CCIT Report # 470 February 2004

Performance Analysis of a Recursive Cyclic Scheduler for Class-based Scheduling

Raphael Rom, Moshe Sidi and Hwee Pink Tan
Department of Electrical Engineering
Technion, Israel Institute of Technology
Technion City, Haifa 32000, Israel
Email: {rom, moshe}@ee.technion.ac.il and hweepink@tx.technion.ac.il

Abstract

In this paper, we consider the problem of determining a cyclic (or loop) scheduler that allocates slots to flows as periodically as possible. We use the second moment of the inter-allocation distance for each flow as its periodicity metric. We establish the optimality of a Weighted-Round Robin with spreading (*WRR*-sp_2) scheduler for a two-flow scenario.

We consider a class-based scheduling scenario where flows are grouped according to their relative bandwidth demands. We propose a *C*-class scheduler that recursively performs inter-class scheduling using the corresponding *C*-1 class scheduler, prior to intra-class scheduling. Optimality is achieved for *C* = 2 with the *WRR*-sp_2 as the inter-class scheduler.

Through numerical results, we show that the recursive scheduler achieves the best periodicity performance at the expense of intra-class fairness, which is desirable for class-based scheduling. Thereby, we expose a trade-off between periodicity and fairness performance in the design of loop schedulers.

I. Introduction

Consider a system that comprises an indivisible resource (time-slot) and *n* clients (or flows) share it by means of time multiplexing: in any given time, a different flow may use the resource. Many applications require that flows are served at some prescribed rate, and this rate should be as smooth as possible even in small time windows. The allocation of time slots to flows is governed by a scheduling algorithm. In other words, given a set of requested shares, $\{x^{(i)}\}_{i=1}^{n}$, the goal of the scheduling algorithm is to produce an assignment of time slots (or a schedule) to flows, while trying to optimize two different measures:

(a) **Fairness**: a schedule is said to have good fairness if the fraction of time slots allocated to each flow is close to its requested share;

(b) **Smoothness**: a schedule is said to have good smoothness if the time slots allocated to each flow are as evenly spaced as possible.

The best possible schedule is one where the allocated shares are exactly the requested shares (perfect fairness) and where each flow is scheduled exactly every *p* time slots (perfectly-periodic schedule). Although schedules that offer fairness while neglecting smoothness are available [1], it is NP-complete to decide whether a set of requests admits a perfectly-periodic schedule [2].

A. Perfectly Periodic Scheduling

Two approaches to the scheduling problem are considered in the literature. The first approach insists on maintaining strict smoothness while relaxing the fairness requirement. In [3] (and references therein), each flow *i* requests that it be scheduled exactly every $\tau^{(i)}$ time slots, and the goal is to determine a scheduler that optimizes the fairness measure under the perfect periodicity constraint. Strict smoothness requirements imply that the periods allocated to some flows will not match their requests. A suitable metric to measure the deviation from perfect fairness for each flow is the fairness ratio, given by the ratio of its requested period and its granted period. There exist schedulers [4] that guarantee that the average fairness ratio (where the weight of each flow is its requested bandwidth) is close to optimal. The maximum measure is studied in [5], where the quality of the schedule is the worst-case fairness ratio over all flows.

B. Non-Periodic Scheduling

An alternative approach is to allow different gaps between consecutive allocations to a flow, while insisting on perfect fairness. This approach was considered in [6] (and references therein), where the authors considered an *online* variant of the resource sharing 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 [7], the author deduced that for throughput optimality for *n*=2 and unit buffer size per queue, the schedule must be *open-loop* (or de-centralized) and *conflict-free*. This work was extended in [8] to the case of *n*>2. It was also verified that an optimal schedule always exists and is stationary and *cyclic* (or loop), i.e., there exists an *N* such that for all *t*, the flow allocated to slot τ is also allocated to slot τ+*N*.

This reduces the problem to an *offline* one, where the objective is to determine a loop schedule of size N in terms of $\{x^{(i)}\}_{i=1}^{n}$, where $\{x^{(i)}\}_{i=1}^{n}$ is computed in terms of the arrival statistics. The authors proposed a Golden Ratio Scheduler that achieves a nearly optimal throughput under online conditions. In [6], the authors considered the case where the buffer size per queue 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.

C. Contribution of This Paper

Although other loop schedulers that ensure perfect approximation have been proposed in the literature, we are not aware of any work that analyzed and evaluated the extent of non-periodicity in these schedulers. In this paper, we propose a periodicity metric and compute the bounds for the metric over the class of loop schedulers. By analyzing the periodicity properties of various loop schedulers, we propose a recursive class-based scheduler that performs inter-class scheduling followed by intra-class scheduling for a class-based scheduling scenario. We demonstrate the gain in periodicity performance achieved by the recursive scheduler over known loop schedulers.

The paper is organized as follows: We define our scheduling problem and define a metric to evaluate the periodicity characteristics of each scheduler in Section II. Then, we describe the mechanism and periodicity properties of several loop schedulers in Section III. We consider the requirements for optimal per-flow periodicity in loop schedulers and establish the lower bound for the periodicity metric in Section IV. In Section V, we define a class-based scheduling scenario and establish an optimal scheduler for two-class scheduling. Based on this scheduler, we propose a recursive class-based scheduler to handle multiple classes. In Section VI, we compare the performance of various loop schedulers in terms of numerical results. Finally, some concluding remarks are given in Section VII.

II. Problem Formulation

Consider a slot allocation problem amongst *n* flows, where each flow *i* demands a share of $x^{(i)}$, where $\sum_{i=1}^{n} x^{(i)} = 1$. Let us use the notation $\underline{y}$ to denote the set $\{y^{(i)}\}_{i=1}^{n}$. We define $\pi_{(N,\underline{x})}$ as the class of *n*-flow loop schedulers that satisfy the perfect fairness constraint over any interval of *N* slots, where *N* is the cycle length. This implies that (for any finite *n*), the elements of $\underline{x}$ are rational. We can then define *N* and $N^{(i)}$, $1 \leq i \leq n$, as follows:

$$
\begin{aligned}
N &= LCD(\underline{x}) \\
N^{(i)} &= x^{(i)} N
\end{aligned}
$$

where $N^{(i)}$ is the number of slots allocated to flow *i* over any interval of *N* slots.

Let $d_\pi^{(i)}(j)$ denote the duration between the $(j-1)^{th}$ and j^{th} allocation to flow *i* under a scheduler $\pi \in \pi_{(N,\underline{x})}$. Since π is a loop scheduler, the following properties must be satisfied for $1 \leq i \leq n$:

$$
\begin{aligned}
d_\pi^{(i)}(N^{(i)}+j) &= d_\pi^{(i)}(j) \\
\sum_{j=k}^{k+N^{(i)}-1} d_\pi^{(i)}(j) &= N
\end{aligned}
\tag{1}
$$

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

$$
E[\underline{d}_\pi^{(i)}]^m = \frac{\sum_{j=1}^{N^{(i)}} [d_\pi^{(i)}(j)]^m}{N^{(i)}}
$$

However, from Eq. (1), we have the following:

$$
\begin{aligned}
E[\underline{d}_\pi^{(i)}] &= \frac{\sum_{j=1}^{N^{(i)}} d_\pi^{(i)}(j)}{N^{(i)}} \\
&= \frac{N}{N^{(i)}} \quad \text{independent of } \pi
\end{aligned}
$$

Hence, the periodicity measure for any scheduler π with respect to flow *i* can be evaluated in terms of $E[\underline{d}_\pi^{(i)}]^2$ instead of $Var[\underline{d}_\pi^{(i)}]$: a smaller value of $E[\underline{d}_\pi^{(i)}]^2$ implies a more periodic slot allocation to flow *i* and vice versa. We note that since the order of the elements in $\underline{d}_\pi^{(i)}$ is unimportant for the evaluation of $E[\underline{d}_\pi^{(i)}]^2$, we can consider $\underline{d}_\pi^{(i)}$ as a set of $N^{(i)}$ elements instead of an ordered sequence.

If we define the allocation vector $\underline{f}_\pi$ of length *N* such that the t^{th} element, $\underline{f}_\pi(t)$, denotes the flow that slot *t* (where $1 \leq t \leq N$) is allocated to under scheduler π, then our scheduling problem can be formulated as follows:

***n*-flow Scheduling Problem**

Determine the allocation vector $\underline{f}_{\pi^*}$ such that for $1 \le i \le n$

$$E[\underline{d}_{\pi^*}^{(i)}]^2 = \min_{\pi \in \pi_{(N,\underline{x})}} E[\underline{d}_{\pi}^{(i)}]^2$$

If $\underline{\mathbf{F}} = \{\underline{f}_{\pi} : \pi \in \pi_{N,\underline{x}}\}$, then

$$|\underline{\mathbf{F}}| = \frac{N!}{\prod_{j=1}^{n} N^{(j)}!}$$

We note that a very large number of $\underline{f} \in \underline{\mathbf{F}}$ are equivalent since they are identical under rotation. However, even after eliminating these, the resultant space is still non-tractable for large *N*.

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 consider various known loop schedulers and evaluate their periodicity performance against a lower bound, which we shall derive.

III. Description of *n*-flow Loop Schedulers

In this section, we will describe the mechanism as well as the periodicity characteristics of several loop schedulers. Without loss of generality, we will assume that $N^{(i)} \le N^{(j)}$ for *i*<*j* and $N^{(i)} \ge 2$. The case of $N^{(i)}$=1 is trivial since $\underline{d}_{\pi}^{(i)}$ = *N* for $\pi \in \pi_{(N,\underline{x})}$, i.e., perfect periodicity is always achieved for flow *i*. We denote by $ns_{\pi}^{(i,j)}(m)$ the cumulative number of slots allocated to flow *j* up to the m^{th} allocation to flow *i* by scheduler π.

A. n-flow Deficit Round Robin Scheduler (DRR_n)

Fair queueing schedulers like Weighted-Fair Queueing (WFQ) achieve nearly perfect fairness, but they are usually expensive to implement. DRR_n [9] 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_n scheduler reduces to the Weighted Round Robin (WRR) policy, which simply allocates a block of $N^{(1)}$ slots to flow 1 followed by a block of $N^{(2)}$ slots to flow 2 and so on. Hence, each flow *i* is allocated slots in blocks of size $N^{(i)}$, with an interval of *N*-$N^{(i)}$ slots between successive blocks. Therefore, we have the following:

$$\begin{aligned} \underline{d}_{DRR_n}^{(i)} &= \{1, \cdots, 1, N - N^{(i)} + 1\} \\ E[\underline{d}_{DRR_n}^{(i)}]^2 &= \frac{N^{(i)} + (N - N^{(i)})^2 + 2(N - N^{(i)})}{N^{(i)}} \end{aligned} \tag{2}$$

The DRR_n scheduler possesses the following property:

Lemma 1: The DRR_n scheduler exhibits the *worst* periodicity amongst the class of loop schedulers, $\pi_{(N,\underline{x})}$, i.e., for $1 \le i \le n$,

$$E[\underline{d}_{DRR_n}^{(i)}]^2 = \max_{\pi \in \pi_{(N,\underline{x})}} E[\underline{d}_{\pi}^{(i)}]^2$$

Proof: Let us consider an arbitrary scheduler $\pi \in \pi_{(N,\underline{x})}$ with $\underline{d}_{\pi}^{(i)}$ given as follows:

$$\underline{d}_{\pi}^{(i)} = \{1 + z_1, 1 + z_2, \cdots, 1 + z_{N^{(i)}-1}, N - N^{(i)} + 1 - \sum_{j=1}^{N^{(i)}-1} z_j\}$$

where $z_j \in Z+$, $1 \le j \le N^{(i)}$-1. We note that for z_j=0, $1 \le j \le N^{(i)}$-1, π = DRR_n.

Using Eq. (2), $E[\underline{d}_{\pi}^{(i)}]^2$ can be expressed in terms of $E[\underline{d}_{DRR_n}^{(i)}]^2$ as follows:

$$E[\underline{d}_{\pi}^{(i)}]^2 = E[\underline{d}_{DRR_n}^{(i)}]^2 + \frac{\sum_{j=1}^{N^{(i)}-1} z_j^2 + [\sum_{j=1}^{N^{(i)}-1} z_j]^2 - 2(N - N^{(i)}) \sum_{j=1}^{N^{(i)}-1} z_j}{N^{(i)}} \tag{3}$$

Since $\underline{d}_{\pi}^{(i)}$ correspond to inter-allocation intervals, we have the following constraint:

$$N - N^{(i)} + 1 - \sum_{j=1}^{N^{(i)}-1} z_j \ge 1$$

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

$$\sum_{j=1}^{N^{(i)}-1} z_j^2 \quad \leq \quad [\sum_{j=1}^{N^{(i)}-1} z_j]^2$$

Substituting into Eq. (3), we have the following:

$$\begin{aligned} E[\underline{d}_{\pi}^{(i)}]^2 &\leq E[\underline{d}_{DRR_n}^{(i)}]^2 + \frac{\sum_{j=1}^{N^{(i)}-1} z_j^2 + [\sum_{j=1}^{N^{(i)}-1} z_j]^2 - 2[\sum_{j=1}^{N^{(i)}-1} z_j]^2}{N^{(i)}} \\ &= E[\underline{d}_{DRR_n}^{(i)}]^2 + \frac{\sum_{j=1}^{N^{(i)}-1} z_j^2 - [\sum_{j=1}^{N^{(i)}-1} z_j]^2}{N^{(i)}} \\ &\leq E[\underline{d}_{DRR_n}^{(i)}]^2 \end{aligned}$$

■

B. n-flow Weighted Round Robin with WFQ-like spreading Scheduler (WRR-sp_n)

The *WRR-sp_n* scheduler [10] 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_n* scheduler is described as follows:

_n_-flow WRR with WFQ-like spreading Scheduler ($WRR - sp_n$)

Let the array A_N contain the sequence $< \frac{j}{N^{(i)}}, i >$: $j \in \{1, \cdots, N^{(i)}\}$, $1 \leq i \leq n$
sorted in lexicographic order.
The vector $\underline{f}_{WRR-sp_n}$ is constructed as follows:
$\underline{f}_{WRR-sp_n}(t)$ = *i* if $A_N(t) =< \frac{j}{N^{(i)}}, i >$

The *WRR-sp_n* scheduler possesses the following property for 1≤ $i \leq$ *n*-1:

Lemma 2: The m^{th} allocation of flow *i* always occurs between the $\lceil \frac{mN^{(j)}}{N^{(i)}} \rceil$ th and $\lceil \frac{mN^{(j)}}{N^{(i)}} \rceil$ - 1 th allocation of flow *j*, where *j*>*i*, 1≤ *m*≤ $N^{(i)}$, i.e.,

$$ns_{WRR-sp_n}^{(i,j)}(m) \quad = \quad \lceil \frac{mN^{(j)}}{N^{(i)}} \rceil - 1$$

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

$$\frac{k}{N^{(j)}} \quad < \quad \frac{m}{N^{(i)}} \quad \text{and}$$
$$\frac{k+1}{N^{(j)}} \quad \geq \quad \frac{m}{N^{(i)}}$$

Hence, we obtain $ns_{WRR-sp_n}^{(i,j)}(m)$ = *k* = $\lceil \frac{mN^{(j)}}{N^{(i)}} \rceil$ -1 ■

C. n-flow Credit Round Robin Scheduler (CRR_n)

The motivation to design the CRR_n scheduler [11] was to reduce the latency of the DRR_n scheduler. As with the DRR_n scheduler, the CRR_n scheduler can be implemented as a loop scheduler, and the pseudo-code is given as follows:

_n_-flow Credit Round Robin Scheduler (CRR_n)

```
Initialize CS_i = N^(i)/N^(n), 1≤ i ≤ n
Set t=1, SP=n, count=0
while t ≤ N
      if count < n
         if CS_SP < 1 SP = SP - 1, count = count + 1
         else f_CRR_n(t) = SP, CS_SP = CS_SP - 1
              SP = SP - 1, t = t + 1, count = 0
      else CS_i = CS_i + N^(i)/N^(n) ∀ i, count = 0
```

The CRR_n scheduler possesses the following property for 1≤ $i \leq$ *n*-1:

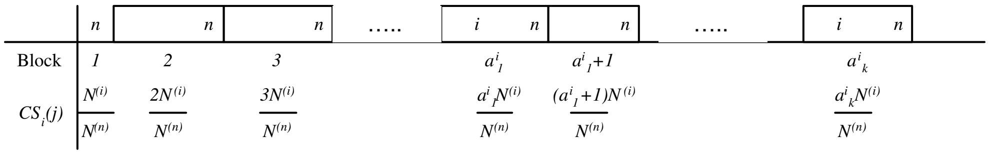


Fig. 1. Illustration of allocation to flow *n* relative to allocation to flow *i* with CRR_n scheduler

Lemma 3: The m^{th} allocation of flow *i* always occurs between the $\lceil\frac{mN^{(n)}}{N^{(i)}}\rceil$ th and $\lceil\frac{mN^{(n)}}{N^{(i)}}\rceil$ - 1 th allocation of flow *n*, 1$\leq$ *m*$\leq$ $N^{(i)}$, i.e.,

$$ns^{(i,n)}_{CRR_n}(m) = \lceil\frac{mN^{(n)}}{N^{(i)}}\rceil - 1$$

Proof: With the CRR_n scheduler, the first slot is always allocated to flow *n*. We can consider subsequent allocations in blocks, where each block terminates with the next flow *n* allocation, as illustrated in Fig. 1, where a^i_k-1 is the number of flow *n* allocations before the k^{th} allocation to flow *i*.

According to the transmission heuristics, a^i_k has to satisfy the following conditions:

$$\frac{a^i_k N^{(i)}}{N^{(n)}} \geq k \text{ and}$$
$$\frac{(a^i_k-1)N^{(i)}}{N^{(n)}} < k$$

Hence, we obtain $ns^{(i,n)}_{CRR_n}(k)$ = a^i_k-1 = $\lceil\frac{kN^{(n)}}{N^{(i)}}\rceil$ -1 ■

D. n-flow Golden Ratio (GR_n) Scheduler

The Golden Ratio Scheduler was proposed in [8] and is described as follows:

<u>*n*-flow Golden Ratio Scheduler (GR_n)</u>

Let ϕ^{-1} = 0.6180339887 and let $a_j = frac(j\phi^{-1})$ where $frac(y)$ = $y - \lfloor y \rfloor$
Let the array A_N contain the sequence a_j, 0$\leq$ $j \leq N$-1, sorted in increasing order.
The vector $\underline{f}_{GR_n}$ is constructed as follows:
$\underline{f}_{GR_n}(t)$ = *i* if $\sum_{m=1}^{i-1} N^{(m)} \leq A_N(t) \leq \sum_{m=1}^{i} N^{(m)}$, 1$\leq i \leq n$

It was established in [6] that if *N* is a Fibonacci number, then $\underline{d}^{(i)}$ comprises at most three values for each *i*; otherwise, more values are generated.

E. n-flow Short-term Fair Scheduler (STF_n)

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

$$sd^{(i)}(t) = x^{(i)} - \frac{a^{(i)}(t)}{t}$$
$$= \frac{N^{(i)}}{N} - \frac{a^{(i)}(t)}{t}$$

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

<u>*n*-flow Short-term Fair Scheduler (STF_n)</u>

Initialize $a^{(i)}(0)$ = 0, 1$\leq i \leq n$
for *t*=1:*N*
  $a^{(i)}(t)$ = $a^{(i)}(t-1)$, 1$\leq i \leq n$
  $sd^{(i)}(t)$ = $\frac{N^{(i)}}{N}$ - $\frac{a^{(i)}(t)}{t}$, 1$\leq i \leq n$
  $\underline{f}_{STF_n}(t)$ = arg $\max_{1\leq i\leq n} sd^{(i)}(t)$
  $a^{(\underline{f}_{STF_n}(t))}(t)$ = $a^{(\underline{f}_{STF_n}(t))}(t)$+1

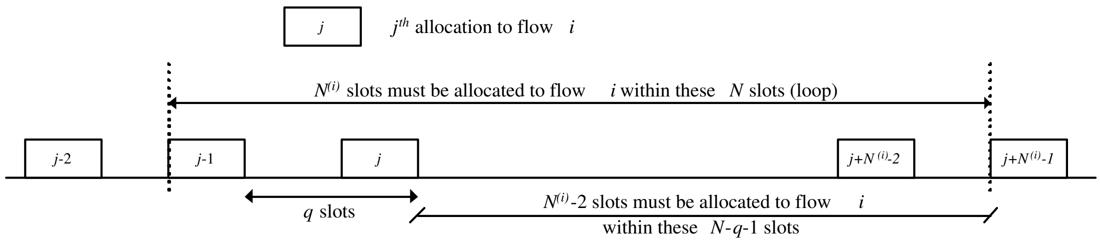


Fig. 2. Evaluation of Prob($d^{(i)}_{RND_n}(j)$ = q)

This scheduler was first suggested in [8], 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.

F. n-flow Random (RND_n) Scheduler

The loop schedulers considered so far are *deterministic* since the allocation vector $\underline{f}_\pi \in \underline{\mathbf{F}}$ is fixed. In this section, we define a *random* scheduler, RND_n, whose allocation vector, $\underline{f}_{RND_n}$ is *uniformly* selected from $\underline{\mathbf{F}}$. We note that $RND_n \in \pi_{(N,\underline{x})}$ because the selected $\underline{f}_{RND_n}$ is used for allocation in each loop.

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

$$\text{Prob}(d^{(i)}_{RND_n}(j)=q) \quad = \quad \frac{\binom{N-q-1}{N^{(i)}-2}}{\binom{N-1}{N^{(i)}-1}} \tag{4}$$

The periodicity metric for each flow *i* is evaluated as follows:

$$E[\underline{d}^{(i)}_{RND_n}]^2 \quad = \quad \frac{N(2N-N^{(i)}+1)}{N^{(i)}(N^{(i)}+1)} \tag{5}$$

By comparing Eq. (5) with Eq. (2), we can show that $E[\underline{d}^{(i)}_{RND_n}]^2 \leq E[\underline{d}^{(i)}_{DRR_n}]^2$ for 1$\leq i \leq n$.

IV. Conditions for Optimal Per-flow Periodicity for *n*-flow Loop Schedulers

In this section, we determine the conditions for optimal allocation periodicity for flow *i*. This can be expressed as a requirement on $\underline{d}^{(i)}_\pi$ in the following lemma:

Lemma 4: The following condition is satisfied:

$$E[\underline{d}^{(i)}_*]^2 \quad = \quad \min_{\pi\in\pi_{(N,\underline{x})}} E[\underline{d}^{(i)}]^2$$

if

$$\underline{d}^{(i)}_* \quad = \quad \{\overbrace{\lfloor\frac{N}{N^{(i)}}\rfloor,\cdots,\lfloor\frac{N}{N^{(i)}}\rfloor}^{N^{(i)}\lfloor\frac{N}{N^{(i)}}\rfloor+N^{(i)}-N},\underbrace{\lceil\frac{N}{N^{(i)}}\rceil,\cdots,\lceil\frac{N}{N^{(i)}}\rceil}_{N-N^{(i)}\lfloor\frac{N}{N^{(i)}}\rfloor}\} \tag{6}$$

Proof: We consider the following cases:

1) N ≡ 0 (mod $N^{(i)}$): Perfect allocation periodicity is achieved for flow *i* when the inter-allocation interval is constant, i.e., $d^{(i)}_*(j)$ = $d^{(i)}_*(k)$. This is achieved if and only if $d^{(i)}_*(j)$ = $\frac{N}{N^{(i)}}$ for 1$\leq j \leq N^{(i)}$.

2) N ≡ j (mod $N^{(i)}$), 1≤ j ≤ $N^{(i)}$-1: In this case, a constant inter-allocation interval for flow *i* cannot be achieved. The best one can achieve is the following for 1$\leq j \leq N^{(i)}$:

$$d^{(i)}_*(j) \quad \in \quad \{Q,Q+1\},\ \text{where } 1\leq Q\leq N-N^{(i)}.$$

For 1$\leq m \leq N^{(i)}$-1, let us assume the following:

$$\underline{d}^{(i)}_* \quad = \quad \{\overbrace{Q,\cdots,Q}^{m},\underbrace{Q+1,\cdots,Q+1}_{N^{(i)}-m}\}$$

Then, since $\sum_{j=1}^{N^{(i)}} d^{(i)}(j)$ = N, we have the following:

$$m \cdot Q + (N^{(i)} - m) \cdot (Q+1) \quad = \quad N$$

from which we have

$$m \quad = \quad N^{(i)} \cdot Q + N^{(i)} - N$$

However, since 1$\leq m \leq N^{(i)}$-1, we have the following constraints on *Q*:

$$\frac{N}{N^{(i)}} - 1 + \frac{1}{N^{(i)}} \leq Q \leq \frac{N}{N^{(i)}} - \frac{1}{N^{(i)}} \tag{7}$$

Since $\lfloor \frac{N}{N^{(i)}} \rfloor$ -1 $< \frac{N}{N^{(i)}} - 1 + \frac{1}{N^{(i)}}$ and $\lceil \frac{N}{N^{(i)}} \rceil > \frac{N}{N^{(i)}} - \frac{1}{N^{(i)}}$, the only integer *Q* that can satisfy Eq. (7) is *Q* = $\lfloor \frac{N}{N^{(i)}} \rfloor$. ■
The corresponding value for $E[\underline{d}_*^{(i)}]^2$ can be used as a *lower* bound for all $\pi \in \pi_{(N,\underline{x})}$ and is given as follows:

$$E[d_*^{(i)}]^2 \quad = \quad \frac{N(2x+1) - N^{(i)}x(x+1)}{N^{(i)}}$$

V. A Recursive Class-based Scheduler

In this section, we consider a *C*-class scheduling scenario [12] that can be specified in terms of the vectors $\underline{N}_C$ = $[N_1\ N_2 \cdots N_C]$ and $\underline{n}_C$ = $[n_1\ n_2 \cdots n_C]$, where each class *c* comprises n_c flows, each with demand N_c. Without loss of generality, we assume that $N_a < N_b$ for $a < b$. In addition, if Gp_c denotes the indices of flows that belong to class *c*, then we have the following:

$$Gp_c \quad = \quad [\sum_{j=1}^{c-1} n_j + 1, \sum_{j=1}^{c-1} n_j + 2, \cdots, \sum_{j=1}^{c} n_j]$$

We propose a recursive class-based scheduler based on the $WRR - sp_n$ scheduler that exhibits good periodicity.

A. 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 group should possess the same periodicity characteristics. Hence, a scheduler π ensures intra-class fairness if $E[\underline{d}_\pi^{(i)}]^2$ = $E[\underline{d}_\pi^{(j)}]^2$ for any *i*,*j* $\in Gp_c$.

A simple example of a scheduler that ensures intra-class fairness is the DRR_n scheduler. This can be observed from Eq. (2), where, for any flow $i \in Gp_c$:

$$\underline{d}_{DRR_n}^{(i)} \quad = \quad \{1, \cdots, 1, N - N_c + 1\} \tag{8}$$

On the other hand, the RND_n scheduler does not ensure intra-class fairness. As an example, we consider 2-class scheduling, where $\underline{N}_2$=[2 3] and $\underline{n}_2$=[2 2]. If $\underline{f}_{RND_n}$=[3,1,2,1,3,2,4,4,3,4], then by evaluating $\underline{d}_{RND_n}^{(i)}$, we observe that although flows 1 and 2 $\in Gp_1$, they have different periodicity characteristics.

B. Periodicity properties of $WRR - sp_n$ for C-class Scheduling

Lemma 2 can be written for class-based scheduling as follows:

Lemma 5: For the *WRR*-sp_n scheduler, flows within each group are allocated in blocks, where the order within group Gp_c is $\sum_{m=1}^{c-1} n_m + 1$, $\sum_{m=1}^{c-1} n_m$+2, $\cdots$, $\sum_{m=1}^{c} n_m$ for 1$\leq c \leq$ *C*. In addition, the m^{th} block of Gp_a will reside between the $\lceil \frac{mN_b}{N_a} \rceil^{th}$ and $\lceil \frac{mN_b}{N_a} \rceil - 1^{th}$ block of Gp_b, where *b*>*a* and 1$\leq m \leq N_a$.

We note from Lemma 5 that flows within each group are always transmitted in blocks, where each flow from that group is allocated exactly once and the order within each block is constant. Hence, the periodicity characteristics for flows belonging to the same group are identical, i.e., intra-class fairness is maintained.

1) Special Case: C=2: Using Lemma 5, $\underline{d}_{WRR-sp_n}^{(i)}$ can be evaluated as follows:

$$\underline{d}_{WRR-sp_n}^{(i)} \quad = \quad \begin{cases} \{\overbrace{n_1 + n_2\lfloor \frac{N_2}{N_1} \rfloor, \cdots, n_1 + n_2\lfloor \frac{N_2}{N_1} \rfloor}^{N_1\lceil \frac{N_2}{N_1} \rceil - N_2}, \underbrace{n_1 + n_2\lceil \frac{N_2}{N_1} \rceil \cdots n_1 + n_2\lceil \frac{N_2}{N_1} \rceil}_{N_2 - (\lceil \frac{N_2}{N_1} \rceil - 1)N_1}\}, & i \in Gp_1; \\ \{\overbrace{n_2, \cdots, n_2}^{\frac{N_2 n - N}{n_1}}, \underbrace{n, \cdots, n}_{\frac{N - n_2 N_2}{n_1}}\}, & i \in Gp_2. \end{cases} \tag{9}$$

For two-class scheduling, Eq. (6) can be written as follows:

$$\underline{d}_*^{(i)} = \begin{cases} \{\overbrace{n_1+\lfloor\frac{n_2N_2}{N_1}\rfloor,\cdots,n_1+\lfloor\frac{n_2N_2}{N_1}\rfloor}^{N_1(n_1+\lfloor\frac{n_2N_2}{N_1}\rfloor)+N_1-N},\underbrace{n_1+\lceil\frac{n_2N_2}{N_1}\rceil,\cdots,n_1+\lceil\frac{n_2N_2}{N_1}\rceil}_{N-N_1(n_1+\lfloor\frac{n_2N_2}{N_1}\rfloor)}\}, & i\in Gp_1; \\ \{\overbrace{n_2+\lfloor\frac{n_1N_1}{N_2}\rfloor,\cdots,n_2+\lfloor\frac{n_1N_1}{N_2}\rfloor}^{N_2(n_2+\lfloor\frac{n_1N_1}{N_2}\rfloor)+N_2-N},\underbrace{n_2+\lceil\frac{n_1N_1}{N_2}\rceil,\cdots,n_2+\lceil\frac{n_1N_1}{N_2}\rceil}_{N-N_2(n_2+\lfloor\frac{n_1N_1}{N_2}\rfloor)}\}, & i\in Gp_2. \end{cases} \quad (10)$$

Comparing Eq. (9) with Eq. (10), we note that $\underline{d}_{WRR-sp_n}^{(i)} \neq \underline{d}_*^{(i)}$ for 1$\leq i \leq n$ and hence, the *WRR-*sp_n scheduler is not optimal in terms of per-flow periodicity. However, we note that when n_1=1 (n_1=*n*-1), the *WRR-*sp_n scheduler offers optimal periodicity for flows in Gp_2 (Gp_1). However, if N_1=1, then optimal (worst-case) periodicity is achieved for flows in Gp_1 (Gp_2).

The corresponding periodicity properties for the STF_n and RND_n schedulers for two-class scheduling can be found in Appendix I and II respectively.

2) Enhancement to WRR-sp_n Scheduler: From Section V-B.1, we observe that the *WRR-*sp_n scheduler results in worst-case periodicity for Gp_2 flows when n_1=*n*-1 and N_1=1. This is due to the default lexicographic ordering in the scheduling mechanism, which can be circumvented by introducing a parameter, *p*, 1$\leq p \leq$ *n*, to the *WRR-*sp_n scheduler (denoted *WRR-*$sp_n(p)$). With the *WRR-*$sp_n(p)$ scheduler, the ordering priority in the event of a tie in the elements $\{\frac{j}{N^{(i)}}\}_{j=1}^{N^{(i)}}$ for 1$\leq i \leq n$ is given by [*p*,*p*+1,*p*+2,$\cdots$,*n*,1,2,$\cdots$,*p*-1]. We note that the scheduler reduces to the original *WRR-*sp_n scheduler when *p* = 1.

C. A Recursive Approach to Class-based Scheduling

Instead of 'blindly' applying any loop scheduler to a class-based scenario, we can define a *Class-based* scheduler that first allocates slots to each class (*inter-class* scheduling) and then distributes the allocated slots within each class to each flow (*intra-class* scheduling). We propose a recursive approach for class-based scheduling. Let $REC^a(\underline{I})$ be a recursive $|\underline{I}|$-class recursive scheduler with inputs *a* $\in \underline{I}$, where $\underline{I} \subset \underline{C}$ = {1,2, $\cdots$,*C*}. The mechanism of $REC^a(\underline{I})$ comprises two stages: The first stage involves obtaining the allocation vectors for flows $\in \{Gp_j\}_{\forall j\in \underline{I}\backslash a}$ and Gp_a respectively. The second stage combines these allocation vectors to obtain $\underline{f}_{REC^a(\underline{I})}$.

The allocation vector for flows $\in \{Gp_j\}_{\forall j\in \underline{I}\backslash a}$ is obtained by evaluating $\underline{f}_{REC^b(\underline{I}\backslash a)}$ for some *b* $\in \underline{I}\backslash a$. Since the flows $\in Gp_a$ are homogeneous, a simple round robin allocation is optimal in terms of periodicity, and the allocation vector is given as follows:

$$\underline{f}_{RR_{n_a\cdot N_a}} = [1,2,\cdots,n_a,1,2,\cdots,n_a,\cdots,\cdots,1,2,\cdots,n_a]$$

We note that the elements of $\underline{f}_{REC^b(\underline{I}\backslash a)}$ and $\underline{f}_{RR_{n_a\cdot N_a}}$ have to be updated accordingly to ensure that the correct flow indices are assigned to flows in each group. Our approach in the second stage is to insert the elements of $\underline{f}_{RR_{n_a\cdot N_a}}$ into $\underline{f}_{REC^b(\underline{I}\backslash a)}$ such that successive elements of $\underline{f}_{RR_{n_a\cdot N_a}}$ are as uniformly separated as possible in $\underline{f}_{REC^a(\underline{I})}$. This is illustrated in Fig. 3, where M_a = $\lceil\frac{|\underline{f}_{REC^b(\underline{I}\backslash a)}|}{|\underline{f}_{RR_{n_a\cdot N_a}}|}\rceil$.

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

D. Optimal Two-Class Scheduling based on the WRR-sp_n Scheduler

From Section V-B.1, we observe that for the special case of *n*=2, since n_1=1 =*n*-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 $\underline{d}_{OPT_2}^{(i)}$ = $\underline{d}_*^{(i)}$ as given in Eq. (10) for 1$\leq i \leq n$. Hence, the OPT_2 scheduler is optimal for two-class scheduling, and the pseudo-code is given below, assuming $n_1N_1 \leq n_2N_2$ (the corresponding scheduler for $n_1N_1 > n_2N_2$ can be obtained by inter-changing the indices 1 and 2):

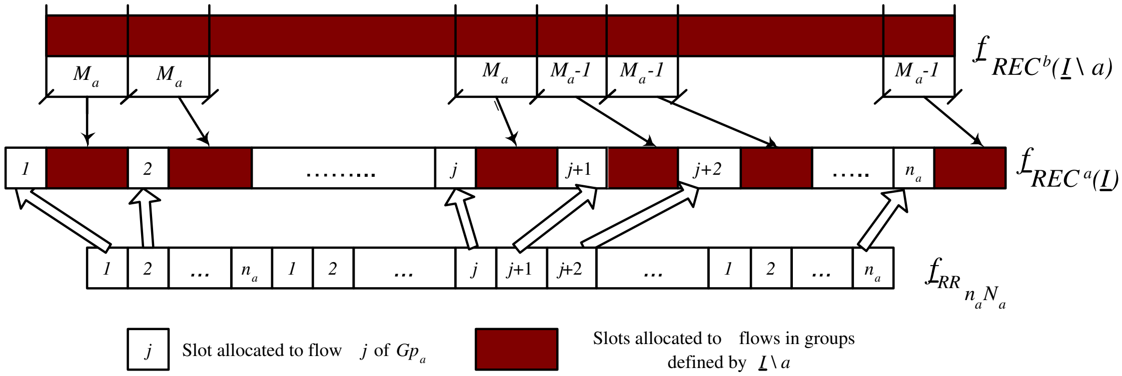


Fig. 3. Illustration of the mechanism of a $REC^{a}(\underline{I})$ scheduler

Optimal Two-Class Scheduler (OPT_2)

Set $\underline{N}$ = $[n_1 \cdot N_1 \ n_2 \cdot N_2]$, n = n_1 + n_2

Define $\underline{Gp}_1$ = $[\overbrace{1, 2, \cdots, n_1}^{1}, \overbrace{1, 2, \cdots, n_1}^{2}, \cdots, \cdots, \overbrace{1, 2, \cdots, n_1}^{n_1 \cdot N_1}]$

Define $\underline{Gp}_2$ = $[\underbrace{n_1+1, n_1+2, \cdots, n}_{1}, \underbrace{n_1+1, n_1+2, \cdots, n}_{2} \cdots, \cdots, \underbrace{n_1+1, n_1+2, \cdots, n}_{n_2 \cdot N_2}]$

Compute $\underline{f}_{WRR-sp_2}$ = *WRR*-$sp_2(\underline{N})$

for c = 1:2

index = find($\underline{f}_{WRR-sp_2}$==c)

$\underline{f}_{OPT_2}$(index) = $\underline{Gp}_c$

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

VI. Numerical Results

In this section, we shall compare the performance of various loop schedulers for multi-class scheduling in terms of a metric that reflects the periodicity over the ensemble of all flows. The metric that we consider (denoted by $wcov_\pi$) is the weighted covariance of $\{\underline{d}_\pi^{(i)}\}_{i=1}^{n}$ achieved by the scheduler π, which is defined as follows:

$$wcov_\pi \quad = \quad \sum_{i=1}^{n} x^{(i)} \cdot \frac{E[\underline{d}_\pi^{(i)}]^2 - (E[\underline{d}_\pi^{(i)}])^2}{(E[\underline{d}_\pi^{(i)}])^2}$$

With perfect periodicity, $wcov_\pi$=0 since $\underline{d}_\pi$ = $E[\underline{d}_\pi^{(i)}]$. Hence, a value close to zero indicates that a scheduler exhibits good periodicity properties.

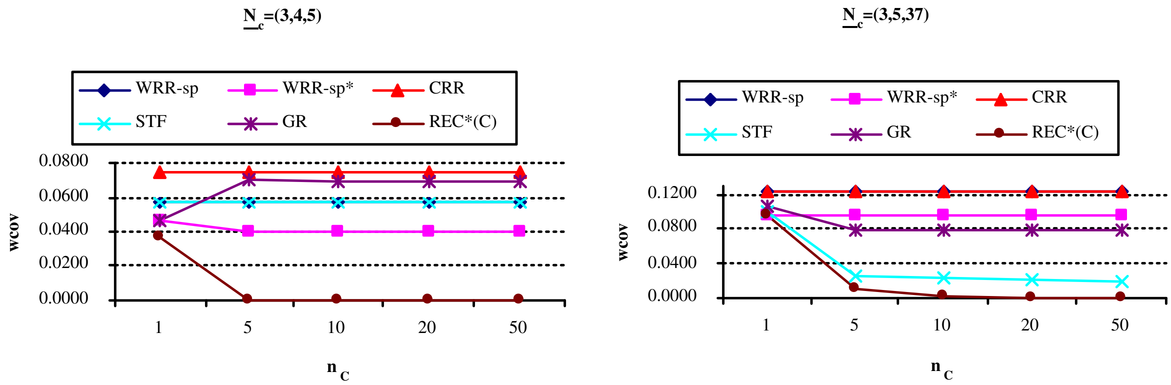
We define the optimal $WRR - sp_n$ scheduler (denoted $WRR - sp_n^*$), where $WRR - sp_n^*$ = *WRR*-$sp_n(p^*)$ such that p^* = $\arg\min_{1 \le p \le n} wcov_{WRR-sp_n(p)}$. Similarly, we define the optimal recursive class-based scheduler (denoted $REC^*(\underline{C})$), where $REC^*(\underline{C})$ = $REC^{a^*}(\underline{C})$ such that a^* = $\arg\min_{\forall a \in \underline{C}} wcov_{REC^a(\underline{C})}$.

Let us consider the following broadband applications with the corresponding typical bandwidth requirements in kbps [13]: 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 N_c=(300,400,500) $\equiv$ (3,4,5).

Assuming uniform flow composition, i.e., n_c = n_C for $1 \le c \le$ *C*, the results for various scenarios for *C* = 3 are shown in Fig. 4 and Fig. 5. The corresponding results for *C*=4 and 5 are shown in Fig. 8 and Fig. 7. Although not depicted in the figures, we note that the RND_n scheduler performs significantly worse than the deterministic schedulers. In addition, the weighted covariance for each scheduler is relatively invariant with n_C for n_C >1 for a given $\underline{N}_C$. Hence, we consider the following cases:

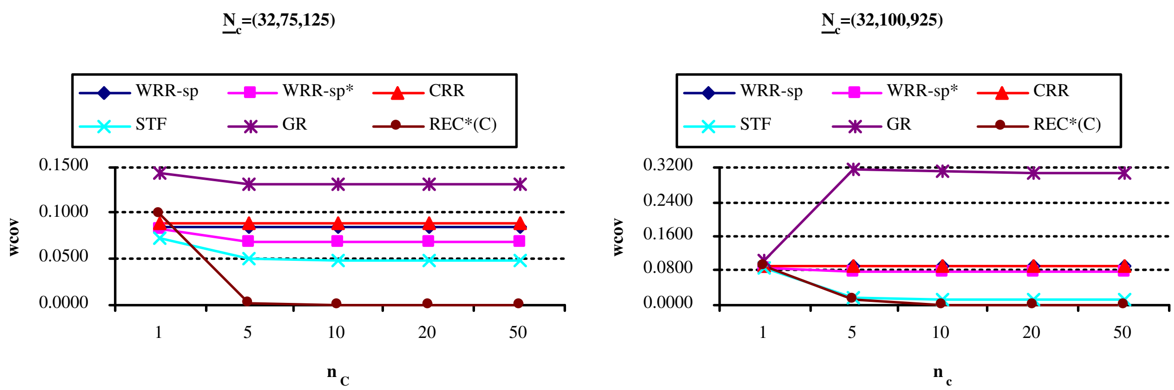
A. n_C >1

Amongst the *WRR*-sp_n, *WRR*-sp_n^* and CRR_n schedulers, the CRR_n performs the *worst*. In addition, an enhancement of the *WRR*-sp_n always exists and the gain in terms of the weighted covariance is significant. Amongst the $REC^*(\underline{C})$, STF_n



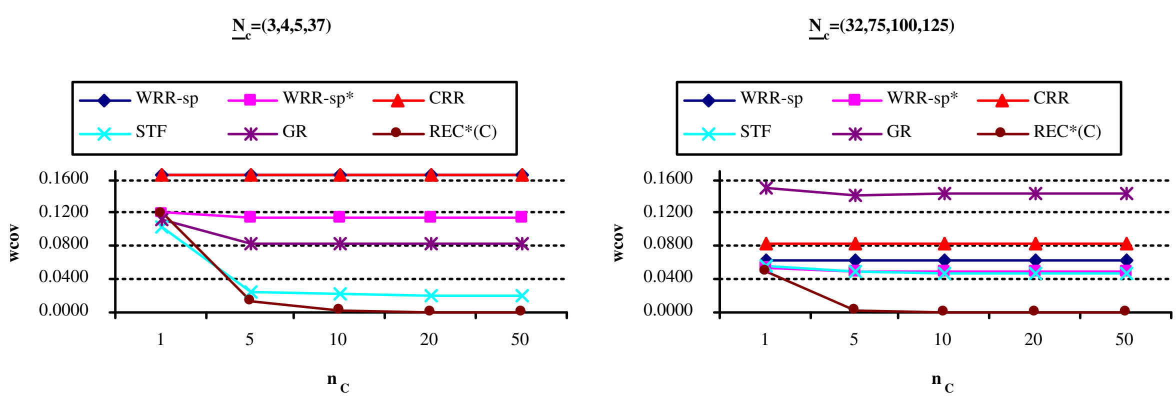
(a) (b)

Fig. 4. Comparison of $wcov_{\pi}$ for various C-class loop schedulers for $\underline{N}_C$ = (a) (3,4,5) and (b) (3,5,37)



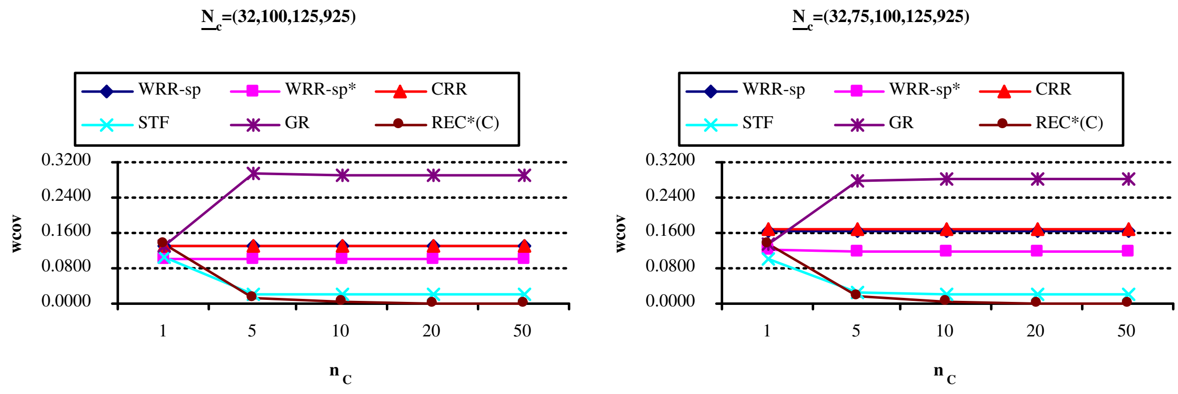
(a) (b)

Fig. 5. Comparison of $wcov_{\pi}$ for various C-class loop schedulers for $\underline{N}_C$ = (a) (32,75,125) and (b) (32,100,925)



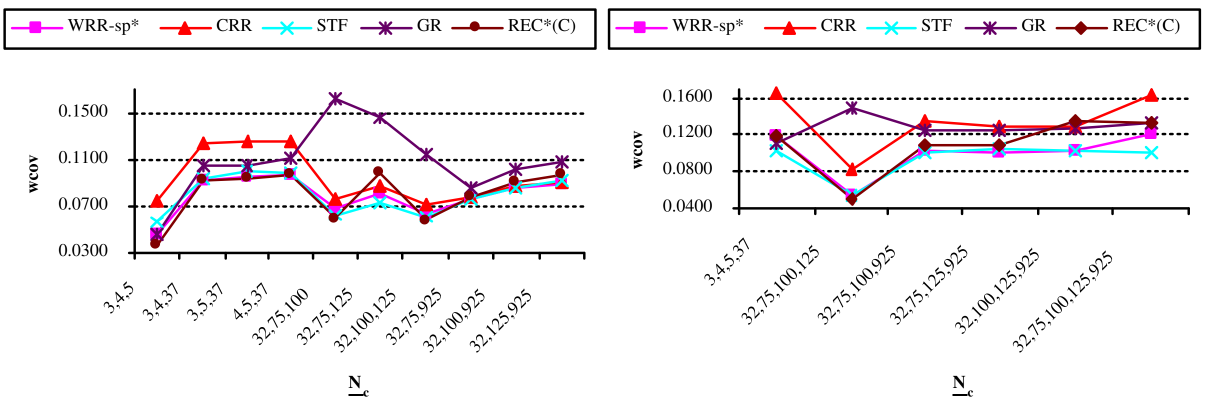
(a) (b)

Fig. 6. Comparison of $wcov_{\pi}$ for various C-class loop schedulers for $\underline{N}_C$ = (a) (3,4,5,37) and (b) (32,75,100,125)



(a) (b)

Fig. 7. Comparison of $wcov_\pi$ for various C-class loop schedulers for $\underline{N}_C$ = (a) (32,100,125,925) and (b) (32,75,100,125,925)



(a) (b)

Fig. 8. Comparison of $wcov_\pi$ for various C-class loop schedulers for n_C=1 and (a) C=3 and (b) C=4 and 5

and GR_n schedulers, the relative performances is always according to the above order, with the $REC^*(\underline{C})$ scheduler achieving the *best* performance amongst all the schedulers. In fact, the weighted covariance achieved by the STF_n and $REC^*(\underline{C})$ are nearly optimal for scenarios where $N_C >> N_c$, 1$\leq c \leq C$-1.

While the *WRR-sp*$_n$, CRR_n and *WRR-sp*$^*_n$ schedulers ensure intra-class fairness for any scheduling scenario, it is not enforced by the STF_n and $REC^*(\underline{C})$ schedulers for certain scenarios, and is never enforced by the GR_n 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-sp*$^*_n$ scheduler should be used; otherwise, the $REC^*(\underline{C})$ scheduler should be used.

B. n_C=1

For an easier comparison of the periodicity performance of the schedulers, we plot the results for n_C=1 in Fig. 8. The STF_n and *WRR-sp*$^*_n$ schedulers offer the best overall periodicity performance, while the GR_n and CRR_n schedulers offer the worst performance. We note that intra-class fairness is irrelevant in this case.

VII. Conclusions

In this paper, we consider the problem of finding a weighted time-division multiplexed loop scheduler for *n* flows that minimizes the average packet delay. The optimization criteria translates to finding a loop scheduler that allocates slots to flows as periodically as possible. We use the second moment of the inter-allocation distance for a flow *i* as the periodicity metric of the scheduler with respect to that flow. We derive the conditions for optimal per-flow periodicity for any *n*-flow loop scheduler.

We consider a class-based scheduling scenario where flows can be grouped according to their relative bandwidth demands. We analyze the periodicity properties of a weighted round robin with spreading (*WRR-sp*$_n$) scheduler for a two-class scenario.

Based on these properties, we establish an optimal scheduler that employs the *WRR-*sp_n scheduler as an inter-class scheduler, and also suggest an enhancement to the *WRR-*sp_n scheduler. We then propose a recursive class-based scheduler based on the two-class optimal scheduler.

We define a metric that reflects the periodicity performance over the ensemble of all flows. We then compare the performance of the above schedulers in terms of numerical results. 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, the enhanced *WRR-*sp_n 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.

APPENDIX I
PERIODICITY PROPERTIES OF STF_n SCHEDULER FOR TWO-CLASS SCHEDULING

The periodicity properties of the STF_n scheduler for two-class scheduling can be specified by the following Lemma:

Lemma 6: Flows in Gp_2 are always allocated in blocks, where the order within each block is n,n-1,n-2,$\cdots$,n-n_1; Flows in Gp_1 are always allocated in the order n_1,n_1-1,$\cdots$,1 and the maximum number of Gp_1 flows allocated between two successive Gp_2 blocks is n_1.

A. Periodicity Properties for Gp_1 flows

We can deduce from Lemma 6 that for 1$\leq j \leq N_1$, $d^{(i)}(j)$ = n_1 + (n-n_1)x_j, where $x_j \geq 1$. Since $\sum_{j=1}^{N_1} d^{(i)}(j)$=$N$, we have the following additional constraint on x_j:

$$\sum_{j=1}^{N_1} x_j = N_2$$

Based on numerical results, we can infer that for 1$\leq j \leq N_1$,

$$x_j = \lceil \frac{jN_2}{N_1} \rceil - \lceil \frac{(j-1)N_2}{N_1} \rceil$$

Hence, $\underline{d}^{(i)}_{STF_n}$ can be written as follows:

$$\underline{d}^{(i)}_{STF_n} = \{\overbrace{n_1 + n_2\lfloor \frac{N_2}{N_1} \rfloor, \cdots, n_1 + n_2\lfloor \frac{N_2}{N_1} \rfloor}^{N_1\lceil \frac{N_2}{N_1} \rceil - N_2}, \underbrace{n_1 + n_2\lceil \frac{N_2}{N_1} \rceil \cdots n_1 + n_2\lceil \frac{N_2}{N_1} \rceil}_{N_2-(\lceil \frac{N_2}{N_1} \rceil -1)N_1}\}$$

$E[\underline{d}^{(i)}_{STF_n}]^2$ can be evaluated as follows:

$$E[\underline{d}^{(i)}_{STF_n}]^2 = n_1^2 N_1 + (2\lceil \frac{N_2}{N_1} \rceil - 1)(n-n_1)^2 N_2 + 2n_1 N_2 - N_1(n-n_1)^2 \lceil \frac{N_2}{N_1} \rceil (\lceil \frac{N_2}{N_1} \rceil - 1) \quad (11)$$

B. Periodicity Properties for Gp_2 flows

Similarly, we can deduce from Lemma 6 that for 1$\leq j \leq N_2$, n-$n_1 \leq d^{(i)}(j) \leq$ n. Let us define $d^{(i)}(j)$ = n-n_1+x_j, where $0 \leq x_j \leq n_1$. Since $\sum_{j=1}^{N_2} d^{(i)}(j)$=$N$, we have the following additional constraint on x_j:

$$\sum_{j=1}^{N_2} x_j = n_1 N_1$$

$E[\underline{d}^{(i)}_{STF_n}]^2$ can be computed in terms of x_j as follows:

$$\begin{aligned} E[\underline{d}^{(i)}_{STF_n}]^2 &= \frac{1}{N_2}\sum_{j=1}^{N_2}(n-n_1+x_j)^2 \\ &= (n-n_1)^2 + \frac{2(n-n_1)}{N_2}\sum_{j=1}^{N_2} x_j + \frac{1}{N_2}\sum_{j=1}^{N_2} x_j^2 \\ &= \frac{(n-n_1)^2 N_2 + 2(n-n_1)n_1 N_1 + \sum_{j=1}^{N_2}[x_j]^2}{N_2} \end{aligned}$$

We can evaluate the upper (lower) bound on $E[\underline{d}_{STF_n}^{(i)}]^2$, denoted by $\overline{E}[\underline{d}_{STF_n}^{(i)}]^2$ ($\underline{E}[\underline{d}_{STF_n}^{(i)}]^2$), by considering the following constrained optimization problem, :

$$\boxed{\begin{array}{ll} \text{Maximize (Minimize)} & \sum_{j=1}^{N_2}[x_j]^2 \text{ such that for } 1\le i \le n \\ & \sum_{j=1}^{N_2} x_j = n_1 N_1 \\ \text{and} & 0 \le x_j \le n_1 \end{array}}$$

1) Expression for $\overline{E}[\underline{d}_{STF_n}^{(i)}]^2$: Using $(a+b)^2 \ge a^2 + b^2$ for *a*,*b*$\ge$ 0, we can show that $\sum_{j=1}^{N_2}[x_j]^2$ is maximized with the following choice of $\{x_j\}_{j=1}^{N_2}$:

$$\{x_j\}_{j=1}^{N_2} = \{\overbrace{n_1,\cdots,n_1}^{N_1},\underbrace{0,\cdots,0}_{N_2-N_1}\}$$

Therefore, $\overline{E}[\underline{d}_{STF_n}^{(i)}]^2$ can be evaluated as follows:

$$\begin{aligned} \overline{E}[\underline{d}_{STF_n}^{(i)}]^2 &= \frac{(n-n_1)^2N_2 + 2(n-n_1)n_1N_1 + n_1^2N_1}{N_2} \\ &= \frac{(n-n_1)^2N_2 + (2n-n_1)n_1N_1}{N_2} \end{aligned} \tag{12}$$

2) Expression for $\underline{E}[\underline{d}_{STF_n}^{(i)}]^2$: For $mN_1 < N_2 - N_1 \le (m+1)N_1$, where 0$\le m \le n_1$-1, $\sum_{j=1}^{N_2}[x_j]^2$ is minimized with the following choice of $\{x_j\}_{j=1}^{N_2}$:

$$\{x_j\}_{j=1}^{N_2} = \{\overbrace{n_1-(m+1),\cdots,n_1-(m+1)}^{N_2-(m+1)N_1},\underbrace{n_1-m,\cdots,n_1-m}_{(m+2)N_1-N_2},\underbrace{1,\cdots,1}_{N_2-N_1}\}$$

Therefore, $\underline{E}[\underline{d}_{STF_n}^{(i)}]^2$ can be evaluated as follows:

$$\underline{E}[\underline{d}_{STF_n}^{(i)}]^2 = \frac{(n-n_1)^2N_2 + 2(n-n_1)n_1N_1 + n_1^2N_1 - m(m+1)N_1 - 2(N_2-N_1)(n_1-m-1)}{N_2} \tag{13}$$

C. Comparison of Periodicity Characteristics with WRR-sp_n Scheduler

Using Eq. (9), the periodicity metric for the *WRR-sp_n* scheduler for two-class scheduling can be evaluated and is given as follows:

$$E[\underline{d}_{WRR-sp_n}^{(i)}]^2 = \begin{cases} n_1^2N_1 + (2\lceil\frac{N_2}{N_1}\rceil - 1)(n-n_1)^2N_2 + \\ 2n_1N_2 - N_1(n-n_1)^2\lceil\frac{N_2}{N_1}\rceil(\lceil\frac{N_2}{N_1}\rceil - 1), & i \in Gp_1; \\ \frac{N_2(n-n_1)^2 + N_1n_1(2n-n_1)}{N_2}, & i \in Gp_2. \end{cases} \tag{14}$$

Comparing Eq. (11), Eq. (12) and Eq. (13) with Eq. (14), we have the following result, where 0$\le m \le n_1$-1 and $mN_1 < N_2 - N_1 \le (m+1)N_1$:

$$\begin{array}{ll} E[\underline{d}_{STF_n}^{(i)}]^2 = E[\underline{d}_{WRR-sp_n}^{(i)}]^2, & i \in Gp_1 \\ E[\underline{d}_{WRR-sp_n}^{(i)}]^2 - \frac{m(m+1)N_1 + 2(N_2-N_1)(n_1-m-1)}{N_2} \le E[\underline{d}_{STF_n}^{(i)}]^2 \le E[\underline{d}_{WRR-sp_n}^{(i)}]^2, & i \in Gp_2 \end{array}$$

In general, for two-class scheduling, while maintaining intra-class fairness, the periodicity characteristics of the STF_n scheduler is at least equal or better than that of the *WRR-sp_n* scheduler.

APPENDIX II
PERIODICITY PROPERTIES OF RND_n SCHEDULER FOR TWO-CLASS SCHEDULING

For two-class scheduling, according to Eq. (4), we have the following:

$$\text{Prob}(d_{RND_n}^{(i)} = q) = \begin{cases} \frac{\binom{N-q-1}{N_1-2}}{\binom{N-1}{N_1-1}}, & 1 \le q \le N - N_1 + 1, \quad i \in Gp_1; \\ \frac{\binom{N-q-1}{N_2-2}}{\binom{N-1}{N_2-1}}, & 1 \le q \le N - N_2 + 1, \quad i \in Gp_2. \end{cases} \tag{15}$$

From Eq. (5), the per-flow periodicity metric is given as follows:

$$E[\underline{d}^{(i)}_{RND_n}]^2 = \begin{cases} \frac{N(2N-N_1+1)}{N_1(N_1+1)}, & i \in Gp_1; \\ \frac{N(2N-N_2+1)}{N_2(N_2+1)}, & i \in Gp_2. \end{cases} \quad (16)$$

Since $\underline{d}^{(i)}_{RND_n}$ takes on more than two values in general, we expect the RND_n scheduler to be sub-optimal in terms of per-flow periodicity. Intuitively, we also expect it to perform worse than the deterministic schedulers. We examine these hypotheses by considering two cases: (a) *n*=2 and (b) *n*>2.

A. n=2

According to Eq. (15), we have the following:

$$\text{Prob}(d^{(1)}_{RND_2} = q) = \frac{\binom{N-q-1}{N_1-2}}{\binom{N-1}{N_1-1}}, \quad 1 \le q \le N_2 + 1$$

$$\text{Prob}(d^{(2)}_{RND_2} = q) = \frac{\binom{N-q-1}{N_2-2}}{\binom{N-1}{N_2-1}}, \quad 1 \le q \le N_1 + 1$$

Comparing with Eq. (6), we note that for N_1 >1, $\underline{d}^{(i)}_{RND_2} \neq \underline{d}^{(i)}_{*}$ in general and hence,

$$E[\underline{d}^{(i)}_{RND_2}]^2 > E[\underline{d}^{(i)}_{WRR-sp_2}]^2 = E[\underline{d}^{(i)}_{STF_2}]^2$$

However, for N_1=1, $d^{(1)}_{RND_2}$=*N* and $E[\underline{d}^{(1)}_{RND_2}]^2$ is always optimal. In addition, the pdf of $\underline{d}^{(2)}_{RND_2}$ is given as follows:

$$\text{Prob}(d^{(2)}_{RND_2} = q) = \begin{cases} \frac{N_2-1}{N_2}, & q = 1; \\ \frac{1}{N_2}, & q = 2. \end{cases}$$

In other words, $\underline{d}^{(2)}_{RND_2}$ = $\{1 \cdots, 1, 2\}$ = $\underline{d}^{(2)}_{*}$. Hence, for $\underline{N}$ = (1, N_2), we have the following:

$$E[\underline{d}^{(i)}_{RND_2}]^2 = E[\underline{d}^{(i)}_{WRR-sp_2}]^2 = E[\underline{d}^{(i)}_{STF_2}]^2$$

B. n>2

Here, we consider several cases with counter-intuitive observations:

1) n_1=n-1,N_1=1: In this case, we only need to evaluate the periodicity characteristics for Gp_2 flows. From Eq. (16), the periodicity metric for the RND_n scheduler is given as follows:

$$E[\underline{d}^{(i)}_{RND_n}]^2 = 1 + \frac{1}{N_2}(n-1)(1 + \frac{2(N_2 + (n-1))}{N_2+1})$$

From Eq. (14), the corresponding metric for the *WRR-*sp_n scheduler is given as follows:

$$E[\underline{d}^{(i)}_{WRR-sp_n}]^2 = 1 + \frac{1}{N_2}(n-1)(1+n)$$

Since *n* $> \frac{2(N_2+(n-1))}{N_2+1}$ for *n*>2, we have the following result:

$$E[\underline{d}^{(i)}_{RND_n}]^2 < E[\underline{d}^{(i)}_{WRR-sp_n}]^2$$

2) n_1=n-1,N_2=2: From Appendix I Section I-C, $E[\underline{d}^{(i)}_{STF_n}]^2$ = $E[\underline{d}^{(i)}_{WRR-sp_n}]^2$, and hence, we have the following result:

$$E[\underline{d}^{(i)}_{RND_n}]^2 < E[\underline{d}^{(i)}_{STF_n}]^2$$

Therefore, for two-class scheduling where n_1=*n*-1 and (N_1,N_2) = (1,2), the random scheduler exhibits better periodicity properties than the deterministic schedulers.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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, no. 3, pp. 341–349, May 1987.

[7] Z. Rosberg, "Optimal Decentralized Control in a Multiaccess Channel with Partial Information," *IEEE Transactions on Automatic Control*, vol. 28, no. 2, pp. 187–193, February 1983.

[8] A. Itai and Z. Rosberg, "A Golden Ratio Control Policy for a Multiple-Access Channel," *IEEE Trans. Information Theory*, vol. 33, no. 3, pp. 341–349, May 1987.

[9] M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin," *IEEE/ACM Transanctions on Networking*, vol. 4, no. 3, pp. 375–385, June 1996.

[10] H. Zhang, "Service disciplines for guaranteed performance service in packet - switching networks," *Proceeding of the IEEE*, vol. 83, no. 10, pp. 1374–1399, October 1995.

[11] 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.

[12] Y. Guo and H. Chaskar, "Class-based Quality of Service over Air Interfaces in 4G Mobile Networks," *IEEE Communications Magazine*, vol. 40, no. 3, pp. 132–137, March 2002.

[13] 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`.