On the Fairness Delay Trade-off in Wireless Packet Scheduling

Aditya Dua and Nicholas Bambos
Department of Electrical Engineering
Stanford University
350 Serra Mall, Stanford CA 94305
Phone: 650-725-5225
Fax: 650-723-4107
Email: {dua,bambos}@stanford.edu

Abstract: We consider the problem of downlink packet scheduling in a time-slotted wireless communication system when a hybrid automatic repeat reQuest (H-ARQ) re-transmission strategy is adopted. User level fairness and average delay per packet are two important metrics used for evaluating the performance of a scheduling policy. However, these are two competing objectives. A good scheduling policy must be flexible enough so that it can be tuned to trade-off one objective for another. We propose one such scheduling policy characterized by a single parameter that can be varied to capture points on the trade-off curve. Our approach is to study two reduced versions of the original problem and construct delay and fairness optimal scheduling policies based on the optimal solutions to these reduced problems. We then leverage intuition from these optimal policies to design a heuristic scheduler that captures the fairness v/s delay trade-off. We demonstrate the efficacy of the proposed scheduler over benchmark schedulers like round-robin (RR), maximum SNR or C/I and proportional fair (PF) via link level simulations. The proposed policy offers a superior fairness and delay performance and is also low complexity from an implementation perspective.

Index Terms: Packet scheduling, HARQ, HSDPA, Fair queuing, Average delay, Fluid model, Dynamic Programming

On The Fairness Delay Trade-off in Wireless Packet Scheduling

Aditya Dua and Nicholas Bambos
Department of Electrical Engineering, Stanford University
350 Serra Mall, Stanford, CA 94305
Phone: 650-725-5525, Fax: 650-723-4107
Email: {dua,bambos}@stanford.edu

***Abstract*— We consider the problem of downlink packet scheduling in a time-slotted wireless communication system when a hybrid automatic repeat reQuest (H-ARQ) re-transmission strategy is adopted. User level fairness and average delay per packet are two important metrics used for evaluating the performance of a scheduling policy. However, these are two competing objectives. A good scheduling policy must be flexible enough so that it can be tuned to trade-off one objective for another. We propose one such scheduling policy characterized by a single parameter that can be varied to capture points on the trade-off curve. Our approach is to study two reduced versions of the original problem and construct delay and fairness optimal scheduling policies based on the optimal solutions to these reduced problems. We then leverage intuition from these optimal policies to design a heuristic scheduler that captures the fairness v/s delay trade-off. We demonstrate the efficacy of the proposed scheduler over benchmark schedulers like round-robin (RR), maximum SNR or C/I and proportional fair (PF) via link level simulations. The proposed policy offers a superior fairness and delay performance and is also low complexity from an implementation perspective.**

***Index Terms*— Packet scheduling, HARQ, HSDPA, Fair queuing, Average delay, Fluid model, Dynamic Programming**

I. Introduction

The third generation (3G) of wireless cellular communication systems aim to support quality-of-service (QoS) intensive services like interactive multimedia and high-speed data [1]. However, bandwidth constraints, power constraints and the unpredictable nature of the wireless channel make this a challenging task for wireless system designers.

Packet scheduling is an important component of the high-speed downlink packet access (HSDPA) concept that has been introduced in the 3GPP Release 5 specifications to provide high data rates and QoS intensive services on the downlink [2]. In addition to fast physical layer re-transmissions, hybrid automatic repeat reQuest (H-ARQ) is also employed in HSDPA to enhance link layer performance [3].

Two important metrics can be used to evaluate the performance of a scheduling policy, namely, average delay experienced per packet and user level fairness [5]. However, these are two competing objectives. A good scheduling policy should be flexible enough so that it can be tuned to trade-off one objective for the other. A packet scheduler typically has access to both queue backlog state and channel state information (CSI) for all downlink users. A “good” scheduler that performs well with respect to both metrics must appropriately incorporate both pieces of information into its scheduling decision. For instance, the maximum SNR or C/I scheduler utilizes only CSI and performs poorly with regard to fairness.

Our goal in this paper is to construct an easily implementable and flexible scheduling policy that can be tuned to achieve a desired fairness v/s delay trade-off. Our approach is to first study two simplified or reduced versions of the scheduling problem. We find the optimal solutions to these reduced problems. They respectively correspond to delay optimal and fairness optimal scheduling policies. We then leverage intuition gained from the structure of these optimal policies to construct a heuristic solution for the original scheduling problem. The result is a family of scheduling policies indexed by a single parameter that can be varied to achieve a desired operating point on the fairness v/s delay trade-off curve. Delay optimal policies for wireless scheduling with H-ARQ were also studied in [4]. However, the authors in [4] do not study fair scheduling.

We study scheduler design in the context of a “buffer-draining” problem. This is a useful model to study with regard to applications like file transfer [6]. The authors in [6] propose a scheduler that jointly uses CSI and file size information. However, they do not consider H-ARQ.

We formally set up the scheduling problem in Section II. We study two reduced versions of the problem and their optimal solutions in Sections III and IV respectively. In Section V we propose heuristic scheduling policies based on the optimal solutions of the reductions. We present performance results based on link-level simulations in Section VI. We provide concluding remarks in Section VII.

II. Problem Definition

We consider a mobile wireless cellular communication system in which time is slotted into fixed size slots on the downlink. There is one queue corresponding to each mobile station (MS) at the base station (BS). In every time slot, the BS schedules a packet from a non-empty queue for transmission. Packets in a queue are served in accordance with the first-come first-served (FCFS) discipline. A H-ARQ based transmission strategy is employed for each queue. We consider a scenario where mobile stations are downloading files from the BS. Each file download corresponds to one “session”. The entire file is assumed available at the BS at the beginning of a session. Equivalently, we are interested in the buffer draining problem. The downlink channel for each MS is a spatially and temporally varying random process. A feedback channel from the MS to the BS provides delayed channel state information (CSI) and acknowledgments of successful/failed (ACK/NACK) packets.

Our goal is to design scheduling policies for the BS that optimize two performance metric(s), namely, *delay* and *fairness*. In our context, delay refers to the average delay per packet averaged over all queues, that is, the average length of a session. While minimizing delay is tantamount to effective use of system resources, optimizing for user level fairness is desirable from a social perspective. Clearly, the two objectives compete with each other.

The scheduling problem in the form posed here is an extremely hard one to solve. We study two reductions of the original model that preserve the essence of the problem and are yet amenable to analysis.

III. Reduction I (Delay Optimal)

Consider two queues, $\mathcal{Q}_1$ and $\mathcal{Q}_2$, being served by a single server (transmitter) $\mathcal{S}$. In each slot, $\mathcal{S}$ schedules a packet from one of the queues for transmission. For simplicity, let us assume that each queue has *exactly* one packet. In other words, we only focus on the head-of-line (HOL) packet of each queue. We model the downlink channels as discrete-time binary-valued $(0-1)$ independent and identically distributed (i.i.d) stochastic processes, independent of each other. The probability of successful transmission on the first attempt is given by $0 \leq s_i \leq 1$ for $\mathcal{Q}_i$. Due to a H-ARQ based transmission strategy, the success probability on subsequent attempts is a function of s_i and the observed SNR on prior failed attempts. The maximum number of transmission attempts for a packet in any queue is $D > 1$. A buffering cost of $c_i > 0$ per slot is incurred for the HOL packet of $\mathcal{Q}_i$. Our goal is to choose a work conserving non-clairvoyant scheduling policy that minimizes the total expected buffering cost. The problem described above is amenable to solution via the methodology of dynamic programming (DP) [7].

Definition: Let n_i be the number of remaining transmission attempts for the HOL packet of $\mathcal{Q}_i$, $i = 1, 2$, $0 \leq n_i \leq D$.

Definition: Let the two-tuple (n_1, n_2) be the **state** of the system.

The system dynamics (under any candidate scheduling policy) are as follows: In state (n_1, n_2), if $\mathcal{Q}_1$ is scheduled, the system moves to state $(0, n_2)$ if the transmission is successful and to state $(n_1 - 1, n_2)$ else. If $\mathcal{Q}_2$ is scheduled, the system moves to state $(n_1, 0)$ if the transmission is successful and to state $(n_1, n_2 - 1)$ else. The initial state is (D, D), and the final state is $(0, 0)$.

Definition: Let $s_i(n_i)$ denote the probability of successful transmission (if scheduled) of the HOL packet of $\mathcal{Q}_i$ when it has n_i transmission attempts remaining.

Definition: Let $V(n_1, n_2)$ be the cost-to-go function in state (n_1, n_2).

Thus, $V(n_1, n_2)$ is the total expected buffering cost starting from state (n_1, n_2) and using the optimal scheduler. For $n_1, n_2 > 0$, we have the following DP recursion,

$$V(n_1, n_2) = \min\{\alpha(n_1, n_2), \beta(n_1, n_2)\} + c_1 + c_2, \quad (1)$$

where

$$\begin{aligned} \alpha(n_1, n_2) &= s_1(n_1)V(0, n_2) + [1 - s_1(n_1)]V(n_1 - 1, n_2) \\ \beta(n_1, n_2) &= s_2(n_2)V(n_1, 0) + [1 - s_2(n_2)]V(n_1, n_2 - 1). \end{aligned}$$

Also, we have the boundary conditions:

$$\begin{aligned} V(n_1, 0) &= [1 - s_1(n_1)]V(n_1 - 1, 0) + c_1,\ n_1 > 0 \\ V(0, n_2) &= [1 - s_2(n_2)]V(0, n_2 - 1) + c_2,\ n_2 > 0 \\ V(0, 0) &= 0. \end{aligned} \quad (2)$$

Thus, it is optimal to schedule $\mathcal{Q}_1$ in state (n_1, n_2) if $\alpha(n_1, n_2) \leq \beta(n_1, n_2)$ and $\mathcal{Q}_2$ else.

Definition: Let $\tau_i(j)$ denote the expected number of slots required to successfully transmit (or drop) the HOL packet of $\mathcal{Q}_i$ $(i = 1, 2)$, with j remaining transmission attempts $(1 \leq j \leq D)$ and counting only the slots in which $\mathcal{Q}_i$ is scheduled.

By definition $\tau_i(1) = 1$. We denote the expected remaining transmission times in state (n_1, n_2) as $\tau_1(n_1)$ and $\tau_2(n_2)$, for $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively. Under static channel conditions, it is reasonable to assume that the success probability increases with each attempt, that is, $\{s_i(n); n \geq 1\}$ is a monotone decreasing sequence in $n\ \forall\ i$. With this assumption, we have the following lemma.

Lemma 1: The sequence $\{\tau_i(n); n \geq 1\}$ is a monotone non-decreasing sequence in $n\ \forall\ i$.

Sketch of Proof: See the Appendix. ■

We now state a key theorem regarding the optimal solution of the DP in (1). The proof of the theorem relies on Lemma 1.

Theorem 1: In state (n_1, n_2) it is optimal to schedule $\mathcal{Q}_1$ if $\frac{\tau_1(n_1)}{c_1} \leq \frac{\tau_2(n_2)}{c_2}$, else it is optimal to schedule $\mathcal{Q}_2$. Also, if it is optimal to schedule $\mathcal{Q}_1$ in state (n_1, n_2), $V(n_1, n_2) = (c_1 + c_2)\tau_1(n_1) + c_2\tau_2(n_2)$ else $V(n_1, n_2) = (c_1 + c_2)\tau_2(n_2) + c_1\tau_1(n_1)$.

Sketch of Proof: See the Appendix. ■

Thus, the optimal scheduling policy[1] is an *index policy* with $\tau_i(n_i)/c_i$ being the index of $\mathcal{Q}_i$. For the special case of $c_i = 1\ \forall\ i$, the optimal policy reduces to the *shortest expected remaining transmission time* policy. Note that this case corresponds to minimizing the average delay over both queues, which was indeed our design objective.

[1] We have presented Theorem 1 for the simple case of two queues with one packet each and an i.i.d channel. While this simple form captures the essence of the optimal scheduler, the scope of the theorem can be extended significantly by incorporating multiple queues, multiple buffered packets and a Markovian channel model. We plan to present extensions in the journal version of the paper.

IV. Reduction II (Fairness Optimal)

The second model reduction deals with the notion of fair scheduler design. Consider two queues $\mathcal{Q}_1$ and $\mathcal{Q}_2$ being served by a single server (transmitter) $\mathcal{S}$. The queues have P_1 and P_2 packets at the start of their respective sessions. The system is time-slotted. In each slot, S schedules one non-empty queue for transmission. We denote by T_i the time (measured in slots) at which the last packet in $\mathcal{Q}_i$ departs. Equivalently, T_i is the length of the session for $\mathcal{Q}_i$. The average delay experienced per packet in $\mathcal{Q}_i$ is denoted by $\psi_i = T_i/P_i$. We define the sessions to be *fair*[2] if both queues experience the same average delay per packet, i.e., $\psi_1 = \psi_2$. We assume ideal channels for now, that is, a packet transmission is successful with probability 1.

A. A Fluid Model

We first study session fair continuous time scheduling in the context of a buffer-draining problem for a fluid queuing model and then establish an equivalence with a discrete time system. Consider a fluid model with two queues being served by a single server that produces work at unit rate. At any time $t \geq 0$, the server serves $\mathcal{Q}_1$ with rate $\beta(t) \in [0, 1]$ and $\mathcal{Q}_2$ at rate $[1 - \beta(t)]$. For a packet size of $L > 0$ the queues have workload $P_1 L$ and $P_2 L$ respectively at $t = 0$. Assume $P_1 > P_2$. The continuous time scheduling policy that achieves session fairness in this fluid model is given by the following lemma:

Lemma 2: For $\mathcal{Q}_i$, define the time-varying quantity $\eta_i(t) = \dfrac{\psi P_i L - t}{W_i(t)}$, where $\psi = 1 + \dfrac{P_2}{P_1}$ and $W_i(t)$ is the workload in $\mathcal{Q}_i$ at time t. Then, a scheduling policy Π^c that allocates the server share such that $\eta_1(t) = \eta_2(t) \; \forall \; t$ such that both queues are non-empty in $[0, t)$ achieves session fairness.

Sketch of Proof: See the Appendix. ■

We now adapt the continuous time policy Π^c to our discrete time slotted system to obtain a session fair scheduler. The following theorem formalizes the result:

Theorem 2: For $\mathcal{Q}_i$, define the time-varying quantity $\eta_i(k) = \dfrac{\psi P_i - k}{Q_i(k)}$, where $\psi = 1 + \dfrac{P_2}{P_1}$, k is the time slot index and $Q_i(k)$ is the number of packets in $\mathcal{Q}_i$ in the k^{th} slot. Then, a scheduling policy Π that schedules $\mathcal{Q}_{\Pi(k)}$ in the k^{th} TTI such that $\Pi(k) = \underset{j}{\operatorname{argmin}} \, \eta_j(k)$ achieves session fairness[3].

Sketch of Proof: See the Appendix. ■

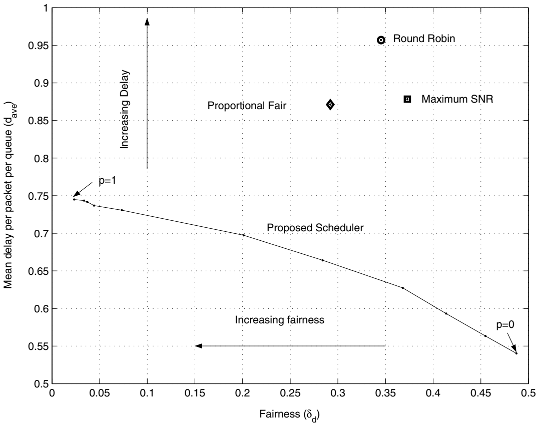

Fig. 1. The fairness v/s delay trade-off: d_{ave} v/s δ_d for Round Robin, maximum SNR, proportional fair and proposed schedulers

Intuitively, the slotted time scheme works because $\eta_i(k)$ increases if $\mathcal{Q}_i(k)$ is scheduled in the k^{th} TTI and decreases otherwise. Since in each TTI the scheme schedules the queue with the smallest η, the η_is of the time-slotted system closely track the η_is of the fluid model. Fairness in the fluid model therefore leads to fairness in the time-slotted model.

V. Heuristic Scheduling Policies

In this section we leverage intuition gained from the optimal solutions to the reduced models studied in Sections III and IV to design heuristic scheduling policies for the original system.

A. Delay aware scheduling

Theorem 1 (and its extensions) established that average delay is minimized by a shortest expected remaining transmission time policy. If a queue has Q packets and the expected remaining transmission time for its HOL packet is τ, a reasonable estimate for the expected remaining transmission time for all packets in the queue is $Q\tau$. We then propose the following scheduling policy:

Policy 1 ($\mathbf{\Pi}^d$*):* Let $\mathcal{S}(k)$ be the set of non-empty queues, and $Q_j(k)$ and $\tau_j(k)$ respectively be the queue size and expected remaining transmission time for the HOL packet of the j^{th} queue ($\mathcal{Q}_j \in \mathcal{S}(k)$) in the k^{th} time slot. The scheduling policy Π^d schedules $\mathcal{Q}_{\Pi^d(k)}$ in the k^{th} slot, such that $\Pi^d(k) = \underset{j \in \mathcal{S}(k)}{\operatorname{argmin}} \, Q_j(k)\tau_j(k)$.

We now present the H-ARQ re-transmission model and outline the computation of τ_i, the expected remaining transmission time for the HOL packet of $\mathcal{Q}_i$.

1) H-ARQ Model: We briefly describe the model used for chase-combining based H-ARQ [3]. The total received SNR for a packet after the n^{th} transmission (assuming previous n-1 attempts were unsuccessful) can be modeled as

$$\gamma_{\text{tot}}(n) = \epsilon^{n-1} \eta(n) \sum_{k=1}^{n} \gamma_k, \tag{3}$$

where γ_k is the received SNR on the k^{th} unsuccessful attempt, $0 < \epsilon < 1$, $\eta(1) = 1$, $\eta(n) = \eta > 1 \; \forall \; n > 1$. Here η

[2] It is clear that not every scheduling policy will achieve session fairness. For example, consider RR scheduling with $P_1 = 20, P_2 = 10$. If we start with $\mathcal{Q}_1$, we get $T_1 = 30, T_2 = 20$. Thus, $\psi_1 = 1.5 \neq 2 = \psi_2$. For the shortest expected remaining time (SERT) scheduler we would have $T_1 = 30, T_2 = 10$, or $\psi_1 = 1.5 \neq 1 = \psi_2$. Thus, both policies are unfair. For a scheduler that achieves session fairness, we have $\psi_1 = \psi_2 = \psi$. This gives $T_1 = 30, T_2 = 15$.

[3] As was the case with Theorem 1, we present Theorem 2 in a simple form owing to space constraints. Yet, the simple form captures the essence of our argument, which is to demonstrate a coupling between the continuous and discrete time systems. We plan to present a more general theorem in the journal version of the paper.

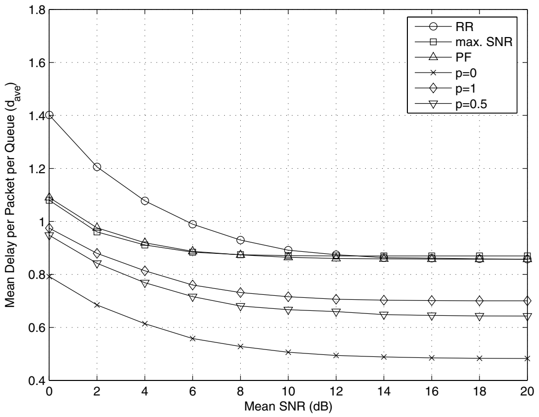

Fig. 2. Delay performance as a function of mean SNR: d_{ave} v/s mean received SNR (dB) for Round Robin, maximum SNR, proportional fair and proposed schedulers

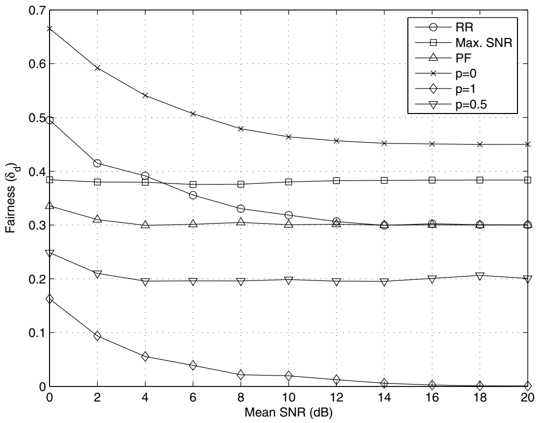

Fig. 3. Fairness performance as a function of mean SNR: δ_d v/s mean received SNR (dB) for Round Robin, maximum SNR, proportional fair and proposed schedulers

and ϵ respectively denote the incremental redundancy gain and chase-combining efficiency.

2) Packet Success Probability and Expected Transmission Time: Consider an HOL packet that has $m > 1$ remaining transmission attempts, given that the maximum permissible number of transmission attempts is D. The total received SNR for the packet on the next transmission is denoted by $\gamma_{\text{tot}}(D - m + 1)$. We map total received SNR γ to the probability of successfully decoding a packet via a monotone increasing function $s(\gamma) : [0, \infty) \mapsto [0, 1]$. Under the assumption that channel conditions do not change from the current instant till the packet has been successfully transmitted or the maximum permissible transmission attempts have been exhausted, we can compute the total received SNR upon each transmission attempt using (3). Finally, we use the recursion in (4) [see Appendix] to compute $\tau(m)$.

B. Fairness aware scheduling

In Section IV, we studied a scheduling policy that achieves session fairness under error-free channel conditions. As an extension, we propose the following policy:

Policy 2 ($\mathbf{\Pi}^f$): Let $\mathcal{S}(k)$ and $\mathcal{S}^c(k)$ respectively denote the set of non-empty and empty queues in the k^{th} time slot. Define the time-varying parameter $\psi(k) = \dfrac{\sum_{j \in \mathcal{S}(k)} P_j \tau_j + \sum_{j \in \mathcal{S}^c(k)} P_j - k}{\max_j P_j}$. Now, for $\mathcal{Q}_i \in \mathcal{S}(k)$ define the time-varying quantity $\eta_i(k) = \dfrac{\psi(k) P_i - k}{Q_i(k)}$. The quantities $Q_j(k)$ and $\tau_j(k)$ are as defined in Policy 1, and $P_j = Q_j(0)$. The scheduling scheme Π^f schedules $\mathcal{Q}_{\Pi^f(k)}$ in the k^{th} TTI, such that $\Pi^f(k) = \underset{j \in \mathcal{S}(k)}{\text{argmin}}\, \eta_j(k)$.

Note that for a 2 queue system with error-free channels ($\tau_j = 1$), Policy 2 reduces to the optimal policy in Theorem 2.

C. Fairness-Delay Trade-off

We now propose a scheduling policy[4], characterized by a single parameter $p \in [0, 1]$, that provides the desired trade-off between fairness and delay based on the choice of p.

Policy 3 ($\mathbf{\Pi}_p^{fd}$): The scheduling policy Π_p^{fd} schedules $Q_{\Pi_p^{fd}(k)}$ in the k^{th} time slot such that

$$Q_{\Pi_p^{fd}(k)} = \begin{cases} Q_{\Pi^f(k)} & \text{with probability } p \\ Q_{\Pi^d(k)} & \text{with probability } (1 - p). \end{cases}$$

VI. Simulation Results

We evaluate the performance of the proposed schedulers via link-level simulations. We consider a downlink scenario with four mobile receivers moving at 3kmph. We assume the ITU PEDA path profile for each MS, with the same average SNR. We model the success probability function as $s(\gamma) = 1 - e^{-\delta\gamma}$ with $\delta = 0.9$. The H-ARQ parameters η and ϵ are set to 1.1 and 0.95 respectively. A maximum of four re-transmissions is permitted per packet.

Suppose $\mathcal{Q}_i$ has P_i packets in its buffer at the start of a session and the last packet in $\mathcal{Q}_i$ departs in the T_i^{th} slot. Then $d_i = T_i/P_i$ is the mean delay per packet for $\mathcal{Q}_i$. We use $d_{\text{ave}} = \dfrac{1}{K}\sum_{k=1}^{K} d_k$ as a measure of average delay, and $\delta_d = \dfrac{1}{K}\left(\max\limits_{1 \le k \le K} d_k - \min\limits_{1 \le k \le K} d_k\right)$ as a measure of user level fairness.

Figure 1 depicts the fairness v/s delay trade-off curve for the proposed scheduling policy, generated by varying the parameter p. The figure also contrasts various points on the trade-off

[4] The proposed scheduling policies are quite practical and low complexity from an implementation perspective. The BS needs the following information (for each queue) to make its scheduling decision: current downlink channel state, outcomes of prior transmission(s) for the HOL packet and the current backlog state. While the latter is directly accessible to the BS, the former two can be attained via feedback from the MS. This is not any different from requirements of benchmark schedulers.

curve with benchmark schedulers. Figures 2 and 3 respectively depict the delay and fairness performance of the proposed scheduler (for $p = 0, 0.5, 1$) and other benchmark schedulers as a function of the mean received SNR. As expected, $p = 0$ and $p = 1$ provide the best delay and fairness performance respectively. A choice of $p = 0.5$ provides a good fairness-delay trade-off. Amongst the benchmark schedulers, maximum SNR provides the best delay performance while PF provides the best fairness performance. Overall, PF provides the best fairness v/s delay trade-off amongst the three, as corroborated by Figure 1. The three benchmarks become nearly identical at very high SNRs.

VII. Conclusions

In this paper, we investigated the problem of downlink packet scheduling in a time slotted wireless cellular system in the presence of a H-ARQ re-transmission strategy. In particular, we explored the fundamental fairness v/s delay trade-off inherent in the scheduling problem and proposed a flexible scheduling policy that can be tuned to trade-off one desired objective for the other. We demonstrated via simulations that the proposed policy outperforms standard benchmark schedulers in terms of both fairness and delay and is quite practical from an implementation perspective. Our on going research involves studying similar trade-offs involved in scheduling of real-time packets with stringent deadline constraints.

References

[1] H. Holma and A. Toskala, Eds., *WCDMA for UMTS*, John Wiley & Sons, 3rd. Ed., 2002.

[2] 3GPP, "High Speed Downlink Packet Access (HSDPA); overall description", 3GPP, Sophia Antipolis, France, Technical Specification 25.308, Ver. 5.4.0, Release 5, Mar. 2002.

[3] F. Frederiksen and T.E. Kolding, "Performance and modeling of WCDMA/HSDPA transmission/H-ARQ schemes", *Proc.IEEE VTC*, Vancouver, British Columbia, Cananda, Sep. 24-29, 2002, pp. 472-476.

[4] J. Huang, R. Berry and M. Honig, "Wireless scheduling with hybrid ARQ", *IEEE Trans. Wireless Commun.*, to appear.

[5] S. Lu, V. Bhargavan and R. Srikant, "Fair scheduling in wireless packet networks", *IEEE/ACM Trans. Networking*, vol. 7, no. 4, Aug. 1999, pp. 76-83.

[6] M. Hu, J. Zhang and J. Sadowsky, "Traffic aided opportunistic scheduling for downlink transmissions: algorithms and performance bounds", *Proc. IEEE INFOCOM*, Hong Kong, Mar. 7-11, 2004, pp. 1652-1661.

[7] J. Walrand, *An Introduction to Queueing Networks*, Prentice Hall, Englewood Cliffs, NJ, 1988.

VIII. Appendix

A. Sketch of Proof of Lemma 1

By conditioning on the outcome of a transmission with k transmission attempts remaining, we get the recursion

$$\tau_i(k) = s_i(k)\cdot 1 + [1 - s_i(k)](\tau_i(k-1)+1),\ k = D, \ldots, 2, \quad (4)$$

with $\tau_i(1) = 1$. This gives $\tau_i(2) = 2 - s_i(2) \geq 1 = \tau_i(1)$, since $s_i(2) \leq 1$. Assume $\tau_i(k) \geq \tau_i(k-1)$, for some $k > 2$. Then, we have $\tau_i(k+1) = 1 + [1 - s_i(k+1)]\tau_i(k) \geq 1 + [1 - s_i(k)]\tau_i(k) \geq 1 + [1 - s_i(k)]\tau_i(k-1) = \tau_i(k)$. The result follows from the principle of mathematical induction.

B. Sketch of Proof of Theorem 1

We prove the theorem via the principle of mathematical induction. We omit the base case due to space constraints.

Inductive Step: Suppose the theorem is true in states (n_1, n_2) and (n_1+1, n_2-1), for $1 < n_1, n_2 < D$. We show that the theorem holds in state (n_1+1, n_2). We divide the proof into 4 cases, depending on whether $\mathcal{Q}_1$ or $\mathcal{Q}_2$ is optimal in state (n_1, n_2) and (n_1+1, n_2-1).

1) *Case 1:* $\mathcal{Q}_1$ is optimal in states (n_1, n_2) and (n_1+1, n_2-1). In this case it is always optimal to schedule $\mathcal{Q}_1$ in state (n_1+1, n_2). Thus, $V(n_1+1, n_2) = \alpha(n_1+1, n_2) + c_1 + c_2 = (c_1 + c_2)\tau_1(n_1+1) + c_2\tau_2(n_2)$.
2) *Case 2:* $\mathcal{Q}_1$ is optimal in state (n_1, n_2) and $\mathcal{Q}_2$ is optimal in state (n_1+1, n_2-1). In this case it is optimal to schedule $\mathcal{Q}_1$ in state (n_1+1, n_2) if $\frac{\tau_1(n_1+1)}{c_1} \leq \frac{\tau_2(n_2}{c_2}$, and $\mathcal{Q}_2$ else. Also, $V(n_1+1, n_2) = (c_1+c_2)\tau_1(n_1+1) + c_2\tau_2(n_2)$ in the former case, and $V(n_1+1, n_2) = (c_1+c_2)\tau_2(n_2) + c_1\tau_1(n_1+1)$ in the latter case.
3) *Case 3:* $\mathcal{Q}_2$ is optimal in state (n_1, n_2) and $\mathcal{Q}_1$ is optimal in state (n_1+1, n_2-1). This case contradicts Lemma 1 and therefore does not arise.
4) *Case 4:* $\mathcal{Q}_2$ is optimal in states (n_1, n_2) and (n_1+1, n_2-1). In this case it is always optimal to schedule $\mathcal{Q}_2$ in state (n_1+1, n_2). Thus, $V(n_1+1, n_2) = \beta(n_1+1, n_2) + c_1 + c_2 = c_1\tau_1(n_1+1) + (c_1+c_2)\tau_2(n_2)$.

We combine the results of the four mutually exhaustive cases and invoke the principle of mathematical induction to claim that the theorem holds.

C. Sketch of Proof of Lemma 2

Since the server produces work at unit rate, we have $\dot{W}_1(t) + \dot{W}_2(t) = -1$, where $\dot{x}(t)$ denotes the time derivative of $x(t)$. Integrating over the interval $[0, t]$ yields $W_1(t) + W_2(t) = P_1L + P_2L - t$. Now, we use this along with the definition of $\eta_1(t)$, $\eta_2(t)$ to get

$$W_2(t) = \frac{(\psi P_1 L - t)(\psi P_2 L - t)}{(\psi P_1 L - t) + (\psi P_2 L - t)}.$$

$W_2(t) = 0 \Rightarrow t = T_2 = \psi P_2 L$. This holds because the first term in the numerator and the denominator are strictly positive for $t < \psi P_2$. The total work at $t = 0$ is $P_1L + P_2L = \psi P_1 L$. Thus, it takes time $\psi P_1 L$ to empty both queues. This gives $T_1 = \psi P_1 L$. Thus, $T_1/P_1 = T_2/P_2 = \psi$ and the scheme achieves session fairness.

D. Sketch of Proof of Theorem 2

The proof is based on the following result: If the workload in $\mathcal{Q}_i$ in the fluid model at time $t = kL$ $(k = 0, 1, 2, \ldots)$ is $W_i(kL)$, the number of packets in $\mathcal{Q}_i$ in the corresponding discrete time model after k TTIs, $Q_i(k)$ is either $\lfloor W_i(kL)/L \rfloor$ or $\lceil W_i(kL)/L \rceil$. Since $W_1(kL) + W_2(kL) = \psi P_1 L - kL$ and $Q_1(k) + Q_2(k) = P_1 + P_2 - k$, if $Q_1(k) = \lfloor W_1(kL)/L \rfloor$, then $Q_2(k) = \lceil W_2(kL)/L \rceil$, and vice-versa. The result can be established via induction. The theorem directly follows from this result.