Stochastic Analysis and Performance Evaluation of Wireless Schedulers

Raphael Rom and Hwee Pink Tan
Department of Electrical Engineering
Technion, Israel Institute of Technology
E-mail: rom@ee.technion.ac.il and hweepink@tx.technion.ac.il

Abstract

In the last few years, wireless scheduling algorithms have been proposed by supplementing wireline scheduling algorithms with a wireless adaptation scheme. However, QoS bounds have either been derived for flows that perceive error-free conditions, or a static worst case channel condition. Such an assumption of the channel condition is unrealistic since channel errors are known to be bursty in nature. Hence, these bounds are inadequate to characterize the scheduler's QoS performance.

Our research focuses on performing an extensive analysis of wireless scheduling in order to derive statistical QoS performance bounds under realistic channel conditions. In this paper, we develop stochastic models for various wireless schedulers. Based on these models, we define and evaluate statistical QoS performance metrics in terms of throughput, delay and fairness under various channel conditions and over different time scales.

Numerical results indicate that no single scheduler out-performs the others in terms of all the QoS metrics under all channel conditions. The choice of an optimal scheduling mechanism depends on the priority of QoS requirements as well as the channel conditions.

Index Terms

Statistical QoS, Wireless Scheduling, Fairness.

I. Introduction

The huge success of mobile telephony brought about by Second Generation (2G) Mobile Networks represents the importance of mobile-connectivity in the user's perspective. Along with a phenomenal growth of Internet users, we envisage the demand for fixed-configuration wireless networks to bring network applications from the wired network to mobile users. The diversity of these applications as well as the unique characteristics of the wireless media make the provision of different, but yet consistent and predictable, grades of Quality of Service (QoS) a challenging task. To achieve this, given the performance requirements (e.g., delay and loss probability) and traffic descriptions (e.g., throughput, burstiness), the scheduling policy will determine how fairly and efficiently resources (wireless channel access) will be allocated to each user being served.

A. Related Work in QoS Provisioning

An abundance of scheduling policies that provide guaranteed QoS for wireline networks exists 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 assumptions regarding the arriving traffic.

Several works have attempted to extend the provision of guaranteed services by wireline scheduling to wireless links. In [4], 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 [5] work-conserving server-node are burstiness-constrained [6], 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 [7], 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 and hence, resources may not be efficiently utilized.

In [8], the author 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 have been characterized and an optimal policy that maximizes the throughput and minimizes the delay has been 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 applies to the special case of symmetric queues (i.e., all queues have similar arrival, service and connectivity statistics).

B. Related work in Wireless Scheduling

Direct application of wireline scheduling to the wireless media is not useful due to the following unique characteristics: (a) high channel error rate (b) bursty and time-varying channel capacity (b) location dependent channel capacity (c) user mobility and (d) power constraint of mobile users.

Instead of extending the QoS provisioning capability of wireline scheduling to wireless links, several wireless scheduling algorithms have been proposed that adapt to the wireless media so as to achieve some performance requirements given some resource allocation constraints. Such constraints are often specified in terms of the long-term fraction of time to be allocated to each user (*time-fraction requirement*). These algorithms can be categorized broadly as follows:

1) Wireline Scheduling with Wireless Adaptation: Wireless Schedulers in this group typically comprise a wireline queuing scheme as well as a wireless adaptation scheme. While scheduling is performed using the wireline algorithm under error-free conditions, the wireless adaptation scheme is employed when these conditions no longer prevail. Most wireless scheduling algorithms proposed recently adopt flow swapping and reassignment as the wireless adaptation scheme.

With this scheme, when a flow i that is scheduled for transmission predicts channel errors, the scheduler selects another flow j that perceives an error-free channel for transmission. After this swapping, flow i *lags* behind the service it is entitled to while flow j gains a *lead* over its entitled service. The scheduler accounts for the 'lost' transmission opportunity 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. The extent to which flow swapping is permitted determines such a trade-off. 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.

Most of the wireless schedulers that employ the above wireless adaptation scheme can be mapped onto a Unified Wireless Fair Queuing Framework [9]. In most of these algorithms, analytical bounds for QoS performance have been derived under error-free conditions; in the case of Wireless-Fair Service (WFS) [10], such bounds have been derived under error-prone conditions, which are more meaningful for assessing the performance of the scheduler. As an example, on average, *all* HOL packets for a *lagging* flow *i*, with time-fraction requirement r_i, given the worst case channel error rate of p_e, will be successfully transmitted within d_i^e slots, given by:

$$d_i^e \leq (\sum_{j \epsilon F} \frac{L_p}{C}) + \frac{L_p}{Cr_i} \frac{1}{(1-p_e)} \tag{1}$$

where F = set of all flows, C = channel capacity in bits/sec and L_p = packet size in bits/packet.

Certain applications, e.g., continuous media, with strict delay constraints, can typically tolerate non-zero losses. Therefore, guaranteed QoS constraints (such as Eq. (1)) may be overly conservative for such applications. This prompted research on providing statistical QoS guarantees (e.g., [11] and references therein). Instead of specifying a single QoS constraint α that will be satisfied by *all* packets, statistical QoS provisioning bounds the fraction of packets, β, that exceeds the QoS constraint, α. 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 I-A.

Another limitation of Eq. (1) lies in the fact that wireless channel errors are typically bursty in nature, and therefore cannot be sufficiently characterized by a worst-case error rate. Hence, the bound given in Eq. (1) will be conservative in a practical wireless channel, and may lead to sub-optimal use of scarce radio resources.

2) Opportunistic Scheduling [12]: In this work, the authors defined $\{U_i^k\}$ as a stochastic process that is associated with user *i*. The value of $\{U_i^k\}$ measures the “worth” of time slot *k* to user *i*, and is in general a function of its channel condition. Usually, the better the channel condition of user *i*, the larger the value of $\{U_i^k\}$. At each scheduling instant, *k*, the system is “rewarded” with a performance value of U_i^k if user *i* is scheduled to transmit.

Instead of extending existing wireline schedulers, the authors defined the scheduling problem as one of maximizing the *average* system performance while satisfying the resource allocation constraint. Hence, a feasible scheduler is one that assigns a time slot to a user with the largest performance value relative to the other users.

The authors proposed an opportunistic scheduling policy that solves the scheduling problem optimally and at the same time improves *every* user’s average performance relative to any non-opportunistic scheduling policy, i.e., traditional wireline schedulers that do not make use of channel information, e.g., WRR. This is verified in terms of simulation results. The authors also proposed an extension to the scheme to take into account short-term performance requirements of users, and demonstrated its effectiveness via simulations. However, it is unclear how the scheme performs in terms of per-flow QoS when handling delay-sensitive traffic.

C. Research Contributions of This Paper

In this paper, we consider a generic wireless scheduler (termed *Wireless-Fair Scheduler* or WFS) as an instance of the Unified Wireless Fair Queuing Framework. Such a scheduler comprises the following components: (a) error-free scheduling mechanism, (b) channel-status monitor (CSM), (c) fairness monitor (FM) and (d) packet dispatcher.

In order to establish the contribution of each component to QoS provisioning, we define variants of the WFS that comprise a partial set of the components listed above. We develop Markov models and perform a stochastic analysis for each scheduler. Based on these models, we derive the statistical overall and per-flow QoS performance of each scheduler under different channel conditions. Such an analysis offers a more complete and accurate characterization of the QoS provision capability in comparison to those obtained via simulation.

This paper expands on the preliminary work performed in [13]. With respect to the discussion in Section I-B, our work offers the following contributions: (a) it extends the QoS analysis performed in [9] to more realistic channel assumptions and (b) supplements the work performed in [12] in terms of per-flow QoS support for delay-sensitive flows.

The rest of the paper is organized as follows: In Section III, we will define the wireless schedulers. In Section III, we will describe our approach for analysis and modeling of the wireless schedulers. QoS parameters will be defined and evaluated in terms on the scheduling models in Section IV. Numerical results for the performance evaluation of these schedulers are presented in Section V. Concluding remarks will be presented in Section VI.

II. Definition of Wireless Schedulers

In this section, we define the scheduling problem and describe the scheduling mechanism of WFS and its variants.

A. Scheduling Problem

We consider a centralized wireless scheduling problem where an access point coordinates the allocation of wireless access to *N* competing users (or flows). An example of such a scenario is in the downlink transmission from a base station in a fixed-configuration wireless network, where the base station forms the access point for mobile users in the vicinity. The resource allocation constraint is given in terms of the vector, $\underline{r}$ = $[r^1, r^2, ...r^N]$, where $r^j \epsilon$ Z+ and $R = \sum_{k=1}^{N} r^k$. The ratio $\frac{r^j}{R}$ denotes the time-fraction requirement of flow *j*, which is the fraction of resources that should be allocated to flow *j* in the long term. We assume that the wireless channel is slotted in time, where all slots are of equal size.

1) Flow Characteristics: Each flow is assumed to be delay sensitive and carry equal-sized packets with transmission time of one slot. We assume that all queues are of infinite length, and hence, packets can only be 'dropped' if their delay bounds have been exceeded.

2) Channel Characteristics: To take into account the spatial and temporal variance of the channel, we define the variable, c_i^j, to denote the channel state of flow *j* in slot *i*, and the vector $\underline{c_i} = [c_i^1, c_i^2, ...c_i^N]$. In the current study, we assume a Two-State Markov Chain error model for the wireless channel, i.e., in any slot *i*, $c_j^i \epsilon \{Good,\ Bad\}$. When the channel is in *Good* state, packet transmission is always successful; when it is in *Bad* state, packet transmission is never successful.

For each flow *j*, the model is specified in terms of the parameters, p^j_{ge} and p^j_{corr}, which are defined as follows:

$$p^j_{ge} = Prob(c^j_i = Bad \mid c^j_{i-1} = Good)$$
$$p^j_{corr} = p^j_{eg} + p^j_{ge}$$

The parameter, p^j_{corr}, is inversely proportional to the level of correlation in the error behavior across successive slots for flow *j*; a value close to 0 indicates high correlation while a value of 1.0 represents the special case of uncorrelated errors. Given these parameters, the steady-state average error rate of flow *j*, $p^j_B = \frac{p^j_{ge}}{p^j_{corr}}$. We assume that users are sufficiently separated spatially such that the channel state of different flows are independent.

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 [14] or higher order [15] 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 Two-State Markov Chain model, which is also analytically more tractable.

B. Definition of WFS

The components of the WFS are depicted in Fig. 1. At the beginning of slot *i*, the error-free scheduler allocates the transmission priority in slot *i* based on the vector $\underline{r}$. The CSM predicts the channel state of each flow in the current slot. Based on these information, as well as the input from the FM, the packet dispatcher selects a flow and transmits its HOL packet.

1) Error-free Scheduler: The error-free scheduler defines the slot allocation and flow transmission priority under error-free conditions according to the vector, $\underline{r}$. Variants of packetized fluid fair queueing algorithms (WFQ) [1], [2], [16], [17], [18], [10] are popular choices since they achieve throughput, delay and fairness bounds under error-free conditions. 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 with spreading (WRR-spreading) is equivalent to WFQ [18]. We choose WRR-spreading as our error-free scheduler since it is much easier to implement than WFQ. For ease of analysis, we shall approximate the mechanism of WRR-spreading with WRR.

Let us define A^{EF}_i such that if $A^{EF}_i = j,\ 1 \leq j \leq N$, then slot *i* is allocated to flow *j* and flow *j* has transmission priority in this slot. Given $\underline{r}$, $\{..., A^{EF}_{i-1}, A^{EF}_i, ..\}$ should resemble the sequence depicted in Fig. 2.

2) CSM: We assume that at the beginning of slot *i*, the CSM is aware of the channel status of each flow in previous slots, i.e., $\underline{c}_{i-1}$,$\underline{c}_{i-2}$...and can use this information to compute the prediction for the current slot, $\tilde{\underline{c}}_i$, of $\underline{c}_i = (c^1_i, c^2_i, ..., c^N_i)$. A simple algorithm for channel prediction is the one-step prediction algorithm, which is given as follows:

$$\tilde{c}^j_i = c^j_{i-1} \tag{2}$$

With this algorithm, a flow *j* 'qualifies' as a candidate for transmission in slot *i* only if $c^j_{i-1} = Good$. In this way, channel efficiency can be optimized by deferring the transmission of an allocated flow that predicts an erroneous channel and reassigning another flow that predicts an error-free channel to transmit instead (flow swapping).

3) FM: The Fairness Monitor keeps track of flow swapping activities in order to ensure some form of fairness provision while channel efficiency is optimized. The notion of per-flow lag (lead) is defined to monitor the amount of additional service that a flow is entitled to (needs to relinquish) in the future in order to compensate for service lost (gained) in the past due to flow swapping.

We define the lead/lag updating mechanism such that the lead and lag of flows *j* and *k* are updated whenever slot swapping takes place between these flows. Let $x^{j,k}_i$ denote the lead of flow *j* relative to flow *k* at the end of slot *i*. When flow *j* transmits in slot *i* that is allocated to flow *k*, $x^{j,k}_{i+1} = x^{j,k}_i + C(\underline{r})$ while $x^{k,j}_{i+1} = x^{k,j}_i - C(\underline{r})$, where $C(\underline{r})$ 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 $x^j_i = \sum_{k=1,k\neq j}^{N} x^{j,k}_i$:

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

The extent to which flow swapping is permitted represents a trade-off between channel efficiency and short-term fairness provision of the scheduler. 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.

4) Packet Dispatcher: At the beginning of each slot *i*, the packet dispatcher selects a flow for transmission based on the input $(A^{EF}_i, \underline{\tilde{c}}_i, \underline{x}_{i-1})$, where $\underline{x}_{i-1} = [x^1_{i-1}, x^2_{i-1}, ...x^N_{i-1}]$. We define the transmission heuristic that achieves optimal channel efficiency while retaining short-term fairness provision. Hence, slots are always allocated to the most lagging flow *j* if it exists; otherwise, transmission priority is defined by A^{EF}_i. However, as long as at least one flow *j* exists such that $\tilde{c}^j_i$=Good, transmission will be attempted in slot *i*. This is depicted in the following pseudocode:

WFS Packet Dispatcher Mechanism at slot i

1) Define F = $\{j : \tilde{c}^j_i = Good, 1 \leq j \leq N\}$
2) if F $\neq \{\}$
   a) G = $\{j : x^j_i < 0, j\epsilon F\}$
   b) if G $\neq \{\}$ then transmit flow k = $arg\min_{j\epsilon G} x^j_i$
   c) else
      i) if $A^{EF}_i \epsilon$ F then transmit A^{EF}_i
      ii) else transmit flow k = $arg\min_{j\epsilon F} x^j_i$

C. *Definition of EFF Scheduler*

We define an abstraction of the WFS (termed Channel-Efficient (or EFF) Scheduler) by removing the FM as shown in Fig. 3. In this case, at the beginning of each slot *i*, the packet dispatcher selects a flow for transmission based on the input $(A_i^{EF}, \underline{\tilde{c}}_i)$. Hence, the goal of the scheduling policy is to maximize channel efficiency, and the transmission heuristic is shown in the following pseudocode:

EFF Packet Dispatcher Mechanism at slot i

1) Define F = $\{j : \tilde{c}_i^j = Good, 1 \leq j \leq N\}$
2) if F $\neq \{\}$
   a) if $A_i^{EF} \epsilon$ F then transmit A_i^{EF}
   b) else transmit any flow j ϵ F

D. *Definition of WRR Scheduler*

We define a further abstraction of the WFS by removing the CSM from the EFF scheduler, resulting in a WRR scheduler as depicted in Fig. 4. In this case, at the beginning of each slot *i*, the packet dispatcher selects a flow for transmission based on the input A_i^{EF}. Hence, transmission may take place only in the event that $\tilde{c}_i^{A_i^{EF}}$=Good; otherwise, the slot is wasted.

E. *Definition of Fair Aggregation (FA) Scheduler*

A final abstraction of the WFS is defined based on the concept of flow aggregation. In flow aggregation, multiple flows are combined into a single aggregate flow prior to scheduling at the wireless link. Examples of application of flow aggregation include ATM networks and the Class-Based QoS Framework proposed in [19]. Flows may be aggregated according to various criteria, e.g., common QoS requirements and/or source-destination specifications.

We define our FA scheduler to comprise a fair aggregator followed by a packet dispatcher. The fair aggregator schedules packets from each flow according to $\underline{r}$ such that they arrive in a single queue at the packet dispatcher in the order as depicted in Fig. 5. Notice that the packet transmission cycle corresponds to the slot allocation cycle in the WRR scheduler.

III. Analysis and Modeling of Wireless Schedulers

In this section, we shall develop Markov models for the wireless schedulers defined in Section II. In order to illustrate the modeling approach, we shall consider the simple case of N=2 and $\underline{r}$= [1,R-1], where R>1, Rϵ Z+.

A. *WFS*

1) Error-free Scheduling - WRR slot allocation: For N-flow scheduling, given $\underline{r}$, in order to implement the slot allocation policy depicted in Fig. 2 under error-free conditions, we define the variable, *count*, which is incremented

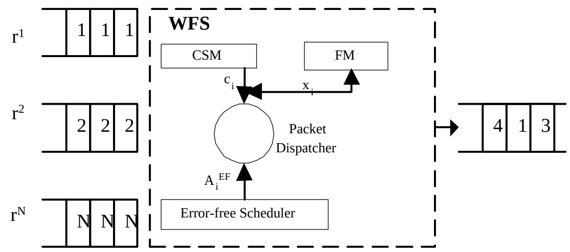


Fig. 1. Components of WFS

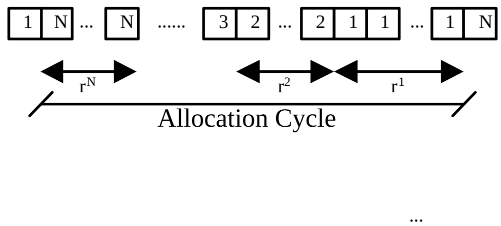


Fig. 2. Illustration of WRR slot allocation

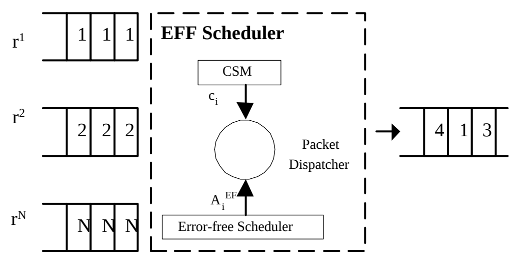


Fig. 3. Components of CSD-EFF Scheduler

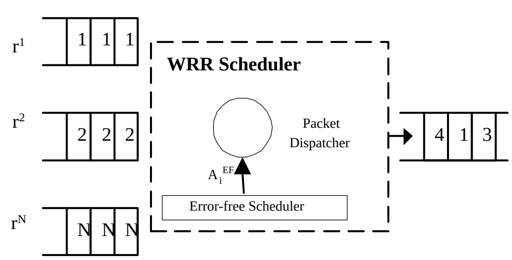


Fig. 4. Components of WRR Scheduler

at the end of each slot, and is reset to zero at the beginning of each allocation cycle (i.e., count modulo R). Given $count_{i-1}$, A_i^{EF} can be computed as follows:

$$A_i^{EF} = j, \; \sum_{k=1}^{j-1} r_k \leq count_{i-1} \leq \sum_{k=1}^{j} r_k - 1, \quad 1 \leq j \leq N \tag{3}$$

For the case of N=2, and $\underline{r}$= [1,R-1], Eq. (3) becomes:

$$A_i^{EF} = \begin{cases} 1, & count_{i-1} = 0; \\ 2, & 1 \leq count_{i-1} \leq R-1. \end{cases} \tag{4}$$

2) FM: Recall that $x_i^{j,k}$ denotes the lead of flow *j* over flow *k* at the end of slot *i*. Since $x_i^{j,k}$ and $x_i^{k,j}$ are updated concurrently whenever flow swapping takes place between flows *j* and *k*, for N=2, this implies that it suffices to define a single variable, x_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 x_i is updated according to the transmission event that occurs in slot *i*. Let us denote the set of all possible events that can occur in any slot *i* by E_i, where E_i is defined as follows:

$$E_i = \left\{ \begin{array}{l} \text{No flow transmits in slot i (NF),} \\ \text{flow 1 transmits successfully in its allocated slot(f1S1),} \\ \text{flow 1 transmits successfully in slot allocated to flow 2(f1S2),} \\ \text{flow 2 transmits successfully in its allocated slot(f2S2),} \\ \text{flow 2 transmits successfully in slot allocated to flow 1(f2S1)} \end{array} \right\} \tag{5}$$

In any slot *i*, whenever no transmission takes place (i.e., event *NF*), neither flow achieves any lead over the other flow. However, when the event *f1S2* takes place, flow 1 gains in lead over flow 2; on the other hand, when the event *f2S1* takes place, flow 1 suffers a lag relative to flow 2. When flows transmit successfully in their allocated slots (i.e., event *f1S1* or *f2S2* 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 allocation). Its lag will be reduced after transmitting successfully in slot *i*.

Since the variable *count* tracks the error-free 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 x$\neq$ 0). Hence, we can generalize the procedure to update *count* as follows:

$$count_i = \begin{cases} count_{i-1} + 1, & x_i = 0; \\ count_{i-1}, & otherwise. \end{cases} \tag{6}$$

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

According to [9], each update in lag or lead either increments or reduces the value of *x* by *one* slot. Based on the above, the effects of each transmission event in slot *i* on x_i can be depicted as follows:

$$x_i = \begin{cases} x_{i-1}, & xsame; \\ x_{i-1} + 1, & xinc; \\ x_{i-1} - 1, & xdec. \end{cases} \tag{7}$$

where

$$xsame \equiv NF \cup [(f1S1 \cup f2S2) \cap x_{i-1} = 0]$$
$$xinc \equiv f1S2 \cup [f1S1 \cap x_{i-1} < 0]$$
$$xdec \equiv f2S1 \cup [f2S2 \cap x_{i-1} > 0]$$

However, according to the algorithms in [9], if some packetized fluid fair queuing mechanism is used as the error-free service, 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 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:

$$x_i = \begin{cases} x_{i-1}, & xsame; \\ x_{i-1} + R - 1, & xinc; \\ x_{i-1} - 1, & xdec. \end{cases} \quad (8)$$

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) Packet Dispatcher: Let us define A_i as the allocation in slot *i* under all channel conditions. Recall that we defined the transmission heuristic such that transmission priority is always given to the most lagging flow if it exists. We can expand Eq. (4) to incorporate this heuristic such that A_i is given as a function of $count_{i-1}$ and x_{i-1} as follows:

$$A_i = \begin{cases} 1, & x_{i-1} < 0 \cup \{x_{i-1} = 0 \cap count_{i-1} = 0\}; \\ 2, & x_{i-1} > 0 \cup \{x_{i-1} = 0 \cap count_{i-1} > 0\}. \end{cases} \quad (9)$$

As a result, the packet dispatcher mechanism in slot *i* is given as follows:

$$\text{Flow to transmit in slot i} = \begin{cases} 1, & \tilde{c}_i^1 = Good \cap [A_i = 1 \cup (A_i = 2 \cap \tilde{c}_i^2 = Bad)]; \\ 2, & \tilde{c}_i^2 = Good \cap [A_i = 2 \cup (A_i = 1 \cap \tilde{c}_i^1 = Bad)]. \end{cases} \quad (10)$$

4) Markov Model of WFS with N=2, $\underline{r}$=[1,R-1]: According to Eq. (9) and (10), the variables $(A_i, \underline{\tilde{c}}_i, count_{i-1}, x_{i-1})$ are required to implement the packet dispatcher mechanism of WFS in slot *i*. Based on Eq. (7) and (8), given x_{i-1}, the value of x_i depends on the transmission event in slot *i*. The value of A_i determines the set of allowable events, i.e., when $A_i = 1$, the allowable events are $\{NF, f1S1, f2S1\}$ whereas if $A_i = 2$, the allowable events are $\{NF, f2S2, f1S2\}$. Based on Eq. (9), the value of A_i can be determined given x_{i-1} and $count_{i-1}$.

The probability of occurrence of each transmission event in slot *i* depends only on $\underline{c}_i$ and $\underline{\tilde{c}}_i$, which depends only on the parameters of the channel error and prediction model. Hence, given these models, if x_{i-1}, $count_{i-1}$

are known for all flow *j*, x_i and $count_i$ can be determined. In other words, the two-flow WFS can be modeled as a two dimensional Markov Chain with state variables given by $\{(x_i, count_i),\ i = 1,2,3...\}$, where the Markov points are defined at the end of each slot. A detailed analysis for the special case of *R*=2 can be found in [20].

B. Markov Model of EFF scheduler with N=2, r=[1,R-1]

We characterize the EFF scheduler by considering it as a special case of the WFS with x_i=0. Hence, Eq. (6) ,(9) and (10) becomes:

$$count_i = count_i + 1 \tag{11}$$

$$A_i = \begin{cases} 1, & count_{i-1} = 0; \\ 2, & count_{i-1} > 0. \end{cases} \tag{12}$$

$$\text{Flow to transmit in slot i} = \begin{cases} 1, & \tilde{c}_i^1 = Good \cap [A_i = 1 \cup (A_i = 2 \cap \tilde{c}_i^2 = Bad)]; \\ 2, & \tilde{c}_i^2 = Good \cap [A_i = 2 \cup (A_i = 1 \cap \tilde{c}_i^1 = Bad)]. \end{cases} \tag{13}$$

Hence, given the channel error and prediction model, if $count_{i-1}$ is known for all flow *j*, $count_i$ can be determined. In other words, the EFF scheduler can be modeled as a one dimensional Markov Chain with state variable given by $\{count_i,\ i = 1,2,3...\}$, where the Markov points are defined at the end of each slot.

C. Markov Model of WRR scheduler with N=2, r=[1,R-1]

The WRR is a special case of the EFF scheduler where flows can only transmit in slots that are allocated to them. Hence, Eq. (11) ,(12) and (13) becomes:

$$count_i = count_i + 1 \tag{14}$$

$$A_i = A_i^{EF} \tag{15}$$

$$\text{Flow to transmit in slot i} = j; \quad A_i = j \cap \tilde{c}_i^j = Good \tag{16}$$

As with the EFF scheduler, the WRR can be modeled as a one dimensional Markov Chain, $\{count_i,\ i = 1,2,3...\}$, where the Markov points are defined at the end of each slot.

D. Markov Model of FA scheduler with N=2, r=[1,R-1]

The FA scheduler transmits packets according to the transmission cycle depicted in Fig. 5. This is equivalent to a modified WRR scheduler that allocates slots to flow *j* until r^j packets have been *transmitted* before switching to flow *j*+1 and so on. In order to implement this slot allocation policy, we modify the definition of $count_i$ such that *count* is incremented at the end of each slot *only* when a successful packet transmission takes place, and is reset to zero at the beginning of each transmission cycle, i.e., Eq. (11) becomes:

$$count_i = \begin{cases} count_{i-1} + 1, & \text{transmission in slot i;} \\ count_{i-1}, & \text{otherwise.} \end{cases} \tag{17}$$

As with the WRR scheduler, the FA scheduler can be modeled as a one dimensional Markov Chain, $\{count_i,\ i = 1,2,3...\}$, where the Markov points are defined at the end of each slot.

IV. DEFINITION AND EVALUATION OF QOS PARAMETERS FOR WIRELESS SCHEDULERS

In this section, we shall derive and evaluate the performance metrics of wireless schedulers based on the Markov models obtained in Section III.

A. Definition of Intervals for Performance Analysis

Let us consider the queue status of each flow *j*, q_i^j, over an interval of time for two-flow scheduling, as shown in Fig. 6, where $q_i^j \epsilon \{Backlogged,\ Idle\}$. At slot t_1, $q_{t_1}^1 = Backlogged$, and since $q_{t_1}^2 = Idle$, all slots are allocated to flow 1 until slot t_2 when $q_{t_2}^2 = Backlogged$. During the interval $[t_2, t_3]$, the scheduling policy determines the slot allocation mechanism. During this interval, the state variables evolve according to Section III. At t_3, $q_{t_3}^j = Idle$, and hence, all slots are allocated to flow 1 until t_4, when once again, $q_{t_4}^2 = Backlogged$.

Since the scheduler operates in disjoint intervals where both flows are backlogged (e.g., $[t_2, t_3]$ and $[t_4, t_5]$ in Fig. 6), we need to define a procedure to determine the values of the state variables at the beginning of each interval where both flows are backlogged. In this study, we initialize the state variables at the beginning of each such interval as follows:

$$
\begin{aligned}
x_0 &= 0 \quad (for\ WFS\ only) \\
Prob(count_0 = k) &= \frac{1}{R}, \quad 0 \le k \le R-1
\end{aligned}
$$

In this way, the instances t_2 and t_4 are regenerative points with respect to the state variables and we define the interval between two successive regenerative points as a *regenerative interval*. Each such interval always begins with a sub-interval where *all* flows are backlogged, defined as a *performance interval*. We shall analyze the performance of each wireless scheduler within the performance interval.

B. Long Term Performance

Let us consider packet *p* of flow *j* that becomes Head-of-Line (HOL) at the beginning of slot *k* and departs at the end of slot *k+n*. If we define the *delay* of packet *p* of flow *j* (denoted d_j^p) as the interval between the instant it became HOL to the instant it departs, then d_j^p = n.

Let us denote the delay probability density function of packet *p* of flow *j* as $\bar{d}_j^p(n)$. Assuming that it exists, the steady-state delay probability density function of flow *j*, $\bar{d}_j(n)$, can be written as follows:

$$\bar{d}_j(n) = \lim_{p \to \infty} \bar{d}_j^p(n) \tag{18}$$

Given $\bar{d}_j(n)$, we define the following long-term performance metrics:

1) Mean delay, μ_{d_j} and Delay jitter, σ_{d_j} : The mean delay and delay jitter for each flow *j* is defined as follows:

$$
\begin{aligned}
\mu_{d_j} &= \sum_{n=1}^{\infty} \bar{d}_j(n) \times n \\
\sigma_{d_j} &= \sqrt{\sum_{n=1}^{\infty} \bar{d}_j(n) \times n^2 - \mu_{d_j}^2}
\end{aligned} \tag{19}
$$

2) Throughput: Let us define $T_j(t)$ as the number of successful flow *j* transmissions in the interval [0,t]. Since the throughput of flow *j* (denoted $S_j(t)$) over the same interval is defined as the average number of transmissions per slot, we have:

$$S_j(t) = \frac{E[T_j(t)]}{t} \tag{20}$$

If μ_{S_j} denotes the steady-state throughput of flow *j* and μ_S denotes the total throughput, we have the following:

$$\begin{aligned} \mu_{S_j} &= \lim_{t\to\infty} S_j(t) = \frac{1}{\mu_{d_j}} \\ \mu_S &= \sum_{j=1}^{N} \mu_{S_j} \end{aligned} \tag{21}$$

3) Fairness: We consider the notion of fairness with respect to the individual flow (absolute fairness) as well as a pair of flows (relative fairness). A flow *j* expects that the fraction of service *obtained* within an interval will be as close to the fraction it *should* get (i.e., r^j) as possible. Between any pair of flows, a flow expects that it should be fairly treated relative to another flow. We consider different fairness metrics for the following cases:

(a) Homogeneous case (r^j=1)

In this case, we are interested in the fairness performance for the individual user. 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:

$$\begin{aligned} \Delta(t) &= \frac{T_j(t)}{t} - \frac{r^j}{R} \\ &= \frac{T_j(t)}{t} - \frac{1}{N} \end{aligned} \tag{22}$$

For good long-term fairness performance, it is desirable to minimize the expected value of $\Delta(t)$. Hence, we compute the first-order statistic, μ_Δ, as follows:

$$\begin{aligned} \mu_\Delta &= \lim_{t\to\infty} E[\Delta(t)] \\ &= \mu_{S_j} - \frac{1}{N} \end{aligned} \tag{23}$$

(b) Non-homogeneous case ($r^j \neq r^k$)

In this case, in addition to fairness for the individual flow, we need to evaluate the relative fairness between two flows, *j* and *k*. The Relative Fairness Bound, RFB(t), is a useful metric for such an evaluation, where

$$RFB(t) = |\frac{\frac{T_j(t)}{t}}{\frac{r^j}{R}} - \frac{\frac{T_k(t)}{t}}{\frac{r^k}{R}}| \tag{24}$$

For good long-term fairness performance, we desire the expected value of RFB(t) to be as small as possible. Hence, we define and evaluate the following metric:

$$\begin{aligned} RFB &= \lim_{t\to\infty} E[RFB(t)] \\ &= |\frac{\mu_{S_j}}{\frac{r^j}{R}} - \frac{\mu_{S_k}}{\frac{r^k}{R}}| \end{aligned} \tag{25}$$

C. Short Term Performance

From the description in Section IV-B.3, we can evaluate the short-term fairness performance by evaluating statistics based on the metrics, $\Delta(t)$ and RFB(t), as a function of *t*. To do so, we need to evaluate the distribution of T_j(t), which we shall illustrate for two-flow WFS. The corresponding evaluation for the other schedulers can be carried out in a similar manner.

Consider a sequence of flow *j* packet departures and their associated delays as depicted in Fig. 7. For simplicity of notations, we shall drop the subscript *j* that denotes flow *j*. We define the following notations:

$$
\begin{aligned}
y^p &\equiv value\ of\ x\ at\ departure\ of\ packet\ p \\
ct^p &\equiv value\ of\ count\ at\ departure\ of\ packet\ p \\
\tilde{d^p} &\equiv \sum_{m=1}^{p} d^m \\
k(\tilde{n}^p, Y^p, C^p) &\equiv Prob(\tilde{d^p} = \tilde{n}^p, y^p = Y^p,\ ct^p = C^p) \\
f(n^{p+1}, Y^{p+1}, C^{p+1} \mid Y^p, C^p) &\equiv Prob(d^{p+1} = n^{p+1}, y^{p+1} = Y^{p+1}, ct^{p+1} = C^{p+1} \mid y^p = Y^p,\ ct^p = C^p) \\
g(\tilde{n}^{p+1}, Y^{p+1}, C^{p+1}, Y^p, C^p, \tilde{n}^p) &\equiv Prob(\tilde{d}^{p+1} = \tilde{n}^{p+1}, y^{p+1} = Y^{p+1}, ct^{p+1} = C^{p+1}, y^p = Y^p,\ ct^p = C^p, \tilde{d^p} = \tilde{n}^p)
\end{aligned}
$$

$\tilde{d^p}$ denotes the actual departure time of packet *p* if we assume that the first packet becomes HOL at time 0. We proceed to compute the distribution of $T(t)$ as follows:

$$
\begin{aligned}
Prob(T(t) = 0) &= Prob(\tilde{d^1} > t) \\
Prob(T(t) = 1) &= Prob(\tilde{d^1} \le t,\ \tilde{d^2} > t) \\
&= \sum_{\tilde{n}^1=1}^{t} \sum_{\tilde{n}^2=t+1}^{\infty} Prob(\tilde{d^2} = \tilde{n}^2, \tilde{d^1} = \tilde{n}^1) \\
&= \sum_{\tilde{n}^1=1}^{t} \sum_{\tilde{n}^2=t+1}^{\infty} Prob(d^2 = \tilde{n}^2 - \tilde{n}^1, \tilde{d^1} = \tilde{n}^1) \\
&= \sum_{C^1=0}^{N-2} \sum_{Y^1} \sum_{\tilde{n}^1=1}^{t} \sum_{\tilde{n}^2=t+1}^{\infty} Prob(d^2 = \tilde{n}^2 - \tilde{n}^1, \tilde{d^1} = \tilde{n}^1, y^1 = Y^1, ct^1 = C^1) \\
&= \sum_{C^1=0}^{N-2} \sum_{Y^1} \sum_{\tilde{n}^1=1}^{t} \sum_{\tilde{n}^2=t+1}^{\infty} Prob(d^2 = \tilde{n}^2 - \tilde{n}^1 \mid y^1 = Y^1, ct^1 = C^1) k(\tilde{n}^1, Y^1, C^1) \\
Prob(T(t) = 2) &= Prob(\tilde{d^2} \le t,\ \tilde{d^3} > t) \\
&= \sum_{C^2=0}^{N-2} \sum_{Y^2} \sum_{\tilde{n}^2=2}^{t} \sum_{\tilde{n}^3=t+1}^{\infty} Prob(d^3 = \tilde{n}^3 - \tilde{n}^2 \mid y^2 = Y^2, ct^2 = C^2) k(\tilde{n}^2, Y^2, C^2)
\end{aligned}
$$

In general, we have the following:

$$
Prob(T(t) = p) = \begin{cases} Prob(\tilde{d^1} > t), & p = 0; \\ \sum_{C^p=0}^{N-2} \sum_{Y^p} \sum_{\tilde{n}^p=p}^{t} \sum_{\tilde{n}^{p+1}=t+1}^{\infty} & \\ Prob(d^{p+1} = \tilde{n}^{p+1} - \tilde{n}^p \mid y^p = Y^p, ct^p = C^p) k(\tilde{n}^p, Y^p, C^p), & 1 \le p \le t; \\ 0, & p > t. \end{cases} \tag{26}
$$

The expression, $k(\tilde{n}^p, Y^p,\ C^p)$, can be recursively computed for $p\geq 1$ as follows:

$$\begin{aligned} k(\tilde{n}^{p+1}, Y^{p+1},\ C^{p+1}) &= \sum_{C^p=0}^{N-2} \sum_{Y^p} \sum_{\tilde{n}^p=p}^{\tilde{n}^{p+1}-1} g(\tilde{n}^{p+1}, Y^{p+1}, C^{p+1}, Y^p, C^p, \tilde{n}^p) \\ &= \sum_{C^p=0}^{N-2} \sum_{Y^p} \sum_{\tilde{n}^p=p}^{\tilde{n}^{p+1}-1} f(\tilde{n}^{p+1} - \tilde{n}^p, Y^{p+1}, C^{p+1} \mid Y^p, C^p) k(\tilde{n}^p, Y^p, C^p) \end{aligned} \tag{27}$$

The corresponding expressions for the other schedulers are given as follows:

$$Prob(T(t) = p) = \begin{cases} Prob(\tilde{d^1} > t), & p = 0; \\ \sum_{C^p=0}^{N-2} \sum_{\tilde{n}^p=p}^{t} \sum_{\tilde{n}^{p+1}=t+1}^{\infty} & \\ Prob(d^{p+1} = \tilde{n}^{p+1} - \tilde{n}^p \mid ct^p = C^p) k(\tilde{n}^p, C^p), & 1 \leq p \leq t; \\ 0, & p > t. \end{cases} \tag{28}$$

$$k(\tilde{n}^{p+1},\ C^{p+1}) = \sum_{C^p=0}^{N-2} \sum_{\tilde{n}^p=p}^{\tilde{n}^{p+1}-1} f(\tilde{n}^{p+1} - \tilde{n}^p, C^{p+1} \mid C^p) k(\tilde{n}^p, C^p) \tag{29}$$

Having obtained the distribution of T(t), we can compute the corresponding metrics as a function of *t* for short-term fairness evaluation for the homogeneous and non-homogeneous cases as follows:

$$\begin{aligned} E[\Delta(t)] &= \frac{E[T_j(t)]}{t} - \frac{1}{N} \\ Var[\Delta(t)] &= \frac{Var[T_j(t)]}{t^2} \\ E[RFB(t)] &= \frac{1}{t} \left| \frac{E[T_j(t)]}{\frac{r^j}{R}} - \frac{E[T_k(t)]}{\frac{r^k}{R}} \right| \end{aligned} \tag{30}$$

D. Evaluation of QoS Parameters

In this section, we shall illustrate the evaluation of QoS parameters for flow *j* for a two-flow WFS with $\underline{r}$= [1,R-1]. Recall that the scheduling mechanism can be modeled as a two-dimensional Markov Chain given by $\{(x_i, count_i)$, i = 1,2,3...$\}$. If we observe these variables at the departure instances of flow *j* packets, as shown in Fig. 7, then $\{(y^p, ct^p)$,p = 1,2,3...$\}$ forms a two-dimensional Markov Chain. The state transition diagram is shown in Fig. 8.

1) Evaluation of long term performance metrics: Based on Section IV-B, the long term performance metrics for flow *j* can be derived from the steady-state delay distribution, $\bar{d^j}(n)$. We shall drop the subscript *j* for simplicity of notations. Hence, $\bar{d}(n)$ can be evaluated as follows:

$$\begin{aligned} \bar{d}(n) &= \lim_{p\to\infty} \bar{d^p}(n) \\ &= \lim_{p\to\infty} \sum_{Y^p, C^p} Prob(d^p = n,\ y^p = Y^p,\ ct^p = C^p) \end{aligned} \tag{31}$$

Since $\{(y^p, ct^p),\ p = 1, 2, 3...\}$ is a Markov Chain, $Prob(d^p = n,\ y^p = Y^p,\ ct^p = C^p),\ p \geq 1$ can be expressed recursively in terms of $f(n^p, Y^p, C^p \mid Y^{p-1}, C^{p-1}),\ p \geq 1$ as follows:

$$\begin{aligned} Prob(d^p = n,\ y^p = Y^p,\ ct^p = C^p) &= \sum_{Y^{p-1}, C^{p-1}} f(n, Y^p, C^p \mid Y^{p-1}, C^{p-1}) Prob(y^{p-1} = Y^{p-1},\ ct^{p-1} = C^{p-1}) \\ Prob(y^p = Y^p,\ ct^p = C^p) &= \sum_n Prob(d^p = n,\ y^p = Y^p,\ ct^p = C^p) \end{aligned} \quad (32)$$

where $Prob(y^0 = Y^0,\ ct^0 = C^0)$ can be obtained from Eq. (18). The limit in Eq. (31) exists since $\{(y^p, ct^p),\ p = 1, 2, 3...\}$ is ergodic. This can be verified by observing the state transition diagram in Fig. 8.

2) Evaluation of short term performance metrics: For short-term performance metrics, based on Eq. (26)-(27), we need to evaluate the following expressions:

$$k(\tilde{n}^1, Y^1, C^1)$$

$$Prob(d^{p+1} = n^{p+1} \mid y^p = Y^p, ct^p = C^p) \qquad p \geq 1$$

$$f(n^{p+1}, Y^{p+1}, C^{p+1} \mid Y^p, C^p) \qquad p \geq 1$$

$Prob(d^{p+1} = n^{p+1} \mid y^p = Y^p, ct^p = C^p)$ can be expressed in terms of $f(n^{p+1}, Y^{p+1}, C^{p+1} \mid Y^p, C^p)$ as follows:

$$Prob(d^{p+1} = n^{p+1} \mid y^p = Y^p, ct^p = C^p) = \sum_{Y^{p+1}} \sum_{C^{p+1}=0}^{R-1} f(n^{p+1}, Y^{p+1}, C^{p+1} \mid Y^p, C^p) \quad (33)$$

$k(\tilde{n}^1, Y^1, C^1)$ can be expressed in terms of $f(n^1, Y^1, C^1 \mid Y^0, C^0)$ as follows:

$$\begin{aligned} k(\tilde{n}^1, Y^1, C^1) &= Prob(\tilde{d}^1 = \tilde{n}^1, y^1 = Y^1,\ ct^1 = C^1) \\ &= Prob(d^1 = \tilde{n}^1, y^1 = Y^1,\ ct^1 = C^1) \\ &= \sum_{Y^0} \sum_{C^0=0}^{R-1} Prob(d^1 = \tilde{n}^1, y^1 = Y^1,\ ct^1 = C^1 \mid y^0 = Y^0,\ ct^0 = C^0) Prob(y^0 = Y^0,\ ct^0 = C^0) \\ &= \sum_{Y^0} \sum_{C^0=0}^{R-1} f(\tilde{n}^1, Y^1, C^1 \mid Y^0, C^0) Prob(y^0 = Y^0,\ ct^0 = C^0) \end{aligned} \quad (34)$$

where $Prob(y^0 = Y^0,\ ct^0 = C^0)$ can be obtained from Eq. (18).

3) Expressions for $f(n^p, Y^p, C^p \mid Y^{p-1}, C^{p-1}),\ p \geq 1$: From Eq. (31)-(34), we note that both the long term and short term performance metrics can be expressed in terms of $f(n^p, Y^p, C^p \mid Y^{p-1}, C^{p-1}),\ p \geq 1$. We can obtain expressions for the latter in terms of transmission events. We begin with the case of *p*=1. According to Eq. (18), Y^0=0 and C^0 is uniformly distributed in the interval [0,R-1].

Let us assume that C^0=0. Since flow *j* can only transmit once over the interval [0,n^1], according to Eq. (8), $Y^1 \leq R-1$. For each possible (Y^1, C^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. In addition to the list of possible transmission events, E_i, defined in Eq. (5), we define the following notations to denote the transmission events over an interval of slots beginning with slot i :

$[S1^a S2^{N-a}]$	Over an interval of N slots, in slots $i, i+1 ... i+a$, flow 1 is allocated and NF occurs; in the remaining slots, flow 2 is allocated and f2S2 ∪ NF occurs
E^r	Event E occurs over slots i,i+1,i+2,...i+r-1
$(S1^a S2^{N-a})$	Over an interval of N slots, flow 1 is allocated and NF occurs in *a* slots; in the remaining slots, flow 2 is allocated and f2S2 ∪ NF occurs

Since $Y^1 < 0$ is possible only if the event f2S1 occurs at least once, we consider two cases:

(a) Event f2S1 never occurs

In this case, $0 \leq Y^1 \leq R-1$. If the event f1S2 occurs, we have $Y^1 = R-1$; otherwise, Y^1=0. Hence, we obtain the following expressions:

$$\begin{aligned} f(n^1, Y^1 = 0, C^1 = 1 \mid C^0 = 0) &= Prob([S1S2^{R-1}]^{\frac{n-1}{R}} f1S1), \quad modulo(n-1, R) = 0 \\ f(n^1, Y^1 = R-1, C^1 = k-1 \mid C^0 = 0) &= Prob([S1S2^{R-1}]^{\frac{n-k}{R}} S1S2^{k-2} f1S2), \quad modulo(n-k, R) = 0, \\ & \quad 2 \leq k \leq R \end{aligned} \tag{35}$$

(b) Event f2S1 occurs at least once

In this case, $Y^1 < R-1$, and we have the following expressions:

$$\begin{aligned} f(n^1, Y^1 \neq 0, C^1 = 0 \mid C^0 = 0) &= \sum_{q=1}^{n+Y^1-R+1} Prob([S1S2^{R-1}]^{\frac{q-1}{R}} f2S1(f2S1^{R-Y^1-2} S1^{n+Y^1-R-q+1}) f1S1) \\ f(n^1, Y^1 = 0, C^1 = 1 \mid C^0 = 0) &= \sum_{q=1}^{n-R+1} Prob([S1S2^{R-1}]^{\frac{q-1}{R}} f2S1(f2S1^{R-2} S1^{n-R-q+1}) f1S1) \end{aligned} \tag{36}$$

The corresponding expressions for other values of C^0 can be obtained in a similar manner.

4) Evaluation of $f(n^p, Y^p, C^p \mid Y^{p-1}, C^{p-1}),\ p \geq 1$ with perfect channel prediction: The above expressions can be evaluated by expressing the probability of occurrence of each type of event in terms of the channel characteristics. For two-flow scheduling, we define a generic 4×4 matrix, $\underline{\mathbf{G}}$, to denote the conditional probabilities of all possible permutations of channel states in the slot pair *(i-1,i)* as follows:

$$\underline{\mathbf{G}} = \begin{bmatrix} GG\_GG & GG\_GB & GG\_BG & GG\_BB \\ GB\_GG & GB\_GB & GB\_BG & GB\_BB \\ BG\_GG & BG\_GB & BG\_BG & BG\_BB \\ BB\_GG & BB\_GB & BB\_BG & BB\_BB \end{bmatrix} \tag{37}$$

For example, the entry BB_GB denotes the probability that $c_i^1 = Good$ and $c_i^2 = Bad$, given that $c_{i-1}^1 = c_{i-1}^2 = Bad$. For our two-state Markov Chain error model, since the channel of both flows are assumed to be independent, $EE\_CE = p_{eg}^1 \times (1 - p_{eg}^2)$.

The probability of occurence of each event defined in Eq. (5) can be expressed in terms of the product of a constant matrix and $\underline{\mathbf{G}}$. For example, consider the event f2S1. For this event to occur in slot *i*, $c_i^1 = Bad$ and

$c_i^2 = Good$. With perfect channel prediction, this corresponds to the entries in the third column of **G**. Hence, we can express the probability of event f2S1 as a 4×4 matrix as follows:

$$Prob(f2S1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \underline{\mathbf{G}} \qquad (38)$$

By defining the corresponding matrices for all possible events, the expressions in Eq. (35) and (36) can be evaluated. In general, for N-flow scheduling, since there are 2^N permutations of channel states, the matrix **G** is of dimension $2^N \times 2^N$.

5) Evaluation of $f(n^p, Y^p, C^p \mid Y^{p-1}, C^{p-1}),\ p \geq 1$ *with imperfect channel 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 one-step channel prediction (as defined in Section II-B.2) on the QoS performance of the WFS and EFF scheduler. We denote these schedulers as WFS-OSP and EFF-OSP respectively, as opposed to WFS-PCK and EFF-PCK which assumes perfect channel prediction.

Let us consider the probability of occurrence of the event f2S1, as in Section IV-D.4, but with one-step prediction. For this event to occur in slot *i*, $c_i^1 = Bad$ and $c_i^2 = Good$. In addition, for flow 2 to qualify for transmission in slot *i*, $c_{i-1}^2 = Good$. Hence, the relevant entries in **G** are GG_BG and BG_BG, and therefore, we can express the probability of event f2S1 as a 4×4 matrix as follows:

$$Prob(f2S1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \underline{\mathbf{G}} \qquad (39)$$

V. PERFORMANCE EVALUATION OF WIRELESS SCHEDULERS

In Section IV, we have shown that the QoS performance metrics for each wireless scheduler can be evaluated in terms of the channel parameters, $\underline{p}_{ge}$ and $\underline{p}_{corr}$, where $\underline{p}_{ge} = [p_{ge}^1, p_{ge}^2, ...p_{ge}^N]$ and $\underline{p}_{corr} = [p_{corr}^1, p_{corr}^2, ...p_{corr}^N]$.

In this section, we present some numerical results to compare the QoS provisioning capabilities of WFS and its variants for a N-flow homogeneous scheduling (i.e., r^j=1) scenario, where

(a) each flow carries delay-sensitive traffic

(b) the scheduling environment is homogeneous to all flows, i.e., $p_{ge}^j = p_{ge}$ and $p_{corr}^j = p_{corr}$

An important example of such a scenario is radio broadcasting via the Internet to a group of N mobile users linked to a common access point, and each user may be tuned to different radio stations. Hence, all users have equal demands in terms of time-fraction requirement. In addition, each user is limited in terms of receiver buffer capacity, and hence, the traffic to each user is sensitive to delay as well as delay jitter.

A. N=2

1) Long Term Performance: Based on the delay density function, $\bar{d}(n)$, we can write the delay distribution function, $\bar{D}(n)$, as follows:

$$\bar{D}(n) = \sum_{k=1}^{n} \bar{d}(k)$$

For a delay-sensitive flow with delay bound, n^{max}, $\beta_{n^{max}} = 1 - \bar{D}(n^{max})$ denotes the probability of delay bound violation. This can also be interpreted as the packet loss rate if the scheduler 'drops' packets whose delay bounds are exceeded. We plot $\beta_{n^{max}}$ as a function of n^{max} for p_B=0.01 in Fig. 9. The corresponding plots for p_B=0.05 are shown in Fig. 10.

When the channel is uncorrelated (i.e., p_{corr}=1.0), we notice a significant degradation in delay bound violation probability in the channel-dependent schedulers (i.e., EFF scheduler and WFS) as a result of channel prediction, with a more severe degradation in the WFS. Under such circumstances, the channel-independent schedulers (i.e., WRR and FA schedulers) offer superior delay performance, with the FA scheduler offering the best performance.

As the channel correlation increases (i.e., as p_{corr} is reduced), the level of degradation as a result of one-step prediction is reduced. However, the channel-independent schedulers still upper-bounds the performance of channel-dependent schedulers. At even higher levels of channel correlation, we expect the performance of the WRR scheduler to improve relative to the FA scheduler. In addition, the performance of channel-dependent schedulers is expected to be tightly upper-bounded by the WRR scheduler.

Next, we plot the delay jitter, σ_d, and throughput, μ_S, as a function of p_{corr} for p_B=0.01 in Fig. 11. The corresponding plots for p_B=0.05 are shown in Fig. 12. Although channel prediction also results in throughput degradation in channel-dependent schedulers, the throughput degradation is relatively less significant compared to that incurred in the delay performance. Hence, the throughput achieved with WFS-OSP and WFS-EFF is still significantly higher than channel-independent schedulers. However, the superior throughput performance in channel-dependent schedulers achieved through flow swapping is traded-off with poorer delay jitter performance relative to channel-independent schedulers.

2) Short Term Performance: We plot $Var[\Delta(t)]$ as a function of t for p_B=0.01 in Fig. 13. The corresponding plots for p_B=0.05 are shown in Fig. 14.

The EFF schedulers perform worst in terms of short-term fairness as expected since there is no mechanism to ensure fairness. The WFS improves the fairness performance of EFF schedulers via the Fairness Monitor. The channel-independent schedulers maintain short-term fairness via the error-free fair scheduling mechanism.

B. N>2

In Section III, we described the approach for analysis and modeling of wireless schedulers for the case of N=2. It was shown that except for the WFS, the wireless schedulers can be modeled by a one-dimensional Markov Chain. The same one-dimensional Markov Chain extends to the case of N>2, and the QoS metrics can be evaluated in terms of products of $2^N \times 2^N$ matrices, since there are 2^N permutations of channel states.

For the WFS, the dimensions of the Markov Chain increases as N increases. In general, we need to define a vector x of length $\frac{N\times(N-1)}{2}$ to characterize the FM and a single variable, count, to characterize the error-free scheduling mechanism. Hence, we require a $\frac{N\times(N-1)}{2}$+1-dimensional Markov Chain to model a N-flow WFS, and analysis of such a model is complex.

However, for N=2, we observed that the delay violation probability achieved by WFS-OSP can be bounded by the WRR scheduler and the EFF-OSP scheduler. Hence, we will use the performance of WRR and EFF-OSP to bound the performance of WFS-OSP for N>2. In addition, since the relative performance of the schedulers are consistent for both error rates considered, we will only present results obtained for p_B=0.01. Results for long term performance are shown in Fig. 15 to 18.

For N>2, 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 N=2. In addition, as N increases, the gain in delay performance achieved by channel-independent schedulers over EFF-OSP is increased. However, we note the following:

(a) when the channel is uncorrelated, the gain in throughput by the EFF-OSP scheduler over the FA and WRR schedulers is invariant with N;

(b) when the channel is correlated, the gain in throughput by the EFF-OSP scheduler over the WRR scheduler is invariant with N but is improved over the FA scheduler.

Results for short term performance are shown in Fig. 19 and 20. We observe the same performance trends as in the case of N=2. However, as N increases, the performance gap amongst all the schedulers is reduced.

VI. CONCLUSIONS

In this work, we presented a Markov-modeling approach for an N-flow Wireless-Fair Scheduler (WFS), and illustrated the approach for the case of N=2. Based on this Markov Model, we defined various QoS performance metrics in terms of throughput, delay and fairness over different time scales. In order to assess the contribution to different QoS metrics of different components of the WFS, we defined and modeled various abstractions of the WFS in a similar manner.

In order to compare the QoS provisioning capability of WFS and its variants, we presented some numerical results for the cases of N=2, N=3 and N=4. We observe that no single scheduler achieves the best performance in terms of all the performance metrics under all channel conditions. In particular, superior long term performance in terms of delay and delay jitter is achieved by channel-independent schedulers at the expense of poorer throughput performance with respect to channel-dependent schedulers. On the other hand, WFS achieves better short-term fairness performance than channel-independent schedulers.

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 an adaptive scheduler that performs FA scheduling when the channel correlation is low, but switches to WRR scheduling when the channel correlation increases.

REFERENCES

[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, 1993.

[3] H. Sariowan and R. L. Cruz, "SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 669–684, October 1999.

[4] K. Lee, "Performance bounds in communication networks with variable-rate links," in *ACM SIGCOMM '95*, August 1995, pp. 126–136.

[5] R. L. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. Information Theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[6] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Transanctions on Networking*, vol. 1, no. 3, pp. 372–385, 1993.

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

[8] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity," *IEEE Trans. Information Theory*, vol. 39, no. 2, pp. 466–478, March 1993.

[9] 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.

[10] 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.

[11] M. Reisslein, K. Ross, and S. Rajagopal, "A Framework for Guaranteeing Statistical QoS," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 27–42, February 2002.

[12] X. Liu, Edwin K.P. Chan, and Ness B. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2064, October 2001.

[13] R. Rom and H. P. Tan, "Performance Tradeoffs in Wireless Scheduling with Flow Aggregation," *To appear in Proc. of the IEEE WCNC*, March 2003.

[14] 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, Sept 1998.

[15] 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.

[16] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *PROCEEDINGS OF IEEE INFOCOM '94*, June 1994, pp. 636–646.

[17] 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, no. 5, pp. 690–704, October 1997.

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

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

[20] 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.

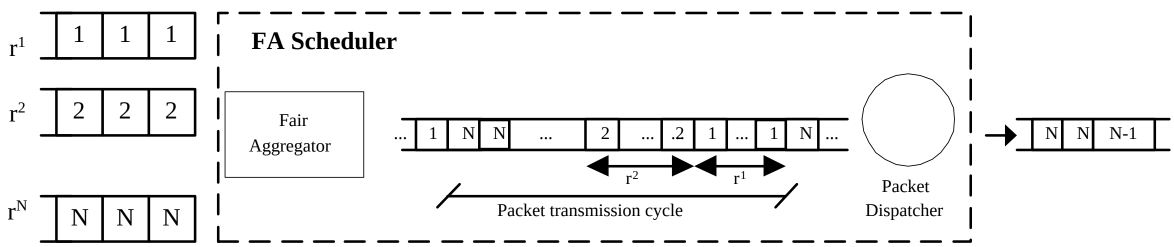


Fig. 5. Components of FA Scheduler

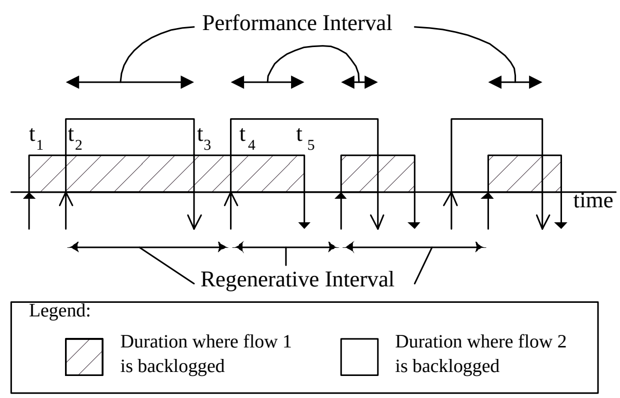


Fig. 6. Queue Status for Two-Flow Scheduling Scenario

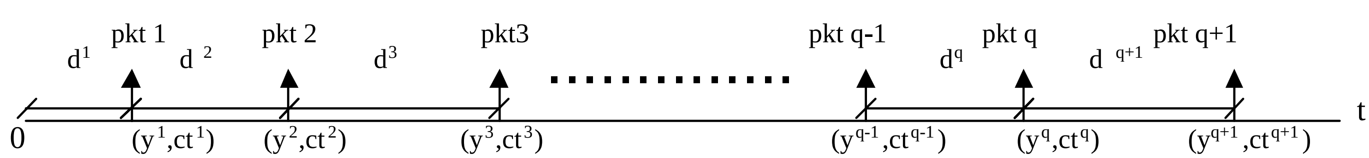


Fig. 7. Computation of probability distribution of T(t)

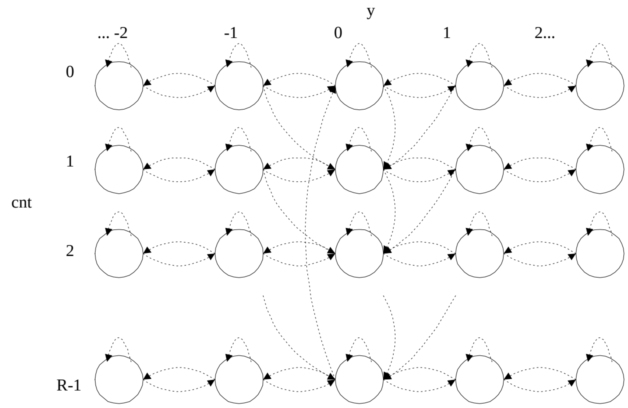
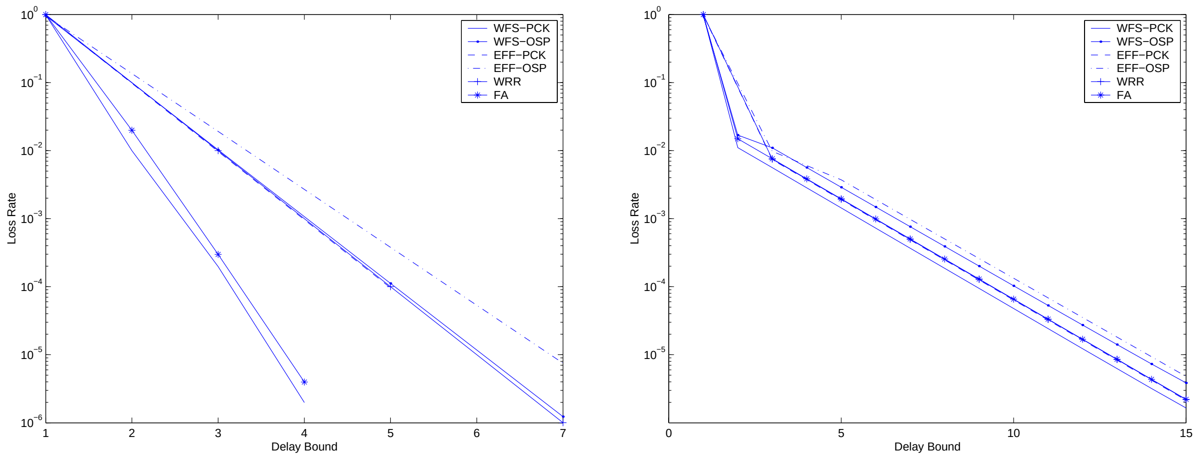
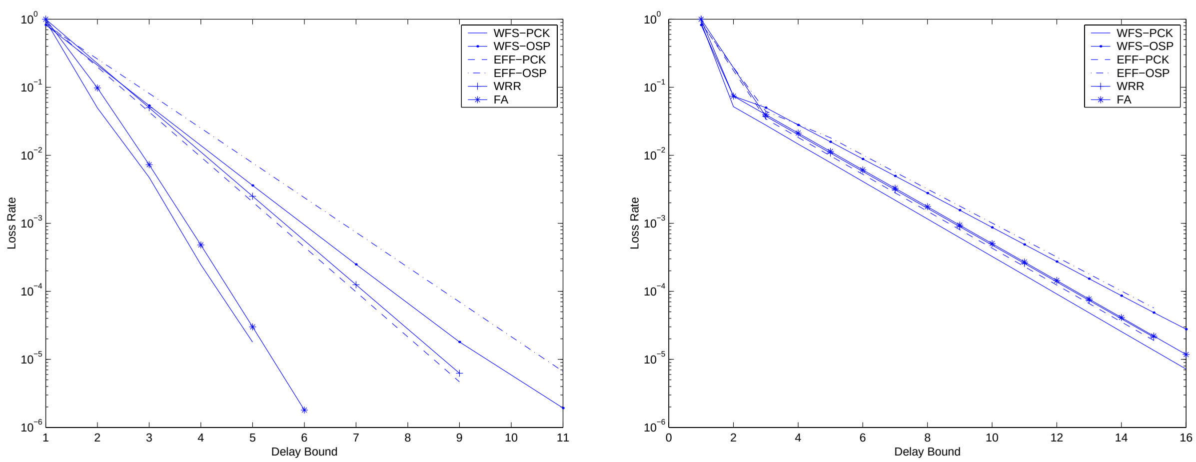


Fig. 8. State Transition Diagram for two-flow WFS-PCK with $\underline{r}$= [1,R-1]



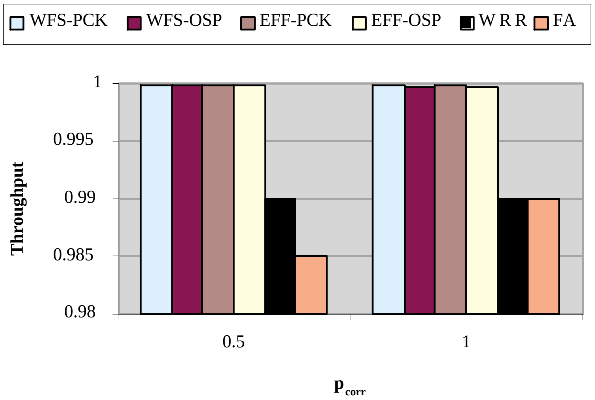
(a) (b)

Fig. 9. Delay violation probability for N=2,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5

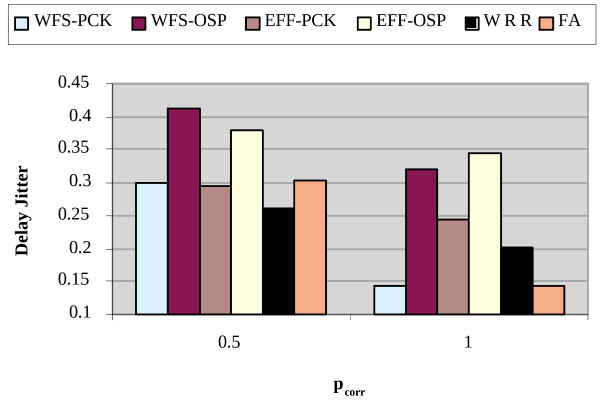


(a) (b)

Fig. 10. Delay violation probability for N=2,p_B=0.05, p_{corr}=(a) 1.0 (b) 0.5

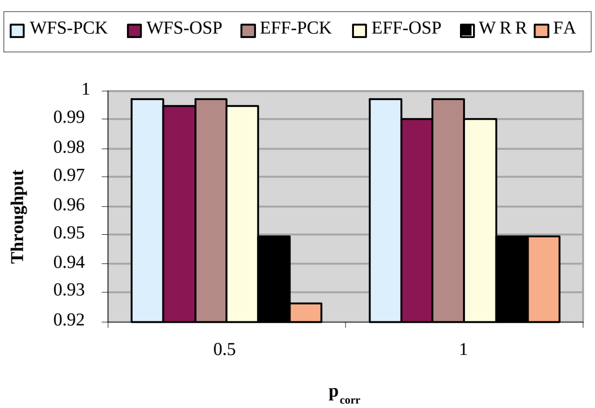


(a)

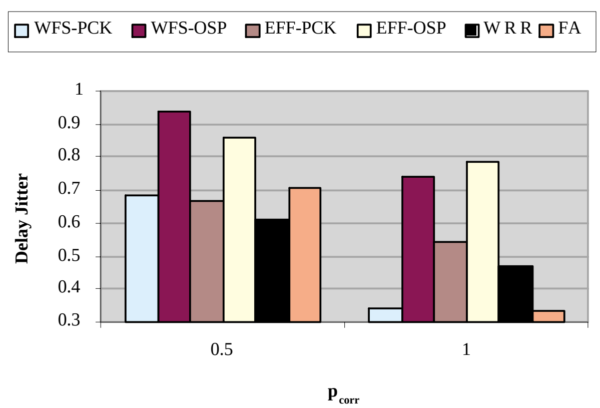


(b)

Fig. 11. (a) Throughput and (b) Delay Jitter for N=2,p_B=0.01

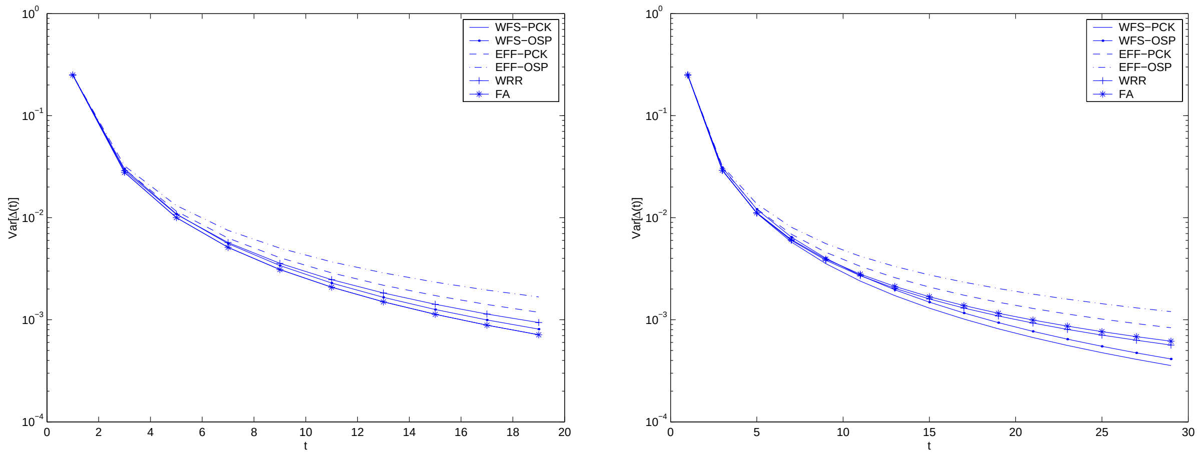


(a)



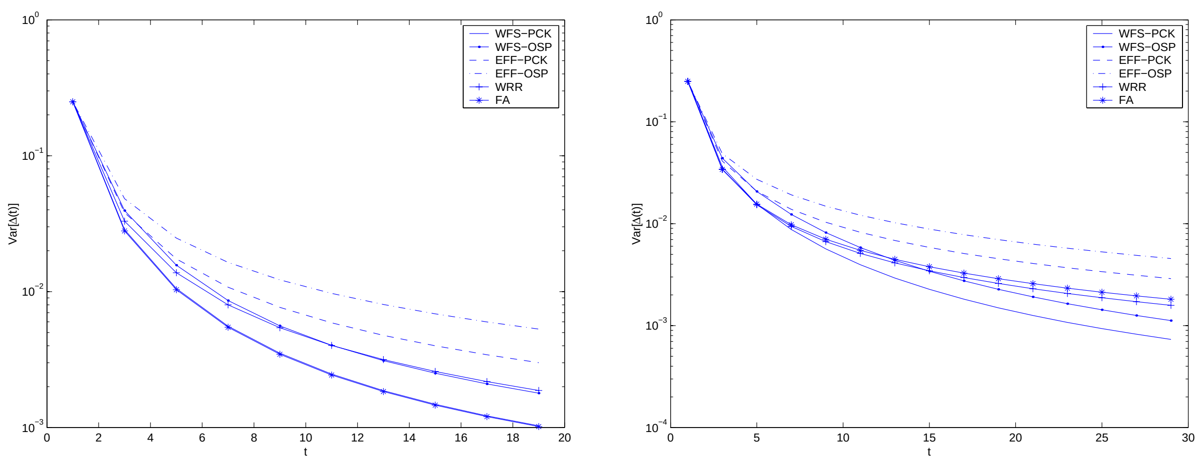
(b)

Fig. 12. (a) Throughput and (b) Delay Jitter for N=2,p_B=0.05



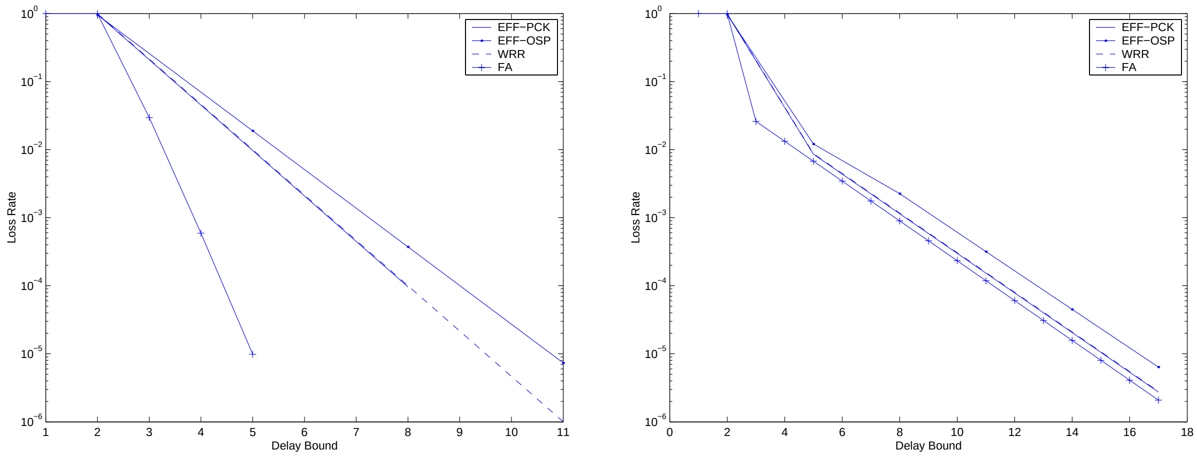
(a) (b)

Fig. 13. Throughput fluctuation as a function of t for N=2,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5



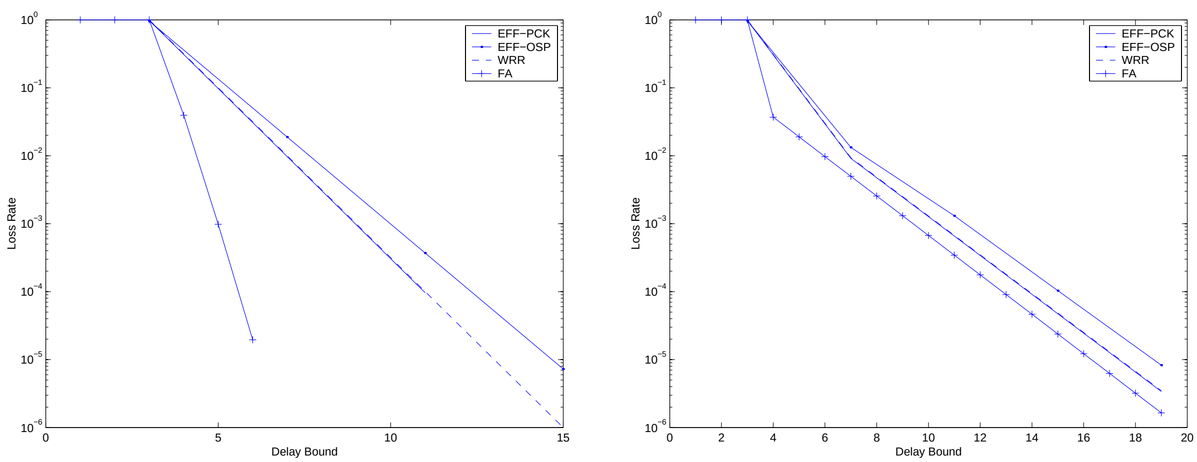
(a) (b)

Fig. 14. Throughput fluctuation as a function of t for N=2,p_B=0.05, p_{corr}=(a) 1.0 (b) 0.5



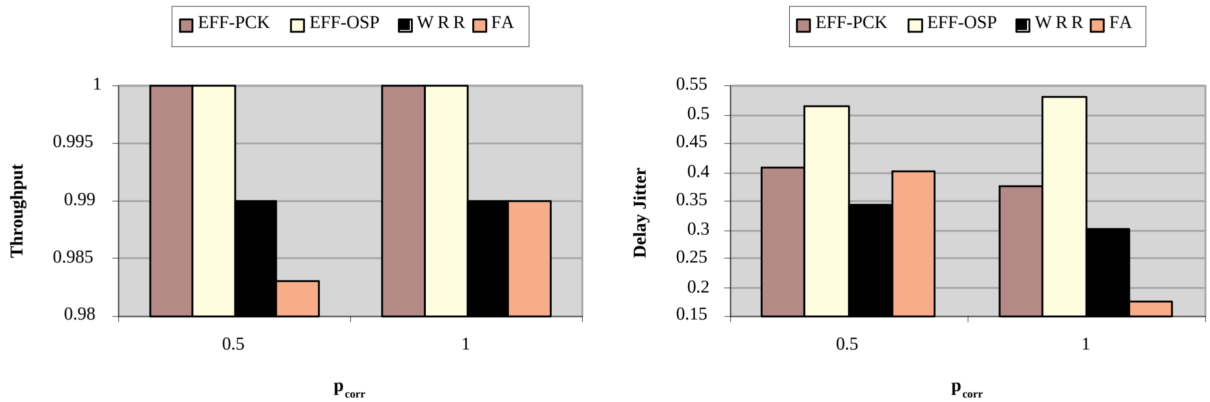
(a) (b)

Fig. 15. Delay violation probability for N=3,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5



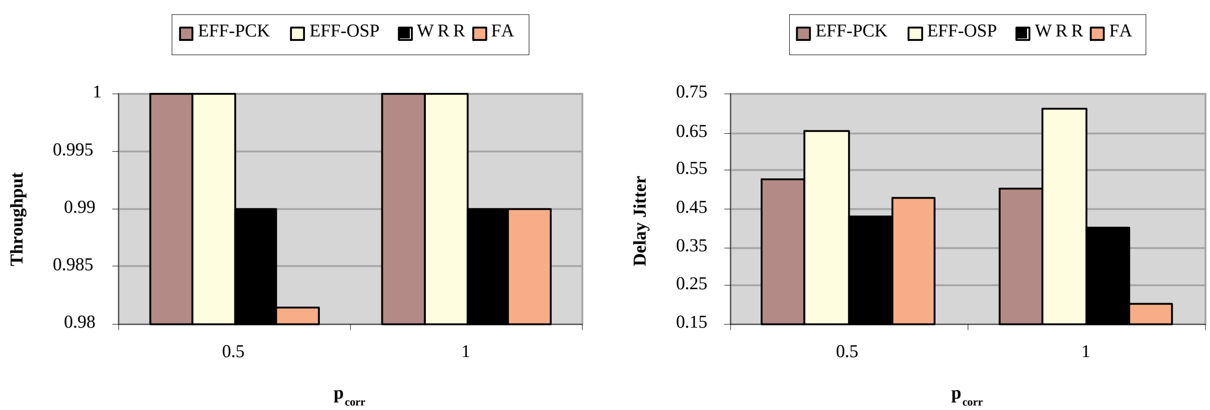
(a) (b)

Fig. 16. Delay violation probability for N=4,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5



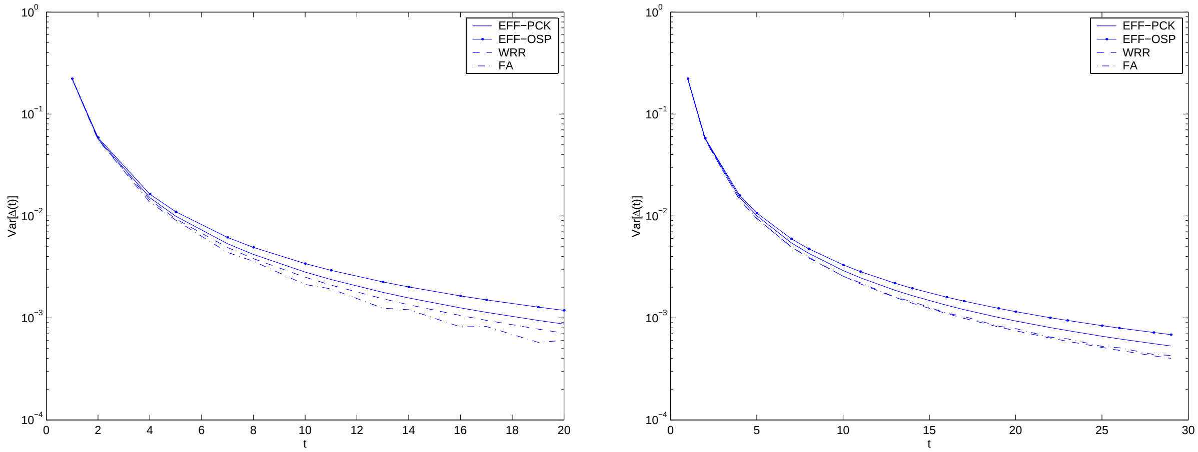
(a) (b)

Fig. 17. (a) Throughput and (b) Delay Jitter for N=3,p_B=0.01



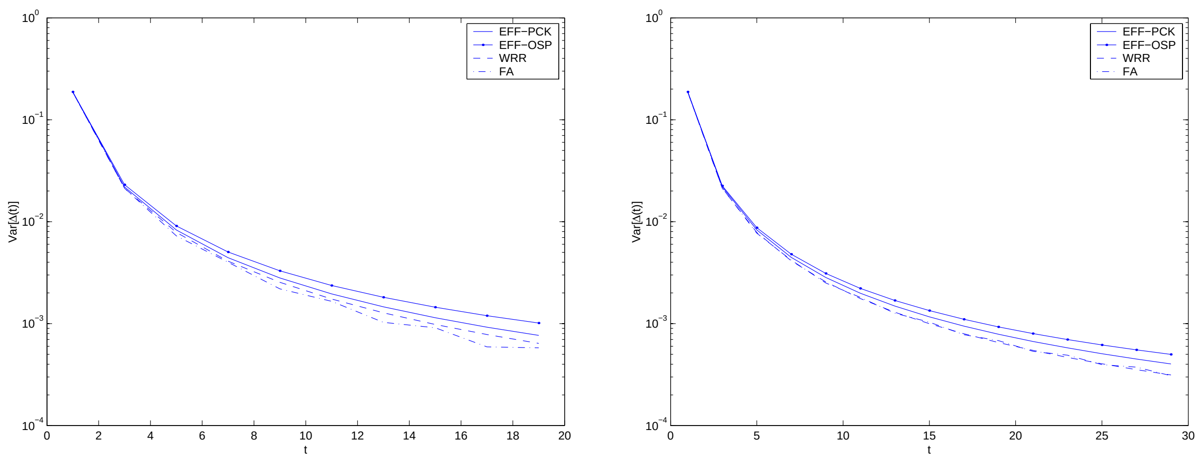
(a) (b)

Fig. 18. (a) Throughput and (b) Delay Jitter for N=4,p_B=0.01



(a) (b)

Fig. 19. Throughput fluctuation as a function of t for N=3,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5



(a) (b)

Fig. 20. Throughput fluctuation as a function of t for N=4,p_B=0.01, p_{corr}=(a) 1.0 (b) 0.5