Scheduling in Wireless Cellular Networks Under Probabilistic Channel Information

Aytac Azgin and Marwan Krunz
Department of ECE, The University of Arizona
{aytaca,krunz}@ece.arizona.edu

Abstract— **As wireless networks continue to grow, users will be demanding more service diversity and differential quality-of-service (QoS). At the packet level, differential QoS is met by means of scheduling for channel access. Several algorithms have been proposed for packet scheduling over the wireless link. The main consideration in the design of these algorithms is the characteristics of the wireless channel. Previous works used a two-state Markov model to characterize the channel and design the scheduling algorithm. This coarse approximation can lead to a considerable amount of inaccuracy in the expected user performance, i.e., when implemented under realistic channel conditions the algorithm may not perform as expected. In this paper, we propose a scheduling algorithm and an associated MAC scheme that enable operation under realistic channel conditions. We use an N-state Markov model to characterize the channel, where $N > 2$. The channel characteristics are incorporated into the scheduler through the use of future channel estimates, making the scheduler more immune to channel variations. Our scheme allows for *adaptive* FEC, whereby the code rate varies according to the forecasted channel state. Comparisons between the proposed algorithm and Wireless Fair Service (WFS) [6] show that the proposed algorithm gives better results in terms of both throughput and delay, while preserving the fairness characteristics. It is also shown that the proposed algorithm is more stable under variations in mobility and network load. Furthermore, by being able to significantly reduce the number of retransmissions, the proposed algorithm makes better utilization of the channel bandwidth and reduces the energy consumed in delivering a packet.**

I. Introduction

In recent years, wireless communications have witnessed a tremendous growth. For this upward trend to continue, greater service diversity and better price/performance differentiation are needed. The challenge in providing different levels of service is how to provide guarantees while maximizing the usable system capacity. Service provisioning must also account for fairness considerations. This is especially true in the wireless environment, where channel errors are location dependent, and can result in diverse throughput and delay characteristics.

At the packet level, service differentiation is typically accomplished by means of scheduling. Scheduling in wireline networks is relatively easy, and it has been extensively studied in the literature (see [10] for a survey). In contrast, scheduling in wireless networks is quite challenging for a number of reasons [6]. First, the available channel capacity perceived by a given user varies depending on channel fading, user mobility, and the obstacles between the transmitter and the receiver. Second, errors over the wireless channel are bursty (i.e., highly correlated) and location dependent. This makes it more difficult to provide delay guarantees. Third, channel access is more complicated due to the presence of the hidden and exposed terminal problems. Finally, mobile users are often power constrained, making it necessary to optimize every bit of the available bandwidth and calling for efficient scheduling schemes that achieve this goal.

This work was supported by NSF under grants CCR 9979310 and ANI 0095626, and by the Center for Low Power Electronics (CLPE) at the University of Arizona. CLPE is supported by NSF (grant # EEC-9523338), the State of Arizona, and a consortium of industrial partners.

An ideal wireless scheduling scheme should provide [6]: (1) short-term throughput bounds for *error-free* flows, (2) long-term throughput bounds for *bounded-error* flows, (3) short-term fairness among *error-free* flows, (4) long-term fairness among *bounded-error* flows, and (5) delay bounds for *bounded-error* flows. A flow is said to be *error-free* if it perceives a "clean" channel at a given time instant. A *bounded-error* flow is one whose fading periods are time limited to allow for subsequent service compensation without disturbing the services given to other flows.

Several wireless scheduling algorithms were previously proposed to achieve some/all of the above objectives. Examples of such schemes can be found in [1, 5, 6, 7, 8]. Among these schemes, the Channel Condition Independent Packet Fair Queueing (CIF-Q) [8], and the Wireless Fair Service (WFS) [6] schemes received special attention as they are closest to meeting the above ideal goals. Both of these schemes achieve good performance in terms of fairness, throughput, and delay guarantees. They both rely on an error-free service model, i.e., a wireline scheduler, to achieve fairness; the Start Time Fair Queueing (STFQ) for CIF-Q and the Weighted Fair Queueing (WFQ) for WFS. A common characteristic of STFQ and WFQ is their use of a per-flow tagging mechanism to tag packets. The scheduler selects the head-of-line (HOL) packet of the flow with the smallest tag. Details of such a tagging mechanism are provided in Section 3.

To account for the channel's behavior, both CIF-Q and WFS associate *leading* and *lagging* counters with each flow. These counters are used to account for "lost" and "gained" services. If a flow that was originally scheduled for service (i.e., packet transmission) during the current time slot is *believed* to be experiencing deep fade at the receiving end, then its slot is assigned to another flow that is likely to experience a clean channel during this slot. As a result of exchanging the transmission slots, the lagging (leading) counter of the deferring flow is incremented (decremented) while the leading (lagging) counter of the other flow that received the service is incremented (decremented). The leading and the lagging counters are used for future compensation of lagging flows by reclaiming the transmission slots from leading flows. Both CIF-Q and WFS allow leading flows to gracefully relinquish their lead to lagging flows to provide short-term guarantees (the relinquishing rate depends on the amount of the lead).

Previously proposed scheduling algorithms, including CIF-Q and WFS, rely on the so-called Gilbert-Elliot (GE) model [2] to predict the state of the channel. The GE model is a two-state Markov model, where in one state the channel is extremely

"bad" (bit error rate (BER) is 0.5) and in the other state the channel is very "good" (BER is zero). In practice, the channel state is determined by the signal-to-noise ratio (SNR) at the receiver, and this SNR takes a continuous range of values. Discretizing it into two regions that correspond to two states is a rather crude approximation. In [9] the authors used information theoretic arguments to show that the GE model minimizes the channel capacity. By employing a specific SNR partitioning approach, the authors in [3] showed that the *wireless effective bandwidth* increases dramatically as the number of states of the employed Markov model is increased. In other words, the 2-state GE model leads to a highly conservative allocation strategy. We believe that the performance of wireless scheduling algorithms can be significantly improved if they are designed under an N-state channel model, where $N > 2$. This claim is substantiated in the results presented in this paper.

Replacing the GE channel model with a higher-order model has important ramifications. Previous scheduling schemes that employed the GE model relied mainly on error detection and packet retransmissions to recover from bad channels. Forward error correction (FEC) was not directly used (it was incorporated into the channel parameters). On the other hand, if a higher-order Markov model is to be employed, the scheduling scheme can take advantage of *adaptive* FEC to recover from *partial* channel errors (BER is greater than zero but less than 0.5). As shown in this paper, integrating higher-order channel prediction ($N > 2$) and adaptive FEC into the design of a wireless scheduling algorithm results in significant improvements in the throughput and packet delay. Another advantage of adaptive FEC is that it can provide a tradeoff between delay and throughput performance, thus providing more flexibility in achieving service differentiation.

The goal of this paper is to introduce a scheduling algorithm and its associated MAC scheme that enable operation under adaptive FEC and multi-state channel predictions. The proposed algorithm is based on computing a flow-specific parameter that combines the flow's actual service rate (i.e., rate taking into account the compensation parameters) with the future channel state estimates. Comparisons of the proposed algorithm with WFS (which already fulfills all the requirements of a wireless fair scheduler with very good throughput and delay results) show that by incorporating probabilistic channel state information into the scheduling mechanism, we can achieve good improvement in performance (i.e., delay and throughput) while preserving the fairness characteristics of WFS. We show that the proposed algorithm is more stable in the presence of mobility and network load variations. Furthermore, it results in fewer retransmissions, which makes it more scalable and more power efficient for mobile terminals.

The rest of the paper is structured as follows. In Section 2, we describe the channel model and the access scheme. Section 3 discusses the proposed scheduling algorithm. We present our simulation results in Section 4. Section 5 concludes the paper.

II. Framework

We consider a cellular wireless network. We assume that a single channel (frequency band) is used for both uplink and downlink flows, and also for data and control packets (hence, only one transmission is allowed at a time). Without loss of generality, we focus on the downlink case, which is typically the case of interest from a scheduling standpoint. A TDMA-like scheme is used for accessing the channel. Before transmitting a packet from the base station to a mobile user, an RTS/CTS handshake must first take place. Variable-size network-layer packets are fragmented into smaller fixed-size link-layer (LL) packets (before adding the FEC bits).

A. Channel Model

A slowly varying Rayleigh fading channel is assumed. The state of the channel at a given mobile node is characterized by an N-state discrete-time Markov chain with a transition probability matrix $\boldsymbol{P} = [p_{ij}]_{i,j\in\mathcal{L}}$, where $\mathcal{L} = \{0, 1, \ldots, N-1\}$. This matrix completely defines the wireless channel. We assume that $\boldsymbol{P}$ does not change frequently, and stays constant for the duration of a connection. Without loss of generality, we assume that $\boldsymbol{P}$ is the same for all mobile nodes. At a mobile station, a nominal BER is associated with each channel state and can be calculated as follows [9]:

$$e_k = \frac{\int_{\Gamma_k}^{\Gamma_{k+1}} \frac{1}{\rho}\exp\{-\frac{x}{\rho}\}(1-F(\sqrt{2x}))dx}{\int_{\Gamma_k}^{\Gamma_{k+1}} \frac{1}{\rho}\exp\{-\frac{x}{\rho}\}dx}, \quad k = 1, \ldots, N \qquad (1)$$

where $[\Gamma_i, \Gamma_{i+1})$ is the SNR range that corresponds to state i ($\Gamma_1 \stackrel{\text{def}}{=} 0$ and $\Gamma_{N+1} \stackrel{\text{def}}{=} \infty$), ρ is the average SNR for the transmitted signal, and F is the CDF of a standard normal random variable (rv).

We divide the N states into a set of $N-1$ states with bounded errors (i.e., errors that can be corrected using reasonable amounts of FEC) and one "bad" state in which packets cannot be correctly received even when using the maximum allowable number of FEC bits (which we set to the size of the payload portion of the LL packet). To predict the channel states at various receivers, we assume that the current state is recorded at the receiving node and then sent back to the transmitter. The feedback coming from the receiver is assumed to arrive correctly at the transmitter. The rationale behind this assumption is that, because channel-state feedback is conveyed to the base station using small control packets (e.g., RTS/CTS packets), one can apply strong FEC to these packets. To cope with channel errors, we use a hybrid FEC/ARQ scheme with a variable FEC code rate that depends on the predicted channel state. This results in variable-size LL packets.

B. Channel Access Scheme

We consider a CSMA/CA based MAC protocol, where each packet transmission is performed as a sequence of RTS-CTS-Data-ACK. The protocol is similar to that in [6], with some modifications to account for adaptive error correction and multi-state channel predictions. As in [6], we assume that RTS messages are initiated by the mobile stations, whether the transmission is uplink or downlink. This puts the polling out of the picture (i.e., the BS is no more responsible for polling the mobile hosts to determine the backlogged ones). In our access scheme, the signaling of an upcoming transmission is performed twice; once before the start of a frame transmission (a frame is a sequence of LL packets that is formed by one scheduling round at the BS)

and again before the start of a packet transmission. The rationale behind such design will be provided shortly.

The frame structure is depicted in Figure 1. A frame consists of one *control* slot and K *data* slots. The variable K corresponds to the maximum frame size, and is selected in a way to ensure an acceptable level of accuracy in channel state prediction.

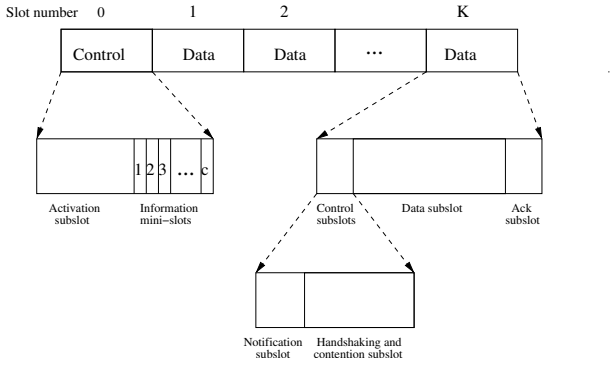

Fig. 1. Format of a transmission frame.

The BS signals the beginning of a frame transmission in the control slot, which is also used to determine and announce the mobile hosts that are *tentatively* selected for transmission in the current frame. In order to perform scheduling for the data slots, a subset of the backlogged mobile hosts are prompted to compute and return the BERs they currently perceive. Note that this BER information is valid only at the beginning of the frame period. For this reason, the amount of error correction applied to the data packet is determined based on the most probable channel state *at the time of packet transmission*.

The control slot starts with an *activation subslot* in which the BS broadcasts an *activation packet*. This packet is used to probe a subset of the users to calculate their BERs [1]. This subset is determined based on an error-free scheduling policy, i.e., information about the channel is not incorporated in such a scheduling policy. The number of flows broadcasted in the activation packet is determined in a way to guarantee an optimum number of data slots in each frame.

In essence, the activation packet acts as a training sequence that is used by the mobile host to determine the channel state (using either the SNR value or the packet error rate). The estimation accuracy increases with the size of the training data. In the case of a Markov channel model, the accuracy also depends on the thresholds of the channel states. For this reason, the size of the activation packet is selected based on the employed channel model.

After a mobile host receives the activation packet, it calculates the BER it perceives. If this host was specified earlier in the activation packet, it acknowledges the BS by transmitting an *information packet* in the *information mini-slot*. This packet includes information on the channel state seen by the respective mobile (or, equivalently, the experienced SNR value) as well as the flow number. The order indicated in the activation packet also indicates the order for the transmission of the information packets. The transmissions are done based on a collision avoidance procedure with a backoff mechanism, as in [6]. The kth mobile sends its information packet $T+(k-1)W$ seconds after the activation packet has been received (k is the order to send as indicated in the activation packet). The variable T includes the processing time needed for the determination of the error rate. The duration W includes the time needed to send the information packet successfully to the BS as well as the processing delay, and at least one propagation delay (defined according to the maximum transmission range of the BS). Based on the information packets, the BS determines the final scheduling order for the current frame and the FEC code rates to be used.

Following the control slot, the transmission of data packets commences in slots with variable durations. Each data slot includes three subslots: a *control subslot*, a *data subslot*, and an *ack subslot*. The control subslot starts with a *notification subslot*, in which the BS broadcasts a list of the flows that can transmit in the current slot. This way, if the RTS packet of the scheduled flow cannot be correctly received by the BS, another flow can be tried. The order in the flow list corresponds to the scheduling order of the flows for that slot. After the notification subslot, the handshaking between the BS and the mobile host selected for transmission takes place in the *handshaking and contention subslot*, as indicated in [6] (the mobile host transmits an RTS packet, and then waits for a CTS packet from the base station). The purpose of this handshaking is to learn the state of the channel at the mobile receiver, and thus be able to detect deep fade before the transmission starts. If none of the flows in the list broadcasted in the notification subslot can accomplish handshaking, flows that perceive a clean channel can contend for channel access using a p-persistent scheme. The RTS packet of a contender includes the most recently observed channel state (if the activation packet for the current frame was successfully received by the contender), which is used by the BS to determine the required amount of FEC bits. If the latest channel state information is not available at the mobile user, a "not available" value is indicated in the channel-state field of the RTS packet, prompting the BS to use a default channel state. The winner of the contention is allowed to transmit in the current data subslot. Note that multiple contention slots can be used to allow prioritization among the flows. Upon the completion of a successful handshaking, data transmission takes place in the data subslot followed by the transmission of its acknowledgement in the ack subslot.

As a final remark, to increase the success rate of the transmissions, the transmission times of the scheduled packets (in terms of the location of the transmission slots in the current frame) can be reordered based on the flow constraints and the channel conditions.

III. Scheduling Algorithm

In this section, we propose a scheduling algorithm that operates under probabilistic channel state information. In our analysis, we assume that flows are delay constrained, where the delays are calculated after the arrival of the packet at the transmitter.

In order to support similar guarantees to those provided by a wireline network, we employ an error-free service model to determine the initial order of transmissions. We use WFQ as the error-free scheduler. In WFQ when a packet p arrives at a queue,

[1] A pilot signal sent over a separate channel can also be used for this purpose, but this requires separate data and control channels.

it is time-stamped with a start tag $S(p)$. Start tags are used to calculate the finish tags ($F(p)$) as follows [6]:

$$\begin{aligned} S({p_i}^k) &= \max(V(A({p_i}^k)), S({p_i}^{k-1}) + L/r_i) \\ F({p_i}^k) &= S({p_i}^k) + L/\phi_i \\ dV/dt &= C(t)/(\textstyle\sum_{i \epsilon B(t)} r_i) \end{aligned} \tag{2}$$

where p_i^k is the kth packet of the ith flow, $V(t)$ is the virtual time value at system time t, $A({p_i}^k)$ is the arrival time of the kth packet of the ith flow, L is the *average* packet length, r_i is the throughput weight for the ith flow, ϕ_i is the delay weight for the ith flow, $C(t)$ is the link capacity at time t, and $B(t)$ is the set of backlogged flows at time t.

The algorithm selects the packet with the minimum finish tag for transmission. Because of channel errors, the error-free service model is not sufficient to determine an efficient scheduling order. To protect flows against channel errors, we need to make the channel errors invisible to these flows through the exchange of slots among flows. Whenever a flow cannot transmit due to deep fade, the scheduler has to assign the slot to another flow. Therefore, the error-free service model is used as a means of measuring the amount of services received and determining whether the flow is leading or lagging.

The proposed algorithm accounts for the future channel state estimates by integrating them together with the compensation mechanism into a single parameter which we refer to as the *success metric*. This metric is defined as the probability of satisfying the delay constraint of the HOL packet of a given flow (or simply, the probability of success). The proposed algorithm tries to schedule the flow for which the HOL packet has a higher chance of being successfully transmitted. Thus, unsuccessful transmission attempts resulting in packet losses at higher layers are often prevented.

In order to find the probability of success for the ith flow, we need to find the maximum number of allowable retransmission attempts, R_i^*, that the HOL packet of flow i is allowed to have before its deadline expires. Note that this value varies from one flow to another due to the variation in the queuing delays, flow weights, and the delay constraints among the flows. It also varies from one scheduling attempt to another. To come up with a good prediction of R_i^*, we have to take into account the compensation parameters (such as lead and lag values) properly, since they determine the actual service rate of a flow.

Let $d_{remain}^{(i)}$ be the remaining delay that the HOL packet for flow i can tolerate until it is successfully transmitted or until it is dropped by the transmitter. The variable $d_{remain}^{(i)}$ is given by $(d_{max}^{(i)} - d_{queue}^{(i)})$, where $d_{max}^{(i)}$ is the delay constraint for flow i and $d_{queue}^{(i)}$ is the time elapsed since the HOL packet of the ith flow entered the queue. From $d_{remain}^{(i)}$, we can calculate R_i^* as follows:

$$R_i^* \approx \left\lfloor d_{remain}^{(i)} \cdot \frac{C(t) \cdot \hat{r_i}}{L \cdot \sum_{j \epsilon B(t)} r_j} \right\rfloor \tag{3}$$

where $\hat{r_i}$ is the *short-term average flow weight* of the ith flow. In calculating R_i^*, we assume that the round trip time is equal to the transmission time and ignore the effect of propagation delay.

The short-term average flow weight is defined as the average weight (normalized service rate) that a flow will have during the lifetime of its current HOL packet. In other words, a correction factor is applied to the flow's original weight to take into account the additional rate the flow gets in return for services allotted to other flows or the rate that the flow gives to other flows in return for the excess service it previously received. In our calculations, we use a rather crude approximation to determine $\hat{r_i}$, setting it to the *instantaneous flow weight* obtained at the time the schedular runs. A pseudocode to calculate this weight is given in Figure 2.

```
1  i = flow number
2  Lead_Rate[i] = minimum(Lead[i], Maximum_Lead[i]) / Maximum_Lead[i]
3  if (Lead[i] > 0)
4     r̂_i = r_i · (1 − Lead_Rate[i])
5  else if (Lag[i] > 0)
6        r̂_i = r_i + Lag[i] / (Σ_{j∈B(t)} Lag[j]) · Σ_{j∈B(t)} Lead_Rate[j] · r_j
7     else
9        r̂_i = r_i
```

Fig. 2. Algorithm for finding the instantaneous flow weight.

Here, we use a similar logic to that of WFS to provide fairness. However, we provide fairness by adjusting the flow weights to obtain an approximate value for the probability of a successful transmission. Subsequently, we select the flow with the highest chance of transmitting its HOL packet successfully. If a flow is lagging, its short-term average weight is going to be larger than its assigned (original) weight, resulting in higher number of transmission slots (or higher chance of being scheduled).

Conversely, a leading flow will have to relinquish some or all of its weight which results in a lower number of transmission slots. This way, we provide both short-term and long-term fairness indirectly by adjusting the flow weights.

The probability of success is given by the probability that the HOL packet's delay $d^{(i)}$ is smaller than or equal to $d_{remain}^{(i)}$, which is equal to:

$$\Pr(d^{(i)} \leq d_{remain}^{(i)}) = \sum_{j=1}^{R_i^*} Q_{succ,i}^{(j)}(s_0) \tag{4}$$

where s_0 is the initial channel state of flow i and $Q_{succ,i}^{(j)}(s_0)$ is the probability that the jth transmission attempt for the ith flow succeeds after $j-1$ unsuccessful transmission attempts with the initial state given as s_0. If the value of $d_{remain}^{(i)}$ reaches zero and the packet has not yet been successfully transmitted, the packet has to be dropped to avoid wasting the bandwidth.

We calculate $Q_{succ,i}^{(j)}(s_0)$ using the channel state transition matrix $\boldsymbol{P}$ and the flow-specific parameters. Let $P_{ij}^{(k)}$ be the k-step transition probability from state i to state j. Then,

$$Q_{succ,i}^{(1)}(s_0) = \sum_{j=1}^{N} {P_{s_0 j}}^{\lfloor (1/\tilde{r_i} + 0.5) \rfloor} \cdot P_{correct,i}^{(1)}(s_0, j) \tag{5}$$

where $\tilde{r_i}$ is the normalized weight for the ith flow ($\tilde{r_i} = \hat{r_i} / \sum_{k \epsilon B(t)} r_k$) and $P_{correct,i}^{(1)}(s_0, j)$ is the probability of being correct in the first transmission attempt given the initial state

as s_0 and the final state as j. The factor $\lfloor (k/\tilde{r}_i + 0.5) \rfloor$ corresponds to the location of the kth transmission slot for the ith flow. We can determine $P^{(1)}_{correct,i}(s_0, j)$ as follows:

$$P^{(1)}_{correct,i}(s_0, j) = \sum_{s \epsilon N^{\eta}_j} \alpha(s, j, n_{i,1}) \cdot \sum_{i=0}^{k^*_s} \left(\begin{array}{c} n_s \\ i \end{array} \right) \cdot {e_j}^i \cdot (1 - e_j)^{n_s - i} \tag{6}$$

where N^{η}_j is the set of states starting from which the Markov chain can reach state j in η steps, $\alpha(s, j, \nu)$ is the probability of making a transition from state s to state j where the location of state s is the closest location to state j at the time the channel state is estimated (e.g., the beginning of a frame transmission period) given that a transition from the initial state s_0 to state j occurs in ν steps, k^*_j is the maximum number of correctable bits in the packet when the channel is in the jth state, and n_j is the packet size including FEC bits when the channel is in state j. $\alpha(s, j, \nu)$ is computed as follows:

$$\alpha(s, j, \nu) = \Pr(estimate = s | S(\nu) = j) =$$
$$\Pr(S(\nu - \eta) = s | S(\nu) = j) = P^{\eta}_{sj} \cdot \frac{P^{\nu-\eta}_{s_0 s}}{\sum_{\tau} P^{\nu-\eta}_{s_0 \tau} \cdot P^{\eta}_{\tau j}} \tag{7}$$

where ν is the location of the transmission slot, η is the distance between the closest estimation slot and the transmission slot ($\eta = (\nu, \ell)$, where ℓ is the average frame duration in slots).

We can obtain $Q^{(j)}_{succ,i}(s_0)$, $j = \{2, \ldots, R^*_i\}$, recursively after computing $Q^{(1)}_{succ,i}(s_0)$:

$$Q^{(\tau)}_{succ,f}(s_0) = \sum_{j=1}^{N} P^{n_{f,1}}_{s_0 j} \cdot (1 - P^{(1)}_{correct,f}(s_0, j)) \cdot Q^{(\tau-1)}_{succ,f}(j), \quad \tau = 2, \ldots, R. \tag{8}$$

In order to minimize the number of computations, the algorithm is only executed when the base station is powered up and when an update is necessary, i.e., when the channel transition matrix changes, which we assume to occur infrequently. This can be achieved by first defining the matrix $[\mathbf{Q^{(1)}_{succ}}]_{i,j}$ for each possible flow weight and initial state combination. The $[i, j]$th element of this matrix corresponds to being successful in the first transmission attempt with initial state given as i and the location of the first transmission attempt (in slots) given as $\lfloor 1/r(j) + 0.5 \rfloor$. Then, using (8), we determine the higher level matrices $[\mathbf{Q^{(\tau)}_{succ}}]_{i,j}$, again for each possible flow weight and initial state combination. At each scheduling instant, by summing the corresponding elements of these matrices (based on the initial state and the instantaneous flow weight), we can determine the success probability for each flow. The flow with the highest success metric is selected for transmission.

Using the proposed algorithm, we can make scheduling more adaptive to channel variations. As a result of this adaptiveness, a change in the channel conditions (e.g., due to mobility) does not cause a significant change in the perceived performance. Another advantage of the proposed success metric is that it can also be used to manage buffer admission. This is done by applying a similar recursive procedure to the one outlined before to obtain the success probability for all packets in the queue (not just the HOL packet). This way, we can determine whether an incoming packet can be served on time or not, and decide accordingly on whether to admit it or not.

IV. Simulations

In this section, we use simulations to evaluate the performance of the proposed algorithm and contrast it with WFS [6]. For simplicity, we ignore the MAC overhead. Due to space considerations, we only give the results when the frame size equals to one, i.e., the scheduling algorithm is run before each packet transmission. For the channel model, we use a 4-state Markov model whose parameters are determined using the SNR partitioning scheme in [11]. Packet transmission is allowed in three of the four states, which represent the good state. We defer the transmission if the channel state is predicted to be in the bad state. For the good states, Reed-Solomon coding is used for error correction. For each flow, packets arrive according to a two-state Markov Modulated Poisson Process (MMPP). Each simulation is run for 50000 time units, with each time unit corresponding to a packet transmission time. The simulations are repeated 10 times. The payload size is chosen to be 424 bits. The delay constraint for each flow is selected to be 50 time units. The number of users M equals to 10 and each user has the same weight.

We evaluate the performance of the proposed algorithm in terms of the achieved throughput, average delay to successfully deliver a packet to the receiver, and the number of retransmissions per delivered packet. For conciseness, we report the percentage change in these performance measures relative to their counterparts in the WFS algorithm. Figure 3 shows the percentage change in throughput versus the normalized traffic intensity for three different levels of mobilities: low mobility (doppler frequency $f_m = 10$ Hz), medium mobility ($f_m = 50$), and high mobility ($f_m = 100$). When the mobility is low, we do not gain from the future channel state estimates since the channel varies very slowly. In this case, the advantage of our algorithm over WFS is negligible. As mobility increases, we start to observe a marked improvement in the throughput. This should be expected since as mobility increases, the future channel state estimates start to gain more importance in the decision process. The accuracy of our decisions can be observed more clearly if we examine the improvement in the delay performance, as shown in Figure 4 for various levels of mobility. The figure shows that the proposed algorithm can achieve a considerable reduction (improvement) in the average delay compared to WFS. In both figures, we also observe that as the network load increases the improvements also increase. This suggests that as the number of backlogged flows increases the efficiency of the proposed algorithm also increases. Since our algorithm relies on the success probability as the basis for scheduling, less bandwidth is wasted on erroneous transmissions when compared to WFS. The percentage decrease in the retransmissions is shown in Figure 5. As can be seen from this figure, a good reduction in the number of retransmissions can be achieved. Table I compares the proposed algorithm with WFS in terms of short-term fairness in throughput (based on Jain's Fairness Index [4]). The results are very close suggesting that the fairness characteristics are preserved in the proposed algorithm.

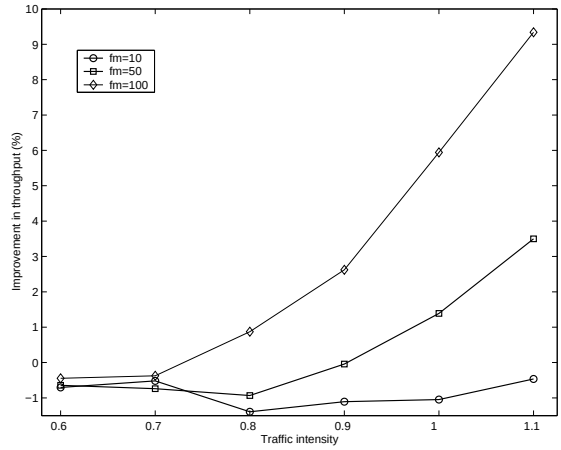

Fig. 3. Percentage change in throughput ($M = 10$).

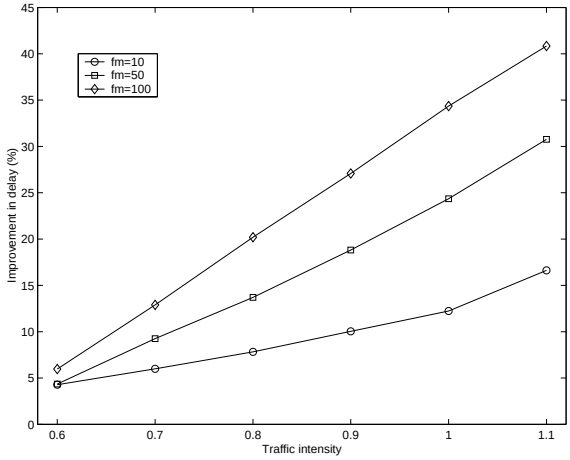

Fig. 4. Percentage change in delay ($M = 10$).

V. Conclusions

In this paper, we introduced a wireless scheduling algorithm that uses a multi-state channel model to predict the future channel states and incorporate these predicted states in the scheduling decision. Because of its channel-dependent nature, the proposed algorithm is highly robust against variations in channel conditions (e.g., due to mobility). A hybrid ARQ scheme was used in which the FEC code rate is adjusted adaptively according to the forecasted channel state. The proposed algorithm employs an *adaptive* virtual rate adjustment mechanism for compensation.

Simulation results indicate that compared to the well-known WFS scheduling algorithm (which does not make extensive use of the channel characteristics), the proposed algorithm achieves good improvements in throughput and delay while preserving the fairness characteristics of WFS. These improvements become more significant as mobility increases. We outlined a MAC protocol (modified from [6]) to go hand-in-hand with the proposed scheduling algorithm. We also described a frame-based transmission strategy for the scheduling algorithm that can operate effectively under dynamic channel conditions.

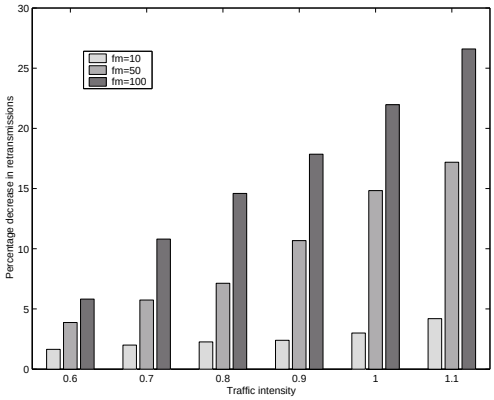

Fig. 5. Percentage decrease in retransmissions ($M = 10$).

	$f_m = 10$	$f_m = 50$	$f_m = 100$
WFS	0.9701	0.9744	0.9756
Proposed	0.9602	0.9716	0.9720

TABLE I
Short-term fairness comparison ($M = 10$).

References

[1] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi. Enhancing throughput over wireless LANs using channel state dependent packet scheduling. In *Proceedings of the IEEE INFOCOM '96 Conference*, pages 1133–1140, 1996.

[2] E. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:1253–1266, Sep. 1960.

[3] M. Hassan, M. Krunz, and I. Matta. Markov-based channel characterization for tractable performance analysis in wireless packet networks. To appear in the *IEEE Transactions on Wireless Communications*, 2003.

[4] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley and Sons Inc., New York, NY, 1991.

[5] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, Aug. 1999.

[6] S. Lu, T. Nandagopal, and V. Bharghavan. A wireless fair service algorithm for packet cellular networks. In *Proceedings of the IEEE MobiCom '98 Conference*, pages 10–20. ACM, 1998.

[7] P. W. Ramanathan and P. Agrawal. Adapting packet fair queueing algorithms to wireless networks. In *Proceedings of the IEEE MobiCom '98 Conference*, pages 1–9. ACM, 1998.

[8] T.S.E. Ng, I. Stoica, and H. Zhang. Packet fair queuing algorithms for wireless networks with location-dependent errors. In *Proceedings of the IEEE INFOCOM '98 Conference*, pages 1103–1111. IEEE, 1998.

[9] H. Wang and N. Moayeri. Finite-state Markov channel - a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, Feb. 1995.

[10] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proceedings of the IEEE*, pages 1374–1396. IEEE, Oct. 1995.

[11] Q. Zhang and S. A. Kassam. Finite-state Markov model for Rayleigh fading channels. *IEEE Transactions on Communications*, 47(11):1688–1692, Nov. 1999.