Quality of Service Provisioning in Wireless Networks

Research Thesis

Submitted in Partial Fulfillment of the
Requirements for the
Degree of Doctor of Philosophy

Hwee Pink Tan

Submitted to the Senate of
The Technion - Israel Institute of Technology
Elul 5764 Haifa August 2004

This research was done under the supervision of Professor Raphael Rom and Professor Moshe Sidi, in the Department of Electrical Engineering. The generous financial help of the Technion is gratefully acknowledged. Without the support of many, I might not have survived this voyage to the end of a new beginning.

First and foremost, my heartfelt gratitude goes to Raphi, who could not have been more patient and supportive during my transition to the new environment. Through his guidance, I picked up the fundamental and yet essential skills of defining research problems, in addition to solving them. His tough stance and demanding questions, which I initially dread with apprehension, helped build up my confidence.

Despite his quiet demeanor, I never fail to be amazed by the wealth of knowledge Moshe possesses. Despite his busy schedule, Moshe had always made time for insightful discussions, which often led to new perspectives and directions. My thanks also go to my thesis readers, Professors Uri Yehiali, Arie Orda and Adam Shwartz.

These years would not have been such a joy without my friends, whose company is a constant reminder that there is more to life than a thesis. First, thanks to Omer, Guy and Yoav for the foolsball sessions. Thanks also to Yuheng and Mei for creating a home away from home for me with your company and your sumptuous home-cooked meals. To my friends Philip, Daniel and Chung Hou back in Singapore, your emails and SMSs helped me tide through moments of despair - thanks for being there for me.

Last and most importantly, my deepest gratitude to my parents, whose constant support made all this possible. Finally, to Tzvika, who has stood by me through all these years with unyielding emotional support and provided me with a warm and cosy home - you are truly the wind beneath my wings.

Contents

List of Figures

Abstract

Wireless scheduling plays an important role in the design of advanced wireless networks as it determines the QoS provisioning over the wireless link. Compared to its wired counterpart, the design of wireless schedulers is a much harder and challenging problem due to the unique characteristics of the wireless channel. While recent work focused on the design of wireless schedulers to meet a given performance objective, our research aims to characterize the QoS performance of a generic wireless scheduler in a downlink centralized scheduling scenario.

We characterize each input flow in terms of its time-fraction requirement (the fraction of resources that it should be allocated) and its channel agility (the likelihood of transition from one channel state to another across successive time slots). These parameters characterize each class in a class-based scheduling scenario. We define a channel-independent scheduler that aggregates all the input flows prior to scheduling, and a generic channel-state dependent wireless-fair scheduler that can be abstracted into various functional components.

In particular, the slot allocation policy determines the QoS performance of the wireless-fair scheduler under error-free conditions. We analyze the allocation periodicity of perfectly-fair (i.e., allocated share = time-fraction requirement) loop schedulers, based on which we propose a recursive implementation. On the other hand, the arbitration scheme selects a flow for transmission in

order to emulate the slot allocation policy under error-prone conditions. We propose novel schemes that assign a priority to each flow instead of uniform arbitration.

Since we assume a Markov-based wireless channel, we model the mechanism of each scheduler as a Markov process. Although the ergodicity of the Markov model is established for a two-flow scenario, our approach is not scalable in a multiple-flow scenario. To address scalability issues of our basic model, we define an abstraction of the wireless-fair scheduler (channel-state dependent scheduler) that preserves the ergodicity of its corresponding Markov model. We introduce a novel concept of constrained state-transition matrices, and develop an analysis framework from which QoS metrics can be derived without the tedious transient computations of Markov analysis.

Numerical results highlight the performance gain achieved with our proposed slot allocation policy and arbitration schemes for the channel-state dependent scheduler. In addition, for a channel-homogeneous scenario, we observe that while flow aggregation achieves better QoS performance in an uncorrelated channel, the channel-state dependent scheduler is superior in a persistent channel. Hence, for a channel-heterogeneous two-class scheduling scenario, we propose a novel hybrid scheduler that partitions the flows according to their channel agility before applying the respective scheduling mechanism to each partition. The proposed scheduler achieves good overall throughput as well as low wireless receiver buffer requirements compared to its component schedulers. This stresses the importance to exploit the long-term error behavior (channel agility), in addition to the instantaneous channel state in the design of wireless schedulers.

Glossary

Symbols for Transmission Events ($\mathcal{E}$)

$\mathcal{E}^j_{a_i}$	Transmission *e*vent for flow j in slot i
$\mathcal{D}^j_{a_i}$	*D*eferred transmission of flow j in slot i
$\mathcal{F}^j_{a_i}$	*F*ailed or de*f*erred transmission of flow j in slot i
$\mathcal{F}$	No transmission in slot i
$\mathcal{S}^j_{a_i}$	*S*uccessful transmission of flow j in slot i
$[\mathcal{AB}]_{N,x}$	Over an interval of N slots beginning with slot i, $\mathcal{A}$ occurs in the first x slots while $\mathcal{B}$ occurs in the remaining slots
$\mathcal{E}^x$	Event $\mathcal{E}$ occurs in slots i,i+1,i+2,...i+x-1
$(\mathcal{AB})_{N,x}$	Over an interval of N slots, $\mathcal{A}$ occurs in x slots while $\mathcal{B}$ occurs in the remaining slots
$\overline{\mathcal{E}}$	Complement of event $\mathcal{E}$

Symbols for probabilistic parameters

$p_x(Y)$	Prob(x=Y)
p_x	Prob(x=X)
P_X	Prob($x \leq X$)
$p_{[x\|y]}$	Prob(x=X | y=Y)
$p_{c^j}(X\|Y)$	Prob(c^j_i=X | c^j_{i-1}=Y)
$p_{\mathcal{E}}$	Prob(Event $\mathcal{E}$ occurs)
$E[x]$	Expectation of x
σ_x	Standard Deviation of x
$Var[x]$	Variance of x
$\underline{\underline{p}}_{\ddot{c}^K}$	Ensemble channel state transition matrix
γ	Prob($\mathcal{S}^f_{a_i}$ | c^f_i=0)
$(p_{\hat{0}}, p_{\hat{1}})$	One-step Prediction parameters

Symbols for discrete variables, constants and functions

a	Index of *a*llocated flow
$Arb()$	Arbitration function that selects a flow for transmission when the allocated flow is not eligible
b	Wireless receiver *b*uffer size
c	Flow *c*hannel state
$\ddot{c}^K$	Ensemble channel state
C	Number of *C*lasses in class-based scheduling
e	*E*mulated SAP allocation
f	*F*low selected by AS for transmission
g	Channel a*g*ility
h	Priority Thres*h*old for Prioritized Arbitration Scheme
i	slot index
j	flow index
κ	Number of flows in each class
K	Total number of flows
l	*L*ead parameter
m	packet index
n	Head-on-Li*n*e packet delay
p	*P*robabilistic parameter
q	Number of packets in the wireless receiver *q*ueue at packet arrival
r	Time-fraction *r*equirement
R	Loop size ($= \sum_{j=1}^{K} r^j$)
s	*S*ervice time at wireless receiver buffer
u	Co*u*nt variable used to implement SAP
v	Priority le*v*el associated with prioritized arbitration scheme
w	*W*aiting time at the wireless receiver queue

Notations associated with variable x

$\underline{x}$	vector associated with variable x
$\underline{\underline{x}}$	matrix associated with variable x
$\hat{x}$	prediction of x
x_{min}	minimum value of x
x_{max}	maximum value of x
$\check{x}$	average of x
$\tilde{x}$	class-based notation for x

Symbols for QoS-related parameters

α	Acceptable packet dropping rate due to delay bound violation
β	Acceptable packet dropping rate due to wireless receiver buffer overflow
ρ	Utilization factor at the wireless receiver
τ	Time interval in slots
N_{max}	Upper bound on HOL delay
t_τ^j	Number of successful flow j transmissions in the interval [0,τ]
T^j	Throughput of flow j
Δ_τ^j	Throughput fluctuation for flow j in the interval [0,τ]
$FM_\tau^{j,k}$	Fairness metric between flow j and flow k over the interval [0,τ]

Symbols for sets

C	Indices of flows that belong to a particular class
F	Set of Perfectly-Fair Loop Schedulers
G	Set of Eligible flows
L	Set of most lagging eligible flows

Abbreviations

2SMC	Two-State Markov Chain
3G	Third Generation Wireless Systems
3GPP	Third Generation Partnership Project
AP	Access Point
AS	Arbitration Scheme
CRR	Credit Round Robin Scheduler
CSD	Channel-State Dependent
CSM	Channel Status Monitor
DISP	Packet Dispatcher
DRR	Debit Round Robin Scheduler
FA	Fair Aggregation
FIFO	First-In-First-Out
FM	Fairness Monitor
GPS	General Processor Sharing
GR	Golden Ratio Scheduler
HOL	Head-of-Line
OPT	Optimal two-class loop scheduler
OSP	One-step Prediction
PA	Prioritized Arbitration
PCK	Perfect Channel Knowledge
pdf	probability density function
QoS	Quality of Service
REC	Recursive loop scheduler
RND	Random Loop Scheduler
RR	Round Robin
SAP	Slot Allocation Policy
SCED	Service Curve Earliest Deadline
STF	Short-term Fair Loop Scheduler
UA	Uniform Arbitration
UWFQF	Unified Wireless Fair Queueing Framework
WFQ	Weighted Fair Queueing
WFS	Wireless Fair Scheduler
WRR	Weighted Round Robin
WRR-sp	WRR with spreading

Chapter 1

Introduction

The importance of mobile connectivity along with the popularity of the Internet is fuelling the development and roll-out of Third Generation (3G) wireless systems and Wireless local area networks (WLAN). While WLANs provide coverage over small areas with high density of demand for high-data-rate wireless services requiring limited mobility, 3G wireless systems are more suited to wide-coverage areas with moderate or low-density demand for wireless usage requiring high mobility. Since these systems are complementary, the integration of 3G wireless and WLANs will be highly significant in making wireless multimedia and other high-data-rate services a reality for a large population. Ongoing work within the 3G Partnership Project (3GPP) seeks to integrate both systems in order to make multi-access solutions effective.

We consider a generic wireless network (common to both 3G wireless systems and WLANs) as depicted in the left hand side of Fig. 1.1, where an access point (AP) is the interface that links wireless receivers to a wired network. All communication that is initiated by or terminated at any wireless receiver is coordinated by the AP. We envisage a demand for such a network to deliver wireless services from the wired network to the wireless receivers (downlink). Examples of such services include multimedia messaging, voice over WLAN and localized content distribution such as maps, newspapers and entertainment guides. In order to be meaningful, the data traffic associated with these services must be delivered to the wireless receivers at specific

data rates and/or within specific delay, packet loss and jitter bounds. These requirements can collectively be termed Quality of Service (QoS).

The end-to-end QoS provision for the wireless network depicted in Fig. 1.1 comprises the end-to-end QoS provision over the wired part of the network and the QoS provision over the last hop, i.e., the wireless link. Traffic management schemes such as call admission, routing, flow control and packet multiplexing (or scheduling) have been developed that guarantee end-to-end QoS constraints over wired networks. On the other hand, QoS provisioning over the wireless link is achieved via admission control, resource allocation and packet scheduling at the AP. Specifically, given the performance requirements (e.g., QoS) and input traffic descriptions (e.g., mean packet arrival rate, burstiness), the wireless scheduler at the AP will allocate wireless channel access to each receiver. Conversely stated, for a given input traffic specification, the design of the wireless scheduler is an important problem in wireless networking for:

- Wireless application development, since it determines the QoS that the network can support;
- Wireless receiver design, since it determines the buffer requirement at each wireless receiver, which is limited due to size and processing power constraints of portable wireless devices.

While the capacity of a wired link is usually assumed to be constant, the following property makes the problem a harder and more challenging one:

Property 1.1 *A typical wireless link is characterized by:*

a. *High channel error rate*

b. *Bursty and time-varying channel capacity*

c. *Location-dependent channel capacity*

In this study, we consider the downlink scheduling problem at each access point, e.g., B, as depicted in the right hand side of Fig. 1.1. Packets (assumed to be fixed-size) arriving at B are queued into K input flows, where flow j comprises packets destined for wireless receiver j. The wireless scheduler allocates fixed-size time slots corresponding to the transmission time of one packet to each flow j according to its time-fraction $r$equirement, r^j, i.e., the proportion of bandwidth that should be allocated to it over a long interval. Hence, any deviation in the actual proportion of allocated bandwidth results in throughput-unfairness. Our objective in this study is to characterize the QoS performance of wireless schedulers in terms of the input-flow and wireless channel characteristics.

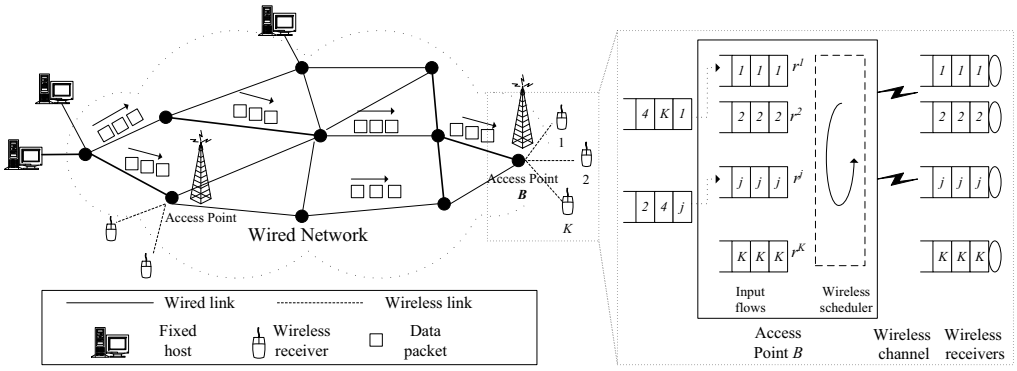

Figure 1.1: A generic wireless network where data packets are delivered to wireless receivers via access points (left) and an illustration of a wireless scheduling problem at an access point B (right).

1.1 Related Work in QoS Provisioning

An abundance of scheduling policies that provide guaranteed QoS for wireline networks exist in the literature, and can be broadly classified under General Processor Sharing (GPS)-based [1, 2] policies and Service Curve-based Earliest Deadline (SCED) [3] policies. A service curve is a function that partially characterizes the service received by a connection at a network element. GPS-based scheduling policies induce a service curve for each connection based on

assumptions regarding the incoming traffic. In contrast, in SCED policies, a service curve is allocated and scheduling policies are then synthesized to support the allocated service curves, independent of any assumption regarding the arriving traffic.

Several works have attempted to extend the provision of guaranteed services by wireline scheduling to wireless links. In [4], the authors studied the delay performance of a simple Automatic Repeat Request (ARQ) error control strategy for communications over a bursty channel for a *single* flow. In [5], the authors modelled a wireless channel in terms of QoS metrics for a single flow. In [6], the author investigated the characteristics and traffic effects of variable-rate communication servers. It is shown that if all input connections to a fluctuation-constrained [7] work-conserving server node are burstiness-constrained [8], deterministic or statistical bounds on queue length and traffic delay in an isolated work-conserving variable-rate server node can be computed as long as the stability criterion is satisfied. In [9], the author considered the allocation of service curves over a time-varying channel modelled by a traffic impairment process. Under SCED scheduling, together with an ARQ policy, service curves can be allocated as long as the traffic impairment process is stochastically upper bounded. The above works are limited since the channel capacity is user-independent. In addition, the scheduling policy is independent of the channel conditions and hence, resources may not be efficiently utilized.

In [10], the authors considered a N-queue, single-server allocation problem, where each queue is characterized by a time-varying connectivity variable (user-dependent channel capacity). At each slot, the allocation decision is based on the connectivity information and on the lengths of the connected queues only. The stability properties of the system are characterized and an optimal policy that maximizes the throughput and minimizes the delay is obtained. However, the results are applicable only for the case where the connectivity variable is uncorrelated across successive time slots. This is an impractical assumption since channel errors are known to be bursty in

nature. In addition, the optimal allocation policies only apply to the special case of symmetric queues (i.e., all queues have similar arrival, service and connectivity statistics).

1.2 Related Work in Wireless Scheduling

Direct application of wireline schedulers to the wireless media is not useful due to the characteristics of the wireless channel defined in Property 1.1. Instead of extending the QoS provisioning capability of wireline scheduling to a wireless link, an alternative approach is to utilize feedback from each wireless receiver to predict its channel state (i.e., whether it is erroneous or error-free). Due to characteristics (b) and (c) in Property 1.1, it is highly likely that at least one flow with an error-free channel exists at any instant. Hence, in the wireless schedulers proposed in [11, 12], channel efficiency can be optimized by restricting the candidates for transmission to those with *predicted* error-free channels. In [13, 14], the authors considered the downlink scheduling problem in a CDMA system. In this case, the channel information is embedded in the measured data rates, and the authors proposed an exponential rule for scheduling that optimizes the throughput. A comprehensive survey of wireless schedulers that differ in the mechanism of selecting the instantaneous 'best' flow to transmit while trading-off amongst various performance constraints such as throughput, fairness and delay can be found in [15].

Most wireless schedulers proposed recently [16, 17, 18, 19, 20, 21, 22] can be mapped onto a Unified Wireless Fair Queuing Framework (UWFQF) [23]. In this framework, a wireless scheduler comprises a wireline scheduler as well as a wireless adaptation scheme. While scheduling is performed using the wireline algorithm under error-free conditions, the wireless adaptation scheme takes over when these conditions no longer prevail.

A common wireless adaptation scheme involves flow swapping and reassignment. With this scheme, when a flow j that is scheduled for transmission

predicts channel errors, another flow k that perceives an error-free channel is selected for transmission. After this swapping, flow j *lags* behind (flow k gains a *lead* over) its time-fraction requirement. The scheduler accounts for the 'lost' transmission opportunity by flow j and attempts to compensate for it at a later time.

In addition to the time-fraction requirement, these schedulers seek to achieve a trade-off between channel efficiency and short-term fairness provision, which is determined by the extent to which flow swapping is permitted. If flow swapping is unbounded, a flow could lag behind another flow by a huge amount due to an error burst (poor short-term fairness). However, if flow swapping is restricted, the scheduler will achieve short-term fairness at the expense of reduced channel efficiency due to 'wasted' slots when no flow transmits.

1.3 Limitations in existing QoS Analysis of Wireless Schedulers

The main contribution in recently proposed wireless schedulers lies in the design of the wireline scheduler as well as the wireless adaptation scheme in order to achieve trade-offs amongst various QoS metrics as well as fairness. Most of the works have relied on simulations to evaluate the performance of the schedulers. Although analytical QoS bounds for some wireless schedulers are derived and presented in [23], they suffer from the following limitations:

1.3.1 Error-free Conditions

Under error-free conditions, the wireless scheduler reduces to its wireline scheduling component. Hence, the QoS bounds for the wireless scheduler derived under such conditions correspond to that of the wireline scheduler. Since these bounds are likely to be degraded under practical channel conditions, and the extent of degradation depends on the channel characteristics, they can only serve as optimistic bounds.

1.3.2 Worst-case Error Rate

In the case of the Wireless-Fair Service [17] scheduler, QoS bounds are derived under error-prone conditions, which are more meaningful for assessing the performance of the scheduler. As an example, on average, *all* Head-of-Line (HOL) packets for a *lagging* flow j, with time-fraction requirement r^j, given the *worst-case* channel error rate of $p_c(1)$, will be successfully transmitted within n^j slots, where

$$n^j \leq (\sum_{j \in \mathbb{F}} \frac{L}{C}) + \frac{L}{Cr^j} \frac{1}{(1 - p_c(1))} \tag{1.1}$$

where $\mathbb{F}$ = set of all flows, C = channel capacity in bits/sec and L = packet size in bits/packet.

Since wireless channel errors are typically bursty in nature, they cannot be sufficiently characterized by a worst-case error rate. Hence, the bound given in Eq. (1.1) will be too conservative for practical wireless channels that conform to Property 1.1, leading to sub-optimal use of scarce radio resources.

1.3.3 Guaranteed QoS Provisioning

Certain applications, e.g., continuous media, with strict delay constraints, can typically tolerate some losses. Therefore, guaranteed QoS constraints (such as Eq. (1.1)) may be overly conservative for such applications. This prompted research on providing statistical QoS guarantees (e.g., [24] and references therein). Instead of specifying a single QoS constraint ν that is satisfied by *all* packets, statistical QoS provisioning bounds the fraction of packets, ω, that violates the QoS constraint, ν. In order to guarantee end-to-end QoS over a wired network, users negotiate with the network in order to limit the amount of traffic (or traffic regulation) they can send over an interval of time. The statistical QoS guarantees are usually expressed as a function of the regulator function (or arrival envelop). Extending these guarantees to include a wireless last-hop is non-trivial as described in Section 1.1.

1.4 Research Goals

The above evidence emphasizes the need for further work on a more complete characterization of the QoS performance of wireless schedulers. This forms the goal of our research, which we achieve through the following sub-tasks (See Fig. 1.2):

Definition and Modelling : We define the scope of our scheduling problem (see Fig. 1.1) in terms of the scheduling scenario, the wireless scheduler, the wireless receivers and desired QoS metrics. (Chapter 2)

Performance Analysis : We derive the performance metrics for a wireless scheduler in terms of the scheduling scenario. (Chapter 3 and 4)

Numerical Results and Applications : We use numerical results to study the trade-offs amongst scheduler design, receiver design as well as QoS metrics for a given scenario, and also to suggest enhancements to the wireless scheduler design. (Chapter 4, 5 and 6)

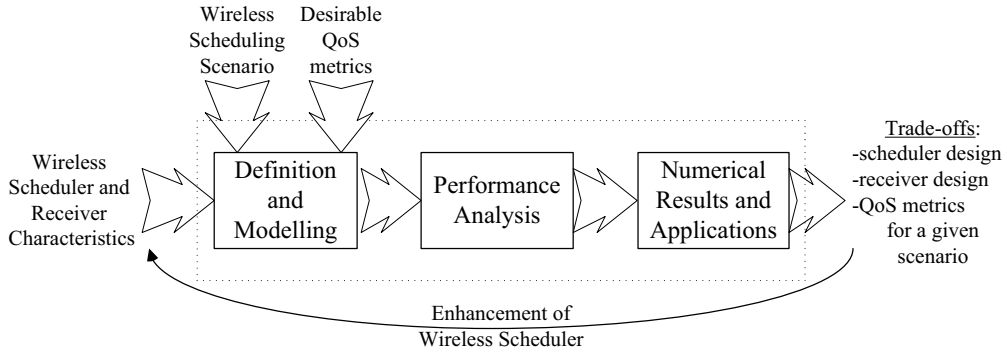

Figure 1.2: Specification of our research tasks.

1.5 Research Contributions

In this study, we consider a class-based scheduling scenario, where each class is specified by the traffic and channel characteristics of each flow that belongs

to it. Such a scenario admits a simple and efficient scheduler that caters to the requirements of diverse applications. Contrary to previous work that focused on wireless scheduler design to achieve a certain performance objective, we consider a generic scheduler model where different abstractions of each component can be defined to fulfil different performance objectives.

We introduce the notion of constrained state-transition matrices, from which we develop a performance analysis framework to derive the delay probability density function (pdf) while skipping tedious computations of transient Markov analysis. Unlike previous work that yield first-order deterministic metrics, our framework permits a wider spectrum of statistical performance metrics (including higher-order metrics) to be computed.

Such a QoS characterization of the wireless scheduler is important for the design of wireless applications and wireless receivers, which is illustrated in terms of the admissibility of the wireless scheduler under QoS (efficiency and real-time) constraints as well as wireless receiver buffer requirement. While most wireless schedulers utilize short-term channel information, we highlight the importance of long-term channel information in wireless scheduler design.

1.6 Organization of the Thesis

The remainder of the thesis is organized as follows. In Chapter 2, we define a Two-State Markov Model for the channel of each flow. The channel parameter, together with the time-fraction requirement, of each flow characterize each class in a class-based scheduling scenario, which is x-homogeneous if all classes are indistinguishable with respect to x. In addition, we define a generic wireless scheduler model as well as performance metrics that can be expressed in terms of the HOL packet delay pdf.

In Chapter 3, we define various abstractions of the scheduler model and also define a Fair Aggregation (FA) scheduler that aggregates the packets into a single flow fairly prior to FIFO transmission. We perform a Markov

analysis for a two-flow scenario to derive the HOL packet delay pdf and verify the ergodicity of various scheduler models. Numerical results illustrate the trade-offs amongst various QoS metrics for each scheduler, as well as amongst different schedulers in a homogeneous scenario. We also highlight the limitation of our modelling approach for a Wireless-Fair Scheduler (WFS) in a multiple-flow scenario.

In Chapter 4, we overcome the above scalability limitation by disabling the fairness module of the WFS, and propose various schemes to embed its functions within other components of the resultant Channel-State Dependent (CSD) scheduler. We develop a performance analysis framework based on the notion of constrained state-transition matrices to analyze the CSD scheduler. We also establish the relationship between the wireless receiver buffer requirement and the QoS metrics of the wireless scheduler. Numerical results highlight the trade-offs amongst QoS, efficiency and wireless receiver buffer requirement for a CSD scheduler in a channel-homogeneous scenario.

We propose a novel idea of a hybrid CSD-FA scheduler in Chapter 5 that exploits the long-term channel behavior in addition to the instantaneous channel state. We apply the framework developed in Chapter 4 for the performance analysis of the hybrid scheduler. Numerical results verify the performance gain obtained relative to a regular CSD scheduler in a two-class channel-heterogeneous scenario.

Next, we address the effects of input-heterogeneity on the design of the CSD scheduler in Chapter 6. We define the periodicity of allocation to each flow as the optimization criteria, and evaluate the periodicity characteristics of various known loop schedulers. Based on the analysis, we propose a recursive implementation which achieves good periodicity performance in an input-heterogeneous class-based scenario.

Finally, Chapter 7 summarizes the research and integrates the results presented in the previous chapters. In addition, possible extensions to the research are suggested.

Chapter 2

Problem Definition and Modelling

In this chapter, we define and model each component of our scheduling problem as depicted in the RHS of Fig. 1.1. This comprises the wireless scheduling scenario (input-flows and wireless channel), the wireless scheduler and the wireless receiver. In addition, we also define the QoS performance metrics considered in this study.

2.1 Wireless Scheduling Scenario

We specify the wireless scheduling scenario by the traffic and channel characteristics of each flow. We define the respective models, and propose a class-based scheduling scenario in the following.

2.1.1 Input-Traffic Model

Each AP comprises K input queues, one corresponding to each wireless receiver within its coverage area. Hence, each packet that arrives at the AP (assumed to be fixed-size with transmission time of one slot) is dispatched into the queue that corresponds to its destination wireless receiver, as depicted in the RHS of Fig. 1.1.

Each input queue (or flow) j can be specified in terms of its packet arrival statistics, its buffer size and its time-fraction requirement, r^j. In this study, we ignore the consideration of the former characteristics with the following assumptions for each input flow:

- Continuous backlog
- Infinite buffer size

Without loss of generality, we assume that $r^j \leq r^k$ if $j < k$.

2.1.2 Wireless Channel Model

Since the performance of a wireless scheduler is influenced by the channel characteristics, it is pertinent to define the channel model considered in our study. A typical channel model that captures the characteristics defined in Property 1.1 is the Gilbert-Elliott channel [25], where the channel state $c_i^j \in \{0,\ 1\}$ behaves according to a stationary Two-State Markov Chain (2SMC). The state-transition diagram for the channel model is given in Fig. 2.1. The wireless receivers are assumed to be sufficiently separated spatially such that the channel states of different flows are independent.

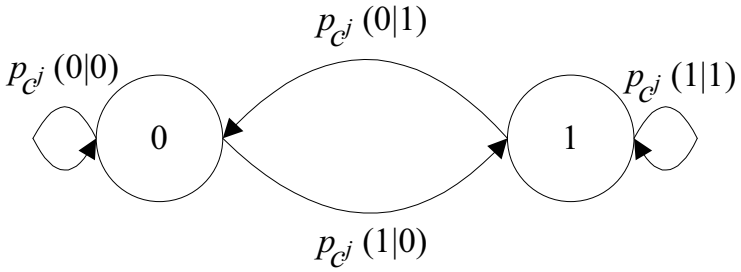

Figure 2.1: State-transition diagram of a Two-State Markov Chain (2SMC) error model for flow j at slot i.

We specify the channel model in terms of ($\underline{p}_c(0)$=[$p_{c^1}(0)$,$p_{c^2}(0)$,$\cdots$,$p_{c^K}(0)$], $\underline{g}$=[g^1,g^2,$\cdots$,g^K] and $\underline{\gamma}$=[γ^1,γ^2,$\cdots$,γ^K]), which are defined as follows:

γ^j **:** When c_i^j=1 (*bad* channel), any attempted transmission by flow j in slot i always fails; on the other hand, when c_i^j=0 (*good* channel), the corresponding probability of a successful transmission is 1-γ^j. We assume that $\gamma^j = \gamma$, $1 \leq j \leq K$, in this study.

$p_{c^j}(0)$ **:** $p_{c^j}(0)$ denotes the steady-state probability of the channel of flow j being in state 0 and is an indication of the quality of the channel. It varies according to the distance of wireless receiver j from the AP. We assume that the coverage area is sufficiently small so that the channel quality of all flows are identical, i.e., $p_{c^j}(0)$=$p_c(0)$ >0 (otherwise, transmissions will never occur).

g^j **:** g^j indicates the level of a*g*ility of the error behavior across successive slots for flow j, and varies according to the mobility of wireless receiver j as well as its environment. For small ϵ, we can categorize the channel according to g^j as follows:

$$g^j = \begin{cases} \epsilon, & \text{Persistent channel;} \\ 1, & \text{Uncorrelated channel;} \\ 2-\epsilon, & \text{Oscillatory channel.} \end{cases}$$

We define the decimal equivalent of the binary sequence $c_i^K c_i^{K-1} \cdots c_i^1$ (denoted by $\ddot{c}_i^K$) as the *ensemble* channel state variable, with state space given by $\{0, 1, 2, \cdots 2^K - 1\}$. Therefore, the corresponding state-transition probability matrix, $\underline{\underline{p}}_{\ddot{c}^K}$, is of dimensions $2^K \times 2^K$ and can be computed, for $K \geq 2$, using the following recurrence relation:

$$\underline{\underline{p}}_{\ddot{c}^K} = \begin{bmatrix} \underline{\underline{p}}_{\ddot{c}^{K-1}} \cdot p_{c^K}(0|0) & \underline{\underline{p}}_{\ddot{c}^{K-1}} \cdot p_{c^K}(1|0) \\ \underline{\underline{p}}_{\ddot{c}^{K-1}} \cdot p_{c^K}(0|1) & \underline{\underline{p}}_{\ddot{c}^{K-1}} \cdot p_{c^K}(1|1) \end{bmatrix} \tag{2.1}$$

where

$$\underline{\underline{p}}_{\ddot{c}^1} = \begin{bmatrix} p_{c^1}(0|0) & p_{c^1}(1|0) \\ p_{c^1}(0|1) & p_{c^1}(1|1) \end{bmatrix}$$

and $p_{c^j}(x|y)$ is the transition probability of c^j from state y to state x. The channel agility, g^j, can be expressed in terms of $p_c(0)$ and $p_{c^j}(0|1)$ as follows:

$$g^j = \frac{p_{c^j}(0|1)}{p_{c^j}(0)}$$

If we define $\underline{p}_{\ddot{c}_i^K} = [p_{\ddot{c}_i^K}(C)]_{C=0}^{2^K-1}$, then, for any $N>0$, we have:

$$\underline{p}_{\ddot{c}_{i+N}^K} = \underline{p}_{\ddot{c}_i^K} \times \prod_{u=1}^{N} \underline{p}_{\ddot{c}^K} \tag{2.2}$$

Although such a model is inadequate for schedulers that employ rate-adaptation schemes for data service, our focus is on delay-sensitive flows where rate-adaptation schemes are not suitable. Recent research also revealed that such a channel model may not be sufficiently accurate for certain fading channels and Markov Chains with more states [26, 27] or higher order [28] have been suggested. However, it is not clear to what extent the accuracy of the model will have an influence on the performance of wireless scheduling. Hence, we perform our analysis based on a 2SMC model, which is also analytically more tractable.

2.1.3 Class-based Scheduling Scenario

A class-based scheduling framework [29] is based on the paradigm of service classes, where flows with a common characteristic are grouped into the same class. This framework is intended to cater to the requirements of diverse networking applications and at the same time, it simplifies scheduling complexity and enables efficient and fair sharing of resources.

We define a general class-based scheduling scenario that comprises C classes. Each class x comprises κ^x flows whose indices are defined by $\mathbf{C}^x$, where

$$\mathbf{C}^x = \{\sum_{y=1}^{x-1} \kappa^y + 1, \sum_{y=1}^{x-1} \kappa^y + 2, \cdots, \sum_{y=1}^{x} \kappa^y\}$$

Let $\tilde{y}$ denote the class-based equivalent of the variable y used to describe flow characteristics. In our study, each class x is characterized in terms of its time-fraction requirement $\tilde{r}^x$ and its channel agility, $\tilde{g}^x$, such that any flow $j \in \mathbf{C}^x$ if and only if $(r^j,g^j) = (\tilde{r}^x,\tilde{g}^x)$. Any class-based scheduling scenario can be specified by the vectors $\underline{\kappa}$, $\underline{\tilde{r}}$ and $\underline{\tilde{g}}$.

For $1 \leq x, y \leq C$, where $x \neq y$, a scheduling scenario is *input-homogeneous* if $\tilde{r}^x$=1, and *input-heterogeneous* if $\tilde{r}^x \neq \tilde{r}^y$; similarly, a scheduling scenario is *channel-homogeneous* if $\tilde{g}^x$=g, and *channel-heterogeneous* if $\tilde{g}^x \neq \tilde{g}^y$. We define a *homogeneous* scheduling scenario as one that is both input- and channel-homogeneous.

We consider the special case of $\underline{\kappa} = [1,\cdots,1]$ in Chapter 3 and 4. In this case, $K = C$, and the scheduling scenario can be specified by the vectors, $\underline{r}$ and $\underline{g}$.

2.2 Wireless Scheduler Model

The generic wireless scheduler model that we consider in this study comprises a Slot Allocation Policy (SAP), Channel Status Monitor (CSM), Fairness Monitor (FM), Arbitration Scheme (AS) and the Packet Dispatcher (DISP), as illustrated in Fig. 2.2. At the beginning of each slot i, the AS assigns a transmission priority to each flow based on the SAP, CSM and FM, and the DISP dispatches the HOL packet of the flow with the highest priority, f_i, for transmission. While the SAP offers QoS performance guarantees under error-free conditions, the AS attempts to emulate its efficiency and fairness performance based on the CSM and FM respectively under error-prone conditions. We describe the mechanism of each component in this section.

2.2.1 Slot Allocation Policy (SAP)

Under error-free conditions, the mechanism of the wireless scheduler is determined by the SAP. Variants of packetized fluid-fair (or weighted-fair) queue-

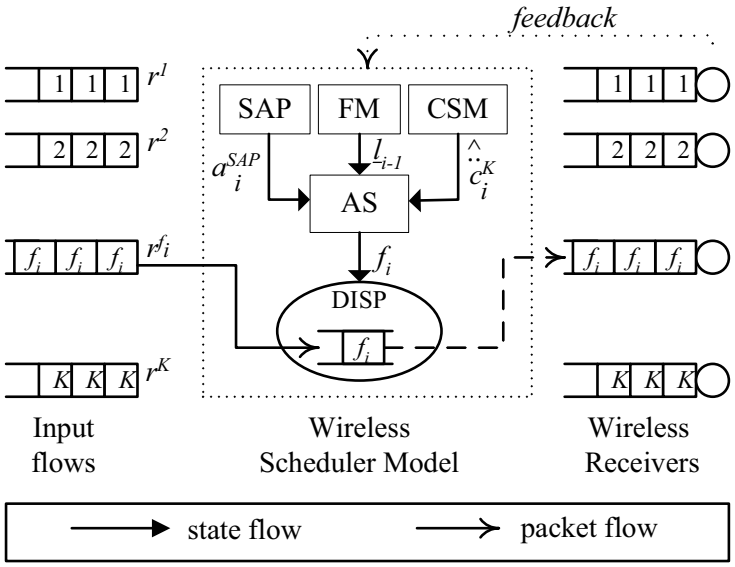

Figure 2.2: Generic wireless scheduler model, with illustration of state flow, downlink packet flow (dashed) and uplink packet flow (dotted) in slot *i*.

ing algorithms (WFQ) [1, 2, 30, 31, 16, 17] are popular choices since they achieve throughput, delay and fairness bounds under error-free conditions. However, we restrict the choice of the SAP to perfectly-fair loop schedulers (denoted by $\mathbf{F}^{\underline{r}}$) as they are simpler to implement and are mathematically tractable. They possess the following properties, where $R = \sum_{x=1}^{C} \kappa^x r^x = \sum_{j=1}^{K} r^j$ is the loop size:

Property 2.1 *If the SAP* $\in \boldsymbol{F}^{\underline{r}}$ *allocates slot* i *to flow* a_i^{SAP}*, then*

- *For any* $i>0$, $a_i^{SAP} = a_{i+R}^{SAP}$;
- *Within any interval of* R *slots,* r^j *slots must be allocated to flow* j*,* $1 \leq j \leq K$.

In fact, for fixed-size packets with transmission time of one slot, if all flows are backlogged at all times during the interval of analysis, then Weighted Round Robin (WRR) with spreading (which can be implemented as a loop scheduler) is equivalent to WFQ [16]. We consider the design and analysis of

perfectly-fair loop schedulers that minimize the HOL packet delay variation in Chapter 6.

2.2.2 Channel Status Monitor (CSM)

The CSM maintains the history of the ensemble channel state based on feedback (see Section 2.2.5) from wireless receivers on the status of each downlink transmission, and uses this information for channel prediction.

Specifically, at the beginning of each slot i, $\ddot{c}_{i-x}^K$, $x>0$ is available and is used to generate the prediction, $\hat{\ddot{c}}_i^K$, of the current channel state, $\ddot{c}_i^K$. We consider a probabilistic one-step predictor (OSP) with parameters ($p_{\hat{0}}$,$p_{\hat{1}}$) defined as follows:

$$\text{Prob}(\hat{c}_i^j = c_{i-1}^j \mid c_{i-1}^j = C) \;=\; \begin{cases} p_{\hat{0}}, & C=0; \\ p_{\hat{1}}, & C=1. \end{cases} \tag{2.3}$$

The predictor parameters ($p_{\hat{0}}$,$p_{\hat{1}}$) are typically close to 1 since most channels are bursty in nature.

For the special case of *perfect* channel knowledge (PCK), we have the following for $1\leq j \leq K$:

$$\hat{c}_i^j \;=\; c_i^j$$

2.2.3 Fairness Monitor (FM)

If transmissions take place according to the SAP, fairness will be maintained. However, due to channel errors, it may be preferable to transmit flow j instead of flow a_i^{SAP} in slot i (flow swapping) in order to minimize 'wasted' slots. When this happens, the time-fraction requirement of both flows are no longer satisfied, resulting in 'unfairness'.

The FM attempts to restore fairness provision by keeping track of flow swapping activities. The notion of per-flow lag (lead) is defined to monitor the

amount of additional throughput that a flow is entitled to (needs to relinquish) in the future in order to compensate for throughput lost (gained) in the past due to flow swapping. This parameter is updated for flows j and k whenever flow swapping takes place between these flows.

If $l_i^{j,k}$ denotes the *l*ead of flow j relative to flow k at the end of slot i, then when flow j transmits in slot i where $a_i^{SAP} \neq j$, $l_i^{j,a_i^{SAP}} = l_{i-1}^{j,a_i^{SAP}} + x(\underline{r})$ while $l_i^{a_i^{SAP},j} = l_{i-1}^{a_i^{SAP},j} - x(\underline{r})$, where $x(\underline{r})$ is a function of $\underline{r}$.

Flow j is defined as *leading, lagging or in-sync (neither leading nor lagging)* at the end of slot i according to the following, where $l_i^j = \sum_{k=1,k\neq j}^{K} l_i^{j,k}$:

$$l_i^j \begin{cases} = 0, & \text{flow } j \text{ is in-sync;} \\ > 0, & \text{flow } j \text{ is leading;} \\ < 0, & \text{flow } j \text{ is lagging.} \end{cases}$$

Since the updating mechanism is symmetric, we have the following property:

$$\sum_{j=1}^{K} l_i^j = 0$$

2.2.4 Arbitration Scheme (AS)

The AS attempts to emulate the performance of the SAP under error-prone conditions based on (a_i^{SAP}, $\hat{\underline{c}}_i^K$,$\underline{l}_{i-1}$). Its mechanism comprises the following:

Eligibility : This component determines which flows are 'eligible' for transmission according to the following criteria:

- **Channel Efficiency :** In order to maximize channel efficiency, a flow j is *eligible* for transmission in slot i only if $\hat{c}_i^j = 0$, since this increases the likelihood of a successful transmission.
- **Fairness :** In order to bound the level of 'unfairness', we define a threshold, l_{max}, such that a flow j is eligible for transmission in

slot i only if $|l_{i-1}^j| \leq l_{max}$. The choice of l_{max} determines the trade-off between channel efficiency and fairness; A large l_{max} implies that a flow could lag behind another flow by a huge amount due to an error burst (unfairness) while a small l_{max} will achieve fairness at the expense of reduced channel efficiency due to 'wasted' slots.

In this study, we consider transmission heuristics that optimize channel efficiency over fairness provisioning. Hence, we choose l_{max} to be a sufficiently large number (but finite) and the set of eligible flows in slot i, $\mathbf{G}_i$, is given as follows:

$$\mathbf{G}_i = \{\arg_{1 \leq m \leq K} \hat{c}_i^m = 0\}$$

Priority Assignment and Selection : This component *assigns* a priority to each eligible flow and *selects* the flow $f_i \in \mathbf{G}_i$ with the highest priority for transmission. We propose various priority assignment schemes in Chapters 3 and 4.

2.2.5 Packet Dispatcher (DISP)

The DISP dispatches the HOL packet of flow f_i for transmission. Under ideal conditions where channel prediction is perfect and γ=0, the transmission will always be successful; however, such conditions do not hold in reality, and hence, packets received erroneously may have to be re-transmitted. The choice of an ARQ mechanism for re-transmission is important since it affects the QoS performance of the wireless scheduler.

In this study, we consider a simple Stop-and-wait ARQ, where a copy of the transmitted packet is stored in a separate buffer in the DISP. The scheduler is notified about the outcome of each transmission through feedback from the wireless receiver, and we assume that all feedbacks are correctly received. With a failed transmission, the packet is enqueued to the HOL of flow f_i for retransmission; otherwise, the copied packet is deleted from the buffer.

2.3 Wireless Receiver Model

Since wireless receivers, e.g., cellular phones and PDAs, need to be portable, they are usually constrained in terms of size. This imposes limitations in terms of memory capacity and battery power, which in turn limits the buffer size and processing power of the wireless receiver. While much research on energy-efficient wireless scheduling (e.g., [32, 33, 34]) focuses on the effects of the latter constraint on the wireless scheduler design, we consider the impact of the QoS performance of the wireless scheduler on the buffer size requirement of the wireless receiver in this study.

Since our scheduling system is time-slotted, a discrete-time $G/G/1$ system can be used to model the queueing behavior at each wireless receiver. Let us designate the variables w^j and s^j as the steady-state *waiting time and service time* (in slots) at wireless receiver buffer j respectively. If the propagation delay over the wireless channel is negligible compared to the inter-arrival time at the wireless receiver, then the latter corresponds to the HOL packet delay, n^j.

If q^j denotes the number of packets in queue j at each packet arrival in steady state, then w^j can be expressed in terms of q^j and s^j as follows:

$$w^j = q^j \cdot s^j$$

The above notations are illustrated in Fig. 2.3.

We establish the buffer size required to sustain an acceptable buffer overflow rate in terms of statistics of n^j and s^j in Chapter 4.

2.4 Definition of QoS Performance Metrics

Let p_{n^j}, $E[n^j]$ and $Var[n^j]$ denote the pdf, mean and variance of n^j respectively. In this section, we define the QoS metrics that are used to quantify the performance of the wireless scheduler, and categorize them into two groups:

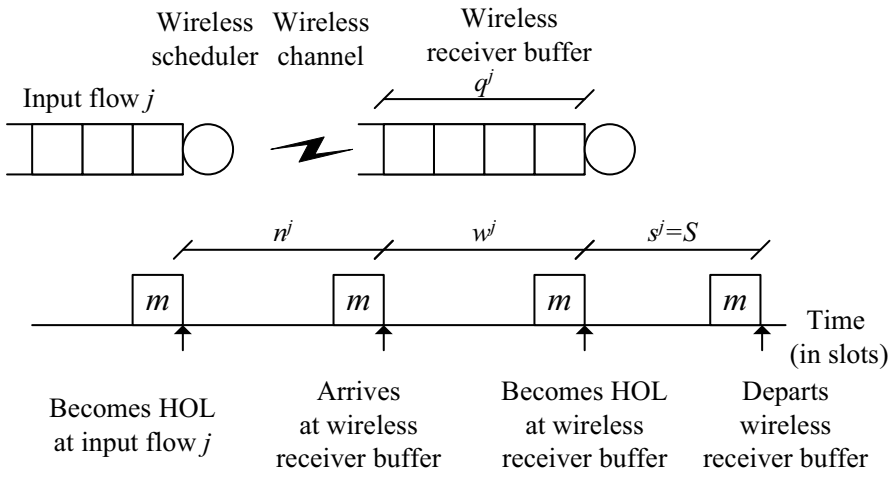

Figure 2.3: Illustration of G/G/1 model for wireless receiver for flow j.

(a) Long-term (or steady-state) and (b) Short-term Performance Metrics.

2.4.1 Long-term Performance Metrics

Overall Throughput

Let t_τ^j denote the number of successful flow j transmissions in the interval $[0,\tau]$. Since the throughput of flow j (denoted T_τ^j) over the same interval is defined as the average number of transmissions per slot, we have:

$$T_\tau^j = \frac{E[t_\tau^j]}{\tau}$$

If T^j denotes the steady-state throughput of flow j, then due to the assumption of continuous backlog in each input flow, we have the following:

$$\begin{aligned} T^j &= \lim_{\tau\to\infty} T_\tau^j \\ &= \frac{1}{E[n^j]} \end{aligned}$$

Since wireless bandwidth is a scarce resource, it is desirable to maximize the overall throughput, T, where

$$\begin{aligned} T &= \sum_{j=1}^{K} T^j \\ &= \sum_{j=1}^{K} \frac{1}{E[n^j]} \end{aligned} \tag{2.4}$$

Throughput-Fairness

We consider the notion of fairness with respect to the individual flow (absolute fairness) as well as a pair of flows (relative fairness).

Absolute Fairness : Any flow j expects that the fraction of service *obtained* within an interval will be as close to the fraction it *should* get as possible. We define our fairness metric in terms of the deviation between the normalized received service and the normalized requested service (or equivalently, the time-fraction requirement) as follows:

$$\Delta_\tau^j = \frac{t_\tau^j}{\tau} - \frac{r^j}{R}$$

For good long-term fairness performance, it is desirable to minimize the expected value of Δ_τ^j. Hence, we compute the first-order statistic, Δ^j, as follows:

$$\begin{aligned} \Delta^j &= \lim_{\tau \to \infty} E[\Delta_\tau^j] \\ &= T^j - \frac{r^j}{R} \end{aligned}$$

Relative Fairness : Between any pair of flows, any flow j expects that it should be fairly treated relative to another flow k. This corresponds to the relative fairness between both flows, which can be evaluated by the

Relative Fairness Bound [30], $FM_{\tau}^{j,k}$, defined as follows:

$$FM_{\tau}^{j,k} = |\frac{\frac{t_{\tau}^{j}}{\tau}}{\frac{r^j}{R}} - \frac{\frac{t_{\tau}^{k}}{\tau}}{\frac{r^k}{R}}|$$

For good long-term fairness performance, we desire the expected value of $FM_{\tau}^{j,k}$ to be as small as possible. Hence, we define and evaluate the following metric:

$$\begin{aligned} FM^{j,k} &= \lim_{\tau \to \infty} E[FM_{\tau}^{j,k}] \\ &= |\frac{T^j}{\frac{r^j}{R}} - \frac{T^k}{\frac{r^k}{R}}| \end{aligned} \tag{2.5}$$

Packet dropping rate due to Delay Bound Violation

Given p_{n^j}, we can write the HOL packet delay distribution function, P_N, as follows:

$$P_N = \sum_{N^j=1}^{N} p_{n^j}$$

We consider real-time applications that carry delay-sensitive traffic such that each packet needs to arrive at the wireless receiver within N_{max} slots after becoming HOL; otherwise, the delay bound is violated and the packet will be 'dropped' by the scheduler. Since 1-$P_{N_{max}}$ denote the probability of delay bound violation, if α is the acceptable packet dropping rate, then the scheduler supports the given real-time application if the following condition holds:

$$1 - P_{N_{max}} \leq \alpha \tag{2.6}$$

Delay Jitter

Jitter is the short-term variation or instability in the duration of a specified time interval, which is the HOL packet delay in our case. Quantitatively, the

HOL packet delay jitter for flow j, σ_{n^j}, is defined as follows:

$$\sigma_{n^j} = \sqrt{Var[n^j]}$$

It is an important QoS metric for multimedia applications since it determines the buffer size, b^j, required at the wireless receiver to ensure continuous playback while keeping packet loss due to overflow within an acceptable level. The relationship between b^j and σ_{n^j} is derived in Chapter 4.

2.4.2 Short-term Performance Metrics

From the description in Section 2.4.1, we can determine the short-term fairness performance by evaluating the metrics, Δ_τ^j and $FM_\tau^{j,k}$, as a function of τ. To do so, we need to evaluate the distribution of t_τ^j. We illustrate the evaluation for a two-flow scenario in Chapter 3.

2.5 Summary

In this chapter, we characterize and model each component of our wireless scheduling problem, which comprises the wireless scheduling scenario, the wireless scheduler and the wireless receiver. The scheduling scenario includes the input-traffic and wireless channel characteristics, based on which we define various class-based scheduling scenarios. We also define the QoS performance metrics considered in this study, and expressed them in terms of statistics of the HOL packet delay. In the next two chapters, we describe our analysis framework for evaluating the HOL packet delay pdf of various wireless schedulers in terms of the input and channel parameters for various scheduling scenarios.

Chapter 3

Stochastic Analysis of Wireless Schedulers

In Chapter 2, we defined the components of the wireless scheduling problem and also defined various QoS metrics that can be expressed in terms of the statistics of the HOL packet delay. In this chapter, we describe our approach to derive p_{n^j} for various wireless schedulers in terms of the input-flow and channel parameters.

Since we assume a Markov-based wireless channel model, it is natural to model the wireless scheduling mechanism as a Markov process, and derive $\underline{p}_n = [p_{n^1}\ p_{n^2}\ \cdots\ p_{n^K}]$ in terms of the channel parameters $(p_c(0),\underline{g})$ for a given $\underline{r}$. Our approach is similar to the work in [4]; while the authors in [4] studied the delay performance of a simple FIFO scheduler with ARQ error-control strategy for communications over a bursty channel of a *single* flow, we consider a more complex problem of a channel-dependent scheduler in a multiple-flow scenario.

In this chapter, we define variants of wireless schedulers based on the generic model and present our approach for the Markov modelling and analysis of these schedulers [35] for the case where $\underline{\kappa}$=[1,$\cdots$,1].

3.1 Definition of Wireless Schedulers

The generic wireless scheduler model defined in Chapter 2 can be abstracted in terms of the mechanism of the SAP, CSM, FM and the AS. In this section, we specify the mechanisms of the SAP, CSM and the FM and define variants of wireless schedulers that differ in terms of the AS.

3.1.1 SAP

We consider a WRR mechanism for the SAP, where each flow j is allocated a contiguous block of r^j slots exactly once within any cycle or loop of R slots. In this study, we consider a simple WRR mechanism where flows are allocated in ascending order of their indices. The resulting allocation sequence, $\underline{a}^{WRR}$, is shown in Fig. 3.1. It is easy to show that $\underline{a}^{WRR}$ satisfies Property 2.1, and hence, WRR $\in \mathbf{F}^{\underline{r}}$.

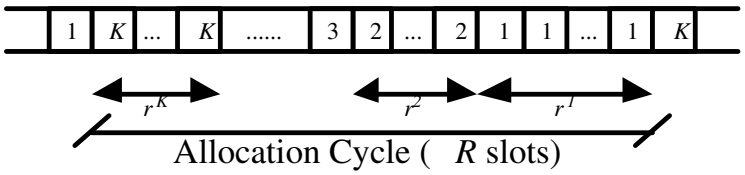

Figure 3.1: Allocation sequence for WRR slot allocation. Flows are allocated in ascending order of their indices, where each flow j is allocated a block of r^j slots.

3.1.2 CSM

We consider two variants for the channel prediction mechanism: (a) *deterministic* OSP, where $p_{\hat{0}} = p_{\hat{1}} = 1.0$ and (b) PCK.

3.1.3 AS

We define three variants of wireless schedulers according to the AS as follows:

Wireless-Fair Scheduler (WFS) : We define a Wireless-Fair Scheduler (WFS) that achieves optimal channel efficiency while maintaining fairness provision [16, 17, 18, 20, 22]. The AS achieves this by assigning a higher priority to a more lagging eligible flow (where ties are broken arbitrarily) so that its service deficit can be compensated as soon as possible; however, if all eligible flows are in-sync, then the highest priority is assigned to flow a_i^{SAP} if it is eligible; otherwise, the arbitration function, $Arb()$, determines the alternative flow to be selected for transmission.

If we define the set $\mathbf{L}_i \subseteq \mathbf{G}_i$ as follows:

$$\mathbf{L}_i = \{\arg\min_{g \in \mathbf{G}_i} l_{i-1}^g\}$$

then the flow f_i with the highest transmission priority is given as follows:

$$f_i \;=\; \begin{cases} a_i^{SAP}, & a_i^{SAP} \in \mathbf{L}_i \text{ and } l_{i-1}^j = 0 \forall j \in \mathbf{L}_i \\ Arb(\mathbf{L}_i), & \text{otherwise.} \end{cases}$$

where

$$\text{Prob}(Arb(\mathbf{L}_i) = j) \;=\; \begin{cases} \frac{1}{|\mathbf{L}_i|}, & j \in \mathbf{L}_i; \\ 0, & \text{otherwise.} \end{cases}$$

Channel-State Dependent Scheduler (CSD) : We define a Channel-State Dependent (CSD) scheduler by disabling the FM in the generic wireless scheduler model. Hence, the goal of this scheduler is to maximize channel efficiency while disregarding fairness provision. Disabling the FM is equivalent to setting l_i^j=0 $\forall$ i,j. As a result, $\mathbf{L}_i = \mathbf{G}_i$, and f_i can be determined as follows:

$$f_i \;=\; \begin{cases} a_i^{SAP}, & a_i^{SAP} \in \mathbf{G}_i \\ Arb(\mathbf{G}_i), & \text{otherwise.} \end{cases}$$

where

$$\text{Prob}(Arb(\mathbf{G}_i) = j) \;=\; \begin{cases} \frac{1}{|\mathbf{G}_i|}, & j \in \mathbf{G}_i; \\ 0, & \text{otherwise.} \end{cases}$$

Channel-State Independent WRR Scheduler (WRR) : We define a further abstraction by disabling both the FM and the CSM in the generic wireless scheduler model. As a result, the AS simply selects flow a_i^{SAP} for transmission in slot i regardless of its channel state.

Fair-Aggregation (FA) Scheduler : We define a variant of the Channel-State Independent Scheduler using the concept of flow aggregation [36], where multiple flows are combined into an aggregate flow prior to scheduling at the wireless link. Example applications of flow aggregation include ATM networks, the Class-Based QoS Framework proposed in [29] and the DiffServ QoS model [37]. Flows may be aggregated according to various criteria, e.g., common QoS requirements and/or source-destination specifications.

We define our Fair-Aggregation (FA) scheduler to comprise a fair aggregator followed by a packet dispatcher. The fair-aggregator schedules packets from each input-flow according to $\underline{r}$ such that they arrive in a single queue at the packet dispatcher in the order as depicted in Fig. 3.2. Notice that the packet transmission cycle corresponds to the slot allocation cycle in the WRR scheduler.

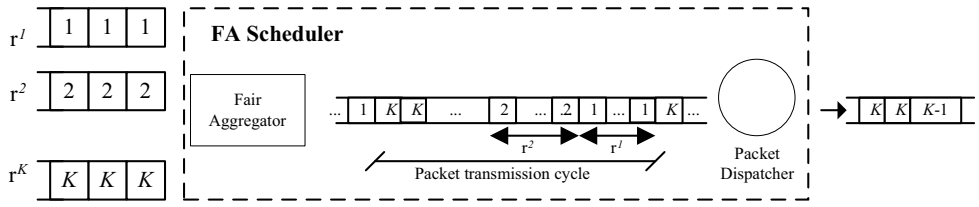

Figure 3.2: Components of a Fair-Aggregation scheduler. The Fair-Aggregator dispatches packets from each input-flow such that they arrive in a single queue according to a WRR allocation sequence at the packet dispatcher.

3.2 Markov Modelling of the Wireless-Fair Scheduler

In this section, we develop a Markov model for the Wireless-Fair Scheduler defined in Section 3.1. In order to illustrate the modelling approach, we consider the simple case of K=2 and $\underline{r}$= [1,R-1], where R>1, $R\in \mathbf{Z}^+$.

3.2.1 SAP

For K-flow scheduling, given $\underline{r}$, in order to implement the WRR allocation policy depicted in Fig. 3.1, we define the co*u*nter variable, u, which is incremented at the end of each slot, and is reset to zero at the beginning of each allocation cycle (i.e., $u \equiv u$ (modulo R)). Given u_{i-1}, a_i^{SAP} can be computed as follows:

$$a_i^{SAP} = \{j : \sum_{k=1}^{j-1} r^k \leq u_{i-1} \leq \sum_{k=1}^{j} r^k - 1,\ 1 \leq j \leq K\} \tag{3.1}$$

For our simple case, Eq. (3.1) becomes:

$$a_i^{SAP} = \begin{cases} 1, & u_{i-1} = 0; \\ 2, & 1 \leq u_{i-1} \leq R-1. \end{cases} \tag{3.2}$$

3.2.2 FM

Recall that $l_i^{j,k}$ denotes the lead of flow j over flow k at the end of slot i. Since $l_i^{j,k}$ and $l_i^{k,j}$ are updated concurrently whenever flow swapping takes place between flows j and k, for K=2, this implies that it suffices to define a single variable, l_i, to denote the lead of flow 1 relative to flow 2 (or the lag of flow 2 relative to flow 1) at the end of slot i.

The value of l_i is updated according to the transmission event that occurs in slot i. Let us denote any event associated with flow j that can occur in any

slot i by $\mathcal{E}^j_{a_i^{SAP}}$, where $1 \leq j$, $a_i^{SAP} \leq 2$. Then, we have the following:

$$\mathcal{E}^j_{a_i^{SAP}} \in \left\{ \begin{array}{l} \text{Flow } j \text{ } f\text{ails or de}f\text{ers its transmission in slot } i \text{ } (\mathcal{F}^j_{a_i^{SAP}}), \\ \text{Flow } j \text{ transmits } s\text{uccessfully in slot } i \text{ } (\mathcal{S}^j_{a_i^{SAP}}) \end{array} \right\} \quad (3.3)$$

When no transmission takes place in any slot (denoted by $\mathcal{F}$), neither flow achieves any lead over the other flow. However, when the event $\mathcal{S}_2^1$ takes place, flow 1 gains in lead over flow 2; on the other hand, when the event $\mathcal{S}_1^2$ takes place, flow 1 suffers a lag relative to flow 2. When flows transmit successfully in their allocated slots (i.e., event $\mathcal{S}_1^1$ or $\mathcal{S}_2^2$ occurs), if they were in-sync in slot i-1, neither flow gains in lead with respect to each other. Otherwise, the flow that is allocated slot i was lagging in slot i-1 (since lagging flows always receive priority in transmission). Its lag will be reduced after transmitting successfully in slot i.

Since the variable u tracks the slot allocation policy based on $\underline{r}$, it is not updated when the scheduler is in 'compensation' phase (i.e., when flows are not in-sync or $l_i \neq 0$). Hence, we can generalize the procedure to update u as follows:

$$u_i = \left\{ \begin{array}{ll} u_{i-1} + 1, & l_i = 0; \\ u_{i-1}, & \text{otherwise.} \end{array} \right. \quad (3.4)$$

Note that under error-free conditions, l_i=0 always and hence, u is incremented at the end of every slot.

According to [23], each update in lag or lead either increments or reduces the value of l by *one* slot. Based on the above, the effects of each transmission event in slot i on l_i can be depicted as follows:

$$l_i = \left\{ \begin{array}{ll} l_{i-1}, & \mathcal{F} \cup [(\mathcal{S}_1^1 \cup \mathcal{S}_2^2) \cap l_{i-1} = 0]; \\ l_{i-1} + 1, & \mathcal{S}_2^1 \cup [\mathcal{S}_1^1 \cap l_{i-1} < 0]; \\ l_{i-1} - 1, & \mathcal{S}_1^2 \cup [\mathcal{S}_2^2 \cap l_{i-1} > 0]. \end{array} \right. \quad (3.5)$$

However, according to the algorithms in [23], if some packetized fluid-fair queuing mechanism is used as the SAP, the scheduler can keep track of the cumulative service received by each flow based on the finishing time of its

HOL packet. Accordingly, when flows become in-sync after an error-burst, they will be allocated service with priority, thus maintaining fairness.

In our case, the simple WRR mechanism does not keep track of the cumulative service already received by each flow. Hence, when the flows become in-sync after an error burst, they will be scheduled as though they have received their fair allocation, which is clearly not the case. In order to restore the fairness property, we propose a weighted lead/lag updating scheme as follows:

$$l_i = \begin{cases} l_{i-1}, & \mathcal{F} \cup [(\mathcal{S}_1^1 \cup \mathcal{S}_2^2) \cap l_{i-1} = 0]; \\ l_{i-1} + R - 1, & \mathcal{S}_2^1 \cup [\mathcal{S}_1^1 \cap l_{i-1} < 0]; \\ l_{i-1} - 1, & \mathcal{S}_1^2 \cup [\mathcal{S}_2^2 \cap l_{i-1} > 0]. \end{cases} \tag{3.6}$$

In this way, whenever flow 1 'consumes' the slots of flow 2 or whenever it 'recovers' from its lag, its lead is incremented by R-1 so that an additional R-1 slots will be allocated to flow 2 in order to satisfy the constraint imposed by $\underline{r}$.

3.2.3 AS

Recall that we defined the transmission heuristic such that transmission priority is always given to the most lagging flow if it exists. We can incorporate this heuristic into Eq. (3.2) as follows:

$$e_i = \begin{cases} 1, & l_{i-1} < 0 \cup \{l_{i-1} = 0 \cap u_{i-1} = 0\}; \\ 2, & l_{i-1} > 0 \cup \{l_{i-1} = 0 \cap u_{i-1} > 0\}. \end{cases} \tag{3.7}$$

e_i can be interpreted as the output of an *e*mulated SAP which will be used to measure the *deviation* from a perfectly-fair scheduler due to non-error-free channel conditions. We note that under error-free conditions, $l_{i-1} = 0$ and hence, $e_i = a_i^{SAP}$.

As a result, the AS is given as follows:

$$f_i = \begin{cases} 1, & \hat{c}_i^1 = 0 \cap [e_i = 1 \cup (e_i = 2 \cap \hat{c}_i^2 = 1)]; \\ 2, & \hat{c}_i^2 = 0 \cap [e_i = 2 \cup (e_i = 1 \cap \hat{c}_i^1 = 1)]. \end{cases} \tag{3.8}$$

3.2.4 Markov Model of WFS

According to Eq. (3.7) and (3.8), the variables $(e_i, \hat{\ddot{c}}_i^2, u_{i-1}, l_{i-1})$ are required to implement the AS/DISP in slot i. Based on Eq. (3.5) and (3.6), given l_{i-1}, the value of l_i depends on the transmission event in slot i. The value of e_i determines the set of allowable events in slot i, i.e., when $e_i = 1$, the allowable events are $\{\mathcal{F}_1^1, \mathcal{F}_1^2, \mathcal{S}_1^1, \mathcal{S}_1^2\}$ whereas if $e_i = 2$, the allowable events are $\{\mathcal{F}_2^2, \mathcal{F}_2^1, \mathcal{S}_2^2, \mathcal{S}_2^1\}$. Based on Eq. (3.7), the value of e_i can be determined given l_{i-1} and u_{i-1}.

The probability of occurrence of each transmission event in slot i depends only on $\ddot{c}_i^2$ and $\hat{\ddot{c}}_i^2$, which in turn depends on the channel and prediction parameters. Hence, given these parameters, if $l_{i-1},\ u_{i-1}$ are known for all flow j, l_i and u_i can be determined. In other words, the two-flow WFS can be modelled as a two-dimensional finite-state Markov Chain with state variables given by $\{(l_i, u_i),\ i = 1, 2, 3 \cdots\}$, where the Markov points are defined at the end of each slot. The corresponding state-transition diagram is shown in Fig. 3.3.

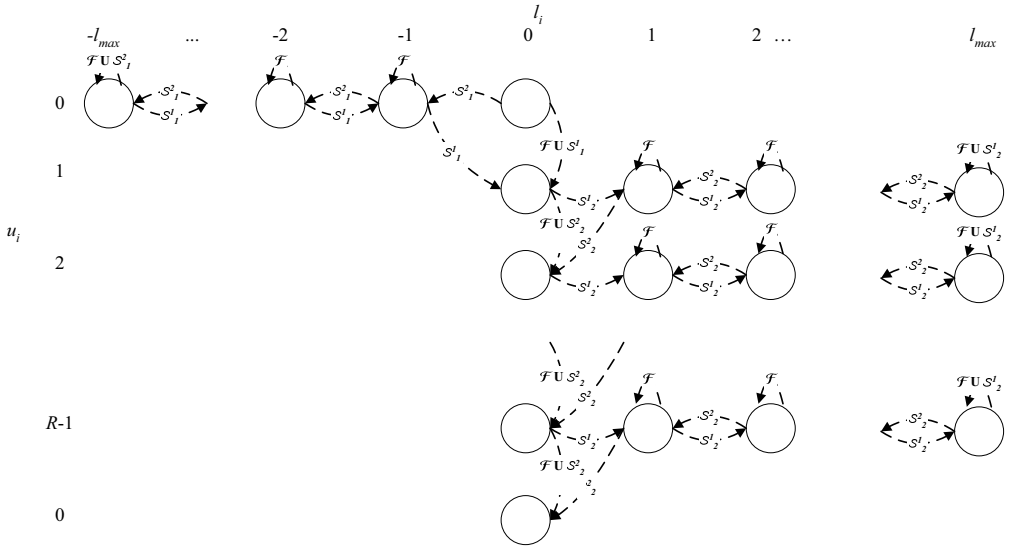

Figure 3.3: State-transition diagram for Markov model of two-flow WFS with $\underline{r} = [1, R\text{-}1]$.

According to the Ergodicity Theorem, a finite-state Markov chain is ergodic as long as it is *irreducible* and *aperiodic*. Since the channel quality, $p_c(0) > 0$, all state transitions in Fig. 3.3 can occur with non-zero probability. Hence, any state x can be reached by any other state y, and therefore, the Markov chain is irreducible. In addition, since states that 'self-loop' (i.e., aperiodic states) exist, the Markov chain is also aperiodic and hence, it is ergodic.

The same approach can be used to derive the Markov models for the other schedulers, and details can be found in [38]. A detailed analysis for the special case of R=2 can be found in [38].

3.3 Evaluation of QoS Parameters

In this section, we illustrate the evaluation of QoS parameters for flow j for the WFS based on its Markov model. We drop the superscript j for simplicity of notations and refer only to variables related to flow j unless otherwise stated.

Recall that the scheduling mechanism can be modelled as a two-dimensional Markov Chain given by $\{(l_i,u_i),\ i = 1,2,3 \cdots\}$. According to Section 2.2.5, the Stop-and-wait ARQ mechanism ensures that packets of each flow are transmitted in the order which they arrive at the queue, i.e., if packet m and m+1 transmit in slots x and y respectively, then $x < y$.

Hence, if we observe $(l_i,\ u_i)$ at the packet departure instances of flow j, as shown in Fig. 3.4, then $\{(l(m), u(m)),\ m = 1,2,3 \cdots\}$ forms a two-dimensional Markov Chain, where $x(m)$ denotes the value of x at the departure of packet m.

3.3.1 Evaluation of long-term performance metrics

Based on Section 2.4.1, the long-term performance metrics can be derived from the steady-state HOL packet delay pdf, p_n, which can be evaluated as

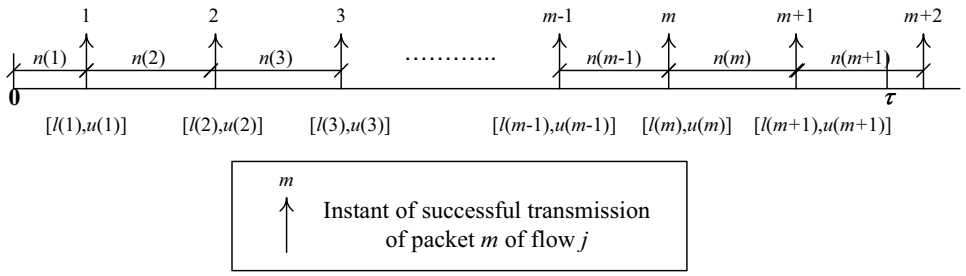

Figure 3.4: Evolution of the variables (n,l,u) at each packet departure of flow j.

follows:

$$\begin{aligned} p_n &= \lim_{m\to\infty} p_{n(m)} \\ &= \lim_{m\to\infty} \sum_{L(m),U(m)} p_{[n(m),l(m),u(m)]} \end{aligned} \tag{3.9}$$

Since $\{(l(m),u(m)),\ m=1,2,3...\}$ is a Markov Chain, $p_{[n(m),l(m),u(m)]},\ m \geq 1$ can be expressed recursively in terms of $p_{[n(m),l(m),u(m)\ |\ l(m-1),u(m-1)]},\ m \geq 1$, as follows:

$$\begin{aligned} p_{[n(m),l(m),u(m)]} &= \sum_{L(m-1),U(m-1)} p_{[n(m),l(m),u(m)\ |\ l(m-1),u(m-1)]} \cdot p_{[l(m-1),u(m-1)]} \\ p_{[l(m),u(m)]} &= \sum_{N(m)} p_{[n(m),l(m),u(m)]} \end{aligned} \tag{3.10}$$

The initial conditions for the above recursion are given as follows:

$$\begin{aligned} l(0) &= 0 \\ p_{u(0)} &= \frac{1}{R}, \quad 0 \leq U(0) \leq R-1 \end{aligned} \tag{3.11}$$

The limit in Eq. (3.9) exists since $\{(l(m),u(m)),\ m=1,2,3\cdots\}$ is ergodic (refer to Section 3.2.4).

3.3.2 Evaluation of short-term performance metrics

According to Section 2.4.2, we need to evaluate p_{t_τ} in order to obtain the short-term fairness metrics. We refer to Fig. 3.4, which depicts a sequence of flow j packet departures and their associated HOL packet delays.

Evaluation of p_{t_τ}

We use the notation, $\phi(m)$, for the cumulative HOL packet delay up to packet m, i.e., $\phi(m) = \sum_{x=1}^{m} n(x)$. $\phi(m)$ also represents the actual departure time of packet m if the first packet becomes HOL at the beginning of slot 1. In addition, $p_{x(m)}$ and $p_{x,y}(A,B)$ are used to denote Prob($x(m)$=X_m) and Prob(x=A,y=B) respectively. We proceed to compute p_{t_τ} as follows:

$$
\begin{aligned}
p_{t_\tau}(0) &= \text{Prob}(\phi(1) > \tau) = 1 - P_N(\tau) \\
p_{t_\tau}(1) &= \text{Prob}(\phi(1) \le \tau,\ \phi(2) > \tau) \\
&= \sum_{\Phi_1=1}^{\tau} \sum_{\Phi_2=\tau+1}^{\infty} p_{[\phi(1),\phi(2)]} \\
&= \sum_{\Phi_1=1}^{\tau} \sum_{\Phi_2=\tau+1}^{\infty} p_{[\phi(1),n(2)]}(\Phi_1, \Phi_2 - \Phi_1) \\
&= \sum_{U_1,L_1} \sum_{\Phi_1=1}^{\tau} \sum_{\Phi_2=\tau+1}^{\infty} p_{[\phi(1),n(2),l(1),u(1)]}(\Phi_1, \Phi_2 - \Phi_1, L_1, U_1) \\
&= \sum_{U_1,L_1} \sum_{\Phi_1=1}^{\tau} \sum_{\Phi_2=\tau+1}^{\infty} p_{[n(2)\ |\ l(1),u(1)]}(\Phi_2 - \Phi_1 \mid L_1, U_1) \cdot p_{[\phi(1),l(1),u(1)]} \\
p_{t_\tau}(2) &= \text{Prob}(\phi(2) \le \tau,\ \phi(3) > \tau) \\
&= \sum_{U_2,L_2} \sum_{\Phi_2=1}^{\tau} \sum_{\Phi_3=\tau+1}^{\infty} p_{[n(3)\ |\ l(2),u(2)]}(\Phi_3 - \Phi_2 \mid L_2, U_2) \cdot p_{[\phi(2),l(2),u(2)]}
\end{aligned}
$$

In general, $p_{t_\tau}(M)$ can be expressed in terms of p_n, $p_{[n(1,M+1)\ |\ l(M),u(M)]}$ and $p_{[n(1,M),l(M),u(M)]}$ as follows:

$$p_{t_\tau}(M) = \begin{cases} 1 - P_N(\tau), & M = 0; \\ \sum\limits_{U_M, L_M} \sum\limits_{\Phi_M = M}^{\tau} \sum\limits_{\Phi_{M+1}=\tau+1}^{\infty} & \\ p_{[n(M+1)\ |\ l(M),u(M)]}(\Phi_{M+1} - \Phi_M \mid L_M, U_M) \cdot & \\ p_{[\phi(M),l(M),u(M)]}, & 1 \le M \le \tau; \\ 0, & M > \tau. \end{cases} \tag{3.12}$$

To evaluate $p_{t_\tau}(M)$, we note the following:

- The expression, $p_{[\phi(m),l(m),u(m)]}$, can be recursively computed for $m \geq 1$ as follows:

$$\begin{aligned} p_{[\phi(m),l(m),u(m)]} &= \sum_{U_{m-1}, L_{m-1}} \sum_{\Phi_{m-1}=m-1}^{\Phi_m - 1} \\ &= p_{[\phi(m)\ |\ l(m-1),u(m-1)]}(\Phi_m - \Phi_{m-1} \mid L_{m-1}, U_{m-1}) \cdot \\ &\quad p_{[\phi(m-1),l(m-1),u(m-1)]} \end{aligned}$$

- $p_{[n(m+1)\ |\ l(m),u(m)]}$ can be expressed in terms of $p_{[n(m+1),l(m+1),u(m+1)\ |\ l(m),u(m)]}$ as follows:

$$p_{[n(m+1)\ |\ l(m),u(m)]} = \sum_{L_{m+1}, U_{m+1}} p_{[n(m+1),l(m+1),u(m+1)\ |\ l(m),u(m)]} \tag{3.13}$$

- $p_{[\phi(1),l(1),u(1)]}$ can be expressed in terms of $p_{[n(1),l(1),u(1)\ |\ l(0),u(0)]}$ as follows:

$$\begin{aligned} p_{[\phi(1),l(1),u(1)]} &= p_{[n(1),l(1),u(1)]} \\ &= \sum_{L_0, U_0} p_{[n(1),l(1),u(1)\ |\ l(0),u(0)]} \cdot p_{[l(0),u(0)]} \end{aligned} \tag{3.14}$$

  where $p_{[l(0),u(0)]}$ can be obtained from Eq. (3.11).

According to Eq. (3.12)-(3.14), as long as $p_{[n(m),l(m),u(m)\ |\ l(m-1),u(m-1)]},\ m \geq 1$ is known, we can compute p_{t_τ}, from which we can derive the short-term

performance metrics in terms of both absolute and relative fairness as a function of τ as follows:

$$\begin{aligned} E[\Delta_\tau] &= \frac{E[t_\tau]}{\tau} - \frac{1}{R} \\ Var[\Delta_\tau] &= \frac{Var[t_\tau]}{\tau^2} \\ E[FM_\tau^{j,k}] &= \frac{1}{\tau}\left|\frac{E[t_\tau^j]}{\frac{r^j}{R}} - \frac{E[t_\tau^k]}{\frac{r^k}{R}}\right| \end{aligned}$$

3.3.3 Evaluation of $p_{[n(m),l(m),u(m) \mid l(m-1),u(m-1)]},\ m \geq 1$

According to Section 3.3.1 and Section 3.3.2, the long-term and short-term performance metrics can be expressed in terms of $p_{[n(m),l(m),u(m) \mid l(m-1),u(m-1)]}$, $m \geq 1$. We illustrate the evaluation of $p_{[n(m),l(m),u(m) \mid l(m-1),u(m-1)]}$ in terms of the channel parameters for the case of m=1. According to Eq. (3.11), $l(0)$=0 and $u(0)$ is uniformly distributed in the interval [0,R-1].

Let us assume that $u(0)$=0. Since flow j can only transmit once over the interval [0,$n(1)$], according to Eq. (3.6), $l(1) \leq R-1$. For each possible $(l(1), u(1))$, our objective is to consider all combinations of permissible transmission events over the duration of flow j packet such that it departs only in slot $n(1)$. We define the following notations for transmission events that occur over an interval of slots beginning with slot i :

$\langle \mathcal{AB} \rangle_{N,x}$ **:** Over an interval of N slots, in slots i, i+1,$\cdots$, i+x-1, $\mathcal{A}$ occurs and in the remaining slots, $\mathcal{B}$ occurs

$\mathcal{E}^x$ **:** Event $\mathcal{E}$ occurs over slots i,i+1,i+2,$\cdots$ i+x-1

$(\mathcal{AB})_{N,x}$ **:** Over an interval of N slots, $\mathcal{A}$ occurs in x slots while $\mathcal{B}$ occurs in the remaining slots

Since $l(1) < 0$ is possible only if the event $\mathcal{S}_1^2$ occurs at least once, we consider two cases:

(a) Event $\mathcal{S}_1^2$ never occurs In this case, $0 \leq l(1) \leq R$-1. If the event $\mathcal{S}_2^1$ occurs, we have $l(1) = R$-1; otherwise, $l(1)$=0. Hence, we obtain the following expressions*:

$$\begin{aligned} p_{[n(1),l(1),u(1) \mid u(0)]} &= p_{([\mathcal{F}\overline{\mathcal{S}}_2^1]_{R,1}^{\frac{N_1-1}{R}}\mathcal{S}_1^1)}, L_1 = 0, \\ & \quad N_1 - 1 \equiv 0 \text{ modulo } R,\ U_1 = 1,\ U_0 = 0 \\ p_{[n(1),l(1),u(1) \mid u(0)]} &= p_{([\mathcal{F}\overline{\mathcal{S}}_2^1]_{R,1}^{\frac{N_1-k}{R}}[\mathcal{F}\overline{\mathcal{S}}_2^1]_{k-1,1}\mathcal{S}_2^1)}, \\ & \quad N_1 - k \equiv 0 \text{ modulo } R,\ L_1 = R-1, \\ & \quad U_1 = k-1,\ U_0 = 0,\ 2 \leq k \leq R \end{aligned} \tag{3.15}$$

(b) Event $\mathcal{S}_1^2$ occurs at least once In this case, $l(1) < R$-1, and we have the following expressions:

$$\begin{aligned} p_{[n(1),l(1),u(1) \mid u(0)]} &= \sum_{m=1}^{N_1+L_1-R+1} p_{([\mathcal{F}\overline{\mathcal{S}}_2^1]_{R,1}^{\frac{m-1}{R}}\mathcal{S}_1^2(\mathcal{S}_1^2\mathcal{F})_{N_1-m-1,R-L_1-2}\mathcal{S}_1^1)}, \\ & \quad L_1 \neq 0,\ U_1 = 0,\ U_0 = 0 \\ p_{[n(1),l(1),u(1) \mid u(0)]} &= \sum_{m=1}^{N_1-R+1} p_{([\mathcal{F}\overline{\mathcal{S}}_2^1]_{R,1}^{\frac{m-1}{R}}\mathcal{S}_1^2(\mathcal{S}_1^2\mathcal{F})_{N_1-m-1,R-2}\mathcal{S}_1^1)}, \\ & \quad L_1 = 0,\ U_1 = 1,\ U_0 = 0 \end{aligned} \tag{3.16}$$

The corresponding expressions for other values of $u(0)$ can be obtained in a similar manner.

Hence, $p_{[n(m),l(m),u(m) \mid l(m-1),u(m-1)]}$, $m \geq 1$, can be expressed in terms of $p_{\mathcal{E}}$. We illustrate the evaluation of $p_{\mathcal{E}}$ for two cases: (a) Perfect Channel knowledge and (b) Deterministic One-step Prediction.

*Note that the notation $\overline{\mathcal{S}}_2^1$ refers to the *complement* of the event $\mathcal{S}_2^1$, i.e., $\mathcal{S}_2^2 \cup \mathcal{F}$. In addition, the notation $p_{\mathcal{E}}$ refers to the probability of occurrence of event $\mathcal{E}$.

Evaluation of $p_{\mathcal{E}}$ for Perfect Channel Knowledge

The probability of occurrence of each event defined in Eq. (3.3) can be expressed in terms of the product of a constant matrix and $\underline{\underline{p}}_{\ddot{c}^2}$ (Refer to Eq. (2.1)). For example, consider the event $\mathcal{S}_1^2$. With perfect channel knowledge, for this event to occur in slot i, $c_i^1 = 1$ and $c_i^2 = 0$, and this corresponds to the entries in the third column of $\underline{\underline{p}}_{\ddot{c}^2}$. Hence, we can express $p_{\mathcal{S}_1^2}$ as a 4×4 matrix as follows:

$$p_{\mathcal{S}_1^2} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \underline{\underline{p}}_{\ddot{c}^2}$$

By defining the corresponding matrices for all possible events, the expressions in Eq. (3.15) and (3.16) can be evaluated.

Evaluation of $p_{\mathcal{E}}$ for Deterministic One-step Prediction

Channel prediction is never perfect in reality and hence, the scheduler does not possess perfect knowledge of the channel of each flow at each scheduling instant. Hence, we analyze the effects of deterministic one-step channel prediction (as defined in Section 3.1.2) on the QoS performance of the WFS and CSD schedulers. We denote these schedulers as WFS-OSP and CSD-OSP respectively, as opposed to WFS-PCK and CSD-PCK, where perfect channel knowledge is assumed.

Again, let us consider the event $\mathcal{S}_1^2$. Since slot i is allocated to flow 1, it has higher priority to transmit. Hence, for flow 2 to be eligible for transmission in slot i, we must have $c_{i-1}^1 = 1$ and $c_{i-1}^2 = 0$. In addition, for flow 2 to transmit successfully, $c_i^2 = 0$. Hence, the relevant entries in $\underline{\underline{p}}_{\ddot{c}^2}$ are $p_{c^1}(0|1) \cdot p_{c^2}(0|0)$ and $p_{c^1}(1|1) \cdot p_{c^2}(0|0)$, and therefore, we can express $p_{\mathcal{S}_1^2}$ as a 4×4 matrix as

follows:

$$p_{S_1^2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \underline{\underline{p}}_{\vec{c}^2}$$

3.4 Numerical Results (K=2, homogeneous scenario)

In Section 3.2, we defined Markov models for various two-flow wireless schedulers, which can be classified as channel-dependent (WFS and CSD) and channel-independent (WRR and FA). Using these models, we derived the corresponding QoS metrics in terms of the channel parameters, ($p_c(0)$,$\underline{g}$).

In this section, we study the effects of the channel agility on the QoS performance of each wireless scheduler. In order to isolate the effects of input- and channel-heterogeneity, we consider a *homogeneous* scenario, where r^j=1 and $g^j = g$ for $1 \leq j \leq 2$. As a result, the performance achieved by all flows are identical.

We present results for $p_c(0)$=0.99 and $g = \{1.0, 0.5\}$ that correspond to uncorrelated and persistent channels respectively. The corresponding results for oscillatory channels (g=2.0) will be presented in a later section.

3.4.1 Long-Term Performance

Throughput

The throughput achievable by each scheduler reflects how efficiently the channel is utilized while the time-fraction requirements are satisfied. We plot the overall throughput, T, as a function of g in Fig. 3.5.

With perfect channel knowledge, the throughput achieved by the channel-

dependent schedulers is higher than that of the channel-independent schedulers due to flow swapping. The WFS achieves the same throughput as the CSD scheduler since the priority of the transmission heuristics is to maximize channel efficiency. The FA scheduler achieves the same throughput as the WRR scheduler when the channel is uncorrelated (i.e., when g=1.0); however, as the channel becomes persistent (i.e., as g is reduced), the throughput of the former is degraded due to HOL blocking.

One-step channel prediction results in throughput degradation and the effect is more significant when the channel is uncorrelated since the prediction accuracy is reduced. Under such channel conditions, no throughput gain is achieved with channel-dependent scheduling. However, as the channel becomes persistent, channel-dependent schedulers achieve significant gain in throughput over channel-independent schedulers.

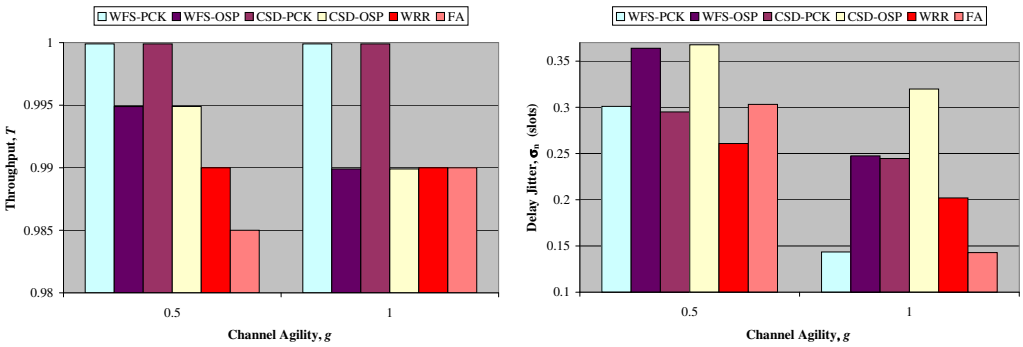

Figure 3.5: Throughput (left) and jitter (right) results for various two-flow wireless schedulers in an uncorrelated (g=1.0) and persistent channel (g=0.5) with $p_c(0)$=0.99.

Delay Jitter

The buffer requirement of the wireless receiver depends on both the delay jitter as well as the average delay (see Chapter 4). For a given average delay and acceptable packet loss rate due to buffer overflow, a lower jitter will result in a lower buffer requirement. We plot the delay jitter, σ_n, as a function of g in Fig. 3.5.

With perfect channel knowledge, WRR introduces a larger delay variance compared to the WFS and FA scheduler when the channel is uncorrelated since a flow may 'miss' its transmission in each allocation cycle due to channel errors. The CSD scheduler performs worst as expected since random flow swapping introduces large variations in the delay. However, as the channel becomes persistent, the performance of the FA scheduler is degraded due to HOL blocking.

One-step channel prediction results in increased delay variation in the channel-dependent schedulers. As a result, for the channel conditions considered, channel-independent schedulers achieve lower delay variation than their channel-dependent counterparts.

Delay violation probability

We plot the delay violation probability, 1-$P_{N_{max}}$, as a function of the delay bound, N_{max}, in Fig. 3.6. For a given delay bound, a lower violation probability is desirable since this results in a lower packet dropping rate.

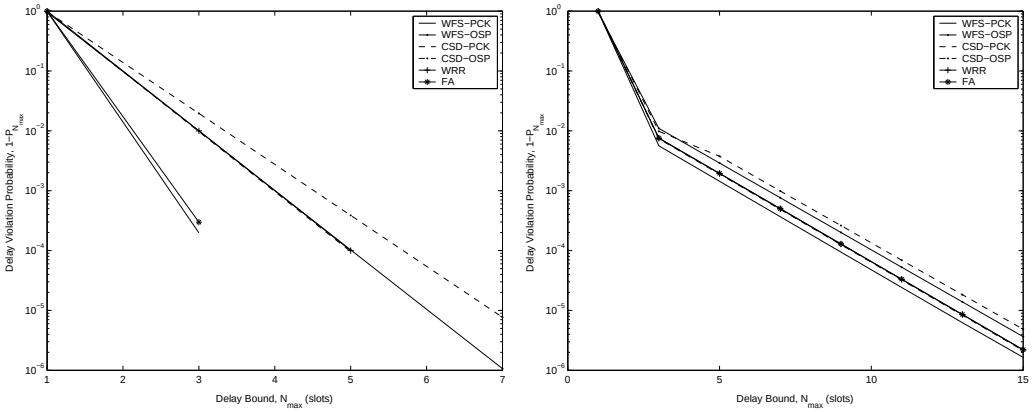

Figure 3.6: Delay violation probability for various two-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

When the channel is uncorrelated, we notice a significant degradation in delay violation probability in the channel-dependent schedulers as a result of channel prediction, with a more severe degradation in the WFS. This is

because the prediction error is the highest when the channel is uncorrelated, and therefore, slots may be 'wasted' due to failed transmissions (an erroneous channel is predicted to be error-free) or unattempted transmissions (an error-free channel is predicted to be erroneous). Under such circumstances, the channel-independent schedulers offer superior delay performance, with the FA scheduler offering the best performance.

As the error behavior becomes more persistent, prediction accuracy is improved and therefore, the level of degradation as a result of one-step prediction is reduced. However, the channel-independent schedulers still upper-bound the performance of channel-dependent schedulers. At even higher levels of channel persistence, we expect the performance of the WRR scheduler to improve relative to the FA scheduler due to HOL blocking. In addition, the performance of channel-dependent schedulers is expected to approach that of the WRR scheduler.

3.4.2 Short-Term Performance

We plot $Var[\Delta_\tau]$ as a function of τ in Fig. 3.7, where a smaller value of $Var[\Delta_\tau]$ at a given τ is desirable since this implies a lower throughput fluctuation from the time-fraction requirement.

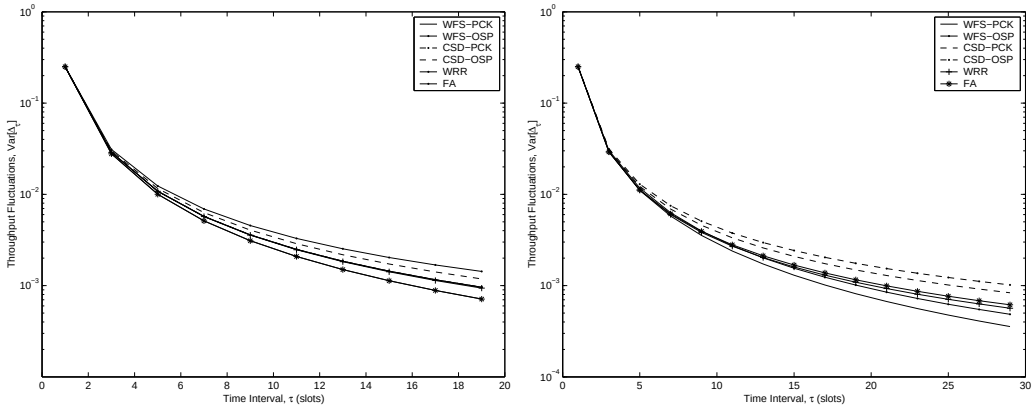

Figure 3.7: Throughput fluctuation as a function of τ for various two-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

The CSD scheduler performs worst in terms of short-term fairness as expected since flow swapping gives rise to 'unfairness' and there is no mechanism to restore fairness. The WFS improves the fairness performance of CSD scheduler via the compensation mechanism of the FM. The channel-independent schedulers maintain short-term fairness via the SAP.

With perfect channel prediction, WRR performs worse than the WFS and FA scheduler when the channel is uncorrelated since a flow may 'miss' its transmission in each allocation cycle due to channel errors. However, as the channel becomes persistent, the performance of the FA scheduler is degraded due to HOL blocking. Imperfect channel prediction results in degraded fairness performance in the channel-dependent schedulers, particularly when the channel is uncorrelated.

We would like to point out that under a homogeneous environment, the CSD scheduler is long-term fair since the flow swapping is symmetrical amongst all flows and hence, eventually, all flows will receive their fair share of the resources due to the perfectly-fair SAP.

3.5 Discussion

3.5.1 Scalability Issues in Markov Modelling of WFS

In Section 3.2, we described our approach for the modelling and analysis of wireless schedulers for the case of K=2. In a general K-flow scheduling scenario, for the WFS, a vector $\underline{l}$ of length $\frac{K\times(K-1)}{2}$ is needed to characterize the FM, in addition to the variable u. For example, for K=3, we need to define the variables $l_i^{1,2}$, $l_i^{1,3}$ and $l_i^{2,3}$. Hence, a $\frac{K\times(K-1)}{2}$+1-dimensional Markov Chain is required to model a K-flow WFS, and such a model is analytically untractable using our approach. However, we make the following observations:

- Intuitively, we expect that the short-term fairness performance of the

CSD scheduler is *worst* for K=2 and is expected to improve as K increases. Consequently, we expect the gain in fairness achieved by the WFS over the CSD scheduler to be reduced as K increases. Hence, the short-term fairness performance of the WFS scheduler can be upper-bounded by the CSD scheduler for K>2.

- In the current study, the CSD scheduler selects an arbitrary flow to transmit amongst the eligible ones in the event that an allocated flow perceives an erroneous channel. The short-term fairness performance of the CSD scheduler can be improved in this aspect by a prioritized selection scheme instead of a random one. This can improve the tightness of the fairness bound, and will be elaborated in Chapter 4.

Hence, we drop the WFS from further consideration, and attempt to incorporate fairness provisioning within the Arbitration Scheme of the CSD scheduler in subsequent chapters.

3.5.2 QoS Performance Evaluation (K>2, homogeneous scenario)

We show in [38, 39] that K-flow wireless schedulers (excluding the WFS) can be modelled by the one-dimensional Markov Chain, $\{u_i,\ i=1,2,3\cdots\}$. Based on these models, the QoS metrics can be evaluated in terms of products of $2^K \times 2^K$ matrices, where the numerical complexity of these matrix products determines the complexity of the analysis.

We consider a homogeneous K-flow wireless scheduling scenario, where each flow carries delay-sensitive traffic. An important example is radio broadcasting via the Internet to a group of K wireless receivers. Hence, all flows have equal demands in terms of time-fraction requirement. In addition, each receiver is limited in terms of receiver buffer capacity, and hence, the traffic to each user is sensitive to delay as well as delay jitter.

Results for long-term performance are shown in Fig. 3.8 to 3.9 for K=3

and Fig. 3.11 to 3.12 for K=4 respectively. We observe similar performance trends amongst the various algorithms in terms of delay violation probability, throughput as well as delay jitter as in the case of K=2. However, as K increases, we note the following:

- for an uncorrelated channel, the delay jitter is increased significantly for the WRR and CSD scheduler, but the increase is less significant for the FA scheduler;

- for a persistent channel, the throughput achieved by the FA scheduler is reduced while that of the WRR and CSD schedulers are invariant with K; in addition, the delay jitter is increased for all algorithms.

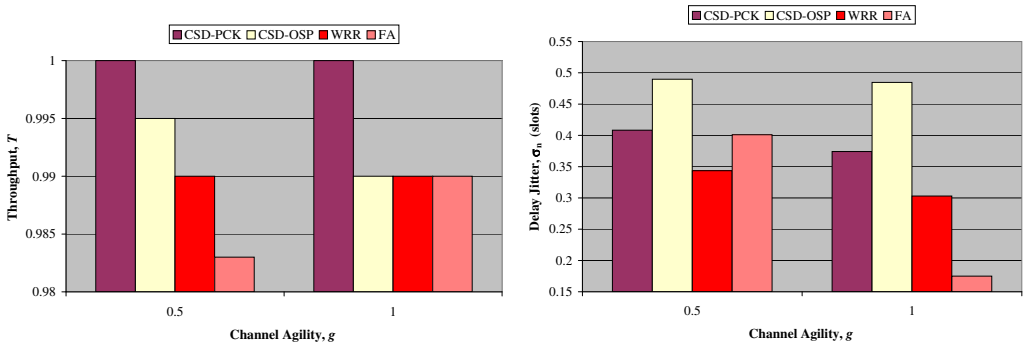

Figure 3.8: Throughput (left) and jitter (right) results for various three-flow wireless schedulers in an uncorrelated (g=1.0) and persistent channel (g=0.5) with $p_c(0)$=0.99.

Results for short-term performance are shown in Fig. 3.10 and 3.13. We observe the same performance trends as in the case of K=2. However, as K increases, the performance gap amongst all the schedulers is reduced. In addition, we verify that the fairness performance of the CSD scheduler is improved as K increases, concurring with our expectation.

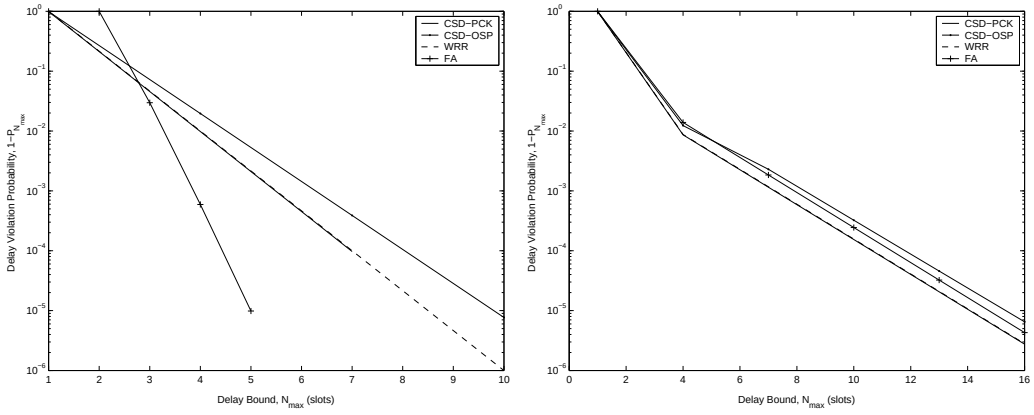

Figure 3.9: Delay violation probability for various three-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

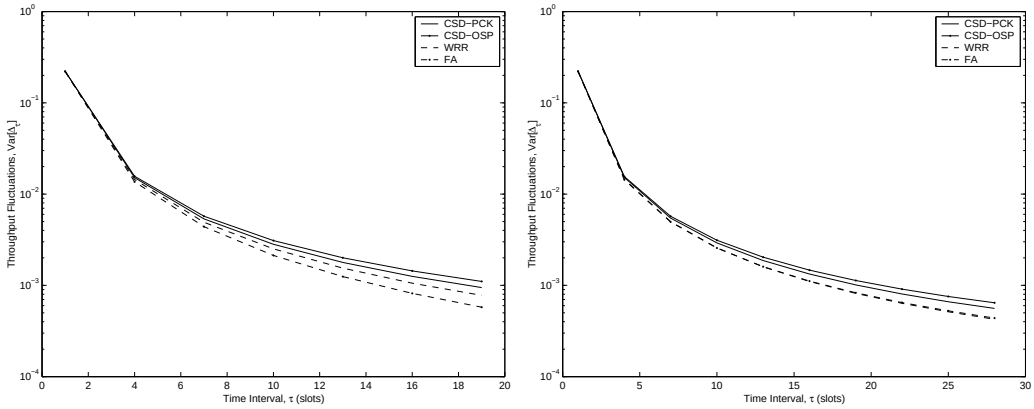

Figure 3.10: Throughput fluctuation as a function of τ for various three-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

3.5.3 Impact of ARQ mechanism on QoS Performance of Channel-dependent Schedulers

In this study, we assume a simple Stop-and-wait ARQ policy at the DISP, where all packets are transmitted in the same order as they arrive at the incoming queues. Although this eliminates the need for re-ordering packets as they arrive at the wireless receiver, a larger HOL packet delay may be incurred as a result of erroneous channel prediction.

Hence, this policy is suitable for applications that are sensitive to packet

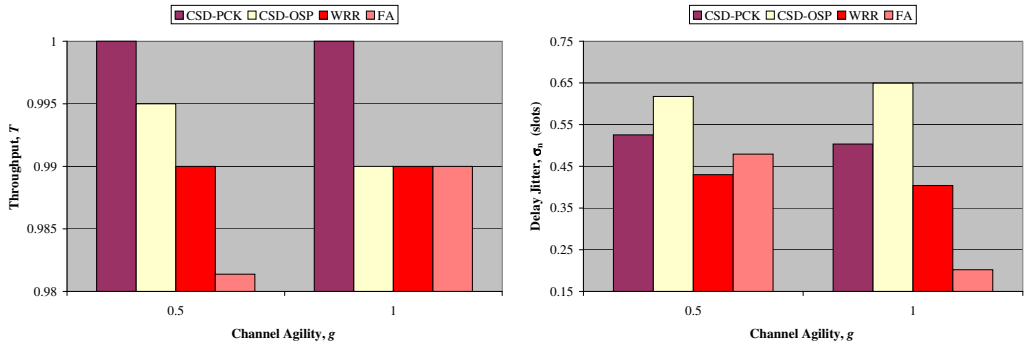

Figure 3.11: Throughput (left) and jitter (right) results for various four-flow wireless schedulers in an uncorrelated (g=1.0) and persistent channel (g=0.5) with $p_c(0)$=0.99.

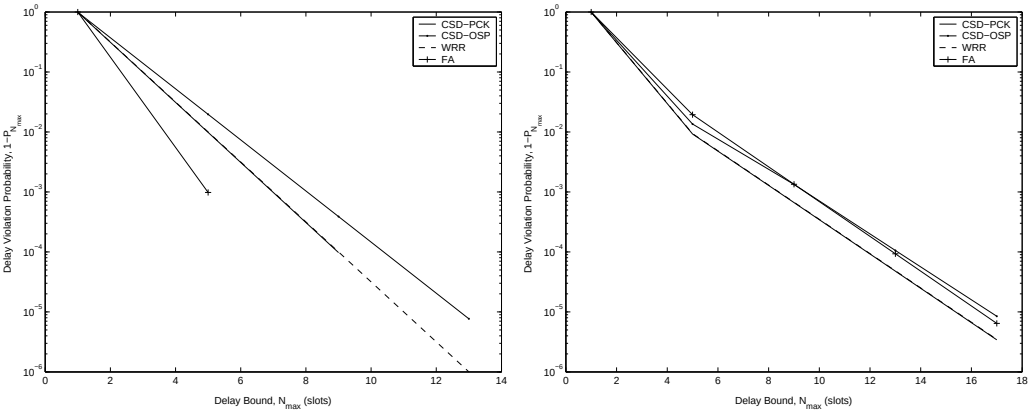

Figure 3.12: Delay violation probability for various four-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

loss rates but can tolerate larger delays, e.g., data transmissions. However, alternative ARQ policies such as Selective Repeat are necessary for real-time and interactive applications that are delay sensitive. In [40] and references therein, a hybrid selective repeat ARQ/Adaptive Modulation System (AMS) combined with a scheduler supplied by channel predictions is proposed for the downlink transmission in 4G systems, and simulation results were presented to demonstrate the importance of ARQ when channel prediction is imperfect. However, the analysis of the proposed hybrid ARQ scheme is difficult.

As part of the future work, we intend to analyze the trade-off between delay

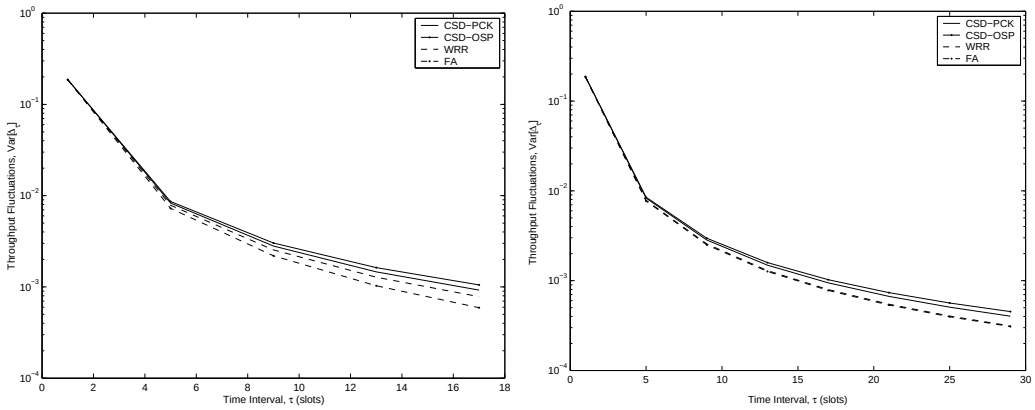

Figure 3.13: Throughput fluctuation as a function of τ for various four-flow wireless schedulers for $p_c(0)$=0.99, g= 1.0 (left) and g = 0.5 (right).

and packet-loss performance due to different ARQ policies as a result of imperfect channel prediction.

3.5.4 QoS Performance Evaluation for Oscillatory Channels ($g \approx 2.0$)

In a *perfectly-oscillatory* channel (g=2.0), the channel state of each flow toggles between 0 and 1 across successive slots. We make the following observations for such a channel:

- Deterministic one-step channel prediction cannot be employed in such channels since the prediction accuracy will be 0, and hence, successful transmissions will never occur;
- The WRR, CSD and WFS schedulers cannot be employed for homogeneous K-flow scheduling, where K is even, since there exist(s) initial channel state(s) that results in infinite starvation of some flows.

Hence, we will only compare the QoS performance of the CSD-PCK, WRR and FA scheduler for a *highly-oscillatory* channel, where $(p_c(0),g) = (0.5,1.95)$. We present the throughput and delay jitter performance for K=2, 3 and 4

in Fig. 3.14.

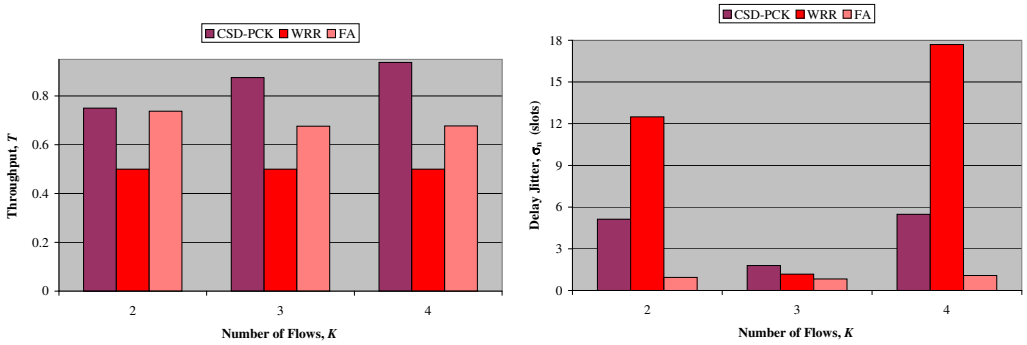

Figure 3.14: Throughput (left) and jitter (right) results for various K-flow wireless schedulers in an oscillatory channel.

The relative throughput performance amongst the schedulers is the same for the various values of K. The CSD-PCK scheduler achieves the highest throughput, and the throughput increases with K since the likelihood of finding at least one flow with an error-free channel is increased. The throughput achieved by the WRR scheduler is the lowest and is invariant with K. The throughput achieved by the FA scheduler falls between the levels achieved by the other schedulers, and is reduced as K increases due to increased likelihood of wasted slots because of HOL blocking.

The relative performance in terms of delay jitter is different for different values of K. Specifically, for $K = 2$ and 4 (or even values of K in general), the WRR and CSD-PCK schedulers achieve significantly higher jitter compared to the FA scheduler. For K=3 (or odd values of K in general), although the FA scheduler achieves the lowest jitter, it is not significantly lower than that achieved by the other schedulers.

We summarize the throughput and jitter performance of the various schedulers for different types of channels in Fig. 3.15 and Fig. 3.16 for K=2 and 3 respectively. The throughput achieved with the CSD-PCK scheduler is the maximum amongst all the schedulers. However, we make the following additional observations:

- When the channel is persistent, the CSD-PCK scheduler is optimal in terms of jitter as well;

- When the channel is uncorrelated, the jitter performance of all the schedulers are similar;

- When the channel is oscillatory, the jitter performance of all the schedulers are similar for odd values of K; however, for even values of K, the FA scheduler achieves a significantly lower jitter than the other schedulers.

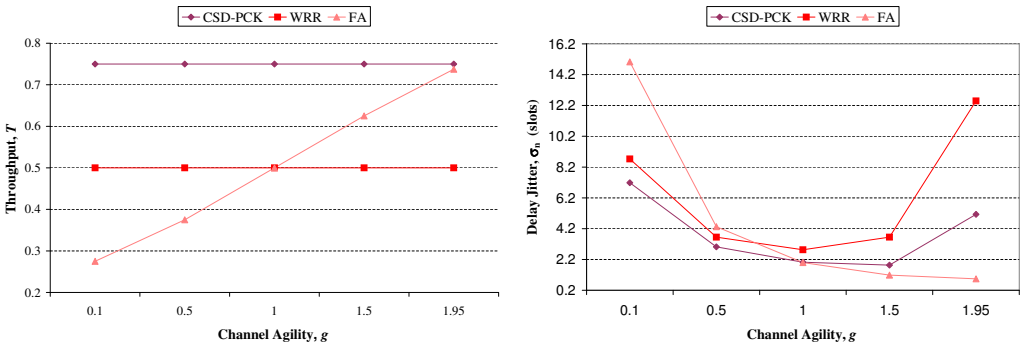

Figure 3.15: Throughput (left) and jitter (right) results for various two-flow wireless schedulers at various levels of channel agility.

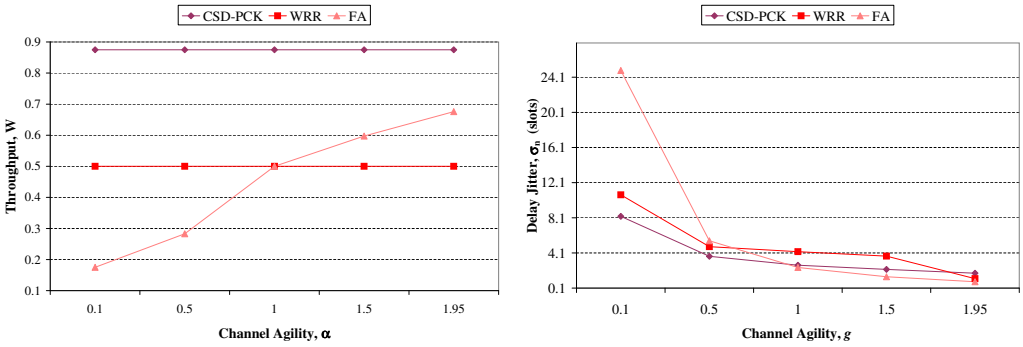

Figure 3.16: Throughput (left) and jitter (right) results for various three-flow wireless schedulers at various levels of channel agility.

3.6 Summary

In this chapter, we present our approach for the Markov modelling and analysis of variants of a generic two-flow wireless scheduler. Based on the Markov models, we evaluate various QoS performance metrics such as throughput, delay and fairness over different time scales as a function of the wireless channel parameters. We also highlight the scalability limitations of our approach for a multiple-flow Wireless-Fair Scheduler, and suggest incorporating the fairness provisioning within the Arbitration Scheme of the Channel-State Dependent scheduler (Chapter 4).

We present some numerical results to compare the QoS performance of each scheduler for a homogeneous scheduling scenario. For each type of channel, we observe a trade-off amongst different schedulers in terms of throughput and delay-jitter performance. In addition, the relative performance of the schedulers is also dependent on the type of channel.

Hence, the scheduling mechanism to be employed should be adaptive to the channel conditions as well as the priority of QoS metrics. For delay-sensitive flows, where maintaining low delay bound violation probability and delay jitter is of highest priority, we propose a hybrid scheduler that performs channel-independent scheduling when the channel is either uncorrelated or oscillatory, but switches to channel-dependent scheduling when the channel becomes persistent (Chapter 5).

Chapter 4

Framework for Performance Analysis of Channel-State Dependent Schedulers

In Chapter 3, we defined various wireless schedulers that assume a simple WRR mechanism for the SAP, a deterministic one-step predictor for channel prediction, and differ in terms of the AS. We presented our approach for Markov modelling and analysis of these schedulers for the scenario where $\underline{\kappa} = [1,\cdots,1]$. In particular, we verified the ergodicity and derived QoS metrics for a two-flow scheduling scenario. However, we also highlighted the scalability limitations of our approach for the WFS in a multiple-flow scenario as the number of states that the FM needs to maintain increases quadratically as K increases.

On the other hand, the CSD scheduler (WFS with FM disabled) does not suffer from this limitation, and the corresponding scheduler model remains ergodic in a multiple-flow (K=3,4) scenario. Hence, in this chapter, we present a framework for the performance analysis of a general K-flow CSD scheduler [41] based on the notion of *constrained* state-transition matrices. Using this framework, we can derive the HOL packet delay pdf more efficiently by skipping the tedious transient computations of Markov analysis. In addition, we propose novel schemes that embed fairness provision into the AS, thus substituting for the functions of the FM.

4.1 Description of Channel-State Dependent Scheduler

We recall that the CSD scheduler is an abstraction of the generic wireless scheduler model depicted in Fig. 2.2, where the input from the FM is disabled. We depict the architecture of the CSD scheduler in Fig. 4.1.

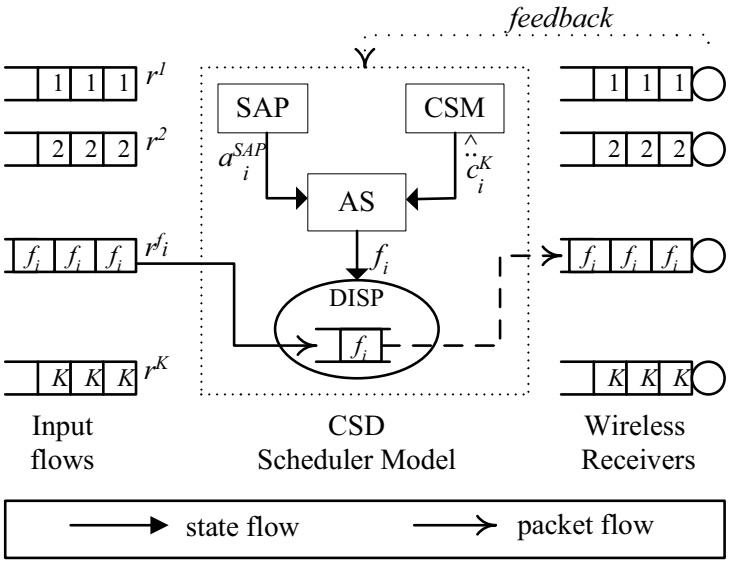

Figure 4.1: Channel-State Dependent scheduler model, with illustration of state flow, downlink packet flow (dashed) and uplink packet flow (dotted) in slot *i*.

According to Chapter 2, we consider SAP $\in \mathbf{F}^{\underline{r}}$ and the probabilistic one-step predictor for channel prediction. Recall that the AS determines the set of eligible flows, $\mathbf{G}_i$, assigns a priority to each eligible flow, and then selects the flow, f_i, with the highest priority to transmit in slot i. Since the AS tries to emulate the SAP, flow a_i^{SAP} has the highest priority to transmit in slot i if it is eligible; otherwise, we consider the following priority assignment schemes:

4.1.1 Uniform Arbitration

The simplest assignment scheme is Uniform Arbitration (UA), where all eligible flows are assigned equal priorities. This scheme is considered in Chapter

3, which is reproduced as follows:

$$f_i = \begin{cases} a_i^{SAP}, & a_i^{SAP} \in \mathbf{G}_i \\ Arb(\mathbf{G}_i), & \text{otherwise.} \end{cases} \quad (4.1)$$

where

$$\text{Prob}(Arb(\mathbf{G}_i) = j) = \begin{cases} \frac{1}{|\mathbf{G}_i|}, & j \in \mathbf{G}_i; \\ 0, & \text{otherwise.} \end{cases} \quad (4.2)$$

4.1.2 Prioritized Arbitration

We propose *prioritized* arbitration schemes for the following scenarios:

- **Input-homogeneous Scenario :** In this scenario, according to Property 2.1, $\mathbf{F}^{\underline{r}}$ reduces to a Round Robin (RR) scheduler. When flow a_i^{SAP} is not eligible for transmission in slot i, we can assign a *higher* priority to an eligible flow whose next allocation is *further* away from slot i. For example, $\underline{a}^{SAP} = \{\cdots,3,4,5,1,\boxed{2},3,4,5,\cdots\}$ for K=5. If a_i^{SAP}=2, then we assign higher priorities to flows 4 and 5 than flows 1 and 3 when flow 2 is not eligible to transmit in slot i. In this way, the transmissions of each flow are spread out as uniformly as possible, thus reducing the HOL packet delay variation (equivalent to short-term fairness provision) to the CSD scheduler.

  Quantitatively, let $v^j_{a_i^{SAP}}$ denote the priority assigned to flow j in slot i, where $1 \leq v^j_{a_i^{SAP}} \leq v_{max}$ and a smaller value denotes a higher priority. The priority assignment mechanism is defined as follows:

$$v^j_{a_i^{SAP}} = \begin{cases} j + 1 + v_{max} - a_i^{SAP}, & \langle -v_{max} + a_i^{SAP} \rangle_0^K \leq j \leq \langle a_i^{SAP} - 1 \rangle_0^K; \\ a_i^{SAP} + v_{max} + 1 - j, & \langle a_i^{SAP} + 1 \rangle_0^K \leq j \leq \langle a_i^{SAP} + v_{max} \rangle_0^K. \end{cases} \quad (4.3)$$

  where

$$\langle x \rangle_0^K = \begin{cases} x + K, & x < 0; \\ x - K, & x > K; \\ x, & \text{otherwise.} \end{cases}$$

and

$$v_{max} = \begin{cases} \frac{K-1}{2}, & K \text{ is odd;} \\ \frac{K}{2}, & K \text{ is even.} \end{cases}$$

We denote the priority assignment scheme by $PA(h)$, where h is a priority t$h$reshold such that an eligible flow with lower priority than h will not be permitted to transmit. The choice of h may represent a trade-off between channel efficiency and short-term fairness under good channel conditions: a larger h implies more flows are permitted to transmit, which may in turn result in poor short-term fairness. We use h=0 to denote the case where flow a_i^{SAP} is the only flow permitted to transmit in slot i. This scheme can be implemented by refining the definition of $\mathbf{G}_i$ in Eq. (4.2) as follows:

$$\mathbf{G}_i = \{j : v^j_{a_i^{SAP}} \leq h,\ v^j_{a_i^{SAP}} = \min_{1 \leq k \leq K, k \neq a_i^{SAP}} v^k_{a_i^{SAP}} \text{ and } \hat{c}_i^j = 0\}$$

- **Input-heterogeneous Scenario :** In this scenario, the throughput-fairness is compromised with UA since it disregards the difference in time-fraction requirement amongst flows (See Section 4.3.2). We propose a priority assignment scheme (denoted by PA) that assigns a weight $\frac{r^j}{r^{a_i^{SAP}}}$ to flow j, so that it is more likely to be selected for transmission than flow k if $r^j > r^k$. As a result, Eq. (4.2) becomes:

$$\text{Prob}(Arb(\mathbf{G}_i) = j) = \begin{cases} \frac{r^j}{r^{a_i^{SAP}}} \cdot \frac{r^1}{r^K} \cdot \frac{1}{|\mathbf{G}_i|}, & j \in \mathbf{G}_i; \\ 0, & \text{otherwise.} \end{cases} \quad (4.4)$$

We note that the normalization factor, $\frac{r^1}{r^K}$, ensures that the RHS of Eq. (4.4) < 1.

4.2 Performance Analysis of Channel-State Dependent Schedulers

In this section, we introduce the notion of *constrained* state-transition matrices, and use it to evaluate the steady-state HOL packet delay pdf, p_{n^j}, for

the CSD scheduler, CSD_{AS}^{SAP}, where AS $\in$ {UA,PA(h),PA} and SAP $\in \mathbf{F}^{\underline{r}}$. The corresponding analysis for perfect channel knowledge can be found in [42].

4.2.1 Notion of Constrained State-Transition Matrices

Let $\mathcal{S}_{a_i^{SAP}}^j$ ($\mathcal{F}_{a_i^{SAP}}^j$) denote a *S*uccessful (de*F*erred or *F*ailed) transmission of flow j in a slot i allocated to flow a_i^{SAP}. The probability of occurrence of $\mathcal{S}_{a_i^{SAP}}^j$ is determined by the AS, the SAP and the values of $(\ddot{c}_{i-1}^K, \ddot{c}_i^K)$. Conversely stated, given the SAP, AS, the occurrence of $\mathcal{S}_{a_i^{SAP}}^j$ imposes a constraint on $\underline{p}_{\ddot{c}_{i-1}^K}$ and $\underline{p}_{\ddot{c}_i^K}$. Hence, we define the *constrained state-transition matrix* for event $\mathcal{S}_{a_i^{SAP}}^j$ as follows:

$$\underline{p}_{\underline{\ddot{c}}^K}(\mathcal{S}_{a_i^{SAP}}^j) = \underline{\underline{D}}_{i-1}^K(\mathcal{S}_{a_i^{SAP}}^j) \times \underline{p}_{\underline{\ddot{c}}^K} \times \underline{\underline{D}}_i^K(\mathcal{S}_{a_i^{SAP}}^j) \tag{4.5}$$

where $\underline{\underline{D}}_x^K(\mathcal{S}_{a_i^{SAP}}^j)$ is a $2^K \times 2^K$ diagonal matrix such that the diagonal element of row y is the probability that $\mathcal{S}_{a_i^{SAP}}^j$ will occur if $\ddot{c}_x^K$=y-1, where $x \in \{i\text{-}1, i\}$. Since the events $\mathcal{S}_{a_i^{SAP}}^j$ and $\mathcal{F}_{a_i^{SAP}}^j$ are complementary with respect to flow j, $\underline{p}_{\underline{\ddot{c}}^K}(\mathcal{F}_{a_i^{SAP}}^j)$ can be evaluated from $\underline{p}_{\underline{\ddot{c}}^K}(\mathcal{S}_{a_i^{SAP}}^j)$ and $\underline{p}_{\underline{\ddot{c}}^K}$ as follows:

$$\underline{p}_{\underline{\ddot{c}}^K}(\mathcal{F}_{a_i^{SAP}}^j) = \underline{p}_{\underline{\ddot{c}}^K} - \underline{p}_{\underline{\ddot{c}}^K}(\mathcal{S}_{a_i^{SAP}}^j) \tag{4.6}$$

In a similar manner, we define the *constrained* pdf of $\ddot{c}_i^K$ for event $\mathcal{E}_{a_i^{SAP}}^j$ as follows:

$$\underline{p}_{\ddot{c}_i^K}(\mathcal{E}_{a_i^{SAP}}^j) = [\text{Prob}(\ddot{c}_i^K = C,\ \mathcal{E}_{a_i^{SAP}}^j \text{ occurs})]_{C=0}^{2^K-1}$$

where $\mathcal{E} \in \{\mathcal{S}, \mathcal{F}\}$. Then, we obtain the following Lemma *:

*Note that the notation $\prod_a^b$ refers to a sequence of matrix products in the order a,a+1,a+2,$\cdots$ b.

Lemma 4.1

$$\underline{p}_{\ddot{c}^K_{i+N}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+N}) \;=\; \underline{p}_{\ddot{c}^K_i}(\mathcal{E}^j_{a_i^{SAP}}) \times \prod_{u=i+1}^{i+N} \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_u^{SAP}})$$

Proof. We begin with the following equation:

$$\underline{p}_{\ddot{c}^K_{i+1}}(\mathcal{E}^j_{a_{i+1}^{SAP}}) \;=\; \underline{p}_{\ddot{c}^K_i} \times \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_{i+1}^{SAP}}) \tag{4.7}$$

Eq. (4.7) is true since the event $\mathcal{E}^j_{a_{i+1}^{SAP}}$ depends on $(\ddot{c}^K_{i+1}, \ddot{c}^K_i)$ and $\ddot{c}^K_{i+1}$ in turn depends on $\ddot{c}^K_i$. Hence, we can write the following:

$$\text{Prob}(\ddot{c}^K_{i+1} = m,\ \mathcal{E}^j_{a_{i+1}^{SAP}}) \;=\; \sum_l \text{Prob}(\ddot{c}^K_{i+1} = m,\ \mathcal{E}^j_{a_{i+1}^{SAP}} \mid \ddot{c}^K_i = l)\text{Prob}(\ddot{c}^K_i = l)$$

Multiplying both sides of Eq. (4.7) by $p_{\mathcal{E}^j_{a_i^{SAP}}}$, we obtain the following:

$$\underline{p}_{\ddot{c}^K_{i+1}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+1}) \;=\; \underline{p}_{\ddot{c}^K_i}(\mathcal{E}^j_{a_i^{SAP}}) \times \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_{i+1}^{SAP}})$$

Similarly, we can write the following:

$$\underline{p}_{\ddot{c}^K_{i+2}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i+1}^{i+2}) \;=\; \underline{p}_{\ddot{c}^K_{i+1}}(\mathcal{E}^j_{a_{i+1}^{SAP}}) \times \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_{i+2}^{SAP}})$$

Substituting Eq. (4.7) into Eq. (4.8), we obtain the following:

$$\underline{p}_{\ddot{c}^K_{i+2}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i+1}^{i+2}) \;=\; \underline{p}_{\ddot{c}^K_i} \times \prod_{u=i+1}^{i+2} \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_u^{SAP}})$$

Multiplying both sides of Eq. (4.8) by $p_{\mathcal{E}^j_{a_i^{SAP}}}$, we obtain the following:

$$\underline{p}_{\ddot{c}^K_{i+2}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+2}) \;=\; \underline{p}_{\ddot{c}^K_i}(\mathcal{E}^j_{a_i^{SAP}}) \times \prod_{u=i+1}^{i+2} \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_u^{SAP}})$$

This can be generalized to obtain Lemma 4.1. □ From Lemma 4.1, we have

$$\text{Prob}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+N} \text{ occurs}) \;=\; \sum_{C=0}^{2^K-1} \underline{p}_{\ddot{c}^K_{i+N}}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+N}) \tag{4.8}$$

$$\;=\; \underline{p}_{\ddot{c}^K_i}(\mathcal{E}^j_{a_i^{SAP}}) \times \prod_{u=i+1}^{i+N} \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{E}^j_{a_u^{SAP}}) \times \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Since the SAP is periodic with period R, $1 \leq i \leq R$, and i is uniformly distributed in this interval. Hence, un-conditioning Eq. (4.8) on i, we have the following:

$$\text{Prob}(\overbrace{\mathcal{E}^j, \mathcal{E}^j, \cdots, \mathcal{E}^j}^{N+1} \text{occurs}) = \frac{1}{R}\sum_{i=1}^{R} \text{Prob}(\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+N} \text{ occurs}) \quad (4.9)$$

4.2.2 Evaluation of p_{n^j}

p_{n^j} can be evaluated according to the following theorem:

Theorem 4.1

$$\begin{aligned} p_{n^j} &= \text{Prob}(\mathcal{S}^j, \overbrace{\mathcal{F}^j, \cdots, \mathcal{F}^j}^{N^j-1}, \mathcal{S}^j \text{occurs}) \\ &= \sum_{i=1}^{R} \underline{p}_{\ddot{c}_i^K}(\mathcal{S}^j_{a_i^{SAP}}) \times \prod_{u=i+1}^{i+N^j-1} \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{F}^j_{a_u^{SAP}}) \times \underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{S}^j_{a_{i+N^j}^{SAP}}) \times \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \end{aligned}$$

Proof. The HOL packet delay for flow j is N^j slots when consecutive successful transmissions of flow j take place N^j slots apart. Substituting $\{\mathcal{E}^j_{a_u^{SAP}}\}_{u=i}^{i+N^j}$ $= \{\mathcal{S}^j_{a_i^{SAP}}, \{\mathcal{F}^j_{a_{i+u}^{SAP}}\}_{u=1}^{N^j-1}, \mathcal{S}^j_{a_{i+N^j}^{SAP}}\}$ into Eq. (4.9), we obtain Theorem 4.1. □

We drop the subscript $\ddot{c}_i^K$ in the notation $\underline{p}_{\ddot{c}_i^K}(\mathcal{S}^j_{a_i^{SAP}})$ since in the computation of p_{n^j}, i always refers to a slot where $\mathcal{S}^j_{a_i^{SAP}}$ takes place. If $\underline{p}(\mathcal{S}^j) = [\underline{p}(\mathcal{S}^j_{a_1^{SAP}})$, $\underline{p}(\mathcal{S}^j_{a_2^{SAP}})$, $\cdots$, $\underline{p}(\mathcal{S}^j_{a_R^{SAP}})]$, then $\underline{p}(\mathcal{S}^j)$ can be evaluated based on a recurrence relation in terms of $\{\underline{p}_{\underline{=}\ddot{c}^K}(\mathcal{S}^j_{a_i^{SAP}})\}_{i=1}^{R}$ by the following theorem:

Theorem 4.2

$$\underline{p}(\mathcal{S}^j) = \underline{p}(\mathcal{S}^j) \times \underline{\underline{Inv}} \times \underline{\underline{Q}}$$

where $\underline{\underline{Inv}}$ *and* $\underline{\underline{Q}}$ *are matrices of size* $R \times R$ *such that each element is a sub-matrix of size* $2^K \times 2^K$*. These sub-matrices are given as follows:*

$$\underline{\underline{Inv}}(row, col) = \begin{cases} [\underline{\underline{I}} - \prod_{x=row+1}^{R} \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{F}^j_{a_x^{SAP}}) \times \prod_{x=1}^{row} \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{F}^j_{a_x^{SAP}})]^{-1}, & row = col; \\ \underline{\underline{Z}}, & \text{otherwise.} \end{cases}$$

$$\underline{\underline{Q}}(row, col) = \begin{cases} \prod_{x=row+1}^{col-1} \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{F}^j_{a_x^{SAP}}) \times \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{S}^j_{a_{col}^{SAP}}), & row < col; \\ \prod_{x=row+1}^{R} \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{F}^j_{a_x^{SAP}}) \times \prod_{x=1}^{col-1} \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{F}^j_{a_x^{SAP}}) \times \underline{\underline{p}}_{\ddot{c}^K}(\mathcal{S}^j_{a_{col}^{SAP}}), & \text{otherwise.} \end{cases}$$

where $\underline{\underline{I}}$ *and* $\underline{\underline{Z}}$ *are the* $2^K \times 2^K$ *Identity and Zeros matrices respectively.*

Proof. Since the allocation sequence, $\underline{a}_{SAP}$, is cyclic, if flow j transmits in slot i, then it will transmit next in slot $i+N$ only if $\mathcal{E} = \{\{\mathcal{F}^j_{a_{i+u}^{SAP}}\}_{u=1}^{N-1}, \mathcal{S}^j_{a_{i+N}^{SAP}}\}$ occurs. Hence, $\underline{p}(S^j_{a_{i+N}^{SAP}})$ can be evaluated by considering $1 \leq i \leq R$, and evaluating the probability of occurrence of $\mathcal{E}$ corresponding to each i. This forms the premise for the proof of Theorem 4.2. □

Based on Eq. (4.6), Theorems 4.1 and 4.2, p_{n^j} can be expressed in terms of $\{\underline{p}_{\ddot{c}^K}(\mathcal{S}^j_{a_i^{SAP}})\}_{i=1}^{R}$, whose evaluation depends on the channel and prediction parameters as well as the AS. We illustrate with the following example:

Example 4.1 *Let us assume AS = UA and* $p_{\hat{0}}$*=*$p_{\hat{1}}$*=1 (i.e., deterministic OSP). According to Eq. (4.5), to determine* $\underline{\underline{p}}_{\ddot{c}^K}(\mathcal{S}^j_{a_i^{SAP}})$*, we have to evaluate the diagonal matrices* $\underline{\underline{D}}^K_{i-1}(\mathcal{S}^j_{a_i^{SAP}})$ *and* $\underline{\underline{D}}^K_i(\mathcal{S}^j_{a_i^{SAP}})$*. Let* $\underline{d}^K_x(\mathcal{S}^j_{a_i^{SAP}})$ *be a row vector comprising the diagonal elements of* $\underline{\underline{D}}^K_x(\mathcal{S}^j_{a_i^{SAP}})$*, i.e.,*

$$\underline{d}^K_x(\mathcal{S}^j_{a_i^{SAP}}) = [\text{Prob}(\mathcal{S}^j_{a_i^{SAP}} \text{ occurs} \mid \ddot{c}^K_x = C)]_{C=0}^{2^K-1}$$

We begin by initializing $\underline{d}^K_x(\mathcal{S}^j_{a_i^{SAP}})$ *to a vector of zeros. For each C, the*

corresponding value of c_x^j for 1$\leq j \leq K$ (denoted $c_x^j(C)$) is given as follows:

$$c_x^j(C) = \begin{cases} 0, & \lceil \frac{C}{2^{j-1}} \rceil \equiv 1 (\text{modulo } 2); \\ 1, & \text{otherwise.} \end{cases} \quad (4.10)$$

Evaluation of $\underline{\boldsymbol{d}}_{i-1}^K(\mathbb{S}_{a_i^{SAP}}^j)$: *The value of $\ddot{c}_{i-1}^K$ determines the eligibility of each flow for transmission in slot i. We consider two cases:*

- $\boldsymbol{j = a_i^{SAP}}$ **:** *According to Eq. (4.1), flow j will be selected for transmission as long as it is eligible, i.e., c_{i-1}^j=0. Using Eq. (4.10), we can determine the range of $\ddot{c}_{i-1}^K$, Index, for which c_{i-1}^j=0, and consequently, $\underline{d}_{i-1}^K(\mathbb{S}_{a_i^{SAP}}^j)(Index)$=1.*

- $\boldsymbol{j \neq a_i^{SAP}}$ **:** *According to Eq. (4.1), flow j may be selected for transmission only if it is eligible while flow a_i^{SAP} is not, i.e., $c_{i-1}^{a_i^{SAP}}$=1 and c_{i-1}^j=0. As above, we can determine the range of $\ddot{c}_{i-1}^K$, Index, for which these conditions are satisfied using Eq. (4.10). Then, for each $C \in Index$, we determine the total number of eligible flows, $|\boldsymbol{G}|$, and $\underline{d}_{i-1}^K(\mathbb{S}_{a_i^{SAP}}^j)(Index)=\frac{1}{|\boldsymbol{G}|}$ according to Eq. (4.2).*

Evaluation of $\underline{\boldsymbol{d}}_i^K(\mathbb{S}_{a_i^{SAP}}^j)$: *The status of flow j's transmission attempt in slot i is determined by c_i^j; a successful transmission can occur only if c_i^j = 0. Using Eq. (4.10), we can determine the range of $\ddot{c}_i^K$, Index, for which c_i^j=0. Consequently, we have $\underline{d}_i^K(\mathbb{S}_{a_i^{SAP}}^j)(Index)$=1-$\gamma$.*

4.3 QoS Performance of CSD schedulers

In this section, we evaluate the wireless receiver buffer requirement and obtain closed-form expressions for the throughput and long-term fairness performance of the CSD_{AS}^{SAP} scheduler.

4.3.1 Wireless Receiver Buffer Requirement in terms of HOL packet delay statistics

In this section, we refer to Fig. 2.3 and quantify the relationship between the buffer size, b^j, required at wireless receiver j and the HOL packet delay jitter, σ_{n^j}, in order to sustain a packet dropping rate, β.

For the broadcast radio application considered in Section 3.4, we can assume a constant-rate server at the wireless receiver, i.e., $s^j = S^j$. Then, we have:

$$\begin{aligned} P_{Q^j} &= \text{Prob}(q^j \leq Q^j) \\ &= \text{Prob}(w^j \leq Q^j \cdot S^j) \end{aligned}$$

Hence, the minimum buffer size, b^j_{min}, required to ensure that the buffer overflow rate does not exceed β is given as follows:

$$b^j_{min} = \frac{\min\{W^j : \text{Prob}(w^j \leq W^j) \geq 1 - \beta\}}{S^j} \tag{4.11}$$

The exact evaluation of the waiting time distribution, P_{W^j}, is given in Appendix A. Let ρ denote the utilization factor at the wireless receiver, i.e., $\rho = \frac{S^j}{E[n^j]}$. Under high offered-load conditions (i.e., $\rho \approx 1$), Eq. (4.11) can be evaluated using the geometric approximation given in Eq. (A.6) in Appendix A, and b^j_{min} is given in terms of the moments of n^j as follows:

$$b^j_{min} \approx \frac{\lceil \frac{\ln \beta}{\ln(1 - \frac{1}{E[w^j]})} - 1 \rceil}{\rho \cdot E[n^j]} \tag{4.12}$$

where

$$E[w^j] = \frac{\sigma^2_{n^j}}{2E[n^j](1-\rho)}$$

and $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

4.3.2 Throughput-Fairness Tradeoff in CSD schedulers

Let us define the following notations:

$$p_{\mathcal{D}} \equiv \text{Prob(a flow } d\text{efers its transmission attempt)}$$
$$p_{\mathcal{S} \mid m} \equiv \text{Prob(a flow transmits } s\text{uccessfully} \mid m \text{ other eligible flows exist)}$$

According to the transmission heuristics of the CSD scheduler, a flow will defer any transmission attempt in any slot if it is not eligible for transmission. In addition, if no other eligible flows exist, a flow will transmit successfully if it is eligible and channel prediction is correct. Using the above heuristics, we can obtain the expressions for ($p_{\mathcal{D}}$,$p_{\mathcal{S} \mid 0}$) as follows:

$$\begin{aligned} p_{\mathcal{S} \mid 0} &= [p_c(0)p_{\hat{0}}[1-(1-p_c(0))g] + (1-p_c(0))(1-p_{\hat{1}})p_c(0)g](1-\gamma) \\ p_{\mathcal{D}} &= p_c(0)(1-p_{\hat{0}}) + (1-p_c(0))p_{\hat{1}} \end{aligned}$$

We have the following theorems:

Theorem 4.3 *The overall throughput achieved by a CSD_{AS}^{SAP} scheduler is independent of the SAP and can be expressed in terms of ($p_{\mathcal{D}}$,$p_{\mathcal{S} \mid 0}$) as follows:*

$$T = \begin{cases} p_{\mathcal{S} \mid 0} \cdot [1 + \frac{p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{1-p_{\mathcal{D}}}], & AS = UA; \\ p_{\mathcal{S} \mid 0} \cdot [1 + \frac{r^1}{r^K} \cdot \frac{p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(1-p_{\mathcal{D}})}], & AS = PA; \\ p_{\mathcal{S} \mid 0}, & AS = PA(h=0); \\ \frac{p_{\mathcal{S} \mid 0} \cdot (1-p_{\mathcal{D}}^{2h+1})}{1-p_{\mathcal{D}}}, & AS = PA(h \geq 1), K \text{ is odd}; \\ \frac{p_{\mathcal{S} \mid 0} \cdot (1-p_{\mathcal{D}}^{2h})}{1-p_{\mathcal{D}}}, & AS = PA(h \geq 1), K \text{ is even}. \end{cases}$$

Theorem 4.4 *For an input-homogeneous scenario, the CSD_{AS}^{SAP} scheduler is always long-term fair. On the other hand, for an input-heterogenous scenario, the long-term fairness performance of a CSD_{AS}^{SAP} scheduler can be expressed in terms of ($p_{\mathcal{D}}$,$p_{\mathcal{S} \mid 0}$) as follows:*

$$FM^{j,k} = \begin{cases} |\frac{p_{\mathcal{S} \mid 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})}(\frac{R-r^j}{r^j} - \frac{R-r^k}{r^k})|, & AS = UA; \\ 0, & AS = PA. \end{cases}$$

Proof. Let $T^{j|x_i}$ denote the throughput of flow j in slot i given x_i. According to Section 4.1, $T^{j|a_i^{SAP}}$=$p_{\mathcal{S}\mid 0}$ if a_i^{SAP}=j; otherwise, we have the following cases:

AS = UA : We have the following:

$$\begin{aligned} T^{j|a_i^{SAP}} &= p_{\text{Flow } a_i^{SAP} \text{ defers its transmission}} \cdot \\ &\quad \sum_{m=0}^{K-2} p_{\text{Flow } j \text{ transmits successfully} \mid m \text{ of remaining } K\text{-2 flows are eligible}} \cdot \\ &\quad p_{m \text{ of remaining } K\text{-2 flows are eligible}} \\ &= p_{\mathcal{D}} \cdot p_{\mathcal{S}\mid 0}\{p_{\mathcal{D}}^{K-2} + \frac{\binom{K-2}{1}p_{\mathcal{D}}^{K-3}(1-p_{\mathcal{D}})}{2} + \\ &\quad \frac{\binom{K-2}{2}p_{\mathcal{D}}^{K-4}(1-p_{\mathcal{D}})^2}{3} + \cdots + \frac{(1-p_{\mathcal{D}})^{K-2}}{K-1}\} \\ &= p_{\mathcal{D}} \cdot p_{\mathcal{S}\mid 0} \sum_{k=1}^{K-2} \frac{\binom{K-2}{k}(1-p_{\mathcal{D}})^k p_{\mathcal{D}}^{K-2-k}}{k+1} \\ &= \frac{p_{\mathcal{S}\mid 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})} \end{aligned}$$

Hence, we have the following:

$$T^{j|a_i^{SAP}} = \begin{cases} p_{\mathcal{S}\mid 0}, & j = a_i^{SAP}; \\ \frac{p_{\mathcal{S}\mid 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})}, & \text{otherwise}; \end{cases} \tag{4.13}$$

Un-conditioning on a_i^{SAP}, we obtain the following expression for T^j:

$$\begin{aligned} T^j &= \sum_{a_i^{SAP}=1}^{K} \frac{r^{a_i^{SAP}}}{R} T^{j|a_i^{SAP}} \\ &= \frac{r^j}{R} p_{\mathcal{S}\mid 0} + \frac{R-r^j}{R} \frac{p_{\mathcal{S}\mid 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})} \end{aligned}$$

AS = PA : Comparing Eq. (4.2) and Eq. (4.4), the expression for $T^{j|a_i^{SAP}}$

is obtained by introducing the factor $\frac{r^j r^1}{r^{a_i^{SAP}} r^K}$ into Eq. (4.13) as follows:

$$T^{j|a_i^{SAP}} = \begin{cases} p_{\mathcal{D}}, & j = a_i^{SAP}; \\ \frac{r^j}{r^{a_i^{SAP}}} \frac{r^1}{r^K} \frac{p_{\mathcal{S}\ |\ 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})}, & \text{otherwise}; \end{cases} \quad (4.14)$$

As a result, we obtain the following expression for T^j:

$$\begin{aligned} T^j &= \sum_{a_i^{SAP}=1}^{K} \frac{r^{a_i^{SAP}}}{R} T^{j|a_i^{SAP}} \\ &= \frac{r^j}{R} p_{\mathcal{S}\ |\ 0} [1 + \frac{r^1 p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{r^K(1-p_{\mathcal{D}})}] \end{aligned}$$

AS = PA(h) : We consider the following two cases:

- ***K* is odd :** According to Eq. (4.3), there are two candidate flows at each priority level. Let us assume that $v^j_{a_i^{SAP}}$=1, i.e., flow j has the highest priority to transmit if it is eligible and flow a_i^{SAP} is not eligible. However, flow j may have to contend for transmission if another flow k that shares its priority level (i.e., $v^k_{a_i^{SAP}}$=$v^j_{a_i^{SAP}}$) is also eligible. In a similar manner as with UA, we obtain the following expression:

$$\begin{aligned} T^{j|v^j_{a_i^{SAP}}=1} &= p_{\mathcal{D}} \cdot p_{\mathcal{S}\ |\ 0} \cdot [p_{\mathcal{D}} + \frac{(1-p_{\mathcal{D}})}{2}] \\ &= \frac{p_{\mathcal{D}} \cdot p_{\mathcal{S}\ |\ 0}(1+p_{\mathcal{D}})}{2} \end{aligned}$$

  Next, we assume that $v^j_{a_i^{SAP}}$=2. In this case, flow j can transmit only if flow a_i^{SAP} as well as both flows with priority level 1 are not eligible. This can be expressed as follows:

$$T^{j|v^j_{a_i^{SAP}}=2} = \frac{p_{\mathcal{D}}^3 \cdot p_{\mathcal{S}\ |\ 0}(1+p_{\mathcal{D}})}{2}$$

This can be generalized for flow j with priority level $v^j_{a_i^{SAP}}$ as follows:

$$T^{j|v^j_{a_i^{SAP}}} = \frac{p_{\mathcal{D}}^{2v^j_{a_i^{SAP}}-1} p_{\mathcal{S}\mid 0}(1+p_{\mathcal{D}})}{2}$$

Un-conditioning on a_i^{SAP}, and taking into account the priority threshold, h, we obtain the following expression for T^j:

$$\begin{aligned} T^j &= \frac{1}{K}T^{j|0} + \sum_{v^j_{a_i^{SAP}}=1}^{h} \frac{2}{K} T^{j|v^j_{a_i^{SAP}}} \\ &= \frac{p_{\mathcal{S}\mid 0}}{K} + \frac{2}{K} \sum_{v^j_{a_i^{SAP}}=1}^{h} p_{\mathcal{D}}^{2v^j_{a_i^{SAP}}-1} \cdot p_{\mathcal{S}\mid 0} \frac{1+p_{\mathcal{D}}}{2} \\ &= \frac{p_{\mathcal{S}\mid 0}(1-p_{\mathcal{D}}^{2h+1})}{K(1-p_{\mathcal{D}})} \end{aligned}$$

- ***K* is even :** According to Eq. (4.3), there are two candidate flows at each priority level > 1, and one candidate flow at priority level $= 1$. Let us assume that $v^j_{a_i^{SAP}}$=1, i.e., flow j has the highest priority to transmit if it is eligible and flow a_i^{SAP} is not eligible. This can be expressed as follows:

$$T^{j|v^j_{a_i^{SAP}}=1} = p_{\mathcal{D}} \cdot p_{\mathcal{S}\mid 0}$$

Next, we assume that $v^j_{a_i^{SAP}}$=2. In this case, flow j can transmit only if flow a_i^{SAP} as well as the flow with priority level 1 are not eligible. However, flow j may have to contend for transmission if another flow k that shares its priority level (i.e., $v^k_{a_i^{SAP}}=v^j_{a_i^{SAP}}$) is also eligible. This can be expressed as follows:

$$T^{j|v^j_{a_i^{SAP}}=2} = p_{\mathcal{S}\mid 0} \cdot p_{\mathcal{D}}^2 \frac{1+p_{\mathcal{D}}}{2}$$

This can be generalized for flow j with priority level $v^j_{a_i^{SAP}}$ as follows:

$$T^{j|v^j_{a_i^{SAP}}} = \begin{cases} p_{S\mid 0} \cdot p_{\mathcal{D}}, & v^j_{a_i^{SAP}} = 1; \\ p_{S\mid 0} \cdot p_{\mathcal{D}}^{2(v^j_{a_i^{SAP}}-1)} \frac{1+p_{\mathcal{D}}}{2}, & v^j_{a_i^{SAP}} > 1. \end{cases}$$

Un-conditioning on a_i^{SAP}, and taking into account the priority threshold, h, as in the case when K is odd, we obtain the following expression for T^j:

$$T^j = \begin{cases} \frac{p_{S\mid 0}}{K}, & h = 0; \\ \frac{p_{S\mid 0}(1-p_{\mathcal{D}}^{2h})}{K(1-p_{\mathcal{D}})}, & h > 0; \end{cases}$$

Substituting the expressions of T^j for each AS into Eq. (2.4), we verify the expressions given in Theorem 4.3. Theorem 4.4 can be verified by substituting the expressions for T^j for each AS into Eq. (2.5). □

4.4 Numerical Results

In this section, we apply our performance analysis framework to investigate the performance trade-offs of the CSD^{SAP}_{AS} scheduler in a channel-homogeneous scenario. We assume a deterministic OSP for channel prediction and γ=0.

4.4.1 Input-homogeneous Scenario

For this scenario, according to Section 4.1.2, SAP = RR and hence, we drop the superscript SAP for simplicity of notations. We have two variants of the CSD scheduler, namely, CSD_{UA} and $CSD_{PA(h)}$. Since the scheduling scenario is homogeneous, $p_{n^j} = p_n$, $1 \leq j \leq K$.

Admissibility of CSD_{AS} scheduler

We consider the following admissibility criteria:

- **Efficiency Requirement :** We specify an efficiency requirement in terms of a minimum overall throughput, T_{min}. This imposes a lower bound on K since an improvement in the channel efficiency of CSD schedulers is achieved through a gain in statistical multiplexing, which increases with K. By substituting the expressions for T using Theorem 4.3, we illustrate the constraint in the LHS of Fig. 4.2 for T_{min}=0.8.

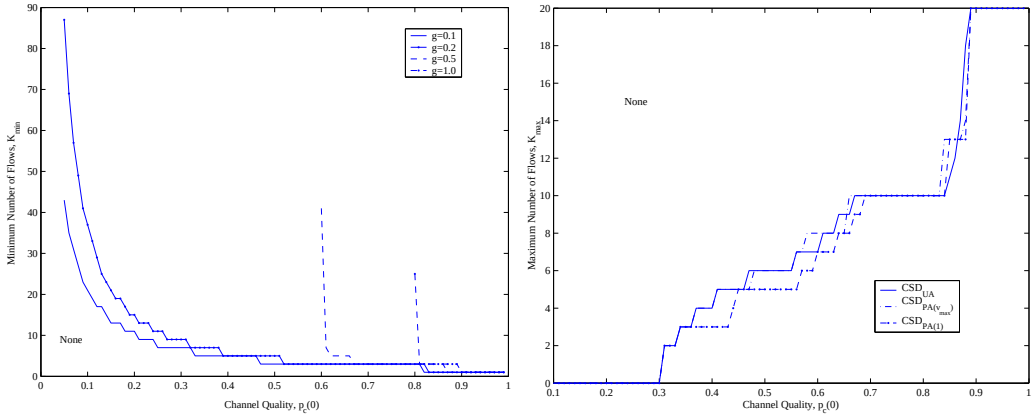

Figure 4.2: Admissibility of the CSD_{AS} scheduler to satisfy (left) efficiency requirement, T_{min}=0.8 and (right) real-time requirement, 1-$P_{N_{max}=20} \leq 0.2$.

Each graph indicates the minimum value of K (denoted K_{min}) as a function of $p_c(0)$ such that the minimum throughput requirement is met for a given channel agility, g. We denote by *NONE* the region where the CSD scheduler is inadmissible. As expected, the admissible region is increased as the channel quality is improved.

We notice a steep transition from one region to another for $p_c(0) \approx T_{min}$ when the channel is uncorrelated (g=1.0). This is because the prediction accuracy is poor, and as a result, the throughput achievable by the CSD scheduler corresponds to that of a channel-independent scheduler, which is solely determined by the channel quality, and is

independent of K. As the channel becomes persistent, the slope of each graph becomes more gradual, resulting in a larger admissible region.

- **Real-time Requirement :** The real-time QoS requirement in Eq. (2.6) imposes a constraint on the maximum number of flows that can be supported at a given value of $p_c(0)$. We consider the CSD_{UA}, $CSD_{PA(v_{max})}$ and $CSD_{PA(1)}$ schedulers in an uncorrelated channel, and illustrate this constraint for $N_{max} = 20$ and $\gamma = 0.2$ in the RHS of Fig. 4.2. The graph for each scheduler indicates the maximum value of K (denoted K_{max}) at each value of $p_c(0)$ such that the real-time QoS requirement is met. We denote by *NONE* the region where none of the above schedulers is admissible.

  We note that as channel conditions improve, each scheduler can handle more input-flows without violating the real-time QoS requirement. However, the admissible region of the CSD scheduler is relatively invariant with the AS.

Comparison of wireless receiver buffer requirement for various AS

We plot b_{min} as a function of K for g=1.0, $\beta = 0.01$ and $\rho = 0.99$ in Fig. 4.3. For each scheduler, the buffer requirement increases sharply as K increases, but levels off at large values of K. This is because the AS has a larger pool of eligible flows to choose from for transmission, and therefore, the delay variation is increased.

In contrast with the real-time requirement, which is relatively invariant with the AS, the buffer requirement is significantly reduced when PA is used instead of UA.

4.4.2 Input-heterogenous Scenario

In this section, we illustrate the trade-off between throughput and fairness performance of the CSD_{AS}^{SAP} scheduler. We drop the superscript SAP since

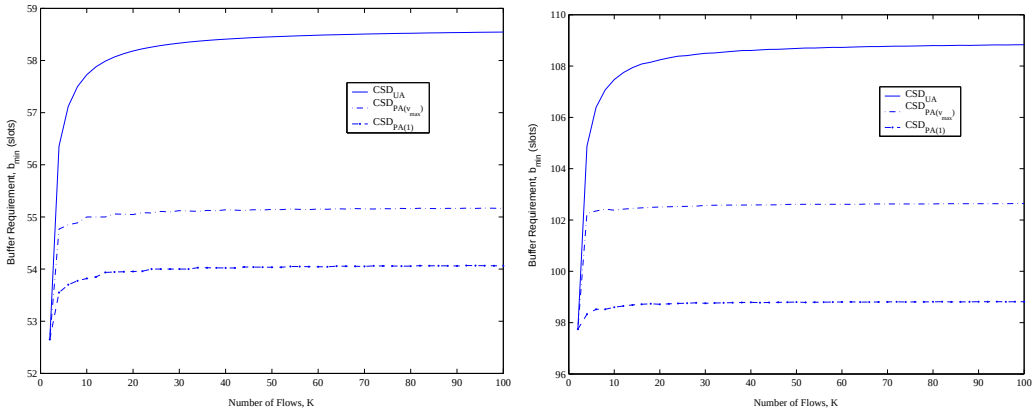

Figure 4.3: Wireless receiver buffer requirement for the CSD_{AS} scheduler with different AS for g=1.0, $p_c(0) = 0.9$ (left) and 0.8 (right).

these metrics are independent of the SAP.

Given the channel parameters ($p_c(0)$,g), using Theorem 4.4, we can define the worst-case fairness metric, FM_{max}, for the CSD_{UA} scheduler as follows:

$$\begin{aligned} FM_{max} &= \max_{j,k} FM^{j,k} \qquad (4.15) \\ &= |\frac{p_{\mathcal{S}\mid 0} \cdot p_{\mathcal{D}}(1-p_{\mathcal{D}}^{K-1})}{(K-1)(1-p_{\mathcal{D}})}(\frac{R-r^1}{r^1} - \frac{R-r^K}{r^K})| \end{aligned}$$

We plot the overall throughput for the CSD_{UA} and CSD_{PA} schedulers using Theorem 4.3 for various $\underline{r}$ and g=0.1 as a function of $p_c(0)$ in the LHS of Fig. 4.4, where the notation $AS(2,3,4)$ denotes the CSD_{AS} scheduler with $\underline{r}$ = [2,3,4]. We plot FM_{max}, for the CSD_{UA} scheduler in the RHS of Fig. 4.4.

As the deviation in the input-flow rates, r^K-r^1, is increased, the throughput obtained with PA is degraded; at the same time, the unfairness due to UA is increased correspondingly. As channel conditions improve, the overall throughput is increased since the likelihood of successful transmission is increased. The unfairness due to UA is increased initially until $p_c(0) \approx 0.75$ and is reduced subsequently.

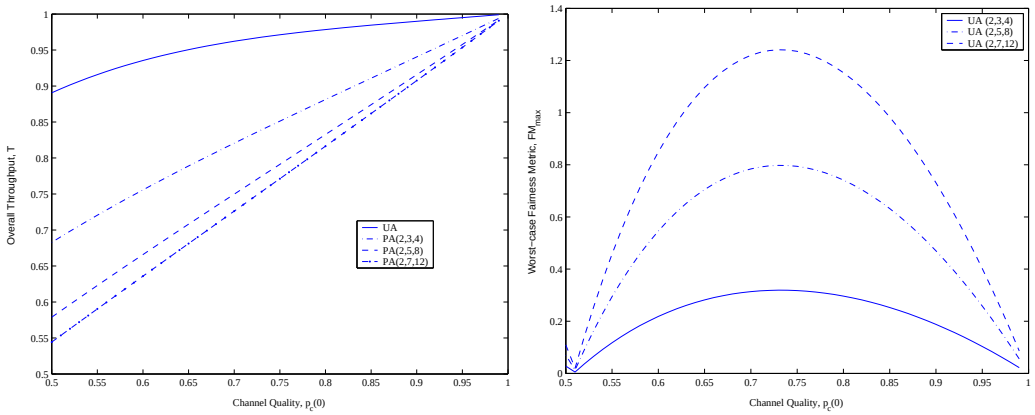

Figure 4.4: Throughput (left) and Worst-case Fairness Metric (right) of the CSD_{AS} scheduler as a function of channel quality, $p_c(0)$, for g=0.1.

4.5 Summary

In this chapter, we present a framework for the performance analysis of a general K-flow CSD scheduler based on the notion of *constrained* state-transition matrices, which permits the efficient evaluation of performance metrics without the tedious transient computations of Markov analysis. We establish the requirement for the wireless receiver buffer in terms of statistics of the HOL packet delay. In addition, we propose and analyze novel schemes that embed fairness provision into the AS, thus substituting for the functions of the FM in the generic wireless scheduler model.

Using numerical results, we evaluate the admissibility of the CSD scheduler under an efficiency constraint and a real-time QoS constraint, and also the buffer requirement at each wireless receiver with different AS. Under a homogeneous scenario, we expose the trade-offs amongst channel efficiency, QoS and buffer requirement between prioritized and uniform arbitration. In addition, we highlight the trade-off between channel efficiency and long-term fairness in an input-heterogeneous scenario with both arbitration schemes.

Chapter 5

A Hybrid Channel-State Dependent / Fair-Aggregation Scheduler for Heterogeneous Channels

Like most wireless schedulers proposed in the literature, the CSD scheduler that we define in this study predicts the instantaneous channel state of each flow and uses this information in its scheduling decision. While the channel state c^j characterizes the short-term behavior of the channel of flow j, the parameters $(p_c^j(0),g^j)$ describe its behavior over a longer period of time. As with the channel state, these parameters can be predicted [28] based on channel measurements.

For a homogeneous scheduling scenario, numerical results in Chapter 3 suggest that while the FA scheduler achieves better QoS performance when the channel is uncorrelated, the CSD scheduler is superior when the channel is persistent. Hence, the channel agility offers an additional dimension of information that can be incorporated into the CSD scheduler design to improve its performance in a channel-heterogeneous scenario. In this chapter, we propose and present a performance analysis of a scheduler that manifests this novel idea, and verify its gain in performance over the original CSD scheduler using numerical results.

5.1 Definition of (K,η) CSD-FA scheduler

In contrast to Chapters 3 and 4, where $\underline{\kappa} = [1,\cdots,1]$, we consider a two-class K-flow channel-heterogeneous scenario defined as follows, where $\epsilon \approx 0$:

$$\begin{aligned} \underline{\kappa} &= [\eta, K-\eta] \\ \underline{\tilde{g}} &= [\epsilon, 1.0] \\ \underline{\tilde{r}} &= [1, 1] \end{aligned} \tag{5.1}$$

We propose a (K,η) CSD-FA scheduler that achieves the relative merits of each scheduler for the scenario defined in Eq. (5.1) by partitioning the input-flows into $\mathbf{C}^1 = \{1,2,\cdots,\eta\}$ and $\mathbf{C}^2 = \{\eta{+}1,\eta{+}2,\cdots,K\}$ prior to scheduling.

The mechanism of the scheduler comprises two stages. In the first stage, the scheduler dispatches packets from flows in $\mathbf{C}^2$ in a round robin manner into a single queue. If we denote this queue by η', then the second stage is equivalent to a η+1-flow CSD scheduler (with flow composition given by $\mathbf{C}^1 \cup \eta'$), where $\underline{r} = [1\ \cdots,1,K\text{-}\eta]$ and $\underline{g} = [\epsilon,\cdots,\epsilon,1.0]$. This is illustrated in Fig. 5.1.

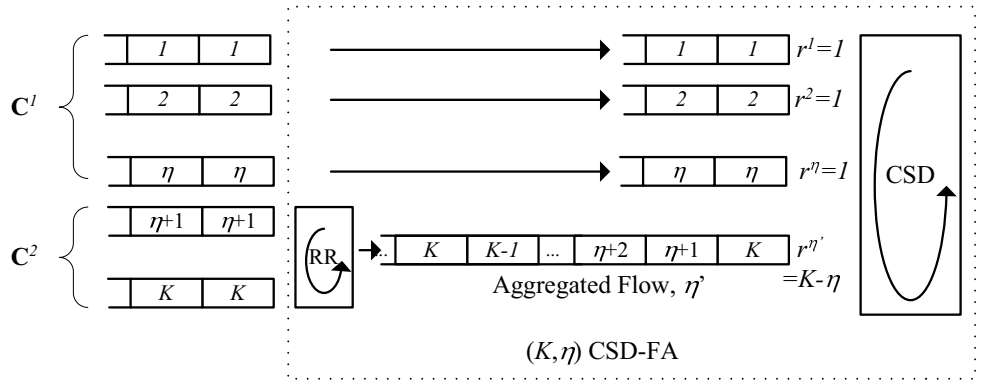

Figure 5.1: Hybrid CSD-FA Scheduler Model: Flows in $\mathbf{C}^2$ are aggregated into a single flow η'; Flows in $\mathbf{C}^1 \cup \eta'$ are then scheduled by a η+1-flow CSD scheduler with $\underline{r} = [1,1,\cdots\ 1,K\text{-}\eta]$ and $\underline{g} = [\epsilon,\cdots,\epsilon,1.0]$.

We note that the (K,η) CSD-FA scheduler is in fact a generalization of the K-flow CSD scheduler and a K-flow FA scheduler; a (K,K) CSD-FA scheduler

is equivalent to a K-flow CSD scheduler while a (K,0) CSD-FA scheduler corresponds to a K-flow FA scheduler.

We illustrate the mechanism of our proposed scheduler by considering a (4,2) *CSD-FA*$^{WRR}_{UA}$ scheduler with deterministic OSP, AS = UA, SAP=WRR and γ=0. According to the above description, the (4,2) *CSD-FA*$^{WRR}_{UA}$ scheduler is equivalent to a 3-flow CSD^{WRR}_{UA} scheduler with $\underline{r}$ = [1,1,2] and $\underline{g}$ = [ϵ,ϵ,1.0], as depicted in the LHS of Fig. 5.2. The allocation sequence, $\underline{a}^{WRR}$, is given as follows:

$$\underline{a}^{WRR} = [\cdots, 2, 2', 2', 1, 2, 2', 2', 1, \cdots] \tag{5.2}$$

Let us assume the following initial conditions: a_0^{WRR}=1 and a flow 3 packet is HOL at flow 2' at the end of slot 0. If TX_i denotes the flow index of the packet transmitted in slot i, then the evolution of TX corresponding to some channel process $\underline{\ddot{c}}^K$ is depicted in the RHS of Fig. 5.2.

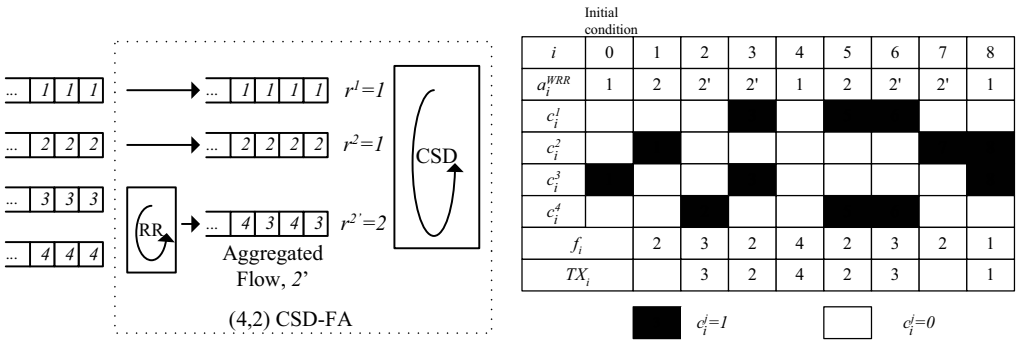

Figure 5.2: Illustration of the mechanism of a (4,2) CSD-FA scheduler: Architecture (left) and illustration of the mechanism (right) of a (4,2) *CSD-FA*$^{WRR}_{UA}$ scheduler with deterministic OSP, AS = UA and SAP = WRR.

Since a_0^{WRR}=1, according to Eq. (5.2), a_1^{WRR}=2; similarly, since c_0^2=0, according to Eq. (2.3), $\hat{c}_1^2$=0. Hence, according to Eq. (4.1), flow 2 is selected for transmission. However, since c_1^2=1, the transmission is unsuccessful. The next slot is allocated to flow 2'. Since the HOL packet of flow 2' belongs to flow 3 and c_1^3=0, flow 2' is selected for transmission. The transmission is successful since c_2^3=0.

Slot 3 is again allocated to flow 2' according to Eq. (5.2). However, since its HOL packet belongs to flow 4 and c_2^4=1, $\hat{c}_3^4$=1, and hence its transmission is deferred. Since c_2^1=c_2^2=0, $\hat{c}_3^1$=$\hat{c}_3^2$=0, and according to Eq. (4.2), flow 1 and 2 are equally likely to be selected for transmission. We assume that flow 2 is selected, and its transmission is successful since c_3^2=0. Subsequent values of TX can be evaluated in a similar manner.

5.2 Performance Analysis of (K,η) CSD-FA scheduler

In this section, we first derive p_{n^j} for $j \in \mathbf{C}^1$ using the framework presented in Chapter 4. Subsequently, we detail the analysis to derive the corresponding p_{n^j} for $j \in \mathbf{C}^2$.

5.2.1 Evaluation of p_{n^j}, $j \in \mathbf{C}^1$

According to Section 5.1, we can define an equivalent η+1-flow CSD_{UA}^{WRR} scheduling scenario with $\underline{g}$=[ϵ,$\cdots$,ϵ,1.0] and $\underline{r}$=[1,$\cdots$,1,K-η]. Using the framework developed in Chapter 4, we can evaluate p_{n^j}, $j \in \mathbf{C}^1$.

In fact, if we define the probabilistic parameters ($p_{\mathcal{S}^x \mid m}$,$p_{\mathcal{D}}$) (as in Theorem 4.3) as follows:

$$\begin{aligned} p_{\mathcal{D}} &\equiv \text{Prob(a flow } d\text{efers its transmission attempt)} \\ p_{\mathcal{S}^x \mid m} &\equiv \text{Prob(a flow} \in \mathbf{C}^x \text{ transmits } s\text{uccessfully} \mid m \\ &\quad \text{other eligible flows exist)} \end{aligned}$$

then $E[n^j]$ can be expressed in terms of ($p_{\mathcal{S}^1 \mid 0}$,$p_{\mathcal{D}}$) by the following theorem:

Theorem 5.1 *For the scheduling scenario defined in Eq. (5.1), the expected HOL packet delay for flow $j \in \mathbf{C}^1$ for a (K,η) CSD-FA$_{UA}^{WRR}$ scheduler is given*

as follows:

$$E[n^j] = \frac{K \cdot \eta(1 - p_{\mathcal{D}})}{p_{\mathcal{S}^1 \mid 0}[\eta(1 - p_{\mathcal{D}}) + (K-1)(p_{\mathcal{D}} - p_{\mathcal{D}}^{\eta+1})]}$$

where ($p_{\mathcal{S}^1 \mid 0}$,$p_{\mathcal{D}}$) *can be expressed in terms of* ($p_c(0)$,ϵ) *and* ($p_{\hat{0}}$,$p_{\hat{1}}$) *as follows:*

$$\begin{aligned} p_{\mathcal{S}^1 \mid 0} &= p_c(0)[p_{\hat{0}}(1 - \epsilon + \epsilon \cdot p_c(0)) + (1 - p_c(0))(1 - p_{\hat{1}})\epsilon](1 - \gamma) \\ p_{\mathcal{D}} &= p_c(0)(1 - p_{\hat{0}}) + (1 - p_c(0))p_{\hat{1}} \end{aligned}$$

The proof of Theorem 5.1 is similar to that of Theorem 4.3 and can be found in [43].

5.2.2 Evaluation of p_{n^j}, $j \in \mathbf{C}^2$

In order to simplify the analysis, we assume that any flow $j \in \mathbf{C}^2$ *only* transmits in slots allocated to $\mathbf{C}^2$. According to Eq. (5.1), since $g^j = 1.0$, $\forall j \in \mathbf{C}^2$,

$$p_{\mathcal{S}^2 \mid 0} = p_c(0)[p_{\hat{0}}p_c(0) + (1 - p_c(0))(1 - p_{\hat{1}})](1 - \gamma)$$

Let us denote $p_{\overline{\mathcal{S}^2} \mid 0}$ = 1-$p_{\mathcal{S}^2 \mid 0}$ as the probability that no successful flow j transmission occurs in a given slot, where $j \in \mathbf{C}^2$, given that there are no other eligible flows.

Assume that flow j transmits in slot i, $1 \leq i \leq \kappa^2$, where κ^2 is the number of flows with uncorrelated channels. From Fig. 5.3, we note that κ^2-1 packets, one from every other flow, must be transmitted before the next flow j packet transmits in slot k, where $1 \leq k$ (modulo K)$\leq \kappa^2$. Since there are κ^2 available transmission slots in the interval $[i{+}1{:}i{+}K]$, we have $k \geq K{+}i$.

Over the interval $[i{+}1{:}k{-}1]$, if we write $k{=}x{\cdot}\ K{+}y$, then there are $x{\cdot}\ \kappa^2{+}y$-i-1 available transmission slots in this interval, out of which κ^2-1 slots must contain successful transmissions. In addition, since the scheduling scenario is homogeneous with respect to flows in $\mathbf{C}^2$, under steady-state conditions,

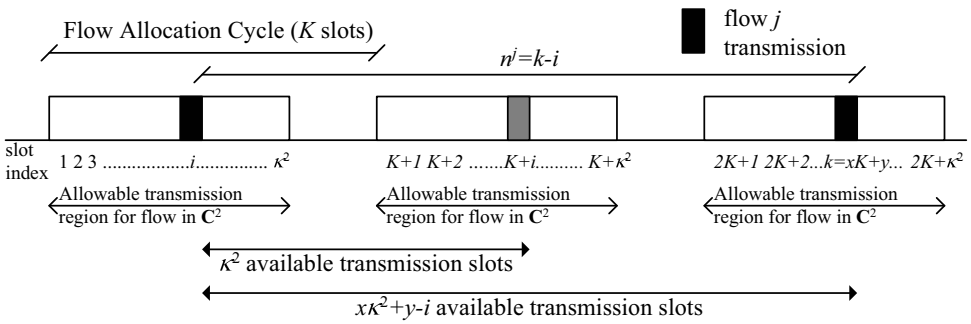

Figure 5.3: Illustration of the evaluation of p_{n^j} for each flow $j \in \mathbf{C}^2$.

i is uniformly distributed in the interval $1 \leq i \leq \kappa^2$. Therefore, we can write the following for $k \geq K+i$ and $1 \leq y,\ i \leq \kappa^2$:

$$\text{Prob}(k = x \cdot K + y) \quad = \quad \frac{\binom{x \cdot \kappa^2 + y - i - 1}{\kappa^2 - 1} p_{S^2 \mid 0}^{\kappa^2 - 1} p_{\overline{S^2} \mid 0}^{y-i} p_{S^2 \mid 0}}{\kappa^2}$$

Since $n^j = k\text{-}i$, p_{n^j} is obtained for $N^j \geq K$ and $1 \leq y,\ i \leq \kappa^2$ as follows:

$$p_{n^j}(x \cdot K + y - i) \quad = \quad \frac{\binom{\kappa^2 + y - i - 1}{\kappa^2 - 1} p_{S^2 \mid 0}^{\kappa^2} p_{\overline{S^2} \mid 0}^{y-i}}{\kappa^2} \tag{5.3}$$

Using Eq. (5.3), we obtain an expression for $E[n^j]$ in the following theorem:

Theorem 5.2 *For the scheduling scenario defined in Eq. (5.1), the expected HOL packet delay for flow $j \in \boldsymbol{C}^2$ for a (K,η) CSD-FA$_{UA}^{WRR}$ scheduler is given as follows:*

$$E[n^j] = \frac{K}{p_{S^2 \mid 0}}$$

Proof. Using Eq. (5.3), we compute $E[n^j]$ as follows:

$$\begin{aligned} E[n^j] \quad &= \quad \sum_{i=1}^{\kappa^2} \Big(\sum_{x=1}^{1} \sum_{y=i}^{\kappa^2} + \sum_{x=2}^{\infty} \sum_{y=1}^{\kappa^2} \Big) [x \cdot K + y - i] \cdot p_{n^j}(x \cdot K + y - i) \\ &= \quad \sum_{i=1}^{\kappa^2} \Big(\sum_{x=1}^{1} \sum_{y=i}^{\kappa^2} + \sum_{x=2}^{\infty} \sum_{y=1}^{\kappa^2} \Big) \frac{p_c(0)^{\kappa^2} [x \cdot K + y - i] \binom{\kappa^2 + y - i - 1}{\kappa^2 - 1} p_c(1)^{y-i}}{\kappa^2} \end{aligned}$$

We can simplify the above expression by noting that the exponent of the term $p_c(1)$ ranges from 0 to ∞, and by evaluating the coefficient of $\{p_c(1)^z\}_{z=0}^{\infty}$, we obtain the following expression:

$$E[n^j] = \frac{p_c(0)^{\kappa^2}}{\kappa^2} \sum_{z=0}^{\infty} K(\kappa^2+z)\binom{z+\kappa^2-1}{\kappa^2-1} p_c(1)^z \tag{5.4}$$

From binomial theorem, we have the following result:

$$\sum_{z=0}^{\infty} \binom{z+x}{x} y^z = \frac{1}{(1-y)^{x+1}} \tag{5.5}$$

Differentiating Eq. (5.5) with respect to y, we obtain

$$\sum_{z=0}^{\infty} z\binom{z+x}{x} y^{z-1} = \frac{x+1}{(1-y)^{x+2}} \tag{5.6}$$

Substituting Eq. (5.5) and Eq. (5.6) into Eq. (5.4), we obtain the expression for $E[n^j]$ as given in Theorem 5.2. □

5.3 Numerical Results

In this section, we compare the overall throughput as well as the wireless receiver buffer requirement between a (K,η) *CSD-FA*$^{WRR}_{UA}$ scheduler and a K-flow CSD^{WRR}_{UA} scheduler for the scheduling scenario given by Eq. (5.1). We denote the metric x corresponding to scheduler π by x_π. We assume a deterministic OSP for channel prediction and γ =0.

Based on Theorem 5.1 and 5.2, we have the following expression for $E[n^j]$ for the (K,η) *CSD-FA*$^{WRR}_{UA}$ scheduler:

$$E[n^j]_{CSD-FA^{WRR}_{UA}} = \begin{cases} \frac{K\cdot\eta(1-p_{\mathcal{D}})}{p_{\mathcal{S}^1\,|\,0}[\eta(1-p_{\mathcal{D}})+(K-1)(p_{\mathcal{D}}-p_{\mathcal{D}}^{\eta+1})]}, & j \in \mathbf{C}^1; \\ \frac{K}{p_{\mathcal{S}^2\,|\,0}}, & j \in \mathbf{C}^2. \end{cases}$$

Hence, according to Eq. (2.4), $T_{CSD-FA_{UA}^{WRR}}$ can be evaluated as follows:

$$\begin{aligned} T_{CSD-FA_{UA}^{WRR}} &= \sum_{j=1}^{K} \frac{1}{E[n^j]} \qquad (5.7) \\ &= \frac{\eta \cdot p_{S^{1} \mid 0}[\eta(1-p_{\mathcal{D}}) + (K-1)(p_{\mathcal{D}} - p_{\mathcal{D}}^{\eta+1})]}{K \cdot \eta(1-p_{\mathcal{D}})} + \frac{(K-\eta) \cdot p_{S^{2} \mid 0}}{K} \\ &= \frac{1}{K}[\frac{p_{S^{1} \mid 0}[\eta(1-p_{\mathcal{D}}) + (K-1)(p_{\mathcal{D}} - p_{\mathcal{D}}^{\eta+1})]}{(1-p_{\mathcal{D}})} + (K-\eta)p_{S^{2} \mid 0}] \end{aligned}$$

The corresponding expression for $T_{CSD_{UA}^{WRR}}$ can be evaluated using Theorem 4.3 and is given as follows:

$$T_{CSD_{UA}^{WRR}} = \frac{1-p_{\mathcal{D}}^{K}}{(1-p_{\mathcal{D}})K}[\eta \cdot p_{S^{1} \mid 0} + (K-\eta)p_{S^{2} \mid 0}]$$

For a channel-heterogenous scenario, $b^j \neq b^k$ for $j \neq k$. Therefore, we evaluate the average buffer requirement of scheduler π, $\check{b}_{\pi}$, defined as follows:

$$\check{b}_{\pi} = \frac{1}{K}\sum_{j=1}^{K} b_{\pi}^{j}$$

5.3.1 Comparison of Throughput and Buffer Requirement of CSD-FA and CSD scheduler

For a given K, the metrics T and $\check{b}$ depend on the flow composition, η, as well as the channel parameters, $p_c(0)$ and ϵ. We illustrate the effects of each parameter on T and $\check{b}$ for $K = 7$, $\beta = 0.01$ and $\rho = 0.99$.

Effects of flow composition

We consider the variation of $\check{b}$ and T with η for $p_c(0)$=0.9 and ϵ=0.1 in Fig. 5.4. As the composition of flows with persistent channels (i.e., η) is increased, T is increased since the accuracy of channel prediction is better for persistent channels. This reduces the likelihood of a wasted slot due to

erroneous prediction. Compared to T_{CSD}, the throughput degradation due to flow-aggregation is relatively invariant with η and is within 2 %.

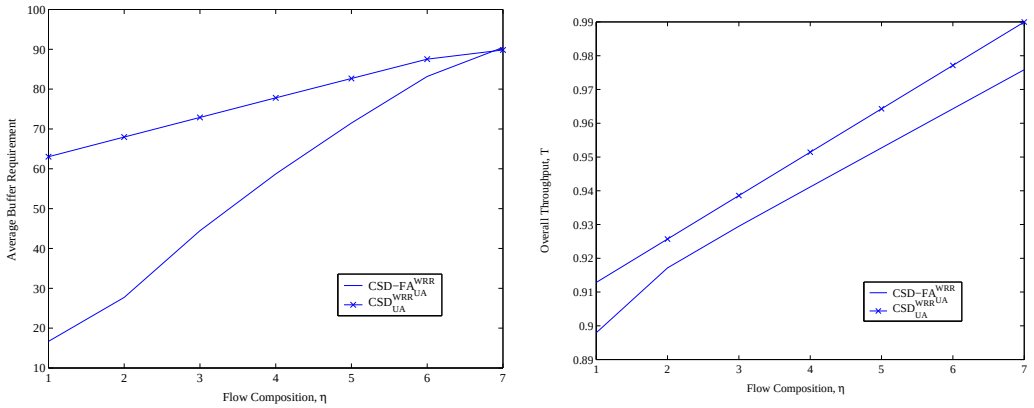

Figure 5.4: Effects of flow composition on average buffer requirement (left) and overall throughput (right) of CSD schedulers for $p_c(0)$=0.9 and ϵ = 0.1.

Since flows with uncorrelated channels have lower delay variation (according to numerical results presented in Chapter 3), the average buffer requirement is increased as the proportion of flows with persistent channels is increased. However, $\check{b}_{CSD-FA^{WRR}_{UA}} \leq \check{b}_{CSD^{WRR}_{UA}}$ due to flow-aggregation, and the resultant reduction in buffer requirement is significant (up to 75%) for small values of η.

We note that when $\eta = K$, both schedulers are identical, and hence they should achieve the same performance in terms of buffer requirement and overall throughput. However, the discrepancy in Fig. 5.4 is due to the assumption made in Section 5.2.2, which results in a conservative approximation for the overall throughput for the *CSD-FA* scheduler.

Effects of channel quality

Next, we consider the variation of $\check{b}$ and T with $p_c(0)$ for η=3 and ϵ=0.1 in Fig. 5.5. As the channel quality is improved (i.e., $p_c(0)$ is increased), T is increased since more transmission attempts will occur for a given interval of slots and the proportion of slots with successful transmissions will be

increased. We note that flow-aggregation actually results in a slight gain in throughput compared to the CSD_{UA}^{WRR} scheduler when the channel quality is poor ($p_c(0)$ <0.7). This trend is reversed when channel conditions improve. However, the difference in throughput performance between both schedulers is marginal (within 2 %).

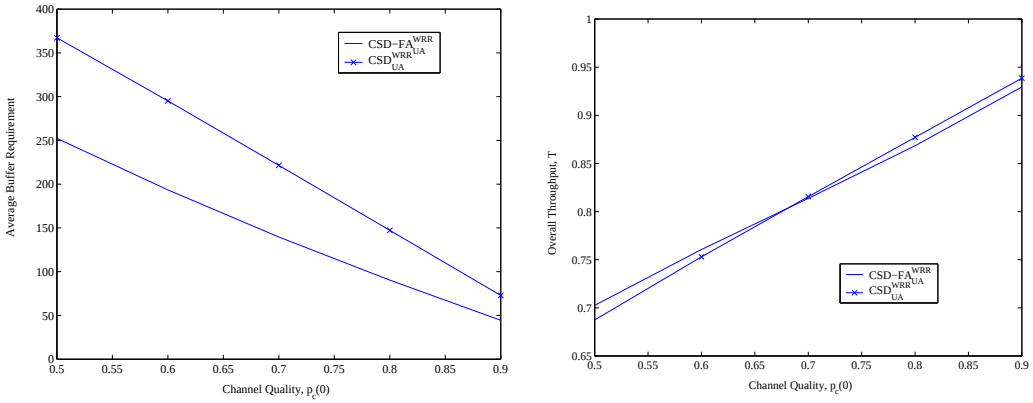

Figure 5.5: Effects of channel quality on average buffer requirement (left) and overall throughput (right) of CSD schedulers for η=3 and $\epsilon = 0.1$.

The buffer requirement is reduced as the channel quality is improved, since delay variation is reduced as flows are more likely to transmit in slots allocated to them. The reduction in buffer requirement as a result of flow-aggregation is significant (up to 30%).

Effects of channel agility

Lastly, we consider the variation of $\check{b}$ and T with ϵ for $p_c(0)$=0.9 and η=3 in Fig. 5.6. As the channel for $\mathbf{C}^1$ flows becomes less persistent (i.e., ϵ is increased), T is reduced since the accuracy of channel prediction is reduced. This increases the likelihood of a wasted slot due to erroneous prediction. Compared to $T_{CSD_{UA}^{WRR}}$, the throughput degradation due to flow-aggregation is relatively invariant with ϵ and is within 2 %.

Since the buffer requirement of any flow $\in \mathbf{C}^2$ is independent of ϵ, the met-

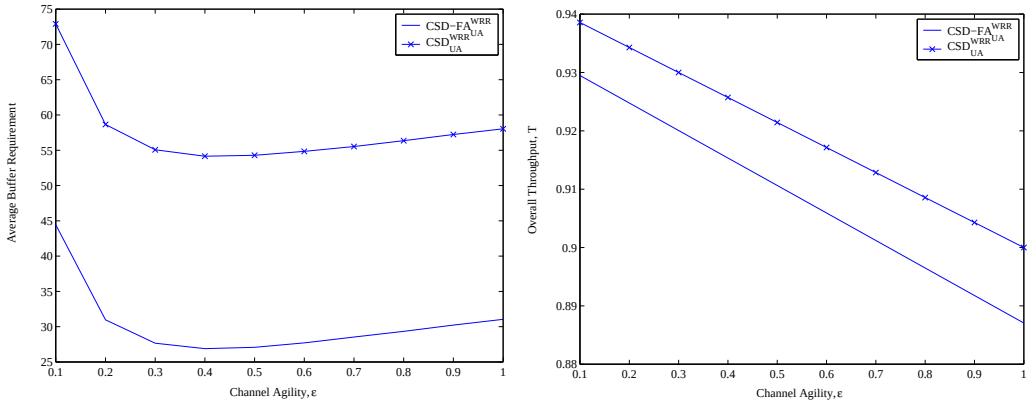

Figure 5.6: Effects of channel agility on average buffer requirement (left) and overall throughput (right) of CSD schedulers for η=3 and $p_c(0) = 0.9$.

ric $\check{b}_{CSD-FA_{UA}^{WRR}}$ is determined by the variation of the buffer requirement of flows $\in \mathbf{C}^1$. It is interesting to note that $\check{b}$ for both schedulers is reduced initially as ϵ is increased, but is increased with further increase in ϵ. However, $\check{b}_{CSD-FA_{UA}^{WRR}} \leq \check{b}_{CSD_{UA}^{WRR}}$, and the reduction in buffer requirement is significant (up to 45 %).

5.3.2 Discussion

A common observation from Section 5.3.1 is a trade-off between throughput and buffer requirements between the $CSD\text{-}FA_{UA}^{WRR}$ and CSD_{UA}^{WRR} scheduler: The $CSD\text{-}FA_{UA}^{WRR}$ scheduler results in a significant reduction in the wireless receiver buffer requirement at the expense of reduced throughput compared to the CSD_{UA}^{WRR} scheduler. In fact, since the throughput degradation is marginal compared to the reduction in buffer requirement, the $CSD\text{-}FA_{UA}^{WRR}$ scheduler is effective in maintaining good overall performance.

Our current analysis assumes a simplistic WRR scheduler for the SAP. In the next chapter, we study the performance of $\pi \in \mathbf{F}^{\underline{r}}$ in terms of its delay variation and we will show that the WRR scheduler exhibits the worst-case performance over $\mathbf{F}^{\underline{r}}$. Hence, the performance of the $CSD\text{-}FA_{AS}^{SAP}$ scheduler

can be enhanced by choosing SAP $\in \mathbf{F}^r$ such that SAP $\neq$ WRR. In addition, prioritized arbitration schemes proposed in Chapter 4 may result in performance enhancement over UA, which is assumed in this chapter.

Our analysis in Section 5.2.2 assumes that each flow $j \in \mathbf{C}^2$ is permitted to transmit only in slots allocated to the aggregate flow η'. As a result, the overall throughput computed using Eq. (5.7) is actually a lower bound to the actual achievable throughput, since slots allocated to $\mathbf{C}^1$ are actually available to flows $\in \mathbf{C}^2$. The corresponding buffer requirement computed represents a lower bound since the delay variation of any flow $j \in \mathbf{C}^2$ is minimized as a result of the assumption.

5.4 Summary

In this chapter, we propose a hybrid scheduler that exploits both the short- and long-term error behavior of the channel of each flow so as to achieve high overall throughput as well as low wireless receiver buffer requirements in a two-class channel-heterogeneous scheduling scenario.

The scheduler first partitions the flows according to their long-term error behavior (persistent/uncorrelated) such that flows with uncorrelated channels are fairly aggregated. The aggregated flow is then scheduled alongside the remaining flows with a CSD scheduler which exploits the short-term error behavior to maximize channel efficiency.

We compare the overall throughput as well as receiver buffer requirements of our proposed scheduler and the CSD scheduler. Our proposed scheduler achieves good overall throughput as well as low receiver buffer requirements, thus stressing the importance to exploit the long-term error behavior in addition to the instantaneous channel state in the design of wireless schedulers.

Chapter 6

Design and Performance Analysis of Loop Schedulers

Let us consider the following scheduling problem: Given the time-fraction requirement, $\underline{r}$, the goal of the scheduler is to produce an allocation sequence (or a schedule) to flows, while trying to optimize two different measures:

Fairness : A schedule is said to have good fairness if the fraction of time slots allocated to each flow is close to its time-fraction requirement.

Smoothness : A schedule is said to have good smoothness if the time slots allocated to each flow are as evenly spaced as possible.

Since $\underline{r}$ is assumed to be constant, the scheduler reduces to an *offline* one. The best possible schedule is one where the allocated shares for each flow j is exactly $\frac{r^j}{R}$ (perfectly-fair schedule) and where each flow j is scheduled exactly every ζ^j time slots (perfectly-periodic schedule). In this study, we consider the design of perfectly-fair schedulers that maximizes smoothness; the design of perfectly-periodic schedulers can be found in [44, 45, 46, 47, 48].

In [49], the authors considered an *online* variant of the scheduling problem. Given that the arrival process of packets to each flow is independent and identically distributed (i.i.d), the goal is to determine a scheduler that optimizes some performance criteria under the perfect-fairness constraint.

In [50], the author deduced that for throughput optimality for K=2 and unit buffer size per input-flow, the schedule must be *open-loop* (or de-centralized) and *conflict-free*. This work was extended in [51] to the case of K>2. It was also verified that an optimal schedule always exists and is stationary and *cyclic* (or loop), i.e., there exists an R such that for all i, the flow allocated to slot i is also allocated to slot $i+R$.

This reduces the problem to an *offline* one, where the objective is to determine a loop schedule of size R, given $\underline{r}$. The authors proposed a Golden Ratio Scheduler that achieves a nearly-optimal throughput under online conditions. In [49], the authors considered the case where the buffer size per input-flow is unlimited. It is shown that the mean queue size (or equivalently, the mean packet delay) is minimized with a perfectly-periodic schedule, which is not always feasible. Although the golden ratio scheduler is not perfectly periodic, it performs extremely well compared to lower bounds for expected packet delay.

In [52], the authors considered the design of perfectly-fair schedulers for the more general case, where $r^j \in \mathbb{R}$. The allocation smoothness is analyzed in terms of the notion of *balanceness* [53] and *regularity* [54]. It is shown that for the special case considered in our study, where $r^j \in \mathbb{Z}$, the schedulers reduce to perfectly-fair loop schedulers (denoted by $\mathbf{F}^{\underline{r}}$). The design of perfectly-fair loop schedulers is an important problem since we assumed that the SAP $\in \mathbf{F}^{\underline{r}}$ and it is the component of the generic wireless scheduler that determines its QoS performance under error-free conditions.

In this chapter, we define an alternative metric (periodicity metric) to quantify the allocation smoothness of various loop schedulers proposed prior to [52]. We show that the WRR scheduler exhibits the worst allocation smoothness in $\mathbf{F}^{\underline{r}}$, and propose a recursive loop scheduler for a class-based scheduling scenario. Since $\mathbf{F}^{\underline{r}}$ reduces to a simple RR scheduler for an input-homogeneous scenario, we focus on input-heterogeneous scenarios.

6.1 Problem Formulation

Under error-free conditions, for a scheduler π, the duration between the $(m-1)^{th}$ and m^{th} allocation to flow j corresponds to the HOL delay of packet m, denoted by $n_\pi^j(m)$. Property 2.1 can be written with respect to $n_\pi^j(m)$ as follows:

Property 6.1 *If $\pi \in \boldsymbol{F}^{\underline{r}}$, then for 1$\leq j \leq K$,*

$$\begin{aligned} n_\pi^j(r^j + m) &= n_\pi^j(m) \\ \sum_{m=k}^{k+r^j-1} n_\pi^j(m) &= R, \text{ for any } k > 0 \end{aligned}$$

Hence, each scheduler π can be uniquely characterized by the elements, $\{n_\pi^j(m)\}_{m=1}^{r^j}$ (which we denote by $\underline{n_\pi^j}$). A suitable metric to evaluate the periodicity of allocation with respect to flow j is the variance of $\underline{n_\pi^j}$, $Var[\underline{n_\pi^j}]$ = $E[\underline{n_\pi^j}]^2$ - $(E[\underline{n_\pi^j}])^2$, where

$$E[\underline{n_\pi^j}]^x = \frac{\sum_{m=1}^{r^j}[n_\pi^j(m)]^x}{r^j}$$

However, according to Property 6.1, we have the following:

$$\begin{aligned} E[\underline{n_\pi^j}] &= \frac{\sum_{m=1}^{r^j} n_\pi^j(m)}{r^j} \\ &= \frac{R}{r^j} \quad \text{independent of } \pi \end{aligned}$$

Hence, the periodicity measure for any scheduler π with respect to flow j can be evaluated in terms of $E[\underline{n_\pi^j}]^2$: a smaller value of $E[\underline{n_\pi^j}]^2$ implies a more periodic slot allocation to flow j and vice versa. We note that the order of the elements in $\underline{n_\pi^j}$ is unimportant for the evaluation of $E[\underline{n_\pi^j}]^2$, and hence, we can consider $\underline{n_\pi^j}$ as a set of r^j elements.

The allocation *sequence* $\underline{a}^\pi$ corresponding to scheduler $\pi \in \mathbf{F}^{\underline{r}}$ can be written in terms of the allocation *vector*, $\underline{\acute{a}}_\pi$, of length R as follows:

$$\underline{a}^\pi = [\cdots, \underline{\acute{a}}_\pi, \underline{\acute{a}}_\pi, \underline{\acute{a}}_\pi, \cdots]$$

Hence, our scheduling problem can be formulated as follows:

K-flow Loop Scheduling Problem

Determine the allocation vector $\underline{\acute{a}}_{\pi^*}$ such that

$$\pi^* = \arg \min_{\pi \in \mathbf{F}^{\underline{r}}} \sum_{j=1}^{K} w^j \cdot E[\underline{n}_\pi^j]^2$$

for some weighting function, $\underline{w}=[w^1,\cdots,w^K]$

If $\mathbf{A} = \{\underline{\acute{a}}_\pi : \pi \in \mathbf{F}^{\underline{r}}\}$, then

$$|\mathbf{A}| \;=\; \frac{R!}{\prod_{j=1}^{K} r^j!}$$

We note that a large number of $\underline{\acute{a}}_\pi \in \mathbf{A}$ are equivalent since they are identical under rotation with respect to $E[\underline{n}_\pi^j]^2$. However, even after eliminating these, the resultant space is still non-tractable for large R.

A dynamic programming approach to derive an optimal scheduler requires the definition of an additive objective function, i.e., one which is computed incrementally. However, the periodicity metric is a cumulative quantity, which renders the approach unsuitable. Therefore, our approach is to evaluate the periodicity performance of various known loop schedulers. By comparing against a lower bound (which we shall derive), we can determine the 'best' scheduler and also quantify its deviation from the optimal scheduler.

6.2 Description of K-flow Loop Schedulers

In this section, we describe the mechanism as well as the periodicity characteristics of several loop schedulers. Without loss of generality, we will assume that $r^j \leq r^k$ for $j<k$ and $r^j \geq 2$. The case of $r^j=1$ is trivial since $\underline{n}_\pi^j = R$ for $\pi \in \mathbf{F}^{\underline{r}}$, i.e., perfect-periodicity is always achieved for flow j. We denote by $ns_\pi^{(j,k)}(m)$ the cumulative number of slots allocated to flow k up to the m^{th} allocation to flow j by scheduler π, and $\acute{a}_\pi(m)$ denotes the m^{th} element of vector $\underline{\acute{a}}_\pi$.

6.2.1 *K*-flow Deficit Round Robin Scheduler (DRR_K)

Fair-queueing schedulers (e.g., WFQ) achieve nearly-perfect fairness, but they are usually expensive to implement. DRR_K [55] is an online scheduler that is an approximation to fair-queueing which is simple to implement and yet achieves good fairness and can also be implemented as a loop scheduler. Within the scope of our scheduling problem, the DRR_K scheduler reduces to the WRR policy, which simply allocates a block of r^1 slots to flow 1 followed by a block of r^2 slots to flow 2 and so on. Hence, each flow j is allocated slots in blocks of size r^j, with an interval of R-r^j slots between successive blocks. Therefore, we have the following:

$$\begin{aligned} \underline{n}^j_{DRR_K} &= \{1, \cdots, 1, R - r^j + 1\} \\ E[\underline{n}^j_{DRR_K}]^2 &= \frac{r^j + (R - r^j)^2 + 2(R - r^j)}{r^j} \end{aligned} \tag{6.1}$$

The periodicity performance of the DRR_K is given by the following theorem:

Theorem 6.1 *The DRR_K scheduler exhibits the worst periodicity amongst $\pi \in \boldsymbol{F}^{\underline{r}}$, i.e., for 1$\leq j \leq K$,*

$$E[\underline{n}^j_{DRR_K}]^2 = \max_{\pi \in \boldsymbol{F}^{\underline{r}}} E[\underline{n}^j_{\pi}]^2$$

Proof. Let us consider an arbitrary scheduler $\pi \in \mathbf{F}^{\underline{r}}$ with $\underline{n}^j_{\pi}$ given as follows:

$$\underline{n}^j_{\pi} = \{1 + z_1, 1 + z_2, \cdots, 1 + z_{r^j - 1}, R - r^j + 1 - \sum_{x=1}^{r^j - 1} z_x\}$$

where $z_x \in \mathbf{Z}^+$, 1$\leq x \leq r^j$-1. We note that for z_x=0, 1$\leq x \leq r^j$-1, $\pi = DRR_K$.

Using Eq. (6.1), $E[\underline{n}^j_{\pi}]^2$ can be expressed in terms of $E[\underline{n}^j_{DRR_K}]^2$ as follows:

$$E[\underline{n}^j_{\pi}]^2 = E[\underline{n}^j_{DRR_K}]^2 + \frac{\sum_{x=1}^{r^j - 1} z_x^2 + [\sum_{x=1}^{r^j - 1} z_x]^2 - 2(R - r^j) \sum_{x=1}^{r^j - 1} z_x}{r^j} \tag{6.2}$$

Since $\underline{n}_{\pi}^{j}$ corresponds to inter-allocation intervals, we have the following constraint:

$$R - r^j + 1 - \sum_{x=1}^{r^j-1} z_x \;\; \geq \;\; 1$$

In addition, according to the triangular inequality, we have:

$$\sum_{x=1}^{r^j-1} z_x^2 \;\; \leq \;\; [\sum_{x=1}^{r^j-1} z_x]^2$$

Substituting into Eq. (6.2), we have the following:

$$\begin{aligned} E[\underline{n}_{\pi}^{j}]^2 &\leq E[\underline{n}_{DRR_K}^{j}]^2 + \frac{\sum_{x=1}^{r^j-1} z_x^2 + [\sum_{x=1}^{r^j-1} z_x]^2 - 2[\sum_{x=1}^{r^j-1} z_x]^2}{r^j} \\ &= E[\underline{n}_{DRR_K}^{j}]^2 + \frac{\sum_{x=1}^{r^j-1} z_x^2 - [\sum_{x=1}^{r^j-1} z_x]^2}{r^j} \\ &\leq E[\underline{n}_{DRR_K}^{j}]^2 \end{aligned}$$

□

6.2.2 *K*-flow Weighted Round Robin with WFQ-like spreading Scheduler (*WRR-*sp_K)

The *WRR-*sp_K scheduler [56] is a variant of the standard WRR scheduler, in which the service order amongst the flows is identical to WFQ. The algorithm for the *WRR-*sp_K scheduler is described as follows:

K-flow WRR with WFQ-like spreading Scheduler ($WRR - sp_K$)

Let the array $\underline{x}$ contain the sequence $< \frac{m}{r^j}, j >: m \in \{1, \cdots, r^j\}$, $1 \leq j \leq K$ sorted in lexicographic order.
The vector $\underline{\acute{a}}_{WRR-sp_K}$ is constructed as follows:
$\acute{a}_{WRR-sp_K}(i) = j$ if $x(i) =< \frac{m}{r^j}, j >$

The *WRR-*sp_K scheduler possesses the following property for $1 \leq j \leq K$-1:

Property 6.2 *The m^{th} allocation of flow j always occurs between the $\lceil \frac{mr^k}{r^j} \rceil$ th and $\lceil \frac{mr^k}{r^j} \rceil$ - 1 th allocation of flow k, where $k>j$, $1 \leq m \leq r^j$, i.e.,*

$$ns^{(j,k)}_{WRR-sp_K}(m) \;=\; \lceil \frac{mr^k}{r^j} \rceil - 1$$

Proof. According to the algorithm, the m^{th} allocation to flow j is characterized by the parameter $\frac{m}{r^j}$. If x denotes the cumulative number of slots allocated to flow k up to the m^{th} allocation of flow j and $j < k$, then x must satisfy the following conditions:

$$\frac{x}{r^k} \;<\; \frac{m}{r^j} \text{ and}$$
$$\frac{x+1}{r^k} \;\geq\; \frac{m}{r^j}$$

Hence, we obtain $ns^{(j,k)}_{WRR-sp_K}(m) = x = \lceil \frac{mr^k}{r^j} \rceil$ -1 □

6.2.3 *K*-flow Credit Round Robin Scheduler (CRR_K)

The motivation to design the CRR_K scheduler [57] was to reduce the latency of the DRR_K scheduler. As with the DRR_K scheduler, the CRR_K scheduler can be implemented as a loop scheduler, and the pseudo-code is given as follows:

_K_-flow Credit Round Robin Scheduler (CRR_K)

Initialize $x^j = \frac{r^j}{r^K}$, $1 \leq j \leq K$
Set i=1, SP=K, $count$=0
while $i \leq R$
    if $count < K$
      if $x^{SP} < 1$ $SP = SP$ - 1, $count = count + 1$
      else $\acute{a}_{CRR_K}(i) = SP$, $x^{SP} = x^{SP}$ - 1
          $SP = SP$ - 1, $i = i + 1$, $count = 0$
    else $x^j = x^j + \frac{r^j}{r^K}$ $\forall$ j, $count = 0$

The CRR_K scheduler possesses the following property for $1 \leq j \leq K$-1:

Property 6.3 *The m^{th} allocation of flow j always occurs between the $\lceil \frac{mr^K}{r^j} \rceil$ th and $\lceil \frac{mr^K}{r^j} \rceil$ - 1 th allocation of flow K, $1 \leq m \leq r^j$, i.e.,*

$$ns^{(j,K)}_{CRR_K}(m) \;=\; \lceil \frac{mr^K}{r^j} \rceil - 1$$

Proof. With the CRR_K scheduler, the first slot is always allocated to flow K. We can consider subsequent allocations in blocks, where each block terminates with the next flow K allocation, as illustrated in Fig. 6.1, where $ns^{(j,K)}_{CRR_K}(m)$ is the number of slots allocated to flow K up to the m^{th} allocation to flow j.

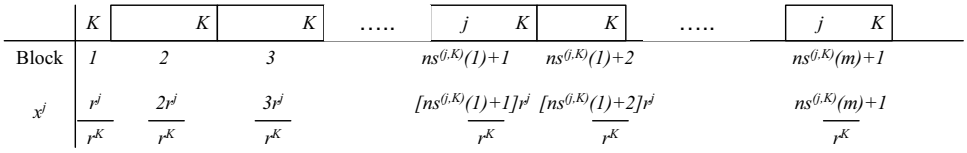

Figure 6.1: Illustration of allocation to flow K relative to allocation to flow j with the CRR_K scheduler.

According to the transmission heuristics given by the pseudo-code, $ns^{(j,K)}_{CRR_K}(m)$ has to satisfy the following conditions:

$$\frac{[ns^{(j,K)}_{CRR_K}(m)+1]r^j}{r^K} \;\geq\; m \;\text{ and}$$

$$\frac{ns^{(j,K)}_{CRR_K}(m)\cdot r^j}{r^K} \;<\; m$$

Hence, we obtain $ns^{(j,K)}_{CRR_n}(m) = \lceil \frac{mr^K}{r^j} \rceil$ -1 □

6.2.4 *K*-flow Golden Ratio (GR_K) Scheduler

The Golden Ratio Scheduler was proposed in [51] and is described as follows:

K-flow Golden Ratio Scheduler (GR_K)

Let $z = 0.6180339887$ and $x(m) = frac(m \cdot z)$ where $frac(y) = y - \lfloor y \rfloor$
Let the array $\underline{y}$ contain the sequence $x(m)$, $0 \leq m \leq R$-1,
sorted in ascending order.
The vector $\underline{\acute{a}}_{GR_K}$ is constructed as follows:
$\acute{a}_{GR_K}(i) = j$ if $\sum_{k=1}^{j-1} \frac{r^k}{R} \leq y(i) \leq \sum_{k=1}^{j} \frac{r^k}{R}$, $1 \leq j \leq K$

It was established in [49] that if R is a Fibonacci number, then $\underline{n}^j_{GR_K}$ comprises at most three values for each j; otherwise, more values are generated.

6.2.5 K-flow Short-term Fair Scheduler (STF_K)

We can characterize the *fairness* performance of any loop scheduler in terms of the *cumulative service-deficit*, $sd^j(i)$, which measures the discrepancy between the requested and allocated fractional bandwidth for flow j up to slot i, $1 \leq i \leq R$. If $y^j(i)$ denote the cumulative number of slots allocated to flow j up to and including slot i, then we have the following:

$$sd^j(i) \;=\; \frac{r^j}{R} - \frac{y^j(i)}{i}$$

A *positive* (negative) value of $sd^j(i)$ implies that flow j has received *less* (more) than its fair share of bandwidth up to slot i. Hence, we consider a scheduler that allocates each slot to the flow with maximum instantaneous service-deficit so as to achieve maximum fairness (Short-term Fair or STF_K scheduler). Whenever there is a tie, priority for allocation is given to the flow with the highest flow index. The pseudo-code for the STF_K scheduler is given as follows:

K-flow Short-term Fair Scheduler (STF_K)

Initialize $y^j(0) = 0$, $1 \leq j \leq K$
for i=1:R
$y^j(i) = y^j(i-1)$, $1 \leq j \leq K$
$sd^j(i) = \frac{r^j}{R}$ - $\frac{y^j(i)}{i}$, $1 \leq j \leq K$
$\acute{a}_{STF_K}(i) = \arg \max_{1 \leq j \leq K} sd^j(i)$
$y^{\acute{a}_{STF_K}(i)}(i) = y^{\acute{a}_{STF_K}(i)}(i)+1$

This scheduler was first suggested in [51], where the authors conjectured, based on numerical calculations, that it is a promising scheduler. However, no analysis of the scheduler was provided in terms of periodicity properties.

6.2.6 K-flow Random (RND_K) Scheduler

The loop schedulers considered so far are *deterministic* since the allocation vector $\underline{\acute{a}}_\pi \in \mathbf{A}$ is fixed. In this section, we define a *random* scheduler, RND_K, whose allocation vector, $\underline{\acute{a}}_{RND_K}$, is *uniformly* selected from $\mathbf{A}$. We note that $RND_K \in \mathbf{F}^{\underline{r}}$ because the selected $\underline{\acute{a}}_{RND_K}$ is used for allocation in each loop.

Let us refer to an allocation sequence based on the RND_K scheduler, and consider a particular loop that begins with the $(m-1)^{th}$ allocation to flow j, as illustrated in Fig. 6.2. Since r^j slots must be allocated to flow j in any loop, the total number of ways the $\{k^{th}\}_{k=m}^{m+r^j-2}$ allocations to flow j can occur within R-1 slots is $\binom{R-1}{r^j-1}$. However, the corresponding expression that ensures that $n^j_{RND_K}(m) = N$ is given by $\binom{R-N-1}{r^j-2}$. Hence,

$$\text{Prob}(n^j_{RND_K}(m) = N) \;=\; \frac{\binom{R-N-1}{r^j-2}}{\binom{R-1}{r^j-1}}, \quad 1 \leq N \leq R - r^j + 1$$

The periodicity metric for each flow j is evaluated as follows:

$$E[\underline{n}^j_{RND_K}]^2 = \sum_{N=1}^{R-r^j+1} \text{Prob}(n^j_{RND_K}(m) = N) \cdot N^2 \quad (6.3)$$

$$= \frac{R(2R - r^j + 1)}{r^j(r^j + 1)}$$

By comparing Eq. (6.3) with Eq. (6.1), we have the following theorem:

Theorem 6.2 *The periodicity metric of the RND_K scheduler is upper-bounded by that of the DRR_K scheduler, i.e., for $1 \leq j \leq K$,*

$$E[\underline{n}^j_{RND_K}]^2 \leq E[\underline{n}^j_{DRR_K}]^2$$

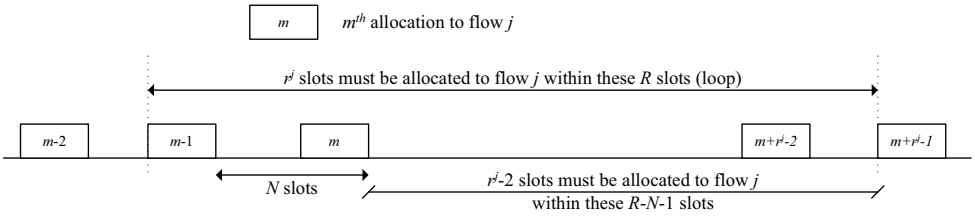

Figure 6.2: Evaluation of Prob($n^j_{RND_K}(m)$=N) for the random scheduler.

6.3 Conditions for Optimal Per-flow Periodicity for *K*-flow Loop Schedulers

In this section, we determine the conditions for optimal allocation periodicity for flow j. This can be expressed as a requirement on $\underline{n}^j_\pi$ in the following theorem:

Theorem 6.3 *If $\underline{n}^j_*$ is defined as follows:*

$$\underline{n}^j_* = \{\overbrace{\lfloor\frac{R}{r^j}\rfloor, \cdots, \lfloor\frac{R}{r^j}\rfloor}^{r^j\lfloor\frac{R}{r^j}\rfloor + r^j - R}, \underbrace{\lceil\frac{R}{r^j}\rceil, \cdots, \lceil\frac{R}{r^j}\rceil}_{R - r^j\lfloor\frac{R}{r^j}\rfloor}\}$$

then we have the following:

$$E[\underline{n}_*^j]^2 \;=\; \min_{\pi \in \boldsymbol{F}^{\underline{r}}} E[\underline{n}_\pi^j]^2$$

Proof. We consider the following cases:

$\boldsymbol{R \equiv 0}$ (modulo r^j) : Perfect allocation periodicity is achieved for flow j when the inter-allocation interval is constant, i.e., $n_*^j(k) = n_*^j(m)$. This is achieved if and only if $n_*^j(k) = \frac{R}{r^j}$ for $1 \leq k \leq r^j$.

$\boldsymbol{R \equiv x}$ (modulo r^j), $\boldsymbol{1 \leq x \leq r^j - 1}$: In this case, a constant inter-allocation interval for flow j cannot be achieved. The best one can achieve is the following for $1 \leq k \leq r^j$:

$$n_*^j(k) \;\in\; \{N, N+1\}, \text{ where } 1 \leq N \leq R - r^j.$$

For $1 \leq m \leq r^j - 1$, let us assume the following:

$$\underline{n}_*^j \;=\; \{\overbrace{N, \cdots, N}^{m}, \underbrace{N+1, \cdots, N+1}_{r^j - m}\}$$

Then, since $\sum_{k=1}^{r^j} n^j(k) = R$, we have the following:

$$m \cdot N + (r^j - m) \cdot (N+1) \;=\; R$$

from which we have

$$m \;=\; r^j \cdot N + r^j - R$$

However, since $1 \leq m \leq r^j - 1$, we have the following constraints on N:

$$\frac{R}{r^j} - 1 + \frac{1}{r^j} \leq N \leq \frac{R}{r^j} - \frac{1}{r^j} \tag{6.4}$$

Since $\lfloor \frac{R}{r^j} \rfloor - 1 < \frac{R}{r^j} - 1 + \frac{1}{r^j}$ and $\lceil \frac{R}{r^j} \rceil > \frac{R}{r^j} - \frac{1}{r^j}$, the only integer N that can satisfy Eq. (6.4) is $N = \lfloor \frac{R}{r^j} \rfloor$.

□

The corresponding value for $E[\underline{n}_*^j]^2$ can be used as a *lower* bound for all $\pi \in \mathbf{F}^{\underline{r}}$ and is given as follows:

$$E[n_*^j]^2 = \frac{R(2N+1) - r^j N(N+1)}{r^j}$$

6.4 A Recursive Loop Scheduler for Input-Heterogeneous Class-based Scheduling

In this section, we consider a C-class input-heterogeneous scheduling scenario, and propose a recursive loop scheduler based on the $WRR - sp_K$ scheduler that exhibits good periodicity properties.

6.4.1 Intra-class Fairness in Class-based Scheduling

In addition to optimizing the periodicity of individual flows, a desirable characteristic in class-based scheduling is the notion of intra-class fairness, i.e., all flows from the same class should possess the same periodicity characteristics. Hence, a scheduler π ensures intra-class fairness if $E[\underline{n}_\pi^j]^2 = E[\underline{n}_\pi^k]^2$ for all j,k $\in \mathbf{C}^x$ for $1 \leq x \leq C$.

A simple example of a scheduler that ensures intra-class fairness is the DRR_K scheduler. This can be observed from Eq. (6.1), where, for any flow $j \in \mathbf{C}^x$:

$$\underline{n}_{DRR_K}^j = \{1, \cdots, 1, R - \tilde{r}^x + 1\} \tag{6.5}$$

On the other hand, the RND_K scheduler does not ensure intra-class fairness. As an example, we consider a two-class input-heterogeneous scheduling scenario, where $\underline{\tilde{r}}$=[2,3] and $\underline{\kappa}$=[2,2]. If $\underline{\acute{a}}_{RND_K}$=[3,1,2,1,3,2,4,4,3,4], then by evaluating $\underline{n}_{RND_K}^j$, we observe that although flows 1, 2 $\in \mathbf{C}^1$, they have different periodicity characteristics.

6.4.2 Periodicity property of $WRR-sp_K$ scheduler for C-class Scheduling

Property 6.2 can be generalized for a class-based scenario as follows:

Property 6.4 *For the WRR-sp_K scheduler, flows within each class are allocated in blocks, where the order within class $\boldsymbol{C}^x$ is $\sum_{m=1}^{x-1}\kappa^m+1$, $\sum_{m=1}^{x-1}\kappa^m+2$, $\cdots$, $\sum_{m=1}^{x}\kappa^m$ for $1\leq x \leq C$. In addition, the m^{th} block of $\boldsymbol{C}^x$ will reside between the $\lceil\frac{m\cdot\tilde{r}^y}{\tilde{r}^x}\rceil^{th}$ and $\lceil\frac{m\cdot\tilde{r}^y}{\tilde{r}^x}\rceil-1^{th}$ block of $\boldsymbol{C}^y$, where $y>x$ and $1\leq m \leq \tilde{r}^x$.*

We note from Property 6.4 that flows within each class are always transmitted in blocks, where each flow from that class is allocated exactly once and the order within each block is constant. Hence, the periodicity characteristics for flows belonging to the same class are identical, i.e., intra-class fairness is maintained.

Special Case: C=2 : Using Property 6.4, $\underline{n}^j_{WRR-sp_K}$ can be evaluated as follows:

$$\underline{n}^j_{WRR-sp_K} = \begin{cases} \{\overbrace{\kappa^1+\kappa^2\lfloor\frac{\tilde{r}^2}{\tilde{r}^1}\rfloor,\cdots,\kappa^1+\kappa^2\lfloor\frac{\tilde{r}^2}{\tilde{r}^1}\rfloor}^{\tilde{r}^1\lceil\frac{\tilde{r}^2}{\tilde{r}^1}\rceil-\tilde{r}^2}, \\ \underbrace{\kappa^1+\kappa^2\lceil\frac{\tilde{r}^2}{\tilde{r}^1}\rceil\cdots\kappa^1+\kappa^2\lceil\frac{\tilde{r}^2}{\tilde{r}^1}\rceil}_{\tilde{r}^2-(\lceil\frac{\tilde{r}^2}{\tilde{r}^1}\rceil-1)\tilde{r}^1}\}, & j\in\mathbf{C}^1; \\ \{\overbrace{\kappa^2,\cdots,\kappa^2}^{\frac{\tilde{r}^2K-R}{\kappa^1}},\underbrace{K,\cdots,K}_{\frac{R-\kappa^2\tilde{r}^2}{\kappa^1}}\}, & j\in\mathbf{C}^2. \end{cases} \quad (6.6)$$

Similarly, from Theorem 6.3, we have the following:

$$\underline{n}^j_* = \begin{cases} \overbrace{\{\kappa^1 + \lfloor \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1} \rfloor, \cdots, \kappa^1 + \lfloor \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1} \rfloor,}^{\tilde{r}^1(\kappa^1+\lfloor \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1}\rfloor)+\tilde{r}^1-R} \\ \underbrace{\kappa^1 + \lceil \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1} \rceil, \cdots, \kappa^1 + \lceil \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1} \rceil\}}_{R-\tilde{r}^1(\kappa^1+\lfloor \frac{\kappa^2\tilde{r}^2}{\tilde{r}^1}\rfloor)}, \quad j \in \mathbf{C}^1; \\ \overbrace{\{\kappa^2 + \lfloor \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2} \rfloor, \cdots, \kappa^2 + \lfloor \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2} \rfloor,}^{\tilde{r}^2(\kappa^2+\lfloor \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2}\rfloor)+\tilde{r}^2-R} \\ \underbrace{\kappa^2 + \lceil \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2} \rceil, \cdots, \kappa^2 + \lceil \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2} \rceil\}}_{R-\tilde{r}^2(\kappa^2+\lfloor \frac{\kappa^1\tilde{r}^1}{\tilde{r}^2}\rfloor)}, \quad j \in \mathbf{C}^2. \end{cases} \tag{6.7}$$

Comparing Eq. (6.6) with Eq. (6.7), we note that $\underline{n}^j_{WRR-sp_K} \neq \underline{n}^j_*$ for $1 \leq j \leq K$ and hence, the *WRR-sp*$_K$ scheduler is not optimal in terms of per-flow periodicity. However, we note that when κ^1=1 (κ^1=K-1), the *WRR-sp*$_K$ scheduler offers optimal periodicity for flows in $\mathbf{C}^2$ ($\mathbf{C}^1$). However, if $\tilde{r}^1$=1, then optimal (worst-case) periodicity is achieved for flows in $\mathbf{C}^1$ ($\mathbf{C}^2$).

The corresponding periodicity properties for the STF_K and RND_K schedulers can be found in [58].

Enhancement to WRR-*sp*$_K$ Scheduler : For $C = 2$, we observe that the *WRR-sp*$_K$ scheduler results in worst-case periodicity for $\mathbf{C}^2$ flows when κ^1=K-1 and $\tilde{r}^1$=1. This is due to the default lexicographic ordering in the scheduling mechanism, which can be circumvented by introducing a parameter, ϱ, $1 \leq \varrho \leq K$, to the *WRR-sp*$_K$ scheduler (denoted *WRR-sp*$_K(\varrho)$). With the *WRR-sp*$_K(\varrho)$ scheduler, the ordering priority in the event of a tie in the elements $\{\frac{m}{r^j}\}_{m=1}^{r^j}$ for $1 \leq j \leq K$ is given by [ϱ,ϱ+1,ϱ+2,$\cdots$,K,1,2,$\cdots$,ϱ-1]. We note that the scheduler reduces to the original *WRR-sp*$_K$ scheduler when $\varrho = 1$.

6.4.3 A Recursive Approach to Class-based Scheduling

Instead of 'blindly' applying any loop scheduler to a class-based scenario, we consider a class-based scheduling paradigm that allocates slots to flows within each class *independently* of other classes (*intra*-class scheduling) and then *combines* the allocation vectors obtained in an optimal way (*inter*-class scheduling). We propose a recursive approach, where $REC^x(\underline{I})$ is a $|\underline{I}|$-class recursive scheduler with input $x \in \underline{I}$, such that $\underline{I} \subset \underline{C} = \{1,2, \cdots, C\}$. The mechanism of $REC^x(\underline{I})$ comprises two stages: The first stage involves obtaining the allocation vectors for flows $\in \{\mathbf{C}^y\}_{\forall y \in \underline{I} \backslash x}$* and $\mathbf{C}^x$ respectively. The second stage combines these allocation vectors to obtain $\underline{\acute{a}}_{REC^x(\underline{I})}$.

The allocation vector for flows $\in \{\mathbf{C}^y\}_{\forall y \in \underline{I} \backslash x}$ is obtained by evaluating the vector $\underline{\acute{a}}_{REC^z(\underline{I} \backslash x)}$ for some $z \in \underline{I} \backslash x$. Since the flows $\in \mathbf{C}^x$ are homogeneous, a simple RR allocation is optimal in terms of periodicity, and the allocation vector is given as follows:

$$\underline{\acute{a}}_{RR_{\kappa^x}} \quad = \quad [1, 2, \cdots, \kappa^x, 1, 2, \cdots, \kappa^x, \cdots, \cdots, 1, 2, \cdots, \kappa^x]$$

We note that the elements of $\underline{\acute{a}}_{REC^z(\underline{I} \backslash x)}$ and $\underline{\acute{a}}_{RR_{\kappa^x}}$ have to be updated accordingly to ensure that the correct flow indices are assigned to flows in each class. Our approach in the second stage is to insert the elements of $\underline{\acute{a}}_{RR_{\kappa^x}}$ into $\underline{\acute{a}}_{REC^z(\underline{I} \backslash x)}$ such that successive elements of $\underline{\acute{a}}_{RR_{\kappa^x}}$ are as uniformly-spaced as possible in $\underline{\acute{a}}_{REC^x(\underline{I})}$. This is illustrated in Fig. 6.3, where $M_x = \lceil \frac{|\underline{\acute{a}}_{REC^z(\underline{I} \backslash x)}|}{|\underline{\acute{a}}_{RR_{\kappa^x}}|} \rceil$.

For each $\underline{I}$, there are $|\underline{I}|$ instances of recursive schedulers, $REC^x(\underline{I})$, corresponding to each $x \in \underline{I}$. Hence, for optimality, we have to evaluate the allocation vector for each instance, and compute the corresponding periodicity performance. According to our algorithm, each $REC^x(\underline{I})$ in turn comprises $|\underline{I}|$-1 instances of recursive schedulers, $REC^z(\underline{I} \backslash x)$, corresponding to each $z \in \underline{I} \backslash x$. This continues until we are reduced to a two-class scheduling scenario.

*The notation $\underline{x} \backslash\ y$ refers to the elements of $\underline{x}$ excluding y.

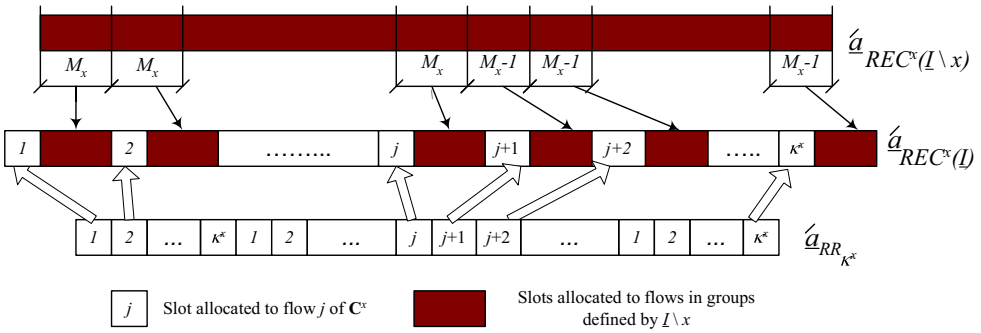

Figure 6.3: Illustration of the scheduling mechanism of the REC_C^x scheduler.

6.4.4 Optimal Two-Class *WRR*-sp_K-based Scheduler

From Section 6.4.2, we observe that for the special case of K=2, since κ^1=1 =K-1, the *WRR*-sp_2 scheduler offers optimal periodicity for *all* flows. Hence, if we define a two-class scheduler, OPT_2, that employs the $WRR - sp_2$ as an inter-class scheduler, then it can be shown that $\underline{n}^j_{OPT_2} = \underline{n}^j_*$ as given in Eq. (6.7) for $1 \leq j \leq K$. Hence, the OPT_2 scheduler is optimal for two-class scheduling, and the pseudo-code is given below, assuming $\kappa^1 \cdot \tilde{r}^1 \leq \kappa^2 \cdot \tilde{r}^2$ (the corresponding scheduler for $\kappa^1 \cdot \tilde{r}^1 > \kappa^2 \cdot \tilde{r}^2$ can be obtained by inter-changing the indices 1 and 2):

Optimal Two-Class Scheduler (OPT_2)

Set $\underline{r} = [\kappa^1 \cdot \tilde{r}^1,\ \kappa^2 \cdot \tilde{r}^2]$, $K = \kappa^1 + \kappa^2$

Define $\mathbf{C}^1 = [\overbrace{1,2,\cdots,\kappa^1}^{1}, \overbrace{1,2,\cdots,\kappa^1}^{2}, \cdots, \cdots, \overbrace{1,2,\cdots,\kappa^1}^{\kappa^1 \cdot \tilde{r}^1}]$

Define $\mathbf{C}^2 = [\underbrace{\kappa^1+1, \kappa^1+2, \cdots, K}_{1}, \cdots, \cdots, \underbrace{\kappa^1+1, \kappa^1+2, \cdots, K}_{\kappa^2 \cdot \tilde{r}^2}]$

Compute $\underline{\acute{a}}_{WRR-sp_2} = WRR\text{-}sp_2(\underline{r})$

for x = 1:2

index = find($\underline{\acute{a}}_{WRR-sp_2}$==$x$)

$\acute{a}_{OPT_2}$(index) = $\mathbf{C}^x$

Based on OPT_2, we can use our recursive approach to find an approximately-optimal allocation for class-based scheduling. Our approach is tractable since the number of classes, C, is small.

6.5 Numerical Results

In this section, we evaluate the (a) periodicity performance under error-free conditions and (b) wireless receiver buffer requirement under error-prone conditions for a CSD_{AS}^{SAP} scheduler for various SAP $\in \mathbf{F}^r$.

We define the weighted covariance of $\{\underline{n}_{SAP}^j\}_{j=1}^K$, $wcov_{SAP}$, as follows:

$$wcov_{SAP} = \sum_{j=1}^{K} \frac{r^j}{R} \cdot \frac{E[\underline{n}_{SAP}^j]^2 - (E[\underline{n}_{SAP}^j])^2}{(E[\underline{n}_{SAP}^j])^2}$$

The metric, $wcov_{SAP}$, reflects the periodicity over the ensemble of all flows. With perfect periodicity, $wcov_{SAP}$=0 since $\underline{n}_{SAP} = E[\underline{n}_{SAP}^j]$. Hence, a value close to zero is an indication of good periodicity performance.

We consider the following broadband applications with the corresponding typical bandwidth requirements in kbps [59]: Streaming Video (Internet Quality) (128), Residential Voice (300), Video Telephony (400), Interactive Games (500) and Streaming video (Video-on-Demand Quality) (3700). We define various C-class scheduling scenarios (where each class comprises flows from a particular application) and compare $wcov_\pi$ obtained for each scenario for each scheduler. For example, if we consider Residual Voice, Video Telephony and Interactive Games, then we have $\underline{\tilde{r}}$=[300,400,500] $\equiv$ [3,4,5]. We assume that the flow composition is uniform, i.e., $\kappa^x = \kappa$ for $1 \leq x \leq C$.

We define the optimal $WRR - sp_K$ scheduler (denoted $WRR - sp_K^*$), where $WRR - sp_K^* = WRR\text{-}sp_K(\varrho^*)$ such that $\varrho^* = \arg \min_{1 \leq \varrho \leq K} wcov_{WRR-sp_K(\varrho)}$. Similarly, the optimal recursive loop scheduler, $REC^*(\underline{C}) = REC^{x^*}(\underline{C})$, where $x^* = \arg \min_{\forall x \in \underline{C}} wcov_{REC^x(\underline{C})}$.

6.5.1 Periodicity Performance of CSD scheduler under Error-free Conditions

Under error-free conditions, the performance of the CSD_{AS}^{SAP} scheduler corresponds to the performance of its SAP. We compare the periodicity performance corresponding to various SAP for different flow configurations and C=3 in Fig. 6.4. The corresponding results for C=4 and 5 are shown in Fig. 6.5. Numerical results for other flow configurations can be found in [58].

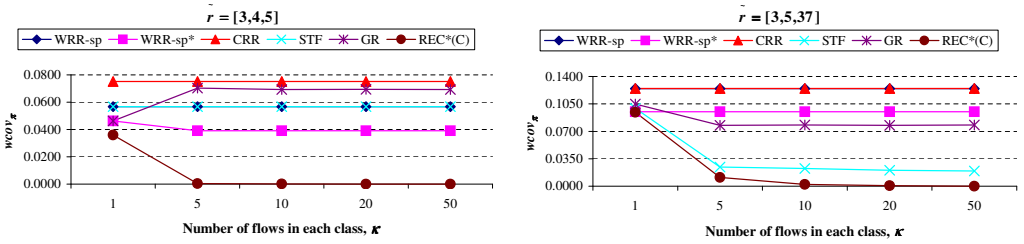

Figure 6.4: Comparison of $wcov_{SAP}$ corresponding to various SAP for $\underline{\tilde{r}}$ = [3,4,5] (left) and [3,5,37] (right) in 3-class scheduling.

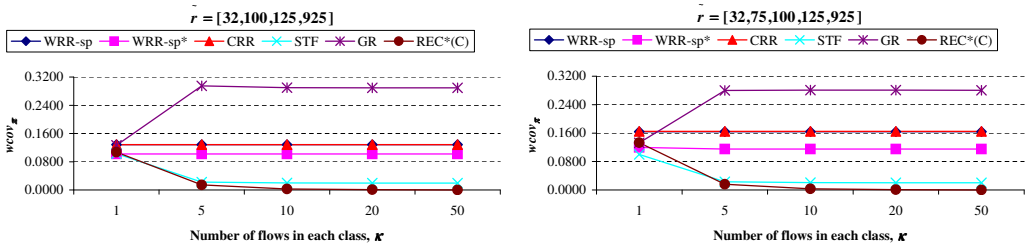

Figure 6.5: Comparison of $wcov_{SAP}$ corresponding to various SAP for $\underline{\tilde{r}}$ = [32,100,125,925] (left) and [32,75,100,125,925] (right) in 4-class and 5-class scheduling.

Although not depicted in the figures, we note that the RND_K scheduler performs significantly worse than the deterministic schedulers (excluding the DRR_K scheduler). In addition, the performance for each scheduler is relatively invariant with κ for κ >1 for a given $\underline{\tilde{r}}$. Hence, we consider the following cases:

$\kappa > 1$: Between the $WRR\text{-}sp_K$ and CRR_K schedulers, the CRR_K performs *worse*. In addition, an enhancement of the $WRR\text{-}sp_K$ always exists and the gain in terms of the weighted covariance is significant. Amongst the $REC^*(\underline{C})$, STF_K and GR_K schedulers, the relative performance is always according to the above order, with the $REC^*(\underline{C})$ scheduler achieving the *best* performance. In fact, $wcov_{STF_K}$ and $wcov_{REC^*(\underline{C})} \approx 0$ for scenarios where $\tilde{r}^C >> \tilde{r}^x$, $1 \leq x \leq C\text{-}1$.

While the $WRR\text{-}sp_K$, CRR_K and $WRR\text{-}sp^*_K$ schedulers ensure intra-class fairness for any scheduling scenario, it is not enforced by the STF_K and $REC^*(\underline{C})$ schedulers for certain scenarios, and is never enforced by the GR_K scheduler for any scenario. Hence, there is a trade-off between achieving good periodicity performance and ensuring intra-class fairness. If the latter needs to be guaranteed for any class-scheduling scenario, then the $WRR\text{-}sp^*_K$ scheduler should be used; otherwise, the $REC^*(\underline{C})$ scheduler should be used.

$\kappa = 1$: For an easier comparison of the periodicity performance of the schedulers, we plot the results for κ=1 in Fig. 6.6. The STF_K and $WRR\text{-}sp^*_K$ schedulers offer the best overall periodicity performance, while the GR_K and CRR_K schedulers offer the worst performance. We note that intra-class fairness is irrelevant in this case.

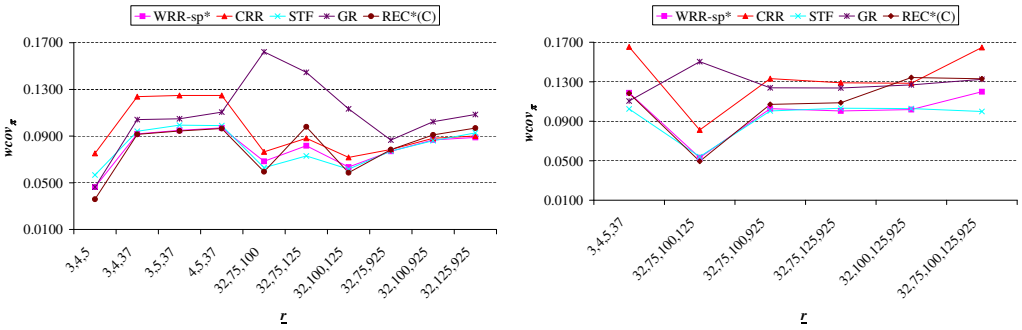

Figure 6.6: Comparison of $wcov_{SAP}$ for various SAP in K-flow scheduling where K=3 (left) and K=4,5 (right).

6.5.2 Wireless Receiver Buffer Requirement of CSD Scheduler under Error-prone Conditions

In Chapters 4 and 5, the wireless receiver buffer requirement for the CSD_{AS}^{SAP} scheduler was evaluated for input-homogeneous scheduling scenarios. In this section, we perform the corresponding evaluation for the CSD_{UA}^{SAP} scheduler in an input-heterogeneous scenario, assuming a deterministic OSP is used for channel prediction and $\gamma = 0$.

Since $b^j \neq b^k$ for $j \neq k$, we evaluate the average buffer requirement of scheduler CSD_{UA}^{SAP}, $\check{b}_{CSD_{UA}^{SAP}}$, defined as follows:

$$\check{b}_{CSD_{UA}^{SAP}} = \frac{1}{K}\sum_{j=1}^{K} b^j_{CSD_{UA}^{SAP}}$$

We plot $\check{b}_{CSD_{UA}^{SAP}}$ for various SAP as a function of g for $\underline{\tilde{r}}$=[3,4,5], $p_c(0)$=0.9, $\beta = 0.01$ and $\rho = 0.99$ in Fig. 6.7. The corresponding plots (excluding the DRR_K scheduler) for $p_c(0)$=0.8 are shown in Fig. 6.8. The error-free case is also included to investigate the effects of channel errors on $\check{b}_{CSD_{UA}^{SAP}}$.

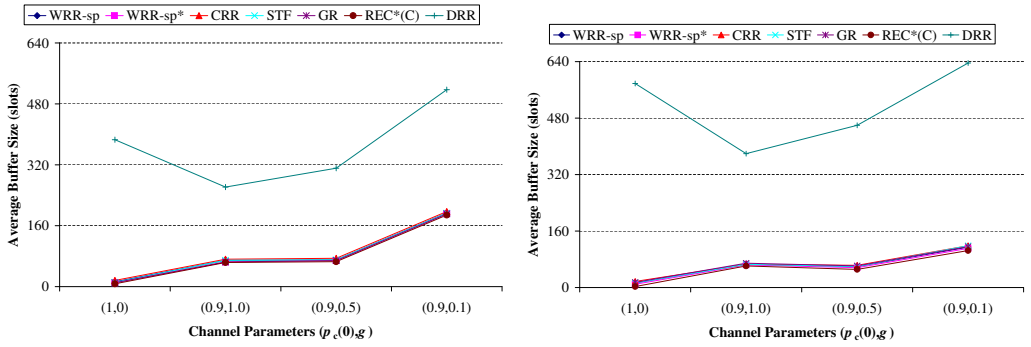

Figure 6.7: Comparison of wireless buffer requirement with the CSD_{UA}^{SAP} scheduler for various SAP for $p_c(0)$=0.9, $\underline{\tilde{r}}$ = [3,4,5] and $\kappa = 1$ (left) and $\kappa = 2$ (right).

For any $(p_c(0),g)$ and κ, the DRR_K scheduler, which is equivalent to a simple WRR_K scheduler, results in significantly larger buffer requirements than the other loop schedulers (whose performances are similar). Hence, the perfor-

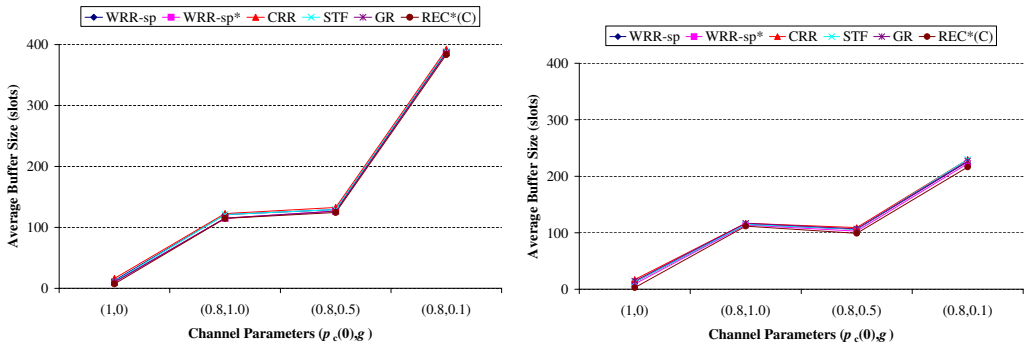

Figure 6.8: Comparison of wireless buffer requirement with the CSD_{UA}^{SAP} scheduler for various SAP for $p_c(0)$=0.8, $\underline{\tilde{r}}$ = [3,4,5] and $\kappa = 1$ (left) and $\kappa = 2$ (right).

mance of the CSD_{UA}^{WRR} scheduler can be improved with any other choice for the SAP. In addition, the following observations are common for all the loop schedulers (excluding the DRR_K scheduler):

Variation with channel quality : The buffer requirement is increased when the channel quality is degraded.

Variation with channel agility : The buffer requirement is relatively invariant for $g \geq 0.5$, but it is significantly increased when g is reduced from 0.5 to 0.1.

Variation with number of flows : When the channel is uncorrelated, the buffer requirement is relatively invariant with κ; however, for a persistent channel, the buffer requirement is significantly reduced (up to 100%) when κ is increased from 1 to 2.

6.6 Summary

In this chapter, we consider the design of a perfectly-fair loop scheduler that minimizes the HOL packet delay variance (or maximizes the periodicity) for an input-heterogeneous scenario under error-free conditions. We describe and analyze the periodicity properties of various loop schedulers and show

that the WRR scheduler exhibits the worst periodicity performance. We also derive the conditions for optimal per-flow periodicity for any K-flow loop scheduler.

We consider a class-based scheduling scenario where flows can be grouped according to their relative bandwidth demands. We propose a recursive implementation of the WRR-with-spreading scheduler, and verify its optimality in terms of periodicity for a two-class scenario.

Using numerical results, we compare the performance of the loop schedulers under error-free conditions. Although the recursive scheduler achieves the best periodicity performance, it fails to guarantee intra-class fairness, which is desirable for class-based scheduling. On the other hand, an enhanced WRR-with-spreading scheduler gives the best periodicity performance amongst those schedulers that maintain intra-class fairness. Hence, there is a trade-off between periodicity and fairness performance in the design of loop schedulers.

In addition, we also investigate the wireless receiver buffer requirement under error-prone conditions when various loop schedulers are implemented as the SAP in a CSD scheduler. The buffer requirement is significantly larger when the WRR scheduler is used as the SAP, which concurs with its worst periodicity performance; the corresponding requirement is similar when any other loop scheduler is used. Hence, the choice of the SAP is important in determining the QoS performance of CSD schedulers.

Chapter 7

Summary

7.1 Main Results

The importance of mobile connectivity along with the popularity of the Internet is fuelling the development and roll-out of Next-Generation wireless networks. We consider a generic wireless network (Fig. 1.1), where an Access Point (AP) is the interface that links wireless receivers (assumed to be buffer-limited) to a wired network. We envisage a demand for such a network to deliver wireless services with diverse QoS requirements from the wired network to the wireless receivers (downlink). For a given input-traffic specification, the design of the downlink wireless scheduler at each AP is critical for QoS provisioning over the wireless link as well as the receiver buffer design. The high error rate as well as time- and location-dependence of the wireless channel renders the problem a hard and challenging one.

Although several wireless schedulers have been proposed recently, their main contribution lies in the design of the scheduling mechanism to achieve a specific performance. To the best of our knowledge, work on the QoS analysis of these schedulers is limited, and hence, our objective is to characterize the QoS performance of a generic wireless scheduler in order to study the trade-offs amongst QoS metrics and wireless scheduler and receiver design for a given scenario. We define the following tasks to achieve this goal:

Definition and Modelling of Scheduling Problem (See Chapter 2) :

The wireless scheduling problem comprises the input-flows, the wireless channel, the wireless scheduler, the wireless receivers, and the desirable QoS metrics. Each input-flow is characterized by its time-fraction requirement, which specifies the fraction of resources that should be allocated to it. We consider a Two-State Markov Chain (2SMC) model for the wireless channel since its typical characteristics can be manifested in such a model. We assume a uniform channel quality for each flow, and hence, the channel agility (uncorrelated/persistent) characterizes the channel of each flow. We define a class-based scheduling scenario, where each class comprises flows with identical time-fraction requirement and channel agility. A scheduling scenario is input (*channel*)-homogeneous if all classes have identical time-fraction requirement (*channel agility*), and is input (*channel*)-heterogeneous otherwise.

We define a generic wireless scheduler that maps to the Unified Wireless-Fair Queuing Framework, from which most wireless schedulers proposed recently can be instantiated. At each scheduling instant, the Arbitration Scheme (AS) selects a flow based on the allocation sequence of the Slot Allocation Policy (SAP), the predicted channel state from the Channel Status Monitor (CSM), the lead/lag status of each flow from the Fairness Monitor (FM), and the Packet Dispatcher (DISP) dispatches the Head-of-Line (HOL) packet of the selected flow for transmission.

The abstraction of the wireless scheduler into various components enables us to isolate the contribution of each component to various QoS metrics (throughput, delay and fairness) defined over different time intervals. In addition, we also define a queuing model for each wireless receiver. We show that the QoS metrics as well as the wireless receiver buffer requirement can be expressed in terms of the statistics of the HOL packet delay of each flow.

Performance Analysis : We model the wireless scheduling mechanism as

a Markov process, from which the HOL packet delay probability density function (pdf) can be derived. We present a stochastic analysis of a Wireless-Fair Scheduler (WFS) for a two-flow scenario, assuming a simple Weighted Round Robin (WRR) scheduler for the SAP. Although the ergodicity of the Markov model is established, our modelling approach is limited in a multiple-flow scenario as the number of states that the FM needs to maintain increases quadratically with the number of flows. (See Chapter 3)

To overcome the above limitation, we propose an abstraction of the generic wireless scheduler model by disabling the FM. The corresponding Markov model of the resulting Channel-State Dependent (CSD) scheduler is ergodic in a multiple-flow scenario. We propose a novel concept of constrained state-transition matrices, based on which we present a framework for the performance analysis of CSD schedulers. With this framework, the HOL packet delay pdf can be derived more efficiently for a given SAP, AS and scheduling scenario without the tedious transient computations of Markov analysis. The complexity of the framework is determined by the computational complexity of products of matrices of dimensions $2^K \times 2^K$, where K is the total number of flows. (See Chapter 4)

The SAP and AS are further abstracted as follows:

- The SAP determines the QoS performance of the wireless scheduler under error-free conditions. We restrict the choice of the SAP to perfectly-fair (i.e., allocated share = time-fraction requirement) loop schedulers as they are simple to implement and are mathematically tractable. We describe and analyze the periodicity properties of various loop schedulers, and also derive the conditions for optimal per-flow periodicity (or minimum HOL packet delay variance). (See Chapter 6)
- The AS selects a flow for transmission amongst the set of eligible flows (i.e., flows with predicted error-free channels) in order

to emulate the performance of the SAP under error-prone conditions. In addition to a uniform arbitration scheme, we propose various prioritized arbitration schemes that assign a priority to each eligible flow in order to embed fairness provision into the CSD scheduler. Our objective is to emulate the functionalities of the FM in the generic scheduler model. (See Chapter 4)

In addition to the WFS and CSD schedulers, we propose a Channel-State Independent Fair-Aggregation (FA) scheduler that aggregates input-flows fairly into a single flow prior to FIFO scheduling. (See Chapter 3)

Numerical Results and Applications : We apply our analysis to derive the HOL packet delay pdf for each scheduler, from which QoS and other performance metrics are evaluated. We make the following observations:

Input-homogeneous Scenario : In this scenario, the choice of the SAP is trivial since the class of perfectly-fair loop schedulers reduces to a simple RR scheduler. In terms of the AS, our proposed prioritized scheme achieves a significant reduction in the wireless receiver buffer requirement for the CSD scheduler compared to uniform arbitration. (See Chapter 4)

When the scheduling scenario is also channel-homogeneous, we observe that while the FA scheduler achieves better QoS performance in an uncorrelated channel, the CSD scheduler is superior in a persistent channel. (See Chapter 3) Hence, for a channel-heterogeneous two-class scheduling scenario, we propose a novel hybrid CSD-FA scheduler that partitions the flows according to their channel agility, and then applies the respective scheduling mechanism to each partition. Numerical results suggest that the proposed scheduler achieves good overall throughput as well as low buffer requirements compared to the CSD scheduler. This stresses the importance to exploit the long-term error behavior

(channel agility), in addition to the instantaneous channel state in the design of wireless schedulers. (See Chapter 5)

Input-heterogeneous Scenario : Under error-free conditions, the performance of the CSD scheduler is given by the performance of its SAP. Based on the analysis of perfectly-fair loop schedulers, we show that the WRR scheduler exhibits the worst periodicity performance, while the WRR-with-spreading (*WRR-sp*) scheduler performs optimally for a two-flow scenario. Based on this observation, we propose a recursive implementation of the *WRR-sp* scheduler for a class-based scheduling scenario. Although the recursive scheduler achieves the best periodicity performance, it fails to guarantee intra-class fairness with respect to periodicity, which is desirable in a class-based scheduling scenario. On the other hand, an enhanced *WRR-sp* scheduler gives the best periodicity performance amongst those that maintain intra-class fairness. Hence, there is a trade-off between absolute periodicity and periodicity-fairness performance in the design of loop schedulers.

Under error-prone conditions, the wireless receiver buffer requirement of the CSD scheduler is significantly larger when the WRR scheduler is implemented as the SAP, which concurs with its worst periodicity performance under error-free conditions; the corresponding requirement is similar when any other loop scheduler is chosen. (See Chapter 6)

7.2 Future Work

Based on our current achievements, we propose a guideline for wireless scheduler design to optimize a particular performance metric for a given scheduling scenario in Fig. 7.1.

According to Fig. 7.1, we need to extend the current analysis to include the remaining scenarios. In addition, we also suggest the following work-items

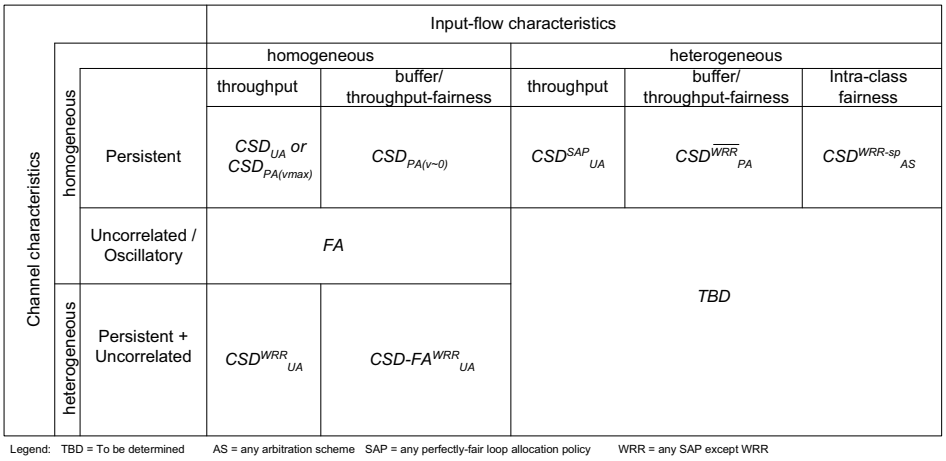

Figure 7.1: Recommendations for Wireless Scheduler Design to optimize given performance metric for various scheduling scenarios.

for future study:

Impact of ARQ mechanism : In our research, we did not address the influence of the ARQ policy on the QoS characterization of the wireless scheduler. We assume a simple Stop-and wait ARQ policy at the DISP, where all packets will eventually be transmitted, and will arrive at the wireless receiver in the same order as they arrive at the input-flows. This policy is suitable for loss-sensitive, but delay-tolerant applications and eliminates the need for re-ordering of packets at the receiver. However, for a delay-sensitive but loss-tolerant application, a trade-off between delay and packet loss can be achieved by dropping packets that exceed a given delay bound. Hence, further investigation of ARQ policies (e.g., [40] and references therein) can improve the QoS characterization of the wireless scheduler.

Impact of Channel Error Model : In our research, we did not address the effects of different wireless channel models on the QoS analysis of wireless schedulers. We considered a simple 2SMC error model, which is inadequate for certain fading channels according to recent works, and Markov chains with more states [26, 27] or higher order [28] have been

suggested. It may be worthwhile to extend our analysis to incorporate these models, although their effects on the QoS provisioning are not clear.

Impact of spatial distribution of wireless receivers : In our research, we did not address the effect of the spatial distribution of the wireless receivers in characterizing the channel; in fact, it is implicitly assumed that they are spatially apart, such that the channel states are independent. However, 'hotspots' may arise in realistic scenarios, and as a result, the channel states of receivers constituting each hotspot may be inter-dependent and be subjected to higher interference levels. Hence, it is important to incorporate this information in defining the scheduling scenario.

Impact of arrival characteristics : Let us consider the end-to-end delay of any packet, which comprises the delay between the source and the access point, and from the access point to the wireless receiver. Without cross-layer adaptation of TCP, we can assume that the first component is independent of the wireless channel characteristics. The second component can be partitioned into the queueing delay and the HOL delay.

In our research, we assumed that each flow is continuously backlogged, i.e., total number of backlogged flows is constant ($=K$). In this way, the steady-state HOL packet delay pdf for each flow can be computed and a G/G/1 model can be defined to evaluate the queueing delay in terms of the arrival characteristics.

On the other hand, under light loading conditions, flows can become idle and hence, the delay characteristics become inter-dependent amongst different flows, giving rise to a more complex delay analysis.

Alternative notions of allocation smoothness : In our research, we defined a metric based on the variance to quantify the allocation smoothness of perfectly-fair loop schedulers. However, recent works ([52] and

references therein) have treated the same problem using the notions of *balanceness* [53] and *regularity* [54]. Hence, it is interesting to compare the different notions of smoothness in terms of their effects on the QoS performance of the wireless scheduler.

Distributed / Uplink Scheduling : In our research, we did not address uplink QoS provisioning, since the distributed nature of uplink scheduling makes the problem harder than centralized downlink scheduling. Although this is a topic of intense recent study [60, 61, 62, 63, 64, 65, 66], much of the research is focused on achieving fairness in an Ad-Hoc Networking environment, but do not consider explicitly the wireless channel error model in their analysis. An extension of our analysis to a distributed scenario will be non-trivial, but is important for supporting interactive applications such as video-conferencing and internet-voice.

Integration with Handoff and Load-balancing : In addition to QoS provisioning via wireless scheduling, the AP is also responsible for mobility management functions such as handoff initiation and load-balancing. These functions are inter-dependent especially when the overlap region between APs is significant.

As an illustration, in our study, we consider the admissibility of wireless schedulers based on an efficiency and real-time constraint. These requirements stipulate a lower and upper bound on the total number of wireless receivers that can be supported respectively. If AP A carries an efficiency constraint while AP B supports flows with a real-time constraint, then AP B may initiate handoff for some receivers in the overlap region to AP A such that both constraints will be satisfied.

Appendix A

Waiting Time Distribution of Discrete-Time G/G/1 System

We consider a $G/G/1$ system for modelling the queuing behavior at wireless receiver j. Let us denote (in slots)

$$\begin{aligned} w^j(k) &= \text{waiting time of the } k^{th} \text{ packet} \\ s^j(k) &= \text{service time of the } k^{th} \text{ packet} \\ n^j(k) &= \text{inter}-\text{arrival time between the } k^{th} \text{ and } (k+1)^{th} \text{ packet} \end{aligned}$$

For simplicity of notations, we will drop the superscript j that denotes the flow index. From Fig. A.1, if we let $\varsigma(k) = s(k)$-$n(k)$, then we have the following recurrence relation:

$$w(k+1) = max\{0, w(k) + \varsigma(k)\} \tag{A.1}$$

Given the pdf of ($s(k)$,$n(k)$), the exact pdf of $w(k)$ can be evaluated according to an iterative approach described in [67]. We shall illustrate this approach by considering a constant-rate server at the wireless receiver, i.e., $s(k) = S$ $\forall$ k.

Since $\varsigma(k) = s(k)$ - $n(k)$ and the variables $s(k)$ and $n(k)$ are independent, p_ς

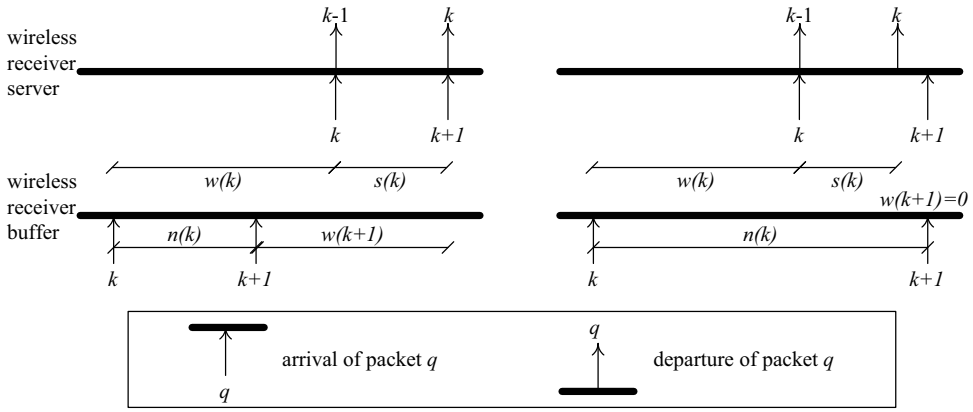

Figure A.1: Timing diagram at wireless receiver buffer: Relationship between $w(k+1)$ and $w(k)$,$s(k)$ and $n(k)$ if packet k departs the server after (left) and before (right) packet $k+1$'s arrival at the buffer.

is given by the following discrete convolution:

$$\begin{aligned} p_\varsigma(\zeta) &= p_n(-\zeta) \otimes p_s(\zeta) \\ &= \sum_{i=-\infty}^{\infty} p_n(-\zeta + i) p_s(i) \\ &= p_n(-\zeta + S) \end{aligned} \tag{A.2}$$

From Eq. (A.1), we obtain the following recursion:

$$\begin{aligned} p_{w(k+1)}(W) &= \pi(p_{w(k)}(W) \otimes p_\varsigma(W)) \\ &= \pi(p_{w(k)}(W) \otimes p_n(-W + S)) \end{aligned} \tag{A.3}$$

where π is a special operator that modifies the pdf of its argument by replacing all of the probability associated with negative values of W with an impulse at $W{=}0$ whose area equals this probability.

As long as the initial waiting time pdf, $p_{w(0)}$, is known, Eq. (A.2) and Eq. (A.3) can be used to find the limiting probability function defined as:

$$p_w = \lim_{k \to \infty} p_{w(k)} \tag{A.4}$$

For the above ergodic distribution to exist, we require that $\rho = \frac{E[s]}{E[n]} < 1$.

Under the assumption of high-offered load scenario, i.e., $\rho \approx 1$, the waiting time for a continuous-time $G/G/1$ system approximates an exponential distribution [67] with mean given as follows:

$$\begin{aligned} E[w] &= \frac{Var[n] + Var[x]}{2E[n](1-\rho)} \\ &= \frac{Var[n]}{2E[n](1-\rho)} \end{aligned} \tag{A.5}$$

For a time-slotted system, we can approximate the exponential distribution of the waiting time by a geometric distribution (i.e., its discrete analogue) given as follows:

$$p_w = \frac{(1 - \frac{1}{E[w]})^W}{E[w]} \tag{A.6}$$

where $E[w]$ as given in Eq. (A.5).

Therefore, in order to improve the speed of convergence of the recursion in Eq. (A.3), we choose the initial waiting time distribution, $p_{w(0)}$, according to Eq. (A.6), i.e.,

$$\begin{aligned} p_{w(0)} &= (1 - \frac{2(E[n] - S)}{Var[n]})^{W(0)} \frac{2(E[n] - S)}{Var[n]} \\ &= (1 - \frac{2E[n](1-\rho)}{Var[n]})^{W(0)} \frac{2E[n](1-\rho)}{Var[n]} \end{aligned}$$

Bibliography

[1] A. Demers, S. Keshev, and S. Shenker, "Analysis and simulation of a Fair Queueing Algorithm," *Journal of Internetworking: Research and Experiance*, vol. 1, pp. 3–26, October 1990.

[2] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - the Single Node Case," *IEEE/ACM Transanctions on Networking*, vol. 1, pp. 344–357, June 1993.

[3] H. Sariowan and R. L. Cruz, "SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service," *IEEE/ACM Transanctions on Networking*, vol. 7, pp. 669–684, October 1999.

[4] M. Zorzi and R. Rao, "ARQ Error Control for Delay-Constrained Communications on Short-Range Burst-Error Channels," *Proc. of the IEEE VTC*, pp. 1528–1532, May 1997.

[5] D. Wu and R. Negi, "A Wireless Channel Model for Support of Quality of Service," *Proceedings of the IEEE Globecom*, vol. 1, pp. 695–699, November 2001.

[6] K. Lee, "Performance Bounds in Communication Networks with Variable-Rate Links," *Proc. of the ACM SIGCOMM*, pp. 126–136, August 1995.

[7] R. L. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. Information Theory*, vol. 37, pp. 114–131, January 1991.

[8] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Transanctions on Networking*, vol. 1, pp. 372–385, June 1993.

[9] R. L. Cruz, "Quallity of service management in integrated service networks," *reprinted from the proc. 1st Semi-Annual Research Review, CWC, UCSD*, June 1996.

[10] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity," *IEEE Trans. Information Theory*, vol. 39, pp. 466–478, March 1993.

[11] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi, "Enhancing throughput over wireless LANs using Channel State Dependent Packet Scheduling," *Proc. of the IEEE INFOCOM*, vol. 3, pp. 1133–1140, March 1996.

[12] X. Liu, E. K. Chan, and N. B. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE Jounnal on Selected Areas in Communications*, vol. 19, pp. 2053–2064, October 2001.

[13] S. Shakkottai and A. L. Stolyar, "Scheduling for Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule," *American Mathematical Society Translations Series 2*, vol. 207, pp. 185–202, 2002.

[14] S. Shakkottai and A. L. Stolyar, "Scheduling Algorithms for a Mixture of Real and Non-Real Time Data in HDR," *Proc. of the 17th International TeleTraffic Congress*, September 2001.

[15] Y. Cao and V. Li, "Scheduling Algorithms in Broadband Wireless Networks," *Proc. of the IEEE*, vol. 89, pp. 76–87, January 2001.

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

[17] 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.

[18] 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.

[19] D. Eckhardt and P. Steenkiste, "Effort-limited Fair (ELF) Scheduling for Wireless Networks," *Proc. of the IEEE INFOCOM*, vol. 3, pp. 1097–1106, March 2000.

[20] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks," *Proc. of the ACM MOBICOM*, pp. 1–9, October 1998.

[21] C. Fragouli, M. Srivastava, and V. Sivaraman, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-based Queuing with Channel-State-Dependent Packet Scheduling," *Proc. of the IEEE INFOCOM*, vol. 2, pp. 572–580, March 1998.

[22] P. Lin, B. Bensaou, Q. Ding, and K. Chua, "CS-WFQ: A Wireless Fair Scheduling Algorithm for Error-Prone Wireless Channels," *Proc. of the IEEE ICCN*, pp. 276–281, October 2000.

[23] 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.

[24] M. Reisslein, K. Ross, and S. Rajagopal, "A Framework for Guaranteeing Statistical QoS," *IEEE/ACM Transanctions on Networking*, vol. 10, pp. 27–42, February 2002.

[25] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, pp. 2316–2323, August 1995.

[26] M. Zorzi and R. R. Rao, "On channel modeling for delay analysis of packet communications over wireless links," *36th Annual Allerton Conference on Commmunications, Control and Computing*, pp. 526–535, September 1998.

[27] Y. Cao and V. Li, "Scheduling Algorithms in Broadband Wireless Networks," *Proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 825–830, September 1998.

[28] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig, "A Markov-based Channel Model Algorithm for Wireless Networks," *Proc. of the ACM MSWiM*, pp. 28–36, July 2001.

[29] Y. Guo and H. Chaskar, "Class-based Quality of Service over Air Interfaces in 4G Mobile Networks," *IEEE Communications Magazine*, vol. 40, pp. 132–137, March 2002.

[30] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications," *Proc. of the IEEE INFOCOM*, pp. 636–646, June 1994.

[31] P. Goyal, H. M. Vin, and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Transanctions on Networking*, vol. 5, pp. 690–704, October 1997.

[32] P. Havinga and G. Smit, "QoS scheduling for energy efficient wireless communications," *International Conference on Information Technology: Recent Advances in Wireless Communications*, pp. 167–171, April 2001.

[33] P. Nuggehalli, V. Srinivasan, and R. R. Rao, "Delay Constrained Energy Efficient Transmission Strategies for Wireless Devices," *Proc. of the IEEE INFOCOM*, vol. 3, pp. 1765–1772, June 2002.

[34] E. U. Biyikoglu, B. Prabhakar, and A. E. Gamal, "Energy-efficient Scheduling of Packet Transmissions over Wireless Networks," *IEEE/ACM Transanctions on Networking*, vol. 10, pp. 487–499, August 2002.

[35] R. Rom and H. P. Tan, "Stochastic Analysis and Performance Evaluation of Wireless Schedulers," *Wiley Journal of Wireless Communications and Mobile Computing Special Issue: Performance Evaluation of Wireless Networks*, vol. 4, pp. 19–41, February 2004.

[36] J. A. Cobb, "Preserving Quality of Service Guarantees in spite of Flow Aggregation," *IEEE/ACM Transanctions on Networking*, vol. 10, pp. 43–53, February 2002.

[37] G. Armitage, *Quality of Service in IP Networks: Foundations for a Multi-Service Internet*. Macmillan, 2000.

[38] R. Rom and H. P. Tan, "Stochastic Analysis of Symmetric Two-flow Wireless-Fair Scheduling," CCIT Tech Report 371, Technion, Israel Institute of Technology, March 2002. Available at `http://www.ee.technion.ac.il/CCIT/info/Publications/Sceintific_e.asp`.

[39] R. Rom and H. P. Tan, "Performance Tradeoffs in Wireless Scheduling with Flow Aggregation," *Proc. of the IEEE WCNC*, vol. 3, pp. 1633–1638, March 2003.

[40] N. C. Ericsson, S. Falahati, A. Ahlen, and A. Svensson, "Scheduling and Adaptive Transmission for the Downlink in 4G Systems," *Proc. of the Future Telecommunication Conference*, November 2001.

[41] R. Rom and H. P. Tan, "Framework for Performance Analysis of Channel-aware Wireless Schedulers," *Submitted to Elsevier Journal on Performance Evaluation*, April 2004. Presented in-part at the 23^{rd} IEEE IPCCC, April 2004.

[42] R. Rom and H. P. Tan, "Framework for Delay Analysis of Channel-aware Wireless Schedulers," CCIT Tech Report 423, Technion, Israel Institute of Technology, May 2003. Available at `http://www.ee.technion.ac.il/CCIT/info/Publications/Sceintific_e.asp`.

[43] R. Rom and H. P. Tan, "Analysis of Trade-offs between Buffer and QoS Requirements in Wireless Networks," *Accepted for publication in the*

16^{th} *ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, August 2004. Also available as CCIT Tech Report 471, Technion, Israel Institute of Technology, February 2004 at `http://www.ee.technion.ac.il/CCIT/info/Publications/Sceintific_e.asp`.

[44] S. Baruah, G. Buttazzo, S. Gorinsky, and G. Lipari, "Scheduling periodic task systems to minimize output jitter," *Proc. of the IEEE International Conference on Real-Time Computing Systems and Applications*, pp. 62–69, December 1999.

[45] A. Bar-Noy, R. Bhatia, J. Naor, and B. Schieber, "Minimizing Service and Operation Cost of Periodic Scheduling," *Proc. of the ACM Symposium on Discrete Algorithms*, pp. 11–20, January 1998.

[46] Z. Brakerski, A. Nisgav, and B. Patt-Shamir, "General Perfectly Periodic Scheduling," *Proc. of the ACM Symposium on Principles of Distributed Computing*, pp. 163–172, July 2002.

[47] A. Bar-Noy, A. Nisgav, and B. Patt-Shamir, "Nearly Optimal Perfectly-Periodic Schedules," *Proc. of the ACM Symposium on Principles of Distributed Computing*, pp. 107–116, August 2001.

[48] A. Bar-Noy, V. Dreizin, and B. Patt-Shamir, "Efficient Periodic Scheduling by Trees," *Proc. of the IEEE INFOCOM*, vol. 2, pp. 791–800, June 2002.

[49] M. Hofri and Z. Rosberg, "Packet Delay under the Golden Ratio Weighted TDM Policy in a Multiple-Access Channel," *IEEE Trans. Information Theory*, vol. 33, pp. 341–349, May 1987.

[50] Z. Rosberg, "Optimal Decentralized Control in a Multiaccess Channel with Partial Information," *IEEE Transactions on Automatic Control*, vol. 28, pp. 187–193, February 1983.

[51] A. Itai and Z. Rosberg, "A Golden Ratio Control Policy for a Multiple-Access Channel," *IEEE Trans. Information Theory*, vol. 33, pp. 341–349, May 1987.

[52] S. Sano, N. Miyoshi, and R. Kataoka, "m-Balanced words: A generalization of balanced words," *Theoretical Computer Science*, vol. 314, pp. 97–120, February 2004.

[53] E. Altman, B. Gaujal, and A. Hordijk, "Balanced Sequences and Optimal Routing," *Journal of the ACM*, vol. 47, pp. 752–775, July 2000.

[54] B. Hajek, "Extremal Splittings of Point Processes," *Mathematics of Operations Research*, vol. 10, pp. 543–556, November 1985.

[55] M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin," *IEEE/ACM Transanctions on Networking*, vol. 4, pp. 375–385, June 1996.

[56] H. Zhang, "Service disciplines for guaranteed performance service in packet - switching networks," *Proceeding of the IEEE*, vol. 83, pp. 1374–1399, October 1995.

[57] V. Do and K. Yun, "An Efficient Frame-Based Scheduling Algorithm: Credit Round Robin," *Proc. of the IEEE Workshop on HPSR*, pp. 103–110, June 2003.

[58] R. Rom, M. Sidi, and H. P. Tan, "Performance Analysis of a Recursive Cyclic Scheduler for Class-based Scheduling," *Accepted for publication in the* 16^{th} *ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, August 2004. Also available as CCIT Tech Report 470, Technion, Israel Institute of Technology, February 2004 at `http://www.ee.technion.ac.il/CCIT/info/Publications/Sceintific_e.asp`.

[59] S. Viswanathan, "Future View of Broadband Demand," FCC TAC Meeting, Intel, April 2003. Available at `http://www.fcc.gov/oet/tac/TAC_III_04_17_03/Future_View_of_Broadband_De%mand.ppt`.

[60] N. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless Lan," *Proc. of the ACM MOBICOM*, pp. 167–178, August 2000.

[61] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints," *Proc. of the ACM MOBICOM*, pp. 200–209, July 2001.

[62] C. Barrack and K. Y. Siu, "A distributed scheduling algorithm for quality of service support in multiaccess networks," *Proc. of the IEEE ICNP*, pp. 200–209, October 1999.

[63] H. Luo and S. Lu, "A topology independent fair queueing model in Ad Hoc Wireless Networks," *Proc. of the IEEE ICNP*, pp. 200–209, August 2000.

[64] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," *Proc. of the ACM MOBICOM*, pp. 87–98, August 2000.

[65] X. Huang and B. Bensaou, "On max-min fairness and scheduling in Wireless Ad-Hoc Networks: analytical framework and implementation," *Proc. of the ACM MOBIHOC*, pp. 221–231, October 2001.

[66] L. Bao and J. Garcia, "A new approach to channel-access scheduling in Ad-Hoc Networks," *Proc. of the ACM MOBICOM*, pp. 210–221, July 2001.

[67] L. Kleinrock, *Queueing Systems Vol II: Computer Applications*. John-Wiley and Sons, 1976.