

Convexity and Optimization in Deficit Round Robin Scheduling for Delay-Constrained Systems

Aniket Mukherjee, Joy Kuri and Chandramani Singh

Abstract—The Deficit Round Robin (DRR) scheduler is widely used in network systems for its simplicity and fairness. However, configuring its integer-valued parameters, known as quanta, to meet stringent delay constraints remains a significant challenge. This paper addresses this issue by demonstrating the convexity of the feasible parameter set for a two-flow DRR system under delay constraints. The analysis is then extended to n -flow systems, uncovering key structural properties that guide parameter selection. Additionally, we propose an optimization method to maximize the number of packets served in a round while satisfying delay constraints. The effectiveness of this approach is validated through numerical simulations, providing a practical framework for enhancing DRR scheduling. These findings offer valuable insights into resource allocation strategies for maintaining Quality of Service (QoS) standards in network slicing environments.

Index Terms—Deficit Round Robin (DRR), Delay requirements, Convexity, Optimization.

I. INTRODUCTION AND RELATED WORK

Deficit Round Robin (DRR) [1], [2] is a popular scheduling algorithm known for its simplicity and fairness in handling multi-flow systems. It works by assigning a quantum to each flow, which determines the amount of data served in each scheduling round. This mechanism allows proportional allocation of service capacity based on flow-specific requirements. Despite its advantages, configuring DRR—particularly determining the optimal quantum values—remains challenging, especially in systems with diverse traffic patterns and strict delay requirements.

In this paper, we study a single-node communication system with n incoming data flows served by a constant-rate server of capacity c . Each flow is modeled using a bursty arrival envelope (leaky bucket) and is associated with specific delay requirements. Using a DRR scheduler, the goal is to allocate service efficiently while meeting these delay constraints, maintaining resource utilization, and minimizing the number of context switches between flows. This balance becomes increasingly difficult as the number of flows grow, underscoring the need for systematic quanta configuration.

Recent studies have focused on delay bounds and worst-case delay analysis for DRR schedulers [3]–[7] using Network Calculus [8] and their application to modern technologies such as network slicing and time-sensitive networking [9], [10]. While the approach Fuzzy Adaptive DRR (FADRR) [11] dynamically adjusts quantum values to meet QoS targets it lacks the precision required for strict delay guarantees. [12], [13] propose binary search-based heuristic algorithms to optimize bandwidth allocation. Our work builds on existing

delay-bound analyses, particularly the recent bounds derived in [4], to address the following fundamental questions:

- **Existence:** Can we find a vector of quanta that satisfies all delay requirements for multiple flows?
- **Optimization:** If such a vector exists, how can we determine the optimal quantum values that balance delay guarantees, minimize switching overhead, and maximize resource utilization?

Using a modified delay bound based on [4], we prove that the feasibility set of quanta is convex. This insight simplifies the optimization process, allowing for efficient computation of optimal quanta that minimize scheduling inefficiencies. By focusing on reducing the number of context switches while ensuring delay constraints are met, we strike a balance between system performance and fairness. The paper is organized as follows: Section II introduces the system model and provides the foundation for the discussions that follow. Section III presents the delay bound, the modified delay bound, and formulates the optimization problem addressed in this paper. Section IV establishes the properties of the optimization problem's constraint set, including conditions for non-emptiness and convexity. Section V proposes two algorithms to determine the optimal quanta values that meet the deadline and proves that the algorithms converge to the optimal solution and presents the simulation results, and Section VI summarizes the key contributions and findings.

II. SYSTEM MODEL

In this section, we present the system model for the problem of finding the optimal quanta of DRR scheduler to meet the delay requirements and also minimize the switching. Our focus is on a single-node system that operates with a constant-rate server, supporting multiple incoming data flows. Each flow exhibits bursty traffic characteristics, which are managed by DRR scheduling algorithm. We describe the system in detail, including the traffic model, the scheduling algorithm, and the key assumptions made to facilitate analysis.

A. System Setup

We consider a communication system where a constant-rate server, with service rate c , serves n incoming data flows. Each flow is indexed by $i \in 1, 2, \dots, n$. The data traffic for each flow is modeled using a bursty rate arrival envelope, commonly referred to as a leaky bucket model. This model allows us to capture both the burstiness and the rate-limited nature of traffic. The bursty behavior of flow i is parameterized by two parameters: the burst size b_i , which is the maximum burst size that the flow can generate, and the arrival rate r_i ,

which specifies the rate at which data arrives for this flow over time. The total data arrival for flow i over the time interval $(0, t)$ is thus bounded above by the following affine function: $b_i + r_i t$. The objective is to efficiently allocate the server's resources by determining the parameter of DRR scheduler for each flow while ensuring that their respective delay constraints are satisfied.

B. Scheduling: Deficit Round Robin (DRR)

Deficit Round Robin (DRR) is a scheduling algorithm commonly used in real-time systems and communication networks to allocate service among multiple queues or flows. Each flow i is assigned a fixed quantum q_i , which determines the amount of data it can transmit in each round of scheduling. The system maintains one queue per input and runs an infinite loop of scheduling rounds. In each round, if a queue is non-empty, its service begins, and its deficit is increased by q_i . The scheduler serves packets from the queue until the deficit is smaller than the size of the head-of-the-line packet or the queue becomes empty. If the queue becomes empty, the deficit is reset to zero; otherwise, the remaining deficit (residual deficit) is carried over to the next round. This ensures that flows receive service proportional to their assigned quantum.

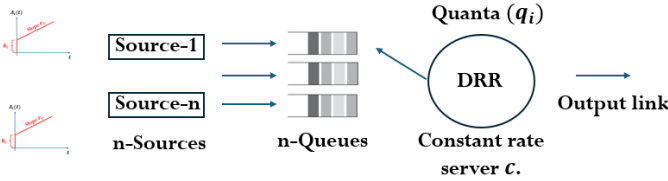


Fig. 1: The system as defined has n -sources, that are bounded by leaky bucket sources (parameterized with r_i, b_i), and a DRR-scheduler with constant rate c , such that the quanta are q_i^{th} for i^{th} queue.

C. Delay Requirements and the Impact of Quanta

Each flow in the system has a specific delay requirement denoted as d_i . The delay requirement represents the maximum permissible delay for the data flow to meet quality-of-service (QoS) constraints. We analyze how the quanta q_i assigned to each flow can affect the delay requirements and whether we can satisfy these requirements for all flows.

TABLE I: System Model

Input Parameters	
b_i	Burst i^{th} flow
r_i	Arrival rate i^{th} flow
d_i	Delay requirement i^{th} flow
L	Max residual deficit
c	Capacity of the server
D_i	Upper bound on the delay of i^{th} flow as per [4]
$\hat{D}_i$	Modified upper bound
Variable	
q_i	Quanta (DRR-parameter) for i^{th} flow

Therefore, to facilitate the analysis and make the problem more tractable, we introduce an modified upper bound on the

delay, which is simpler to analyze. The introduction of this upper bound reveals that the set of quanta satisfying the delay requirements form a convex set, which greatly simplifies the search for an optimal solution. All the parameters and variables are stated in Table I

III. DELAY BOUND ANALYSIS AND OPTIMIZATION PROBLEM

In this section we define the delay bound of i^{th} as presented in [4] and adapt it to our system. The delay bound for flow i is given by the following expression:

$$D_i = \max \left(\frac{\Psi_i(b_i)}{c}, \frac{\Psi_i(\alpha_i(\tau_i))}{c} - \tau_i \right), \quad (1)$$

where:

$$\tau_i = \frac{q_i - (b_i + L) \bmod q_i}{r_i}, \quad (2)$$

$$\Psi_i(x) = x + \sum_{j \neq i} \phi_{i,j}(x), \quad (2)$$

$$\phi_{i,j}(x) = \left\lfloor \frac{x + L}{q_i} \right\rfloor q_j + (q_j + L). \quad (3)$$

In the above equations, we define several important components: D_i represents the delay bound for flow i , which is the maximum delay the flow can experience. $\Psi_i(x)$ represents the total service demand for flow i , which includes the data generated by flow i itself as well as the contributions from other interfering flows. The term $\phi_{i,j}(x)$ is the contribution from flow j to the service demand of flow i , and it is computed based on the quantum assigned to flow i and the interference from other flows. $\alpha_i(\tau_i)$ is the arrival curve for flow i , which specifies the maximum number of bits that can arrive at flow i by time τ_i .

We propose a modified delay bound, $\hat{D}_i$. This modification simplifies the analysis while ensuring that the original delay bound D_i is always less than or equal to $\hat{D}_i$.

Lemma 1 (Modified Delay Bound). *Let D_i denote the delay bound as defined in [4]. Then, the modified delay bound $\hat{D}_i$ satisfies $D_i \leq \hat{D}_i$, where:*

$$\hat{D}_i = \frac{b_i + L}{c} \left(1 + \frac{\sum_{j \neq i} q_j}{q_i} \right) + \frac{\sum_{j \neq i} q_j}{c} + \frac{(n-2)L}{c} + \left(q_i \left(\frac{r_i - c}{r_i c} \right) + \frac{\sum_{j \neq i} q_j}{c} \right)^+. \quad (4)$$

Proof. See Appendix A of [14]. $\square$

The modified delay bound in Lemma 1 is met with equality when $(b_i + L) \bmod q_i = 0$. To simplify the analysis, we assume that the quanta q_i are real-valued, rather than integer-valued. This assumption simplifies the problem and enables us to leverage the structure of the modified upper bound, making it easier to investigate the convexity of the parameter set that satisfies the delay requirements, specifically ensuring that $\hat{D}_i \leq d_i$ for each flow.

It is important to note that the set of (q_1, q_2) that satisfy the original delay constraints, $D_i \leq d_i$, is not convex. To

demonstrate this, consider the following parameter values: $b_1 = b_2 = 10, r_1 = r_2 = 1, d_1 = d_2 = 1, c = 40, L = 3$. We compute the delay bounds for both flows using (1) at the points $(q_1, q_2) = (5, 9), (6, 10)$ and $(7, 11)$. The corresponding delay bounds are: $(D_1, D_2) = (1, 0.575), (1.075, 0.625)$ and $(0.875, 0.675)$. From these values, we observe that the points $(5, 9)$ and $(7, 11)$ satisfy the delay constraints, whereas the point $(6, 10)$ does not. This confirms that the set of (q_1, q_2) that satisfy the delay requirement $D_i \leq d_i$ is not convex.

Introducing the modified bound $\hat{D}_i$ is important because it makes the problem easier to handle. This change helps us to analyze the shape of the feasible set, which is a key step for solving the optimization problem.

From Lemma 1, we know that $D_i \leq \hat{D}_i \leq d_i$. This means that if the modified upper bound $\hat{D}_i$ satisfies the delay requirement d_i , then the original delay bound D_i will also be met. So, ensuring $\hat{D}_i \leq d_i$ is enough to guarantee that the delay requirement is satisfied.

We show that the set of quanta $q = (q_1, \dots, q_n)$ that satisfy the modified delay bounds for both flows, i.e., $\hat{D}_i \leq d_i$ for all $i \in \{1, \dots, n\}$ forms a convex set. To do this, we define function $f^i(q)$ that represent the deviations from the delay requirements for each flow i.e., $\{q > 0 | \hat{D}_i \leq d_i\} = \{q > 0 | f^i(q) \leq 0\}$.

The function $f^i(q)$, which represents the deviation for flow i , is given by:

$$f^i(q) = \frac{b_i + L}{c} \left(1 + \frac{\sum_{j \neq i} q_j}{q_i} \right) - d_i + \frac{\sum_{j \neq i} q_j}{c} + \frac{(n-2)L}{c} + \left(q_i \left(\frac{r_i - c}{r_i c} \right) + \frac{\sum_{j \neq i} q_j}{c} \right)^+ \quad (5)$$

where the terms are defined as before. We express the sets of quanta q that satisfy the delay constraints for each flow as:

$$\mathcal{D}^i = \{q > 0 | f^i(q) \leq 0\} \quad (6)$$

To maximize the number of packets served per round, our goal is to maximize the total quanta while ensuring that the delay constraints for each of the n flows are satisfied. This optimization problem is formulated as follows:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n q_i \\ & \text{subject to} && q \in \bigcap_{j=1}^n \mathcal{D}^j, \end{aligned} \quad (7)$$

where the feasible region is defined by the intersection of the constraint sets $\mathcal{D}^j$ for each flow $j = 1, 2, \dots, n$. These constraint sets determine the permissible values for the quanta of each flow, ensuring that the system adheres to the required delay constraints. The objective of this optimization problem is to maximize the total sum of all quanta, thereby maximizing the number of packets that can be served in each scheduling round while maintaining system efficiency and fairness.

IV. PROPERTIES OF CONSTRAINT SET

In this section, we establish necessary conditions to ensure that the feasible set of the optimization problem defined in (7) is non-empty we also demonstrate that the feasible set is convex. First, we derive a necessary condition for the existence of a feasible solution in the case of n flows. To prove convexity, we begin by analyzing the two-flow case, demonstrating that its feasible set is convex. We then extend this result to the general n -flow case, proving that the feasible set remains convex for any number of flows.

A. Necessary Conditions for a Non-Empty Feasible Set

In this subsection, we establish the necessary conditions for the feasible set to be non-empty. The feasible set is given by

$$\bigcap_{i=1}^n \{q > 0 | f^i(q) \leq 0\}. \quad (8)$$

From (5), we observe that the condition $f^i(q) \leq 0$ implies

$$\frac{b_i + L}{c} \left(1 + \frac{\sum_{j \neq i} q_j}{q_i} \right) - d_i + \frac{\sum_{j \neq i} q_j}{c} + \frac{(n-2)L}{c} \leq 0. \quad (9)$$

Using the implication in (9), we derive the necessary condition for the non-emptiness of the feasible set.

Lemma 2. *If the set $\bigcap_{j=1}^n \mathcal{D}^j$ is non-empty, then*

$$c \geq \sum_i \frac{b_i + L}{q_i}. \quad (10)$$

Proof. Let $q \in \bigcap_{j=1}^n \mathcal{D}^j$, which implies that for any $i \in \{1, \dots, n\}$,

$$\sum_{j \neq i} q_j \leq \frac{q_i (cd_i - b_i - (n-1)L)}{b_i + L + q_i}.$$

Furthermore, we have

$$\frac{\sum_j q_j}{q_i} + \frac{\sum_{j \neq i} q_j}{b_i + L} \leq \frac{cd_i}{b_i + L} - \frac{b_i + nL}{b_i + L}.$$

Since $\frac{\sum_{j \neq i} q_j}{b_i + L} > 0$ and $\frac{b_i + nL}{b_i + L} > 0$, we can rewrite the above inequality as

$$\frac{b_i + L}{cd_i} \leq \frac{q_i}{\sum_j q_j}.$$

We have n such inequalities for each $i \in \{1, \dots, n\}$. Summing all these inequalities, we obtain

$$\sum_i \frac{b_i + L}{cd_i} \leq c.$$

□

The implication of Lemma 2 is that it establishes the minimum bandwidth necessary to meet the system's deadline constraints. Specifically, the total capacity must exceed the sum of the clearing times necessary to process the bursts of all flows. This ensures that the system has sufficient resources to

handle the maximum load without violating timing constraints. By providing a clear criterion for bandwidth allocation, the lemma highlights the critical relationship between flow characteristics and system performance.

B. Convexity of feasible set: Two flow case

In this subsection, we demonstrate that the feasible set $\mathcal{D}^1 \cap \mathcal{D}^2$ is convex. From the definition of $f^i(q)$ in (5), we can express the same function for the two-flow case, specifically for flow 1:

$$f^1(q_1, q_2) = \frac{b_1 + L}{c} \left(1 + \frac{q_1 + q_2}{q_1} \right) - d_1 + \frac{q_2}{c} + \left(q_1 \left(\frac{r_1 - c}{r_1 c} \right) + \frac{q_2}{c} \right)^+. \quad (11)$$

To facilitate our analysis, we introduce the following auxiliary functions that decompose different components of (11):

- **Linear constraint term:**

$$a(q_1, q_2) = \frac{q_1(r_1 - c)}{r_1 c} + \frac{q_2}{c}, \quad (12)$$

This term appears inside the max function $(\cdot)^+$, determining when the second component in $f^1(q_1, q_2)$ is active.

- **When linear term is inactive ($a(q_1, q_2) \leq 0$):**

$$f^1(q_1, q_2) = \frac{b_1 + L}{c} \left(1 + \frac{q_2}{q_1} \right) + \frac{q_2}{c} - d_1, \quad (13)$$

This function represents the deviation when the affine term in (11) does not contribute, i.e., when $a(q_1, q_2) \leq 0$. Additionally, the feasibility condition associated with this case is governed by:

$$h_1^1(q_1) = \frac{q_1(cd_1 - b_1 - L)}{(b_1 + L) + q_1}, \quad (14)$$

This function represents an upper bound on q_2 when $f^1(q_1, q_2) \leq 0$.

- **When linear term is active ($a(q_1, q_2) > 0$):**

$$f^1(q_1, q_2) = \frac{b_1 + L}{c} \left(1 + \frac{q_2}{q_1} \right) + \frac{2q_2}{c} - d_1 + q_1 \left(\frac{r_1 - c}{r_1 c} \right), \quad (15)$$

This function represents the deviation when the affine term in (11) contributes, i.e., when $a(q_1, q_2) > 0$. The associated feasibility condition is given by:

$$h_2^1(q_1) = \frac{q_1^2(c - r_1) + q_1 r_1(cd_1 - b_1 - L)}{r_1(b_1 + L) + 2r_1 q_1}, \quad (16)$$

This function represents an upper bound on q_2 when the max term in f^1 is active.

The term cd_1 represents the total work completed by the server in the interval $(0, d_1)$, assuming a service rate of c . Meanwhile, $b_1 + L$ represents the work that can arrive instantaneously for the i^{th} flow. To ensure the delay target is met, the server must process more work than what has arrived. This requires $cd_1 \geq b_1 + L$, meaning that the delay target

is only achievable if the server has enough time to handle the initial worst-case burst. If this condition is not met, the burst cannot be fully processed within the required delay. By analyzing these functions, we can systematically establish the convexity of the feasible set $\mathcal{D}^1 \cap \mathcal{D}^2$.

We are interested in the inequalities $f_1^1(q_1, q_2) \leq 0$ and $f_2^1(q_1, q_2) \leq 0$, which can be written as:

$$q_2 \leq h_1^1(q_1), \quad (17)$$

$$q_2 \leq h_2^1(q_1). \quad (18)$$

The functions $h_1^1(\cdot)$ and $h_2^1(\cdot)$ are defined in (14), (16). These inequalities describe the feasible region for q_1 and q_2 .

We rewrite the set $\mathcal{D}^1$ based on the sign of $a(q_1, q_2)$. Since $a(q_1, q_2)$ is a linear, it partitions the space into two disjoint half spaces. Let us denote the half-space where $a(q_1, q_2) \leq 0$ as the set A :

$$A = \{(q_1, q_2) \in \mathbb{R}_+^2 \mid a(q_1, q_2) \leq 0\}. \quad (19)$$

From the definition, we know that both A and A^c are convex sets.

Let the sets B_1 and B_2 be defined as:

$$B_1 = \{f_1^1(q_1, q_2) \leq 0\} = \{q_2 \leq h_1^1(q_1)\}, \quad (20)$$

$$B_2 = \{f_2^1(q_1, q_2) \leq 0\} = \{q_2 \leq h_2^1(q_1)\}. \quad (21)$$

Therefore, the set $\mathcal{D}^1$ can be equivalently written as

$$\mathcal{D}^1 = (A \cap B_1) \cup (A^c \cap B_2), \quad (22)$$

The sets A , B_1 , and B_2 are all shown in the Figure 2. Before proving that $\mathcal{D}^1$ is a convex set, we will establish the following properties:

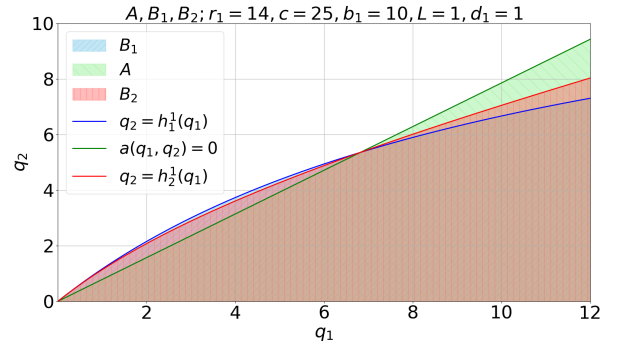


Fig. 2: The shaded region in blue, red and green shows the points that belong to the sets B_1, B_2 and A . The blue, red and green lines denote the equations $q_2 = h_1^1(q_1), q_2 = h_2^1(q_1), a(q_1, q_2) = 0$ respectively, with the parameter values given by $b_1 = 10, L = 1, C = 25, d_1 = 1$. The Brown region denotes the set $\{A \cup B_1\}$ and the small pink region denotes the set $\{A^c \cup B_2\}$.

Remark 1. For $a(q_1, q_2) \geq 0$, the following holds: $f_2^1(q_1, q_2) \geq f_1^1(q_1, q_2)$, $B_2 \subseteq B_1$, and $B_1 \cap B_2 = B_2$. For $a(q_1, q_2) \leq 0$, the implications reverse: $f_1^1(q_1, q_2) \geq f_2^1(q_1, q_2)$, $B_1 \subseteq B_2$, and $B_1 \cap B_2 = B_1$.

We will use the observations noted in remark 1 while proving the convexity of the set $\mathcal{D}^1$

Lemma 3. *The set $A^c \cap B_2$ is convex.*

Proof. See Appendix B of [14] $\square$

The set $A \cap B_2$ is empty if $d_1 r_1 \leq b_1 + L$, a condition utilized in the next theorem.

Lemma 4. *The set $\mathcal{D}^1$ is convex.*

Proof. Case 1:

Let $A^c \cap B_2 = \emptyset$. In this case, $\mathcal{D}^1 = A \cap B_1$, where the set A is a half-space and is convex. Since $h_1^1(q_1)$ is concave, the set $\{q_2 \leq h_1^1(q_1)\}$, denoted as B_1 , is convex. The intersection of two convex sets, A and B_1 , implies that $\mathcal{D}^1$ is convex.

Case 2:

When $A^c \cap B_2 \neq \emptyset$, we know that $(A^c \cap B_2)$ is a convex set and that $h_2^1(q_1)$ is a concave function. Therefore, using the remark 1 the set $\mathcal{D}^1$ can be written as:

$$\begin{aligned} \mathcal{D}^1 &= (A \cap B_1) \cup (A^c \cap B_2) \\ &= (A \cap B_1 \cap B_2) \cup (A^c \cap B_1 \cap B_2) \\ &= (B_1 \cap B_2) \\ &= (q_2 \leq H_1(q_1)), \end{aligned}$$

$$\text{where } H_1(q_1) = \min \{h_1^1(q_1), h_2^1(q_1)\}.$$

Since both $h_1^1(q_1)$ and $h_2^1(q_1)$ are concave functions, their pointwise minimum, $H_1(q_1)$, is also concave. Therefore, the set $\mathcal{D}^1 = \{q_2 \leq H_1(q_1)\}$ is convex. $\square$

Thus, we have shown that $\mathcal{D}^1$ is a convex set. By symmetry, $\mathcal{D}^2$ is also convex. Therefore, the intersection $\mathcal{D}^1 \cap \mathcal{D}^2$ is convex. To unify both cases, we update the definition of $H_1(q_1)$ as follows:

$$H_1(q_1) = \begin{cases} h_1^1(q_1) & \text{if } d_1 r_1 \leq b_1 + L, \\ \min \{h_1^1(q_1), h_2^1(q_1)\} & \text{if } d_1 r_1 > b_1 + L. \end{cases} \quad (23)$$

We now state some properties of the function $H_i(q_i)$:

Remark 2. *The function $H_i(q_i) : (0, \infty) \rightarrow (0, \infty)$, follow these properties mentioned below,*

- $\lim_{q_i \rightarrow 0} H_i(q_i) = 0$
- $H_i(q_i)$ is a concave, strictly increasing, continuous function.
- $\left. \frac{dH_i(q_i)}{dq_i} \right|_{q_i=0} = \frac{cd_i}{b_i+L} - 1$
- $\left. \frac{dH_i(q_i)}{dq_i} \right|_{q_i=\infty} = 0$

C. Convexity of feasible set: n flow case

In this subsection, we aim to demonstrate that the constraint set for the n -flow problem, which ensures that the delay requirements of all flows are satisfied, is a convex set. Convexity of the constraint set is essential as it allows the n -flow optimization problem to be framed as a convex

optimization problem. This reformulation enables the use of efficient algorithms to determine the optimal solution.

Theorem 1. *The constraint set for the n -flow problem, $\bigcap_{j=1}^n \mathcal{D}^j$, is a convex set.*

Proof. The proof proceeds by reformulating the delay constraints for each flow using auxiliary variables, simplifying the problem to a two-dimensional convex constraint set. Leveraging this structure, we verify the convexity of the reformulated set. Finally, using the properties of affine transformations and the intersection of convex sets, we conclude that the original constraint set is convex.

The set $\mathcal{D}^i$ as defined in (6) and the function $f^i(q)$ as defined in (5), we substitute $k_n^i = \sum_{j \neq i} q_j$ and $J_i = d_i - \frac{L(n-2)}{c}$ and reformulate the function $f^i(q_1, q_2, \dots, q_n)$ as:

$$\begin{aligned} f_k^i(q_1, k_n^i) &= \frac{b_i + L}{c} \left(1 + \frac{k_n^i}{q_i} \right) + \frac{k_n^i}{c} - J_i \\ &\quad + \left(q_i \left(\frac{r_i - c}{r_i c} \right) + \frac{k_n^i}{c} \right)^+. \end{aligned} \quad (24)$$

We define, $\hat{\mathcal{D}}^i = \{(q_1, k_n^i) \in \mathbb{R}_+^2 \mid f_k^i(q_1, k_n^i) \leq 0\}$. From Theorem IV-B, we know that $\hat{\mathcal{D}}^i$ is a convex set. Using an affine transformation to return to the original variables, we conclude that $\mathcal{D}^i$ is also convex [15].

Since the intersection of convex sets is also convex, $\bigcap_{i=1}^n \mathcal{D}^i$ is a convex set. $\square$

V. OPTIMIZATION PROBLEM

In this section, we propose two algorithms to solve the two flow and the n -flow case optimization problem as given in (7). The Two flow case optimization problem is given by,

$$\begin{aligned} &\text{maximize} \quad q_1 + q_2 \\ &\text{subject to} \quad (q_1, q_2) \in \mathcal{D}^1 \cap \mathcal{D}^2, \end{aligned} \quad (25)$$

A. Optimization: 2-flow case

We aim to maximize the amount of data served in one round and meet the delay requirements. Since q_1 represents the number of packets served per round for flow 1, maximizing the sum $q_1 + q_2$ increases the amount of packets served in a single round. The optimization problem defined in (25) is a convex optimization problem.

Lemma 5. *[Existence and Uniqueness of Fixed Point] The system of equations:*

$$H_1(q_1) = q_2, \quad (26)$$

$$H_2(q_2) = q_1. \quad (27)$$

have a unique solution (q_1^, q_2^*) , where $q_1^* > 0$ and $q_2^* = H_1(q_1^*)$.*

Proof. See Appendix C of [14] $\square$

A brief outline of the proof of Theorem 5 is as follows. Define $T(q_1) = H_2(H_1(q_1))$, a concave increasing function [16]. First, we analyze $T'(q_1)$ and find $T'(0) > 1$, ensuring $T(q_1)$ initially grows faster than q_1 but eventually slower as $T'(q_1)$ decreases. Second, we establish the existence of a fixed

point q_1^* such that $T(q_1^*) = q_1^*$, as $T(0) = 0$ and $T(q_1) < q_1$ for large q_1 . Finally, the concavity of $T(q_1)$ ensures the fixed point is unique by ruling out the possibility of multiple fixed points. Moreover, the fixed point q_1^* represents the equilibrium of the system described by $T(q_1)$.

Lemma 6. *The optimal solution to the optimization problem described in (25) is given by the pair of values that satisfy the following system of equations:*

$$\begin{aligned} H_1(q_1) &= q_2, \\ H_2(q_2) &= q_1. \end{aligned} \quad (28)$$

Proof. See Appendix D of [14] See $\square$

In this section, we have formulated a convex optimization problem aimed at minimizing the number of switchings between two flows by maximizing the sum of the quanta $q_1 + q_2$. The existence and uniqueness of the optimal solution were established, with the solution satisfying the system of equations $H_1(q_1) = q_2$ and $H_2(q_2) = q_1$. Through analysis, we showed that the optimal solution exists and is unique, providing a set of conditions under which this solution can be determined.

Theorem 2. *Consider the fixed-point iteration:*

$$x_{k+1} = T(x_k), \quad (29)$$

where the initial value $x_0 > 0$ is given. Then, the sequence $\{x_k\}$ generated by this iteration, converges to a unique fixed point $q_1^* > 0$, i.e.,

$$\lim_{k \rightarrow \infty} x_k = q_1^*. \quad (30)$$

Furthermore, the fixed point q_1^* satisfies the equation:

$$T(q_1^*) = q_1^*. \quad (31)$$

Proof. See Appendix E of [14] $\square$

Algorithm 1 Fixed-point iteration for finding q^* for two-flow case

- 1: **Input:** Initial guess $x_0 > 0$, tolerance $\epsilon > 0$, $T(x)$
- 2: **Output:** Fixed point q^*
- 3: **Initialize:** Set x_0
- 4: Set $k \leftarrow 1$
- 5: Set $x_1 \leftarrow T(x_0)$
- 6: **while** $|x_k - x_{k-1}| > \epsilon$ **do**
- 7: Compute $x_{k+1} \leftarrow T(x_k)$
- 8: Update $k \leftarrow k + 1$
- 9: **end while**
- 10: **Return** $q_1^* = x_k$; $q_2^* = H_1(q_1^*)$

In Figure 3, the trajectory of q_1 , as determined by the Algorithm 1, is shown to converge to the optimal value q_1^* . The specified parameters for this case are $b = [10, 15]$, $r = [1, 2]$, $d = [1, 0.5]$, $c = 50$, and $L = 1$. Starting with an initial guess of $x_0 = 1$, the algorithm successfully converges to the optimal solution, $(q_1^*, q_2^*) = (3.181, 8.749)$.

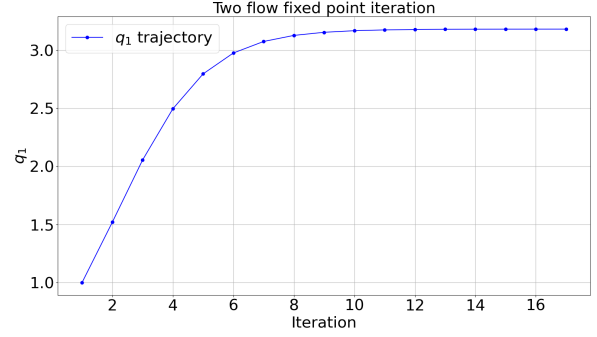


Fig. 3: Trajectory of q_1 under Algorithm 1, illustrating convergence to the optimal value $q_1^* = 3.181$. The parameters for this case are $b = [10, 15]$, $r = [1, 2]$, $d = [1, 0.5]$, $c = 50$, and $L = 1$, with an initial guess of $x_0 = 1$.

B. Optimization: n -flow case

We now extend the problem to the case of minimizing the number of switchings in the n -flow scenario, where the goal is to maximize the total quanta while satisfying the flow constraints for each of the n flows. The optimization problem is defined as follows:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^n q_i \\ &\text{subject to} && (q_1, q_2, \dots, q_n) \in \bigcap_{j=1}^n \mathcal{D}^j, \end{aligned} \quad (32)$$

where $q_1, q_2, \dots, q_n$ represent the quanta for each flow, and the feasible region is given by the intersection of the constraint sets $\mathcal{D}^j$ for each $j = 1, 2, \dots, n$. These constraint sets represent the feasible values for the quanta of each individual flow, ensuring the system operates within the required delay requirements, such as delay and throughput constraints for each flow. The objective of this problem is to maximize the sum of all quanta, which corresponds to minimizing the number of switching between the flows.

Since this is a convex optimization problem, it can be solved using any standard convex optimization solver. These solvers iteratively explore the feasible space, adjusting the quanta to maximize the objective function. The convexity of the problem guarantees that any local optimum is also a global optimum, ensuring that the solution found by such solvers is optimal for the given system.

We propose a fixed-point iterative method to find the optimal solution, as described below.

From (24), the constraint set can be expressed as

$$\sum_{j \neq i} q_j \leq H_i(q_i). \quad (33)$$

Since the function $H_i(q_i)$ is strictly increasing and concave, we define

$$g_i(q_i) = H_i(q_i) + q_i. \quad (34)$$

Thus, the constraint can be rewritten as

$$\sum_j q_j \leq g_i(q_i). \quad (35)$$

Since $g_i(q_i)$ is increasing and concave, let us assume $\sum_i q_i = \theta$. This leads to

$$\sum_i g_i^{-1}(\theta) \leq \theta. \quad (36)$$

Lemma 7. *There exists a unique $\theta^* > 0$, such that*

$$\sum_i g_i^{-1}(\theta^*) = \theta^* \quad (37)$$

Proof. Let, $\Gamma(\theta) = \sum_i g_i^{-1}(\theta)$. We know that $\Gamma(\theta)$ is an increasing and convex function. Also, we know that

$$\lim_{\theta \rightarrow 0} \Gamma'(\theta) = \frac{1}{c} \sum_j \frac{b_j + L}{d_j} \leq 1 \quad (38)$$

$$\lim_{\theta \rightarrow \infty} \Gamma'(\theta) = n > 1 \quad (39)$$

The inequality in (38) is from the non-empty condition given in Lemma 2. By intermediate value theorem, there exists at least one $\theta > 0$, such that $\Gamma(\theta) = \theta$. Let there exist two distinct fixed points θ_1 and θ_2 such that $\Gamma(\theta_1) = \theta_1$ and $T_2(\theta_2) = \theta_2$, WLOG we assume $\theta_1 < \theta_2$. $T(\theta)$ is strictly increasing $T(\theta_1) < T(\theta_2)$. The line connecting $(T(\theta_1), \theta_1)$ and $(T(\theta_2), \theta_2)$ is 1. By, mean value theorem there exist $c \in (\theta_1, \theta_2)$ such that $T'(c) = 1$ [17]. But, $T(\theta)$ is concave, $T(\theta)$ is increasing and for any $\theta < c$ the slope is less than 1, and it cannot intersect the line $y = x$ at θ_1 . This contradicts the assumption that θ_2 is a fixed point. $\square$

Lemma 8. *The optimal solution to the optimization problem in (7) is given by q^* , where each component satisfies*

$$q_i^* = g_i^{-1}(\theta^*).$$

The parameter θ^ is uniquely determined by the fixed-point equation:*

$$\Gamma(\theta^*) = \theta^*. \quad (40)$$

Proof. The feasibility condition for the optimization problem (7) is:

$$\sum_i g_i^{-1}(\theta) \leq \theta, \quad (41)$$

$$\Gamma(\theta) \leq \theta. \quad (42)$$

Since $\Gamma(\theta)$ is an increasing function, Theorem 7 guarantees the existence of a unique $\theta^* > 0$ satisfying $\Gamma(\theta^*) = \theta^*$.

Furthermore, $\Gamma(\theta)$ is convex, and at θ^* , we have $\Gamma'(\theta^*) \geq 1$. By the first-order convexity condition, for any $\theta > \theta^*$,

$$\Gamma(\theta) \geq \Gamma(\theta^*) + \Gamma'(\theta^*)(\theta - \theta^*) \geq \theta.$$

This implies that values $\theta > \theta^*$ are not feasible. Similarly, for $\theta < \theta^*$, by definition, we obtain

$$\sum_j q_j < \sum_j q_j^*.$$

Hence, the optimal solution to the problem in (7) is given by $q_i^* = g_i^{-1}(\theta^*)$. $\square$

Theorem 3. *Consider the fixed-point iteration:*

$$\theta_{k+1} = \Gamma^{-1}(\theta_k) \quad (43)$$

where the initial value $\theta_0 > 0$ is given. Then, the sequence $\{\theta_k\}$ generated by this iteration converges to a unique fixed point θ^ , i.e.,*

$$\lim_{k \rightarrow \infty} \theta_k = \theta^* \quad (44)$$

Furthermore, the fixed point θ^ satisfies the equation*

$$\Gamma(q_1^*) = q_1^* \quad (45)$$

Proof. We know that $\Gamma(\theta)$ is an increasing and convex function. Therefore, $\Gamma^{-1}(\theta)$ is an increasing and concave function. The properties are similar to the function $T(x)$, hence the proof is same as theorem 2. $\square$

These algorithms can be viewed as a fixed-point iteration connected to the nonlinear Perron-Frobenius theory, establishing links to well-established mathematical frameworks presented in [18]–[20]

Algorithm 2 Fixed-point iteration for finding q^* for n -flow case

- 1: **Input:** Initial guess $\theta_0 > 0$, tolerance $\epsilon > 0$, $g_i^{-1}(\theta), \Gamma^{-1}(\theta)$
 - 2: **Output:** Fixed point q^*
 - 3: **Initialize:** Set θ_0
 - 4: **Set** $k \leftarrow 1$
 - 5: **Set** $\theta_1 \leftarrow \Gamma^{-1}(\theta_0)$
 - 6: **while** $|x_k - x_{k-1}| > \epsilon$ **do**
 - 7: Compute $\theta_{k+1} \leftarrow \Gamma^{-1}(\theta_k)$
 - 8: Update $k \leftarrow k + 1$
 - 9: **end while**
 - 10: **Return** $q_i^* = g_i^{-1}(\theta_k)$
-

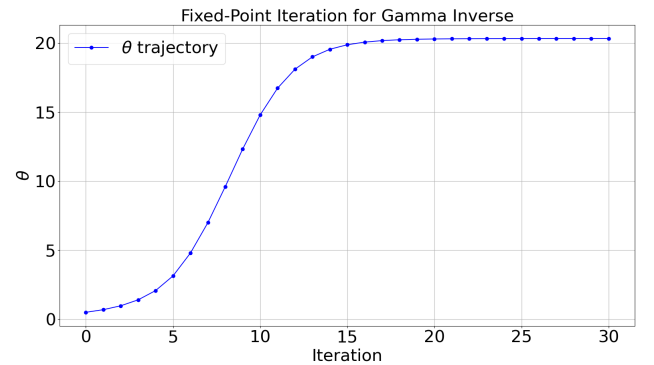


Fig. 4: Trajectory of q_1 under Algorithm 1, illustrating convergence to the optimal value $\theta^* = 20.312$. The parameters for this case are $b = [10, 15, 10]$, $r = [1, 2, 1]$, $d = [1, 1, 0.5]$, $c = 100$, and $L = 1$, with an initial guess of $\theta_0 = 0.5$.

In Figure 4, the trajectory of θ , as determined by the Algorithm 2, is shown to converge to the optimal value θ^* . The specified parameters for this case are $b = [10, 15, 10]$, $r = [1, 2, 1]$, $d = [1, 1, 0.5]$, $c = 100$, and $L = 1$. Starting with an initial guess of $\theta_0 = 0.5$, the algorithm successfully converges to the optimal solution, $\theta^* = 20.312$.

No. Flows	Burst (Kb)	Packet Length (Kb)	Rate (Kb/s)	Delay Target (ms)	Capacity (Kb/s)	Optimal Quanta (b)	Simulated Delay (ms)
2	10	3	1	1	40	14000	0.2
	10		1	1		14000	0.2
2	10	5	1	1	80	11764	0.5
	15		2	0.5		28571	0.1
3	10	5	1	1	150	1144	0.5
	15		2	0.5		4211	0.2
	15		2	0.5		4211	0.2
4	10	5	1	1	200	2641	0.6
	15		2	0.5		10593	0.2
	15		2	0.5		10593	0.2
	10		1	0.75		3872	0.4

TABLE II: Quanta values from Algorithms 1 and 2 compared to DRR simulation delays.

We simulate the DRR scheduler using the optimal quanta obtained from the convex optimization solution and observe the actual delays. Since the optimization provides real-valued quanta, we convert them to integer values by taking the floor function. This conversion does not guarantee that the delay requirement will be met exactly. However, our simulations show that the delay constraints are still satisfied, as shown in Table II. This is because the optimization approach was conservative, using an upper bound in the delay calculation, which leads to an over-designed solution. As a result, even after rounding down the quanta, the observed delays remain within acceptable limits.

VI. CONCLUSION

This paper presents an optimization framework designed to maximize the number of packets served in a round while optimizing total quanta and meeting delay requirements. We have formulated a convex optimization problem, derived the necessary conditions for the optimal solution, proposed two algorithms to find the optimal solution and proved the existence and uniqueness of a fixed point for the governing system of equations. Our theoretical results provide a solid understanding of the interaction between the two flows and how to maximize the total number of packets served in a round of DRR scheduler. In the practical simulation, we mapped real-valued quanta to integer-valued quanta by applying the floor function, as integer quanta are commonly required in network systems. The simulation demonstrated that the delay requirements were successfully met. However, it is important to note that the simulation does not guarantee the delay constraints will always be satisfied in all cases. The use of the floor function introduces an approximation, which may not always hold, resulting in some uncertainty. Ensuring that the delay requirements are consistently met despite this approximation, remains an open problem. In future work, we plan to explore methods for establishing more robust guarantees to formally

ensure delay constraints are satisfied. Additionally, we aim to refine our approach to manage better integer-valued constraints without compromising system performance and efficiency.

REFERENCES

- [1] M. Shreedhar and G. Varghese, "Efficient fair queueing using Deficit Round-Robin," *IEEE/ACM Trans. Netw.*, vol. 4, p. 375–385, June 1996.
- [2] L. Lenzini, E. Mingozzi, and G. Stea, "Aliquem: a novel DRR implementation to achieve better latency and fairness at $O(1)$ complexity," in *IEEE 2002 Tenth IEEE International Workshop on Quality of Service (Cat. No.02EX564)*, pp. 77–86, 2002.
- [3] M. Boyer, G. Stea, and W. M. Sofack, "Deficit Round Robin with network calculus," in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, pp. 138–147, 2012.
- [4] S. M. Tabatabaee and J.-Y. Le Boudec, "Deficit Round-Robin: A Second Network Calculus Analysis," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2216–2230, 2022.
- [5] A. Bouillard, "Individual Service Curves for Bandwidth-Sharing Policies Using Network Calculus," *IEEE Networking Letters*, vol. 3, no. 2, pp. 80–83, 2021.
- [6] Y. Hua and X. Liu, "Scheduling design and analysis for end-to-end heterogeneous flows in an avionics network," in *2011 Proceedings IEEE INFOCOM*, pp. 2417–2425, 2011.
- [7] A. Soni, X. Li, J.-L. Scharbarg, and C. Fraboul, "Optimizing Network Calculus for Switched Ethernet Network with Deficit Round Robin," in *2018 IEEE Real-Time Systems Symposium (RTSS)*, pp. 300–311, 2018.
- [8] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*, vol. 2050, 2006.
- [9] E. Coronado, R. Riggio, J. Villalón, and A. Garrido, "Lasagna: Programming Abstractions for End-to-End Slicing in Software-Defined WLANs," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–15, 2018.
- [10] M. H. Ali and M. H. Ali, "A Priority-Based Deficit Weighted Round Robin Queueing for Dynamic Bandwidth Allocation Algorithm in Gigabit Passive Optical Network," in *Intelligent Energy Management Technologies* (M. Shorif Uddin, A. Sharma, K. L. Agarwal, and M. Saraswat, eds.), (Singapore), pp. 65–71, Springer Singapore, 2021.
- [11] A. M. Alsahag, B. M. Ali, N. K. Noordin, and H. Mohamad, "Fair uplink bandwidth allocation and latency guarantee for mobile WiMAX using fuzzy adaptive deficit round robin," *Journal of Network and Computer Applications*, vol. 39, pp. 17–25, 2014.
- [12] A. Soni, J.-L. Scharbarg, and J. Ermont, "Quantum assignment for qos-aware afdx network with deficit round robin," in *Proceedings of the 27th International Conference on Real-Time Networks and Systems, RTNS '19*, (New York, NY, USA), p. 70–79, Association for Computing Machinery, 2019.
- [13] A. Soni, J.-L. Scharbarg, and J. Ermont, "Efficient configuration of a QoS-aware AFDX network with Deficit Round Robin," in *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, vol. 1, pp. 251–258, 2020.
- [14] A. Mukherjee, J. Kuri, and C. Singh, "Convexity and Optimization in Deficit Round Robin Scheduling for Delay-Constrained Systems," 2025.
- [15] S. R. Lay, *Convex sets and their applications*. Courier Corporation, 2007.
- [16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [17] R. Rockafellar, "Convex Analysis," *Princeton Mathematical Series*, vol. 28, 1970.
- [18] L. Zheng, Y.-W. P. Hong, C. W. Tan, C.-L. Hsieh, and C.-H. Lee, "Wireless Max-Min Utility Fairness With General Monotonic Constraints by Perron-Frobenius Theory," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7283–7298, 2016.
- [19] L. Zheng, D. W. H. Cai, and C. W. Tan, "Max-Min Fairness Rate Control in Wireless Networks: Optimality and Algorithms by Perron-Frobenius Theory," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 127–140, 2018.
- [20] Y. Gao, C. W. Tan, Y. Huang, Z. Zeng, and P. R. Kumar, "Characterization and Optimization of Delay Guarantees for Real-Time Multimedia Traffic Flows in IEEE 802.11 WLANs," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1090–1104, 2016.