



## Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:  
<http://oatao.univ-toulouse.fr/24789>

### Official URL

DOI : <https://doi.org/10.1109/RTSS.2018.00046>

**To cite this version:** Soni, Aakash and Li, Xiaoting and Scharbarg, Jean-Luc and Fraboul, Christian *Optimizing Network Calculus for Switched Ethernet Network with Deficit Round Robin*. (2019) In: 39th IEEE Real-Time System Symposium (RTSS 2018), 11 December 2018 - 14 December 2018 (Nashville, United States).

Any correspondence concerning this service should be sent to the repository administrator: [tech-oatao@listes-diff.inp-toulouse.fr](mailto:tech-oatao@listes-diff.inp-toulouse.fr)

# Optimizing Network Calculus for Switched Ethernet Network with Deficit Round Robin

Aakash SONI  
ECE Paris - INPT/IRIT, Toulouse  
France  
aakash.soni@ece.fr

Xiaoting Li  
ECE Paris  
France  
xiaoting.li@ece.fr

Jean-Luc Scharbarg  
IRIT-ENSEEIH, Toulouse  
France  
Jean-Luc.Scharbarg@enseeiht.fr

Christian Fraboul  
IRIT-ENSEEIH, Toulouse  
France  
Christian.Fraboul@enseeiht.fr

**Abstract**—Avionics Full Duplex switched Ethernet (AFDX) is the de facto standard for the transmission of critical avionics flows. It is a specific switched Ethernet solution based on First-in First-out (FIFO) scheduling. Worst-case traversal time (WCTT) analysis is mandatory for such flows, since timing constraints have to be guaranteed. A classical approach in this context is Network Calculus (NC). However, NC introduces some pessimism in the WCTT computation. Moreover, the worst-case often corresponds to very rare scenarios. Thus, the network architecture is most of the time lightly loaded. Typically, less than 10 % of the available bandwidth is used for the transmission of avionics flows on an AFDX network embedded in an aircraft. One solution to improve the utilization of the network is to introduce Quality of Service (QoS) mechanisms. Deficit Round Robin (DRR) is such a mechanism and it is envisioned for future avionics networks. A WCTT analysis has been proposed for DRR. It is based on NC. It doesn't make any assumption on the scheduling of flows by end systems. The first contribution of this paper is to identify sources of pessimism of this approach and to propose an improved solution which removes part of this pessimism. The second contribution is to show how the scheduling of flows can be integrated in this optimized DRR approach, thanks to offsets. An evaluation on a realistic case study shows that both contributions bring significantly tighter bounds on worst-case latencies.

**Index Terms**—Deficit Round Robin, Network Calculus, worst-case traversal time, switched Ethernet network, offsets

## I. INTRODUCTION

Up to now, Quality of Service (QoS) mechanisms are not used in practice in the context of avionics. The de facto standard is the AFDX network, which mainly implements a FIFO service discipline in switch output ports. Actually, two priority levels are available, but they are rarely used. Different approaches have been proposed for Worst-case traversal time analysis in the context of avionics, in particular Network Calculus (NC) [1], Trajectories [2] and Model Checking [3]. Due to the problem of combinatorial explosion, Model Checking doesn't scale. Trajectories and NC approaches compute a sure but often pessimistic upper bound on end-to-end delay. NC has a strong mathematical background with successful implementation to certify A380 AFDX backbone[4].

The pessimism of WCTT analysis as well as the fact that worst-case scenarios have a very low probability to occur lead to a very lightly loaded network. Typically, less than 10 % of the available bandwidth is used for the transmission of avionics flows on an AFDX network embedded in an aircraft

[3]. One solution to improve the utilization of the network is to introduce Quality of Service (QoS) mechanisms. Deficit Round Robin (DRR) and Weighted Round Robin (WRR) are such mechanisms and they are envisioned for future avionics networks. We have proposed a first evaluation of WRR in the context of avionics in [5]. In this paper we focus on DRR.

Deficit Round Robin (DRR) was proposed in [6] to achieve fair sharing of the capacity of a server among several flows. The main interest of DRR is its simplicity of implementation. As long as specific allocation constraints are met, it can exhibit  $O(1)$  complexity. A lot of work has been devoted to DRR [7], [8], [6], [9], [10]. They point out the undeniable high latency of DRR scheduler and propose some improvements. One of the most efficient implementations called "Aliquem" is proposed in [10]. It shows a remarkable gain in latency and fairness while still preserving  $O(1)$  complexity. A comparison of DRR scheduler with First-In-First-Out (FIFO) and Static Priority (SP) scheduler used in AFDX network is shown in [11]. The end-to-end delay (ETE) bounds are computed and the paper shows the comparatively better performance of DRR scheduler over FIFO and SP scheduler, given an optimized network configuration. Another DRR implementation is proposed in [9], which combines the DRR with SP scheduling, to improve schedulability and makes more efficient use of hardware resources. A detailed analysis and improvement of DRR latency bound for homogeneous flows is given in [8]. Some mathematical errors of [8] are pointed out and corrected in [12]. Analysis of a server with DRR scheduler using NC method is first discussed in [7] which also proposes improvement in DRR latency. [7] generalizes the analysis to network with heterogeneous flows.

The first contribution of the paper is to identify sources of pessimism of existing worst-case end-to-end delay calculation using NC for a network with DRR schedulers and to propose an improved solution. An evaluation on an industrial size configuration shows that the proposed approach outperforms existing ones.

The approach in [7] as well as the optimized one in this paper don't make any assumption on the scheduling of flows by source end systems. The second contribution of this paper is to show how this scheduling can be integrated in our optimized WCTT analysis for DRR. We have presented such an integration in the existing WCTT analysis in [13].

The paper is organized as follows. The considered network model is presented in section II. It is followed by a brief recall of the DRR scheduling policy, its latency and delay calculation using Network Calculus in section III. Section IV exhibits sources of pessimism in DRR WCTT analysis. The main contribution is given in section V, where we propose an optimized NC approach for DRR scheduler based networks. In Section VI further improvements to classical NC approach are given, including the integration of end system scheduling. An evaluation on an industrial configuration is given in section VII. Section VIII concludes the paper and gives directions for future works.

## II. NETWORK AND FLOW MODEL

In this paper, we consider a real-time switched Ethernet network. It is composed of a set of end systems, interconnected by switched Ethernet network via full-duplex links. Thus, there are no collisions on links. Each link offers a bandwidth of  $R$  Mbps in each direction.

Each end system manages a set of flows, and each switch forwards a set of flows through its output ports, based on a statically defined forwarding table. This forwarding process introduces a switching latency, denoted by  $sl$ . Each port  $h$  of a switch  $S_x$ , denoted by  $S_x^h$ , can be connected at most to one end system or another switch. Each output port, of a switch or of an end system, has a set of buffers managed by a scheduler supporting a scheduling policy, for example: First-In-First-Out (FIFO), Fixed Priority (FP) queuing or Round Robin (RR) etc. In this paper, the considered network uses Deficit Round Robin (DRR) scheduler at each output port.

Sporadic flows are transmitted on this network. Each sporadic flow  $v_i$  gives rise to a sequence of frames emitted by a source end system with respect to the minimum inter-arrival duration imposed by a traffic shaping technique. This minimum inter-arrival duration is called the period  $T_i$  of flow  $v_i$ . If the duration between any two successive emissions of a flow  $v_i$  is  $T_i$ , then, the flow  $v_i$  is periodic. The size of each frame of flow  $v_i$  is constrained by a maximum frame length ( $l_i^{max}$ ) and a minimum frame length ( $l_i^{min}$ ). Each flow  $v_i$  follows a predefined path  $\mathcal{P}_i$  from its source end system till its last visited output port, and then arrives at its destination end system.

Figure 1 shows an example of a switched Ethernet network configuration which consists of 4 switches,  $S_1$  to  $S_4$ , interconnecting 10 end systems,  $e_1$  to  $e_{10}$ , through full duplex links to transfer 20 flows,  $v_1$  to  $v_{20}$ . In this work, each output port of a switch has a set of buffers controlled by a Deficit Round Robin (DRR) scheduler. The links provide a bandwidth of  $R = 100$  Mbits/s. Table I summarizes flow features (inter-arrival duration  $T_i$  as well as minimum and maximum frame size  $l_i^{min}$  and  $l_i^{max}$ ).

## III. DEFICIT ROUND ROBIN

In this section, we briefly recall the DRR scheduling policy. A more detailed description can be found in [6] and [7]. We

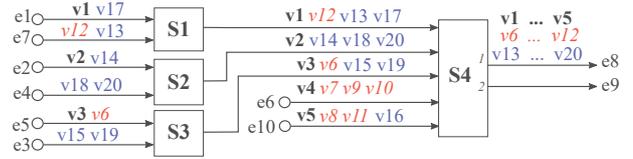


Fig. 1: Switched Ethernet network (Example 1)

TABLE I: Network Flow Configuration

Flows $v_i$	$T_i(\mu\text{sec})$	$l_i^{max}(\text{byte})$	$l_i^{min}(\text{byte})$
$v_{12}, v_{20}$	512	100	80
$v_1, v_7, v_8, v_9, v_{17}$	512	99	80
$v_2, v_4, v_5, v_{10}, v_{13}, v_{16}, v_{18}$	256	100	80
$v_3, v_{11}, v_{14}, v_{15}, v_{19}$	256	99	80
$v_6$	96	100	80

then summarize the DRR worst-case analysis in [7], [8]. This analysis is based on network calculus [1].

### A. DRR scheduler principle

DRR was designed in [6] for a fair sharing of server capacity among flows. DRR is mainly a variation of Weighted Round Robin (WRR) which allows flows with variable packet length to fairly share the link bandwidth.

The flow traffic in a DRR scheduler is divided into buffers based on few predefined classes. Each class receives service sequentially based on the presence of a pending frames in a class buffer and the credit assigned to the class. Each class buffer follows FIFO queuing to manage the flow packets. The DRR scheduler service is divided into rounds. In each round all the active classes are served. A class is said to be *active* when it has some flow packet in output buffer waiting to be transmitted. The basic idea of DRR is to assign a credit quantum  $Q_x^h$  to each flow class  $C_x$  at each switch output port  $h$ .  $Q_x^h$  is the number of bytes which is allocated to  $C_x$  for each round at port  $h$ . At any time, the current credit of a class  $C_x$  at a port  $h$  is called its deficit  $\Delta_x^h$ . Each time  $C_x$  is selected by the scheduler,  $Q_x^h$  is added to its deficit  $\Delta_x^h$ . As long as  $C_x$  queue is not empty and  $\Delta_x^h$  is larger than the size of  $C_x$  queue head-of-line packet, this packet is transmitted and  $\Delta_x^h$  is decreased by this packet size. Thus, the scheduler moves to next class when either  $C_x$  queue is empty or the deficit  $\Delta_x^h$  is too small for the transmission of  $C_x$  queue head-of-line packet. In the former case,  $\Delta_x^h$  is reset to zero. In the latter one,  $\Delta_x^h$  is kept for the next round.

The credit quantum  $Q_x^h$  is defined for each port  $h$ . It must allow the transmission of any frame from class  $C_x$  crossing  $h$ . Thus,  $Q_x^h$  has to be at least the maximum frame size of  $C_x$  flows at port  $h$ . Let  $\mathcal{F}_{C_x}^h$  be the set of flows of class  $C_x$  at output port  $h$ . Let  $l_{C_x}^{max,h}$  and  $l_{C_x}^{min,h}$  be the max and min frame size among all class  $C_x$  flows at output port  $h$ . We have:

$$l_{C_x}^{max,h} = \max_{i \in \mathcal{F}_{C_x}^h} l_i^{max}, \quad l_{C_x}^{min,h} = \min_{i \in \mathcal{F}_{C_x}^h} l_i^{min} \quad (1)$$

Algorithm 1 shows an implementation of DRR at a switch output port  $h$  with  $n$  traffic classes. First, deficits are set to 0



**Definition 1.** *Theoretical service rate:* The quantum  $Q_x^h$  allocated to traffic class  $C_x$  at port  $h$  defines the theoretical service rate  $\rho_x^h$  of  $C_x$  at  $h$ , i.e. the minimum service rate that  $C_x$  should get on the long term. We have

$$\rho_x^h = \frac{Q_x^h}{\sum_{1 \leq j \leq n^h} Q_j^h} \times R \quad (2)$$

In the example in Figure 2, output port  $S_4^1$  is shared by  $n^{S_4^1} = 3$  classes ( $C_1$ ,  $C_2$  and  $C_3$ ). All of them are assigned a quantum of 199 bytes (1592 bits). Thus, the theoretical service rate for any class  $C_x$  can be computed by Equation (2):

$$\rho_x^{S_4^1} = \frac{199}{199 * 3} \times 100 \text{ Mbits/s} = \frac{100}{3} \text{ Mbits/s}$$

However the service provided to  $C_x$  at  $h$  in a given time interval might be more or less than the theoretical one.

**Definition 2.** *Actual service rate:* The actual service rate is the service rate received by a given class  $C_x$  at a port  $h$  in a given time interval.

The actual service rate of a given class depends on the packets which effectively cross the output port. First, as previously defined, a class is active in an output port  $h$  when it has pending packets in  $h$ . In a given time interval, active classes share the available bandwidth. For instance, considering port  $S_4^1$  in Figure 2,  $C_2$  and  $C_3$  each get half of the bandwidth in any interval where they are active and  $C_1$  is not. Thus, a class can receive more than its theoretical service rate when some other classes are inactive. Second, since frames are transmitted sequentially, each class is served on its turn, thus getting 100 % of service for some duration. Third, since a packet cannot be transmitted in the current round if its size is more than the remaining credit of its class, a class might get less than its theoretical service rate in a round. Conversely, since the credit which is not used by a class in a round might be used in the following round, a class can get more than its theoretical service rate in a round.

The aim of a WCTT analysis is to maximize the latency of a given flow. It can be obtained by minimizing the actual service rate of its class. In [8], it is based on the DRR scheduler latency.

**Definition 3.** *DRR scheduler latency:* The DRR scheduler latency  $\Theta_x^h$  experienced by a class  $C_x$  flow at output port  $h$  is defined as the maximum delay before  $C_x$  flow is served at its theoretical service rate  $\rho_x^h$ .

[8] determines a lower bound on the service that  $C_x$  receives in a given interval. To that purpose, it introduces two delays at the beginning of the considered interval:

- the delay  $X_{C_x}^h$  before class  $C_x$  receives service for the first time in the interval,
- a delay  $Y_{C_x}^h$  to take into account the fact that, when  $C_x$  receives service for the first time, it can be a reduced service.

These delays are illustrated in Figure 3

- $t_i$  is the starting time of round  $i$ ,
- Round 1 starts at the arrival time of a packet of the class under study with no backlog for this class,
- $X_{C_x}^h$  is part of the first round, starting at time  $t_1$  and ending at time  $t'_1$ ,
- $Y_{C_x}^h$  is part of the second round, starting at time  $t'_2$  and ending at time  $t''_2$ .

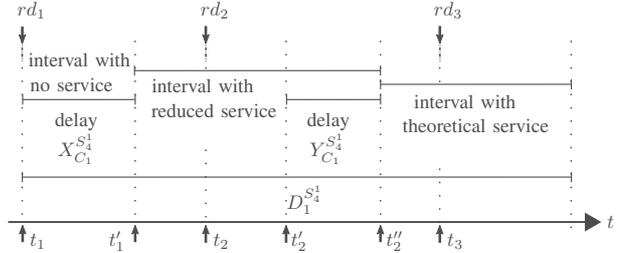


Fig. 3: DRR scheduler latency

Let us come back to the example in Figure 2 in order to illustrate the first duration (the delay  $X_{C_x}^h$  before first service). The first  $C_1$  packet arrives at time  $t_1$ , where it just misses its turn to receive service. Before receiving the first service, it has to wait till time  $t'_1$  while all the other active classes ( $C_2$ ,  $C_3$ ) are served. This delay has been analyzed and upper bounded in [8]. It is denoted by  $X_x^h$  for class  $C_x$  in node  $h$ . It has been shown in [8] that it is maximized when class  $C_x$  has to wait for all the other classes with maximum transmission capacity. This maximum delay can be computed by the following formula:

$$X_x^h = \frac{\sum_{j=\{1,2,\dots,n^h\}, j \neq x} (Q_j^h + \Delta_j^{max,h})}{R} \quad (3)$$

where  $\Delta_j^{max,h}$  is the maximum deficit of class  $C_j$  in node  $h$  at the end of its service. Since class  $C_j$  packets are served as long as the remaining deficit of class  $C_j$  is not smaller than the size of class  $C_j$  head-of-line packet, the remaining deficit has to be smaller than the largest  $C_j$  packet. Thus, we have:

$$\Delta_x^{max,h} = l_{C_x}^{max,h} - 1 \quad (4)$$

where  $l_{C_x}^{max,h}$  is the size of the largest  $C_x$  packet.

This maximum delay is observed for class  $C_1$  in round  $rd_1$  in Figure 2. Indeed, classes  $C_2$  and  $C_3$  have a maximum remaining deficit at time  $t_1$ :

$$\begin{aligned} \Delta_2^{max,h} &= l_{C_2}^{max,h} - 1 = 100 - 1 = 99 \text{ bytes} \\ \Delta_3^{max,h} &= l_{C_3}^{max,h} - 1 = 100 - 1 = 99 \text{ bytes} \end{aligned}$$

Both classes ( $C_2$ ,  $C_3$ ) get maximum service between  $t_1$  and  $t'_1$ . They both have a credit of  $199 + 99 = 298$  bytes. It corresponds to the cumulative size of pending packets for  $C_2$  ( $v_6, v_7$  and  $v_8$  packets) as well as  $C_3$  ( $v_{13}, v_{14}$  and  $v_{15}$  packets). Thus, the delay until first  $C_1$  pending packet (from  $v_5$ ) gets transmitted is computed by Equation (3):

$$X_1^{S_4^1} = \frac{(298 + 298) * 8}{100} = 47.68 \mu s$$

The second delay  $Y_{C_x}^h$  comes from reduced service. It can also be illustrated with the example in Figure 2. In  $rd_1$ ,  $C_1$  receives a reduced service (100 bytes corresponding to the transmission of a  $v_5$  packet). Indeed, the remaining deficit (99 bytes) is smaller than head-of-line  $C_1$  packet (100 bytes for  $v_4$  packet). Thus,  $C_1$  receives at least its theoretical service rate in  $rd_2$ , after the service of  $C_2$  and  $C_3$  (199 bytes for each class in Figure 2). It means that, between  $t'_1$  and  $t''_2$ ,  $C_1$  receives a service of 100 bytes. Since, between  $t'_1$  and  $t''_2$ , 498 bytes are transmitted (packets from  $v_5, v_9, v_6, v_{16}$  and  $v_{17}$ ),  $C_1$  receives an average service of

$$\frac{100}{498} \times 100 \simeq 20 \text{ Mbps}$$

instead of one third of the available bandwidth, i.e. 33.33 Mbps. Another solution to compute the average service for  $C_1$  between  $t'_1$  and  $t''_2$  is to split the interval in two parts:

- in the first part,  $C_1$  receives an average service of one third of the available bandwidth,
- in the second part, it receives no service.

Since  $C_1$  gets a service of 100 bytes at  $t'_1$ , it gets on average one third of the available bandwidth between  $t'_1$  and  $t''_2$ . Indeed, 300 bytes are transmitted between these two instants. Then,  $C_1$  gets no service between  $t'_2$  and  $t''_2$ . These intervals are illustrated in Figure 3

In [8], the computation of the largest possible duration of such an interval with no service is formalized. The authors in [8] prove an upper bound on this duration and show a scenario leading to this upper bound. We compute the duration corresponding to such a scenario and show that it corresponds to a worst-case. This worst-case duration  $Y_x^h$  for a class  $C_x$  in a node  $h$  is given by:

$$Y_x^h = \frac{Q_x^h - \Delta_x^{max,h} + \sum_{\substack{1 \leq j \leq n^h \\ j \neq x}} Q_j^h}{R} - \frac{Q_x^h - \Delta_x^{max,h}}{\rho_x^h} \quad (5)$$

The first fraction computes the duration between  $t'_1$  and  $t''_2$ , while the second one corresponds to the duration between  $t'_1$  and  $t'_2$ . The delay  $t''_2 - t'_2$  is the impact of the reduced service on class  $C_x$ . The first fraction corresponds to the situation where class  $C_x$  receives its minimum possible credit  $Q_x^h - \Delta_x^{max,h}$  (its deficit for the following round is maximized) while other classes receive exactly the credit corresponding to their quantum. The second fraction computes the duration of a round where class  $C_x$  receives its minimum possible credit and its theoretical service rate.  $Y_x^h$  can be greater if one class  $C_j$  ( $j \neq x$ ) receives more than its quantum in round  $rd_2$ :  $C_j$  receives a credit of  $Q_j^h + d$  with  $0 < d \leq \Delta_j^{max,h}$ . In that case,  $C_j$  has a deficit of at least  $d$  from round  $rd_1$ . However, in the computation of  $X_x^h$ , we consider that class  $C_j$  consumes its maximum possible credit  $Q_j^h + \Delta_j^{max,h}$  in  $rd_1$ , leading to no deficit. Thus, adding a credit of  $d$  to class  $C_j$  in  $rd_2$  comes to remove a credit of at least  $d$  from  $C_j$  in round  $rd_1$ . Therefore it does not increase the sum  $X_x^h + Y_x^h$

Considering the example in Figure 2, it gives:

$$\begin{aligned} Y_1^{S_4^1} &= \frac{(199 - 99) * 8 + (199 + 199) * 8}{100} - \frac{(199 - 99) * 8}{\frac{100}{3}} \\ &= 15.84 \mu s \end{aligned}$$

This scenario in Figure 2 corresponds to a worst-case for class  $C_1$ , with maximum values for  $X_1^{S_4^1}$  and  $Y_1^{S_4^1}$ .

Finally, the DRR scheduler latency  $\Theta_x^h$  is defined as the delay before  $C_x$  packets are served at their theoretical service rate at port  $h$ . Thus:

$$\Theta_x^h = X_x^h + Y_x^h \quad (6)$$

In the example in Figure 2, we have:

$$\Theta_1^{S_4^1} = X_{C_1}^{S_4^1} + Y_{C_1}^{S_4^1} = 63.52 \mu s$$

2) *Network Calculus applied to DRR scheduling*: WCTT analysis for DRR has been modeled with Network Calculus in [7]. In this paragraph, this modeling is summarized. The Network Calculus (NC) theory is based on the (min, +) algebra. It has been proposed for worst-case backlog and delay analysis in networks [1]. It models traffic by arrival curves and network elements by service curves. Upper bounds on buffer size and delays are derived from these curves.

a) *Arrival Curve*: The traffic of a flow  $v_i$  at an output port  $h$  is over-estimated by an arrival curve, denoted by  $\alpha_i^h(t)$ . The leaky bucket is a classical arrival curve for a sporadic traffic:

$$\alpha_i^h(t) = r \times t + b, \text{ for } t > 0 \text{ and } 0 \text{ otherwise.}$$

It can be used to model a flow  $v_i$  at its source end system  $e_k$ . We have:

$$\alpha_i^{e_j}(t) = \frac{l_i^{max}}{T_i} \times t + l_i^{max}, \text{ for } t > 0 \text{ and } 0 \text{ otherwise.}$$

It means that  $v_i$  is allowed to send at most one frame of maximum length  $l_i^{max}$  bits every minimum inter-frame arrival time  $T_i \mu s$ .

Any flow  $v_i$  can be modeled in a similar manner at any switch output port  $h$  it crosses. However, since a frame of flow  $v_i$  can be delayed by other frames before it arrives at port  $h$ , a jitter  $J_i^h$  has to be introduced. It is the difference between the worst-case delay and the best-case delay for a frame of flow  $v_i$  from its source end system to port  $h$  [2].

Since flows of class  $C_x$  are buffered in their class queue and scheduled by FIFO policy, an overall arrival curve is used to constrain the arrival traffic of class  $C_x$  at port  $h$ . It is denoted by  $\alpha_{C_x}^h$  and calculated by:

$$\alpha_{C_x}^h(t) = \sum_{i \in \mathcal{F}_{C_x}^h} \alpha_i^h(t) \quad (7)$$

where  $\mathcal{F}_{C_x}^h$  is the set of  $C_x$  flows crossing port  $h$ .

As an example, let us consider the output port  $S_4^1$  in Figure 2. 5 flows of class  $C_1$  are scheduled by FIFO, leading

to  $\mathcal{F}_{C_1}^{S_4^1} = v_1, v_2, v_3, v_4, v_5$ . The overall arrival curve of class  $C_1$  can be computed by:

$$\alpha_{C_1}^{S_4^1}(t) = \sum_{i \in \mathcal{F}_{C_1}^{S_4^1}} \alpha_i^{S_4^1}(t)$$

which is illustrated by blue line in Figure 4a.

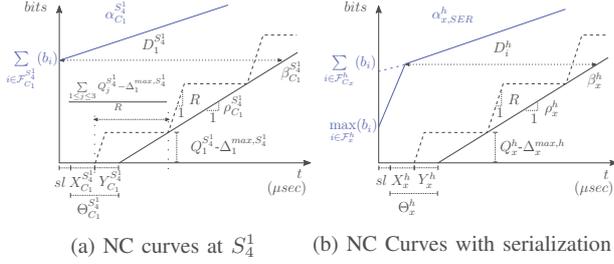


Fig. 4: NC curves at  $S_4^1$

*b) Service Curve:* According to NC, the full service provided at a switch output port  $h$  with a transmission rate of  $R$  (bits/s) is defined by:

$$\beta^h(t) = R[t - sl]^+$$

where  $sl$  is the switching latency of the switch, and  $[a]^+$  means  $\max\{a, 0\}$ .

According to [8] and [7], the full service is shared by all DRR classes at an output port  $h$  and each class  $C_x$  has a predefined service rate  $\rho_x^h$  based on its assigned credit quantum  $Q_x$  as explained in Section III-B Equation (2). Besides a reduced service rate, each class  $C_x$  could experience a DRR scheduler latency  $\Theta_x^h$  before receiving service with the predefined rate  $\rho_x^h$ . The scheduler latency can be calculated by Equation (6). Therefore, based on the NC approach, the residual service  $\beta_{C_x}^{DRR}$  to each class  $C_x$  is given by:

$$\beta_{C_x}^h(t) = \rho_x^h[t - \Theta_x^h - sl]^+ \quad (8)$$

$Y_x^h$  delay is considered right after  $X_x^h$ , in order to get a convex service curve.

In the example of the output port  $S_4^1$ , class  $C_1$  service curve is:

$$\beta_{C_1}^{S_4^1}(t) = \rho_1^{S_4^1} * [t - \Theta_1^{S_4^1} - sl]^+ = \frac{100}{3}(t - 63.52 - sl)^+$$

which is illustrated in Figure 4a.

The actual service curve is a staircase one (shown by the dashed black line in Figure 4a), as a flow alternates between being served and waiting for its DRR opportunity, as explained in [8]. For computation reason, NC approach employs the convex curve represented by equation (8) which is an underestimated approximation of actual staircase curve.

*c) Delay bound:* According to NC, the delay experienced by a  $C_x$  flow  $v_i$  constrained by the arrival curve  $\alpha_{C_x}^h(t)$  in a switch output port  $h$  offering a strict DRR service curve  $\beta_{C_x}^h(t)$  is bounded by the maximum horizontal difference between the curves  $\alpha_{C_x}^h(t)$  and  $\beta_{C_x}^h(t)$ . Let  $D_i^h$  be this delay. It is computed by:

$$D_i^h = \sup_{s \geq 0} (\inf_{\tau \geq 0} \{ \alpha_{C_x}^h(s) \leq \beta_{C_x}^h(s + \tau) \}) \quad (9)$$

Therefore, the end-to-end delay upper bound of a  $C_x$  flow  $v_i$  is denoted by  $D_i^{ETE}$  and it is calculated by:

$$D_i^{ETE} = \sum_{h \in \mathcal{P}_i} D_i^h \quad (10)$$

Based on the equation (9) and (10), the delay bound calculated for flow  $v_1$  of class  $C_1$  is found to be  $D_1^{S_4^1} = 234.91 \mu s$  and  $D_1^{ETE} = 387.63 \mu s$ .

#### IV. PESSIMISM OF DRR WCTT ANALYSIS

The delay upper bound  $D_i^h$  for flow  $v_i$  from class  $C_x$  presented in the previous section assumes that, at each output port  $h$ , every interfering class  $C_y$  consumes maximum service. More precisely, it assumes that, in any DRR round  $rd_k$ , each class  $C_y$  ( $y \neq x$ ) is always active and transmits frames of at least the size of its quantum value  $Q_y^h$ . Such an assumption might be pessimistic. Indeed, the traffic from one or several  $C_y$  classes might be too low to consume quantum values  $Q_y^h$  in each round. The effect of such a pessimism on service curve is shown in Figure 5.

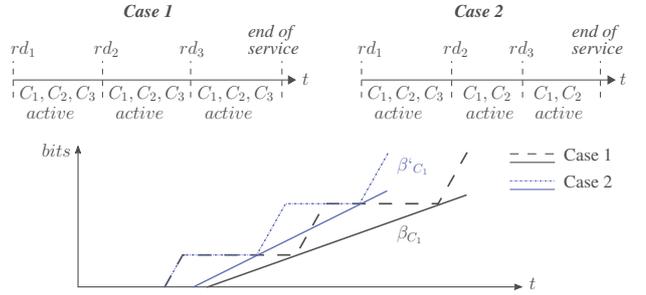


Fig. 5: Pessimism in DRR Service

This pessimism can be illustrated with the example in Figure 6. This example is based on the network architecture in Figure 1. The difference is that part of  $C_2$  and  $C_3$  flows that are transmitted from  $S_4$  to  $e_8$  in Figure 1 are transmitted to  $e_9$  in Figure 6.

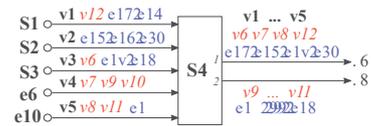


Fig. 6: Switched Ethernet network (Example 2)

We focus on output port  $S_4^1$  to calculate the delay experienced by flow  $v_1$  from class  $C_1$ . In the given example, it is

worth noting that the considered class  $C_1$  has more flows to be served as compared to the class  $C_2$  and  $C_3$ . The corresponding DRR schedule rounds are shown in Figure 7. The scenario in

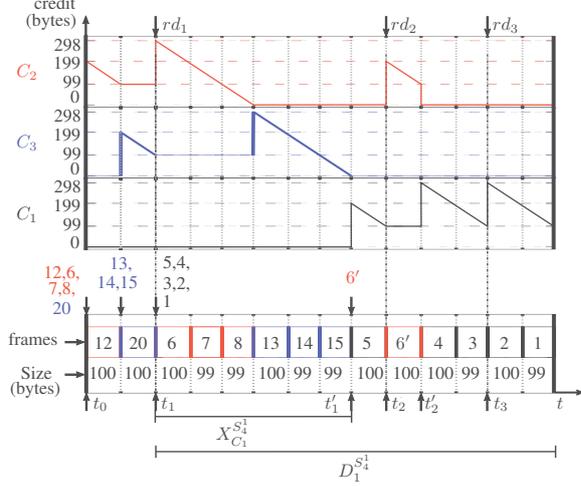


Fig. 7: DRR cycles at  $S_4^1$

the example given in Figure 7 is similar to the scenario in Figure 2, where the first  $C_1$  packet arrive at time  $t_1$  and it experience a delay  $X_{C_1}^{S_4^1} = 47.68 \mu s$  before being served for the first time. As shown in Figure 7, at  $t'_1$ , class  $C_3$  has served all its frames in the buffer, by transmitting frames from  $v_{13}$ ,  $v_{14}$  and  $v_{15}$ . Its deficit is reduced to  $\Delta_3^{S_4^1} = 0$  and it has no more frames to further delay class  $C_1$  flows. Similarly, class  $C_2$  consumes all its deficit by transmitting frames from  $v_6$ ,  $v_7$  and  $v_8$  and its deficit is also reduced to  $\Delta_2^{S_4^1} = 0$ . Another frame from flow  $v_6$  has arrived at  $t'_1$ . At  $t_2$ , class  $C_2$  receive service of 100 bytes (corresponding to frame of  $v_6$ ). At  $t'_2$ , there are no more frames from class  $C_2$  and  $C_3$  to be served, class  $C_1$  gets the deficit of  $\Delta_1^{S_4^1} = 199 + 99 = 298 \text{ bytes}$  to serve frames from  $v_4$  and  $v_3$ . Since the remaining deficit is less than the size of next frame from  $v_2$ , and there are no other active classes, class  $C_1$  receives an additional credit equal to its quantum value. Thus, it has a deficit of  $\Delta_1^{S_4^1} = 298 \text{ bytes}$  to serve  $v_2$  and  $v_1$  flows. Between  $t'_1$  and  $t'_2$ ,  $C_1$  receives a service of 100 bytes. Since, between  $t'_1$  and  $t'_2$ , 200 bytes are transmitted, it means  $C_1$  received an average service of

$$\frac{100}{200} \times 100 = 50 \text{ Mbps}$$

which is greater than its theoretical service rate (from equation (2))  $\rho_1^{S_4^1} = \frac{199}{199+199+199} \times 100 = 33.33 \text{ Mbps}$ . Thus, in this scenario class  $C_1$  flows do not experience any reduced service. Therefore, the total latency observed by class  $C_1$  flows before they could be served at their theoretical service rate is  $47.68 \mu s$ , which is much less than that considered by the NC approach i.e.  $47.68 + 15.84 = 63.52 \mu s$ . In this case, the delay bounds, calculated from Equation (9) and (10), for flow  $v_1$  of class  $C_1$  are  $D_1^{S_4^1} = 215.9 \mu s$  and  $D_1^{ETE} = 366.87 \mu s$ .

However, the scenario in Figure 7 might not lead to the worst-case. In next section, we show how to upper bound the effective impact of interfering classes.

## V. TAKING INTO ACCOUNT THE EFFECTIVE LOAD OF INTERFERING CLASSES

As illustrated in the previous section, maximizing the service of interfering classes when computing the worst-case end-to-end delay for packets of a given class  $C_x$  can introduce pessimism, since these interfering classes might generate too few traffic to consume all their allocated service. In this section we show how it is possible to upper bound the traffic of these interfering classes that impact the worst-case delay of  $C_x$  packets. Each time this upper bound is smaller than the maximized service for interfering classes, the difference can be safely removed from the worst-case delay of  $C_x$  packets. The approach considers the following steps.

- 1) We compute the worst-case delay  $D_x^h$  of class  $C_x$  flows in node  $h$ , using the NC approach from [8], [7] presented in previous sections (equation (9) and (10)). Indeed, all the flows of a given class experience same delay at a given output port.
- 2) We determine the service load  $SL_y^h(t)$  available for class  $C_y$  at a node  $h$  between 0 and  $D_x^h$ . It corresponds to the maximized service of interfering classes  $C_y$  used by the NC approach from [8], [7].
- 3) We calculate the effective maximum load  $L_y^{max,h}(t)$  of a class  $C_y$  at a node  $h$  between 0 and  $D_x^h$ . It gives an upper bound on the  $C_y$  traffic which has to be served at node  $h$  between 0 and  $D_x^h$ .
- 4) If  $L_y^{max,h}(D_x^h) < SL_y^h(D_x^h)$ , the difference can be safely removed from the worst-case delay  $D_x^h$  of class  $C_x$  flows in node  $h$ .

The first step is a direct application of what has been presented in previous sections. The three other steps are detailed in the following paragraphs.

### A. Maximized service of interfering classes

The NC approach in [8], [7] considers a maximized service for an interfering class  $C_y$  that is not the same for all the time intervals. The time starts when the first  $C_x$  frame arrives at node  $h$ .

- The first interval  $(0, X_x^h]$  corresponds to the delay  $X_x^h$  before  $C_x$  is served for the first time. In this interval, the NC approach assumes that  $C_y$  gets a service of at most  $Q_y^h + \Delta_y^{max,h}$  bytes.
- The second interval starts at  $X_x^h$  and stops when  $C_x$  gets service for the second time. In this interval,  $C_x$  gets a service of  $Q_x^h - \Delta_x^{max,h}$  bytes while each other class  $C_y$  gets a service of  $Q_y^h$  bytes.
- The following intervals are all identical. As in the second one, each  $C_y$  class gets a service of  $Q_y^h$  bytes. The difference with the second interval is that  $C_x$  class also gets a service of  $Q_x^h$  bytes. Thus, each of these intervals is a bit longer than the second one.

Thus, the maximized service load  $SL_y^h(t)$  is defined as follows:

$$SL_y^h(t) = \begin{cases} 0 & t < X_x^h \\ Q_y^h + \Delta_y^{max,h} & X_x^h \leq t < t_N \\ Q_y^h + \Delta_y^{max,h} + \left(1 + \left\lfloor \frac{R \times (t - t_N)}{\sum_{j=\{1,2,\dots,n_h\}} Q_j^h} \right\rfloor\right) Q_y^h & t_N \leq t \end{cases} \quad (11)$$

where  $t_N$  is the end of the second interval:

$$t_N = X_x^h + \frac{1}{R} \left( \sum_{\substack{j=\{1,2,\dots,n_h\} \\ j \neq x}} Q_j^h + Q_x^h - \Delta_x^{max,h} \right)$$

It should be noticed that the load corresponding to a given interval is taken into account at the end of the interval. For instance,  $Q_y^h + \Delta_y^{max,h}$  is added to  $SL_y^h(t)$  at the end of the first interval, i.e.  $t = X_x^h$ . Thus,  $SL_y^h(t)$  is an under-bound of the maximized service load considered in [8]. We obtain a step function, as illustrated in Figure 8.

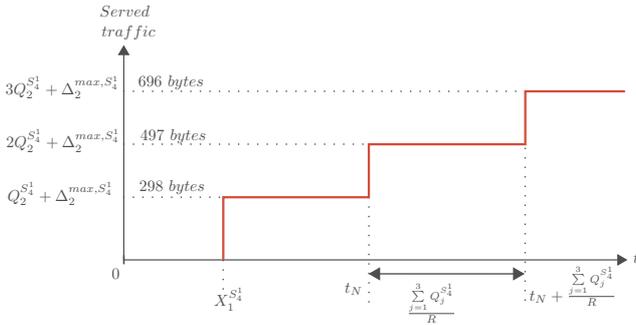


Fig. 8: Served traffic of class  $C_2$

Considering the example in Figure 6, for any value  $t$ , we have:

$$SL_{C_2}^{S_4}(t) = \begin{cases} 0 & t < 47.68 \\ 298 \text{ bytes} & 47.68 \leq t < 87.52 \\ 298 + \left(1 + \left\lfloor \frac{100 \times (t - 87.52)}{597} \right\rfloor\right) 199 \text{ bytes} & 87.52 \leq t \end{cases}$$

By applying the NC computation, we obtain  $D_1^{S_4} = 223.32 \mu s$ . Thus, the maximized load of an interfering class  $C_y$  in the service duration  $t = D_1^{S_4}$  of class  $C_1$  flow  $v_1$  is:

$$(SL_{C_2}^{S_4}(D_1^{S_4})) = (SL_{C_3}^{S_4}(D_1^{S_4})) = 895 \text{ bytes}$$

### B. Effective maximum load of interfering classes

The effective maximum load  $L_y^{max,h}(t)$  of an interfering class  $C_y$  at a node  $h$  in a given duration  $t$  is based on the arrival curves of NC. Thus,  $C_y$  load at the beginning of the duration is the sum of all the bursts of  $C_y$  flow arrival curves and it increases, following the long-term rate of each  $C_y$  flow arrival curve. Since the delay of class  $C_x$  flows in node  $h$  is

upper bounded by  $D_x^h$ , only packets arriving within a duration  $D_x^h$  have a chance to delay a given  $C_x$  packet. Thus,  $C_y$  load that can delay a given  $C_x$  packet is upper bounded by:

$$L_y^{max,h}(D_x^h) = \alpha_{C_y}^h(D_x^h) \quad (12)$$

where  $\alpha_{C_y}^h(t)$  is the overall arrival curve of the class  $C_y$  flows, calculated by Equation (7).

Considering the example in Figure 6, we have  $D_1^{S_4} = 223.32 \mu s$ , as depicted in the upper part in Figure 9a. The lower part in Figure 9a shows the overall arrival curve of  $C_2$  flows in  $S_4$ . Thus, we have:

$$L_{C_2}^{max,S_4}(D_1^{S_4}) = 858 \text{ bytes}$$

Similarly, for  $C_3$  flows, we have:

$$L_{C_3}^{max,S_4}(D_1^{S_4}) = 784 \text{ bytes}$$

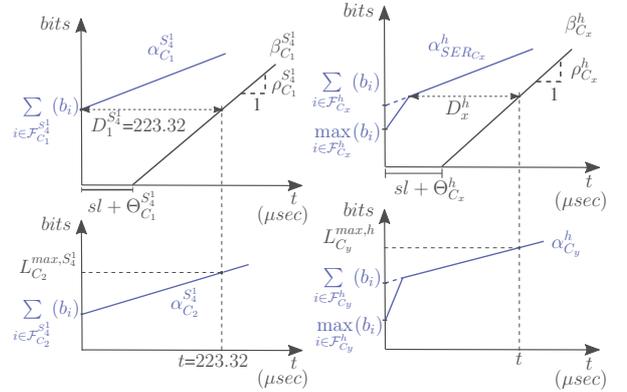


Fig. 9: Arrival traffic (a) Arrival traffic of  $C_2$  (b) Arrival traffic of  $C_y$  with grouping

Fig. 9: Arrival traffic

### C. Limitation of the service to the load

In previous paragraphs, we have computed for each class  $C_y$  interfering at node  $h$  with class  $C_x$  under study:

- the service load  $SL_y^h(t)$  taken into account by the NC approach in [8], [7],
- an upper bound  $L_y^{max,h}(t)$  on the effective load.

When this upper bound is smaller than the service load taken into account by the NC approach, the difference can be safely removed from the delay of  $C_x$  flow  $v_i$  packets in node  $h$ . Indeed, in a round, when a class  $C_y$  has nothing more to transmit, DRR moves to the next class, which is then served earlier. Therefore, following  $C_x$  packets will be served earlier, leading to a reduced delay.

Thus, for each interfering class  $C_y$  we can remove the following value:

$$\max(SL_y^h(D_x^h) - L_y^{max,h}(D_x^h), 0) \quad (13)$$

Since  $n_h - 1$  classes are interfering with  $C_x$  in node  $h$ , the optimized delay  $D_{i,opt}^h$  for  $C_x$  flow  $v_i$  in node  $h$  is given by:

$$D_{i,opt}^h = D_i^h - \frac{\sum_{y \in 1 \dots n_h, y \neq x} \max(SL_y^h(D_x^h) - L_y^{max,h}(D_x^h), 0)}{R} \quad (14)$$

Therefore, the end-to-end delay upper bound of a class  $C_x$  flow  $v_i$  can be computed by:

$$D_{i,opt}^{ETE} = \sum_{h \in \mathcal{P}_i} D_{i,opt}^h \quad (15)$$

In the example in Figure 6,  $D_{1,opt}^{S_4}$  is  $211.59\mu s$  and  $D_{1,opt}^{ETE}$  is  $300.86\mu s$ , which gives 19.8% improvement in ETE delay computation as compared to the  $D_1^{ETE} = 375.34\mu s$ .

## VI. FURTHER IMPROVEMENTS

In this section, we present three further improvements of the worst-case delay computation. The first one concerns the additional delay  $Y_x^h$  generated by the reduced service for class  $C_x$  when it is first served. The second one is based on the integration of the serialization effect: packets sharing an input link cannot arrive in the output node of the link at the same time. The third one concerns flow scheduling at end systems: integration of offset. The effect of integrating these improvements in NC approach is evaluated on an industrial configuration and shown in Table IV.

### A. Optimization of $Y_x^h$ latency

The duration  $Y_x^h$  of class  $C_x$  at an output port  $h$  is computed by considering that, when  $C_x$  is served for the first time, it consumes exactly its minimum possible service, i.e.  $Q_x^h - \Delta_x^{max,h}$ , where  $\Delta_x^{max,h}$  is the size of the largest  $C_x$  packet crossing port  $h$  minus one byte. If we assume a port  $h$  where all  $C_x$  packets have a size of 100 bytes and the quantum  $Q_x^h$  is 150 bytes, we have:

$$\Delta_x^{max,h} = 100 - 1 = 99 \text{ bytes}$$

Thus,

$$Q_x^h - \Delta_x^{max,h} = 150 - 99 = 51 \text{ bytes}$$

In that case, the computation of  $Y_x^h$  considers that  $C_x$  transmits 51 bytes when it is served for the first time.

However, since DRR imposes that  $Q_x^h$  is at least the size of the largest  $C_x$  frame (here  $150 \geq 100$ ), one  $C_x$  packet is guaranteed to be transmitted when  $C_x$  is served for the first time. In our case, 100 bytes will be transmitted during the first  $C_x$  service. Thus, considering only 51 bytes significantly underestimate this first service and it leads to an overestimation of  $Y_x^h$ .

In the general case, at least one  $C_x$  packet with minimum size is transmitted. Thus, the minimum first service for  $C_x$  cannot be less than this minimum size  $l_{C_x}^{min,h}$ . DRR guarantees that it cannot be less than  $Q_x^h - \Delta_x^{max,h}$ . Therefore it cannot be less than

$$\max\{Q_x^h - \Delta_x^{max,h}, l_{C_x}^{min,h}\}$$

This upgraded minimum first service for  $C_x$  can be integrated in the computation of  $Y_x^h$ . We get:

$$Y_x^h = \frac{\max(Q_x^h - \Delta_x^{max,h}, l_{C_x}^{min,h}) + \sum_{\substack{1 \leq j \leq n_h \\ j \neq x}} Q_j^h}{R} - \frac{\max(Q_x^h - \Delta_x^{max,h}, l_{C_x}^{min,h})}{\rho_x^h} \quad (16)$$

### B. Integration of DRR/FIFO Serialization

As explained in Section III-B2a, the arrival curve for a class  $C_x$  in a port  $h$  is obtained by summing the arrival curves of all  $C_x$  flows in  $h$ . This operation assumes that one frame from each flow arrives exactly at the same time in  $h$ . This situation might be impossible. Let us come back to the example in Figure 1.  $C_3$  flows  $v_{14}$ ,  $v_{18}$  and  $v_{20}$  share the link between  $S_2$  and  $S_4$ . Therefore they cannot arrive in  $S_4$  at the same time. They are serialized on the link.

This serialization effect has been integrated in the NC approach for FIFO [2]. The idea is to consider that the largest packet among the flows sharing the link arrives first. Packets from the other flows arrives by decreasing size, at the speed of the link.

This approach can be adapted to DRR scheduling. Indeed, DRR scheduling considers each class separately. Therefore, the serialization effect can be integrated as in [2], on a class by class basis. The principle of a curve integrating serialization is illustrated in Figure 4b.

These serialized arrival curves are then directly used for the worst-case delay computation.

While considering serialization with our optimized NC approach, one should pay attention while calculating effective maximum load of interfering classes (section V-B). In this case one should use the arrival curve and time value  $t (\geq D_x^h)$  as shown in Figure 9b.

### C. Flow scheduling at end system

In a switched Ethernet network each end system schedules flow transmission individually. The flow scheduling introduces temporal separation between flows and hence reduce the effective traffic in the network. The scheduling of flows emitted by given end system is characterized by the assignment of offsets which constrain the arrival of flows at output ports. The offset integration in NC was first proposed in [14] for First-In-First-Out (FIFO) scheduler. A Similar approach can be used for DRR schedulers.

The idea is that, if the flows are temporally separated at source end system and they share the same input link then they cannot arrive at an output port at the same time. Such flows can be aggregated as a single flow.

[14] defines *relative offset*  $O_{r,b,i}^h$  at an output port  $h$  as the minimum time interval between arrival time of a frame from a reference flow  $v_b$  and arrival time of a frame from another flow  $v_i$  after  $v_b$ . Such offset computation algorithm is given in [14], however, the aggregation technique can work with any offset assignment algorithm.

In NC, the integration of offset affects the computation of arrival curves. In DRR scheduler, flows of each class  $C_x$ , from

same source end system, can be aggregated as a single flow. This is valid because the flows of a class  $C_x$  transmitted from same source end system are affected by temporal separation and cannot delay each other. Class  $C_x$  flows can be aggregated by taking into account the relative offset at the given node. At an output port  $h$ , for  $n$  flows from class  $C_x$  the overall arrival curve  $\alpha_{C_x}^h$ , can be computed as :

- Make  $i$  subsets of class  $C_x$  flows, based on the flows sharing same source end system. Each subset  $SS_j$  has  $n_j$  flows such that  $n_1 + n_2 + \dots + n_i = n$ .
- Aggregate the flows of each subset  $SS_i$  as one flow and characterize its arrival curve  $\alpha_{SS_i}^h$ .

$$\alpha_{SS_i}^h = \max\{\alpha_{v_1\{v_2, v_3, \dots, v_i\}}^h, \dots, \alpha_{v_i\{v_1, v_2, \dots, v_{i-1}\}}^h\}$$

where,  $\alpha_{m\{n\}}^h$  is the arrival curve obtained when flow  $v_m$  arrives before flow  $v_n$  at output port  $h$ , with temporal separation of  $O_{r,m,n}^h$ .

- The overall arrival curve is the sum of the arrival curve of each subset, i.e.  $\alpha_{C_x}^h = \sum_{j=1}^i \alpha_{SS_j}^h$ .

Let us come back to the example in Figure 1, where  $C_3$  flows  $v_{13} \dots v_{20}$  compete at output port  $S_4^1$ ,  $n = 8$ , from 6 end systems  $e_1, e_7, e_2, e_4, e_3$  and  $e_{10}$ . Thus, there are 6 subsets:  $SS_1 = [v_{17}]$  ( $n_1 = 1$ ),  $SS_2 = [v_{13}]$  ( $n_2 = 1$ ),  $SS_3 = [v_{14}]$  ( $n_3 = 1$ ),  $SS_4 = [v_{18}, v_{20}]$  ( $n_4 = 2$ ),  $SS_5 = [v_{15}, v_{19}]$  ( $n_5 = 2$ ) and  $SS_6 = [v_{16}]$  ( $n_6 = 1$ ). The flows  $v_{18}$  and  $v_{20}$  will be aggregated as one flow to make aggregated arrival curve as they share source end system  $e_4$ . Similarly,  $v_{15}$  and  $v_{19}$  will also be aggregated as one flow. The arrival curve for subsets with only single flow is same as the arrival curve of the flow. For details about the aggregated curve computation, readers can refer to [14].

The obtained overall arrival curve can be used to compute delay using equation (9). It can also be used to calculate effective maximum load  $L_{C_y}^h(t)$  of interfering class  $C_y$  to optimize the delay using equation (14).

## VII. EVALUATION

In this section, we compare the approach in [8] (classical NC DRR) with our optimized approach (optimized NC DRR).

### A. Illustrative Example

First we consider the sample examples in Figures 1 and 6. Figure 10 shows end-to-end delay bounds obtained by both approaches for each VL of the configuration in Figures 1 and 6. For example in Figure 1, we have average gain 13.45%, and maximum gain 19.75%. For Figure 6, average gain is 16.84% and maximum 29.67%.

### B. Realistic case study

Next, we consider an industrial-size configuration, inspired from [2]. It includes 96 end systems, 8 switches, 984 flows, and 6412 paths (due to VL multi-cast characteristics).

We take into account three types of flows, namely critical flows, multimedia flows and best-effort data flows. Each flow

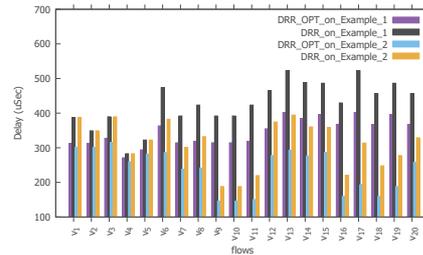


Fig. 10: End-to-end delay bound in network example 1 & 2

TABLE III: DRR Scheduler Configuration for Industrial Network

Class	Number of Flows	Max Frame Length (byte)	$Q_x$ (byte)	T (msec)	Category
$C_1$	128	150	3070	4 - 128	Critical
$C_2$	590	500	1535	2 - 128	Multimedia
$C_3$	266	1535	1535	2 - 128	Best-effort

type is assigned to one DRR class. Table III shows the DRR scheduler configuration.

The results of the optimized NC DRR approach are compared with the delays obtained by the classical NC DRR approach of [7] & [8]. For each path  $\mathcal{P}_x$  of each flow  $v_i$ , the upper bound  $D_{i,NC}^{\mathcal{P}_x}$  computed by the classical NC approach is taken as the reference value and it is normalized to 100. Then the upper bound  $D_{i,opt}^{\mathcal{P}_x}$  of optimized NC approach is normalized as

$$D_{i,opt,norm}^{\mathcal{P}_x} = \frac{D_{i,opt}^{\mathcal{P}_x}}{D_{i,NC}^{\mathcal{P}_x}} \times 100$$

For illustration purpose, the paths are sorted in increasing order of  $D_{i,opt,norm}^{\mathcal{P}_x}$ .

### C. Comparison of classical NC with DRR and Optimized NC with DRR

Table V shows the delay computation improvement on applying proposed optimization technique for NC with DRR. As shown in Table V and Figure 11a, 11b, 11c & 11d, the average improvement of the E2E delay bound computed in the given industrial configuration is around 47%. This is a significant improvement which shows that the proposed optimizations are relevant on an industrial configuration. It means that, on such configurations, the load in switch output ports is not equally shared between classes.

### D. Comparison of Optimized NC with DRR and Classical NC with FIFO

In Figure 11e and 11f we have compared the delays (computed by optimized NC approach) of the different flow classes when using DRR with the delays computed when using FIFO. It can be observed that the critical flows have smaller delays (shown in Figure 11e) than the two other flow classes (shown in Figure 11f) This is due to the fact that, as shown in Table III, DRR approach allows critical flows to have more allocated quanta in each round and hence produces smaller delay.

TABLE IV: Comparison result of improvements applied to classical NC approach

	$NC_{SER}$	$NC_{OFF}$	$NC_{SER,OFF}$	
NC	5.04	26.98	28.83	Avg Gain %
	19.57	70.05	70.05	Max Gain %

TABLE V: Comparison result of optimization applied to different NC approaches

$NC_x$	$NC_x^{OPT}$	
	avg gain %	max gain %
Classic	47.77	77.55
Offset	48.66	75.38
Grouping	47.2	75.11
Offset + Grouping	48.19	76.59

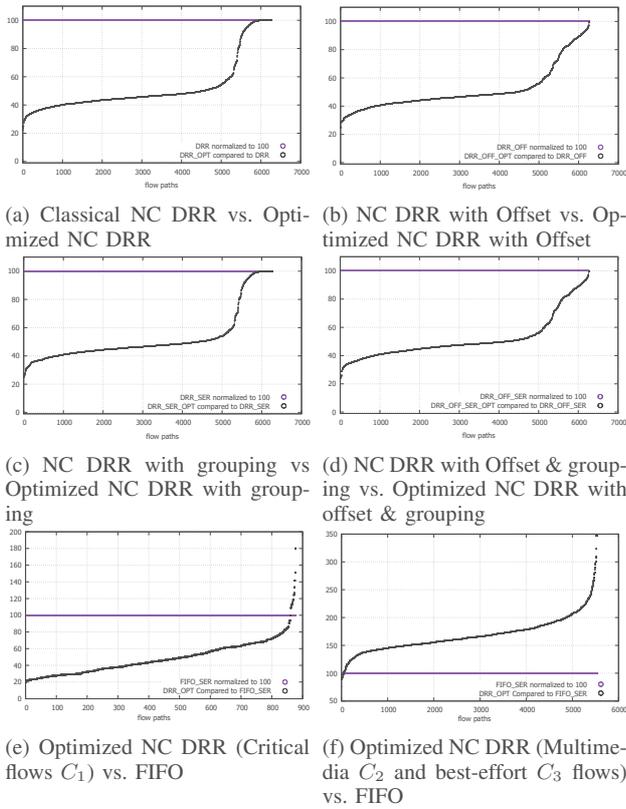


Fig. 11: Evaluation of Optimized NC approach

#### E. Comparison of Optimized NC with DRR and optimized NC with WRR

In this section, we compare our optimized WCTT analysis for DRR with the WCTT analysis for WRR presented in [5]. The WRR analysis cannot cope with the realistic case study presented in previous sections. Indeed, this analysis considers the largest frame size in each class. Since each class includes frames with very different sizes, the computation is very pessimistic and does not converge. This problem is not fully linked to the analysis. It mainly comes from the fact that

WRR cannot cope efficiently with packets with very different sizes in a given class.

A different case study is considered in [5]. It has a similar network architecture as well as a similar number of flows. However, as shown in Table VI, frame sizes are homogeneous per class. Table VII shows that DRR leads to better results for Classes  $C_1$  and  $C_2$  and not for  $C_3$ . Therefore, in that homogeneous case, no algorithm outperforms the other one.

TABLE VI: Configuration of WRR and DRR schedulers

class	No. of flows	DRR Quantum (bytes)	WRR weight (no. of packets)	frame size range
$C_1$	718	$4 \times l_{max}$	4	415-475
$C_2$	194	$2 \times l_{max}$	2	911-971
$C_3$	72	$1 \times l_{max}$	1	1475-1535

TABLE VII: Performance comparison of DRR and WRR schedulers

class	DRR vs WRR	
	avg difference (%)	max difference (%)
$C_1$	29.16	52.7
$C_2$	29.6	52.3
$C_3$	-35.4	-68.8

## VIII. CONCLUSION

Deficit Round Robin (DRR) scheduling policy has been defined for a fair sharing of server capacity among flows. Bandwidth sharing between traffic classes is fixed by the definition of quantum. DRR is envisioned for future avionics switched Ethernet networks, in order to improve bandwidth usage. It has good fairness properties and acceptable implementation complexity but a non-negligible latency which must be accurately evaluated.

This paper presents an improved method for the WCTT of DRR policy using network calculus. Our approach minimizes the pessimism in delay calculation using network calculus and gives tighter upper bounds on end-to-end delay as compared to previous studies. On an industrial-size case study, the proposed approach outperforms existing ones by 47 %. This improvement should allow to increase the number of flows transmitted on the network or to reduce the number of switches. We also show that, thanks to quantum, it is possible to achieve better performance for critical flows as compare to other scheduling policies like FIFO.

The WCTT analysis proposed in this paper is mandatory when critical flows are transmitted on the network. However, one goal of using DRR for avionics network is to be able to share the network between critical flows and less/not critical ones. For those later flows, an upper bound on the delay is not the most relevant metric. Thus, the WCTT analysis has to be coupled with a study of the delay distribution. Such a distribution can be obtained by simulation.

Allocating flows to traffic classes and assigning a quantum to each class has a significant impact on the end-to-end delays. We plan to precisely measure this impact in order to propose guidelines for the tuning of classes and quantum, based on traffic profiles.

## REFERENCES

- [1] J.-Y. L. Boudec and P. Thiran, *Network Calculus: a theory of deterministic queuing systems for the internet*. LNCS, April 2012, vol. 2050.
- [2] H. Bauer, J.-L. Scharbag, and C. Fraboul, "Improving the worst-case delay analysis of an afdx network using an optimized trajectory approach," *IEEE Trans. Industrial Informatics*, vol. 6, no. 4, Nov 2010.
- [3] H. Charara, J.-L. Scharbag, C. Fraboul, and J. Ermont, "Methods for bounding end-to-end delays on an afdx network," *Real-Time Systems. 18th Euromicro Conference on. IEEE*, p. 10, July 2006.
- [4] "Aircraft data network, parts 1,2,7 aeronautical radio inc." ARINC Specification 664, Tech. Rep., 2002 - 2005.
- [5] A. Soni, X. Li, J.-L. Scharbag, and C. Fraboul, "Wcct analysis of avionics switched ethernet network with wrs scheduling," *26th International Conference on Real-Time Networks and Systems (RTNS)*, 2018.
- [6] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on networking, 1996, vol. 4, no 3, p. 375-385*, p. 11, 1996.
- [7] M. Boyer, G. Stea, and W. M. Sofack, "Deficit round robin with network calculus," *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on (pp. 138-147)*. IEEE, p. 10, October 2012.
- [8] S. S. Kanhere and H. Sethu, "On the latency bound of deficit round robin," *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on. IEEE, 2002. p. 548-553.*, p. 7, October 2002.
- [9] M. Boyer, N. Navet, M. Fumey, J. Miggie, and L. Havet, "Combining static priority and weighted round-robin like packet scheduling in afdx for incremental certification and mixed-criticality support," *5TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*, 2013.
- [10] L. Lenzi, E. Mingozzi, and G. Stea, "Aliquem: a novel drr implementation to achieve better latency and fairness at  $o(1)$  complexity," *IEEE 2002 Tenth IEEE International Workshop on Quality of Service*, 2002.
- [11] Y. Hua and X. Liu, "Scheduling design and analysis for end-to-end heterogeneous flows in an avionics network," *2011 Proceedings IEEE INFOCOM*, April 2011.
- [12] A. Kos and S. Tomazic, "A more precise latency bound of deficit round-robin scheduler," *Elektrotehniški vestnik*, vol. 76, pp. 257–262, January 2009.
- [13] A. Soni, X. Li, J.-L. Scharbag, and C. Fraboul, "Integrating offset in worst case delay analysis of switched ethernet network with deficit round robin," *IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018.
- [14] X. Li, J.-L. Scharbag, and C. Fraboul, "Improving end-to-end delay upper bounds on an afdx network by integrating offsets in worst-case analysis," *IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–8, Sept 2010.