Worst-case Timing Analysis of Ring Networks with Cyclic Dependencies using Network Calculus

Ahmed Amari and Ahlem Mifdaoui
University of Toulouse - ISAE-Supaero
France

Abstract—**The ring networks guarantee a high availability level, while limiting cabling costs. However, their performance analysis is still a relevant concern due to cyclic dependencies. In this paper, we handle this challenging issue based on Network Calculus formalism. We propose a new approach, Pay Multiplexing Only at Convergence points (PMOC), to compute accurate delay bounds while integrating the impact of cyclic dependencies. We first define and prove the end-to-end service curve, guaranteed for a flow of interest under Arbitrary multiplexing. Afterwards, we detail the methodology of delay bounds computation and illustrate it in a special case of ring networks, called regular ring networks. Finally, we analyse the sensitivity and tightness of the derived delay bounds, and compare them against the related work results. We highlight a noticeable enhancement of delay bounds accuracy, thus network resource efficiency and scalability.**

***Index Terms*—Network Calculus; Ring; Cyclic dependencies; Performance evaluation; Delay upper bounds.**

I. Introduction

The recent research efforts towards defining new communication solutions, guaranteeing high availability level with limited cabling costs and complexity, have renewed the interest in ring-based networks. Such a topology provides actually an implicit redundant path by introducing only one additional connection between the two end nodes. The ring-based networks have been prominently used for several safety-critical applications, such as automation with the implementation of many Real Time Ethernet (RTE) profiles, e.g., EtherCAT [6] and Profinet-IRT [10]; and recently in other application fields like automotive, e.g. RACE [12], and avionics, e.g. AeroRing [1]. A relevant issue for such networks is proving time predictability, considered as a key requirement for safety-critical applications. Hence, to deal with their performance evaluation, accurate timing analysis to compute worst-case delays or at least upper bounds has to be considered.

For the most common Real-Time Ethernet (RTE) profiles supporting ring topologies, conducting such performance analyses has been greatly simplified due to their implemented time-triggered communication scheme, e.g. Master/slave or TDMA. Unlike these existing approaches, we are interested in this paper in event-triggered ring-based networks, which guarantee high resource utilization efficiency and (re)configuration flexibility but increase timing analysis complexity. The implementation of such a communication scheme on top of a ring topology actually induces cyclic dependencies, i.e., there exist interfering flows with paths forming cycles.

To cope with this arising issue of cyclic dependencies, only few techniques have been proposed in the literature, mainly based on Network Calculus [8]. The high modularity and scalability of such a framework make it particularly efficient to conduct timing analysis of complex communication networks [9], e.g., it has been recently used to certify the avionics standard AFDX [4]. Existing approaches are based on *iterative local analysis*, by successively computing the delay bound in each crossed node either directly, i.e., *Delay-based* methods [5][3] [7], or from the backlog bound, i.e., *Backlog-based* methods [13] [8]; and summing these delays up results in end-to-end delay bounds. However, these approaches lead to overly pessimistic upper bounds, which decrease the network scalability and resource efficiency as it will be illustrated in Section VI.

To handle these limitations, we propose in this paper a new *global analysis* based on Network Calculus, *Pay Multiplexing Only at Convergence points (PMOC)*, to enhance the end-to-end delay bounds accuracy of such networks. This consists in considering the flow serialization phenomena along the path of a ***flow of interest (f.o.i)***, by paying the bursts of interfering flows only at the convergence points[1]. Similar concepts have been developed in the literature for feedforward networks, i.e., with no cyclic dependencies, such as the Pay Bursts Only Once (PBOO) in [8] and the Pay Multiplexing Only Once (PMOO) in [11]. However, tightening the delay bounds of non-feedforward networks is still an open problem in the literature. The main objective of this paper is to handle such an issue for ring-based networks, a particular case of non-feedforward networks.

Hence, our main contributions are at both fundamental and practical levels:

(i) First, we introduce the concept of Pay Multiplex Only at Convergence points (PMOC) and define the guaranteed end-to-end service curve for a f.o.i along its path, in a ring network with cyclic dependencies under Arbitrary Multiplexing (Theorem 3). Then, we detail the main steps of our followed methodology to compute the delay upper bounds. Furthermore, this methodology has been illustrated in a special case of ring networks, called *regular ring networks*, for which the

[1]In ring-based networks, two flow paths may join at a node, called the convergence point, then disjoin after having a common subpath to maybe join again at another convergence point.

978-1-5386-1898-1/17/$31.00 © 2017 IEEE

computation of end-to-end delay bounds can be considerably simplified, under a necessary and sufficient condition on the maximum flow rate (Conjecture 1). These results constitute our main theoretical contributions.
(ii) Second, we analyse the sensitivity of the derived delay bounds with respect to the flows rate and path length, but also the network size. In doing this, we highlight the significant impact of the flows characteristics on the computed bounds. We also assess their tightness through a comparison with a WCD (Worst-Case Delay) Lower bound, i.e., an achievable worst-case delay. Finally, we review the existing approaches by identifying their main limitations, and benchmark the available results against ours. We highlight a noticeable enhancement of the delay bounds, thus network resource efficiency and scalability. These results constitute our main practical contributions.

The rest of the paper is organized as follows: we start with presenting the main concepts of the Network Calculus framework in Section II, and detailing the main system assumptions and model in Section III. Afterwards, we present our main contributions and report evaluation results in Sections IV and V, respectively. Finally, we report the main related work in Section VI, and draw the main conclusions and future work in Section VII.

II. Network Calculus Background

Network Calculus describes data flows by means of cumulative functions, defined as the number of transmitted bits during the time interval $[0, t]$. Consider a system S receiving input data flow with a Cumulative Arrival Function (CAF), $A(t)$, and putting out the same data flow with a Cumulative Departure Function (CDF), $D(t)$. Furthermore, S fulfills the causality condition, i.e., $\forall t \in \mathbb{R}^+, A(t) \geq D(t)$. Then, at the instant t, the delay and backlog are simply the vertical and horizontal distances between $A(t)$ and $D(t)$, respectively. However, to compute upper bounds on the worst-case delay and backlog, we need to introduce the maximum arrival curve, which provides an upper bound on the number of events, e.g., bits or packets, observed during any interval of time.

Definition 1. *(Arrival Curve)[8] A function α is an arrival curve for a data flow with the CAF A, iff:*

$$\forall t, s \geq 0, s \leq t, A(t) - A(s) \leq \alpha(t-s)$$

A widely used curve is the leaky-bucket curve, which guarantees a maximum burst σ and a maximum rate ρ, i.e., the traffic flow is (σ, ρ)-constrained. In this case, the arrival curve is defined as $\gamma_{\sigma,\rho}(t) = \sigma + \rho \cdot t$ for $t > 0$.

To conduct worst-case performance analysis, we need to put constraints on the input traffic through the maximum arrival curve notion. In return, we need to guarantee a minimum offered service within crossed nodes. This is feasible through the concept of minimum service curve, defined as follows:

Definition 2. *(Simple Minimum Service Curve)[8] The function β is the simple service curve for a data flow with the CAF A and the CDF D, iff:*

$$\forall t \geq 0, D(t) \geq \inf_{s \leq t}(A(s) + \beta(t-s))$$

A very useful and common model of service curve is the rate-latency curve $\beta_{R,T}$, with R the minimum guaranteed rate and T the maximum latency before starting the service. This rate-latency function is defined as $\beta_{R,T}(t) = [R \cdot (t-T)]^+$, where $[x]^+$ is the maximum between x and 0. Moreover, to model a node implementing aggregate scheduling, i.e., multiplexes the crossing flows at the input and demultiplexes them at the output, we need to define the *left-over service curve* guaranteed to each flow within the crossed node, considering the impact of contention with other traffic flows. The computation of such a left-over service curve depends on the implemented scheduling policy, and its derivation needs strict service curve property in the general case.

Definition 3. *(Strict service curve)[8] The function β is a strict service curve for a data flow with the CDF $D(t)$, if for any backlogged period [2] $]s, t]$, $D(t) - D(s) \geq \beta(t-s)$.*

The main results concerning the left-over service curves computation are as follows:

Theorem 1. *(Left-over service curve - Arbitrary Multiplex)[2] let f_1 and f_2 be two flows crossing a server that offers a strict service curve β such that f_1 is α_1-constrained, then the left-over service curve offered to f_2 is:*

$$\beta_2 = (\beta - \alpha_1)_\uparrow$$

where $f_\uparrow(t) = \max\{0, \sup_{0 \leq s \leq t} f(s)\}$

Knowing the arrival and service curves, one can compute the upper bounds on performance metrics for a data flow, according to the following theorem.

Theorem 2. *(Performance Bounds) Consider a flow constrained by an arrival curve α crossing a system S that offers a service curve β, then:*
Delay [3]: $\forall\ t:\ d(t) \leq h(\alpha, \beta)$
Backlog [4]: $\forall\ t:\ q(t) \leq v(\alpha, \beta)$
Output arrival curve [5]: $\alpha^*(t) = \alpha \oslash \beta(t)$

The calculus of these bounds is greatly simplified in the case of a leaky bucket arrival curve and a rate-latency service curve. In this case, the delay and backlog are bounded by $\frac{b}{R} + T$ and $b + r \cdot T$, respectively; and the output arrival curve is $b + r \cdot (T + t)$.

III. System Model

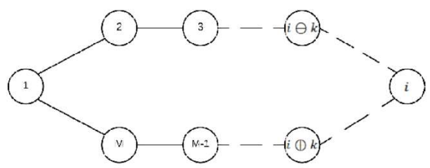

Fig. 1: Ring-based Network Example

[2] A backlogged period $]s, t]$ is an interval of time during which the backlog is non null, i.e., $A(s) = D(s)$ and $\forall u \in]s, t]$, $A(u) - D(u) > 0$

[3] $h(f, g)$: the maximum horizontal distance between f and g

[4] $v(f, g)$: the maximum vertical distance between f and g

[5] $f \oslash g(t) = \sup_{\forall u \geq 0}\{f(t+u) - g(u)\}$

We are interested in computing an upper bound on Worst-Case Delay for a flow of interest (f.o.i) in a ring network with cyclic dependencies. To conduct such a timing analysis, we consider the following assumptions and notations, using upper indices to indicate nodes or a set of nodes, and lower indices to indicate flows:

(i) We consider a unidirectional ring topology, as shown in Fig. 1, connecting M nodes, labelled from 1 to M, and serving a fixed set of flows I. The unidirectional topology is not restrictive, since a full-duplex ring can be considered as two independent unidirectional rings that can be analyzed separately;

(ii) Each flow $i \in I$ follows a fixed path from its initial source until the final sink, defined as $\mathbb{P}_i = (0, i.ft, i.ft \oplus 1, ..., i.ft \oplus (h_i - 1))$, where 0 is a virtual node representing the source, $i.ft$ the first hop and h_i the number of hops of flow i with $h_i \leq M$ and the notations $l \oplus k$ and $l \ominus k$ designate the $k-th$ node downstream and upstream from node l, respectively. Due to the ring topology, the first downstream node for node M is node 1 and the first upstream node for node 1 is node M. Moreover, for any flow i, the node $i.ft \ominus 1$ along its path corresponds to its source, i.e., 0. Afterwards, we define its subpath through $n \in [1, h_i]$ hops as $\mathbb{P}_i(n) = (0, i.ft, ..., i.ft \oplus (n-1))$, i.e., $\mathbb{P}_i = \mathbb{P}_i(h_i)$. It is worth noting that we consider only the output ports of crossed nodes within the subpath $\mathbb{P}_i(n)$. Finally, we assume that no two flows have the same path, since we can aggregate such flows (if any) and thus consider the aggregate flow;

(iii) Within the network, flows are treated according to an aggregate scheduling, i.e. flows are aggregated following a common parameter, such as the priority. Within an aggregate, flows are served under arbitrary multiplexing in each crossed node. It is worth noting that the worst-case for such a kind of multiplexing covers the worst-case of any type of multiplexing, e.g., First In First Out (FIFO) and Fixed Priority (FP);

(iv) We denote $i \ni k$ the set of flows crossing the node k, i.e., $i \ni k = \{i \in I / k \in \mathbb{P}_i\}$;

(v) Consider $\mathbb{K}_f(n)$ the set of interfering flows with a f.o.i f along its subpath $\mathbb{P}_f(n)$; so that $\mathbb{K}_f(n) = \{i \neq f / \exists k \in \mathbb{P}_f(n) / i \ni k\}$. Moreover, for any flow $i \in \mathbb{K}_f(n)$, consider its first (last) convergence point label with the f.o.i f along the subpath $\mathbb{P}_f(n)$ as $Mft(i, f, n)$ $(Mlt(i, f, n))$;

(vi) Each flow $i \in I$ has the CAF A_i^k and the CDF D_i^k at the node k. In addition, it is constrained by one leaky bucket of rate ρ_i and an initial burst σ_i^0 generated by the virtual node 0, thus it admits an arrival curve $\alpha_i^0(t) = \sigma_i^0 + \rho_i \cdot t$ at the input of the node $i.ft$. We define also its input arrival curve at each crossed node k along its path $\mathbb{P}_i$, as $\alpha_i^{k \ominus 1}(t) = \sigma_i^{k \ominus 1} + \rho_i \cdot t$, i.e., $0 \leq s \leq t$, $A_i^k(t) - A_i^k(s) \leq \alpha_i^{k \ominus 1}(t - s)$;

(vii) Each node k serves the traffic of an aggregate according to a strict service curve having a rate-latency form, with a rate R^k and a latency T^k, $\beta^k(t) = [R^k \cdot (t - T^k)]^+$, i.e., for any backlogged period $]s, t]$, $\sum_{i \ni k} D_i^k(t) - D_i^k(s) \geq [R^k \cdot (t - s - T^k)]^+$;

(viii) We consider the cases where the following condition is satisfied: for any node $k \in [1, M]$, $\frac{\sum_{i \ni k} \rho_i}{R^k} \leq 1$. This condition is necessary to guarantee finite delay bounds within each crossed node.

The main notations used in this paper are summarized in Tab. I in Appendix A.

IV. PMOC Approach

In this section, we detail the main idea and steps of our proposed PMOC approach, to compute delay upper bounds in ring networks with cyclic dependencies. We first introduce the main concept progressively through an example. Then, we define and prove the closed-form service curve in such a network, under arbitrary multiplexing. Afterwards, based on these defined service curves, the necessary and sufficient condition to compute end-to-end delay bounds is discussed, and finally specified for a special case of ring networks.

A. Illustrative Example

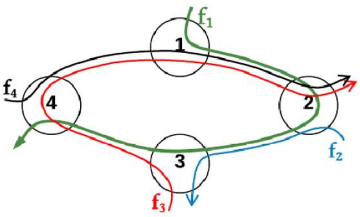

Fig. 2: Ring network with cyclic dependency.

Fig. 2 illustrates an example of the cyclic dependency problem. Consider as a f.o.i f_1 with the path $\mathbb{P}_{f_1} = (0, 1, 2, 3)$. It is worth noting that node 4 is not part of $\mathbb{P}_{f_1}$, since f_1 crosses only the input port of 4, i.e., as explained in assumption (ii) of Section III, we consider only the output ports of crossed nodes along the path. To compute the end-to-end delay bound of f_1, we need to integrate the impact of all the interfering flows along its path, $\mathbb{K}_{f_1}(3) = \{f_2, f_3, f_4\}$. Hence, at the input of node 1, we need to quantify the arriving bursts of flows f_3 and f_4. Moreover, the burst of f_4 at the input of node 1 depends on the burst of f_3 at the input of node 4, which in its turn depends on the burst of the f.o.i f_1 at the input of node 3. As we can notice, to analyse the impact of interfering flows on the f.o.i f_1, we need to quantify its impact on these interfering flows; thus the cyclic dependency. There is actually no start point, where all the flows bursts are known, to launch the delay computation.

To overcome this difficulty, the main idea of PMOC approach is to compute the tightest possible upper bound on these unknown bursts, when considering the flow serialization phenomena along the path of the f.o.i, and integrating the impact of interfering flows only at the convergence points. As illustrated in Fig. 2, because of the ring topology, there are only two possible convergence points with a f.o.i: (i) If the convergence point is the interfering flow source, then the burst impacting the f.o.i is known; (ii) If the convergence point is the source of the f.o.i, then the burst impacting the f.o.i is unknown.

Consider the example of computing the unknown burst of f_4 at the input of node 1. To compute such a propagated burst, we need to quantify the minimum guaranteed service of f_4 until reaching the input of its convergence point with the f.o.i f_1, i.e., the service along $\mathbb{P}_{f_4}(1) = (0, 4)$. However, this service depends on the burst of f_3 at the input of node 4, which depends in its turn on the minimum guaranteed service of f_3 until reaching the input of node 4, i.e., the service along $\mathbb{P}_{f_3}(1) = (0, 3)$. Detailing such dependencies for all flows crossing the network reveals actually the need to quantify the service curve guaranteed to each flow f along each of its subpaths, i.e., the service along $\mathbb{P}_f(n)$, $\forall n \leq h$.

Expressing the service curves and the propagated bursts, for any flow along any of its subpaths, defines a system of linear equations. The latter is solved using matrices, when a given necessary and sufficient condition is verified. Afterwards, we can compute the delay upper bounds. These different steps of our proposed PMOC approach are detailed in next sections.

B. Service Curve for the Flow of Interest

We focus herein on the first step of the PMOC approach, which consists in defining the guaranteed service curve for a f.o.i along any of its subpaths in a ring network. Such a curve under arbitrary multiplexing within the crossed nodes is defined in Th. 3.

Theorem 3. *(Service Curve in Ring Network under Arbitrary Multiplexing) The service curve offered to a f.o.i f along its subpath, $\mathbb{P}_f(n)$, in a ring network under arbitrary multiplexing with strict service curve nodes of the rate-latency form $\beta_{R,T}$ and leaky bucket constrained arrival curves $\alpha_{\sigma,\rho}$, is a rate-latency curve, with a rate $R^{\mathbb{P}_f(n)}$ and a latency $T^{\mathbb{P}_f(n)}$, defined as follows:*

$$R^{\mathbb{P}_f(n)} = \min_{k \in \mathbb{P}_f(n)} [R^k - \sum_{i \ni k, i \neq f} \rho_i] \tag{1a}$$

$$T^{\mathbb{P}_f(n)} = \sum_{k \in \mathbb{P}_f(n)} T^k + \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^0 \cdot 1_{\{f \ni i.ft\}} + \rho_i \cdot \sum_{k \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^k}{R^{\mathbb{P}_f(n)}}$$
$$+ \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^{f.ft \ominus 1} \cdot 1_{\{i \ni f.ft / i.ft \neq f.ft\}}}{R^{\mathbb{P}_f(n)}}, \tag{1b}$$

where $1_{\{cdt\}}$ is equal to 1 if cdt is true and zero otherwise.

The proof of Th. 1 is provided in Appendix A. As shown in Eq. (1b), some flow bursts are payed twice. These particular flows have actually two convergence points with the f.o.i: their own source and the f.o.i source. This fact respects the principle of the PMOC approach, introduced in Section IV-A.

We detail here the end-to-end service curve of the f.o.i f_1 in the example of Fig. 2, when the assumptions of the system model detailed in Section III are fulfilled, and all the crossed nodes offer the same service curve $\beta_{R,T}$. According to Th. 3, this service curve is a rate-latency curve, with a rate $R^{\mathbb{P}_{f_1}(3)} = \min[R-\rho_3-\rho_4, R-\rho_2, R-\rho_3]$ and a latency $T^{\mathbb{P}_{f_1}(3)} = 3 \cdot T + \frac{1}{R^{\mathbb{P}_{f_1}(3)}} \cdot (\sigma_2^0 + \rho_2 \cdot T + \sigma_3^0 + \rho_3 \cdot (2 \cdot T) + \rho_4 \cdot T) + \frac{1}{R^{\mathbb{P}_{f_1}(3)}} \cdot (\sigma_3^4 + \sigma_4^4)$.

C. Computation of Delay Upper Bounds

Now that we have expressed the service curve guarantees for each f.o.i along any of its subpaths, we can move to the second step of the PMOC approach, which consists in computing the delay bounds. All the system constraints in a ring network under arbitrary multiplexing, which depend on some variables are as follows:

(i) **Service Curve Constraint**

$\forall f \in I, \forall n \leq h_f$, for any $[s, t]$,

$$R^{\mathbb{P}_f(n)} = \min_{k \in \mathbb{P}_f(n)} [R^k - \sum_{i \ni k, i \neq f} \rho_i]$$

$$T^{\mathbb{P}_f(n)} = \sum_{k \in \mathbb{P}_f(n)} T^k + \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^0 \cdot 1_{\{f \ni i.ft\}} + \rho_i \cdot \sum_{k \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^k}{R^{\mathbb{P}_f(n)}} + \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^{f.ft \ominus 1} \cdot 1_{\{i \ni f.ft / i.ft \neq f.ft\}}}{R^{\mathbb{P}_f(n)}}$$

(ii) **Output Arrival Curve Constraint**

$\forall f \in I, \forall n \leq h_f$,

$$\sigma_f^{f.ft \oplus (n-1)} = \sigma_f^0 + \rho_f \cdot T^{\mathbb{P}_f(n)}$$

(iii) **Delay bound**

$\forall f \in I, \forall n \leq h_f$,

$$EED_f^{\mathbb{P}_f(n)} = \frac{\sigma_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

Hence, this set of linear constraints can be written in a matrix form as follows:

(i) **Service Curve Constraint**

$$\overbrace{\begin{bmatrix} T^{\mathbb{P}_f(1)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T} = \overbrace{\begin{bmatrix} c1_{f1} \\ \vdots \\ c1_{fh_f} \\ \vdots \end{bmatrix}}^{C1} + \overbrace{\begin{bmatrix} a1_{f,1} & \cdot\cdot & a1_{f,h_f} & \cdot\cdot \\ \vdots & \ddots & \ddots & \\ a1_{fh_f,1} & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \vdots & \vdots & \ddots & \end{bmatrix}}^{A1} \cdot \overbrace{\begin{bmatrix} \sigma_f^{f.ft} \\ \vdots \\ \sigma_f^{f.ft \oplus (h_f - 1)} \\ \vdots \end{bmatrix}}^{\sigma}$$

where T is the vector that holds the latencies of the offered service (given in Eq. (1b)), $A1$ is the matrix of the coefficients of unknown propagated bursts and $C1$ is the vector of constants, i.e, the latencies T^i and initial bursts transmission times appearing in the service curve constraint.

(ii) **Output Arrival Curve Constraint**

$$\overbrace{\begin{bmatrix} \boldsymbol{\sigma}_f^{f.ft} \\ \vdots \\ \boldsymbol{\sigma}_f^{f.ft\oplus(h_f-1)} \\ \vdots \end{bmatrix}}^{\boldsymbol{\sigma}} = \overbrace{\begin{bmatrix} c2_{f1} \\ \vdots \\ c2_{fh_f} \\ \vdots \end{bmatrix}}^{C2} + \overbrace{\begin{bmatrix} a2_{f,1} & \cdot\cdot & a2_{f,h_f} & \cdot\cdot \\ \vdots & \ddots & \ddots & \\ a2_{fh_f,1} & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \vdots & \vdots & \ddots & \end{bmatrix}}^{A2} \cdot \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(1)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

where $\boldsymbol{\sigma}$ is the vector of the unknown propagated bursts, $A2$ is the matrix of the coefficients of the corresponding unknown offered service latencies, i.e., the flow rate, and $C2$ is the vector of constants, i.e. the initial bursts appearing in the output arrival curve constraint.

(iii) **Delay bound**

$$\overbrace{\begin{bmatrix} EED^{\mathbb{P}_f(1)} \\ \vdots \\ EED^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{EED} = \overbrace{\begin{bmatrix} c3_{f1} \\ \vdots \\ c3_{fh_f} \\ \vdots \end{bmatrix}}^{C3} + \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(1)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

where $C3$ is the vector of constants, i.e., the initial bursts transmission times appearing in the delay bound constraint.

When propagating the different constraint, this matrix form is transformed to the following ($\mathbb{M}$*):

$$\begin{cases} (Id - A1 \times A2) \times T = C1 + A1 \times C2 \\ EED = C3 + T \end{cases} \quad (2)$$

Based on the matrix form $\mathbb{M}$*, we deduce in the following corollary a necessary and sufficient condition on the existence of delay upper bounds for each f.o.i along any of its subpaths, in the general case of ring networks under arbitrary multiplexing. This condition will be detailed in the next section for a special case of ring networks.

Corollary 1. *(Delay Bound under Arbitrary Multiplexing) In a ring network under arbitrary multiplexing, the delay upper bound of each f.o.i f along its subpath $\mathbb{P}_f(n)$ exists and is at most equal to*

$$EED_f^{\mathbb{P}_f(n)} = \frac{\boldsymbol{\sigma}_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

if and only if the matrix $(Id - A1 \times A2)$ in $\mathbb{M}$ is invertible, i.e., its determinant is not zero.*

Proof. Based on known results in linear algebra, we can see from $\mathbb{M}$* that the vector of latencies T exists, if and only if the square matrix $(Id - A1 \times A2)$ is invertible. Under this necessary and sufficient condition, we have $T = (Id - A_1 \times A_2)^{-1} \times (C1 + A1 \times A2)$. Consequently, $EED = C3 + (Id - A_1 \times A_2)^{-1} \times (C1 + A1 \times A2)$ exists. This finishes the proof of Corollary 1. □

D. Special Case: Regular Ring Networks

We introduce herein a particular case of ring networks, called regular ring networks, for which we deduce a specific necessary and sufficient condition for the existence of delay upper bounds, in comparison to the general one in Cor. 1.

Definition 4. *(Regular Ring Network) A ring network connecting M nodes is a regular ring network with a degree h, where $2 \leq h \leq M$, when it satisfies the following assumptions: (i) all the nodes guarantee the same rate-latency service curve, $\boldsymbol{\beta}_{R,T}$; (ii) each node $l \in [1, M]$ is generating a $(\boldsymbol{\sigma}, \boldsymbol{\rho})$-constrained flow, destined to all its k-th downstream nodes from l, $\forall k \leq h$.*

It is worth noting that a ring network with a broadcast communication pattern is a regular ring network with a degree $h = M$. We have the following conjecture on the delay bounds in regular ring networks, based on a more specific necessary and sufficient condition than the one in Cor. 1:

Conjecture 1. *(Delay Bound in Regular Ring Networks) In a regular ring network under arbitrary multiplexing and with a degree h, the delay upper bound of each f.o.i f along its subpath $\mathbb{P}_f(n)$ exists and is at most equal to*

$$EED_f^{\mathbb{P}_f(n)} = \frac{\boldsymbol{\sigma}_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

if and only if the following equivalent conditions are verified:
(i) (Flow rate Cdt.) The maximum rate of each generated $(\boldsymbol{\sigma}, \boldsymbol{\rho})$-constrained flow is as follows: $\boldsymbol{\rho} < \frac{R}{2 \cdot (h-1)}$;
(ii) (Utilization rate Cdt.) The maximum utilization rate of the network, $U_{max} = h \cdot \boldsymbol{\rho}/\boldsymbol{R}$, is as follows: $U_{max} < \frac{h}{2 \cdot (h-1)}$. Thus, as $h \to \infty$, the maximum utilization rate tends to 50%.

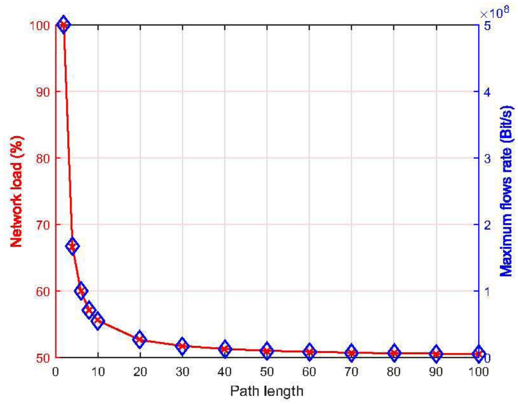

Fig. 3: Maximum network utilization and flow rate vs network degree, i.e. flow path length, for which the determinant of the matrix $(Id - A1 \times A2)$ in $\mathbb{M}$* vanishes

This conjecture is based on the observation of the behavior of the maximum utilization rate (resp. maximum flow rate), satisfying the necessary and sufficient condition of Cor. 1, for regular ring networks when varying the degree $h \in [2, M]$, as illustrated in Fig. 3. We actually have built the associated matrix form $\mathbb{M}$* for $h \in [2, 100]$ and $R = 1Gb/s$. Then, based on a symbolic computation tool, we have computed the

maximum utilization rate of the network (resp. maximum flow rate), for which the determinant of the matrix $(Id - A1 \times A2)$ in $\mathbb{M}^*$ vanishes. As we can see, The maximum network utilization rate decreases from 100% for $h = 2$ to 50.5% for $h = 100$, **while the maximum flow rate decreases from** $\frac{R}{2}$ for $h = 2$ to $\frac{R}{198}$ for $h = 100$. These values are coherent with the **upper bounds defined in the Conjecture 1, which are** $\frac{h}{2\cdot(h-1)}$ for the maximum network utilization rate and $\frac{R}{2\cdot(h-1)}$ for the **maximum flow rate. It is worth noting that the maximum** utilization rate in Conjecture 1 is more restrictive than the one in Section III, i.e., $h \cdot \rho/R \leq 1$.

Example

We now explicit the matrix form $\mathbb{M}^*$ and the necessary and **sufficient condition on the existence of delay bounds for a** simple example. Consider a regular ring network with 3 nodes, labeled from 1 to 3, and a degree $h = 2$ under arbitrary multiplexing, as illustrated in Fig. 4. Each node i sends a (σ^0, ρ)**-constrained flow** f_i and guarantees a service curve $\beta_{R,0}$. The aim is to compute the end-to-end delay bound of the f.o.i f_1.

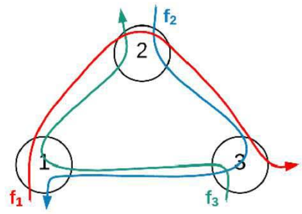

Fig. 4: Example of a regular ring network with $M = 3$ and $h = 2$

First, we explicit the different parameters of the matrix form $\mathbb{M}^*$ of such a network as follows:

$$T^T = (T^{\mathbb{P}_{f_1}(1)}, T^{\mathbb{P}_{f_1}(2)}, T^{\mathbb{P}_{f_2}(1)}, T^{\mathbb{P}_{f_2}(2)}, T^{\mathbb{P}_{f_3}(1)}, T^{\mathbb{P}_{f_3}(2)})$$
$$C1^T = \frac{\sigma^0}{R-\rho} \cdot (0,1,0,1,0,1)$$
$$A_1 = \frac{1}{R-\rho} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$
$$\sigma^T = (\sigma^1_{f_1}, \sigma^2_{f_1}, \sigma^2_{f_2}, \sigma^3_{f_2}, \sigma^3_{f_3}, \sigma^1_{f_3})$$
$$C2^T = \sigma^0 \cdot (1,1,1,1,1,1,) \text{ and } A2 = \rho \cdot I_{(h \times M)}$$

Then, to verify the necessary and sufficient condition defined in Cor. 1, we express the determinant of the matrix $(Id - A_1 \times A_2)$, which is as follows:

$$(\rho - \frac{R}{2}) \cdot (-2\rho^2 + 2R\rho - 2R^2)/(R-\rho)^3$$

This function vanishes for the maximum flow rate $\rho = \frac{R}{2}$. This value is coherent with the Conjecture 1, where the **upper bound of the maximum flow rate is** $< R/2 \cdot (h-1)$, i.e., $R/2$ for $h = 2$. **Hence, if the flow rate condition is verified, i.e.,** $\rho < R/2$, then the end-to-end delay upper bound of the f.o.i f_1, $EED^{\mathbb{P}_{f_1}(2)}_{f_1}$, exists and is at most equal to $\frac{\sigma^0}{R^{\mathbb{P}_{f_1}(2)}} + T^{\mathbb{P}_{f_1}(2)}$, where $R^{\mathbb{P}_{f_1}(2)} = R - \rho$ and $T^{\mathbb{P}_{f_1}(2)} = \frac{2\sigma^0}{R-\rho} + \frac{\sigma^0\rho(\rho^2 - R\rho + R^2)}{(R-\rho)(R^3 - 3R^2\rho - 2\rho^3)}$.

V. Numerical Results

In this section, we detail some numerical results of the delay upper bounds of a f.o.i. in a ring network with cyclic dependencies, under different scenarios, when applying our approach PMOC. First, we describe the considered case study and scenarios. Then, we report the sensitivity analysis of the **computed upper bounds with respect to flows rate and path** length, for various values of network size M. Finally, we assess their tightness through a lower bound on WCD (Worst-Case Delay) in several scenarios.

A. *Case study and scenarios*

We consider the case study with the following assumptions: (i) The topology is a unidirectional ring topology connecting M nodes, as the one illustrated in Fig. 1; (ii) All nodes guarantee a rate-latency service curve $\beta_{R,T}$ with $R = 1Gb/s$ and $T = 600ns$; (iii) Each node generates one leaky-bucket **constrained flow with a burst** σ and a rate ρ; (iv) The network is a regular ring network with a degree h.

To analyze the sensitivity of the computed delay bounds and to assess their tightness, we consider various network **configurations, where each network configuration is defined** with the tuple (σ, ρ, h, M). **We vary the flow rate** ρ or the **flow path** h of this tuple at a time, to highlight its impact on the computed delay bounds.

B. *Sensitivity analysis*

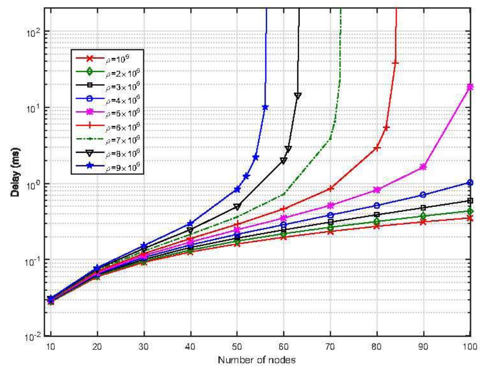

Fig. 5: The impact of flow rate on the delay bound vs network size for $(\sigma = 128\textbf{bytes}, \rho \in [1-9]Mb/s, h = M, M \in [10-100])$.

Fig. 5 shows the impact of the flow rate on the delay bounds. As we can notice, there are two distinguishable **behaviors of the delay bounds: (i) when the flow rate condition in Conjecture 1 is verified, the delay bounds grow logarithmically in terms of the flow rate, e.g., for** $M = 40$, when the rate increases from 1 Mb/s to 9Mb/s, the delay bound grows from almost 10^{-2}ms to 3.10^{-2}ms; (ii) when **this condition is violated, the delay bound tends to infinity,**

e.g., for $\rho = 8Mb/s$, the delay bound diverges for a network size higher than $M = 63$, which corresponds to the condition $\rho < \frac{R}{2(M-1)} \Leftrightarrow M < \frac{R}{2\rho} + 1 = 63.5$. This fact infers an exponential growth of the delay bounds with the network size, when the flow rate condition achieves its limit. These results show the inherent impact of the flow rate on the delay bounds.

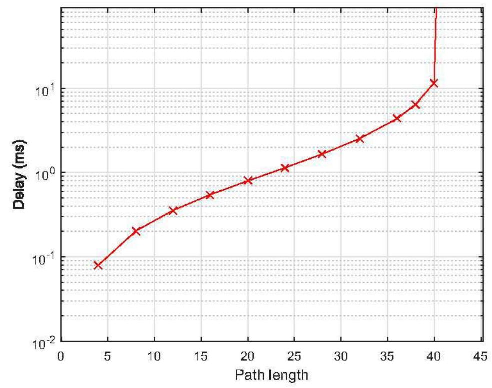

Fig. 6: The impact of the flow path on delay bound for ($\sigma = 1500$*bytes*$, \rho = 12Mb/s, h \in [4-45], \forall M > h$).

Fig. 6, shows the impact of the flow path length on the delay bounds. As it is shown, the delay bound has similar behavior in terms of flow path length than its rate, i.e., grows logarithmically when the flow rate condition is verified. Increasing the flow path length induces a higher number of interfering flows along the path; thus a higher service latency and lower service rate according to the PMOC approach. It is worth noting that the delay bounds for regular ring networks depend only on the network degree h, i.e., flow path length. For instance, the delay bound is $0.79ms$ for $h = 20$ independently from the network size. This result is coherent with Conjecture 1.

These results show that the delay bounds computed with the PMOC approach are particularly sensitive to the flow rate and path length. This fact is mainly due to the conditions defined in Conjecture 1, which depend on both parameters and infer an exponential behavior of the delay bounds when they achieve their limit.

C. Tightness analysis

To investigate the tightness of our approach, we compare the delay bounds obtained with our proposed method to an achievable worst-case delay, denoted as WCD lower bound. The latter is computed when considering an intuitive worst-case scenario, which consists in integrating for each flow of interest only the impact of downstream flows interferences within each crossed node, and ignoring the impact of the upstream flows at its source node, i.e., this is the unknown variable due to cyclic dependency. The size of the interval between the computed upper delay bounds and WCD lower bounds will give us an idea about the delay bound tightness, i.e., this interval includes the exact worst-case delay.

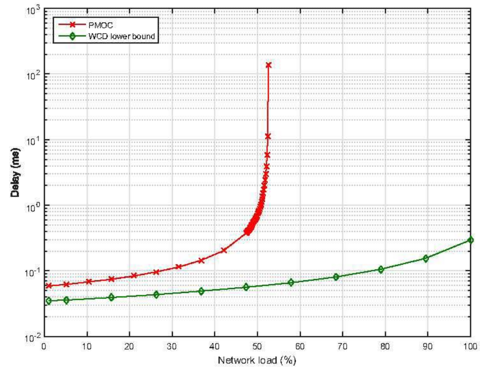

Fig. 7: Impact of the flow rate on delay bound tightness for ($\sigma = 128$*bytes*$, \rho = [0.5-50]Mb/s, h = M, M = 20$).

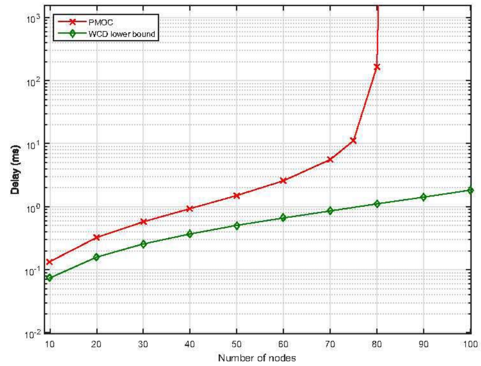

Fig. 8: Impact of network size on delay bound tightness for ($\sigma = 787$*bytes*$, \rho = 6.3Mb/s, h = M, M \in [10-100]$).

As illustrated in Figs. 7 and 8, if the utilization rate condition is verified, then the gap between the delay bound computed with the PMOC approach and the WCD lower bound is still bounded, e.g., up to 0.2ms for a network load up to 42%; else, we can not conclude on the delay bound tightness since it tends to infinity.

These results show that: if the network utilization rate condition is verified, then the delay bounds computed with the PMOC approach are noticeably accurate.

VI. Related work

The most relevant approaches related to computing end-to-end delay bounds in networks with cyclic dependencies are based on an iterative approach, i.e., successively analyzing the delay bound in each crossed node in the network resulting in end-to-end delay bounds computation. These approaches are focusing on, either each crossed node delay bound [5] [3] [7], or each crossed node backlog bound [13] [8]. For the particular case of ring-based network, Cruz [5] defines an interesting approach, called *Time Stopping Method*, which consists of two steps. First, a finite burstiness bound for the transmitted flows is assumed to compute the delay bounds. Then, the feasibility conditions to solve these equations are defined. The latter infers a maximum network utilization rate upper bounded by $\frac{2}{(M-1)}$. This fact implies that the network has to be under utilized to satisfy such a condition, which limits the network resource-efficiency. Another interesting approach in this area has been proposed in [13] and then generalized in [8] to prove the ring stability through the existence of a backlog bound, called ***Backlog-based Method***. The maximum bound on the delay within a node is the processing time of the maximum backlogged traffic and the end-to-end delay communication bound is the sum of the crossed nodes delays. Hence, the delay bound is growing as $\Theta(M^4)$, which inherently limits the network scalability.

To benchmark the delay bounds obtained with the PMOC approach against these main existing approaches, we consider a regular ring network of M nodes with a degree $h = M$, where each node has a transmission capacity 1Gb/s and generates a traffic flow with an arrival curve $\alpha \sim (128bytes, 128Kb/s)$ and a deadline of $1ms$.

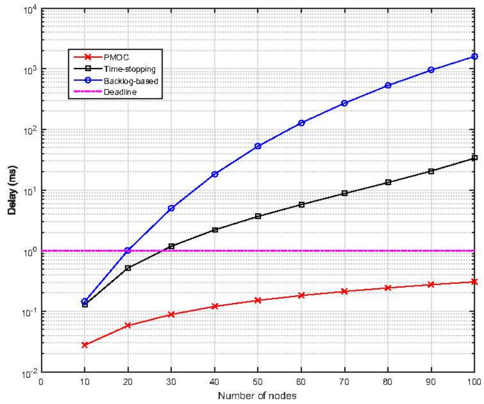

Fig. 9: End-to-end delay bounds vs number of nodes for ($\sigma = 128bytes, \rho = 128Kb/s, h = M, M \in [10-100]$).

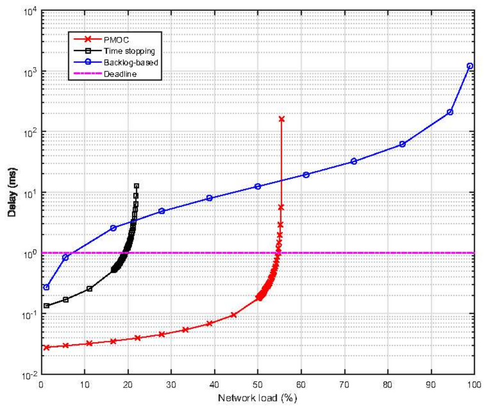

Fig. 10: End-to-end delay bounds vs network utilization rate for ($\sigma = 128bytes, \rho \in [1-100]Mb/s, h = M, M = 10$).

Fig. 9 shows a comparison of the different approaches when enlarging the network size. Obviously, the delay bounds increase with the network size. As we can notice, the PMOC approach offers tighter delay bounds for large-scale networks, while guaranteeing the flow deadline, in comparison with the conventional methods. Hence, the maximum network size respecting the flow deadline is about 20 and 27 nodes with the backlog-based and time-stopping methods, respectively, whereas it achieves 100 nodes with PMOC approach.

Fig. 10 illustrates the impact of increasing the congestion with the different methods. As we can see, the time stopping method diverges for a global utilization rate around 22.22%, which corresponds to $\frac{2}{M-1}$; whereas it achieves 55.55% with our proposed approach, which corresponds to the upper bound $\frac{M}{2(M-1)}$ in Conjecture 1. Furthermore, the maximum network utilization rate respecting the flow deadline is only about 7.1% and 19.36% with the backlog-based and time-stopping methods, respectively, compared to 54.6% with PMOC.

Discussion: this comparative analysis shows that using PMOC approach yields enhanced network performance in terms of resource efficiency and network scalability, in comparison with the conventional timing analyses. To guarantee the temporal deadlines, the Time Stopping method actually limits the network utilization rate, whereas the Backlog-based method limits the network size.

VII. Conclusions and Future Work

In this paper, we have introduced a new approach, called PMOC, to compute delay bounds in ring networks with cyclic dependencies. This proposed approach is based on a global analysis method, integrating the flow serialization phenomena along the flow path, to allow the computation of tighter end-to-end delay bounds. Hence, we have defined and proved the guaranteed end-to-end service curve of any f.o.i crossing such a network under arbitrary multiplexing. Afterwards, the methodology to compute delay bounds have been presented in the general case, and illustrated for regular ring networks. Finally, the numerical results have highlighted the accuracy of our proposed approach, in comparison to conventional methods, which yields enhanced network performance in terms of resource efficiency and network scalability.

The next step of this work is to extend the PMOC approach to compute delay bounds in the general case of non-feedforward networks.

Acknowledgment

The authors would like to thank Prof. Jens B. Schmitt from University of Kaiserslautern for the fruitful discussion and interesting comments, which helped to improve this paper.

Appendix A
Proof of Theorem 4 and Notations

Proof. We distinguish three possible categories for an interfering flow i with the f.o.i f: (i) *category 1*: having only one convergence point with f, which is its first hop, i.e., $i.ft$; (ii) *category 2* having only one convergence point with f, which is the first hop of f, i.e., $f.ft$; (iii) *category 3* having two distinct convergence points with f, i.e., $i.ft$ and $f.ft$ if $i.ft \neq f.ft$.

Consider flow f_1 in Fig. 2 as the f.o.i, then flows f_2, f_4 and f_3 are in categories 1, 2 and 3, respectively.

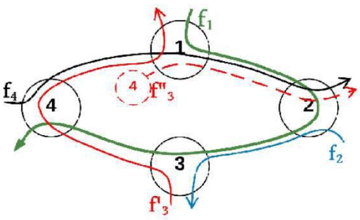

Fig. 11: Splitting virtually the flows of Fig. 2

To prove the Th. 3, we split an interfering flow i of category 3 in two subflows to cut virtually the cyclic dependency with the f.o.i f, as illustrated in Fig. 11 for flow f_3: (i) $i1$: the subflow of i along its subpath $\mathbb{P}_{i1} = (0, i.ft, i.ft \oplus 1, ..., f.ft \ominus 1)$, which is (σ_i^0, ρ_i)-constrained; (ii) $i2$: the subflow of i along its subpath $\mathbb{P}_{i2} = (f.ft \ominus 1, f.ft, ..., i.ft \oplus (h_i - 1))$, which is $(\sigma_i^{f.ft\ominus 1}, \rho_i)$-constrained. It is worth noting that this splitting operation is only needed to prove the theorem and not to apply the PMOC approach. As we can notice, $i1$ fulfills the conditions of category 1, whereas $i2$ fulfills the ones of category 2. Thus, splitting virtually the flows of category 3 in $\mathbb{K}_f(n)$ in two subflows leads to a transformed set $\overline{\mathbb{K}_f(n)}$. The latter can be rewritten according to the conditions of categories 1 and 2 as follows:

$$\{i \in \overline{\mathbb{K}_f(n)} / f \ni i.ft\} \cup \{i \in \overline{\mathbb{K}_f(n)} / i \ni f.ft, i.ft \neq f.ft\}$$

For the f.o.i f_1 in Fig. 2, the only flow of category 3 is the flow f_3. So, f_3 is virtually split as $(f_3', f"_3)$ as shown in Fig. 11, where $\mathbb{P}_{f_3'} = \{0, 3, 4\}$ and $\mathbb{P}_{f"_3} = \{4, 1\}$. It is worth noting that according to this model, the virtual node representing the source of flow $f"_3$ is node 4. Moreover, the set of interfering flows with the f.o.i f_1, $\mathbb{K}_{f_1}(3)$, is transformed to $\overline{\mathbb{K}_{f_1}(3)} = \{f_2, f_3'\} \cup \{f_4, f"_3\}$.

Consider a flow of interest f with a subpath $\mathbb{P}_f(n)$. Any crossed node $l \in \mathbb{P}_f(n)$ admits a strict service curve. Hence, according to Def. 3, for any instant $t_l \geq 0$, there exists $t_{l\ominus 1} \leq t_l$ the start of the backlogged period such that:

$$D_f^l(t_l) - D_f^l(t_{l\ominus 1}) + \sum_{i \ni l, i \neq f} (D_i^l(t_l) - D_i^l(t_{l\ominus 1})) \geq \beta^l(\Delta_l) \quad (3)$$

where $\Delta_l = t_l - t_{l\ominus 1}$. The time indices are chosen to match the node indices. Then, we sum up the expression in Eq. (3) when varying $l \in \mathbb{P}_f(n)$, which infers:

$$\begin{aligned}&\sum_{l \in \mathbb{P}_f(n)} D_f^l(t_l) - D_f^l(t_{l\ominus 1}) \qquad (4)\\ &\geq \sum_{l \in \mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{l \in \mathbb{P}_f(n)} \sum_{i \ni l, i \neq f} (D_i^l(t_l) - D_i^l(t_{l\ominus 1}))\end{aligned}$$

We list the following main results, which allow transforming Eq. (4) into Eq. (5): (i) the definition of $\overline{\mathbb{K}_f(n)}$ induces $\sum_{l \in \mathbb{P}_f(n)} \sum_{i \ni l, i \neq f} \Leftrightarrow \sum_{i \in \overline{\mathbb{K}_f(n)}} \sum_{l \in \mathbb{P}_f(n) \cap \mathbb{P}_i}$; (ii) at the start of a backlogged period s, $D_f^{i\oplus 1}(s) = A_f^{i\oplus 1}(s)$; (iii) because of the ring topology, we have $A_f^{i\oplus 1}(s) = D_f^i(s)$, and consequently $D_f^{i\oplus 1}(s) = D_f^i(s)$; (iv) the definitions of $Mft(i, f, n)$ and $Mlt(i, f, n)$ in Tab. I simplify the arrangement of Eq. (5).

$$\begin{aligned}
&D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1})\\
\geq &\sum_{l \in \mathbb{P}_f(n)} \beta^l(\Delta_l)\\
- &\sum_{i \in \overline{\mathbb{K}_f(n)}} D_i^{Mlt(i,f,n)}(t_{Mlt(i,f,n)}) - D_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1})\\
\geq &\sum_{l \in \mathbb{P}_f(n)} \beta^l(\Delta_l)\\
- &\sum_{i \in \overline{\mathbb{K}_f(n)}} A_i^{Mlt(i,f,n)}(t_{Mlt(i,f,n)}) - A_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1})\\
\geq &\sum_{l \in \mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{i \in \overline{\mathbb{K}_f(n)}} \alpha_i^{Mft(i,f,n)\ominus 1}\Big(\sum_{l=Mft(i,f,n)}^{Mlt(i,f,n)} \Delta_l\Big)
\end{aligned} \quad (5)$$

It is worth noting that to substitute the cumulative traffic functions of flows in $\overline{\mathbb{K}_f(n)}$ in Eq. (5) by their arrival curves, we have used the causality constraint of cumulative traffic functions, i.e., $\forall t, A_i^k(t) \geq D_i^k(t)$ and the property of the start of backlogged period at $t_{Mft(i,f,n)\ominus 1}$, i.e., $D_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1}) = A_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1})$.

On the other hand, rewriting the input arrival curve of a flow i at node k, $\alpha_i^{k\ominus 1}$, using $\overline{\alpha_i}(\Delta_l) = \rho_i \cdot \Delta_l$, infers:

$$\begin{aligned}
\alpha_i^{k\ominus 1}\Big(\sum_{l=1}^{m} \Delta_l\Big) &= \sigma_i^{k\ominus 1} + \rho_i \cdot \sum_{l=1}^{m} \Delta_l\\
&= \alpha_i^{k\ominus 1}(\Delta_1) + \sum_{l=2}^{m} \overline{\alpha_i}(\Delta_l)
\end{aligned} \quad (6)$$

Hence, Eq. (5) can be rewritten using Eq. (6) as follows:

$$\begin{aligned}
&D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1})\\
\geq &\sum_{l \in \mathbb{P}_f(n)} [\beta^l(\Delta_l)\\
- &\sum_{i \ni l, i \neq f} \alpha_i^{l\ominus 1}(\Delta_l) \cdot 1_{\{l=Mft(i,f,n)\}} + \overline{\alpha_i}(\Delta_l) \cdot 1_{\{l \neq Mft(i,f,n)\}}]\\
\geq &\sum_{l \in \mathbb{P}_f(n)} [(R^l - \sum_{i \ni l, i \neq f} \rho_i) \cdot\\
&(\Delta_l - T^l - \frac{\sum_{i \ni l, i \neq f} \sigma_i^{Mft(i,f,n)\ominus 1} + T^l \cdot \sum_{i \ni l, i \neq f} \rho_i}{R^l - \sum_{i \ni l, i \neq f} \rho_i})]^+\\
\geq &\min_{l \in \mathbb{P}_f(n)} (R^l - \sum_{i \ni l, i \neq f} \rho_i) \cdot [\sum_{l \in \mathbb{P}_f(n)} \Delta_l - \sum_{l \in \mathbb{P}_f(n)} T^l\\
- &\sum_{l \in \mathbb{P}_f(n)} \frac{\sum_{i \ni l, i \neq f} \sigma_i^{Mft(i,f,n)\ominus 1} + T^l \cdot \sum_{i \ni l, i \neq f} \rho_i}{R^l - \sum_{i \ni l, i \neq f} \rho_i})]^+
\end{aligned} \quad (7)$$

Knowing the definition of $\overline{\mathbb{K}_f(n)}$, we can easily verify that

$$\sum_{l\in\mathbb{P}_f(n)} T^l \cdot \sum_{i\ni l, i\neq f} \rho_i \Leftrightarrow \sum_{i\in\overline{\mathbb{K}_f(n)}} \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j$$

Hence, Eq. (7) becomes:

$$\begin{aligned}
& D_f^{f.ft\ominus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \quad (8)\\
\geq\ & \min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)\cdot[t_{f.ft\oplus(n-1)} - t_{f.ft\ominus 1} - \sum_{l\in\mathbb{P}_f(n)} T^l\\
-\ & \sum_{i\in\overline{\mathbb{K}_f(n)}} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)}]^+\\
\geq\ & \min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)\cdot[t_{f.ft\oplus(n-1)} - t_{f.ft\oplus 1} - \sum_{k\in\mathbb{P}_f(n)} T^k\\
-\ & \sum_{i\in\overline{\mathbb{K}_f(n)}, f\ni i.ft} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)}\\
-\ & \sum_{\substack{i\in\overline{\mathbb{K}_f(n)}\\ i\ni f.ft\\ i.ft\neq f.ft}} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)}]^+
\end{aligned}$$

Moreover, for each interfering flow i in category 3 split as (i_1, i_2), with i_1 and i_2 in categories 1 and 2, we have:

$$\begin{aligned}
& \sigma_{i1}^{Mft(i_1,f,n)\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_{i1}} T^j \quad (9)\\
+\ & \sigma_{i2}^{Mft(i_2,f,n)\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_{i2}} T^j\\
=\ & \sigma_i^{i.ft\ominus 1} + \sigma_i^{f.ft\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap(\mathbb{P}_{i1}\cup\mathbb{P}_{i2})} T^j\\
=\ & \sigma_i^0 + \sigma_i^{f.ft\ominus 1} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j
\end{aligned}$$

(10)

Using Eq. (9) and (8), we deduce:

$$\begin{aligned}
R^{\mathbb{P}_f(n)} &= \min_{l\in\mathbb{P}_f(n)}(R^l - \sum_{i\ni l,i\neq f}\rho_i)\\
T^{\mathbb{P}_f(n)} &= \sum_{k\in\mathbb{P}_f(n)} T^k\\
&+ \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^0 . 1_{\{f\ni i.ft\}} + \rho_i \cdot \sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{R^{\mathbb{P}_f(n)}}\\
&+ \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^{f.ft\ominus 1}\cdot 1_{\{i.ft\neq f.ft/i\ni f.ft\}}}{R^{\mathbb{P}_f(n)}} \quad (11)
\end{aligned}$$

This finishes the proof of the theorem.

□

TABLE I: Notations

M	Number of nodes in the network
I	Set of flows served within the network
$l\oplus k$	$k-eth$ node downstream from node l
$l\ominus k$	$k-eth$ node upstream from node l
$i\ni k$	Set of flows crossing the node k
$\mathbb{P}_i$	Path of flow i from its source until its final sink
h_i	Number of hops within $\mathbb{P}_i$
$\mathbb{P}_i(n)$	Subpath of flow i from its source through n hops, $n\leq h_i$
$\mathbb{K}_f(n)$	Set of interfering flows with flow f along $\mathbb{P}_f(n)$
$\mathbb{K}_{\leq f}(n)$	Set of flows interfering with the flow f along $\mathbb{P}_f(n)$ with a priority higher or equal to f one.
$\overline{\mathbb{K}_f(n)}$	Transformed $\mathbb{K}_f(n)$ when cutting virtually the cycles
$Mft(i,f,n)$	First convergence point label of flows i and f along $\mathbb{P}_f(n)$
$Mlt(i,f,n)$	Last convergence point label of flows i and f along $\mathbb{P}_f(n)$
$\beta^k(t)$	Service curve guaranteed within node k
$\alpha_i^0(t)$	Arrival curve of flow i at its initial source
$\alpha_i^{k\ominus 1}(t)$	Input arrival curve of flow i at the output port of node k
R^k	Service rate of node k
T^k	Service latency of node k
$\sigma_i^{k\ominus 1}$	Input Maximum burst at the output port of node k
ρ_i	Maximum rate of flow i
$EED_f^{\mathbb{P}_f(n)}$	The end-to-end delay bound of flow f along $\mathbb{P}_f(n)$

REFERENCES

[1] Ahmed Amari, Ahlem Mifdaoui, Fabrice Frances, Jérôme Lacan, David Rambaud, and Loic Urbain. AeroRing: Avionics Full Duplex Ethernet Ring with High Availability and QoS Management. In *European Congress on Embedded Real Time Software and systems 2016*, pages pp–150, 2016.

[2] Anne Bouillard, Laurent Jouhet, and Eric Thierry. Service curves in Network Calculus: Dos and Don'ts. Technical report, 2009.

[3] Anna Charny and Jean-Yves Le Boudec. Delay Dounds in a Network with Aggregate Scheduling. In *Quality of Future Internet Services*. Springer, 2000.

[4] AEE Committee et al. Aircraft Data Network Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664. *Annapolis, Maryland: Aeronautical Radio*, 2002.

[5] Rene L Cruz. A Calculus of Delay Part II: Network Analysis. *IEEE Trans. Inform. Theory*, 1991.

[6] Dirk Jansen and Holger Buttner. Real-time Ethernet: the EtherCAT solution. *Computing and Control Engineering*, 15(1):16–21, 2004.

[7] Bengt Jonsson, Simon Perathoner, Lothar Thiele, and Wang Yi. Cyclic Dependencies in Modular Performance Analysis. In *Proceedings of the 8th ACM international conference on Embedded software*, 2008.

[8] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer Science & Business Media, 2001.

[9] Simon Perathoner, Ernesto Wandeler, and et al. Influence of Different Abstractions on the Performance Analysis of Distributed Hard Real-Time Systems. *Design Automation for Embedded Systems*, 2009.

[10] Raimond Pigan and Mark Metter. *Automating with PROFINET: Industrial Communication Based on Industrial Ethernet*. Wiley-VCH, 2008.

[11] Jens B Schmitt, Frank A Zdarsky, and Ivan Martinovic. Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 14th GI/ITG Conference-*, 2008.

[12] Stephan Sommer, Alexander Camek, Klaus Becker, Christian Buckl, Andreas Zirkler, Ludger Fiege, Michael Armbruster, Gernot Spiegelberg, and Alois Knoll. Race: A Centralized Platform Computer Based Architecture for Automotive Applications. In *Electric Vehicle Conference (IEVC), IEEE International*, 2013.

[13] Leandros Tassiulas and Leonidas Georgiadis. Any Work-Conserving Policy Stabilizes the Ring with Spatial Re-use. *IEEE/ACM Transactions on Networking (TON)*, 4(2):205–208, 1996.