Incorporating Retransmission in Quality-of-Service Guaranteed Multiuser Scheduling Over Wireless Links

Xin Wang, *Member, IEEE*, Irena Li, Di Wang, Hanqi Zhuang, *Senior Member, IEEE*, and Salvatore D. Morgera, *Fellow, IEEE*

***Abstract*—A cross-layer optimization scheme combining retransmission with multiuser diversity is investigated for wireless communications. To this end, the joint design of adaptive modulation and coding (AMC) with the retransmission-based automatic repeat request (ARQ) protocol is first outlined. This design is then employed to devise multiuser scheduling schemes that can optimally capture the available multiuser diversity and retransmission-induced spectral efficiency gain. In addition, the proposed stochastic scheduling algorithms can operate even when the underlying fading channel distribution is unknown *a priori* while asymptotically converging to the optimum benchmark with guarantees on prescribed fairness and rate/delay requirements for a heterogeneous network traffic. Numerical results are provided to verify the advantage of these novel methods for multiuser transmissions over Nakagami block-fading channels.**

***Index Terms*—Cross-layer design, multiuser diversity, retransmission, wireless networking.**

I. Introduction

THE ADAPTIVE modulation and coding (AMC) technique has become a key component in wireless transmissions, as indicated by its adoption in current wireless standards, including the IEEE 802.11/15/16 and the 3rd Generation Partnership Project (3GPP) standards [1]–[3]. With AMC, an efficient resource allocation can be effected in wireless networks. Whereas the AMC at the physical (PHY) layer could achieve Shannon's limit for single-user channels with power and rate adaptation [4], extra degrees of freedom become available with multiuser links and may be exploited at the medium access control (MAC) layer by intelligently scheduling the most reliable channel for transmission. Hence, cross-layer optimization of the AMC and multiuser scheduling can approach the multiuser channel capacity, which, relative to single-user fading links, is enhanced (as the number of users grows) by the so-called multiuser diversity gain without increasing the power or rate. Building on such a cross-layer channel-adaptive approach, a class of "opportunistic" scheduling algorithms has been developed for nonreal-time [5]–[8] and real-time traffic [9]–[12]; see also a unified framework in [13].

In addition to the spatial multiuser diversity, a spectral efficiency gain can also be captured by proper retransmissions over fading channels. The value of this retransmission gain has recently been explored in [14] and [15], where it is shown that the cross-layer combining of the retransmission-based automatic repeat request (ARQ) protocol at the MAC layer with the AMC scheme at the PHY layer can enable considerable spectral efficiency gain for point-to-point wireless links.

Whereas the cross-layer designs in [14] and [15] focused on point-to-point links, the opportunistic multiuser scheduling in [5]–[13] did not account for the possible gain from retransmission. In contrast with these works, we consider incorporating the ARQ retransmission into the design of multiuser scheduling over wireless links. This becomes possible through a novel stochastic optimization approach to devising the stochastic scheduling algorithms based on a joint design of the ARQ protocol and the AMC scheme. With such a cross-layer ARQ–AMC design, the scheduling decisions in the proposed schemes could be significantly different from their counterparts based on transmission rate adaptation with the simple AMC scheme. The convergence analysis for the stochastic schemes in these two cases, however, follows a similar line. As such, this paper provides a novel approach to practical scheduling designs capable of jointly collecting both multiuser diversity and retransmission gain to enhance the spectral efficiency and the network performance for wireless networks with heterogeneous (nonreal-time and real-time) applications. For these applications, the proposed schemes can ensure fairness and quality of service (QoS) by maximizing a suitable utility function of average user rates and/or delays under average rate/delay constraints per user. To cope with the uncertainty in the wireless channel, our novel schemes are also capable of essentially learning the underlying fading channel distribution on the fly

Manuscript received October 30, 2008; revised March 3, 2009 and April 17, 2009. First published May 2, 2009; current version published October 2, 2009. This work was supported in part by the U.S. Department of Defense under Research Grant DCA-100-02-D400 and in part by the U.S. National Science Foundation under Grant CNS 0831671. This paper was presented in part at the 42nd Conference on Information Sciences and Systems, Princeton University, Princeton, NJ, March 19–21, 2008 and the 43rd Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, MD, March 18–20, 2009. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The review of this paper was coordinated by Prof. O. B. Akan.

X. Wang, D. Wang, H. Zhuang, and S. D. Morgera are with the Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: xin.wang@fau.edu; wdi@fau.edu; zhuang@fau.edu; smorgera@fau.edu).

I. Li was with the Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA. She is now with the NASA Johnson Space Center, Houston, TX 77058 USA (e-mail: ili@fau.edu).

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/TVT.2009.2021983

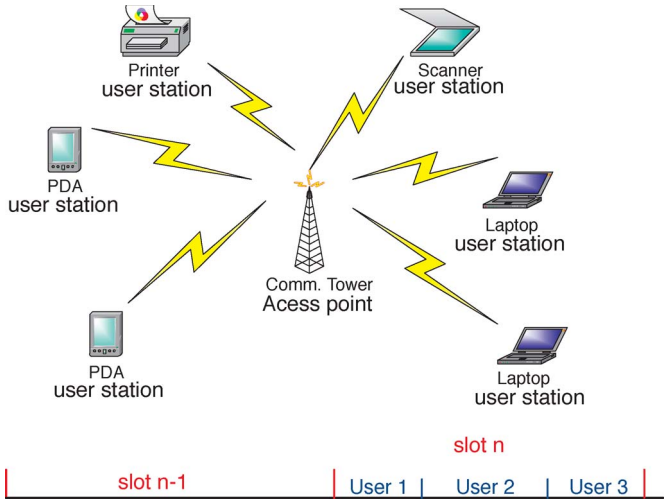


Fig. 1. Wireless network topology.

and asymptotically converging to the optimal scheduling policy with fairness and QoS guarantees for heterogeneous network traffic. Simulation results are also provided to demonstrate the performance gain of the proposed novel schemes over the existing alternatives.

This paper is organized as follows: We describe the system model under consideration and outline a joint ARQ and AMC design entailing the retransmission-induced spectral efficiency gain in Section II. The optimal multiuser scheduling jointly exploiting the multiuser diversity and the retransmission gain is then developed for nonreal-time, real-time, and heterogeneous network traffic in Section III. Simulation results are presented in Section IV to evaluate the proposed schemes, followed by the conclusions in Section V.

II. System Model and a Joint AMC-ARQ Design

A. Modeling Preliminaries

As depicted in Fig. 1, we consider a star-like wireless network topology, where multiple users are connected to an access point (AP) over wireless links. Notice that this star topology also describes the connections between the relay station and the multiple nodes in mobile ad hoc networks and wireless sensor networks. For specificity, we focus on a downlink setup, where the AP transmits data packets to K connected users, but our results can be extended to the uplink as well. Packet transmissions from the AP to the users are naturally slot based, and in each slot, the AP communicates with the users using time-division multiplexing.

To schedule active user connections, the AP maintains a separate queue for each incoming packet stream destined to different users, as shown in Fig. 2. An intelligent scheduler at the AP allocates resources among user connections based on the collected channel state information (CSI) and the queue-length information.

For each connection from the AP to a user, multiple transmission modes are available at the PHY layer, with each mode representing a pair of specific modulation format [e.g., 16 quadratic-amplitude modulation (16-QAM)] and

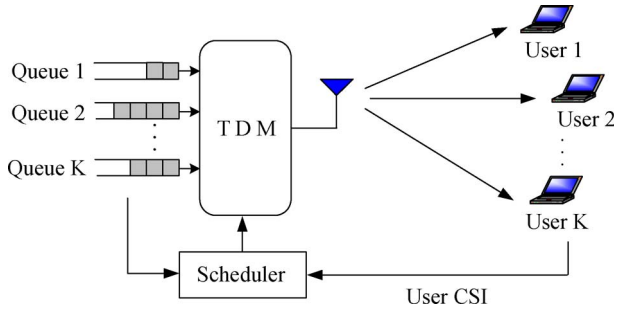


Fig. 2. Downlink multiuser scheduling for real-time connections.

an error control code (e.g., a convolutional code), as in the IEEE 802.11/15/16 and 3GPP standards. Based on the CSI fed back from the user, the AMC controller at the AP selects a modulation-coding pair for transmission.

At the MAC layer, the packet error detections rely on cyclic redundancy check (CRC) codes, with which nearly perfect detection can be assumed [16]. Based on CRC detection, an ARQ protocol is implemented. If an erroneous packet is detected, then a retransmission request will be fed back to the AP from the user. A retransmission of the requested packet will then be scheduled by the AP. Each packet from the AP to the kth user $k = 1, \ldots, K$ can be retransmitted up to N_r times upon packet errors. If a packet is not correctly received after N_r retransmissions, then it is dropped and declared to be lost.

B. Joint Design of AMC With ARQ

Based on the system model, we describe a joint ARQ and AMC design that will be adopted in the development of the proposed multiuser scheduling schemes in the sequel. At the PHY layer, the AP performs AMC based on the given CSI. With $\rho_0 := 0$ standing for no transmission (the zeroth mode), $M + 1$ AMC modes with rate ρ_m bits per second per hertz $m = 0, 1, \ldots, M$ (and $\rho_m < \rho_{m+1}$) can be employed for transmissions. Suppose that a prescribed packet error rate (PER) $\check{P}_{e,\mathrm{MAC}}$ needs to be maintained at the MAC layer to guarantee error performance per AP-user connection. For the given retransmission limit N_r, the PER required at the PHY layer is clearly

$$\check{P}_{e,\mathrm{PHY}} = [\check{P}_{e,\mathrm{MAC}}]^{\frac{1}{N_r+1}}. \tag{1}$$

With the required $\check{P}_{e,\mathrm{PHY}}$, AMC selection is carried out per AP-user link. Specifically, based on its received SNR γ_k, user k can determine the optimal AMC mode index number as

$$m^*(\gamma_k) = \max\left\{m \,|\, P_{e,\mathrm{PHY}}(m, \gamma_k) \le \check{P}_{e,\mathrm{PHY}}\right\} \tag{2}$$

where $P_{e,\mathrm{PHY}}(m, \gamma_k)$ denotes the PER when the mth AMC mode is used with SNR γ_k. Clearly, if user k is scheduled, then the AP shall transmit using the $m^*(\gamma_k)$th mode with rate $\rho_{m^*(\gamma_k)}$, which is the maximum rate user k can support under the required $\check{P}_{e,\mathrm{PHY}}$.

It is easy to see that the PER function $P_{e,\mathrm{PHY}}(m, \gamma_k)$ in (2) is a decreasing function of γ_k, i.e., the PER decreases as the SNR γ_k increases for a fixed mode m. On the other hand,

$P_{e,\mathrm{PHY}}(m,\gamma_k)$ should be an increasing function of m since the PER, in general, increases if a higher transmission rate is maintained for a given γ_k [20]. Therefore, the AMC mode selection rule in (2) indeed partitions the user k's SNR range into $M+1$ consecutive intervals, where the boundary points $\gamma_{k,m}$ can be obtained by solving $P_{e,\mathrm{PHY}}(m,\gamma_{k,m})=\check{P}_{e,\mathrm{PHY}}$, $m=1,\ldots,M$. With $\gamma_{k,0}:=0$ and $\gamma_{k,M+1}:=\infty$, the rule (2) is then equivalent to

mode $m^*(\gamma_k)$ is selected for user k

$$\text{when } \gamma_k \in \left[\gamma_{k,m^*(\gamma_k)}, \gamma_{k,m^*(\gamma_k)+1}\right). \quad (3)$$

In accordance with (3), the AP should not transmit to user k (i.e., $m^*(\gamma_k)=0$) when $\gamma_k<\gamma_{k,1}$ to avoid a deep fade. Moreover, it is possible to select a larger $m^*(\gamma_k)$ and, thus, a higher rate for a higher γ_k. Notice that for the AP to perform the desirable AMC selection, user k only needs to feed back the *quantized* (Q-) CSI $m^*(\gamma_k)$ using $\log_2(M+1)$ bits.

Finally, it is worth clarifying how retransmission is capable of bringing gain in spectral efficiency. Conventionally, retransmissions are regarded as a scheme ensuring communication reliability at the cost of spectral efficiency. It is true that retransmissions of the same packet incur extra bandwidth consumption. However, this cost can be compensated by a potential gain in spectral efficiency, as follows. Due to the allowance of retransmissions, a packet error will be claimed at the link layer only if multiple (re)transmissions of this packet fail. Therefore, because of the independent fading that those multiple transmissions could experience, the required PER at the physical layer is alleviated. Consequently, a higher rate can be selected by the AMC schemes for transmissions, which leads to a potential spectral efficiency gain.

Specifically, if retransmission is not allowed, then we have $N_r=0$ in (1), and thus, the PER requirement at the PHY layer coincides with that at the MAC layer, i.e., $\check{P}_{e,\mathrm{PHY}}=\check{P}_{e,\mathrm{MAC}}$. On the other hand, with $N_r>0$ retransmissions allowed, the required $\check{P}_{e,\mathrm{PHY}}>\check{P}_{e,\mathrm{MAC}}$ is alleviated for the AMC selection in (2). This implies that for the same SNR γ_k, the AP may transmit using a mode with a higher rate, which renders better spectral efficiency. Of course, the latter gain should be countered by the possible retransmission of the same packet. However, since the ARQ protocol activates retransmission only when necessary, the composite impact of allowing retransmission can probably entail a gain in spectral efficiency when the parameter N_r is properly selected. In other words, there is likely a retransmission gain available through the judicious utilization of wireless fading channels. This intuition has been corroborated by [14] for point-to-point communications and will be also corroborated by the simulation results in Section IV for multiuser links.

III. Optimal Multiuser Scheduling

In this section, we build on the joint AMC-ARQ design in Section II to develop novel stochastic scheduling schemes that are capable of jointly collecting the available multiuser diversity and retransmission gain in wireless networks. For our scheduling designs, we adopt the following operating conditions.

- oc-1) The wireless links between the AP and the users are modeled as frequency-flat block fading channels, where the SNR vector $\boldsymbol{\gamma}:=[\gamma_1,\ldots,\gamma_K]^T$ is fixed per slot n but is allowed to vary from slot to slot according to a stationary and ergodic random process with distribution function $F(\boldsymbol{\gamma})$.
- oc-2) Each user terminal k has the full information of $\gamma_k\ \forall k$, and the AP obtains a quantized CSI vector $\boldsymbol{m}^*(\boldsymbol{\gamma}):=[m^*(\gamma_1),\ldots,m^*(\gamma_K)]^T$ via a finite-rate feedback from the users per slot n.

The block-fading model in oc-1) is widely adopted in communication system design and reflects wireless links with slowly moving terminals in practice [14], [20]. The full receive CSI assumption at users in oc-2) is possible via training-based channel estimation [13], [20], and the quantized transmit CSI at the AP is entailed by the AMC selection rule in (2). Under the operating conditions oc-1) and oc-2), we first consider the scheduling of nonreal-time connections.

A. Scheduling Nonreal-Time Traffic

For nonreal-time connections, it is reasonable to assume that all the data arrive and are stored in the queues before scheduling; hence, the queues can be assumed infinitely backlogged for one-hop communications in Fig. 1. Recall that the AP transmits to the users based on time-division multiplexing. Let $\boldsymbol{\tau}(\boldsymbol{\gamma}):=[\tau_1(\boldsymbol{\gamma}),\ldots,\tau_K(\boldsymbol{\gamma})]^T$ denote the time-sharing fractions of a slot allocated for user $k=1,\ldots,K$ upon $\boldsymbol{\gamma}$. Assuming, without loss of generality, that the slot duration $T_s=1$, clearly, a scheduling policy $\boldsymbol{\tau}:=\{\boldsymbol{\tau}(\boldsymbol{\gamma}),\ \forall\boldsymbol{\gamma}\}$ in the feasible set $\mathcal{F}$ should satisfy $\sum_{k=1}^K\tau_k(\boldsymbol{\gamma})\le 1\ \forall\boldsymbol{\gamma}$. Upon denoting $\bar{\boldsymbol{r}}(\boldsymbol{\tau}):=[\bar{r}_1(\boldsymbol{\tau}),\ldots,\bar{r}_K(\boldsymbol{\tau})]^T$ as the resultant average user rates for a given policy $\boldsymbol{\tau}$, and selecting a utility function $U(\bar{\boldsymbol{r}})$, the optimal scheduler then wishes to solve

$$\max_{\boldsymbol{\tau}\in\mathcal{F}} U\left(\bar{\boldsymbol{r}}(\boldsymbol{\tau})\right) \qquad \text{subject to (s. to)} \quad \bar{\boldsymbol{r}}(\boldsymbol{\tau})\ge\check{\boldsymbol{r}} \qquad (4)$$

where $\check{\boldsymbol{r}}:=[\check{r}_1,\ldots,\check{r}_K]^T$ are the minimum average rate requirements to guarantee the QoS of user connection $k=1,\ldots,K$, and henceforth, the inequalities of vectors are defined as element wise. To ensure the solvers of (4) can in principle attain the global optimum, we assume the following.

- A1) The utility function $U(\cdot)$ is selected to be a concave and increasing function.
- A2) The set of time allocation policies satisfying $\bar{\boldsymbol{r}}(\boldsymbol{\tau})>\check{\boldsymbol{r}}$ is not empty.

The concavity of U guarantees that the optimization problem in (4) is a convex optimization. Choosing U to be increasing is reasonable, since the benefit of user k should increase as the average rate $\bar{r}_k$ increases. These justify the assumptions in A1). On the other hand, A2) ensures that the problem in (4) is strictly feasible. It simply requires that the minimum average rate requirements $\check{\boldsymbol{r}}$ are affordable by the given wireless channels, i.e., $\check{\boldsymbol{r}}$ belongs to the interior of the achievable average rate region of the channels between the AP and the users. Notice

that the achievable average rates are always bounded due to the finite transmit power; hence, the utility U is also bounded.[1] Assumption A2) can be ensured by a connection admission control scheme, which admits user connections with feasible QoS requests and/or drop connections when their requested QoS cannot be fulfilled (see [13]).

The utility maximization paradigm in (4) has proven successful in devising efficient and fair-scheduling schemes [6], [8], [9], [13]. In fact, the maximization of a carefully chosen utility function can balance the overall throughput and fairness among users. For instance, using the logarithmic utility function leads to the development of the well-known proportional fair-scheduling algorithm, under which all users are served with equal probability, regardless of their different average SNRs, when there are no minimum rate constraints, i.e., $\check{\boldsymbol{r}} = 0$ [5].

Under A1) and A2), the problem in (4) is a strictly feasible convex optimization problem that can efficiently be solved using rich convex programming tools [21]. Interestingly, this problem can also be tackled via a stochastic primal–dual method without knowing the fading distribution function $F(\boldsymbol{\gamma})$. In this method, we initialize using two vectors $\hat{\bar{\boldsymbol{r}}}[0] := [\hat{\bar{r}}_1[0], \ldots, \hat{\bar{r}}_K[0]]^T$ and $\hat{\boldsymbol{\lambda}}[0] := [\hat{\lambda}_1[0], \ldots, \hat{\lambda}_K[0]]^T$, where $\hat{\bar{\boldsymbol{r}}}$ stands for the estimates of the average user rates, and $\hat{\boldsymbol{\lambda}}$ stands for the estimates of Lagrange multipliers associated with the minimum average rate constraints [21].

With $\hat{\bar{\boldsymbol{r}}}[n]$ and $\hat{\boldsymbol{\lambda}}[n]$ available (from the previous iteration) and $m^*(\gamma_k[n])\ \forall\, k$ known at the AP per slot n, the scheduler then adopts a "winner-takes-all" strategy, as in the opportunistic scheduling [5], [6], [9], [13]. Specifically, upon denoting $\nabla U(\hat{\bar{\boldsymbol{r}}}[n]) := [\nabla U_1(\hat{\bar{\boldsymbol{r}}}[n]), \ldots, \nabla U_K(\hat{\bar{\boldsymbol{r}}}[n])]^T$ as the gradient (vector) of $U(\hat{\bar{\boldsymbol{r}}}[n])$, we pick the winner index

$$k^*[n] = \arg \max_{k=1,\ldots,K} \left(\nabla U_k \left(\hat{\bar{\boldsymbol{r}}}[n] \right) + \hat{\lambda}_k[n] \right) \rho_{m^*(\gamma_k[n])} \quad (5)$$

and assign the entire slot to this winner, i.e.,

$$\tau_k^* (\boldsymbol{\gamma}[n]) = \begin{cases} 1, & \text{if } k = k^*[n] \\ 0, & \text{if } k \neq k^*[n]. \end{cases} \quad (6)$$

In other words, the optimal policy per slot n is to schedule a user capable of supporting the largest weighted rate, where the user weights are given by $\nabla U_k(\hat{\bar{\boldsymbol{r}}}[n]) + \hat{\lambda}_k[n]\ \forall\, k$. Notice that here the scheduling decision in (5) is based on the rate adaptation $\rho_{m^*(\gamma_k[n])}$ dictated by the joint ARQ–AMC design. This could significantly be different from that based on a simple AMC rate adaptation in, e.g., [5], [6], [9], and [13].

According to (6), the AP transmits to user $k^*[n]$ with a *scheduled* rate $\rho_{m^*(\gamma_{k^*}[n])}$. Taking into account erroneous packet receptions notified by the user, the *effective* transmit rate $r_{k^*}[n]$ can be evaluated at the AP. Using $r_{k^*}[n]$, the scheduler then updates $\hat{\bar{\boldsymbol{r}}}$ and $\hat{\boldsymbol{\lambda}}$ as follows: $\forall\, k$

$$\hat{\bar{r}}_k[n+1] = \begin{cases} (1-\beta)\hat{\bar{r}}_k[n] + \beta r_{k^*}[n], & k = k^*[n] \\ (1-\beta)\hat{\bar{r}}_k[n], & k \neq k^*[n] \end{cases} \quad (7)$$

$$\hat{\lambda}_k[n+1] = \begin{cases} \left[\hat{\lambda}_k[n] + \beta \left(\check{r}_k - r_{k^*}[n] \right) \right]^+, & k = k^*[n] \\ \left[\hat{\lambda}_k[n] + \beta \check{r}_k \right]^+, & k \neq k^*[n] \end{cases} \quad (8)$$

where β is a small step size, and $[x]^+ := \max(0, x)$. The updated $\hat{\bar{\boldsymbol{r}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$ will be utilized for the scheduling decision in the next slot.

Notice that $\rho_{m^*(\gamma_{k^*}[n])}$ is the transmission rate scheduled for the winner user k^* by the AP, which determines the number of packets to be transmitted at slot n. On the other hand, $r_{k^*}[n]$ is the actual rate received by user k^*, which is determined by the number of packets correctly received. Since packet errors can individually be notified, $r_{k^*}[n]$ is a number between 0 and $\rho_{m^*(\gamma_{k^*}[n])}$. Moreover, it is the actual rate $r_{k^*}[n]$ that should be used to update the estimate of the average rate $\check{\bar{r}}_k$, which will in turn be utilized for the subsequent scheduling decisions. In fact, it follows from the definition of PER $P_{e,\mathrm{PHY}}(m, \gamma)$ that these two rates are related by

$$\mathbb{E}\left[r_{k^*}[n] \right] = \rho_{m^*(\gamma_{k^*}[n])} \left(1 - P_{e,\mathrm{PHY}} \left(m^* \left(\gamma_{k^*}[n] \right), \gamma_{k^*}[n] \right) \right) \quad (9)$$

where we take expectation over the additive noise.

To summarize, the proposed multiuser scheduling algorithm operates as follows.

Algorithm 1: Scheduling for nonreal-time traffic

1) **Initialize** with any $\hat{\bar{\boldsymbol{r}}}[0]$ and $\hat{\boldsymbol{\lambda}}[0]$ and per slot n.
2) **Repeat online**: With $\hat{\bar{\boldsymbol{r}}}[n]$ and $\hat{\boldsymbol{\lambda}}[n]$ available from the last iteration and given the current Q-CSI $\boldsymbol{m}^*(\boldsymbol{\gamma}[n])$, the AP schedules the connections in accordance with the winner-takes-all policy determined by (5) and (6) and then uses (7) and (8) to obtain $\hat{\bar{\boldsymbol{r}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$.

Algorithm 1, which is a simple stochastic scheduling algorithm, is also asymptotically optimal. In (7) and (8), $\hat{\bar{\boldsymbol{r}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$ are updated based on the instantaneous (i.e., a stochastic estimate of average) rates. For (16), the updates of primal variables (i.e., estimates of average rates) are the same as those in the standard "opportunistic" algorithms [5], [6], [9]. In addition, in (8), the way to update dual variables (i.e., estimates of Lagrange multipliers associated with rate constraints) follows a stochastic subgradient projection approach. Hence, overall, the iterations in (7) and (8) follow the greedy primal–dual (GPD) approach originated in [9], where the convergence of such a scheme was established based on the fluid limit and Lyapunov drift arguments. Specifically, consider the corresponding "fluid-scaled" continuous-time random process $\hat{\bar{\boldsymbol{r}}}(t/\beta)$ and $\hat{\boldsymbol{\lambda}}(t/\beta)$ for the discrete-time GPD process $\hat{\bar{\boldsymbol{r}}}[n]$ and $\hat{\boldsymbol{\lambda}}[n]$ in (7) and (8) with $n = \lfloor t/\beta \rfloor$ (i.e., n is the largest integer less than or equal to t/β). Then, as $\beta \to 0$, a deterministic process called fluid sample path (FSP) arises as the limiting case of the foregoing fluid-scaled process. Assuming that the fading process is Markovian, it can also be shown that any weak limit of the GPD trajectory follows such an FSP with probability 1 as $\beta \to 0$. Since the FSP is asymptotically optimal, which can be proven via a Lyapunov argument under A1) and A2), so is the GPD scheme.

[1] A uniformly bounded U can facilitate the convergence analysis of the proposed stochastic primal–dual iterations since it guarantees a Lipschitz condition for system evaluation [6].

Furthermore, the latter weak convergence result has been enhanced by our recent work in [13] and [17], where it was shown that the iterations in (7) and (8) follow a stochastic feasible direction principle, and their convergence can be established using the stochastic averaging tools in [18]. As a consequence, the Markov assumption can be relaxed, and only the stationarity of the fading in oc-1) is required. Specifically, with $\bar{\boldsymbol{r}}^* = \bar{\boldsymbol{r}}(\boldsymbol{\tau}^*)$ denoting the optimal solution for (4), under A1) and A2), the estimate $\hat{\bar{\boldsymbol{r}}}[n]$ asymptotically converges to $\bar{\boldsymbol{r}}^*$, i.e., $\lim_{n\to\infty} \hat{\bar{\boldsymbol{r}}}[n] \to \bar{\boldsymbol{r}}^*$, in probability as $\beta \to 0$, and the corresponding time allocation converges to the globally optimal allocation for (4). This implies that the stochastic scheduling policy specified in (6) asymptotically converges to the optimal policy solving (4).

From Algorithm 1, the scheduler at the AP "greedily" assigns the entire time per slot to a single winner with the largest weighted rate, where the weight of user k is adaptively provided by $\nabla U_k(\hat{\bar{\boldsymbol{r}}}[n]) + \hat{\lambda}_k[n]$. Upon convergence, the optimal scheduling then follows such a winner-takes-all policy with user weights $\nabla U_k(\bar{\boldsymbol{r}}^*) + \lambda_k^* \; \forall\, k$, where λ_k^* denotes the optimal Lagrange multiplier. When user k experiences a good channel state, i.e., $\gamma_k[n]$ is large, then its weighted rate becomes large, and it is likely to be scheduled for slot n. It is worth clarifying that the term "winner-takes-all" should not be misunderstood. Although one winner is chosen per slot, the optimally scheduled winner (as well as its transmit rate) varies across fading realizations. As the random $\boldsymbol{\gamma}$ varies per slot, the AP can then capitalize on multiuser diversity as it schedules the user terminal with the "best" channel (in terms of the largest weighted rate). On the other hand, the user weights $\nabla U_k(\bar{\boldsymbol{r}}^*) + \lambda_k^*$ account for the desirable fairness and prescribed rate requirements. Notice that different from prior works [5]–[13], the joint AMC-ARQ design specified in Section II is incorporated for transmit rate allocation in the proposed scheduling scheme to also collect the retransmission-induced spectral efficiency gain.

Algorithm 1 is valuable since it is arguably as simple as any heuristic scheme. In addition, the value of the proposed scheduling scheme can further be appreciated if we take into account that it operates without knowing the fading distribution $F(\boldsymbol{\gamma})$ *a priori*. In other words, the simple stochastic updates in (7) and (8) are capable of learning the fading statistics of the intended wireless links "on-the-fly" and optimally exploiting the available multiuser diversity and retransmission gain with guarantees on the prescribed average rate requirements.

B. Scheduling Real-Time Traffic

We next consider scheduling for real-time traffic, which remains an active research topic with rich challenges to be tackled. Typically, real-time services, such as video conferencing and streaming, request guarantees on both throughput and latency. Adherence to these requirements necessitates linking delay with rate, which is a task that involves the size of each user's queue. Whereas we can reasonably assume infinite backlogged queues in nonreal-time scheduling, for real-time traffic, it is necessary to account for queueing delays. Let $\boldsymbol{q} := [q_1, \ldots, q_K]^T$ denote the vector of the current queue lengths (in bits or packets). For AP transmission based on time-division multiplexing, the time allocation $\boldsymbol{\tau}(\boldsymbol{q}, \boldsymbol{\gamma}) := [\tau_1(\boldsymbol{q}, \boldsymbol{\gamma}), \ldots, \tau_K(\boldsymbol{q}, \boldsymbol{\gamma})]^T$ now collects the time-sharing fractions of a slot allocated for user $k = 1, \ldots, K$ upon the current queue length vector $\boldsymbol{q}$ and CSI vector $\boldsymbol{\gamma}$. Again, a scheduling policy $\boldsymbol{\tau} := \{\boldsymbol{\tau}(\boldsymbol{q}, \boldsymbol{\gamma}) \; \forall\, \boldsymbol{q}, \boldsymbol{\gamma}\}$ is feasible, i.e., it is in the feasible set $\mathcal{F}$ if it satisfies $\sum_{k=1}^{K} \tau_k(\boldsymbol{q}, \boldsymbol{\gamma}) \leq 1 \; \forall\, \boldsymbol{q}, \boldsymbol{\gamma}$. Notice that in contrast with the nonreal-time case, the time allocation here also depends on the queue length vector $\boldsymbol{q}$.

For real-time connections, let $\boldsymbol{a}[n] := [a_1[n], \ldots, a_K[n]]^T$ collect the number of (bit or packet) arrivals during slot n and $\bar{\boldsymbol{a}} := [\bar{a}_1, \ldots, \bar{a}_K]^T$ denote the average arrival rates. For a given schedule policy $\boldsymbol{\tau}(\cdot, \cdot)$, the queue lengths then evolve according to $\forall\, k = 1, \ldots, K$

$$q_k[n+1] = \left[q_k[n] - \tau_k\left(\boldsymbol{q}[n], \boldsymbol{\gamma}[n]\right) \rho_{m^*(\gamma_k[n])}\right]^+ + a_k[n] \quad (10)$$

where $\rho_{m^*(\gamma_k)}$ is the maximum rate user k can support under the required $\check{P}_{e,\mathrm{PHY}}$ per (2), and thus, $\tau_k(\boldsymbol{q}[n], \boldsymbol{\gamma}[n]) \rho_{m^*(\gamma_k[n])}$ represents the possible maximum number of departures for user k under the given scheduling policy. Clearly, the actual departures cannot exceed the current queue length $q_k[n]$; hence, the $[\cdot]^+$ operator is enforced in the queue evolution.

Based on the well-known Little's law [19], the resultant average queueing delays $\bar{\boldsymbol{d}}(\boldsymbol{\tau}) := [\bar{d}_1(\boldsymbol{\tau}), \ldots, \bar{d}_K(\boldsymbol{\tau})]^T$ for a given policy $\boldsymbol{\tau}(\cdot, \cdot)$ can be obtained from the expected queue lengths $\forall\, k = 1, \ldots, K$

$$\begin{aligned} \bar{d}_k(\boldsymbol{\tau}) &= \bar{a}_k^{-1} \mathbb{E}_{\boldsymbol{q},\boldsymbol{\gamma}} \left[q_k[n+1]\right] \\ &= \bar{a}_k^{-1} \mathbb{E}_{\boldsymbol{q},\boldsymbol{\gamma}} \left[\left[q_k - \tau_k(\boldsymbol{q}, \boldsymbol{\gamma}) \rho_{m^*(\gamma_k)}\right]^+\right] + 1 \end{aligned} \quad (11)$$

where the average arrival rate $\bar{a}_k$ can be assumed known or can be estimated beforehand when real-time connections maintain a fixed average data rate. The identity (11) relates the average delays with the time schedules and forms a solid basis on which we pursue desirable scheduling algorithms for real-time traffic.

When real-time services demand a fixed data rate, and there is little benefit in providing them with larger rates than what they need, the major objective becomes the minimization of delays (instead of maximizing the spectral efficiency). To guarantee the QoS for real-time connections, suppose that the average user delays need to be controlled under prescribed maximum values in $\check{\boldsymbol{d}} := [\check{d}_1, \ldots, \check{d}_K]^T$. Upon selecting a utility function $U(\bar{\boldsymbol{d}})$,[2] as with (4), the optimal scheduler then wishes to solve

$$\max_{\boldsymbol{\tau} \in \mathcal{F}} U\left(\bar{\boldsymbol{d}}(\boldsymbol{\tau})\right) \qquad \text{subject to (s. to)} \quad \bar{\boldsymbol{d}}(\boldsymbol{\tau}) \leq \check{\boldsymbol{d}} \quad (12)$$

where we assume the following.

- A3) The utility function $U(\cdot)$ is selected to be a concave and *decreasing* function.
- A4) The set of time allocation policies satisfying $\bar{\boldsymbol{d}}(\boldsymbol{\tau}) < \check{\boldsymbol{d}}$ is not empty.

As opposed to (4), here, we maximize a utility function of average user delays to entail fairness among the users. For the same reasons in Section III-A, the chosen utility function U is

[2]We abuse the notation a little by using U to denote the utility functions for both nonreal-time and real-time traffic, but in general, the chosen utility functions for these two cases are different.

concave. In distinction to A1), here, U is a decreasing function with respect to $\bar{d}_k\ \forall k$. This is reasonable, since the benefit of user k should increase as its average delay $\bar{d}_k$ decreases. As with A2), A4) ensures that the problem (12) is strictly feasible.

Under A3) and A4), again, the problem in (12) can be solved by using a stochastic optimization method such as a primal–dual algorithm without knowing the fading distribution function $F(\boldsymbol{\gamma})$. In this approach, we initialize with two nonnegative vectors $\hat{\bar{\boldsymbol{d}}}[0] := [\hat{\bar{d}}_1[0], \ldots, \hat{\bar{d}}_K[0]]^T$ and $\hat{\boldsymbol{\lambda}}[0] := [\hat{\lambda}_1[0], \ldots, \hat{\lambda}_K[0]]^T$, where $\hat{\bar{\boldsymbol{d}}}$ stands for the estimates of the average user delays, and $\hat{\boldsymbol{\lambda}}$ denotes the estimates of Lagrange multipliers associated with the maximum average delay constraints, i.e., $\bar{\boldsymbol{d}}(\boldsymbol{\tau}) \leq \check{\boldsymbol{d}}$.

With $\hat{\bar{\boldsymbol{d}}}[n]$ and $\hat{\boldsymbol{\lambda}}[n]$ available (from the previous iteration) and $m^*(\gamma_k[n])$, $q_k[n]\ \forall k$, which is known at the AP per slot n, the scheduler then adopts a "winner-goes-first" strategy. Specifically, upon denoting $\nabla U(\hat{\bar{\boldsymbol{d}}}[n]) := [\nabla U_1(\hat{\bar{\boldsymbol{d}}}[n]), \ldots, \nabla U_K(\hat{\bar{\boldsymbol{d}}}[n])]^T$ as the gradient (vector) of $U(\hat{\bar{\boldsymbol{r}}}[n])$, we calculate the weighted rate per user as

$$\bar{a}_k^{-1}\left(-\nabla U_k\left(\hat{\bar{\boldsymbol{d}}}[n]\right) + \hat{\lambda}_k[n]\right)\rho_{m^*(\gamma_k[n])} \qquad \forall k \tag{13}$$

where the adaptive weight per user

$$\hat{w}_k[n] := \bar{a}_k^{-1}\left(-\nabla U_k\left(\hat{\bar{\boldsymbol{d}}}[n]\right) + \hat{\lambda}_k[n]\right)$$

is always nonnegative since U is a decreasing function per A3), and $\hat{\lambda}_k$ is a nonnegative Lagrange multiplier associated with an inequality constraint.

Based on (13), we sort the users in the decreasing order of the weighted rates to obtain an index vector $\boldsymbol{u}[n] := [u_1[n], \ldots, u_K[n]]^T$, such that $\hat{w}_{u_k}[n]\rho_{m^*(\gamma_{u_k}[n])} \geq \hat{w}_{u_{k+1}}[n]\rho_{m^*(\gamma_{u_{k+1}}[n])}\ \forall k \in [1, K-1]$. Then, the optimal scheduling policy is to allocate to the user $u_k[n]$

$$\tau^*_{u_k}(\boldsymbol{q}[n], \boldsymbol{\gamma}[n]) = \min\left(\frac{q_{u_k}[n]}{\rho_{m^*\left(\gamma_{u_k}[n]\right)}}, 1 - \sum_{j=1}^{k-1}\tau^*_{u_j}(\boldsymbol{q}[n], \boldsymbol{\gamma}[n])\right). \tag{14}$$

Basically, this is a "winner-goes-first" policy, where the scheduler first considers assigning the entire slot to user $u_1[n]$ with the maximum weighted rate per joint queue and channel realization $(\boldsymbol{q}[n], \boldsymbol{\gamma}[n])$. If only part of the slot is required to serve all the data in $u_1[n]$'s queue, however, then the remaining time will be assigned to user $u_2[n]$, which has the second largest weighted rate. This allocation continues until the entire slot is assigned or the data in all the user queues are cleared. Different from the "winner-takes-all" policy for the nonreal-time case, here, the scheduling policy in (14) depends on the queue sizes. It can be shown that such a policy maximizes the marginal increase of the utility $U(\hat{\bar{\boldsymbol{d}}}[n+1]) - U(\hat{\bar{\boldsymbol{d}}}[n])$ for the given $\hat{\bar{\boldsymbol{d}}}[n]$, $\hat{\boldsymbol{\lambda}}[n]$, $\boldsymbol{q}[n]$, and $\boldsymbol{\gamma}[n]$ (see [13] and [17]).

According to (14), the AP schedules to evacuate $\tau^*_k(\boldsymbol{q}[n], \boldsymbol{\gamma}[n])\rho_{m^*(\gamma_k[n])}$ bits or packets from user k's queue. Taking into account erroneous packet receptions notified by the user, the AP can evaluate the *actual* departure $r_k[n]$, whose expected value is approximately

$$\mathbb{E}[r_k[n]] = \tau^*_k(\boldsymbol{q}[n], \boldsymbol{\gamma}[n])\,\rho_{m^*(\gamma_k[n])} \times \left(1 - P_{e,\mathrm{PHY}}\left(m^*(\gamma_k[n]), \gamma_k[n]\right)\right)$$

where the expectation is over the noise. Based on the actual departures $r_k[n]$, the queues evolve according to

$$q_k[n+1] = q_k[n] - r_k[n] + a_k[n] \qquad \forall k. \tag{15}$$

With $q_k[n+1]$, the scheduler can then update $\forall k$

$$\hat{\bar{d}}_k[n+1] = (1-\beta)\hat{\bar{d}}_k[n] + \beta q_k[n+1]/\bar{a}_k \tag{16}$$

$$\hat{\lambda}_k[n+1] = \left[\hat{\lambda}_k[n] + \beta\left(q_k[n+1]/\bar{a}_k - \check{d}_k\right)\right]^+ \tag{17}$$

where β is a small step size. The updated $\hat{\bar{\boldsymbol{d}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$ will be utilized for the scheduling decision in the next slot.

To summarize, the proposed scheduling algorithm operates as follows.

Algorithm 2: Scheduling for real-time traffic

1) **Initialize** with any nonnegative $\hat{\bar{\boldsymbol{d}}}[0]$ and $\hat{\boldsymbol{\lambda}}[0]$ and per slot n.
2) **Repeat online:** With $\hat{\bar{\boldsymbol{d}}}[n]$ and $\hat{\boldsymbol{\lambda}}[n]$ available from the last iteration and given the current queue length vector $\boldsymbol{q}[n]$ as well as Q-CSI $\boldsymbol{m}^*(\boldsymbol{\gamma}[n])$, the AP schedules the connections in accordance with the winner-goes-first policy determined by (14) and then uses (16) and (17) to obtain $\hat{\bar{\boldsymbol{d}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$.

In (16) and (17), the updates of $\hat{\bar{\boldsymbol{d}}}[n+1]$ and $\hat{\boldsymbol{\lambda}}[n+1]$ are based on the instantaneous (i.e., a stochastic estimate of average) delays $q_k[n+1]/\bar{a}_k$. (Recall the relation between the queue length and the delay by Little's law.) As with (7) and (8), the iterations in (16) and (17) follow a GPD approach. Thus, it can be shown that the stochastic scheduling policy specified in (14) can asymptotically converge to the policy maximizing the utility with average delay guarantees in (12) under A3) and A4).

In the proposed scheduling policy (14), the scheduler at the AP adopts a greedy "winner-goes-first" strategy that assigns time fractions to users per slot in a decreasing order of the weighted user rates. Clearly, when user k experiences a good channel state, i.e., $\gamma_k[n]$ is large, then its weighted rate becomes large, and it is likely to go first, i.e., it is first scheduled for slot n. As the random $\boldsymbol{\gamma}$ varies per slot, the AP can then capitalize on multiuser diversity as it first schedules the user terminals with the "best" channels (in terms of large weighted rate). On the other hand, the user weights $\hat{w}_k[n] := \bar{a}_k^{-1}(-\nabla U_k(\hat{\bar{\boldsymbol{d}}}[n]) + \hat{\lambda}_k[n])$ are adapted to account for the desirable fairness and prescribed maximum average delay requirements. In addition, with the joint AMC-ARQ design incorporated for transmit-rate

TABLE I
TRANSMISSION MODES WITH UNCODED M-QAM MODULATION

	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Modulation	BPSK	QPSK	8-QAM	16-QAM	32-QAM	64-QAM	128-QAM
Rate (bits/sec/Hz)	1	2	3	4	5	6	7
α_m	67.7328	73.8279	58.7332	55.9137	50.0552	42.5594	40.2559
g_m	0.9819	0.4945	0.1641	0.0989	0.0381	0.0235	0.0094
γ_{pm} (dB)	6.3281	9.3945	13.9470	16.0938	20.1103	22.0340	25.9677

allocation, the proposed scheduling scheme again capitalizes on the retransmission-induced spectral efficiency gain. Overall, the simple stochastic Algorithm 2 is capable of learning the fading statistics of the intended wireless links "on-the-fly" and optimally exploiting the available multiuser diversity and retransmission gain for real-time scheduling. This is a universal approach to utility-based scheduling for a real-time traffic with stationary packet arrival process, and it overcomes the limitations of existing alternatives, e.g., the alternatives in [22] that only work for a linear utility function with Poisson arrival process or generic concave utility but without new arrivals.

C. Scheduling Heterogeneous Traffic

The common utility maximization paradigm for nonreal-time and real-time scheduling naturally suggests a unified approach to scheduling heterogeneous traffic with a mixture of nonreal-time and real-time traffic as follows. Let $\bar{\boldsymbol{r}} := [\bar{r}_1, \ldots, \bar{r}_I]^T$ and $\bar{\boldsymbol{d}} := [\bar{d}_1, \ldots, \bar{d}_J]^T$ denote the average rates and delays for I nonreal-time and J real-time connections, respectively. With selected utility functions $U_{nrt}(\bar{\boldsymbol{r}})$ and $U_{rt}(\bar{\boldsymbol{d}})$, we then wish to solve

$$\max_{\boldsymbol{\tau}\in\mathcal{F}} U_{nrt}\left(\bar{\boldsymbol{r}}(\boldsymbol{\tau})\right) + U_{rt}\left(\bar{\boldsymbol{d}}(\boldsymbol{\tau})\right) \qquad \text{s.to} \quad \bar{\boldsymbol{r}}(\boldsymbol{\tau}) \geq \check{\boldsymbol{r}} \quad \bar{\boldsymbol{d}} \leq \check{\boldsymbol{d}} \tag{18}$$

where $\check{\boldsymbol{r}} = [\check{r}_1, \ldots, \check{r}_I]^T$ and $\check{\boldsymbol{d}} = [\check{d}_1, \ldots, \check{d}_J]^T$ denote the prescribed minimum average rate and maximum average delay requirements, respectively.

To solve (18), stochastic scheduling for heterogeneous traffic is obtained by simply combining Algorithms 1 and 2. Specifically, per slot n, we compare the weighted rates $(\nabla U_{nrt,i}(\hat{\bar{\boldsymbol{r}}}[n]) + \hat{\lambda}_i[n])\rho_{m^*(\gamma_i[n])}$ or $\bar{a}_k^{-1}(-\nabla U_{rt,j}(\hat{\bar{\boldsymbol{d}}}[n]) + \hat{\lambda}_j[n])\rho_{m^*(\gamma_j[n])}$. If the winner user with maximum weighted rate is a nonreal-time user, then we assign to this user the entire slot, or if the winner user is a real-time user, then we allocate the remaining time among the other users with the same rule only if part of the slot is required to serve all the data in this user's queue. All $\hat{\bar{r}}_i[n]$, $\hat{\lambda}_i[n]$, $\hat{\bar{d}}_j[n]$, and $\hat{\lambda}_j[n]$ are then updated in accordance with this allocation, as in (7) and (8) and (16) and (17). This constitutes a stochastic scheme that can jointly capture the multiuser diversity and retransmission gain in scheduling heterogeneous traffic.

IV. NUMERICAL RESULTS

In this section, we test the proposed schemes in a simulated IEEE 802.16 downlink, where the system bandwidth is $B = 1$ MHz, and the slot length is $T_s = 1.08$ ms. The user fading processes are independent and subject to the general Nakagami-m flat fading[3] with average SNR $\bar{\gamma}_k$, i.e., the instantaneous SNR γ_k per slot is a random variable distributed according to a Gamma probability density function

$$f(\gamma_k) = \frac{m^m \gamma_k^{m-1}}{\bar{\gamma}_k^m \Gamma(m)} \exp\left(-\frac{m\gamma_k}{\bar{\gamma}_k}\right) \tag{19}$$

where $\Gamma(m) := \int_0^\infty t^{m-1}e^{-t}dt$ is the Gamma function, and $m \geq 1/2$ is the Nakagami fading parameter. In all the simulations, we assume that the Nakagami fading parameter $m = 1$, $\forall\, k$ (which corresponds to Rayleigh fading), for the wireless links.

The AP transmits to the user data packets, each of which consists of 1080 bits. As shown in Table I, seven uncoded M-QAM modulation modes could be employed for AP transmissions. With these modes, the PER can closely be approximated using an exponential curve as in [14]

$$P_{e,\mathrm{PHY}}(m,\gamma_k) \approx \begin{cases} 1, & \text{if } 0 < \gamma_k < \gamma_{pm} \\ \alpha_m \exp(-g_m \gamma_k), & \text{if } \gamma_k \geq \gamma_{pm} \end{cases} \tag{20}$$

where α_m, g_m, and γ_{pm} are obtained by fitting (20) to the exact PER, and their values are listed in Table I per mode index m.

In all the simulations, the prescribed PER at the MAC layer $\check{P}_{e,\mathrm{MAC}} = 0.01$. For a given retransmission limit N_r, the PER $\check{P}_{e,\mathrm{PHY}}$ at the PHY layer is given by (1). Inverting (20) from $P_{e,\mathrm{PHY}}(m,\gamma_k) = \check{P}_{e,\mathrm{PHY}}$, we then obtain the boundary points as

$$\gamma_{k,0} = 0 \tag{21}$$

$$\gamma_{k,m} = \frac{1}{g_m} \ln\left(\frac{\alpha_m}{\check{P}_{e,\mathrm{PHY}}}\right), \quad m = 1, \ldots, M \tag{22}$$

$$\gamma_{k,M+1} = \infty. \tag{23}$$

Using these values, the AMC selection at the AP follows (3).

Test Case 1: Suppose first that all the user links maintain nonreal-time services, have the same average SNRs $\bar{\gamma}_k = \bar{\gamma}$, and have no minimum rate requirements (i.e., $\check{\boldsymbol{r}} = \mathbf{0}$), and the utility function is selected as the sum of the average user rates (i.e., $U(\bar{\boldsymbol{r}}) := \sum_{k=1}^K \bar{r}_k$). Such a utility (sum of average rates) also indicates the spectral efficiency when it is normalized in units of bits per second per hertz. Notice that when evaluating the average user rates or the spectral efficiency, we only count the "good" throughput. In other words, the erroneous packets (which require retransmissions) are not taken into account.

[3]We abuse the notation to use m to denote the Nakagami parameter only in this paragraph; this should not be confused with the mode index m used throughput the paper.

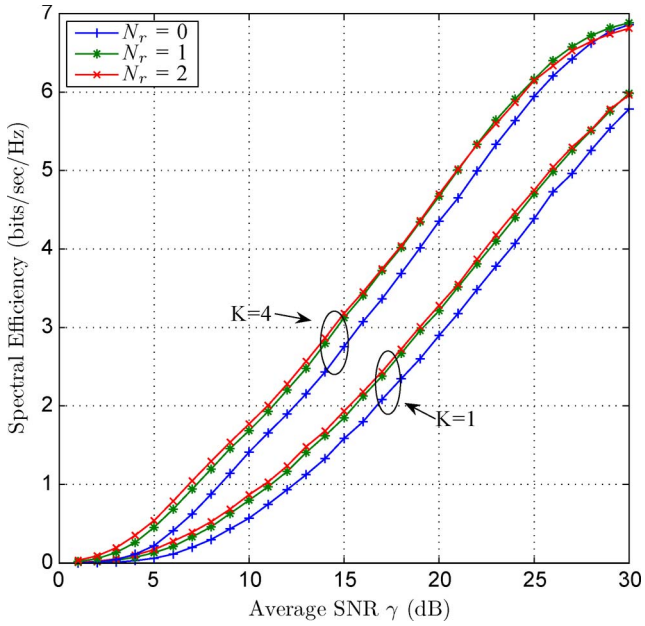


Fig. 3. Spectral efficiency for different N_r values when the number of users $K = 1$ and $K = 4$.

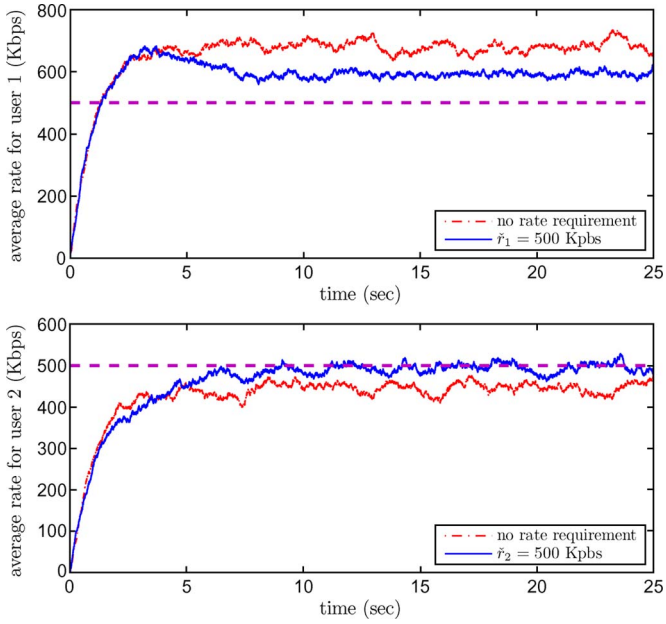


Fig. 4. Average rate evolutions for the proposed nonreal-time scheduling scheme. (Dashed lines indicate $\check{r}_k = 500$ kb/s.)

Fig. 3 depicts the resultant spectral efficiency when different N_r values are adopted for single-user $(K = 1)$ and multiuser $(K = 4)$ cases. It is shown that only allowing $N_r = 1$ retransmissions can bring about 2-dB SNR improvement in spectral efficiency for both single-user and multiuser cases thanks to the retransmission gain, whereas such an improvement quickly degrades as N_r increases [14]. It is also evident that the multiuser diversity is capable of bringing significant gain since the spectral efficiency for the four-user case presents a 5-dB gain over that of the single-user case. Overall, the combination of multiuser diversity and retransmission gain can account for the 7-dB increment in spectral efficiency in this example.

Test Case 2: We assume that there are $K = 2$ nonreal-time users with average SNRs: $\bar{\gamma}_1 = 10$ dB and $\bar{\gamma}_2 = 8$ dB. The utility function is chosen as $U(\bar{\boldsymbol{r}}) := \sum_{k=1}^{K} \ln(\bar{r}_k)$. We consider two cases: 1) There are no rate requirements, and 2) the minimum average rate requirements for the users are $\check{r}_k = 500$ kb/s $\forall\, k$. For both cases, Fig. 4 depicts the evolution of $\hat{\bar{r}}_k[n]$ in (7) when a step size $\beta = 0.001$ is employed. It is clear that as the number of slots n grows, the average user rates converge. When there are no rate requirements, the average user rates converge to around $\bar{r}_1 \approx 700$ kb/s and $\bar{r}_2 \approx 450$ kb/s, which maximize the selected utility function. With the selected logarithmic utility, the proposed scheme indeed performs a proportional fair-scheduling. The two users are served with equal probability. User 1 results in a higher average rate than user 2 simply because the former has a better average channel quality, i.e., average SNR. When minimum rate requirements $\check{r}_k = 500$ kb/s are present, the average user rates converge to around $\bar{r}_1 \approx 600$ kb/s and $\bar{r}_2 \approx 500$ kb/s, i.e., both rate requirements are met. In contrast to case 1, these rates maximize the selected utility function under the conditions $\bar{r}_k \geq \check{r}_k\ \forall\, k$. To this end, we sacrifice some resources for user 1 in case 1 to guarantee the average rate requirement for user 2. The results for case 2 clearly demonstrate that the proposed scheme approaches the optimal schedules with QoS guarantees. Notice that $\hat{\bar{\boldsymbol{r}}}[n]$ converges to the exact $\bar{\boldsymbol{r}}^*$ only when a vanishing $\beta \to 0$ is adopted. For a constant $\beta > 0$, $\hat{\bar{\boldsymbol{r}}}[n]$ will reach the neighborhood of the optimal $\bar{\boldsymbol{r}}^*$ and hover around it. This explains the variation of $\hat{\bar{r}}_k[n]$ after convergence in Fig. 4.

Test Case 3: We now consider a downlink where there are $K = 2$ real-time user connections with average SNRs: $\bar{\gamma}_1 = 10$ dB, and $\bar{\gamma}_2 = 12$ dB. The arrival processes to user queues are assumed to be Bernoulli distributed with given average rates $\bar{a}_k$ and parameters $\pi_k \in (0, 1)$ [11]. As a result, the instantaneous arrival rate at time slot n for user k is given by

$$a_k[n] = \begin{cases} 0, & \text{with probability } \pi_k \\ \bar{a}_k/(1-\pi_k), & \text{with probability } 1-\pi_k. \end{cases} \tag{24}$$

The parameters for the arrival processes are set to $\bar{a}_1 = 300$ kb/s, $\pi_1 = 0.6$, and $\bar{a}_2 = 300$ kb/s, $\pi_2 = 0.4$. The utility function is chosen as $U(\bar{\boldsymbol{d}}) := \sum_{k=1}^{K}(-\bar{d}_k^2)$, and the maximum average delay requirements for the users are $\check{d}_k = 4.0$ ms $\forall\, k$. Fig. 5 depicts the evolution of $\hat{\bar{d}}_k[n]$ in (16) when a step-size $\beta = 0.001$ is employed for the following: 1) $N_r = 0$, i.e., no retransmission allowed, and 2) retransmission limit $N_r = 2$, respectively. It is shown that as the number of slots n grows, the average user delays converge. When retransmissions are not allowed, the resultant average delay for user 1 is higher than the prescribed maximum value, i.e., the QoS cannot be guaranteed. On the other hand, allowing $N_r = 2$ retransmissions results in lower average delays for both users compared with the $N_r = 0$ case. The average delay requirements for both users are fulfilled thanks to the retransmission gain incorporated in the proposed scheduling scheme.

Test Case 4: Finally, we suppose that there are $K = 4$ active users. Among them, users 1 and 2 have real-time connections with Bernoulli-distributed arrival processes determined by parameters $\bar{a}_1 = 300$ kb/s, $\pi_1 = 0.6$, $\bar{a}_2 = 300$ kb/s, and

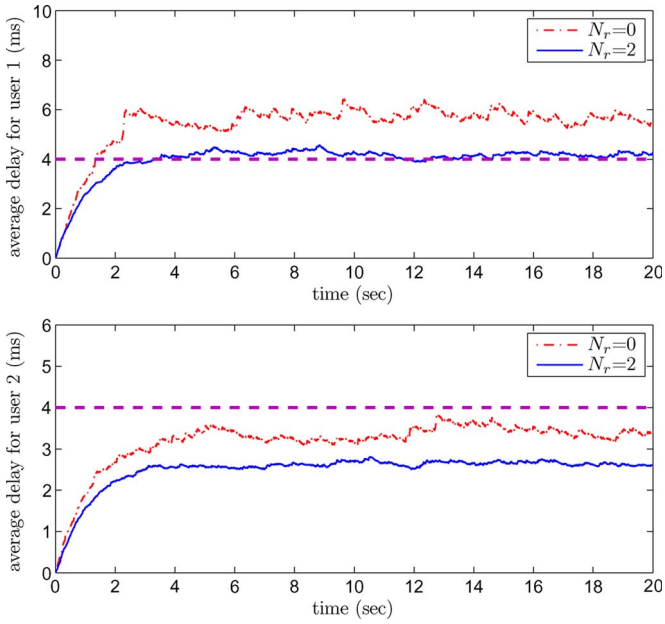


Fig. 5. Average delay evolutions for the proposed real-time scheduling scheme. (Dashed lines indicate $\check{d}_k = 4.0$ ms.)

$\pi_2 = 0.4$, and maximum average delay requirements $\check{d}_1 = \check{d}_2 = 4$ ms. Users 3 and 4 have nonreal-time connections with minimum average rate requirements $\check{r}_3 = \check{r}_4 = 300$ kb/s. For real-time users 1 and 2, we set $U_{rt}(\bar{\boldsymbol{d}}) = \sum_{k=1}^{2}(-\bar{d}_k^2)$, and for nonreal-time users 3 and 4, we set $U_{nrt}(\bar{\boldsymbol{r}}) = \sum_{k=3}^{4} \ln(\bar{r}_k)$. Fig. 6 depicts the evolution of $\hat{\bar{d}}_k[n]$ or $\hat{\bar{r}}_k[n]$ when a step-size $\beta = 0.001$ is employed for the following: 1) $N_r = 0$, i.e., no retransmission allowed, and 2) retransmission limit $N_r = 2$, respectively. Convergence of the average rates and delays is clearly shown. When retransmissions are not allowed, the resultant average delay for user 1 is higher than the prescribed maximum value, and the resultant average delay for user 4 is less than the prescribed minimum value; hence, QoS cannot be guaranteed for these two users. This is because that when the retransmission gain is not exploited, the intended wireless channel cannot afford the requested QoS, i.e., a feasible time allocation satisfying conditions A2) and A4) does not exist. With retransmissions incorporated by allowing $N_r = 2$, the user rate and delay performance is significantly improved. In this case, all the users' QoS are met. Fig. 6 reveals that the proposed stochastic scheme is capable of exploiting the retransmission gain in addition to multiuser diversity for the QoS-guaranteed scheduling of the heterogeneous network traffic.

V. Conclusion

Making use of stochastic optimization tools and principles, we have incorporated retransmissions in multiuser wireless links and described a class of novel scheduling algorithms that are capable of jointly collecting the available multiuser diversity and retransmission gain to enhance the overall network performance. The stochastic scheduling schemes presented are simple to operate, even when the underlying fading channel distribution is unknown *a priori*, and asymptotically guarantee the prescribed fairness and rate/delay requirements for a heterogeneous network traffic. These promising features make the proposed scheduling algorithms an attractive candidate for current and future wireless standards.

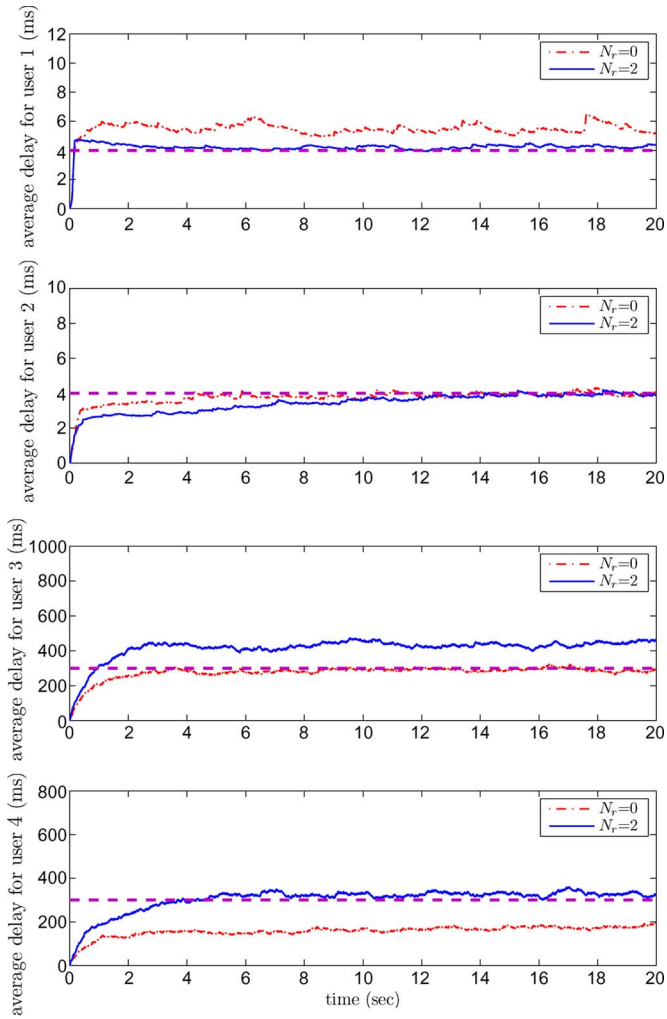


Fig. 6. Average delay and average rate evolutions for the proposed scheduling scheme for heterogeneous traffic. (Dashed lines indicate $\check{d}_k = 4.0$ ms, $k = 1, 2$, or $\check{r}_k = 300$ kb/s, $k = 3, 4$.)

References

[1] *802.11: Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11-1997, 1997. IEEE Std. 802.11 Working Group.

[2] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems (Revision of IEEE Standard 802.16-2001)*, IEEE Std. 802.16-2004, 2004. IEEE Std. 802.16 Working Group.

[3] The 3rd Generation Partnership Project (3GPP) TR 25.848 V4.0.0, *Physical Layer Aspects of UTRA High Speed Downlink Packet Access (Rel. 4)*, 2001.

[4] A. J. Goldsmith and P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, no. 11, pp. 1896–1992, Nov. 1997.

[5] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and A. Viterbi, "CDMA/HDR: A bandwidth efficient high speed wireless data service for nomadic users," *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70–77, Jul. 2000.

[6] H. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

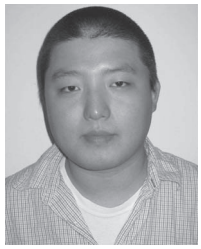
[7] X. Liu, E. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 2053–2064, Oct. 2001.
[8] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *Proc. INFOCOM Conf.*, Miami, FL, Mar. 13–17, 2005, vol. 4, pp. 2415–2424.
[9] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal–dual algorithm," *Queueing Syst.*, vol. 50, no. 4, pp. 401–457, 2005.
[10] G. Song, "Cross-layer resource allocation and scheduling in wireless multicarrier networks," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, Apr. 2005.
[11] S. Shakkottai and A. Stolyar, "Scheduling algorithms for a mixture of real-time and nonreal-time data in HDR," in *Proc. 17th Int. Teletraffic Congr.*, Salvador da Bahia, Brazil, 2001, pp. 793–804.
[12] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, pp. 191–217, 2004.
[13] X. Wang, G. B. Giannakis, and A. G. Marques, "A unified approach to QoS-guaranteed scheduling for channel-adaptive wireless networks," *Proc. IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.
[14] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sep. 2004.
[15] X. Wang, Q. Liu, and G. B. Giannakis, "Analyzing and optimizing adaptive modulation-coding jointly with ARQ for QoS-guaranteed traffic," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 710–720, Mar. 2007.
[16] S. B. Wicker, *Error Control Systems for Digital Communications and Storage*. Englewood Cliffs, NJ: Prentice–Hall, 1995.
[17] X. Wang and G. B. Giannakis, "A stochastic framework for scheduling in wireless packet access networks," in *Proc. IEEE Int. Conf. Commun.*, Glasgow, U.K., Jun. 24–28, 2007, pp. 4052–4057.
[18] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice–Hall, 1995.
[19] L. Kleinrock, *Queueing Systems*, vol. I. New York: Wiley, 1975.
[20] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[22] J. Huang, R. Berry, and M. Honig, "Wireless scheduling with hybrid ARQ," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2801–2810, Nov. 2005.

Xin Wang (M'04) received the B.Sc. and M.Sc. degrees in electrical engineering from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical engineering from Auburn University, Auburn, AL, in 2004.

From September 2004 to August 2006, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. Since September 2006, he has been an Assistant Professor with the Department of Electrical Engineering, Florida Atlantic University, Boca Raton. His research interests include medium access control, cross-layer design, stochastic resource allocation, and signal processing for communication networks.

Irena Li received the B.S. degree in aerospace engineering from the University of Florida, Gainesville, in 2006 and the M.S. degree in electrical engineering from Florida Atlantic University, Boca Raton, in 2008.

From 2007 to 2008, she was a Graduate Research Assistant with Florida Atlantic University, where she worked in the area of wireless communications. She is currently a Space Shuttle Flight Controller with the NASA Johnson Space Center, Houston, TX.



Di Wang received the B.Sc. degree in electrical engineering in 2005 from Heilongjiang University, Harbin, China, and the M.Sc. degree in electrical engineering in 2008 from Florida Atlantic University, Boca Raton, where he is currently working toward the Ph.D. degree with the Department of Electrical Engineering.

His current research interest focuses on stochastic resource allocation.

Hanqi Zhuang (SM'93) received the B.S. degree in engineering from the Shanghai University of Technology (currently Shanghai University), Shanghai, China, in 1981 and the M.S. and Ph.D. degrees in engineering from Florida Atlantic University, Boca Raton, in 1986 and 1989, respectively.

He is currently a Professor of electrical engineering with Florida Atlantic University. His current research interests are in computer vision, robotics, and telecommunications. He has received research grants from various federal agencies and local industries. He has chaired or cochaired 12 Ph.D. committees, published over 50 papers in referred international journals, and given numerous presentations at conferences and institutions. He is currently an Associate Editor of the *International Journal of Computer Applications* and the *International Journal of Biometrics*.

Salvatore D. Morgera (F'90) received the Sc.B. degree (with Honors) in physics, the Sc.M. degree in electrical engineering, and the Ph.D. degree in electrical engineering from Brown University, Providence, RI.

He was a Professor with Concordia University, Montreal, QC, Canada, and a Senior Scientist with the Submarine Signal Division, Raytheon Company, Portsmouth, RI. Prior to joining Florida Atlantic University, Boca Raton, he was a Professor and the Director of the Information Networks and Systems Laboratory, Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada; a major project leader with the Canadian Institute for Telecommunications Research, a Government of Canada Network of Centres of Excellence; the President of the Quebec Research Council, Le Fonds Nature et Technologies; and Special Assistant to the President, Communications Research Center, Industry Canada, Government of Canada. Since 1998, he has been a Professor, the Chair of electrical engineering, and the Director of the Bioengineering Program, Florida Atlantic University. He has more than 40 years of leadership in industry, government, and academia. He has published more than 95 journal papers and 113 conference papers and a book entitled *Digital Signal Processing—Applications to Communications and Algebraic Coding Theories* (Academic). He has conducted research in various aspects of wireless networks, particularly in the areas of quality of service and hybrid automatic repeat request radio link protocols, biometrics for identity management, and bioengineering.

Dr. Morgera is an IEEE Distinguished Lecturer for the Communications Society, Tau Beta Pi Eminent Engineer, Order of Engineers, Professional Engineer, and Vice Chair of the Florida Engineers in Education, Florida Engineering Society.