Interleaved Weighted Round-Robin: A Network Calculus Analysis

Seyed Mohammadhossein Tabatabaei  
EPFL  
Lausanne, Switzerland  
hossein.tabatabaei@epfl.ch

Jean-Yves Le Boudec  
EPFL  
Lausanne, Switzerland  
jean-yves.leboudec@epfl.ch

Marc Boyer  
ONERA/OTIS, University of Toulouse  
F-31055 Toulouse, France  
Marc.Boyer@onera.fr

Abstract—Weighted Round-Robin (WRR) is often used, due to its simplicity, for scheduling packets or tasks. With WRR, a number of packets equal to the weight allocated to a flow can be served consecutively, which leads to a bursty service. Interleaved Weighted Round-Robin (IWRR) is a variant that mitigates this effect. We are interested in finding bounds on worst-case delay obtained with IWRR. To this end, we use a network calculus approach and find a strict service curve for IWRR. The result is obtained using the pseudo-inverse of a function. We show that the strict service curve is the best obtainable one, and that delay bounds derived from it are tight (i.e., worst-case) for flows of packets of constant size. Furthermore, the IWRR strict service curve dominates the strict service curve for WRR that was previously published. We provide some numerical examples to illustrate the reduction in worst-case delays caused by IWRR compared to WRR.

I. INTRODUCTION

Weighted Round-Robin (WRR) is a scheduling algorithm that is often used for scheduling tasks, or packets, in real-time systems or communication networks. The capacity is shared among several clients or queues by giving each of them a weight, which is a positive integer, and by providing more service to those with larger weights. Specifically, queues are visited one after the other, and when a queue \( i \) with weight \( w_i \) has an emission opportunity, it sends \( w_i \) packets, or less if fewer packets are present. The advantage of WRR is that it is fair and simple. However, the service is bursty because up to \( w_i \) packets can be served consecutively for queue \( i \), which can cause a large worst-case waiting time for other queues. Interleaved Weighted Round-Robin (IWRR) mitigates this effect [1]. With IWRR, a queue \( i \) with weight \( w_i \) has \( w_i \) emission opportunities per round and can send up to one packet at every emission opportunity. In contrast, with WRR, it has one emission opportunity per round and can send up to \( w_i \) packets at every emission opportunity. Hence, IWRR spreads out emission opportunities of each queue in a round, which is expected to result in a smoother service and lower worst-case delays. There exist several versions of IWRR; we focus on the simplest one, where queue \( i \) has emission opportunities in the first \( w_i \) cycles within a round (see Section III for a formal description of IWRR and Section IV for WRR variants).

We are interested in delay bounds for the worst case, as is typical in the context of deterministic networking. To this end, a standard approach is network calculus. Specifically, with network calculus, the service offered to a flow of interest by a system is abstracted by means of a service curve. A bound on the worst-case delay is obtained by combining the service curve with an arrival curve for the flow of interest. An arrival curve is a constraint on the amount of data that the flow of interest can send; such a constraint is necessary to the existence of a finite delay bound. The exact definitions are recalled in Section II.

The network calculus approach was applied to WRR in [2, Sec. 8.2.4], where a strict service curve is obtained. As explained in Section II, a strict service curve is a special case of a service curve hence can be used to derive delay (and backlog) bounds. Our first contribution is to obtain a strict service curve for IWRR. Compared to WRR, the interleaving in IWRR makes the analysis more difficult, and the method of proof in [2] cannot easily be extended. To circumvent this difficulty, we rely heavily on the method of pseudo-inverse, recalled in Section II. As expected, the IWRR strict service curve dominates that of WRR, hence the resulting delay bounds for IWRR are always less than or equal to those for WRR.

The strict service curve enables us to obtain delay bounds by using network calculus, but such bounds might not always be tight, i.e., they might not always be equal to worst-cases. This is because the strict service curve is an abstraction of the system. Our second contribution is to show that, for flows with packets of constant sizes, the strict service curve obtained for IWRR provides tight delay bounds. We show that the same result holds for the existing strict service curve of WRR. Extending such results to flows with packets of variable sizes is left for further study.

The strict service curve obtained for IWRR has some description complexity, see also Fig. 3. Therefore, we provide simplified lower bounds that can be used, at the expense of sub-optimality, when analytic, closed-form expressions are important.

After giving some necessary background on network calculus and the lower-pseudo inverse technique in Section II, we describe our system model in Section III. We describe the state of the art in Section IV. In Section V, we present our strict service curve for IWRR, the proof of which we present in Section VI. In Section VII, we show that both the IWRR and WRR strict service curves are the best possible and that
they give tight delay bounds for a flow with constant packet sizes. We use numerical examples to illustrate the worst-case latency improvement of IWR over WRR obtained with our method in Section VIII. Due to lack of space, proofs of results other than Theorem 1 are in [3].

II. BACKGROUND

We use the framework of network calculus [2], [4], [5]. A flow is represented by a cumulative arrival function \( R(t) \) in \( \mathbb{R}^+ \), where \( \mathcal{F} \) denotes the set of wide-sense increasing functions \( f : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \cup \{+\infty\} \) and \( R(t) \) is the number of bits observed on the flow between times 0 and \( t \). We say that a flow \( R \) has \( \alpha \in \mathcal{F} \) as arrival curve if for all \( s \leq t \), \( R(t) - R(s) \leq \alpha(t-s) \). A frequently used arrival curve is \( \alpha = \gamma_{r,b} \), defined by \( \gamma_{r,b}(t) = rt + b \) for \( t > 0 \) and \( \gamma_{r,b}(t) = 0 \) for \( t = 0 \) (token bucket arrival curve, with rate \( r \) and burst \( b \)). An arrival curve \( \alpha \) can always be assumed to be sub-additive, i.e., to satisfy \( \alpha(s + t) \leq \alpha(s) + \alpha(t) \) for all \( s, t \).

For two functions \( f \) and \( g \) in \( \mathcal{F} \), the min-plus convolution is defined by \((f \diamond g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\} \). An example of min-plus convolution used in this paper is illustrated in Fig. 1.

Consider a system \( S \) and a flow through \( S \) with input and output functions \( R \) and \( R^* \) and let \( \beta \in \mathcal{F} \). We say that the system \( S \) offers a service curve to the flow if \( R^* \geq R \odot \beta \), which often means that for every \( t \geq 0 \) there exists some \( s \leq t \) such that \( R^*(t) \geq R(s) + \beta(t-s) \) [2, Sec. 3.2.2]. We say that system \( S \) offers a strict service curve \( \beta \in \mathcal{F} \) to the flow if \( R^*(t) - R^*(s) \geq \beta(t-s) \) whenever \( s,t \) is a backlogged period (i.e., \( R^*(\tau) \geq R(\tau) \) for all \( s \) such that \( s < \tau \)). If \( \beta \) is a strict service curve, then it is a service curve, but the converse is not always true [4, Section 1.3]. A frequently used service curve is the rate-latency function \( \beta_{r,T} \) that is the function in \( \mathcal{F} \) defined by \( \beta_{r,T}(t) = r[t-T]^+ \), where we use the notation \( [x]^+ = \max\{x,0\} \). Saying that a system offers a service curve \( \beta_{r,T} \) to a flow expresses that the flow is guaranteed a service rate \( r \), except for possible interruptions that might impact the delay by at most \( T \). Saying that a system offers a strict service curve \( \beta_{r,T} \) to a flow expresses that the flow is guaranteed a service rate \( r \), except for possible interruptions that might not exceed \( T \) in total per backlogged period. A strict service curve \( \beta \) can always be assumed to be super-additive, i.e., to satisfy \( \beta(s + t) \geq \beta(s) + \beta(t) \) for all \( s, t \) (otherwise, it can be replaced by its super-additive closure [2, Prop. 5.6]).

Assume that a flow, constrained by arrival curve \( \alpha \), traverses a system that offers a service curve \( \beta \) to the flow and that respects the ordering of the flow (FIFO per-flow). The delay of the flow is upper bounded by \( h(\alpha,\beta) \) (horizontal deviation), defined by

\[
h(\alpha,\beta) = \inf_{t \geq 0} \{ \alpha(t) - \beta(t+d) \}
\]

Our technique of proof uses the lower pseudo-inverse. The lower pseudo-inverse \( f^\dagger \) of a function \( f \in \mathcal{F} \) is defined by

\[
f^\dagger(y) = \inf\{x | f(x) \geq y \} = \sup\{x | f(x) < y \}
\]

We use the following property from [6, Sec. 10.1]:

\[
\forall x, y \in \mathbb{R}^+, y \leq f(x) \Rightarrow x \geq f^\dagger(y)
\]

III. SYSTEM MODEL

We consider a weighted round-robin subsystem that serves \( n \) input flows, has one queue per flow, and uses a weighted round-robin algorithm (described later) to arbitrate between flows. The weighted round-robin subsystem is itself placed in a larger system, and can compete with other queuing subsystems. For example, consider the case of a constant-rate server with several priority levels, without preemption, and where the weighted round-robin subsystem is at a priority level that is not the highest, as in [14, Section 8.6.8.3]. Assuming some arrival curve constraints for the higher priority traffic, the service received by the entire weighted round-robin subsystem can be modelled using a strict service curve [2, Section 8.3.2].

This motivates us to assume that the aggregate of all flows in the weighted round-robin subsystem receives a strict service curve, say \( \beta \in \mathcal{F} \) that we call “aggregate strict service curve”. If the weighted round-robin subsystem has exclusive access to a transmission line of rate \( c \), then \( \beta(t) = ct \) for \( t \geq 0 \). We assume that \( \beta(t) \) is finite for every (finite) \( t \) and, without loss of generality, we assume \( \beta \) to be super-additive. Furthermore, we need an additional technical assumption, primarily for establishing the tightness result: we assume that \( \beta \) is Lipschitz-continuous, i.e., there exists a constant \( K > 0 \) such that \( \frac{\beta(t) - \beta(s)}{t-s} \leq K \) for all \( 0 \leq s < t \); this does not appear to be a restriction as the rate at which data is served has a physical limit.

The arbitration algorithm assumed in this paper is IWR, shown in Algorithm 1. When a packet of flow \( i \) enters the weighted round-robin subsystem, it is put into queue \( i \). The weight of flow \( i \) is \( w_i \). IWR runs an infinite loop of rounds. In one round, each queue \( i \) has \( w_i \) emission opportunities; one packet can be sent during one emission opportunity. The inner loop defines a cycle, where each queue is visited but only those with a weight not smaller than the cycle number have an emission opportunity. The send instruction is assumed to be the only one with a non-null duration. Its actual duration depends on the packet size but also on the amount of service
available to the entire weighted round-robin subsystem. See Figure 2 for an illustration.

<table>
<thead>
<tr>
<th>cycle 1</th>
<th>cycle 2</th>
<th>cycle 3</th>
<th>cycle 1</th>
<th>cycle 2</th>
<th>cycle 3</th>
<th>cycle 1</th>
<th>cycle 2</th>
<th>cycle 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
<td>1 2 3 1</td>
</tr>
</tbody>
</table>

Fig. 2: Emission opportunities on two successive rounds for IWRR with three flows and $w_1 = 2, w_2 = 3, w_3 = 5$. Mind that this is not the temporal behaviour: each opportunity can lead to an empty interval if the queue is empty at this time. Furthermore, the duration of each non-empty interval depends on the packet size and the aggregate service available (we do not assume constant rate service).

Algorithm 1 Interleaved Weighted Round-Robin

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>$w_{\text{max}} = \max{w_1, \ldots, w_n}$</td>
<td>Integer weights $w_1 \leq w_2 \leq \ldots \leq w_n$ are used.</td>
</tr>
<tr>
<td>2:</td>
<td>while True do</td>
<td>A round starts.</td>
</tr>
<tr>
<td>3:</td>
<td>for $C \leftarrow 1$ to $w_{\text{max}}$ do</td>
<td>A cycle starts.</td>
</tr>
<tr>
<td>4:</td>
<td>for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>5:</td>
<td>if $C \leq w_i$, then</td>
<td></td>
</tr>
<tr>
<td>6:</td>
<td>if (not empty(i)) then</td>
<td></td>
</tr>
<tr>
<td>7:</td>
<td>print(now, i);</td>
<td>A service for queue $i$.</td>
</tr>
<tr>
<td>8:</td>
<td>send(head(i));</td>
<td></td>
</tr>
<tr>
<td>9:</td>
<td>removeHead(i);</td>
<td></td>
</tr>
<tr>
<td>10:</td>
<td>end if</td>
<td></td>
</tr>
<tr>
<td>11:</td>
<td>end for</td>
<td>A cycle finishes.</td>
</tr>
<tr>
<td>12:</td>
<td>end if</td>
<td>A round finishes.</td>
</tr>
<tr>
<td>13:</td>
<td>end while</td>
<td></td>
</tr>
</tbody>
</table>

Here, we use the context of communication networks, but the results equally apply to real-time systems: Simply map flow to task, packet to job, packet size to job execution time and strict service curve to “delivery curve” [7], [8].

IV. STATE OF THE ART

One of the first use of round-robin scheduling in the network context appeared in [9], with a fairness objective, i.e., a fair way to share the bandwidth among sessions. It is also mentioned in [10] as a way to implement “fair queuing”.

The term “Weighted Round-Robin” was coined in [1] as a generalisation of round-robin to share the bandwidth “in proportion to prescribed weights” in the context of ATM (i.e., with constant-size packets). Two versions of the algorithm are presented in [1]. The former is presented in Algorithm 1: at cycle $C$ (with $C$ between 1 and $w_{\text{max}}$), only flows with weight $w_i \geq C$ can emit one packet. We call this version IWRR. The latter version assumes that there exists for each flow $i$ a bit-list of length $w_{\text{max}}$, $o_i \in \{0, 1\}^{w_{\text{max}}}$, such that $w_i = \sum_{k=1}^{w_{\text{max}}} o_i[k]$. A flow $i$ can emit a packet at cycle $C$ only if $o_i[C] = 1$. A strategy is given to build these vectors in [1] and is refined with fairness objectives in [11]. Call LIWRR (list-based IWRR) this version.

IWRR is modified into WRR/SB in [12] to enable some flow to send slightly more packets than permitted in a cycle, and to decrease accordingly at the next cycle.

As mentioned in Section I, plain WRR (which we simply call “WRR”) enables each flow $i$ to send up to $w_i$ packets every time it is selected [13]. A “Multiclass WRR” is also defined in [13]. Surprisingly, the authors of [13] were not aware of [1] and have re-invented LIWRR. Note that even if WRR was designed for packets of constant size, it has been applied in network of variable size packets such as Ethernet [14, Sec. 8.6, Sec. 8.6.8.3, Sec. 37], in request balancing in cloud infrastructures [15], in the Linux VirtualServer scheduling [16], in network of chip [17], and so on. In fact, looking for expression “weighted round-robin” in the title or abstracts of papers index by Scopus returns more than 400 entries (March 2020), and Google references more than 4000 patents with this expression (March 2020). Unfortunately, when authors refer to WRR, they often do not explicit which version of WRR it is.

A WRR server is also a latency-rate server, with latency and rates given in [18] for packets of constant size. The latency result is generalised to LIWRR in [19]. Even if the notion of latency-rate server is very close to the one of a service curve $\beta_{R,T}$ in network calculus, both notions are slightly different, and results cannot be directly imported from one theory to the other [20]. In [17], the authors consider a Network on Chip (NoC), with WRR arbitration at the flit level. A flit is the elementary data unit of the NoC, one flit is sent per CPU/NoC cycle. Assuming that the weights are such that packets are never fragmented by the arbiter, a strict service curve $\beta_{R,T}$ for flow $i$ is found, with $R_i = \frac{w_i}{\sum_k w_k}$, $T_i = \sum_{j \neq i} w_j$.

WRR arbitration in an Ethernet switch is also considered in [21], with the assumption that all flows of an output ports have the same constant packet size. It then computes, in the network calculus framework, a residual service with curve $\beta_{R,T}$, with $R_i = \frac{w_i}{\sum_k w_k}$, $T_i = \frac{\sum_{j \neq i} w_j}{\sum_k w_k}$, where $C$ is the link rate. We assume that the missing packet size in the $T_i$ term was a typo. This network calculus result on conventional WRR arbitration in Ethernet is refined in [22], considering packets of variable size, leading to residual service with strict service curve $\beta_{R,T}$, with $R_i = \frac{w_i}{\sum_k w_k}$, $T_i = \frac{\sum_{j \neq i} w_j}{\sum_k w_k}$ (cf. eq. (1) and (2) in [22]) where $\sum_{k} w_i^{\min}$ and $\sum_{k} w_i^{\max}$ are, respectively, lower and upper bounds on the size of the packets in the flow $i$. It refines this result by subtracting the part of the bandwidth not used by interfering flows (considering their arrival curves).

Observe that computing a residual service with a $\beta_{R,T}$ curve is pessimistic as it assumes that, once the worst latency is payed, each packet is served with the long-term residual rate. Whereas, in reality, each packet, when it is selected for emission, is transmitted at full link speed up to completion. A residual service for the conventional WRR with a curve that is an alternation of full services and plateaus is given in [2, Sec. 8.2.4]. This effect of “full speed up to completion” can also be captured when computing the local delay of a server.
with $\beta_{RT}$ service curve [23].

V. STRICT SERVICE CURVES FOR IWRR

Our first result is a strict service curve for IWRR that, as we show in Section VII, is the best possible. We compare it to WRR and also give simpler, lower approximations.

Theorem 1 (Strict Service Curve of IWRR). Let $S$ be a server shared by $n$ flows that uses IWRR as explained in Section III, with weight $w_i$ for flow $i$. Recall that the server offers a strict service curve $\beta$ to the aggregate of the $n$ flows. For any flow $i$, $l_i^{\min}$ [resp. $l_i^{\max}$] is a lower [resp. upper] bound on the packet size.

Then, $S$ offers to every flow $i$ a strict service curve $\beta_i$ given by $\beta_i(t) = \gamma_i(\beta(t))$ with

$$\gamma_i = \lambda_i \oplus U_i$$

$$U_i(x) \overset{def}{=} \sum_{k=0}^{w_i-1} \phi_{i,k}(x)$$

$$L_{\text{tot}} = w_i l_i^{\min} + \sum w_j l_j^{\max}$$

$$\psi_i(x) \overset{def}{=} \left\lfloor \frac{x}{w_i} \right\rfloor w_j + \lfloor y_i - w_i \rfloor^+$$

$$\phi_{i,j}(x) \overset{def}{=} \left\lfloor \frac{x}{w_i} \right\rfloor w_j + \lfloor y_i - w_i \rfloor^+$$

In the above, $\nu_{a,b}$ is the stair function, $\lambda_i$ is the unit rate function and $\oplus$ is the min-plus convolution, all are described in Fig. 1.

Furthermore, $\beta_i$ is super-additive.

The proof is in [3]. Fig. 3 illustrates how the strict service curve for IWRR improves on that for WRR, by providing a smoother, and generally larger, service.

The service curve found in Theorem 1 is the best possible one but has a complex expression. If there is interest in a simpler expression, any lower bounding function is a strict service curve; in particular, the strict service curve $\beta_i'$ for WRR is also a valid, though suboptimal, strict service curve for IWRR. There is often interest in service curves that are rate-lateny functions. Observe that, if the aggregate service curve $\beta$ is a rate-latency function, then replacing $\gamma_i$ by a rate-latency lower-bounding function also yields a rate-latency function for $\beta_i$, and vice-versa. Therefore, we are interested in rate-latency functions that lower bound $\gamma_i$.

Among all of such these, there is not a single best one, as some have a smaller latency while others have a larger rate. We say that a rate-latency function $\beta_{r,T}$ that lower bounds $\gamma_i$ is non-dominated if there is no other rate-latency function $\beta'_{r,T}$ that lower bounds $\gamma_i$ and dominates $\beta_{r,T}$, i.e., such that $r' \geq r$ and $T' \leq T$. The following result gives all such non-dominated rate-latency functions. Let $r^* = \min_{t \in \mathbb{N}} \{ w_i \}$ and $r_{w_i-1} = 1$, and

$$r_k = \psi_i(k l_i^{\min}) - \psi_i(k l_i^{\min}) \quad 0 \leq k < w_i - 1 \quad (11)$$

$$k^* = \min\{0 \leq k < w_i \mid r_k \geq r^*\} \quad (12)$$

$$r^*_{k+1} = \min_{t \in \mathbb{N}} \{ r_k, r^* \} \quad (13)$$

Theorem 3. With the assumptions in Theorem 1 and the definitions (11)-(13), a rate-latency function $\beta_{r,T}$ lower bounds $\gamma_i$ and is non-dominated if and only if $r = r^*_{k+1}$ and $T = \psi_i(k l_i^{\min}) - \frac{k l_i^{\min}}{r}$, or $r^*_{k+1} \leq r < r^*_{k}$ and $T = \psi_i(k l_i^{\min}) - \frac{k l_i^{\min}}{r}$ for some integer $k$ with $0 < k \leq k^*$. Among all such rate-latency functions, the one with lowest latency is $\beta_{r^*_{k+1},T}$ and the one with largest rate is $\beta_{r^*_{k+1},T}$.

The proof is in [3]. Fig 3 illustrates $\beta_{r^*_{k+1},T}$ and $\beta_{r^*_{k+1},T}$. in some examples. Observe that $k \mapsto r^*_{k}$ is wide-sense increasing with $k$ for $0 \leq k \leq k^*$, but the values of $r^*_{k}$ are not necessarily all distinct. It can also occur that $k^* = 0$ (as in the top panel of Fig. 3); in which case, there is one optimal rate-latency service curve. In general, however, this does not occur, and a simple lower bounding approximation can be obtained with the supremum of all non-dominated rate-latencies, which can be shown is equal to $\max \{ \beta_{r^*_{k+1},T} \}$.

VI. PROOF OF THEOREM 1

The idea of proof is as follows. We consider a backlog period [s, t] of flow of interest i, and we let $p$ be the number of packets of flow i that are entirely served during this period. For every other flow $j$, the number of packets that are entirely served is upper bounded by a function of $p$, given in Lemma 3. $p$ is also upper bounded by a function of the amount of service received by flow i in Lemma 5. Combining these two results
gives an implicit inequality for the total amount of service in (26). By using the technique of pseudo-inverse, this inequality is inverted and provides a lower bound for the amount of service received by the flow of interest.

A. Key Variables and Basic Properties

Let \( \{s,t\} \) be a backlogged period of flow \( i \). Let \( (\tau_k, fl_k) \) be couples of \((\text{instant}, \text{flow})\), printed at line 8 of Algorithm 1. Note that \( \tau_k < \tau_{k+1} \) as the send instruction has a non-null duration (because the aggregate service curve \( \beta \) is Lipschitz continuous). Let \( \sigma(0), \sigma(1), \ldots \) be the sequence of service opportunities for flow \( i \) at or after \( s \), i.e., \( \sigma(0) = \min\{m \mid \tau_m \geq s, fl_m = i\} \) and \( \sigma(k) = \min\{m \mid \tau_m > \tau_{\sigma(k)-1}, fl_m = i\} \). The 4th service opportunity for flow \( i \) occurs at time \( \tau_{\sigma(k-1)+1} \); we say that it is “complete” if \( \tau_{\sigma(k-1)+1} \leq t \), i.e., the interval taken by this service is entirely in \([s,t]\). Let \( p \geq 0 \) be the number of complete service opportunities. Observe that it is possible that \( p = 0 \), and it might happen that \( \tau_{\sigma(p)} < t \) or \( \tau_{\sigma(p)} \geq t \).

In each service of flow \( i \), during a backlogged period, it sends one packet with a length \( \geq l_{\text{min}}^i \), thus, for all \( k = 0 \ldots (p-1) \), \( R^i_\sigma(\tau_{\sigma(k)+1}) - R^i_\sigma(\tau_{\sigma(k)}) \geq l_{\text{min}}^i \), therefore

\[
R^i_\sigma(\tau_{\sigma(p)}) - R^i_\sigma(\tau_{\sigma(0)}) \geq pl_{\text{min}}^i \quad (14)
\]

B. Amount of Service to Other Flows

In order to upper bound the number of emission opportunities for another flow \( j \), we first find an expression, in Lemma 1, for the number of emission opportunities for flow \( j \) between two consecutive emission opportunities for flow \( i \). Lemma 2 then finds an upper bound on the number of emission opportunities for flow \( j \) in \((s, \tau_{\sigma(p)})\), as a function of the cycle number (variable \( C \) in Algorithm 1) at \( \tau_{\sigma(0)} \). Lastly, Lemma 3 maximizes the previous upper bound over all values of \( C \).

**Lemma 1.** The number of emission opportunities for flow \( j \neq i \) between two consecutive emission opportunities for flow \( i \), given that the latter emission opportunity for flow \( i \) occurs at cycle \( C \), is equal to \( q_{i,j}(C) \)

\[
\begin{cases} 
0 & \text{if } 1 < C \leq w_i \text{ and } w_j < C \\
1 & \text{if } 1 < C \leq w_i \text{ and } w_j \geq C \\
[w_j - w_i]^+ + 1 & \text{if } C = 1 
\end{cases} 
\quad (15)
\]

**Proof.** According to Algorithm 1, flow \( i \) has emission opportunities only in the first \( w_i \) cycles of each round. Both emission opportunities are either in the same round (Case 1) or in two consecutive rounds (Case 2). As \( C \) is the cycle number for the second emission opportunity for flow \( i \), Case 1 can occur only when \( 1 < C \leq w_i \), and Case 2 can occur when \( C = 1 \). For Case 1, we further differentiate between \( w_j < C \) and \( w_j \geq C \).

**Case 1a:** \( 1 < C \leq w_i \) and \( w_j < C \): Queue \( j \) does not have an emission opportunity in cycle \( C \) because \( w_j < C \). Also, we must have \( w_j < w_i \), thus queue \( j \) does not have an emission opportunity after \( i \) in cycle \( C - 1 \). Hence, \( q_{i,j}(C) = 0 \).

**Case 1b:** \( 1 < C \leq w_i \) and \( w_j \geq C \): If \( w_j > w_i \), then queue \( j \) has an emission opportunity after queue \( i \) in cycle \( C - 1 \). If
$w_j = w_i$, then queue $j$ has an emission opportunity before $i$ in cycle $C$, or after $i$ in cycle $C - 1$. Else, $C \leq w_j < w_i$ and queue $j$ has an emission opportunity in cycle $C$, before $i$. In all cases, $q_{i,j}(C) = 1$.

**Case 2:** $C = 1$: The first emission opportunity for $i$ is in the last cycle of a round that includes $i$ (cycle $w_i$). If $w_j > w_i$, then queue $j$ has an emission opportunity in the rest of cycle $w_i$ and also has emission opportunities during the next $(w_i - w_j)$ cycles of the last round. In this case, $q_{i,j}(C) = w_i - w_j + 1$, which is also the value in the last line of (15). Else if $w_j = w_i$, queue $j$ has an emission opportunity before $i$ in this cycle or after $i$ in cycle $w_i$ of the first round, thus $q_{i,j}(C) = 1$, which is also the value in the last line of (15). Else, $w_j < w_i$ and queue $j$ has an emission opportunity before $i$ in this cycle. Here too, $q_{i,j}(C) = 1$, the value in the last line of (15).

**Lemma 2.** The number of emission opportunities for flow $j \neq i$ in $(s, \tau_{(p)})$, for any backlogged period $(s, t)$ of flow $i$ with $p$ complete services, given that the first service starts at cycle number $C$ (cycle number at time $\tau_{(0)}$) is upper bounded by

$$q'_{i,j}(C, p) \overset{\text{def}}{=} \sum_{k=0}^{p} q_{i,j}((C + k - 1) \mod w_i + 1)$$

(16)

Also, let $C'(p)$ be the cycle number at $\tau_{(p)}$. Then,

$$C'(p) = (C + p - 1) \mod w_i + 1$$

(17)

**Proof.** By induction on $p$.

**Base Case:** $p = 0$

In this case, $q'_{i,j}(C, 0)$ is the number of emission opportunities for flow $j$ between two consecutive emission opportunities for flow $i$ that by Lemma 1, is equal to $q_{i,j}(C)$. As $1 \leq C \leq w_i$, $C - 1 \mod w_i + 1 = C$ thus $q_{i,j}(C)$, $q_{i,j}((C - 1) \mod w_i + 1)$. This shows (16). Also, by definition, $C'(0) = C$; using again $(C - 1) \mod w_i + 1 = C$ shows that (17) holds.

**Induction step:**

We assume that (16) and (17) hold for $p = 1$, and we want to show that they also hold for $p$.

First, let’s prove (17). There are two possible cases: (a) if $0 \leq C'(p - 1) < w_i$, then both $(p - 1)\text{th}$ and $p\text{th}$ emission opportunities occur in the same round, thus $C'(p) = C'(p - 1) + 1$. By the induction hypothesis, $(C + p - 2) \mod w_i + 1 < w_i$, i.e., $(C + p - 2) \mod w_i < w_i - 1$. Note that, for any integer $x$

$$(x+1) \mod w = \begin{cases} (x \mod w) + 1 & \text{if } (x \mod w) < w - 1 \\ 0 & \text{otherwise} \end{cases}$$

(18)

By using (18), we obtain that $C'(p)$ is given by (17) as required. (b) In the second case, $C'(p - 1) = w_i$ then the next emission opportunity occurs in the first cycle of the next round, thus $C'(p) = 1$. Here too, applying (18) shows that $C'(p)$ is given by (17) as required.

Then, we prove (16). Let $N$ be the number of emission opportunities for flow $j$ in $[s, \tau_{(p)}]$. $N$ is the sum of $N_1$, the number of emission opportunities in $[s, \tau_{(p-1)}]$, and $N_2$, the number of emission opportunities in $(\tau_{(p-1)}, \tau_{(p)})$. By the induction hypothesis, $N_1 \leq q'_{i,j}(C, p - 1)$. Also, by Lemma 1, we have $N_2 \leq q_{i,j}(C'(p))$. Thus, by using (17) which was just shown to also hold for $p$, we obtain

$$N \leq \sum_{k=0}^{p-1} q_{i,j}((C + k - 1) \mod w_i + 1) + q_{i,j}((C + p - 1) \mod w_i + 1)$$

(19)

where the right-hand side is equal to $q'_{i,j}(C, p)$ as required.

**Lemma 3.** For any backlogged period $(s, t)$ of flow $i$ with $p$ complete services, the number of emission opportunities for flow $j \neq i$ in $(s, \tau_{(p)})$ is upper bounded by $\phi_{i,j}(p)$, defined in (8).

**Proof.** Lemma 2 gives the number of emission opportunities for flow $j \neq i$ in $(s, \tau_{(p)})$, for any backlogged period $(s, t)$ of flow $i$ with $p$ complete services, when the first service starts at cycle number $C$ (cycle number at time $\tau_{(0)}$). To obtain the lemma, we maximize this result over $C$. We show the following properties.

**(P1)** For any integer $C \in [1, w_i]$, 

$$\sum_{k=0}^{w_i-1} q_{i,j}((C + k - 1) \mod w_i + 1) = w_j$$

(20)

The mapping $k \mapsto (C + k - 1) \mod w_i + 1$ is one-to-one from $\{0, \ldots, w_i - 1\}$ onto $\{1, \ldots, w_i\}$, thus the left-hand side of (20) is equal to $\sum_{k=0}^{w_i-1} q_{i,j}(k)$ that as we show now, is equal to $w_j$. First, we have $q_{i,j}(1) = [w_j - w_i]^{+} + 1$. Also, $q_{i,j}(k) = 1$ when $k > 1$ and $w_j \geq k + 1$. Thus, $\sum_{k=2}^{w_i-1} q_{i,j}(k) = \min(w_i - k, w_j - 1)$ and finally the left-hand side is equal to $[w_j - w_i]^{+} + \min(w_i - 1, w_j - 1) + 1$, which is equal to $w_j$.

**(P2)** For any integers $C \in [1, w_i]$ and $p \geq 0$, $q'_{i,j}(C, p) = \frac{p}{w_i} w_j + \sum_{k=0}^{p} w_i q_{i,j}((C + k - 1) \mod w_i + 1)$

(21)

$q_{i,j}$ is a periodic function with period $w_i$. By (P1), the sum over one complete period is $w_j$. Also, we can write $p = \frac{p}{w_i} w_i + p \mod w_i$. Thus, we have $\frac{p}{w_i} w_i$ complete rounds, and the sum in (21) is the remainder.

**(P3)** $q_{i,j}$ is a wide-sense decreasing function. This means that for any integer $k \in [1, w_i]$, $q_{i,j}(k + 1) \leq q_{i,j}(k)$. If $k = 1$, this follows from $q_{i,j}(1) \geq 1$ and $q_{i,j}(2) \leq 1$. Else if $k \leq w_i < k + 1$, then $q_{i,j}(k + 1) = 0$ and $q_{i,j}(k) = 1$. Else, they are equal. Hence, in all cases the property holds.

**(P4)** For any integer $C \in [1, w_i]$ and $p \geq 0$,

$$q'_{i,j}(C, p) \leq q'_{i,j}(1, p)$$

(22)

By using (P2), we should show that $\sum_{k=0}^{p} q_{i,j}((C + k - 1) \mod w_i + 1)$ is upper bounded by $\sum_{k=0}^{p} q_{i,j}(k \mod w_i + 1)$. Note that here we have $k \mod w_i = k$. Both sides are the sum of $a \overset{def}{=} p \mod w_i + 1$ unique elements of the set $\{q_{i,j}(k)\}_{k \in [1, w_i]}$. By (P3), the
right-hand side is the maximum sum of $a$ unique elements of this set.\\\[(P5)\text{ For any integer } p \geq 0, q_{i,j}(1,p) = \phi_{i,j}(p)\] (23)

We apply (P2) with $C = 1$ to compute $q_{i,j}(1,p)$. Then, the sum in the right-hand side of (21) is equal to $\sum_{k=0}^{p} \mod \omega_i q_{i,j} (k+1)$, as $k \mod \omega_i = k$. Then, by using the same argument after (20), it is equal to $(w_j - w_k)^{p} + \min(p \mod \omega_i, w_j - 1)$, which, by (8), is precisely $\phi_{i,j}(p)$.

The lemma then follows directly from (P4) and (P5). \hfill \Box 

Lemma 4. For every flow $j \neq i$, $R^t_j(t) \leq R^t_i(\tau_{ij}(p))$ (24)

Proof. If $t \leq \tau_{ij}(p)$, the result follows from $R^t_j$ being wide-sense increasing. Else, we have $t > \tau_{ij}(p)$; this implies that flow $i$ is served during $[\tau_{ij}(p), t]$; thus for any other flow $j$: $R^t_j(t) = R^t_j(\tau_{ij}(p))$.

C. Amount of Service to Flow of Interest

Lemma 5. The number of complete services, $p$, of flow of interest, $i$, in $(s,t]$ is upper bounded by:

$$p \leq \left\lfloor \frac{R^t_i(t) - R^t_i(s)}{p^{\text{min}}} \right\rfloor$$

(25)

Proof. First, $R^t_i(s) \leq R^t_i(\tau_{ij}(p))$ as $s \leq \tau_{ij}(p)$ and $R^t_i$ is wide-sense increasing. Second, consider the two cases in VI-A. If $t \geq \tau_{ij}(p)$, the property holds. Else, the scheduler in not serving flow $i$ in $[\tau_{ij}(p), t]$, thus $R^t_i(t) = R^t_i(\tau_{ij}(p))$. Hence, in both cases $R^t_i(t) \geq R^t_i(\tau_{ij}(p))$. By (14), $R^t_i(t) - R^t_i(s) \geq p^{\text{min}}$. Then, observe that $p$ is integer. \hfill \Box 

D. Total Amount of Service

Lemma 6. For any backlogged period $(s,t]$ of the flow of interest $i$, $\beta(t-s) \leq \psi_i(R^t_i(t) - R^t_i(s))$ (26)

where $\psi_i$ is defined in (7).

Proof. As the interval $(s,t]$ is a backlogged period, by the definition of the strict service curve for the aggregate of flows, $\beta(t-s) \leq \sum_j [R^t_j(t) - R^t_j(s)]$. We upper bound $R^t_j(t)$ for all $j \neq i$ by applying Lemma 4,

$$\beta(t-s) \leq \sum_{j \neq i} R^t_j(t) - R^t_j(s) + \sum_{j \neq i} R^t_j(\tau_{ij}(p)) - R^t_j(s)$$

(27)

Each flow $j$ has at most $\phi_{i,j}(p)$ emission opportunities during $[s, \tau_{ij}(p)]$ (Lemma 3) and can send at most one packet of maximum size in each. Thus, $\beta(t-s) \leq (R^t_i(t) - R^t_i(s)) + \sum_{j \neq i} \phi_{i,j}(p)p^{\text{max}}$

(28)

Also, Lemma 5 finds an upper bound on $p$. Therefore,

$$\beta(t-s) \leq (R^t_i(t) - R^t_i(s)) + \sum_{j \neq i} \phi_{i,j} \left( R^t_i(t) - R^t_i(s) \right) p^{\text{max}}$$

(29)

where the right-hand side is equal to $\psi_i(R^t_i(t) - R^t_i(s))$. \hfill \Box 

E. Lower Pseudo-inverse of $\psi_i$

Our next step is to invert (26) by computing the lower-pseudo inverse of $\psi_i$. As the calculus of pseudo inverses applies to wide-sense increasing functions, we first show:

Lemma 7. $\psi_i$, defined in (7), is wide-sense increasing.

Proof. It is sufficient to show that $\phi_{i,j}$, defined in (8), is a wide-sense increasing function. For any non-negative integers $x$ and $y$ such that $y \leq x$, we can write $x = kw_i + (x \mod w_i)$ and $y = k'm_i + (y \mod w_i)$, where $k, k'$ are non-negative integers. We must have $k \leq k'$. If $k = k'$, we know that $y \mod w_i \leq x \mod w_i$. Hence, $\phi_{i,j}(x) \leq \phi_{i,j}(y)$. Else, $k > k'$ and $\frac{x}{w_i} > \frac{y}{w_i}$. Thereby, $\phi_{i,j}(x)$ is at least one $w_i$ larger than $\phi_{i,j}(y)$. Hence, $\phi_{i,j}(y) < \phi_{i,j}(x)$. \hfill \Box 

Lemma 8. Let $g_0, g_1, \ldots, g_N$ be a non-negative sequence such that $g_{k+1} \geq g_k$. The sequence can be extended to a function in $\mathbb{F}$ by $g(x) = \inf(x \mod w_i)$ and let $g^2$ be its lower pseudo-inverse, so that $g^2(y) = k + 1 \in \mathbb{N} \Leftrightarrow g_k \leq y \leq g_{k+1}$. Define $f \in \mathbb{F}$ by $f(x) = x \mod w_i + 1$.

Proof. Observe that convolving $g^2$ with $\lambda_i$ consists in smoothing the unit steps with a slope of 1 (Fig. 1). Thus, $(\lambda_i \odot g^2)(y) = k + 1$ whenever $g_k \leq y \leq g_{k+1}$ and $(\lambda_i \odot g^2)(y) = k$ whenever $g_k + 1 \leq y \leq g_{k+1}$. Also, $f$ is piecewise linear and can be inverted in closed form on every interval where it is linear. A direct calculation gives $f^2(y) = k + y - g_k$ whenever $g_k \leq y \leq g_{k+1}$ and $f^2(y) = k + 1$ whenever $g_k + 1 \leq y \leq g_{k+1}$.

Lemma 9. Let $f \in \mathbb{F}$ and $m > 0$. Define $h \in \mathbb{F}$ by $h(x) = mf(x)$. Then, for all $y \geq 0$, $h^2(y) = mf^2(y)$.

Proof. Let $B(f,g) \overset{df}{=} (x \geq 0, h(x) \geq y)$ so that $f^2(y) = \inf B(f,y)$. Observe that $x \in B(f,y) \Leftrightarrow \frac{x}{y} \in B(f, \frac{y}{m})$.

Lemma 10. Let $a \in \mathbb{F}$ and $l > 0$. Define $b \in \mathbb{F}$ by $b(x) = l f(x)$. Then, for all $x \geq 0$, $b(x) = l \inf f(x)$.

Proof. Do the change of variable $u = lv$ in the expansion $(\lambda_i \odot b)(x) = \inf_{0 \leq s \leq x} (u + b(x - u))$ and obtain $(\lambda_i \odot b)(x) = \inf_{0 \leq s \leq l} (u + a(x - u)) = l \inf (\lambda_i \odot a)$.

We can now compute the lower-pseudo inverse of $\psi_i$. First, define the sequence $g$ by $g_k = \frac{1}{p^{\text{min}}} \inf \psi_i(kp^{\text{min}})$. As Lemma 8, $g$ can be extended to a piecewise constant function whose lower-pseudo inverse, $g^2$, can be directly computed:

$$g^2(x) = \frac{1}{p^{\text{min}}} \sum_{k=0}^{w_i-1} \int_{k}^{k+w_i} \min \left( \frac{x + \int_{\frac{k}{w_i}}^{\frac{k+w_i}{w_i}} \right)$$

(30)

Second, observe that for all $x \geq 0$, $\psi_i(x) = \psi_i\left( \frac{x}{p^{\text{min}}} \right) + x \mod p^{\text{min}}$. Define $f$ and $h$ from $g$ as in Lemmas 8 and 9 with $l = m = p^{\text{min}}$, so that $h = \psi_i$. Apply Lemmas 8 and 9 and obtain $\psi_i(x) = p^{\text{min}} (\lambda_i \odot g^2)(x)$.

Now apply Lemma 10 with $a = g^2$, $l = p^{\text{min}}$, and $b = U_i$ to obtain

$$\psi_i = \lambda_i \odot U_i$$

(31)
Proof of Theorem 1

Proof. Lemma 6 gives, in (26), an upper bound on the total amount of service as a function of the service received by the flow of interest. We invert (26) by the lower-pseudo inverse technique in (3) and obtain

\[ \bar{R}_i^*(t) - R_i^*(s) \geq (\lambda_i \otimes U_i) (\beta(t - s)) = \beta_i (t - s) \]  

(32)

Last, we need to prove that \( \beta_i \) is super-additive. This follows from the tightness result in Theorem 4 (the proof of which is independent of rest of this proof). Indeed, the super-additive closure \( \beta_i \) of \( \beta_i \) is also a strict service curve, and \( \beta_i (t) \geq \beta_i (t) \) for all \( t \in [2, \text{Prop. 5.6}] \). By Theorem 4, we also have \( \beta_i (t) \leq \beta_i (t) \) for all \( t \), hence \( \beta_i = \beta_i \). 

\[ \Box \]

VII. Tightness

We first show that the strict service curve we have obtained is the best possible. Proofs of results in this Section are in [3].

A. Tightness of Strict Service Curve

Theorem 4 (Tightness of the IWRR Service Curve). Consider a weighted round-robin subsystem that uses the IWRR scheduling algorithm, as defined in Section III, Assume the following system parameters are fixed: the number of input flows, the weight \( u_j \) allocated to every flow \( j \), the bounds on packet sizes \( b_{\min} \) and \( b_{\max} \) for every flow \( j \), and the strict service curve \( \beta \) for the aggregate of all flows. Let \( l \) be the index of one of the flows.

Assume that \( \beta_i \in \mathcal{F} \) is a strict service curve for flow \( i \) in any system that satisfies the specifications above. Then \( \beta_i \leq \beta_i \) where \( \beta_i \) is given in Theorem 1.

Interestingly, we obtain a similar result for WRR. Recall that \( \beta_i \) is the strict service curve for flow \( i \), described in (9), which was obtained in [2, Sec. 8.2.4].

Theorem 5 (Tightness of the WRR Service Curve). Theorem 4 is also valid if we replace IWRR with WRR. Specifically, using WRR as a scheduling policy, \( \beta_i \) is the largest possible strict service curve for flow \( i \).

B. Tightness of Delay Bounds With Constant Packet Sizes

Having obtained the best-possible strict service curve does not guarantee that the delay bounds derived from it are tight, i.e., are worst-case delays. This is because a service curve is only an abstraction of the system; and we have obtained a strict service curve, and non-strict service curves might provide better results. However, we show that, for flows of packets of constant size, we do obtain tight delay bounds. We show that it holds for IWRR and for WRR.

Recall that a delay bound requires the knowledge of an arrival curve \( \alpha_i \) for the flow of interest. If this flow generates only packets of length \( l \), then \( \alpha_i \) can be assumed to be a multiple of \( l \) and sub-additive. A delay bound for this flow is then equal to \( h(\alpha_i, \beta_i) \) (see (1)).

Theorem 6 (Tightness of Delay Bound for IWRR with Constant Packet Size). Consider a system, as in Theorem 4, with the additional assumption that, for the flow of interest, \( \alpha_i = \alpha_i \).

Let \( \alpha_i \in \mathcal{F} \) be a sub-additive function that is an integer multiple of \( l \), and assume that flow \( i \) has \( \alpha_i \) as arrival curve. The network calculus bound delay is tight, i.e., there exists a trajectory where the delay of one packet of flow \( i \) is equal to \( h(\alpha_i, \beta_i) \).

Theorem 7 (Tightness of Delay Bound for WRR with Constant Packet Size). Theorem 6 is also valid for the WRR policy.

VIII. Numerical Examples

To compare IWRR and WRR worst-case delays, we provide some numerical examples. First, we consider a system of 8 input flows \( f_1, \ldots, f_8 \) with respective weights \( \{22, 27, 28, 30, 34, 41, 45\} \) and \( b_{\min} = b_{\max} = l = 7119 \text{ bit} \). Let the aggregate service, \( \beta \), be a constant bit rate of 10 Mb/s. For every flow \( i \), we compute the IWRR and WRR strict service curves \( \beta_i^\prime \). Then, for every \( i \), we generate \( N = 1000 \) leaky-bucket arrival curves \( \gamma_k \), \( k = 1 \ldots N \), with rate \( r = 0.5 \text{ Mb/s} \) and burst \( b_k \) picked uniformly at random in \([1, 20]\) packets. Then, we use \( \alpha_i = \left\lfloor \frac{r}{b_k} \right\rfloor \) to satisfy the conditions of Theorems 6 and 7 and to compute \( d_{\text{IWRR}} = h(\alpha_i^\prime, \beta_i) \) and \( d_{\text{WRR}} = h(\alpha_i^\prime, \beta_i^\prime) \). Fig. 4 gives the box-and-whisker plots of the \( d_{\text{IWRR}} - d_{\text{WRR}} \) series. The median of WRR delay bounds \( d_{\text{WRR}} \) are also provided to illustrate the improvement.

Second, we repeated the same study for \( M = 10000 \) sets of system parameters. For each system, we choose the weights of 8 flows by picking them uniformly at random between 10 and 50, and we pick a packet length \( l \) uniformly at random between 64 to 1522 bytes. For each experiment, we call flow 1 the flow with the smallest weight, flow 2 with second smallest weight, and so on. As the scale of delay bounds depends on the choices of weights and the packet length, the \( d_{\text{IWRR}} - d_{\text{WRR}} \) series are divided by \( d_{\text{WRR}}^\prime \), the median of WRR delay bounds for flow \( i \). Fig. 4 gives the box-and-whisker plots of the \( \frac{d_{\text{IWRR}} - d_{\text{WRR}}}{d_{\text{WRR}}^\prime} \) series. Using IWRR improves worst-case delays, as expected, and the improvement is larger for flows with larger weights.

IX. Conclusion

IWRR is a variant of WRR with the same long-term rate and the same complexity. We have provided a residual strict service curve for IWRR and have showed that it is the best possible one under general assumptions. For flows with packets of constant size, we have showed that the delay bounds derived from it are worst-case. We have proved that IWRR worst-case delay is not greater than WRR and shown on experiments that the gain is significant (20%-60%) in practice, which speaks in favour of using IWRR as a replacement to WRR. Our result assumes that the aggregate of all IWRR queues receives a strict service curve guarantee, and we find a strict service curve guarantee for every IWRR queue. Therefore, our results apply to hierarchical schedulers. In future research, we plan to improve the results with supplementary hypotheses on flows, considering arrival curves and packet size distribution, with “packet curves” [24].
and

random in bounds, for several systems with weights picked uniformly at IWRR delay bounds normalized to the median of WRR delay bounds.

Box-and-whisker plots of difference between WRR and median of WRR Worst-case Delay

Absolute Improvement of Delay Bounds of IWRR wrt WRR on one Configuration

Flows

Fig. 4: Box-and-whisker plots of difference between WRR and IWRR delay bounds with weights \{22, 27, 28, 30, 34, 41, 45\} and \(I = 7119\) bit with random arrival curves. Median WRR delay bounds are also provided.

Relative Improvement of Delay Bounds of IWRR wrt WRR on a Set of Configurations

Flows

Fig. 5: Box-and-whisker plots of difference between WRR and IWRR delay bounds normalized to the median of WRR delay bounds, for several systems with weights picked uniformly at random in \{10, 50\}, assigned to flow by increasing order, and a packet length picked uniformly at random in \{64, 1522\} bytes.

REFERENCES
