueing algorithms," in *Proc. ACM Mobicom*, Aug. 1999.

[10] T. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in *Proc IEEE INFOCOM*, New York, 1998, vol. 3.

[11] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," in *Proc. ACM Workshop Wireless Mobile Multimedia*, Seattle, WA, Aug. 1999.

[12] Y. Cao and V. Li, "Scheduling algorithms in broadband wireless networks," *Proc. IEEE*, vol. 89, no. 1, pp. 76–87, Jan. 2001.

[13] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, Apr. 2004.

[14] M. Zorzi and R. Rao, "Error control and energy consumption in communications for nomadic computing," *IEEE Trans. Comput.*, vol. 46, pp. 279–289, Mar. 1997.

[15] L. A. Johnston and V. Krishnamurthy, "Opportunistic file transfer over a fading channel: A POMDP search theory formulation with optimal threshold policies," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 394–405, Feb. 2006.

[16] S. Lin, D. Costello, and M. Miller, "Automatic-repeat-request error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5–17, Dec. 1984.

[17] D. L. Lu and J. F. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. Commun.*, vol. 41, pp. 721–730, May 1993.

[18] M. Zorzi, R. R. Rao, and L. B. Milstein, "Arq error control on fading mobile radio channels," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 445–455, May 1997.

[19] Y. J. Cho and C. K. Un, "Performance analysis of ARQ error controls under Markovian block error pattern," *IEEE Trans. Commun.*, vol. 42, pp. 2051–2061, Feb.–Apr. 1994.

[20] E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, 1960.

[21] S. H. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multi-channel opportunistic access," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4040–4050, Sep. 2009.

[22] S. Murugesan, P. Schniter, and N. B. Shroff, "Multiuser scheduling in a Markov-modeled downlink environment," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 2008.

[23] H. Viswanathan, "Capacity of Markov channels with receiver csi and delayed feedback," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 761–771, Mar. 1999.

[24] L. Ying and S. Shakkottai, "On throughput optimality with delayed network-state information," in *Proc. 2008 Inf. Theory Appl. Workshop*, San Diego, CA, Feb. 2008.

[25] V. S. Annapureddy, D. V. Marathe, T. R. Ramya, and S. Bhashyam, "Outage probability of multiple-input and single-output (MISO) systems with delayed feedback," *IEEE Trans. Commun.*, vol. 57, pp. 319–326, 2009.

[26] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[27] H. S. Wang and N. Moayeri, "Finite-state Markov channel-A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163–171, 1995.

[28] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, pp. 1688–1692, 1999.

[29] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Oper. Res.*, Sep. 1973.

[30] S. Murugesan and P. Schniter, "Opportunistic multiuser scheduling in a three state Markov-modeled downlink," CoRR vol. 904, 2009.

[31] S. Murugesan, P. Schniter, and N. B. Shroff, "Multiuser scheduling in a Markov-modeled downlink using randomly delayed ARQ feedback," arXiv preprint:1002.3312v2, 2011.

Sugumar Murugesan (S'10–M'11) received the B.E. degree in Electronics and Communication Engineering from the College of Engineering, Anna University, India in 2004 and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from The Ohio State University, USA, in 2006 and 2010, respectively. He is a recipient of the OSU University Fellowship (2004–2005). Dr. Murugesan is now an Assistant Research Professor with the Department of ECEE at Arizona State University, and currently serves on the TPC of IEEE INFOCOM 2012. His research interests include communication theory, wireless networks, sequential decision processes and smart power grids.

Philip Schniter (S'03–M'93–SM'05) received the B.S. and M.S. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 1992 and 1993, respectively. From 1993 to 1996 he was employed by Tektronix Inc. in Beaverton, OR as a systems engineer, and in 2000, he received the Ph.D. degree in Electrical Engineering from Cornell University in Ithaca, NY. Subsequently, he joined the Department of Electrical and Computer Engineering at The Ohio State University in Columbus, OH, where he is now an Associate Professor and a member of the Information Processing Systems (IPS) Lab. In 2003, he received the National Science Foundation CAREER Award, and in 2008–2009 he was a visiting professor at Eurecom (Sophia Antipolis, France) and Supélec (Gif-sur-Yvette, France). Dr. Schniter's areas of interest include statistical signal processing, wireless communications and networks, and machine learning.

Ness B. Shroff (S'91–M'93–SM'01–F'07) received his Ph.D. degree from Columbia University, NY in 1994 and joined Purdue university immediately thereafter as an Assistant Professor. At Purdue, he became Professor of the school of Electrical and Computer Engineering in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In 2007, he joined The Ohio State University, and holds the Ohio Eminent Scholar Endowed chaired position of Networking and Communications in the departments of ECE and CSE.

His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, control, and security of these networks. Dr. Shroff has served on the editorial boards of IEEE/ACM Trans. on Networking, the Computer Networks Journal, and IEEE Communications Letters. He was the technical program co-chair of IEEE INFOCOM03, ACM Mobihoc 2008, and General co-chair of IEEE WICON'08 and IEEE CCW'99. He was also a co-organizer of NSF workshops on Fundamental Research in Networking (2003) and Future Directions in Wireless Networks (2008). Dr. Shroff has received numerous paper awards including the IEEE INFOCOM 2008 best paper award, IEEE INFOCOM 2006 best paper award, the IEEE IWQoS 2006 best student paper award, the 2005 best paper of the year award for the Journal of Communications and Networking, the 2003 best paper of the year award for Computer Networks, and the NSF CAREER Award in 1996 (his INFOCOM 2005 paper was also selected as one of two runner-up papers for the best paper award).