CCIT Report #371

February 2002

Stochastic Analysis of Symmetric Two-Flow Wireless-Fair Scheduling

Raphael Rom and Hwee Pink Tan

March 25, 2002

Abstract

Future Wireless Networks are expected to offer diverse services with different grades of Quality of Service (QoS). Wireless scheduling plays an important role in QoS provisioning since it determines how scarce resources will be allocated to concurrently support the diverse demands of users.

In this report, we shall develop an analytical model for a simple symmetric two-flow wireless scheduling algorithm based on the fair queuing paradigm (*Wireless-Fair Scheduling*). By proper selection of the analysis interval, time instances as well as definition of a suitable state variable, the behavior of the scheduler can be characterized by a one dimensional Markov Chain. Based on the model, we shall derive the delay and fairness performance of the two-flow scheduler in terms of channel state parameters.

In addition, we shall also define a *Wired-Fair* scheduling scheme which is fair and but not channel-efficient and a *Channel-Efficient* scheduling scheme which is not fair. We shall compare the performance of these schemes under different channel conditions in terms of channel-efficiency and fairness.

Contents

Chapter 1

Introduction

Future Generation wireless networks are envisaged to bring existing wireline applications, including high speed data and multimedia, to mobile users via a wireless environment. Users are expected to carry diverse traffic types with vastly different Quality of Service (QoS) requirements. This poses a challenge to the network's traffic management mechanism to provide seamless QoS to the mobile users.

Scheduling determines how resources will be allocated amongst contending users, and hence, is an integral component for QoS provisioning. Whilst an abundance of such algorithms, e.g., Fair Queuing scheduling [1], Virtual Clock [2] and EDD [3] that provides guaranteed QoS exists for wireline networks, research activities on scheduling over a wireless media (Wireless Scheduling) took off only in the last few years. Direct application of wireline scheduling to the wireless media is not useful due to the following unique characteristics: (a) high error rate and bursty errors (b) location dependent and time varying capacity (c) scarce bandwidth (d) user mobility and (e) power constraint of mobile users.

A comprehensive survey of wireless scheduling algorithms is given in [4] and [5]. Most of the algorithms proposed can be mapped onto a Unified Wireless Fair Queuing Framework [4]. Under this framework, scheduling is performed according to a wireline fair queuing algorithm under error-free conditions. However, under error-prone conditions, the scheduler swaps transmission amongst flows based on their respective channel conditions to maximize channel efficiency. To ensure fairness, an accounting system is maintained to monitor the reassignment activities in order to make up for 'lost' transmissions in the future. In terms of per-flow QoS requirements, the performance of these algorithms are evaluated in terms of the following parameters: (a) channel-access delay bound, (b) short and long-term throughput guarantees and (c) long and short-term fairness. Based on simulation results in [4], it is suggested the CIF-Q [6] and WFS [7] algorithms offer the best QoS performance.

However, it is noted that the analytical bounds obtained for the above algorithms seem somewhat incomplete. For the CIF-Q algorithm, performance bounds are derived for error-free flows only. For the WFS algorithm, in addition to error-free flows, performance bounds conditioned on the channel characteristics are derived for error-prone flows, i.e., the worst case performance under a static error condition (e.g., worst case error rate). Such a deterministic bound is often conservative and hence, is not representative of the scheduler performance. Since the behavior of the wireless channel can typically be modeled as a stochastic process, statistical performance bounds can be derived by modeling the wireless scheduler as a stochastic process. These bounds are expected to better describe the scheduler performance.

In this report, we shall develop an analytical model for a wireless fair scheduling algorithm that maps to the Unified Wireless Fair Queuing Framework. Based on this model, we shall derive statistical performance bounds for packet delay and fairness for various scheduler designs and channel error models.

Chapter 2

Scheduling Scenario

We consider a typical centralized scheduling scenario, where N flows (denoted flow 1,2,..N) contend for a shared resource. In this case, the shared resource is the access to a common wireless channel. The channel is assumed to be slotted in time, where all slots are of equal size. The traffic of each flow is characterized by its rate weight (denoted $r_1, r_2, ..r_N$ where $\sum_{i=1}^{N} r_i$ = 1).

Since the purpose of this paper is to establish the delay and fairness properties of the scheduling mechanism, we assume that at the beginning of each slot, the scheduler has perfect knowledge of the channel state of each flow, i.e., if the flow perceives an error-free channel or an error-prone channel. A flow's transmission is successful only if it perceives an error-free channel.

In addition, although the channel state of each flow is assumed to be independent of that of other flows, we only consider time-dependent, location-independent channel errors so as to disregard the 'unfairness' introduced by location-dependence of the channel states of different flows.

Chapter 3

Wireless-Fair Scheduling

Fair queuing is a popular scheduling approach that provides throughput and fairness guarantees as well as bounded-delay link access in a wired network. These desirable properties will be degraded if fair queuing is directly applied for scheduling in a wireless media because of the following characteristics:

(a) *High transmission error rate*

Since flows are scheduled independent of channel conditions, slots are wasted (which is significant compared to a wired link) when the flow allocated to transmit perceives channel error, resulting in reduced channel efficiency.

(b) *Time-dependent and Bursty transmission errors*

Due to the burstiness of channel errors, the fairness property no longer holds over intervals comprising erroneous slots. In order to maintain fairness, channel efficiency can be significantly reduced if the flow undergoing an error-burst is repeatedly polled by the scheduler.

(b) *Flow-dependent transmission errors*

In addition to time dependence, the transmission error over the wireless channel may also be spatially dependent (e.g., due to user mobility), and hence differ amongst flows. This can introduce additional unfairness to the scheduler.

Based on the above characteristics, wireless adaptation techniques can be introduced to the fair queuing paradigm such that the resulting wireless-fair queuing algorithm will:

(a) emulate the performance of fair queueing under error-free conditions (thus achieving throughput, fairness and delay guarantees), and

(b) maintain optimal channel efficiency (thus minimizing degradations to delay and throughput guarantees) and fairness guarantees under error-prone conditions.

3.1 Wireless Adaptation Mechanism

Most wireless-fair scheduling algorithms that were recently proposed in the literature perform wireless adaptation by (a) reassigning flows for transmission based on their channel states and (b) subsequently compensating for the reassignment. This is further elaborated as follows:

When a flow that is scheduled for transmission in the next slot predicts channel error, another flow that perceives a clean channel in the given slot (which is likely to exist due to the location-dependent nature of channel errors) will transmit instead to minimize slot wastage and hence optimize channel efficiency. The scheduler accounts for the ‘loss’ slot and attempts to compensate the flow for it at a later time. The extent to which the scheduler minimizes slot wastage and maximize slot compensation represents a trade-off between channel efficiency and short-term fairness provision.

3.2 Unified Wireless Fair Queuing Framework

A Unified Wireless Fair Queuing Framework has been defined in [4]. With our assumption of perfect channel knowledge, and that each packet is kept in its queue until successful transmission (i.e., infinite retransmission bound), the framework is simplified and comprises the following components:

3.2.1 Error-free Service

The error-free service is the scheduling scheme employed in an error-free environment to provide throughput, fairness and delay guarantees. It is typically some packetized approximation of the fluid fair queueing paradigm (see [1],[8],[9],[10],[11],[7]).

3.2.2 Lead and Lag Accounting Service

The notion of per-flow lag(lead) is defined to keep track of the amount of additional service that a flow is entitled to (needs to relinquish) in the future in order to compensate for service lost (gained) in the past. It is used as an input to the compensation service to select the flow to transmit in the next slot. The definition of the service differs in terms of the choice of the reference and the existence of bounds for lead/lag.

3.2.3 Compensation Service

This component enables lagging flows to reclaim ‘lost’ service (due to channel error) from leading flows by defining the mechanism (how, when and which) for which a flow relinquishes or transmits in a given slot.

Chapter 4

Analytical Model

In Section 3, we have described a general framework for designing a wireless-fair scheduler based on time-slot swapping and compensation. In this section, we will define a specific wireless-fair scheduler and build a model in order to analyze its performance.

4.1 Assumptions

Several assumptions are in place here. We shall consider a scheduling scenario with N=2 where $r_1 = r_2 = 0.5$ (symmetric two-flow wireless-fair scheduling). The arrival process of one flow is assumed to be independent of the other. In addition, for each flow, the arrival process in one interval is independent of the arrival process in a non-overlapping interval. Each message comprises one packet, and all packets are of equal size with transmission time of one slot.

We also assume that the queue of each flow is of infinite length and at any instant in time, each queue can be in one of two states, namely, *empty* or *backlogged*.

4.2 Definition of Symmetric Two-Flow Wireless-Fair Scheduler

According to Section 3, we can define our symmetric two-flow wireless-fair scheduler by specifying its error-free service, lead and lag accounting service and its compensation service.

4.2.1 Error-free service

For a system where packets are of fixed size equal to the slot width, if all flows are backlogged at all times during the interval of analysis, then Weighted Round Robin with spreading (WRR-spreading) is equivalent to Weighted Fair Queueing (WFQ) [11]. Hence, WRR-spreading will be used as the error-free service. For a symmetrical two-flow system, WRR-spreading reduces to simple alternate slot allocation.

4.2.2 Lead and Lag Accounting Service

Since we do not simulate fluid fair queueing, the lead/lag of each flow will be computed relative to each other such that at any time instant, $\sum_{j=1}^{2} lead\ of\ flow\ j = 0$.

For a two-flow system, we can define a single variable, *x*, to denote the lead of flow 1 relative to flow 2 (or the lag of flow 2 relative to flow 1). Flow 1 is defined as *leading, lagging or in-sync (neither leading nor lagging)* according to the following:

$$x \begin{cases} = 0, & flows\ are\ in-sync; \\ > 0, & flow\ 1\ is\ leading; \\ < 0, & flow\ 1\ is\ lagging. \end{cases}$$

The value of *x* is updated (a) whenever a flow transmits in a slot given up by another flow or (b) whenever a lagging flow is 'catching up' on its lag by transmitting in its own slot.

4.2.3 Compensation Service

We consider the compensation service where absolute priority is given to the lagging flow, i.e, as long as there exists a lagging flow, slots are always allocated to it until it 'recovers' from its lag, i.e, when *x* = 0; otherwise, slots are allocated according to the error-free service.

4.3 Definition of Intervals for Performance Analysis

Fig. 4.1 depicts an example of the queue status of both flows over an interval of time.

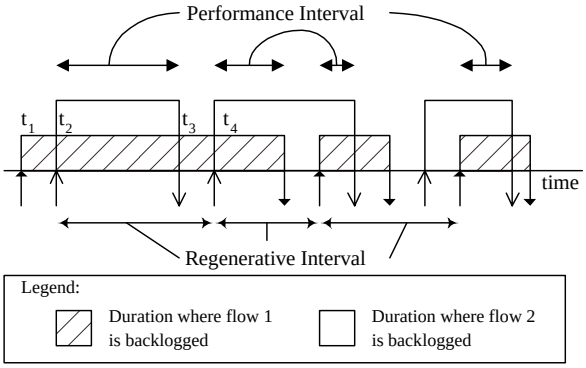


Figure 4.1: Queue status for two-flow scheduling scenario

At time t_1, flow 1 becomes backlogged, and since it is the only flow that is backlogged, all slots are allocated to it until t_2 when both flows become backlogged. During the interval $[t_2, t_3]$, slots are allocated alternately to each flow. At t_3, flow 2 empties, and hence, all slots are allocated to flow 1 until t_4, when flow 2 becomes backlogged again.

Let us define a variable, x_i to denote the lead of flow 1 in slots (or equivalently the lag of flow 2) at the end of slot i. A positive x_i indicates that flow 1 is leading over flow 2 by x_i slots, while a negative x_i indicates that flow 1 is lagging from flow 2 by x_i slots. If we consider all intervals where both flows are backlogged (e.g., $[t_2, t_3]$ in Fig. 4.1), then x is initialized to zero at the beginning of all such intervals.

If we consider an interval between the instant when both flows become backlogged to the next instant when both flows become backlogged again (e.g., $[t_2, t_4]$ in Fig. 4.1), we notice that the evolution of *x* within such an interval is independent of that in past intervals because at the beginning of each interval, *x* is reset. Hence, the instances where *x* is reset are regenerative points with respect to *x* and we define the interval between two successive regenerative points as a *regenerative interval*.

Let us consider a regenerative interval. It always begins with a sub-interval where both flows are backlogged, followed by at least one sub-interval where only one flow is backlogged. Sometimes, it may also include a sub-interval where both flows are empty. We are interested in the performance analysis over the sub-interval where both flows are backlogged (e.g., $[t_2, t_3]$ in Fig. 4.1) since the other cases are trivial. We define such an interval as a *performance interval*.

4.4 Mechanism of Wireless-Fair Scheduler

Let us consider the performance interval as shown in Fig. 4.2, where time is slotted and slots are numbered 1,2,3,...from the beginning of each performance interval.

For any slot *i*, the lead of flow 1 at the end of slot *i* is denoted by x_i. Let A_i denote the flow that is allocated in slot *i*, where $A_i = \{S1,\ S2\}$. If $A_i = Sj$, then slot *i* is allocated to flow *j*.

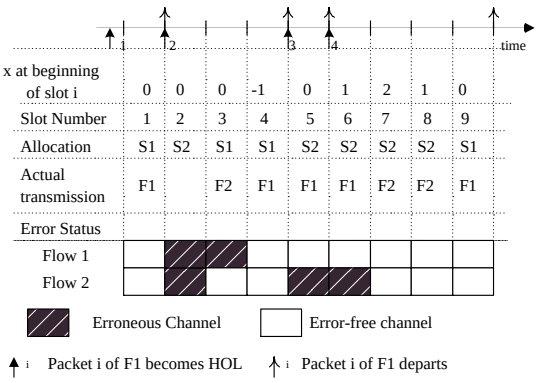


Figure 4.2: Performance Interval for a two-flow scheduling scenario

Under an error-free environment, the error-free service reduces to a simple slot allocation policy based on the status of the queue to each flow. At any instant of time, as long as only one flow is backlogged, all slots are allocated to that flow. Once the other flow becomes backlogged while that flow stays backlogged, slots are allocated in an alternate manner (i.e., $A_i = \overline{A_{i-1}}$) until one queue empties. No transmission takes place when neither queue is backlogged.

However, under an error-prone environment, the above slot allocation policy is sub-optimal in terms of channel efficiency and is also unable to guarantee fairness to both flows. This is where the lead and lag accounting service as well as the compensation service (described in Section **??**) come into play.

In Fig. 4.2, the first flow 1 packet arrives to the system while a flow 2 packet is transmitting in slot 0. Hence, the performance interval begins with slot 1 and x_0 = 0. Let us assume that $A_1 = S1$, i.e., flow 1 has priority of transmission in slot 1. Flow 1 perceives an error-free channel in slot 1 and transmits successfully. Since x_0 = 0 and neither flow gained in transmission relative to the other, $x_1 = x_0$ = 0. To ensure fairness, slot 2 is allocated S2 (i.e., $A_2 = \overline{A_1}$). However, both flows perceive erroneous channels in slot 2 and since neither flow can transmit, $x_2 = x_1$ = 0. Although slot 3 is allocated S1, flow 2 transmits as it perceives an error-free channel while flow 1 perceives an erroneous channel. Hence, flow 2 gains a lead of 1 slot (or equivalently, flow 1 gains a lead of -1), and therefore, x_3 = -1. Since flow 2 leads at the beginning of slot 5, the scheduler attempts to compensate flow 1 by allocating subsequent slots to S1 until both flows are *in-sync* (i.e., *x* = 0 where neither flow leads nor lags relative to the other). In slot 4, flow 1 transmits as it perceives an error-free channel, thereby incrementing *x* such that $x_4 = 0$.

Since $x_4 = 0$, the system resumes alternate slot allocation, and slot 5 is allocated to S2. However, in this slot, flow 1 transmits as it perceives an error-free channel while flow 2 perceives an erroneous channel, thus giving flow 1 a lead of 1 slot over flow 2 (or equivalently, $x_5 = 1$). Subsequent slots are allocated S2 to compensate flow 2 until both flows are in-sync again. However, the channel of flow 2 is undergoing an error burst and flow 1 increases its lead by transmitting in slot 6. The system continues to allocate slots to flow 2 until sufficient flow 2 transmissions take place, and *x* is decremented with each transmission to zero at the end of slot 8.

4.5 Stochastic Characterization

Let us characterize the behavior of the symmetric two-flow wireless-fair scheduler by looking at the evolution of *x* over each slot interval.

Let us denote the set of all possible events that can occur in any slot i by *E* as follows:

$$E = \left\{ \begin{array}{l} No\ Flow\ transmits (NF), \\ flow\ 1\ transmits\ when\ A_i = S1\ (f1S1), \\ flow\ 1\ transmits\ when\ A_i = S2\ (f1S2), \\ flow\ 2\ transmits\ when\ A_i = S2\ (f2S2), \\ flow\ 2\ transmits\ when\ A_i = S1\ (f2S1) \end{array} \right\}$$

4.5.1 Effects of each transmission event on x

In any slot *i*, whenever no transmission takes place (i.e., event NF), neither flow achieves any lead over the other flow. However, when the event f1S2 takes place, flow 1 gains a lead of one slot over flow 2; on the other hand, when the event f2S1 takes place, flow 1 suffers a lag of 1 slot relative to flow 2.

When flows transmit in their allocated slots (i.e., event f1S1 or f2S2 occurs), if they were in-sync before the transmission, neither flow gains with respect to each other. Otherwise, the flow that is allocated the slot is lagging (since lagging flows always receive priority in allocation) before the transmission and perceives a 'clean' channel in slot *i*. Hence, its lag will be reduced after the transmission.

In summary, the effects of each transmission event in slot *i* on x_i can be depicted as follows:

$$x_i = \left\{ \begin{array}{ll} x_{i-1}, & NF \cup (f1S1 \cup f2S2) \cap x_{i-1} = 0; \\ x_{i-1} + 1, & f1S2 \cup f1S1 \cap x_{i-1} < 0; \\ x_{i-1} - 1, & f2S1 \cup f2S2 \cap x_{i-1} > 0. \end{array} \right.$$

4.5.2 Effects of *x* on A_i

The scheduler always allocates a given slot *i* to a lagging flow if it exists (i.e., when $x_{i-1} \neq 0$); otherwise, the allocation is alternate. Hence, the effects of *x* on A_i can be depicted as follows:

$$A_i = \left\{ \begin{array}{ll} \overline{A_{i-1}}, & x_{i-1} = 0; \\ S1, & x_{i-1} < 0; \\ S2, & x_{i-1} > 0. \end{array} \right.$$

Given x_{i-1}, the value of x_i depends on the transmission event in slot *i*. The value of A_i determines the set of allowable events, i.e., when A_i = S1, the allowable events are $\{NF, f1S1, f2S1\}$ whereas if A_i = S2, the allowable events are $\{NF, f2S2, f1S2\}$. Within each set of allowable events, the probability of occurrence of each transmission event in slot *i* depends only on the channel state of each flow in slot *i*.

The value of A_i can be determined from x_{i-1} if $x_{i-1} \neq 0$; otherwise, the value of A_{i-1} is needed since in this case, $A_i = \overline{A}_{i-1}$.

In summary, given $x_{i-1},\ A_{i-1}$ and the channel statistics of each flow in slot *i*, x_i can be determined. In other words, the wireless scheduler can be modeled as a two dimensional Markov Chain with state variables given by $\{(x_i, A_i),\ i = 1, 2, 3...\}$ defined at each slot interval (Markov points).

4.6 Simplification of 2-D Markov Model of Wireless Scheduler

Let us consider a performance interval (i.e., an interval during which both flows are continuously backlogged). Instead of observing the value of *x* at each slot interval, let us consider only the departure instances of flow 1 packets, or equivalently, the instances when the flow 1 packets become $head - of - line$ (HOL). Let us consider a packet of flow 1 which departs from the system in slot *i*-1 with x_{i-1} = 0. In slot *i*-1, either one of the events, f1S1 or f1S2, could have occurred. Let us assume that f1S2 occurred in slot *i*-1. Since f1S2 always results in an increment of *x*, this implies

that $x_{i-2} = x_{i-1} - 1$ = -1. However, if $x_{i-2} = -1$, then A_{i-1} = S1, which is a contradiction since then, the event f1S2 cannot take place in slot *i*-1. Hence, the event f1S1 must have occurred in slot *i*-1, i.e., A_{i-1} = S1. Therefore, S2 is always allocated in slot *i* when x_{i-1}=0 since $A_i = \overline{A}_{i-1}$. Therefore, given the value of *x* when a packet of flow 1 becomes HOL, the distribution of *x* when it departs the system can be computed given the channel statistics.

Let us define the state variable of the system, y_q, to be the value of *x* when the q^{th} packet of flow 1 departs the system within a performance interval, for $q \geq 1$. y_q is also the value of *x* when the $q+1^{th}$ packet becomes HOL. The description in the previous paragraph implies that the probability distribution of y_q can be determined given the channel statistics of each flow as well as y_{q-1}. Hence, the two dimensional Markov Chain, $\{(x_{i-1}, A_i),\ i = 1, 2, 3...\}$, defined in Section 4.5 reduces to a one dimensional Markov Chain, $\{y_q,\ q = 1, 2, 3...\}$, where Markov Points are defined at departure instances of flow 1 packets.

Since the scheduling mechanism is symmetrical with respect to both flows, the same Markov Chain representation is obtained by considering only the departures of flow 2 packet instead of flow 1 packets.

Chapter 5

Performance Evaluation

In Section 4.4, a 1-D Markov Chain representation of a symmetric two-flow wireless-fair scheduler, $\{y_q,\ q = 1, 2, 3...\}$, has been developed over any performance interval, where y_q denotes the lead of flow j, $j \epsilon \{1, 2\}$, when its q^{th} packet departs the system. Given y_{q-1}, as long as the channel states of each flow are known, y_q can be computed. In this section, we shall derive the performance of the scheduler based on its Markov Chain representation.

The performance of the wireless-fair scheduler can be evaluated based on several QoS parameters, e.g., packet delay, throughput, loss probability and fairness. In this analysis, we shall establish the performance of the scheduler in terms of fairness and delay. Since the system is symmetrical, henceforth, we shall consider only packet departures of flow 1.

Let us consider packet *q* of flow 1 which becomes HOL in slot *k*. We say that it has a *duration* of *n* slots if it departs in slot *k+n*. This is illustrated in Fig. 5.1. We shall also define *n* to be the delay of packet *q*. Hence, we shall use the words *duration* and *delay* interchangeably in the rest of the paper.

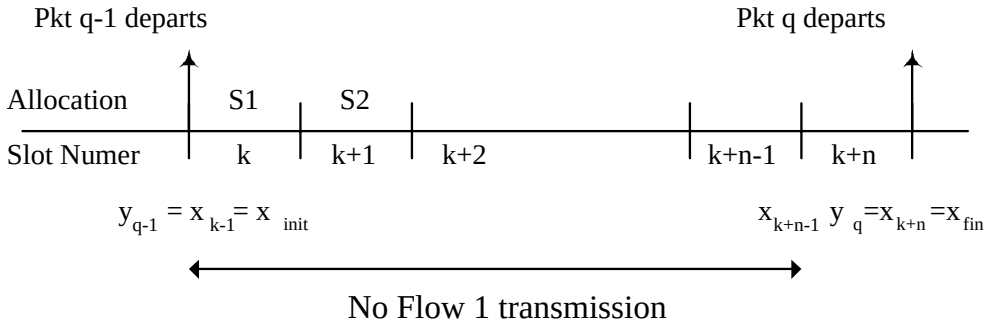


Figure 5.1: Definition of packet duration / delay

5.1 Fairness density function

The value of *y* at any departure instance indicates the gain in cumulative service received by flow 1 at the expense of flow 2 up to and including the departing packet. Hence, $|y|$ represents the disparity in cumulative service received by both flows, and is a coarse measure of the level of 'unfairness' of the scheduler. If we compute the probability density function of *y* (termed the *fairness density function*), then the spread of the density function indicates the level of long-term 'unfairness' of the scheduler; the wider the spread, the more 'unfair' the scheduler is and vice versa. An ideal fair

scheduler will have a fairness density function given by a dirac-delta function, $\delta(x)$ such that

$$\delta(x) = \begin{cases} 1, & x = 0; \\ 0, & otherwise. \end{cases}$$

We proceed to derive the fairness density function of the two-flow symmetrical wireless-fair scheduler. Let us use the following notations:

$$\begin{aligned} p_{x_{init}x_{fin}}(q) &\equiv Prob(y_q = x_{fin} \mid y_{q-1} = x_{init}) \\ &\quad where\ q \geq 2 \\ h_q(x) &\equiv Prob(y_q = x),\ where\ q \geq 1 \end{aligned}$$

where $p_{x_{init}x_{fin}}(q)$ denotes the state transition probability matrix for packet *q* and $h_q(x)$ is probability density function of y_q at flow 1 departure instants. Since y_q is the state variable of the Markov Chain, $h_q(x)$ is related to $h_{q-1}(x)$, $q \geq 2$ as follows:

$$h_q(x) = \sum_{x_{init}} p_{x_{init}x}(q) h_{q-1}(x_{init}) \tag{5.1}$$

Hence, as long as $h_1(x)$ and $p_{x_{init}x}(q)$ are known, $h_q(x)$ can be recursively computed by Eq. (5.12).

Since the scheduler is symmetrical to both flows, the fairness density function at a packet departure of *any* flow, $f_q(x)$, is given by

$$f_q(x) = 0.5 \times (h_q(x) + h_q(-x)) \tag{5.2}$$

The steady state fairness density function, ***f(x)***, is given as follows:

$$f(x) = \lim_{q \to \infty} f_q(x) \tag{5.3}$$

Since ***f(x)*** is symmetrical about *x* (as shown in Eq. (5.3)), the steady state fairness distribution function, ***F(x)***, can be written as follows:

$$F(x) = f(0) + \sum_{y=1}^{x} 2 \times f(y) \tag{5.4}$$

5.2 Packet delay distribution

We have defined the packet delay or duration to be the number of slots from the instant the packet becomes HOL to the instant it departs the system, as illustrated in Fig. 5.1. Since Markov points of the scheduler model are defined at packet departure instants of flow 1 packets, the packet delay distribution actually corresponds to the distribution of the Markov intervals.

Consider packet *q* of flow 1 whose duration is *n* slots. Let us use the following notations:

$$\begin{aligned} r_{x_{init}n}(q) &\equiv Prob(delay\ of\ packet\ q\ = n\ slots \\ &\quad \mid y_{q-1} = x_{init})\ where\ q \geq 2 \\ g_q(n) &\equiv Prob(delay\ of\ packet\ q\ = n\ slots) \\ &\quad where\ q \geq 1 \end{aligned}$$

Assuming that $r_{x_{init}n}(q)$ and $h_{q-1}(x)$ are known, $g_q(n),\ q \geq 2$ can be computed recursively as follows:

$$g_q(n) = \sum_{x_{init}} r_{x_{init}n}(q) h_{q-1}(x_{init}) \tag{5.5}$$

The steady state packet density function, *g(n)*, and distribution function, *G(n)*, are then given as follows:

$$\begin{aligned} g(n) &= \lim_{q\to\infty} g_q(n) \\ G(n) &= \sum_{k=1}^{n} g(k) \end{aligned} \tag{5.6}$$

5.3 Computation of $h_1(x)$

From Eq.(5.12)-(5.6), we observe that as long as $p_{x_{init}x_{fin}}(q)$, $r_{x_{init}n}(q)$ and $h_1(x)$ are known, $h_q(x)$ and $g_q(n)$ ($q \geq 2$) can be computed and hence, f(x) and g(n) can be obtained. In this section, we shall illustrate the computation of $h_1(x)$.

Let us consider packet 1 of flow 1 and assume that it has a duration of *n* slots. Since it becomes HOL in slot 1, it will depart in slot *n*. We begin by computing $h_1(x,n)$, which is the probability density function of *x* at the end of slot *n*, where *n* is the departure slot. To do so, we will consider the transmission events between slots 1 to *n*.

Since the beginning of a performance interval always coincides with the beginning of a regenerative interval (where *x* is reset), the slot allocation policy starts anew at the beginning of each performance interval. Hence, in slot 1 where $x_0 = 0$, the allocation is given as follows:

$$Prob(A_1 = S1) = Prob(A_1 = S2) = 0.5 \tag{5.7}$$

Since the packet can only transmit successfully in slot *n*, for slot *k*, $1 \leq k \leq n-1$, one of the following two events can occur:

(a) No transmission takes place (Event NF)

This event can occur if both flows perceive an erroneous channel in slot *k*, regardless of the channel allocation in slot *k*, and $x_k = x_{k-1}$.

(b) Flow 2 transmits (Event f2S2 or f2S1)

The probability of occurrence of this event depends on A_k. If slot *k* is allocated S1, then this event occurs if flow 1 perceives an erroneous channel while flow 2 perceives an error-free channel, and $x_k = x_{k-1} - 1$. However, if slot *k* is allocated S2, then this event occurs if flow 2 perceives an error-free channel, and $x_k = x_{k-1}$.

By considering each possible A_1, and all allowable events for each A_1 in slot 1, the probability distribution of x_1 can be computed. By repeating this process up to slot *n*, $h_1(x,n)$ can be obtained. By considering all possible *n*, we obtain the following expression for $h_1(x)$:

$$h_1(x) = \sum_{n=1}^{\infty} h_1(x,n) \tag{5.8}$$

5.4 Computation of $p_{x_{init}x_{fin}}(q)$ and $r_{x_{init}n}(q)$

We consider packet *q* of flow 1, as illustrated in Fig. 5.1, that becomes HOL in slot *k* and departs from the system in slot *k*+*n*. Let us look at the transmission events that are possible in each slot *i*, $k \leq i \leq k+n$, constrained by the fact that the packet can only be successfully transmitted in slot *k*+*n*. In slots k,k+1,..*k*+*n*-1, either flow 2 transmits or neither flow transmits. Since *x* can only be incremented by a flow 1 transmission, *x* can be incremented at most once over the packet duration, and hence, we obtain the following constrain:

$$x_{fin} \leq x_{init} + 1 \tag{5.9}$$

From Section 4.4, the allocation in slot *k* can be determined as long as we know if $x_{init} < 0$, $x_{init} > 0$ or $x_{init} = 0$. Hence, we shall consider the three cases separately and for each case, we

shall look at all combinations of permissible transmission events over the duration of packet *q* such that

(a) it transmits successfully only at slot *k+n* (i.e., to compute $r_{x_{init}n}(q)$)

(b) it departs with a lead of x_{fin} (i.e., to compute $p_{x_{init}x_{fin}}(q)$)

We shall use the following notations to denote all possible transmission events in each slot or over an interval of slots:

$f1S1$	flow 1 transmits in slot *i* where A_i=S1
$f1S2$	flow 1 transmits in slot *i* where A_i=S2
$f2S1$	flow 2 transmits in slot *i* where A_i=S1
$f2S2$	flow 2 transmits in slot *i* where A_i=S2
NF	No flow transmits in slot *i*
$S1S2$	In slots *i*, *i*+1 where A_i=S1 and A_{i+1}=S2, NF and (NF $\cup$ f2S2) occur in slot *i* and *i*+1 respectively
$S2S1$	In slots *i*, *i*+1 where A_i=S2 and A_{i+1}=S1, (NF $\cup$ f2S2) and NF occur in slot *i* and *i*+1 respectively
$(E)^m$	*m* successive occurrences of Event E

We shall show here the derivation only for the case of $x_{init} < 0$ here. In this case, slot *k* is allocated S1, independent of the value of *q*. Hence, $p_{x_{init}x_{fin}}(q) \equiv p_{x_{init}x_{fin}}$ and $r_{x_{init}n}(q) \equiv r_{x_{init}n}$.

Since flow 1 cannot transmit before slot *k+n*, the value of *x* cannot be incremented over the slot interval *k*,*k*+1,..*k*+*n*-1, i.e., $x_i < 0,\ k \leq i \leq k+n-1$, and therefore, S1 is allocated from slot k:*k*+*n*-1. In the above interval, one of two events can occur, namely, f2S1 or NF. In slot *k+n*, since $x_{k+n-1} < 0$, S1 is allocated and the only possible event is f1S1.

If there are *m* occurrences of f2S1 in slot *k*:*k*+*n*-1, where $0 \leq m \leq n-1$, then, NF occurs in each of the remaining *n-m-1* slots. By considering all possible values of *m*, we can express $r_{x_{init}n}$ as follows:

$$r_{x_{init}n} = \sum_{m=0}^{n-1} Prob([(f2S1)^m (NF)^{n-m-1}](f1S1)) \tag{5.10}$$

where $[(A)(B)]$ denotes all possible permutations of the events A and B and Prob(*(A)(B)*) denotes the probability of occurrence of event A followed by event B.

Each occurrence of f2S1 in slots *k*:*k*+*n*-1 decrements *x* by a value of 1, while the event f1S1 in slot *k+n* increments it by 1. Hence, for a given *n*, for a packet to depart with *x* = x_{fin}, there must be $x_{init} - x_{fin} + 1$ occurrences of f2S1 in slots *k*:*k*+*n*-1. If we let $c = x_{init} - x_{fin} + 1$, we obtain

$$p_{x_{init}x_{fin}} = \sum_{n=c+1}^{\infty} Prob([(f2S1)^c (NF)^{n-1-c}](f1S1)) \tag{5.11}$$

5.5 Wired-Fair Scheduling

A scheduling algorithm that allocates resources based only on the error-free service described in Section 4.2.1 achieves fairness in a wired environment where transmissions are assumed to be error-free. We call such an algorithm a $wired-fair\ scheduler$. In a wireless environment, while retaining its fairness properties, the performance of such an algorithm is expected to be degraded in terms of channel efficiency, and we are interested to investigate the extent of this degradation.

Since our scheduling system comprises two flows with equal resource requirements, the $wired-fair\ scheduler$ allocates slots in an alternate manner as described in Section 4.4. However, each flow can only transmit in its own allocated slot (i.e., flow i can only transmit in a slot allocated Si).

5.5.1 Packet delay distribution

We refer to Fig. 5.1 and consider packet *q* of flow which becomes HOL in slot *k* and departs in slot *k+n*. For the first packet of flow 1 (*q*=1), the delay density function, $g_1(n)$, where $n \geq 1$ is given as follows:

$$g_1(n) = \begin{cases} Prob(f1's\ channel\ in\ error\ for\ every\ even\ slot\ k,\ 2 \leq k < n\ AND\ error-free\ in\ slot\ n), & n\ even\ and\ A_1 = S2; \\ Prob(f1's\ channel\ in\ error\ for\ every\ odd\ slot\ k,\ 1 \leq k < n\ AND\ error-free\ in\ slot\ n), & n\ odd\ and\ A_1 = S1; \\ 0, & otherwise. \end{cases}$$

Next, consider the case where $q \geq 2$. Since packet *q* of flow 1 becomes HOL in slot *k*, packet q-1 must have departed the system in slot *k*-1, i.e., when A_{k-1}=S1 and the channel for flow 1 was error-free. Since slots are allocated in an alternate manner, $A_k = S2$ always. Hence, the delay density function, $g_q(n)$, where $q > 1$ and $n \geq 1$, is given as follows:

$$g_q(n) = \begin{cases} Prob(f1's\ channel\ in\ error\ for\ every\ even\ slot\ k,\ 2 \leq k < n\ AND\ error-free\ in\ slot\ n), & n\ even; \\ 0, & otherwise. \end{cases}$$

5.5.2 Fairness density function

Since channel-symmetry is assumed (i.e., both flows are subject to the same channel conditions), the Wired-Fair Scheduler is ideally-fair and hence, has a fairness density function, f(x) = δ(x) such that

$$\delta(x) = \begin{cases} 1, & x = 0; \\ 0, & otherwise. \end{cases}$$

5.6 Channel-Efficient Scheduling

In a wireless environment, in the absence of wireless adaptation (as in the Wired-Fair Scheduler), a slot may be 'wasted' when the flow that is allocated the slot perceives an erroneous channel. We define an simple wireless adaptation scheme to the Wired-Fair Scheduler as follows: when a flow cannot transmit in a slot allocated to it due to channel errors, another flow is allowed to transmit in its place if it perceives an error-free channel. In this way, the channel efficiency will be maximized. We term such a scheduler a *Channel-Efficient* Scheduler.

Let us consider packet *q* of flow 1, as illustrated in Fig. 5.1, that becomes HOL in slot *k* and departs from the system in slot *k+n*. Let us define z_q to be the allocation in slot *k*, i.e.,

$$z_q \equiv A_k \epsilon \{S1, S2\}$$

Given z_q and *n*, $\{A_i,\ i\epsilon k : k+n\}$ can be determined and hence, z_{q+1} can be computed. The value of *n* depends on the transmission events in slots *k:k+n*, that in turn depends on the channel state of each flow. Hence, we can define $\{z_q, q = 1, 2, 3..\}$ as a 1-D Markov Chain that characterizes the behavior of the Channel-Efficient Scheduler.

In order to compute the state transition probability matrix, we define the following notations:

$$\begin{aligned} PS_1(q) &\equiv Prob(A_k = S1) \\ PS_2(q) &\equiv Prob(A_k = S2) \end{aligned}$$

where

$$PS_1(q) + PS_2(q) \quad = \quad 1 \quad \forall q$$

We define the following notations to denote the transmission events in single slot *i* (or multiple slots beginning with slot *i*):

$f1S1$	flow 1 transmits in slot *i* where A_i=S1
$f1S2$	flow 1 transmits in slot *i* where A_i=S2
$f2S1$	flow 2 transmits in slot *i* where A_i=S1
$f2S2$	flow 2 transmits in slot *i* where A_i=S2
NF	No flow transmits in slot *i*
$f1E$	flow 1 channel in error in slot *i* where A_i=S1
$S1S2$	In slots *i*, *i*+1 where A_i=S1 and A_{i+1}=S2, f1E and (NF $\cup$ f2S2) occur in slot *i* and *i*+1 respectively
$S2S1$	In slots *i*, *i*+1 where A_i=S2 and A_{i+1}=S1, (NF $\cup$ f2S2) and f1E occur in slot *i* and *i*+1 respectively
$(E)^m$	*m* successive occurrences of Event E

The 1-D Markov Chain of the Channel-Efficient Scheduler can be characterized in terms of the its state transition probability matrix (for $q \geq 2$) as follows:

$$\begin{aligned} PS_2(q) \quad = \quad & \sum_{n=2,\ n\ even}^{\infty} PS_2(q-1)Prob[(S2S1)^{\frac{n-2}{2}}(NF \cup f2S2)(f1S1)] + \\ & \sum_{n=1,\ n\ odd}^{\infty} PS_1(q-1)Prob[(S1S2)^{\frac{n-1}{2}}(f1S1)] \\ PS_1(q) \quad = \quad & \sum_{n=2,\ n\ even}^{\infty} PS_1(q-1)Prob[(S1S2)^{\frac{n-2}{2}}(f1E)(f1S2)] + \\ & \sum_{n=1,\ n\ odd}^{\infty} PS_2(q-1)Prob[(S2S1)^{\frac{n-1}{2}}(f1S2)] \end{aligned}$$

5.6.1 Packet delay distribution

For packet *q* of flow 1, where $q \geq 1$, based on Eq. (5.12), the delay density function, $g_q(n)$, where $n \geq 1$, can be computed as follows:

$$g_q(n) = \begin{cases} PS_1(q) \times Prob[(S1S2)^{\frac{n-2}{2}}(f1E)(f1S2)] & n\ even; \\ +PS_2(q) \times Prob[(S2S1)^{\frac{n-2}{2}}(NF \cup f2S2)(f1S1)], & \\ & \\ PS_1(q)Prob[(S1S2)^{\frac{n-1}{2}}(f1S1)] & n\ odd. \\ +PS_2(q)Prob[(S2S1)^{\frac{n-1}{2}}(f1S2)], & \end{cases}$$

5.6.2 Fairness density function

We define x_i to be the lead of flow 1 at the end of slot *i*, as for the Wireless-Fair Scheduler. In this case, x_i is updated as follows:

$$x_i = \begin{cases} x_{i-1}, & NF \cup f1S1 \cup f2S2; \\ x_{i-1}+1, & f1S2; \\ x_{i-1}-1, & f2S1. \end{cases}$$

In addition to z_q, we can define the variable y_q to be the value of *x* when packet *q* of flow 1 departs the system within a performance interval. Given y_q,z_q, the value of *n* as well as the transmission events in slots *k:k+n*, y_{q+1} can be computed. Hence, instead of the 1-D Markov Chain, $\{z_q, q = 1, 2, 3..\}$, we can define $\{(y_q, z_q), q = 1, 2, 3..\}$ as a 2-D Markov Chain for the scheduler for fairness evaluation, with state transition probabilities defined as follows:

$$\begin{aligned} r_{a_{init}a_{fin}}(q) &\equiv Prob(z_q = a_{fin} \mid z_{q-1} = a_{init}) \\ p_{x_{init}a_{init}x_{fin}}(q) &\equiv Prob(y_q = x_{fin} \mid y_{q-1} = x_{init},\ z_q = a_{init}) \end{aligned}$$

If we define the following:

$$h_q(x) \equiv Prob(y_q = x),$$

then

$$h_q(x) = \sum_{x_{init}} \sum_{a_{init}} p_{x_{init}a_{init}x}(q) h_{q-1}(x_{init}) \tag{5.12}$$

Hence, as long as $h_1(x)$ and $p_{x_{init}a_{init}x}(q)$ are known, $h_q(x)$ can be recursively computed by Eq. (5.12), from which the steady-state fairness density function, f(x), can be obtained.

5.6.3 Computation of $p_{x_{init}a_{init}x_{fin}}(q)$

As in the Wireless-Fair Scheduler, we have the following constraint on x_{fin} and x_{init}:

$$x_{fin} \leq x_{init}+1$$

To derive the expressions for $p_{x_{init}a_{init}x_{fin}}(q)$, we have to consider the following three cases separately (a) $x_{fin} = x_{init}+1$, (b) $x_{fin} = x_{init}$ and (c) $x_{fin} < x_{init}$. We show here the expressions for case (b).

$$p_{x_{init}a_{init}x_{fin}}(q) = \begin{cases} Prob[(S1S2)^{\frac{n-1}{2}}(f1S1)], & n\ odd,\ a_{init} = S1; \\ (_1^{\frac{n-1}{2}})Prob[(NF \cup f2S2)(f2S1)(S2S1)^{\frac{n-3}{2}}(f1S2)], & n\ odd, a_{init} = S2; \\ Prob[(S2S1)^{\frac{n-2}{2}}(NF \cup f2S2)(f1S1)], & n\ even,\ a_{init} = S2; \\ (_1^{\frac{n-2}{2}})Prob[(f2S1)(NF \cup f2S2)(S1S2)^{\frac{n-4}{2}}(NF)(f1S2)], & n\ even,\ a_{init} = S1; \\ Prob[(S1S2)^{\frac{n-2}{2}}(f2S1)(f1S2)], & n\ even\ a_{init} = S1. \end{cases}$$

where the following events are re-defined as follows:

$S1S2$	In slots *i*, *i*+1 where A_i=S1 and A_{i+1}=S2, NF and (NF ∪ f2S2) occur in slot *i* and *i*+1 respectively
$S2S1$	In slots *i*, *i*+1 where A_i=S2 and A_{i+1}=S1, (NF ∪ f2S2) and NF occur in slot *i* and *i*+1 respectively

5.6.4 Computation of $h_1(x)$

Let us consider packet 1 of flow 1 that becomes HOL at the beginning of a performance interval. Since this always coincides with the beginning of a regenerative interval (where *x* is reset), the slot allocation policy starts anew at the beginning of each performance interval and x_0=0. Hence, the allocation in the slot 1, z_1, is given as follows:

$$PS_1(1) = PS_2(1) = \frac{1}{2} \tag{5.13}$$

We note that $h_1(x)$ is actually equivalent to $p_{x_{init}a_{init}x_{fin}}(q)$ for *q*=1, x_{init}=0 with Prob($z_1 = a_{init}$) defined according to Eq. (5.13). Hence, $h_1(x)$ can be evaluated as follows:

$$\begin{aligned} h_1(x) &= PS_1(1)p_{x_{init}=0a_{init}=S1x}(1) + \\ &\quad PS_2(1)p_{x_{init}=0a_{init}=S2x}(1) \end{aligned} \tag{5.14}$$

Chapter 6

Results

The expressions obtained for *f(x)* and *g(n)* in Section 5 can be evaluated in terms of the probability of occurrence of combinations of transmission events. The evaluation of these probabilities depend on the channel error model which we assume for the wireless channel.

6.1 Channel Error Model

We consider two types of wireless channels that differ in terms of the level of burstiness of the error behavior over time, namely, (a) Bursty or Two-State Markov Chain Error Model and (b) Zero-burstiness or Random Error Model.

6.1.1 Two-State Markov Chain (2SMC) Error Model

Channel errors over wireless links are typically bursty in nature and hence, the error behaviour can be suitably modeled as a Markov Chain. We consider a Two-State Markov Chain (2SMC) error model where renewal points are defined at the beginning of each slot and between successive renewal points, the channel is in one of two states, $Good$, Bad. The state transition diagram is given in Fig. 6.1.

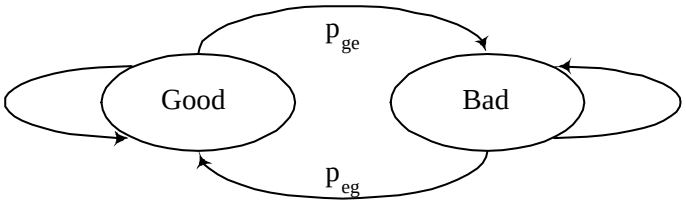


Figure 6.1: State transition diagram of a 2SMC error model

The Markov Chain is specified in terms of two parameters, namely, p_{ge} and p_{corr}, which are defined as follows:

$$\begin{aligned} p_{ge} &= Prob(Bad \mid Good) \\ p_{corr} &= p_{eg} + p_{ge} \quad where \;\; 0 \leq p_{corr} \leq 1 \end{aligned}$$

The steady state probabilities of the channel being in either state are given as follows:

$$P_G = \text{Prob(Good)} = \frac{p_{eg}}{p_{eg}+p_{ge}}$$
$$P_B = \text{Prob(Bad)} = \frac{p_{ge}}{p_{eg}+p_{ge}}$$

The value of p_{corr} is inversely proportional to the level of burstiness of the error behavior of the channel: the lower the value, the higher the correlation between successive slots. In the performance analysis that follows, p_{corr} = 0.1 will be used.

6.1.2 Random Error Model (REM)

This represents the special case where the error behavior is completely uncorrelated across successive slots. The error behavior in any given slot can be described by a single parameter, P_E, where

$$P_E = \text{average error rate in each slot}$$

Since the scheduler has perfect channel knowledge, given the channel state of the previous slot, the scheduler can compute the probability of the channel to each flow being in a certain state. Hence, the probability of occurrence of transmission events and therefore, *f(x)* and *g(n)*, can be evaluated.

6.2 Performance of Symmetric Two-Flow Wireless-Fair Scheduling in different channels

In this section, we shall evaluate the delay and fairness performance of Wireless-Fair Scheduling in terms of *G(n)* and *F(x)* in different channels. Since we have assumed that the channel is flow-independent, the channel conditions for both flows will be identical (i.e., with the same average error rate = $P_B = P_E$).

6.2.1 Delay performance

The delay distribution, *G(n)*, for symmetric two-flow wireless-fair scheduling in different channels are shown in Fig. 6.2. The mean packet delay and its standard deviation (std) are tabulated in Table 6.1.

We observe the loss/delay trade-off of the scheduler under all channel conditions independent of the average error rate. A reduction in delay bound always results in an increase in loss rate. However, the loss/delay trade-off is less pronounced when the channel is correlated.

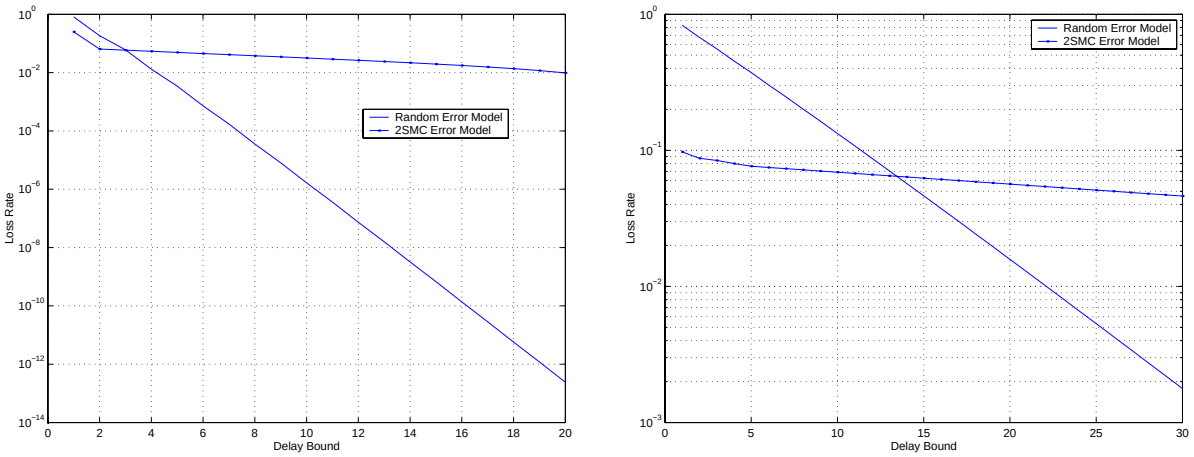
Under low error conditions, as seen in Fig. 6.2(a), the scheduler performs significantly better in a correlated channel in the region of low delay bound and high loss rate. At larger delay bounds, the scheduler performs better in an uncorrelated channel.

Assume that flow switching is negligible at the given error rate and consider a packet of flow 1 that becomes HOL in slot *i*+1 with a delay of *n* slots. It will see slot allocation *S2,S1,S2,S1,*...and will transmit only in slot *i*+2, *i*+4, .. etc. If the channel is correlated, there is a very high probability ($(1-p_{ge})^2 \sim$1) that it will transmit successfully in slot *i*+2. However, when the channel is uncorrelated, the corresponding probability is of the order of $1 - p_E$, which is lower, since $p_{ge} < p_E$.

For $n > 2$, this implies that the channel of flow 1 became erroneous in slot *i*+2. Hence, Prob(n=4) is of the order of $p_{ge} \times p_{eg}$ for a correlated channel and $p_E \times (1 - p_E) >> p_{ge} \times p_{eg}$ for an uncorrelated channel. In a similar way, Prob($n \mid n > 4$, uncorrelated channel) > Prob($n \mid n > 4$,

Error Model \ Channel state	p_E = 0.2		p_E = 0.8	
	mean	std	mean	std
REM	2.08	0.80	5.50	5.17
2SMC EM	1.93	3.80	4.83	19.81

Table 6.1: Mean and std of packet delay for symmetric two-flow wireless-fair scheduling



(a) (b)

Figure 6.2: Delay distribution, G(n), for two-flow wireless-fair scheduling for (a)P_B = 0.2 and (b) P_B = 0.8

correlated channel). Therefore, eventually, at a certain threshold *m*, Prob($n \leq m$, uncorrelated channel) > Prob($n \leq m$, correlated channel).

Similar observations can be made under high error conditions, as seen in Fig. 6.2(b), except that the delay threshold, *m*, is now 14 slots. In this case, significant flow switching occurs, and hence, we have to consider separately the cases when a flow 1 packet becomes HOL with $x > 0$, *x*=0 and $x < 0$.

The explanation given for the low-error case can be applied here for the case of *x*=0. Here, Prob($n = i + 2 \mid$ correlated channel) >> Prob($n = i + 2 \mid$ uncorrelated channel) since $p_{ge} << p_E$. Hence, it will require a larger *m* before Prob($n \leq m$, uncorrelated channel) > Prob($n \leq m$, correlated channel) than the low-error case. Similar observations can be made for the cases when $x \neq 0$.

Hence, Wireless-Fair Scheduling performs better in a correlated channel for delay-sensitive applications (e.g., loss rate $\geq 10^{-1.5}$ and $n \leq 3$ under low error conditions and loss rate $\geq 10^{-1.5}$ and $n \leq 14$ under high error conditions). For error-sensitive applications, the algorithm performs better when the channel is uncorrelated.

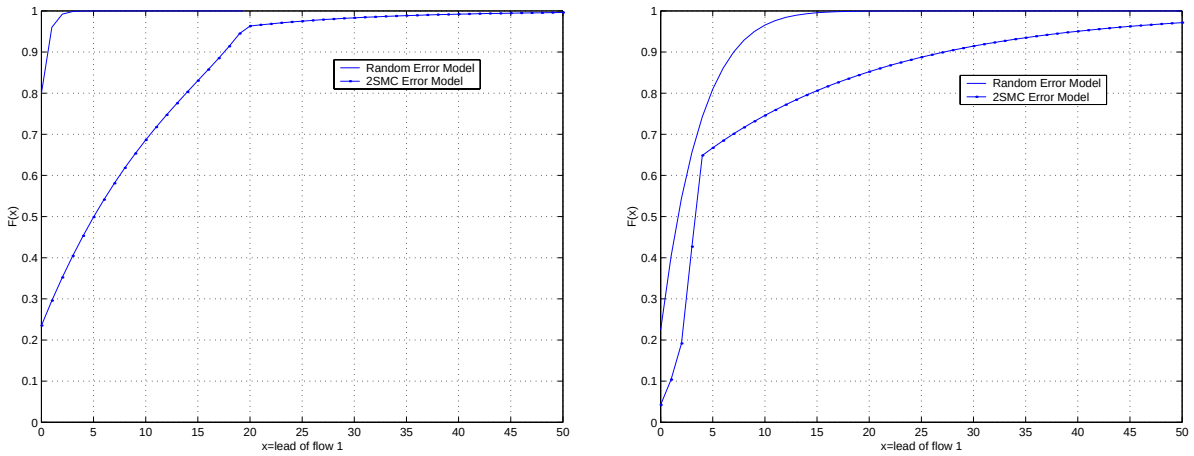
6.2.2 Fairness performance

The fairness distribution, *F(x)*, for two-flow symmetric wireless-fair scheduling in different channels are shown in Fig. 6.3. The mean and standard deviation of *x* are tabulated in Table 6.2.

It is observed that under all channel conditions, the algorithm retains fairness properties significantly better in an uncorrelated channel. For example, in Fig. 6.3(a), while the disparity between the cumulative service received by both flows is within 2 slots with a probability of 0.99 in an uncorrelated channel, the corresponding figure is 37 slots in a correlated channel.

Channel state / Error Model	p_E = 0.2		p_E = 0.8	
	mean	std	mean	std
REM	0.25	0.56	2.71	2.79
2SMC EM	7.65	8.22	13.63	15.98

Table 6.2: Mean and std of *x* for symmetric two-flow wireless-fair scheduling



(a) (b)

Figure 6.3: Fairness Distribution, F(x), for two-flow wireless-fair scheduling for (a)P_B = 0.2 (b) P_B = 0.8

Let us consider the scenario where x_{i-1}=0 and a packet departure in slot *i* results in $x_i \neq 0$ (i.e., either f1S2 or f2S1 occurs). If f1S2 occurred in slot *i*, x_i=1 and the next slot will be allocated to S2. In a correlated channel, the probability of f1S2 occurring in slot *i*+1 (resulting in x_{i+1}=2) is (1-p_{ge}) $\times$ $(1-p_{eg}) \sim 1$ and hence, there is a high likelihood that *x* will deviate from 0. In the case of an uncorrelated channel, the corresponding probability is (1-p_E) $\times$ $p_E << 1$ and hence, there is less tendency to deviate from 0. A similar explanation is applicable for the case where f2S1 occurred in slot *i*. Hence, under this scheduling algorithm, the disparity between the service received by both flows (i.e., reducing 'fairness') is likely to be higher when the channel is correlated.

6.3 Performance Comparison of Wireless Scheduling Algorithms over Random Error Model

6.3.1 Channel efficiency

The delay distribution, *G(n)*, for the Wired-Fair, Channel-Efficient and Wireless-Fair Scheduling algorithms for p_E = 0.2 and 0.8 are shown in Fig. 6.4(a) and (b) respectively.

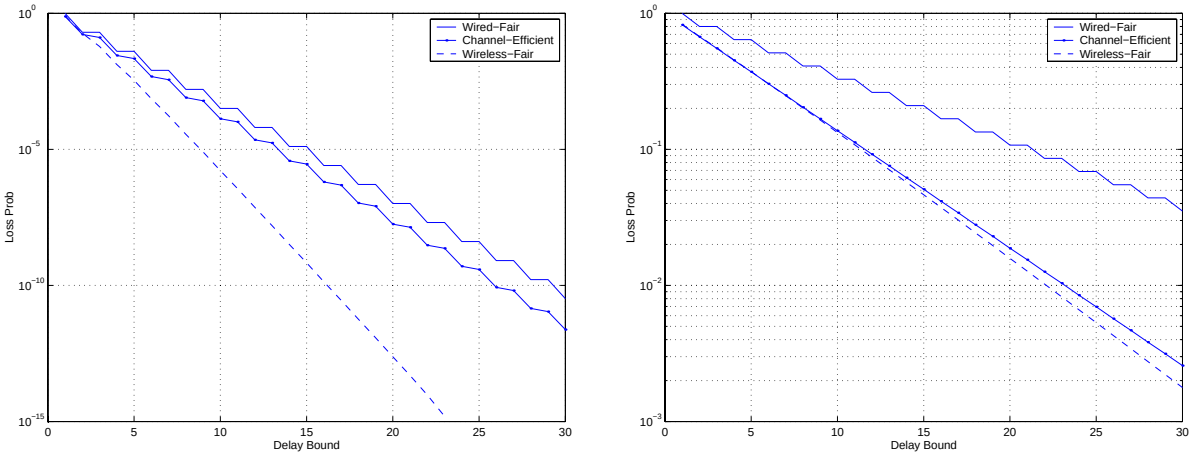
Under both error conditions, the Wireless-Fair algorithm performs better than the Channel-Efficient algorithm, which in turn performs better than the Wired-Fair algorithm.

For example, under low error conditions, referring to Fig. 6.4(a), a delay-sensitive flow with a delay bound of 10 slots will suffer a loss rate of between 10^{-3} to 10^{-4} with the first two algorithms, while the corresponding figure for the Wireless-Fair algorithm is 10^{-6}. On the other hand, for a loss-sensitive flow with a loss rate of 10^{-5}, the respective maximum delay are 13 to 15 slots and 9 slots.

6.3.2 Fairness

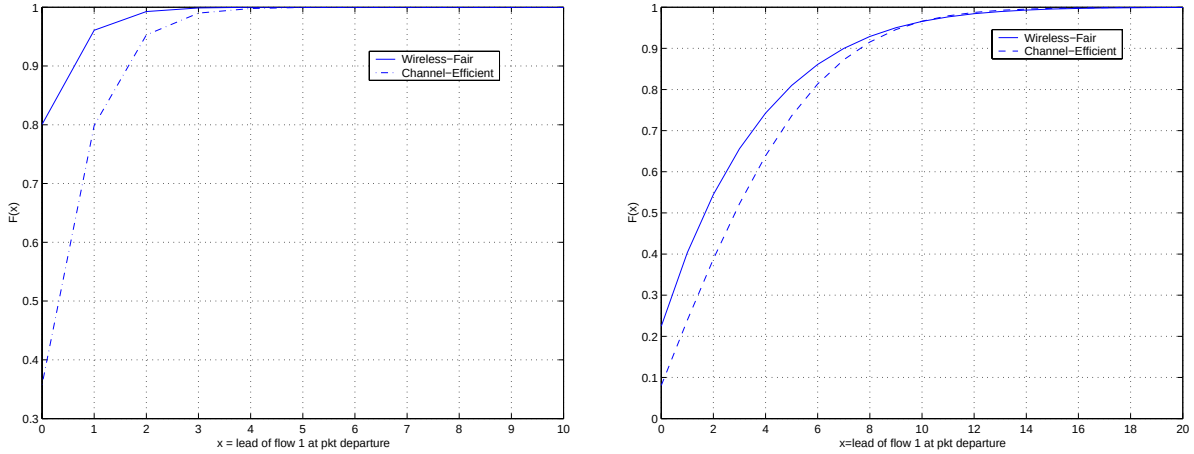
The fairness distribution function, F(x), obtained with each scheduling algorithm is plotted in Fig. 6.5 for (a) p_E=0.2 and (b) p_E=0.8 respectively.

Based on Fig. 6.5(a), under low error conditions, the disparity between the service received by each flow is between 2 to 4 slots with a probability of 0.99. On the other hand, under high error conditions, as observed in Fig. 6.5(b), the disparity falls within 12 to 14 slots with the same probability. This is because when a flow is lagging, it is harder for it to transmit successfully in order to reduce its lag when the channel conditions worsen.



(a) (b)

Figure 6.4: Delay distribution, G(n), for various scheduling algorithms for (a) P_E = 0.2 and (b) P_E = 0.8 under Random Error Model



(a) (b)

Figure 6.5: Fairness distribution, F(x), for various scheduling algorithms for (a) p_E = 0.2 (b) p_E = 0.8 under Random Error Model

For the Wireless-Fair algorithm, under high error conditions, this implies that a lagging flow may be allocated up to 12 consecutive slots in order for it to reclaim its 'lost' slots (fairness), thus 'starving' the other flow (separation) over this duration. Hence, the trade off between fairness and separation is significant under high error conditions.

Between the Channel-Efficient and the Wireless Fair algorithm, the Channel-Efficient algorithm is 'less fair'. This is because in the Wireless-Fair algorithm, slots are allocated such that priority is always given to the lagging flow to allow it to 'recover'; in the Channel-Efficient algorithm, slots are allocated independent of the channel conditions, i.e., the lead/lag status of the flows.

6.4 Performance Comparison of Wireless Scheduling Algorithms over 2SMC Error Model

6.4.1 Channel efficiency

The delay distribution function, G(n), obtained with each scheduling algorithm is plotted in Fig. 6.6 for (a) p_B=0.2 and (b) p_B=0.8 respectively.

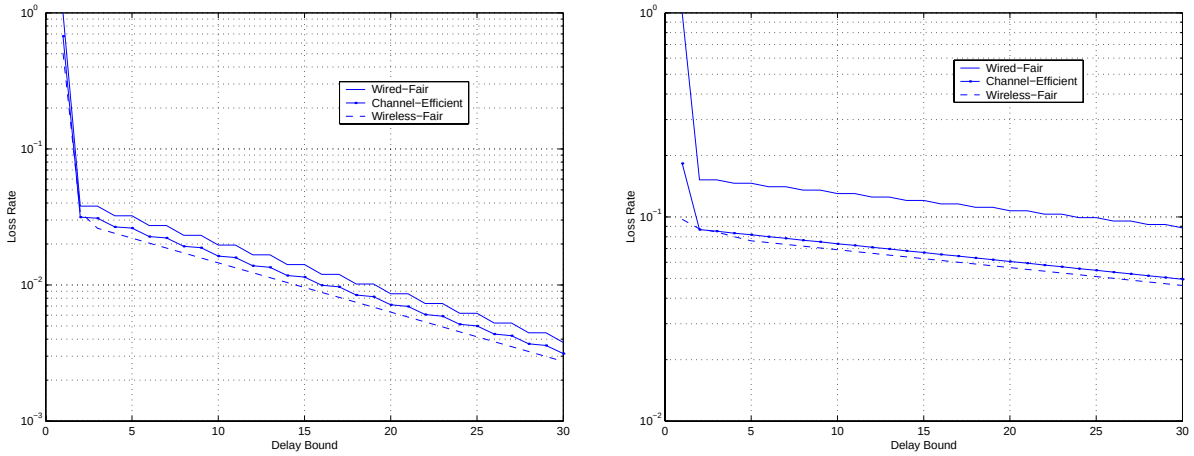


Figure 6.6: Delay distribution, G(n), for various scheduling algorithms for (a) P_B = 0.2 and (b) P_B = 0.8 under 2SMC Error Model

Under all conditions, the Wireless-Fair algorithm performs better than the Channel-Efficient algorithm, which in turn performs better than the Wired-Fair algorithm. However, the difference in performance under low error conditions is marginal.

This is because if we assume that flows transmit mainly in slots allocated to them under low error conditions, then the 3 algorithms are actually equivalent. However, flow switching becomes significant when error conditions are degraded, and hence, the gain in performance becomes visible.

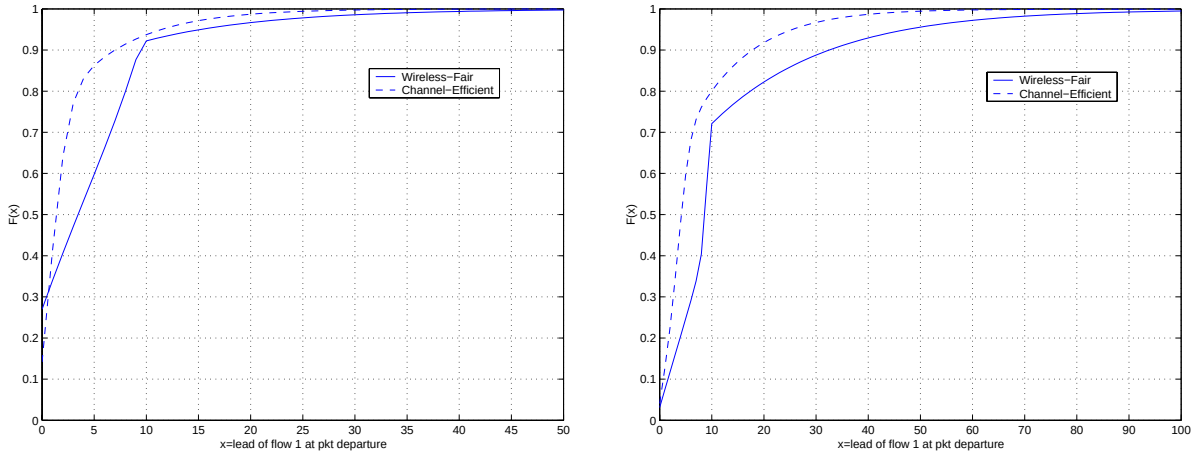
6.4.2 Fairness

The fairness distribution function, F(x), obtained with each scheduling algorithm is plotted in Fig. 6.7 for (a) p_B=0.2 and (b) p_B=0.8 respectively.

Based on Fig. 6.7, it is observed that instead of enhancing the fairness of the Channel-Efficient Algorithm, the wireless adaptation scheme employed in the Wireless-Fair algorithm actually results in a degradation in fairness. For example, under low error conditions, (i.e., Fig. 6.7(a)), with the Channel-Efficient algorithm, the disparity between the service received by each flow is about 12 slots with a probability of 0.95; for the Wireless Fair algorithm, the corresponding figure is about 15 slots.

The above observation can be explained as follows. Let us assume that flow 1 becomes lagging at the end of slot *i*. This implies that its channel was in error in slot *i* while flow 2 enjoyed an error-free channel. In the Wireless-Fair algorithm, the scheduler persistently allocates slots to the lagging flow to allow it to 'recover'. Hence, $A_{i+1} = S1$ and owing to the burstiness of the channel, the channel states are likely to remain the same in slot *i*+1 as in slot *i*. Hence, flow 2 is highly likely to extend its lead over flow 1 in slot *i*+1, resulting in $A_{i+2} = S1$ and the above cycle repeats itself, further stretching the disparity between the flows.

On the other hand, in the Channel-Efficient algorithm, since flow 2 can only extend its lead by transmitting in an S1 slot, this probability is lower since slots are allocated alternately, i.e.,



(a) (b)

Figure 6.7: Fairness distribution, F(x), for various scheduling algorithms for (a) p_B = 0.2 (b) p_B = 0.8 under 2SMC Error Model

...$S1$, $S2$, $S1$, $S2$, .. instead of ..$S1$, $S1$, $S1$, .. as described for the Wireless Fair algorithm. Hence, the wireless adaptation scheme does not perform well in terms of fairness in a highly correlated channel.

Chapter 7

Conclusions and Future Directions

Several wireless-fair scheduling algorithms have been proposed in the literature, which are based on adapting wireline-fair scheduling algorithms to the wireless channel. The wireless adaptation mechanism comprises *flow switching* to optimize channel efficiency and *fairness compensation* to retain the fairness properties offered by the wireline algorithm.

In this report, we have developed an analytical model for a wireless-fair scheduler for the simplest case of a symmetric two-flow scheduling scenario. By appropriately choosing the interval and time instances, the symmetric two-flow scheduler can be characterized stochastically as a one dimensional Markov Chain.

Based on the analytical model, we have derived the performance of the wireless-fair scheduling algorithm under different channel conditions. We have considered both uncorrelated as well as correlated channel error models in our performance evaluation. In addition, In order to benchmark its performance in terms of channel efficiency and fairness under different types of channel, we have defined a *wired-fair scheduler* (which is equivalent to the wireless-fair scheduler without the wireless adaptation mechanism) and a *channel-efficient scheduler* (which is equivalent to the wireless-fair algorithm without the fairness compensation component of wireless adaptation).

Results indicate that for all channel types, we obtain the expected gain in channel efficiency for the Wireless-Fair and Channel-Efficiency algorithms over the Wired-Fair algorithm. In terms of fairness, when the channel errors are uncorrelated, the Wireless-Fair algorithm performs better than the Channel-Efficient algorithm. However, the fairness compensation mechanism fails to retain fairness properties well when the channel is correlated.

The analytical model can be extended to the more general case of asymmetric two-flow wireless-scheduling, where $r_1 \neq r_2$. The performance evaluation obtained for such a scheduler with $r_1 = \frac{1}{3}$ may provide some insight into the best-case performance of a three-flow symmetric scheduler. This may be extrapolated to an N-flow symmetric scheduler by analyzing the asymmetric scheduler with $r_1 = \frac{1}{N}$.

In this analysis, we have assumed that channel errors are spatially independent in order to attribute any 'unfairness' obtained solely to the mechanism of the wireless scheduler. Ongoing work seeks to address the 'unfairness' introduced by location-dependent errors, where different flows may perceive different channel conditions at the same time.

Bibliography

[1] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm," *Proc. of the ACM SIGCOMM*, pp. 3–12, September 1989.

[2] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," *Proc. of the ACM SIGCOMM*, pp. 19–29, September 1990.

[3] D. Kandlur, K. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," *Proc. of the Intl Conf. Distributed Computer System*, pp. 300–307, May 1991.

[4] T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair queuing algorithms," *Proc. of the ACM MOBICOM*, pp. 132–142, October 1999.

[5] Y. Cao and V. Li, "Scheduling algorithms in broad-band wireless networks," *Proc. of the IEEE*, vol. 89, pp. 76–87, January 2001.

[6] T. Ng, I. Stoica, and H. Zhang, "Packet fair queuing algorithms for wireless networks with location-dependent errors," *Proc. of the IEEE INFOCOM*, vol. 3, pp. 1103–1111, March 1998.

[7] S. Lu, T. Nandagopal, and V. Bharghavan, "A wireless fair service algorithm for packet cellular networks," *Proc. of the ACM MOBICOM*, pp. 10–20, October 1998.

[8] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control - the single node case," *IEEE/ACM Trans. on Networking*, vol. 1, pp. 344–357, June 1993.

[9] S. Golestani, "A self-clocked fair queuing scheme for broadband applications," *Proc. of the IEEE INFOCOM*, vol. 2, pp. 636–646, June 1994.

[10] P. Goyal, H. Vin, and H. Chen, "Start-time fair queuing: A scheduling algorithm for integrated service access," *Proc. of the ACM SIGCOMM Computer Communications Review*, vol. 26, pp. 157–168, October 1996.

[11] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *Proc. of the ACM SIGCOMM*, pp. 63–74, August 1997.