Bachelor Thesis

Towards an Improved Performance Analysis of Deficit Round Robin in the Network Calculus

by

Vlad-Cristian Constantin

May 31, 2021

Technische Universität Kaiserslautern
Department of Computer Science
Distributed Computer Systems (DISCO) Lab

Supervisor: Prof. Dr.-Ing. Jens B. Schmitt
Examiner: M. Sc. Paul Nikolaus

Abstract

This paper investigates new methods of performance analysis of the bandwidth-sharing policy called Deficit Round Robin. We will conduct our reasoning by using Network Calculus and Numerical Evaluation. Deficit Round Robin is a practical implementation of the fair-queuing paradigm presented by Generalized Processor Sharing. Deficit Round Robin has been studied by several papers, one of which already proposed a leftover service curve. By means of algorithm study and network calculus, we introduce an improved leftover service curve for Deficit Round Robin. This improvement is later shown in the numerical evaluation chapter by comparing the delays corresponding to both approaches. We conclude that our newly proposed leftover service is better than in previous papers.

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Alle wörtlich oder sinngemäß übernommenen Zitate sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Kaiserslautern, den May 31, 2021

Vlad-Cristian Constantin

Contents

List of Figures

1 Introduction

Queues have been a social construct for centuries, due to their practicality and effectiveness. Since the early 1900s, it has also been used to model communication networks. A.K. Erlang has been a pioneer in this soon-to-be research area. With him working at the Copenhagen Telephone Exchange, Erlang soon pondered upon modeling telephone calls at the exchange in a rigorous mathematical manner.

Expanding upon Erlang's work in Queueing Theory, the scientific community took interest in this new field, with mathematicians leading the research efforts. Scientists like Felix Pollaczek, Aleksandr Yakovlevich Khinchin and Andrei Nikolajewitsch Kolmogorow have made tremendous progress in this research area, their focus being on congestion in practical contexts.

The foundations of queueing theory have been introduced in the 50s. The M/M/1 (Poisson arrival, exponential service, single server) queue has been one of the first challenges of this field, with the first solution being found in 1954 by Bailey. A more probabilistic approach was introduced by Kendall, also famous for his notation of queueing nodes. Later in the 60s, approximation methods have been examined, with results being shown by Kingman. He also suggested a formula for the mean waiting time in G/G/1 queues (G stands for general, i.e. random probability distribution). Many other techniques for analysis have been proposed since the 70s, one example being inversion of generating functions and Laplace transforms suggested by Abate and Dubner (1968).

However, all aforementioned advancements have been made in the stochastic analysis of the networked systems. A new type of analysis has been introduced by Cruz in 1991 [Cruz, 1991]. This is now known as deterministic network calculus and it deals in a non-probabilistic way with the analysis of network properties. Through this approach, one may perform a worst-case analysis. A good argument in favor of this method of analysis is that more often that not, providing an exact analysis for real systems is difficult, especially if one is considering the stochastic model. This is due to the fact that some assumptions needed for probabilistic approach are not always fulfilled, as in the correlation between arrival processes of different network nodes. Cruz et. al. thus introduce constraints on different parameters (e.g. delay) which need to be satisfied in order to analyze the corresponding system.

We start by introducing the Network Calculus Fundamentals in the first chapter. In Chapter 3 we present a scheduling algorithm called Generalized Processor Sharing (GPS) and the improved leftover service curve suggested in [Chang, 2000]. Deficit

Round Robin is the obvious next topic of discussion, since it is the most practical and efficient approximation of GPS. In Chapter 4 we propose an improved leftover service curve and a new definition for general resource allocation. In Chapter 5 a numerical evaluation is accomplished, in order to strengthen our claim that the newly presented service curve is indeed an improvement over the one suggested in [Boyer et al., 2012]. We finally conclude our findings and ponder upon next research opportunities with regard to our results.

2 Network Calculus Fundamentals

2.1 Min-Plus Algebra

In order to introduce and define an algebra, a rigorous mathematician would firstly require an algebraic structure, such as monoids or dioids. A classical example is the $(\mathbb{R}, +, \cdot)$ dioid. However, our algebra operates on $\mathbb{R} \cup \{+\infty\}$ with the minimum operation taking the addition's place and addition as our last operation, in comparison with the aforementioned dioid. The properties of the triple $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$ have proven its affiliation to the class of commutative dioids.

2.1.1 Min-Plus Convolution and Deconvolution

Firstly, we need to define the mathematical functions used throughout this paper:

Definition 2.1. (Set of Increasing Functions). We define $\mathcal{F}$ to be the set increasing function in $\mathbb{R}$ with $f(t) = 0$ for $t < 0$

$$\mathcal{F} := \{f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\} | \forall s \leq t : f(s) \leq f(t), \forall x < 0 : f(x) = 0\}.$$

In network calculus, a somewhat more preferable name for this type of functions is *wide-sense increasing*, since in our definition increasing does not refer completely to strictly increasing functions ($f(s) \leq f(t)$). As such, we define the set $\mathcal{F}_0$ with $f(0) = 0$ so

$$\mathcal{F}_0 := \{f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\} | \forall s \leq t : 0 \leq f(s) \leq f(t), f(0) = 0\}.$$

Definition 2.2. (Min-Plus Convolution). Let f and g be two functions in $\mathcal{F}$. The Min-Plus Convolution of these two function is defined as follows

$$f \otimes g(t) := \inf_{0 \leq s \leq t} f(t-s) + g(s).$$

If $t < 0, f \otimes g(t) := 0$.

Example 2.3. We consider the following functions and illustrate their convolution

$$\gamma_{r,b}(t) = \begin{cases} 0, & \text{if } t \leq 0 \\ b + r \cdot t, & \text{otherwise} \end{cases}$$

and

$$\beta_{R,T}(t) = R \cdot \max(0, t - T) = R \cdot [t - T]^+.$$

Here we used the function called *positive part*

$$[\cdot]^+ := x \rightarrow \max(0, x).$$

In [Schmitt and Nikolaus, 2020] it has been shown that the min-plus convolution of the two functions has the following form

$$\gamma_{r,b} \otimes \beta_{R,T}(t) = \begin{cases} 0, & \text{if } t \leq T \\ \min\{b + r \cdot (t - T), R \cdot (t - T)\}, & \text{otherwise.} \end{cases}$$

This is illustrated in Figure 2.1.

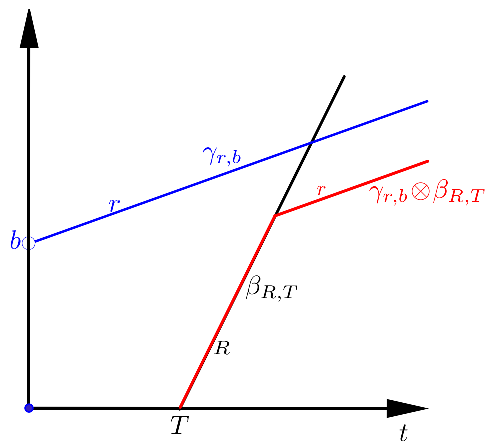


Figure 2.1: The convolution of $\gamma_{r,b}(t)$ and $\beta_{R,T}(t)$ [Schmitt and Nikolaus, 2020]

Definition 2.4. (Min-Plus Deconvolution). Let f and g be two functions in $\mathcal{F}$. The Min-Plus Deconvolution of these two function is defined as follows

$$f \oslash g(t) := \sup_{u \geq 0} \{f(t + u) - g(u)\}.$$

It has also been shown in [Schmitt and Nikolaus, 2020] that the deconvolution of our two functions from Example 2.3 follows the formula

$$\gamma_{r,b} \oslash \beta_{R,T}(t) = \gamma_{r,b+r \cdot t}(t).$$

This can be better visualized in Figure 2.2

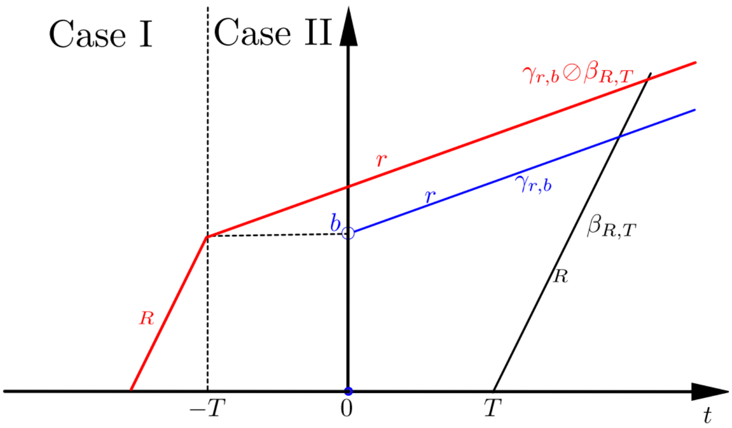


Figure 2.2: The deconvolution of $\gamma_{r,b}(t)$ and $\beta_{R,T}(t)$ [Schmitt and Nikolaus, 2020]

2.1.2 Vertical and Horizontal Deviation

Definition 2.5. Let $f, g \in \mathcal{F}$. The vertical and horizontal deviation of these two functions are defined as follows

$$v(f,g) := \sup_{t \geq 0}\{f(t) - g(t)\} = f \oslash g(0)$$

and respectively

$$\begin{aligned} h(f,g) &:= \sup_{t \geq 0}\{\inf\{d \geq 0 | f(t) \leq g(t+d)\}\} \\ &= \sup_{t \geq 0}\{\inf\{d \geq 0 | f(t-d) \leq g(t)\}\} \\ &= \inf\{d \geq 0 | \sup_{t \geq 0}\{f(t-d) - g(t) \leq 0\}\} \\ &= \inf\{d \geq 0 | f \oslash g(d) \leq 0\}. \end{aligned}$$

2.1.3 Arrival and Service Curves

Definition 2.6. (Arrival Process). The *arrival process* cumulatively counts the number of work units that arrive in a system $\mathcal{S}$ in the interval $[0, t)$ [Bouillard et al., 2018]. We can subsequently make the following assumptions of the function A

$$\begin{aligned} &A(0) = 0, \\ &A(s) \leq A(t), \text{ for all } s \leq t. \end{aligned}$$

Here we assume that A is left-continuous, since we can easily infer the discrete and continuous processes based on this property of A. For a better visualization, please refer to Figure 2.4.

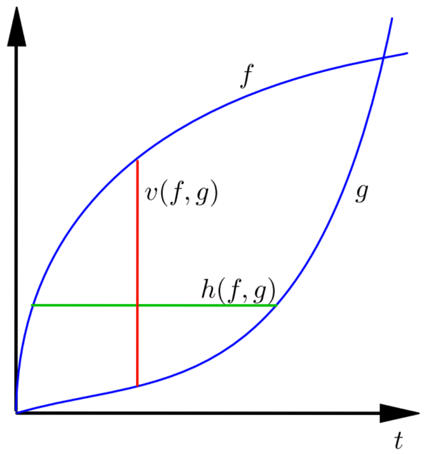


Figure 2.3: The vertical and horizontal deviations of two sample functions [Schmitt and Nikolaus, 2020]

Definition 2.7. (Departure Process) Similar to the arrival process, a *departure process* cumulatively counts the number of work units that depart from a system $\mathcal{S}$ in the interval $[0, t)$.

In a system $\mathcal{S}$ with both arrivals and departures, or better expressed, input and output, we assume the following property called *causality*

$$A(t) \geq D(t), \text{ for all } t \in \mathbb{R} \cup \{+\infty\}.$$

This implies that our system does not create work units and only those that entered the system can depart it. Furthermore, work units can also get lost, since we have "$\geq$" rather than "$=$".

We know assume a lossless system $\mathcal{S}$ in order to define the following two terms:

Definition 2.8. (Backlog at time t). The *backlog* of a system $\mathcal{S}$ is the vertical distance between arrival process A and departure process D at time t,

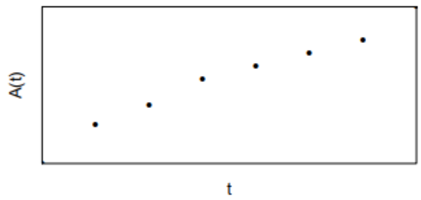
$$q(t) := A(t) - D(t).$$

Due to the causality property that we assume of any system $\mathcal{S}$, the relation $q(t) \geq 0$ always holds.

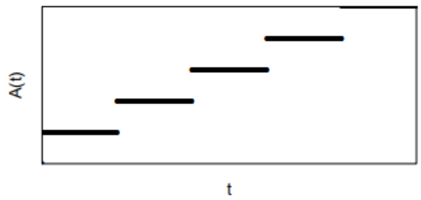
Definition 2.9. (Virtual delay at time t). The virtual delay of data arriving at system $\mathcal{S}$ at time t is the time until this data would be served, assuming FIFO order of service,

$$d(t) := \inf\{\tau \geq 0 : A(t) \leq D(t + \tau)\}$$

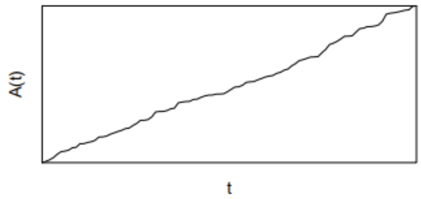
In Figure 2.5, the arrival and departure processes are illustrated along with backlog and virtual delay examples.



(a) Discrete-time random arrivals of the form $A(t) = \sum_{i=1}^{t} a_i$, where $a_i \geq 0$



(b) Left-continuous arrivals of the form $A(t) = \lceil x \rceil = \min\{n \in \mathbb{N} \mid n \geq x\}$



(c) Continuous random arrivals of the for $A(t) = \int_0^t a(x)\mathrm{d}x$, where $a(x) \geq 0$ for all x

Figure 2.4: Examples of different arrival processes A [Schmitt and Nikolaus, 2020]

Definition 2.10. (Arrival Curve). Given a function $\alpha \in \mathcal{F}_0$. We define α is an *arrival curve* for an arrival process A if for all $s \leq t$

$$A(t) - A(s) \leq \alpha(t - s).$$

Example 2.11.

- Peak-rate limit: $\alpha(t) = r \cdot t$
- Affine arrival curve:

$$\alpha(t) = \gamma_{r,b}(t) = \begin{cases} 0, & \text{if } t \leq 0 \\ b + r \cdot t, & \text{otherwise} \end{cases}$$

Definition 2.12. (Strict Service Curve). A system allegedly offers presents a *strict service curve* β to a flow if, during any (continuously) backlogged period (s, t), the following relation with regards to the output of the system holds

$$D(t) - D(s) \geq \beta(t - s).$$

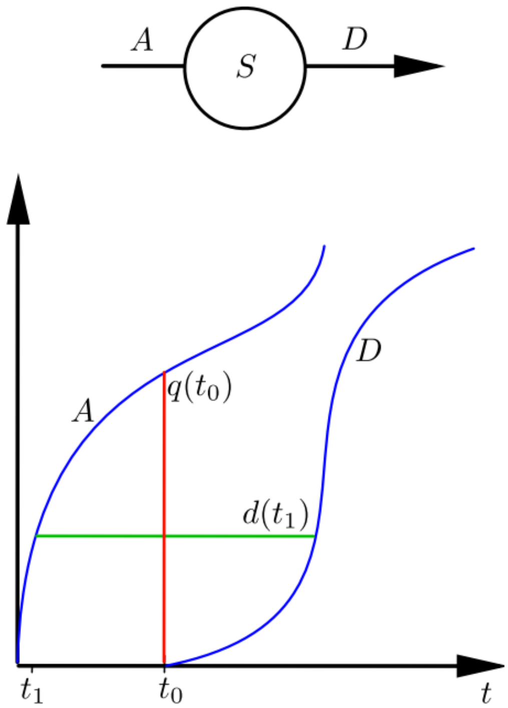


Figure 2.5: System $\mathcal{S}$ with with arrival process A; departure process D, backlog at time t_0, $q(t_0)$, and virtual delay at time t_1; $d(t_1)$ [Schmitt and Nikolaus, 2020]

Definition 2.13. (Service Curve). A system $\mathcal{S}$ is said to offer a (minimum) *service curve* β to A, which is an arrival process of a flow through $\mathcal{S}$, if β in $\mathcal{F}_0$ and for all $t \in \mathbb{R}$

$$D(t) \geq A \otimes \beta(t) = \inf_{0 \leq s \leq t} \{A(t-s) + \beta(s)\}.$$

Definition 2.14. (Rate-Latency Service Curve). The *rate-latency service curve* is defined as

$$\beta_{R,T}(t) := R \cdot [t-T]^+ = \begin{cases} R \cdot (t-T), & \text{if } t > T \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 2.15. *(Backlog Bound). We assume an arrival process A of a system $\mathcal{S}$. Furthermore, let A be constrained by an arrival curve α and $\mathcal{S}$ offer a strict service curve β. Then the following statement holds true for all t*

$$q(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\} = \alpha \oslash \beta(0) = v(\alpha, \beta).$$

Theorem 2.16. *(Delay Bound). We assume an arrival process A of a system $\mathcal{S}$. Furthermore, let A be constrained by an arrival curve α and $\mathcal{S}$ offer a strict service curve β. Then the following statement holds true for all t*

$$d(t) \leq \inf\{\tau \geq 0 | \alpha \oslash \beta(-\tau) \leq 0\} = h(\alpha, \beta).$$

Example 2.17. (Delay Bound for Token Bucket and Rate-Latency). We assume arrival curve $\alpha(t) = \gamma_{r,b}(t)$ and service curve $\beta(t) = \beta_{R,T}(t)$ satisfying the stability condition ($r \leq R$). We will prove that

$$h(\gamma_{r,b}, \beta_{R,T}) = \frac{b}{R} + T.$$

Proof.

$$\begin{aligned}
h(\gamma_{r,b}, \beta_{R,T}) &= \inf\{d \geq 0 | \gamma_{r,b} \oslash \beta_{R,T}(-d) \leq 0\} \\
&= \inf\{d \geq 0 | \gamma_{r,b+r\cdot t}(-d) \leq 0\} \\
&= \inf\{d \geq 0 | b + r \cdot T + r \cdot (-d) \leq 0\} \\
&= \inf\{d \geq 0 | b + r \cdot T \leq (-r) \cdot (-d)\} \\
&= \inf\{d \geq 0 | \frac{b}{r} + T \leq d\} \\
&= \frac{b}{R} + T
\end{aligned}$$

□

3 GPS Scheduling

An established scheduling algorithm, however of only theoretical significance, is Generalized Processor Sharing, or GPS [Parekh and Gallager, 1993]. It relies on the fluid model, which is not realistic for practical systems where we have packets and work units.

Definition 3.1. (GPS Resource Allocation [Parekh and Gallager, 1993]). Let n flows cross a GPS node. Thus, a fair allocation of resources in accordance with the weights $\phi_i \in \mathbb{R}_{>0}, i = 1, ..., n$. Namely, for any interval (s,t) in which a flow f_i is (continuously) backlogged, the following relation holds true:

$$\frac{D_i(t) - D_i(s)}{\phi_i} \geq \frac{D_j(t) - D_j(s)}{\phi_j} \text{ for all } j = 1, ..., n.$$

Theorem 3.2. *(GPS Leftover Service curve [Parekh and Gallager, 1993],[Schmitt and Nikolaus, 2020], [Bouillard et al., 2018]). Let a GPS node be traversed by n flows. Furthermore, let the node offer a strict service curve $\beta(t)$ to the aggregate of the flows $f_1, ..., f_n$. Then,*

$$\beta^i_{GPS}(t) := \frac{\phi_i}{\sum_{j=1}^n \phi_j} \cdot \beta(t).$$

3.1 An improved GPS Leftover Service Curve

Definition 3.3. ((σ, ρ) Constraint). A sequence A is (σ, ρ)-upper constrained if for all $0 \leq s \leq t$

$$A(t) - A(s) \leq \rho \cdot (t - s) + \sigma.$$

This definition was firstly introduced by Cruz et. al. [Cruz, 1991] and further applied by Chang in his work [Chang, 2000].

Definition 3.4. (Work Conserving Link [Schmitt and Nikolaus, 2020]) A link with capacity $C \geq 0$ is said to be work conserving if, during any backlogged period (s, t), the output of the system is equal to $C \cdot (t - s)$.

In order to calculate the next leftover service curve, we need to make the following assumptions: D_i, $i = 1, ..., n$ are (σ_i, ρ_i)-upper constrained, $\mathcal{M}$ is a nonempty subset of {1,2,...,n} and s_{i+1} is the beginning slot of the last busy period of the i^{th} input up to time t. We begin by summing the inequality in **Definition 3.1.** for all $j \in \mathcal{M}$

$$\begin{aligned}\sum_{j\in\mathcal{M}} \phi_j(D_i(t) - D_i(s)) &\geq \sum_{j\in\mathcal{M}} \phi_i(D_j(t) - D_j(s)) \\ &= \phi_i\left(C\cdot(t-s_i) - \sum_{j\notin\mathcal{M}}(D_j(t) - D_j(s)\right) \\ &\geq \phi_i\left(C\cdot(t-s_i) - \sum_{j\notin\mathcal{M}}(\rho_j(t-s_i)+\sigma_i)\right).\end{aligned}$$

Due to the above assumptions, we can further derive the next inequality

$$D_i(t) \geq A_i(s_i) + \frac{\phi_i}{\sum_{j\in\mathcal{M}}\phi_j}\left(\left(C - \sum_{j\notin\mathcal{M}}\rho_j\right)(t-s_i) - \sum_{j\notin\mathcal{M}}\sigma_i\right).$$

As our derivation works for all nonempty subsets $\mathcal{M}$, we continue as follows

$$\begin{aligned}D_i(t) &\geq A_i(s_i) + \max_{\mathcal{M}}\left\{\frac{\phi_i}{\sum_{j\in\mathcal{M}}\phi_j}\left(\left(C - \sum_{j\notin\mathcal{M}}\rho_j\right)(t-s_i) - \sum_{j\notin\mathcal{M}}\sigma_i\right)\right\} \\ &\geq \min_{0\leq s\leq t}\left\{A_i(s_i) + \max_{\mathcal{M}}\left\{\frac{\phi_i}{\sum_{j\in\mathcal{M}}\phi_j}\left(\left(C - \sum_{j\notin\mathcal{M}}\rho_j\right)(t-s_i) - \sum_{j\notin\mathcal{M}}\sigma_i\right)\right\}\right\}.\end{aligned}$$

Thus β_i is a service curve to A, where

$$\beta_i(t) = \max_{\mathcal{M}}\left\{\frac{\phi_i}{\sum_{j\in\mathcal{M}}\phi_j}\left(\left(C - \sum_{j\notin\mathcal{M}}\rho_j\right)(t-s_i) - \sum_{j\notin\mathcal{M}}\sigma_i\right)\right\}.$$

A more generalized formula for β_i was reported by Fidler [Fidler, 2010],

$$\beta_i = \max_{\mathcal{M}}\left\{\frac{\phi_i}{\sum_{j\in\mathcal{M}}\phi_j}\left(C\cdot t - \sum_{j\notin\mathcal{M}}F^j(t)\right)\right\},$$

where $F^j(t)$ denotes the departure envelope of flow j.

Example 3.5. Now we provide the improved service curve for the special case $\beta(t) = \beta_{R,T}(t)$ and $A_i = \gamma_{r_i,b_i}$.

Here we used the suggestion from [Fidler, 2010], where

$$\sum_{j \notin M} D_j(s_i, t) \leq \sum_{j \notin M} \left(A_j \oslash \left(\frac{\phi_j}{\sum_{k=1}^{n} \phi_k} \cdot \beta_{R,T}(s_i, t) \right) \right).$$

$$\begin{aligned}
\beta_i(t) &= \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta_{R,T}(t) - \left(\sum_{j \notin M} D_j(t) \right) \right) \right\} \\
&\geq \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta_{R,T}(t) - \left(\sum_{j \notin M} \left(A_j \oslash \left(\frac{\phi_j}{\sum_{k=1}^{n} \phi_k} \cdot \beta_{R,T}(t) \right) \right) \right) \right) \right\} \\
&\geq \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta_{R,T}(t) - \left(\sum_{j \notin M} \gamma_{r_j, b_j + r_j \cdot T}(t) \right) \right) \right\} \\
&= \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\beta_{R,T}(t) - \gamma_{\sum_{j \notin M} r_j, \sum_{j \notin M} b_j + \sum_{j \notin M} r_j \cdot T}(t) \right) \right\} \\
&= \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(R \cdot [t - T]^+ - \left(\left(\sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j \right) + \sum_{j \notin M} r_j \cdot t \right) \right) \right\} \\
&= \begin{cases} \max\limits_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(R \cdot t - R \cdot T - \left(\sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j + \sum_{j \notin M} r_j \cdot t \right) \right) \right\}, & \text{if } t > T \\ 0, & \text{otherwise,} \end{cases} \\
&= \begin{cases} \max\limits_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\left(R - \sum_{j \notin M} r_j \right) \cdot t - \left(R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j \right) \cdot \frac{R - \sum_{j \notin M} r_j}{R - \sum_{j \notin M} r_j} \right) \right\}, & \text{if } t > T \\ 0, & \text{otherwise,} \end{cases} \\
&= \begin{cases} \max\limits_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \left(\left(R - \sum_{j \notin M} r_j \right) \cdot \left(t - \frac{R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j}{R - \sum_{j \notin M} r_j} \right) \right) \right\}, & \text{if } t > T \\ 0, & \text{otherwise,} \end{cases} \\
&= \max_{M \subseteq \mathbb{M}} \left\{ \frac{\phi_i}{\sum_{j \in M} \phi_j} \cdot \beta_{\left(R - \sum_{j \notin M} r_j\right), \left(\frac{R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j}{R - \sum_{j \notin M} r_j} \right)}(t) \right\} \\
&= \max_{M \subseteq \mathbb{M}} \left\{ \beta_{\frac{\phi_i}{\sum_{j \in M} \phi_j} \cdot \left(R - \sum_{j \notin M} r_j\right), \left(\frac{R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j}{R - \sum_{j \notin M} r_j} \right)}(t) \right\}
\end{aligned}$$

3.2 Deficit Round Robin

Deficit Round Robin is a cheap and effective approximation method of fair queuing. It has been introduced by [Shreedhar and Varghese, 1995] in 1995. This method attains nearly perfect throughput fairness and takes $\mathcal{O}(1)$ processing work per packet. Thus, the implementation cost of this mechanism is very low. At that time, Deficit Round Robin (DRR) was considered the first approximation method which achieves near-perfect fairness with regard to throughput. Moreover, DRR is regarded as an efficient alternative implementation of the GPS paradigm. However, the latency of DRR is greater than P-GPS [Boyer et al., 2012].

Deficit Round Robin has seen an abundance of applications. For example, in the works of [Cicconetti et al., 2006] on 802.16, it has been utilized as the downlink scheduler. Furthermore, Scarlet [Ananthanarayanan et al., 2011] makes use of an approach similar to Deficit Round Robin for the spread of replicas of blocks based on popularity. Another system that makes use of Deficit Round Robin is Pisces [Shue et al., 2012]. This scheme attains per-tenant weighted fair shares of the aggregate resources of the shared service, by decomposing the fair sharing problem into four complementary methods, one of which is Deficit Round Robin.

3.2.1 Deficit Round Robin Algorithm [Shreedhar and Varghese, 1995]

In Figure 3.1 the algorithm of Deficit Round Robin is presented. In the next illustrations, Figures 3.2 and 3.3, Deficit Round Robin is exemplified:

```
Input: Per flow quantum: Q_1..Q_n (Integer)
Data: Per flow deficit: DC[1..n] (Integer)
Data: Counter: k (Integer)
for i= 1 to n do
    DC[i] ← 0 ;
end
while True do
    for i= 1 to n do
        if not empty(i) then
            // Print is a pseudo
               instruction, used to
               simplify the proof
            print (now(), i) ;
            DC[i] ← DC[i] + Q_i ;
            while (not empty(i)) and
            (size(head(i)) ≤ DC[i]) do
                send(head(i)) ;
                DC[i] ← DC[i] − size(head(i)) ;
                removeHead( i ) ;
            end
            if empty(i) then
                DC[i] ← 0
            end
        end
    end
end
```

Algorithm 1: DRR algorithm

Figure 3.1: Deficit Round Robin Algorithm [Boyer et al., 2012]

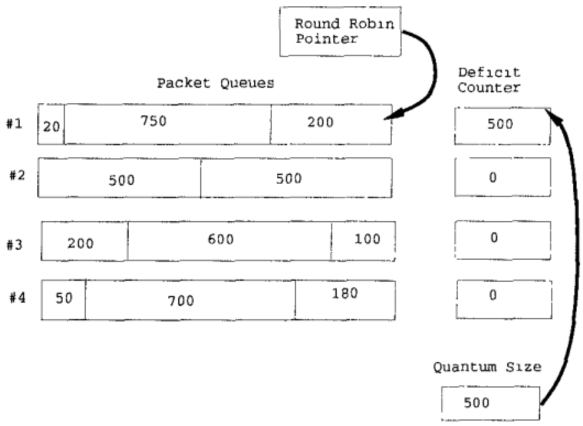


Figure 3.2: Deficit Round Robin: At the start, all the *Deficit Counter* variables are initialized to zero. The round robin pointer points to the top of the active list. When the first queue is serviced the *Quantum* value of 500 is added to the *Deficit Counter* value. The remainder after servicing the queue is left in the *Deficit Counter* variable [Shreedhar and Varghese, 1995]

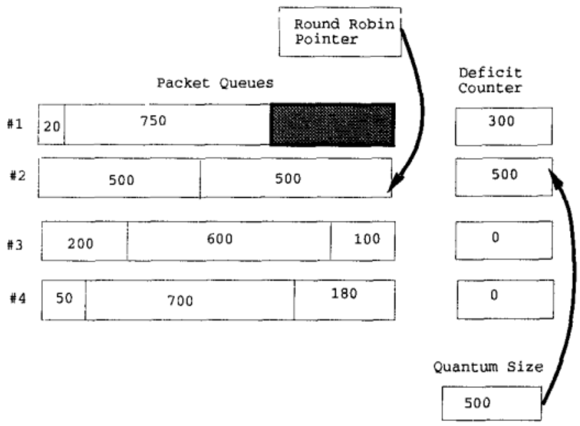


Figure 3.3: Deficit Round Robin (2): After sending out a packet of size 200, the queue had 300 bytes of its quantum left. It could not use it the current round, since the next packet in the queue is 750 bytes. Therefore, the amount 300 will carry over to the next round when it can send packets of size totaling 300 (deficit from previous round) + 500 (quantum).
[Shreedhar and Varghese, 1995]

3.2.2 Deficit Round Robin using Network Calculus

The most significant and relevant finding in the paper [Boyer et al., 2012] is the left-over service curve

Theorem 3.6. *(Leftover DRR Service Curve). Let S be a server, offering a strict service curve of super-additive curve β, shared by n flows. Each flow has a maximum packet size l_i^m and a Quantum Q_i. Then S offers to each flow a strict leftover service curve β_i^{DRR} defined as*

$$\beta_i^{DRR}(t) = \left[\frac{Q_i}{F}\beta(t) - \frac{Q_i(L - l_i^m) + (F - Q_i)(Q_i + l_i^m)}{F}\right]^+$$

with $F = \sum_{i=1}^n Q_i$ and $L = \sum_{i=1}^n l_i^m$.

Corollary 3.7. (DRR Leftover Rate-Latency Service Curve). We make the same assumptions as in **Theorem 3.6** with the addition of $\beta(t) = \beta_{R,T}(t)$. Then the service curve presented in **Theorem 3.6** is a rate-latency service curve

$$\beta_i^{DRR}(t) = R'\left[t - T'\right]$$

with $R' = \frac{Q_i}{F}R$, $T' = T + \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i + l_i^m)}{Q_i}}{R}$, $F = \sum_{i=1}^n Q_i$ and $L = \sum_{i=1}^n l_i^m$.

Proof.

$$\begin{aligned}
\beta_i^{DRR}(t) &= \left[\frac{Q_i}{F}\beta(t) - \frac{Q_i(L - l_i^m) + (F - Q_i)(Q_i + l_i^m)}{F}\right]^+ \\
&= \left[\frac{Q_i}{F}R\,[t - T]^+ - \frac{Q_i R(L - l_i^m)}{R\cdot F} - \frac{(F - Q_i)(Q_i + l_i^m)Q_i R}{R \cdot F \cdot Q_i}\right]^+ \\
&= \frac{Q_i}{F}R\left[[t - T]^+ - \frac{L - l_i^m}{R} - \frac{(F - Q_i)(Q_i + l_i^m)}{R \cdot Q_i}\right]^+ \\
&= \frac{Q_i}{F}R\left[[t - T]^+ - \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i + l_i^m)}{Q_i}}{R}\right]^+ \\
&= \begin{cases} \frac{Q_i}{F}R\left[t - \left(T + \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i + l_i^m)}{Q_i}}{R}\right)\right]^+, & \text{if } t > T \\ 0 = \frac{Q_i}{F}R\left[t - \left(T + \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i + l_i^m)}{Q_i}}{R}\right)\right]^+, & \text{otherwise} \end{cases} \\
&= \frac{Q_i}{F}R\left[t - \left(T + \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i + l_i^m)}{Q_i}}{R}\right)\right]^+
\end{aligned}$$

□

4 An improved DRR leftover service curve

Our focus is to derive an improved leftover service curve for DRR, by having an inequality similar to GPS Resource Allocation in **Definition 3.1**.

Definition 4.1. (General Resource Allocation). Let n flows traverse a node implementing a bandwidth-sharing policy. Then for the following weights $\phi_i \in \mathbb{R}_{>0}, i = 1, ..., n$ and for any interval (s,t) in which flow f_i is (continuously) backlogged, the following criterion is fulfilled

$$\phi_j D_i(s,t) \geq \phi_i \left(D_j(s,t) - H_{i,j} \right) \text{ for all } i \neq j. \tag{4.1}$$

Here we make use of the notation $D_i(s,t) = D_i(t) - D_i(s)$. We name $H_{i,j}$ as our penalty term.

To be noted that a similar work has been accomplished in [Bouillard, 2021]. Furthermore, it further strengthens the validity of our findings, since the methods used differ and yet our results are the same. However, our work has been done completely in parallel to Bouillard's.

Theorem 4.2. *(DRR Resource Allocation). For a flow i traversing a node implementing DRR and any backlogged period (s,t) corresponding to f_i, we have for all $i \neq j$*

$$Q_j D_i(s,t) \geq Q_i \left(D_j(s,t) - \left(\frac{l_i^m Q_j}{Q_i} + l_j^m + Q_j \right) \right),$$

where Q_j is the Quantum of flow j and l_j^m is the maximum packet size of flow j.

We can observe that this inequality corresponds to (4.1), with $\phi_i = Q_i$ for all flows and $H_{i,j} = \frac{l_i^m Q_j}{Q_i} + l_j^m + Q_j$.
By observing the code in Figure 3.1, we can draw the following conclusions:

- if flow i is continuously backlogged, line 15 in the algorithm is never executed and at least $pQ_i - l_i^m$ is served;
- for each other flow $j \neq i$, there are at most (p+1) executions of the loop lines 6-15 and at most $(p+1)Q_j + l_j^m$ is served.

By virtue of these observations, we can create two inequalities, one for the aforementioned backlogged flow i and every other flow.

Proof of ***Theorem 4.2****:*

$$
\begin{aligned}
D_i(s,t) &\geq pQ_i - l_i^m \\
D_j(s,t) &\leq (p+1)Q_j + l_j^m
\end{aligned}
$$

$$
\begin{aligned}
\frac{D_i(s,t)}{Q_i} &\geq \frac{D_j(s,t)}{Q_j} + p - \frac{l_i^m}{Q_i} - (p+1) - \frac{l_j^m}{Q_j} \\
&\geq \frac{D_j(s,t)}{Q_j} - \frac{\frac{l_i^m Q_j}{Q_i} + l_j^m + Q_j}{Q_j}
\end{aligned}
$$

□

The first inequality is used in the proof of the leftover service curve in [Boyer et al., 2012] and additionally in [Bouillard et al., 2018], page 198. The second inequality is utilized in [Bouillard et al., 2018, page 199]. Furthermore, these two inequalities hold true for two flows $i \neq j$ backlogged during (s, t).

The third inequality is formed by multiplying the first inequalities with $\frac{1}{Q_i}$ and with $\frac{1}{Q_j}$ respectively. Then we subtract the second inequality from first one.

4.1 Derivation of the improved leftover service curve

In order to calculate the improved leftover service curve, we employ the same technique as in **Chapter 3.1**, where we introduced $\mathcal{M}$ as a nonempty subset of {1,2,...,n}. Afterwards, we sum the inequality of **Theorem 4.2** for all $j \in \mathcal{M}$ and $j \neq i$

$$
\begin{aligned}
\sum_{j \in \mathcal{M}} Q_j D_i(s,t) &\geq \sum_{j \in \mathcal{M}} Q_i \left(D_j(s,t) - H_{i,j} \right) \\
D_i(s,t) \sum_{j \in \mathcal{M}} Q_j &\geq Q_i \sum_{j \in \mathcal{M}} \left(D_j(s,t) - H_{i,j} \right).
\end{aligned}
$$

Furthermore, we make the same assumptions as in **Chapter 3.1**, thus the derivation is analogous

$$
\begin{aligned}
D_i(s,t) &\geq \frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \sum_{j \in \mathcal{M}} \left(D_j(s,t) - H_{i,j} \right) \\
&\geq \frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \left(\beta(s,t) - \sum_{j \notin \mathcal{M}} D_j(s,t) - \sum_{j \in \mathcal{M}} H_{i,j} \right).
\end{aligned}
$$

Subsequently, the leftover service curve is defined as

$$
\beta_i(t) = \max_{M \subseteq \mathbb{M}} \left\{ \frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \left(\beta(t) - \sum_{j \notin \mathcal{M}} D_j(t) - \sum_{j \in \mathcal{M}} H_{i,j} \right) \right\}.
$$

Moreover, we make the same assumptions as in **Example 3.5**. The leftover service curve derivation is analogous with the one from the aforementioned example

$$
\begin{aligned}
\beta_i(t) &= \max_{M \subseteq \mathbb{M}} \left\{ \frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \left(\beta(t) - \sum_{j \notin \mathcal{M}} D_j(t) - \sum_{j \in \mathcal{M}} H_{i,j} \right) \right\} \\
&\geq \max_{M \subseteq \mathbb{M}} \left\{ \frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \left(\beta_{\left(R - \sum_{j \notin M} r_j\right), \left(\frac{R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j}{R - \sum_{j \notin M} r_j} \right)}(t) - \sum_{j \in \mathcal{M}} H_{i,j} \right) \right\}.
\end{aligned}
$$

Corollary 4.3. (Improved rate-latency service curve). The improved leftover service curve is a rate-latency service curve

$$
\beta_i(t) = \max_{M \subseteq \mathbb{M}} \left\{ \beta_{\frac{Q_i}{\sum_{j \in \mathcal{M}} Q_j} \cdot \left(R - \sum_{j \notin M} r_j\right), \left(\frac{R \cdot T + \sum_{j \notin M} r_j \cdot T + \sum_{j \notin M} b_j + \sum_{j \in \mathcal{M}} H_{i,j}}{R - \sum_{j \notin M} r_j} \right)}(t) \right\}.
$$

The proof is analogous to *Corollary 3.7*.

5 Numerical Evaluation

In order to illustrate that our leftover service curve is a legitimate improvement of the one presented in [Boyer et al., 2012], a numerical evaluation is required. In this chapter, we will calculate the virtual delays in a greedy/lazy scenario ([Fidler, 2010], [Burchard and Liebeherr, 2018] and [Le Boudec and Thiran, 2001]). Subsequently, we will compare the virtual delays created using the service curves presented by boyer and by our new approach. Moreover, we will simulate DRR under the same conditions (greedy/lazy scenario). The delays will be then compared through plots.

5.1 DRR Simulation

We start by introducing our simulation of DRR. As stated before, our input is greedy and our output lazy. Due to the scope of this paper, this being a worst-case analysis, our main focus during the simulation is to create the worst-case conditions for our flow of interest(foi). This can be achieved by having the foi at the end of the list of flows. Furthermore, our focus is not entirely on the flow of interest, but on the cross-flows as well (second to last one in the list).

Our simulation is a discrete-time-based one. Thus, during every second of it, we would create arrivals and departures at each flow. We assigned a Quantum to each flow. DRR also necessitates a maximum packet size, which we set for each flow at the value of 1.

Firstly, since our input is greedy, we chose to model our arrivals using the token-bucket arrival curve from **Example 2.11**. So at second 0, a burst of packets of size 1 will arrive at each flow i corresponding to its b_i. At second 1, we have our normal arrivals matching the rate r_i each flow. Since we want to execute our simulation for n seconds, we have a parameter *time*, so we would not simulate further. Thus, the infinite while loop at line 4 in Figure 3.1 will be replaced to accommodate our design of the problem. We then simulate our departures based on DRR. We make use of a time counter for two reasons: first one regarding time of simulation is mentioned above, the second one is to make sure that at each second, server (constant rate server, or rate-latency server with latency 0) rate departures are executed (lazy output). When the server, at second i for example, already served a number equal to server rate of packets, then we increment by 1 our time counter and continue serving the flows. Naturally, we also create arrivals each second. If our time counter exceeds the time parameter of our simulation function, then we stop.

Our design also includes FIFO per flow, so each flow is modeled as a Queue. During our simulation, we track to total number of units (packets) sent (which is compared as aforementioned with the rate of the server). We also track the packets arriving and departing the flow of interest and a cross-flow (second to last one in the list of flows). At the end, our simulation returns the maximum packet delay of flow of interest and cross-flow.

For the execution of our simulation function, we chose our parameters to be on integer type (quanta, arrival rates, bursts etc.).

One very important thing to note is that by implementing the greedy/lazy scenario, the virtual delay is equal to the horizontal deviation [Schmitt and Nikolaus, 2020, page 47]. Moreover, since we elected a discrete-time-based simulation, for each flow, the horizontal deviation is equal to the maximum packet delay.

5.2 Virtual Delay Calculations

Next, we will calculate the virtual delays corresponding to leftover service curves presented by Boyer [Boyer et al., 2012] and in our paper (Corollaries 4.3 and 3.7). Since our service curves have rate-latency characteristic, we can compute the horizontal deviation by using the formula presented in **Example 2.17**. Again, the greedy/lazy scenario plays an important role here, since the delay bound from **Example 2.17** is no longer a bound, but exact [Schmitt and Nikolaus, 2020, page 47].

We start with the leftover rate-latency service curve presented in Corollary 3.7. Thus, the virtual delay of flow i with arrival curve γ_{r_i,b_i} and service curve, naturally under the stability condition, is then

$$d(t) = \frac{b_i}{\frac{Q_i R}{F}} + T + \frac{(L - l_i^m) + \frac{(F-Q_i)(Q_i+l_i^m)}{Q_i}}{R}$$

where $F = \sum_{i=1}^{n} Q_i$, $L = \sum_{i=1}^{n} l_i^m$ and T=0 (as modeled in our simulation).
This delay will be calculated and plotted against various parameters (burst of flow of interest/cross-flow, utilization, quanta of burst of interest).

An analogous approach can be employed for our newly derived leftover rate-latency service curve from Corollary 4.3

$$d(t) = \frac{b_i}{\frac{Q_i}{\sum_{j\in\mathcal{M}} Q_j} \cdot \left(R - \sum_{j\notin M} r_j\right)} + \frac{R \cdot T + \sum_{j\notin M} r_j \cdot T + \sum_{j\notin M} b_j + \sum_{j\in\mathcal{M}} \left(H_{i,j} \mathbb{1}_{\{i\neq j\}}\right)}{R - \sum_{j\notin M} r_j},$$

where we used the indicator function to illustrate that $H_{i,i} = 0$ and further note that for our numerical evaluation the latency $T = 0$.

We now need a heuristic to calculate an efficient subset M, so that our service-curve is greater than the one derived by Boyer et.al., or so that the our delay is smaller than the one calculated from [Boyer et al., 2012]. We choose subsequently to sort the list of bursts in a descending order (for running time reasons) and to pick those flows in M, whose bursts are greater than equal the maximum burst. However, we do not conclude that this method of choosing the subset is optimal and the question of finding an optimum remains open.

5.3 Delay Comparison

As stated before, we conclude our findings through numerical evaluation by virtue of graphs. Based on each plot, we can make the following observations:

- Figure 5.1:
  - the delay grows as the utilization grows. Here we varied the server rate R and calculated the utilization as the sum of the arrival rates of the flows divided by the server rate. This behavior is expected and illustrated in the figure by all of our metrics (simulation and service curves of DRR, the one from Boyer et.al. and the improved one);
  - as stated earlier, the delay from our new leftover service curve is an obvious improvement over the one from [Boyer et al., 2012];
  - the results of our simulation may seem misleading, due to the delay always being an integer. However, this would be only an approximation of the delay. Were the delay overestimated, it would still be smaller than the delay from Boyer et.al. Were it underestimated (by a value strictly smaller than 1), the delay would be nevertheless smaller than the "Boyer" one, thus warranting a search for a better leftover service curve.
- Figure 5.2:
  - this growth of the delay with the increase of burst of flow of interest is anticipated and shown in this Figure. Moreover, the fact that the delay calculated using the new service curve is smaller than the one derived from the service curve in [Boyer et al., 2012] is perfectly illustrated;
  - the overestimation of the delay from the simulation can be analogously stated for this figure. However, if the simulation underestimates the delay (let's say, worst case by 1-ϵ), then the delay from our experiment will be greater than the one from "Boyer". Nonetheless, this would be because of poor heuristic utilization and not as a result of our new service curve not

being an improvement over the old one. This argument can be used for all other figures, where underestimation of the simulation delay may cause problems.

- Figure 5.3:
  - in this Figure, we chose to show the behavior of five variations of burst of flow of interest. This illustrates that the expectation of the improvement of delay is fulfilled, even for small variations of burst (for integer values);
  - the argument in case of a problematic under-evaluation of the simulation delay is similar as the one above.
- Figure 5.4:
  - here we illustrate a very interesting fact: if $M = 1, 2, ..., n$, with 0 latency (of the rate-latency server service curve), then the service curve presented by Boyer et.al. and ours are equal

$$\begin{aligned}\sum_{j=1}^{n} H_{i,j}\mathbb{1}_{\{i\neq j\}} &= \sum_{j=1}^{n}\left(\frac{l_i^m Q_j}{Q_i} + l_j^m + Q_j\right)\mathbb{1}_{\{i\neq j\}}\\ &= \sum_{i\neq j}\left(\frac{l_i^m Q_j}{Q_i} + l_j^m + Q_j\right)\\ &= \frac{l_i^m}{Q_i}(F - Q_i) + (L - l_i^m) + (F - Q_i)\\ &= \frac{l_i^m}{Q_i}(F - Q_i) + \frac{(L - l_i^m)Q_i}{Q_i} + \frac{(F - Q_i)Q_i}{Q_i}\\ &= (L - l_i^m) + \frac{(Q_i + l_i^m)(F - Q_i)}{Q_i}.\end{aligned}$$

  Then by deriving analogously as in our two Corollaries (3.7 and 4.3), we can conclude that the two service curves are equal;
  - analogous argument in case of simulation delay being under-valued.
- Figure 5.5:
  - in the case of cross-flow delay versus utilization, we can conclude that our approach is more efficient, even if the difference in small between the two delays ("Boyer" and ours), for time-critical systems, it can be of a significant importance for the validity of our approach;
  - analogous argument in case of simulation delay being underestimated.
- Figure 5.6:

– here the behavior of the delay when increasing the Quantum of flow of interest is opposite as in Figure 5.1. We expect that with the increase of Quanta, the delay for flow of interest would sink, fact illustrated by both of our plots using the "old" and "newly-"presented approaches;

– similar argument in case of simulation delay being under-evaluated.

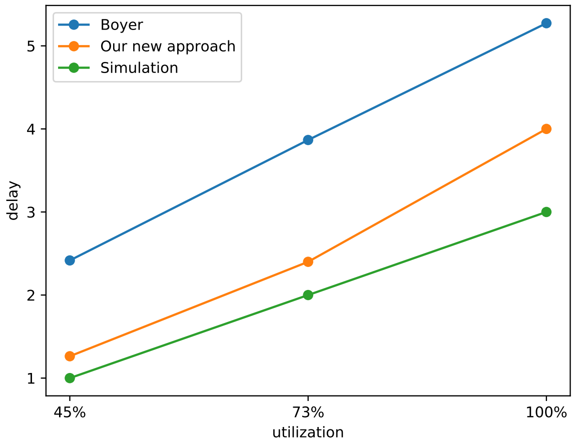


Figure 5.1: Delay versus utilization

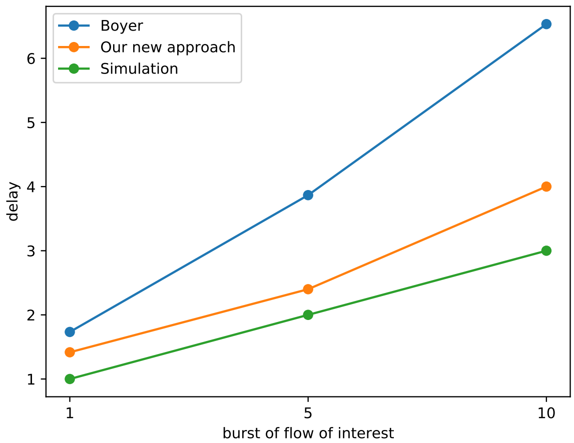


Figure 5.2: Delay versus burst variation of Flow of Interest

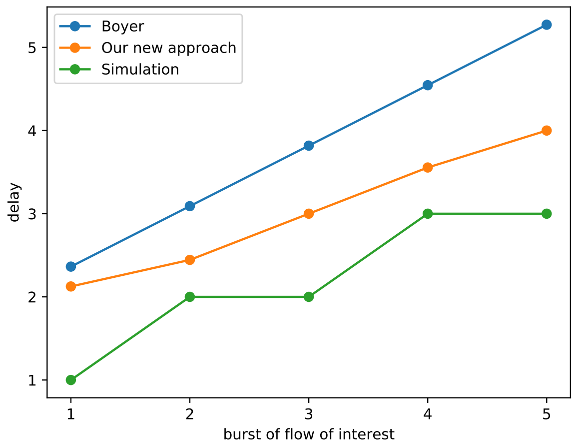


Figure 5.3: More bursts evaluated

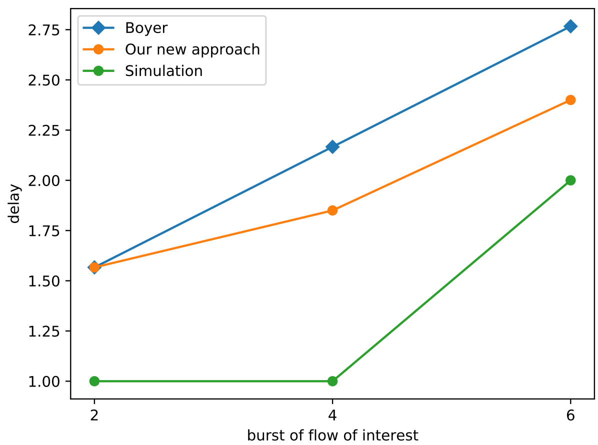


Figure 5.4: Special burst variation case

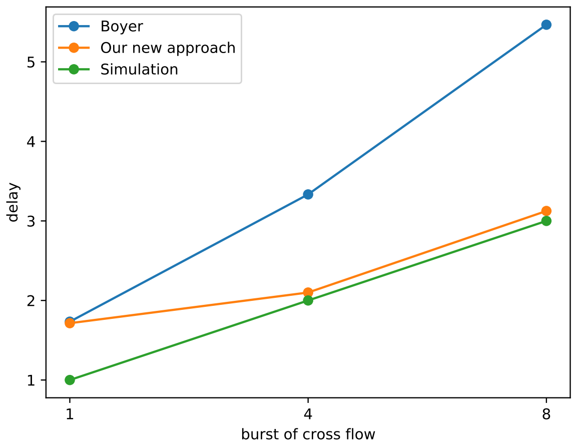


Figure 5.5: Delay versus burst variation of Cross-Flow

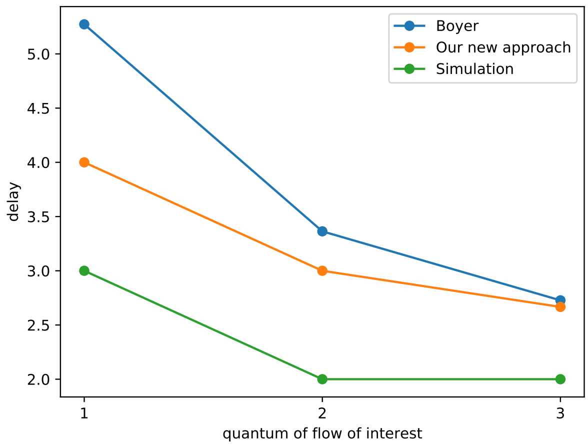


Figure 5.6: Delay versus quantum variation

6 Conclusion

In this paper we introduced a new leftover service curve for Deficit Round Robin. It, by direct comparison to the one from [Boyer et al., 2012], takes into account the arrival curves of the flows traversing the system. Furthermore, we introduced the General Resource Allocation for all bandwidth-sharing policies (including GPS). This definition helped us when looking for different policies to use our approach on, thus stumbling upon the two derived inequalities for DRR and calculating an improved leftover service curve. This improvement is better illustrated in our numerical evaluation chapter.

Our approach based on the suggestion in [Chang, 2000] looks more than promising. Since both leftover service curves can be presented as rate-latency, the corresponding delays are then mathematically and formally comparable. The finding of a heuristic which will calculate an optimal subset for our service curve remains an open question and it will certainly be the main focus of future research papers. Other possible studies may include the derivation of better service curves for other bandwidth-sharing policies, for instance Surplus Round Robin or Weighted Round Robin.

Bibliography

[Ananthanarayanan et al., 2011] Ananthanarayanan, G., Agarwal, S., Kandula, S., Greenberg, A., Stoica, I., Harlan, D., and Harris, E. (2011). Scarlett: coping with skewed content popularity in mapreduce clusters. In *Proceedings of the sixth conference on Computer systems*, pages 287–300.

[Bouillard, 2021] Bouillard, A. (2021). Individual service curves for bandwidth-sharing policies using network calculus. *IEEE Networking Letters*.

[Bouillard et al., 2018] Bouillard, A., Boyer, M., and Le Corronc, E. (2018). *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons.

[Boyer et al., 2012] Boyer, M., Stea, G., and Sofack, W. M. (2012). Deficit round robin with network calculus. In *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, pages 138–147. IEEE.

[Burchard and Liebeherr, 2018] Burchard, A. and Liebeherr, J. (2018). A general per-flow service curve for gps. In *2018 30th International Teletraffic Congress (ITC 30)*, volume 2, pages 31–36. IEEE.

[Chang, 2000] Chang, C.-S. (2000). *Performance guarantees in communication networks*. Springer Science & Business Media.

[Cicconetti et al., 2006] Cicconetti, C., Lenzini, L., Mingozzi, E., and Eklund, C. (2006). Quality of service support in ieee 802.16 networks. *IEEE network*, 20(2):50–55.

[Cruz, 1991] Cruz, R. L. (1991). A calculus for network delay. i. network elements in isolation. *IEEE Transactions on information theory*, 37(1):114–131.

[Fidler, 2010] Fidler, M. (2010). Survey of deterministic and stochastic service curve models in the network calculus. *IEEE Communications surveys & tutorials*, 12(1):59–86.

[Le Boudec and Thiran, 2001] Le Boudec, J.-Y. and Thiran, P. (2001). *Network calculus: a theory of deterministic queuing systems for the internet*, volume 2050. Springer Science & Business Media.

[Parekh and Gallager, 1993] Parekh, A. K. and Gallager, R. G. (1993). A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM transactions on networking*, 1(3):344–357.

[Schmitt and Nikolaus, 2020] Schmitt, J. and Nikolaus, P. (2020). Lecture notes in worst-case analysis of distributed systems.

[Shreedhar and Varghese, 1995] Shreedhar, M. and Varghese, G. (1995). Efficient fair queueing using deficit round robin. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 231–242.

[Shue et al., 2012] Shue, D., Freedman, M. J., and Shaikh, A. (2012). Performance isolation and fairness for multi-tenant cloud storage. In *10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, pages 349–362.