Exact Worst-case Delay for FIFO-multiplexing Tandems

Anne Bouillard*
Department of Informatics
ENS/INRIA, France
45 Rue d'Ulm, 75230 Paris CEDEX 05
Anne.Bouillard@ens.fr

Giovanni Stea†
Department of Information Engineering
University of Pisa, Italy
Largo L. Lazzarino 2, 56122, Pisa
g.stea@iet.unipi.it

***Abstract*—This paper computes the actual worst-case end-to-end delay for a flow in a tandem of FIFO multiplexing service curve nodes, where flows are shaped by concave, piecewise linear arrival curves, and service curves are convex and piecewise linear. Previous works only computed bounds on the above quantity, which are not always tight. We show that the solution entails taking the maximum among the optimal solution of a number of Linear Programming problems. However, the number and size of LP problems grows exponentially with the tandem length. Furthermore, we present approximate solution schemes to find both upper and lower delay bounds on the worst-case delay. Both of them only require to solve just one LP problem, and they produce bounds which are generally more accurate than those found in the previous work. Finally, we elaborate on how the worst-case scenario should be constructed.**

I. Introduction

Network calculus (NC, [8], [9], [11]) is a theory for deterministic network performance analysis, which has found many applications, from Quality of Service problems in IP networks, to wireless sensor networks [10], [15], Ethernet installations [17], and Systems-on-Chip [7]. NC allows one to compute delay bounds over a multi-hop path by composing service guarantees at single nodes. In several network architectures, nodes (e.g., switches or routers) buffer traffic of different flows in the same physical queue. For instance, Behavior Aggregates from Differentiated Services networks (DiffServ [4]), or traffic trunks in Multi-Protocol Label Switching (MPLS, [14]), are composed of possibly many flows sharing the same queue at each node. We call this paradigm *flow aggregation* or *multiplexing*. In many cases, flows are buffered First-Come-First-Served, or FIFO, in the same queue. Network Calculus has already been employed to compute upper and lower bounds on the worst-case delay (WCD) of a flow traversing a tandem of FIFO-multiplexing nodes.

As far as upper bounds are concerned, the method known as Least Upper Delay Bound (LUDB) has been proposed in [3]. It is based on equivalent service curves, and it consists in formulating and solving a piecewise-linear programming (P-LP) problem. A tool, called DEBORAH (DElay BOund Rating AlgoritHm), has been proposed to solve the problem. It transforms the P-LP problem into a number of linear programming (LP) problems, each one of which produces an upper bound on the WCD, solves all the LPs and takes the minimum solution (which is, in fact, the least upper bound). As far as lower bounds on the WCD are concerned, a trajectorial method is described in [2], [3] and implemented in DEBORAH. It consists in setting arrival functions at each node empirically, and having nodes delay traffic as much as possible. The LUDB method has two main shortcomings: it can be inaccurate, and it is costly from a computation standpoint, despite the optimizations. As far as accuracy is concerned, although the LUDB may be tight (by *tight* we mean equal to the WCD) in some specific cases, e.g., sink-tree networks ([12]) it has been proved in [1] that this is not always the case, even in simple tandems. Furthermore, the lower and upper bounds may diverge in some scenarios, which hints at either or both of them being inaccurate. As for complexity, the methods for both upper and lower bounds have exponential complexity. Although their implementation is carefully tuned to keep the number of operations to a minimum, and several heuristics are presented, it may still take minutes to analyze a tandem of more than 15 nodes. Hence these methods cannot be applied in online contexts, e.g. for admission control.

Other works in the recent past (e.g. [5], [16]) have focused on computing delay bounds for flows in feed-forward networks of blind multiplexing nodes. *Blind* means that no assumption is made regarding the flow multiplexing criterion: for instance, both a FIFO multiplexing scheme and a strict priority multiplexing scheme in which the tagged flow (i.e., the one being analyzed) is always multiplexed at the lowest priority fit this definition. For blind-multiplexing networks the exact WCD has been computed using linear programming techniques in [5]. While the problem is NP-hard for general feed-forward networks, it can be solved with a single linear program of polynomial size for tandems. Instead of relying on equivalent service curves, a set of input-output inequalities at each node is formulated, which allow us to relate the ingress and egress dates of a single bit. Then the optimization problem that maximizes the distance between the egress and ingress dates (i.e., the delay) for the flow of interest is solved. Even though the bounds computed therein are tight

*This author has carried out the work presented in this paper at LINCS (www.lincs.fr). This work has been partially funded by the French National Research Agency ANR (PEGASE SEGI 009 02).

†This work has been partially funded by the Italian Ministry for University and Research under grant PRIN 2009XN4ZRR.

VALUETOOLS 2012, October 09-12, Cargèse, France

DOI 10.4108/valuetools.2012.250090

for a blind multiplexing scenario, they are not so under the FIFO hypothesis, and the gap between the WCD and these bounds can be shown to be unbounded as the node utilization approaches 100%.

In this paper, we present a linear programming approach for *exact* WCD computation under the FIFO hypothesis. This is the first paper to solve the above problem. Unlike in [5], we show that enforcing the monotonicity of cumulative arrival and departure functions requires a total ordering of the dates, whereas the input-output relationships only give a partial ordering. Therefore, we need to formulate and solve as many LPs as the number of total orderings compatible with the input-output relationships. Although LPs are polynomial, their number and size grow exponentially with the tandem length, which hampers scalability. As a second contribution, we show that our formulation readily allows for *approximated* upper and lower bounds to be computed much faster. In fact, two LP problems can be derived by removing and adding constraints, respectively, whose optima bound the WCD from both sides. Another tool exists (i.e., DEBORAH) that solves the same problem (in less general settings). However, we show that our bounds are more accurate than DEBORAH's. As a third contribution, we review the conjecture of a worst-case scenario for a FIFO tandem (first appeared in [1]): we report some (unfortunately non conclusive) proofs of properties that the worst-case scenario should satisfy, and show that, within the limits of the analyzed case studies, the conjecture is not disproved.

The rest of the paper is organized as follows: Section II reports some background and notation on Network Calculus. In Section III we introduce the hypotheses and state the problem formally. We describe our contribution in Section IV. In Section V we prove properties related to worst-case scenarios. We report evaluation results in Section VI. Finally, conclusions are reported in Section VII.

II. Network Calculus background

This section introduces basic Network Calculus concepts, using the same notation as in [11]. In Network Calculus, a data flow is described by means of a wide-sense increasing and left-continuous cumulative function $R : \mathbb{R}_+ \to \mathbb{R}_+$, where $R(t)$ is the number of bits seen on the flow in time interval $[0, t[$.

A wide-sense increasing function α is an *arrival curve* for a flow characterized by a cumulative function A (or A is α-*upper constrained*) if:

$$\text{for all } \tau \leq t, \quad A(t) - A(\tau) \leq \alpha(t - \tau).$$

As an example, a common *leaky-bucket shaper*, with sustainable *rate* ρ and *burst size* σ, enforces the (concave) affine arrival curve $\gamma_{\sigma,\rho}(t) = \sigma + \rho t$. Let A and D be the *Cumulative Arrival* and *Cumulative Departure* functions (CAF and CDF) characterizing the same data flow at the input and output of a network element, respectively. Then, the network element can be modeled by the *service curve* β if:

$$\forall t \geq 0, \quad D(t) \geq \inf_{0 \leq s \leq t} A(s) + \beta(t - s). \qquad (1)$$

The flow is said to be guaranteed the (minimum) service curve β. The infimum on the right side of (1), as a function of t, is called the min-plus convolution of A and β, and is denoted by $A \otimes \beta$. Min-plus convolution is commutative and associative. Several network elements, such as delay elements, links, and regulators, can be modeled by service curves. Many packet schedulers can be modeled through *rate-latency* service curves, defined as follows:

$$\beta_{\theta,R}(t) = R(t - \theta)_+$$

for some $\theta > 0$ (the latency) and $R > 0$ (the rate). Notation $(.)_+$ denotes $\max(., 0)$ and should not be confused with $f(t^+)$ which instead denotes the right limit of f at t in the following. Note that rate-latency curves are convex. A fundamental result of Network Calculus is that the service curve of a tandem of network elements traversed by the same data flow is obtained by convolving the service curves of each network element. A well-known property of convolution is given by the following lemma.

Lemma 1. *If f and g are left-continuous and wide-sense increasing, then for all $t \geq 0$, there exists $s \leq t$ such that $f \otimes g(t) = f(s) + g(t - s)$.*

In a network element that serves bits in FIFO order, one can determine the delay of each bit by comparing the CAF and the CDF. More specifically, the delay of a bit leaving at time t is equal to:

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

In other words, it is the horizontal distance between the CAF and the CDF measured at point t. If $A(t)$ and $D(t)$ are continuous, then it is $D(t + d(t)) = A(t)$, and $d(t)$ is the smallest value that satisfies this equation. The same holds for a tandem of N nodes traversed by a flow: the end-to-end delay of the bit arriving at time t is the horizontal distance from point $\{t, A(t)\}$ on the CAF at node 1 to the CDF at node N. An *upper bound* on the delay for a flow can be computed by combining its arrival curve α and the service curve β of the (tandem of) node(s) it traverses. The delay bound is

$$h(\alpha, \beta) = \sup_{t \geq 0} [\inf\{d \geq 0 \mid \alpha(t - d) \leq \beta(t)\}]. \qquad (2)$$

a) FIFO Multiplexing: Under FIFO multiplexing, traffic of flows arriving at a node are buffered First-Come-First-Served in a single queue. This means that a bit of a tagged flow arriving at time t will be transmitted only when all the traffic arrived before time t (and belonging to any flow traversing that node) has been transmitted. Network Calculus allows one to derive *equivalent service curves* [11, Chapter 6.2] for the individual flows, which can then be employed to compute delay bounds through (2). In fact, this lies at the core of the LUDB method [3].

However, the fact that the node is FIFO (henceforth referred to as the *FIFO hypothesis*) allows one to compute the CDFs of *single flows*, given their CAFs, *without* resorting to equivalent service curves. All it takes is the *aggregate* CDF, or at least - if only a service curve is known - a lower bound on that CDF, as per (1). The operations required for computing the CDF of a tagged flow at a node are:

- *FIFO multiplexing* of several CAFs at the entrance of a node, so as to compute the aggregate CAF;
- *Input-output transformation* from the aggregate CAF to the aggregate CDF, according to a node's service curve (i.e., to inequality (1));
- *FIFO de-multiplexing* of flows at the exit of a node, i.e. computation of per-flow CDFs from the aggregate CDF, exploiting the FIFO hypothesis.

The procedure is exemplified in Figure 1, using two piecewise-linear CAFs A_1 and A_2 and a rate-latency service curve β (this can obviously be generalized to any number and shape of CAFs and any service curve). FIFO multiplexing (bottom left) is a summation of CAFs: $A = A_1 + A_2$. I-O transformation (top) corresponds to computing an aggregate CDF which is wide-sense increasing and satisfies (1), e.g., the lower-bound CDF obtained by assuming that equality holds in (1). FIFO de-multiplexing (bottom right) exploits the FIFO hypothesis: more specifically, for all $t \in \mathbb{R}_+$, there is a unique $\tau \le t$ such that $A(\tau) \le D(t) \le A(\tau^+)$. Then, $D_1(t)$ and $D_2(t)$ must satisfy $A_i(\tau) \le D_i(t) \le A_1 i(\tau^+)$, $i \in \{1, 2\}$ (and $D_1(t)+D_2(t) = D(t)$). However, $D_1(t)$ and $D_2(t)$ may not be uniquely defined, when neither A_1 nor A_2 are continuous in τ. In that case, any wide-sense increasing D_1 and D_2 satisfying the above equalities are possible CDFs.

If A_2 is discontinuous in τ (e.g., $A_2(\tau^+) = A_2(\tau)+\sigma$), but A_1 is not, then on some non trivial interval D_1 is constant, while D_2 has the same slope as D. If in the interval $[t_1, t_2]$, D is affine with slope R, and on the *corresponding* interval $[\tau_1, \tau_2]$ (i.e., the interval when the bits that depart in $[t_1, t_2]$ arrive at the input), A_i is affine with slope ρ_i, $i \in \{1, 2\}$, then D_i is affine on the interval $[t_1, t_2]$ with slope $\frac{\rho_i}{\rho_1+\rho_2}R$.

III. System model

We analyze a tandem of N *nodes*, numbered from 1 to N, connected by forward links from node i to $i + 1$, $1 \le i < N$. The tandem is traversed by *flows*, i.e. distinguishable streams of traffic. Flow (i, j), with $1 \le i \le j \le N$, is the flow that enters the network at node i, traverses all nodes from i to j included, and then departs. We assume that, for each couple (i, j), there is at most one flow. This is not restrictive, since piecewise affine and concave arrival curves are additive. At each node, *FIFO multiplexing* is in place, meaning that all flows traversing the node are buffered in a single queue First-Come-First-Served.

We make the following assumptions and will use the following notations in the rest of the paper:

- $F^{(h)}_{(i,j)}$ is the CAF of flow (i, j) at node $h \in [i, j]$;
- $F^{(j+1)}_{(i,j)}$ represents the CDF of flow (i, j) at node j;

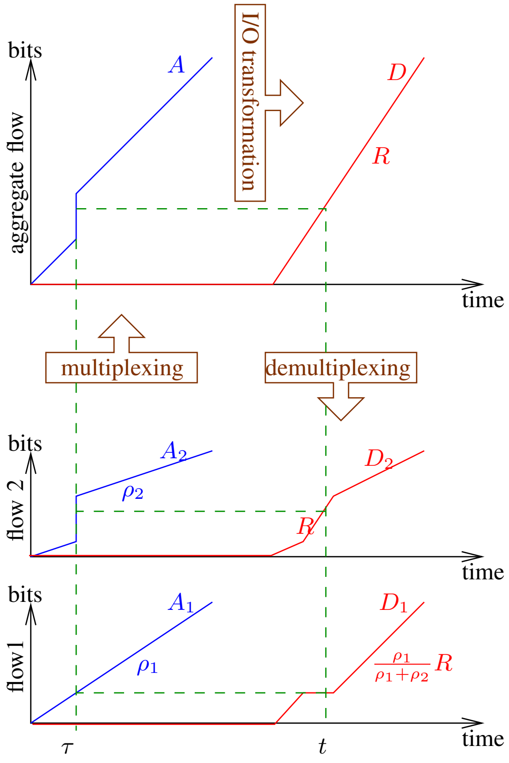

Fig. 1. Input-output relationship at a FIFO node.

- the aggregate CAF at node h is $A^{(h)} = \sum_{(i,j)\ni h} F^{(h)}_{(i,j)}$ and the aggregate CDF is $D^{(h)} = \sum_{(i,j)\ni h} F^{(h+1)}_{(i,j)}$, where $(i, j) \ni h$ means that $h \in [i, j]$;
- node h offers a service curve β_h to $A^{(h)}$, and β_h is assumed to be wide-sense increasing, piecewise affine and convex;
- the arrival process of flow (i, j), $F^{(i)}_{(i,j)}$, is $\alpha_{(i,j)}$-upper constrained, where $\alpha_{(i,j)}$ is assumed to be wide-sense increasing, piecewise affine and concave.

A system is said to be stable if there exists a constant C such that for each server, the backlog is always upper bounded by C. Let $R_h = \lim_{t\to\infty} \beta_h(t)/t$ and $\rho_{(i,j)} = \lim_{t\to\infty} \alpha_{(i,j)}(t)/t$. We assume that the system is stable, that is, $\forall h \in [1, N]$, $R_h \ge \sum_{(i,j)\ni h} \rho_{(i,j)}$ (see [11] for example).

A *scenario* for an N-node tandem described as above is a family of functions $(F^{(h)}_{(i,j)})_{1\le i\le j\le N, i\le h\le j+1}$ such that:

1) $\forall i, j, h$, $F^{(h)}_{(i,j)}$ is wide-sense increasing and left-continuous;
2) $\forall i \le h \le j$, $F^{(h)}_{(i,j)} \ge F^{(h+1)}_{(i,j)}$;
3) $\forall i \le j$, $F^{(i)}_{(i,j)}$ is $\alpha_{(i,j)}$-upper constrained;
4) $\forall h \in [1, N]$, $D^{(h)} \ge A^{(h)} \otimes \beta_h$;
5) Nodes satisfy the FIFO hypothesis.

We are interested in finding the WCD for the *tagged flow* $(1, N)$, i.e. the maximum delay that one of its bit can experience.

IV. Linear programming approach

In this section, we show that worst-case scenarios are indeed linear programs. Unfortunately, the number and size of linear programs that are required to compute the WCD are exponential in the number of flows and nodes, which may not be tractable at large scales. We also provide algorithms to compute *upper* and *lower* bounds on the WCD. We first exemplify the method on a single-node scenario for the sake of readability, and then generalize it to a tandem of arbitrary length. Furthermore, we provide a comprehensive example at the end of the section.

b) Single node: Let us first focus on the simple yet meaningful example of one node traversed by two flows, whose CAFs/CDFs are denoted as A_1, A_2 and D_1, D_2. For each departure date t_1 of a bit of D, there exists another date t_2 when that bit arrived, and a date t_3 which verifies the convolution property stated in Lemma 1 at date t_1. As $\beta \geq 0$, $t_3 \leq t_2$. Then, the WCD is computed by solving the following problem, also illustrated in Figure 2 ($i \in \{1, 2\}$):

Maximize $t_1 - t_2$ under constraints

$D_1(t_1) + D_2(t_1) \geq$ $(A_1(t_3) + A_2(t_3)) + \beta(t_1 - t_3)$	service curve
$\forall i, D_i(t_1) = A_i(t_2)$	FIFO hypothesis
$\forall i, A_i(t_2) - A_i(t_3) \leq \alpha_i(t_2 - t_3)$	arrival curve
$\forall i, A_i(t_3) \leq A_i(t_2)$	non-decreasing CAF
$t_3 \leq t_2 \leq t_1$	date ordering

Assume that, in the above constraints, $A_i(t_x)$, with $x \in \{1, 2\}$, and $D_i(t_1)$, are *variables* (instead of functions computed at some date). Then all the above constraints are linear. In fact, if α_1 and α_2 are piecewise affine concave, then the arrival curve constraints can be linearized as follows:

$$A_i(t_2) - A_i(t_3) \leq \alpha_i(t_2 - t_3) = [\min_m Y_m](t_2 - t_3)$$
$$\Leftrightarrow \forall m,\ A_i(t_2) - A_i(t_3) \leq Y_m(t_2 - t_3),$$

where Y_m are affine curves. Thus, each arrival curve constraint yields as many linear constraints as its linear pieces. The same holds, *mutatis mutandis*, for service curve constraints, given that β is piecewise affine and convex. The objective function is clearly linear as well, hence computing the WCD entails solving an LP problem.

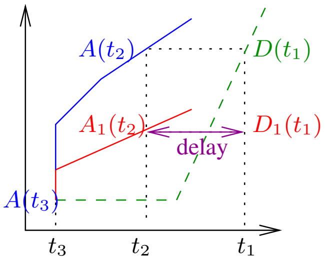

Fig. 2. Constraints for a single server.

c) Tandem networks: Consider now a tandem of N nodes and a tagged flow $(1, N)$, and focus on node N. Most of the above constraints can be written down, generalized to the set of all flows traversing N. However, arrival constraints for flows (i, N), with $i < N$, cannot be formulated, since these arrival curves have effect at node i. The tagged flow $(1, N)$ is among the latter. Therefore, given a departure date at node N, t_1, we can compute two input dates, related to the FIFO and service curve constraints at node N, t_2 and t_3. Those dates also describe the *output* of node $N-1$. Thus, we can iterate the previous reasoning at node $N-1$, for each of the two dates: the bit that exits node $N-1$ at time t_2 entered that node at time t_4, and t_5 is a time instant in the past that verifies the service curve constraint formulated at time t_2. Similarly, for date t_3 at node $N-1$ we can identify dates t_6 and t_7, respectively. It is easy to see that we can go backwards until node 1, doubling the number of dates and constraints at each node, and adding arrival constraints whenever we hit the ingress node of a flow. This way, we eventually set up a set of constraints that relate the dates of departure (at node N) and arrival (at node 1) of a bit of the tagged flow. Thus, we compute its WCD by solving the LP problem that maximizes the above difference under these constraints.

More formally, the variables of our problem are the following:

- *dates:* $t_1, \ldots, t_{2^{N+1}-1}$, where t_{2k} and t_{2k+1} correspond to the FIFO hypothesis and the service curve constraints with regards to t_k, respectively;
- *function values:* $F^{(h)}_{(i,j)}(t_k)$ for $h \in [i, j+1]$ and $k \in [2^{N+1-h}, 2^{N+2-h} - 1]$.

The number of dates (hence of variables) grows exponentially with the tandem length, since it doubles at each node as we go backwards. Unfortunately, in a multi-node scenario, these dates are only partially ordered. For $k < 2^N$, we have (*) $t_{2k+1} \leq t_{2k} \leq t_k$. Moreover, if $2^h \leq k, k' < 2^{h+1}$ and (**) if $t_k \leq t_{k'}$, then $t_{2k} \leq t_{2k'}$ (cumulative functions are non-decreasing) and $t_{2k+1} \leq t_{2k'+1}$ (same as above, plus β_j is convex). These relations and the transitivity only lead to a partial order of $t_{2^h}, \ldots, t_{2^{h+1}-1}$. For example, for a two-node tandem, we have $t_1 \geq t_2 \geq t_3$, and $t_4 \geq t_5$, $t_4 \geq t_6$, $t_5 \geq t_7$ and $t_6 \geq t_7$. However, t_5 and t_6 cannot be ordered. A partial order creates a problem. Given a partial order of dates, we can enforce the same partial order on the corresponding function values, but this is not enough to ensure the monotonicity of the CAFs and CDFs. In fact, it may well happen that, in the above example, the optimal solution to the LP problem is for $t_5 < t_6$ and $F^{(h)}_{(i,j)}(t_5) > F^{(h)}_{(i,j)}(t_6)$, for some i, j, h, which violates monotonicity. In this case, the solution does not correspond to any feasible scenario, hence it is not the WCD. To ensure monotonicity in a LP approach, one needs to generate *total* orders on the dates.

Since a date corresponds to a unique node in the tandem, for a date t_k there exists a unique h such that $F^{(h)}_{(i,j)}(t_k)$ is defined, hence it is sufficient to generate total orders on $t_{2^h}, \ldots, t_{2^{h+1}-1}$ for each $h \in [1, N+1]$. Let *Ord* the set of such orders that are compatible with (*) and (**). For each order $\preceq \in Ord$, we now can set the following constraints for the LP $\lambda_\preceq$: forall $h \in [1, \dots, N+1]$, $k, k' \in [2^{N+1-h}, 2^{N+1-h}-1]$, (i,j) such that $h \in [i, j+1]$,

- $t_k \preceq t_{k'} \Rightarrow \{t_k \leq t_{k'},\ F^{(h)}_{(i,j)}(t_k) \leq F^{(h)}_{(i,j)}(t_{k'})\} \in \lambda_\preceq$ (date order, monotonicity);
- $i < h \Rightarrow \{F^{(h)}_{(i,j)}(t_k) = F^{(h-1)}_{(i,j)}(t_{2k})\} \in \lambda_\preceq$ (FIFO hypothesis);
- $\{D^{(h)}(t_k) \geq A^{(h-1)}(t_{2k+1}) + \beta_h(t_k - t_{2k-1})\} \in \lambda_\preceq$ (service curve);
- $i = h$ and $t_k \preceq t_{k'} \Rightarrow \{F^{(h)}_{(h,j)}(t'_k) - F^{(h)}_{(h,j)}(t_k) \leq \alpha_{(i,j)}(t_{k'} - t_k)\} \in \lambda_\preceq$ (arrival curve).

The objective of the LP is then $\max t_1 - t_{2^N}$. Let $\Lambda = \{\lambda_\preceq \mid \preceq \in Ord\}$ and call d_λ the maximal solution of $\lambda \in \Lambda$. We now state the main result:

Theorem 1. *The WCD for flow $(1,N)$ is $W = \max_{\lambda\in\Lambda} d_\lambda$.*

We give the proof in two separate lemmas. The first one (Lemma 2) states that W bounds from the above the delay obtainable from any scenario. The second one (Lemma 3) shows that there is a scenario where W is attained.

Lemma 2. *Let F be a scenario of the network with delay d. There exists a linear program $\lambda \in \Lambda$ such that $d \leq d_\lambda$.*

Proof: Let $F = (F^{(h)}_{(i,j)})$ be a scenario and t_1 be the date of departure of the bit of interest (the one that experiences the largest delay d). By definition, cumulative functions are left-continuous. Then, by Lemma 1 we can find a date t_3 such that $D^{(h)}(t_1) \geq A^{(h)}(t_3) + \beta_h(t_1 - t_3)$ and a date t_2 such that $\forall i$, $F^{(N)}_{(i,N)}(t_2) \leq F^{(N+1)}_{(i,N)}(t_1) \leq F^{(N)}_{(i,N)}(t_2^+)$. For an LP, we define the variables and constraints just as above, but with $F^{(N)}_{(i,N)}(t_2) = F^{(N+1)}_{(i,N)}(t_1)$.

By induction, we can go on defining dates and variables $F^{(h)}_{(i,j)}(t_k)$: if variables t_k and $F^{(h)}_{(i,j)}(t_k)$ are defined, one can define t_{2k}, t_{2k+1}, $F^{(h-1)}_{(i,j)}(t_{2k})$ and $F^{(h-1)}_{(i,j)}(t_{2k+1})$ if $h-1 \geq i$ the following way: t_{2k+1} is such that $D^{(h-1)}(t_k) \geq A^{(h-1)}(t_{2k+1}) + \beta_{h-1}(t_k - t_{2k+1})$ and t_{2k} as $F^{(h-1)}_{(i,j)}(t_{2k}) \leq F^{(j+1)}_{(i,j)}(t_{k2^{h-j-1}}) \leq F^{(h-1)}_{(i,j)}(t^+_{2k})$. Variables and constraints are defined accordingly, except for t_{2k}: we set $F^{(h-1)}_{(i,j)}(t_{2k}) = F^{(h)}_{(i,j)}(t_k)$.

The reason we cannot define t_{2k} directly by $F^{(h-1)}_{(i,j)}(t_{2k}) = F^{(h)}_{(i,j)}(t_k)$ is that the functions are not necessarily continuous. As a consequence, it may be the case that several dates defined have the same value, with different values for the functions (see Figure 3). This is not a problem for linear programs. Indeed, when all the constraints above are defined, one can set a total order on the dates for each server h: If $t_k < t_{k'}$, then we set the constraint $t_k \leq t_{k'}$. If $t_k = t_{k'}$ and there exists (i,j) such that $F^{(h)}_{(i,j)}(t_k) < F^{(h)}_{(i,j)}(t_{k'})$, then set $t_k \leq t_{k'}$. Because we started from a feasible scenario, one cannot have $F^{(h)}_{(i,j)}(t_k) < F^{(h)}_{(i,j)}(t_{k'})$ and $F^{(h)}_{(i',j')}(t_k) > F^{(h)}_{(i',j')}(t_{k'})$.

Once this order is defined, one can define constraints to ensure that the functions are non-decreasing and that the arrival processes $F^{(i)}_{(i,j)}$ are $\alpha_{(i,j)}$-upper constrained.

The constraints defined correspond to a linear program $\lambda \in \Lambda$, and the original scenario satisfies all the constraints. So, $d \leq d_\lambda$. ■

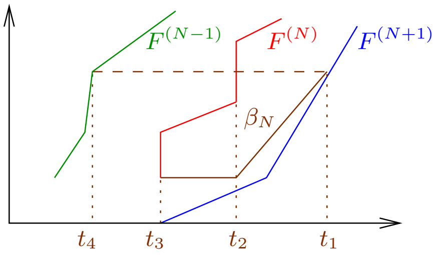

Fig. 3. Construction of the dates for the LP (with only one flow for simplicity). We have the constraint $F^{(N+1-h)}(t_{2^h}) = F^{(N+1)}(t_1)$, $h \in \{a, 2\}$.

Lemma 3. *Let $\lambda \in \Lambda$. For any solution of λ, there exists a scenario of the system whose delay is no smaller than this solution.*

Proof: Let $\lambda \in \Lambda$ and a solution of this linear program. We will define a scenario that verifies all the constraints. Fix i, j and h and consider the variables of the form $F^{(h)}_{(i,j)}(t_k)$ and t_k for which $F^{(h)}_{(i,j)}(t_k)$ is defined. Let $T_k = \{t_\ell \mid t_\ell = t_k\}$. We set $F^{(h)}_{(i,j)}(t_k) = \min_{t\in T_k} F^{(h)}_{(i,j)}(t)$ and $F^{(h)}_{(i,j)}(t_k^+) = \max_{t\in T_k} F^{(h)}_{(i,j)}(t)$. If $h = i$, from those values, we extrapolate $F^{(h)}$ as the largest function that is $\alpha_{(i,j)}$-upper constrained. It is proved in [6] that this is possible. If $h \neq i$, we extrapolate

$$F^{(h)}_{(i,j)}(t) = \min[F^{(h-1)}_{(i,j)}(t), \min_{t_k \geq t} F^{(h)}_{(i,j)}(t_k)].$$

One can check that:

- those functions are wide-sense increasing (because of the initial constraints);
- arrival functions are α-constrained (by construction);
- the FIFO order is preserved: it is preserved for the dates defined by the linear program and outside of those dates, the traffic is made of bursts only;
- $D^{(h)} \geq A^{(h)} \otimes \beta_h$ by construction.

We have then built a scenario that verifies all the constraints and that has the same delay as the solution of the linear program. Then, $d \geq d_\lambda$. ■

Note that Theorem 1 can be generalized to compute the WCD of any flow (i,j) by only modifying the objective function to Maximize $t_{2^{N-j}} - t_{2^{N-i+1}}$.

d) Bounds on the WCD: As already pointed out, computing the WCD requires solving as many LPs as elements in Λ. This number grows exponentially with the tandem length. Furthermore, the number of constraints grows exponentially as well with the tandem length. However, it is easy to find good bounds on the WCD at a reasonable computation cost, i.e., solving just *one* LP. An *upper bound* can be found by giving up total ordering. We only keep the partial date and function value ordering, and solve *one* LP. This problem lacks some constraints w.r.t. the WCD problem (notably, those related to monotonicity), hence yields an upper bound by definition, call it V_{LP}. This upper bound may verify all monotonicity constraints (something that can be easily checked *a posteriori*), in which case it is *the* WCD. Otherwise it is only an upper bound.

Similarly, any solution of any LP $\lambda \in \Lambda$ is a *lower bound* on the WCD. An algorithmically efficient way to obtain a lower bound is to reduce the number of dates, by imposing further constraints on the latter. We do this by setting $t_{2k+1} = t_{2k'+1}$ for $2^h \leq k, k' < 2^{h+1} - 1$. This way the number of different dates is quadratic in the number of nodes, and this induces a complete order on the dates for every h. Hence, the solution to the LP is a lower bound on the WCD, call it v_{LP}. This is the worst-case delay bound obtained when the servers offer *strict service curves* ([11]) and where for each server h, for the dates of interest, there is only one backlogged period starting at time $t_{2^{n-h+1}-1}$.

For example, for two nodes, the order of the dates is $t_1 \geq t_2 \geq t_3$, $t_4 \geq t_6 \geq t_5 = t_7$.

We now report a complete example, based on the tandem of Figure 4.

Example 1. *Consider* $\alpha_{(1,2)}(t) = \alpha_{(1,1)}(t) = 1 + t/3$, $\alpha_{(2,2)}(t) = \min(t, 11 + t/3)$, $\beta_1(t) = \beta_2(t) = (t-1)_+$. *The dates are* t_1, t_2, t_3, t_4, t_5, t_6 *and* t_7, *the function values are* $F^{(1)}_{(1,2)}(t_k)$ *and* $F^{(1)}_{(1,1)}(t_k)$ *for* $k \in \{4,5,6,7\}$, $F^{(2)}_{(1,2)}(t_k)$, $F^{(2)}_{(1,1)}(t_k)$ *and* $F^{(2)}_{(2,2)}(t_k)$ *for* $k \in \{2,3\}$ *and* $F^{(3)}_{(1,2)}(t_1)$ *and* $F^{(3)}_{(2,2)}(t_1)$. *Note that, with our notation* $A^{(2)}(t_k) = F^{(2)}_{(1,2)}(t_k) + F^{(2)}_{(2,2)}(t_k)$ *and* $D^{(1)}(t_k) = F^{(2)}_{(1,2)}(t_k) + F^{(2)}_{(1,1)}(t_k)$.

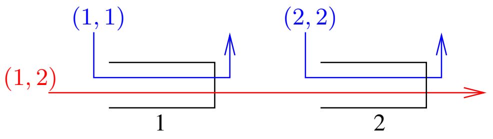

Fig. 4. Network of two nodes and three flows for Example 1.

The objective is to maximize $t_1 - t_4$.

The date constraints are:

- $t_1 \geq t_2 \geq t_3$; $t_2 \geq t_4 \geq t_5$; $t_3 \geq t_6 \geq t_7$; $t_4 \geq t_6$.

The service constraints are:

- $F^{(3)}_{(1,2)}(t_1) + F^{(3)}_{(2,2)}(t_1) \geq F^{(2)}_{(1,2)}(t_3) + F^{(2)}_{(2,2)}(t_3) + \beta_2(t_1 - t_3)$;
- $F^{(2)}_{(1,2)}(t_2) + F^{(2)}_{(1,1)}(t_2) \geq F^{(1)}_{(1,2)}(t_5) + F^{(1)}_{(2,2)}(t_5) + \beta_1(t_2 - t_5)$;
- $F^{(2)}_{(1,2)}(t_3) + F^{(2)}_{(1,1)}(t_3) \geq F^{(1)}_{(1,2)}(t_7) + F^{(1)}_{(2,2)}(t_7) + \beta_1(t_3 - t_7)$.

The FIFO hypotheses are:

- $F^{(3)}_{(1,2)}(t_1) = F^{(2)}_{(1,2)}(t_2)$; $F^{(3)}_{(2,2)}(t_1) = F^{(2)}_{(2,2)}(t_2)$;
- $F^{(2)}_{(1,2)}(t_2) = F^{(1)}_{(1,2)}(t_4)$; $F^{(2)}_{(1,1)}(t_2) = F^{(1)}_{(1,1)}(t_4)$;
- $F^{(2)}_{(1,2)}(t_3) = F^{(1)}_{(1,2)}(t_6)$; $F^{(2)}_{(1,1)}(t_3) = F^{(1)}_{(1,1)}(t_6)$;

We know that $t_3 \leq t_2$, *that* $t_7 \leq t_6 \leq t_4$ *and* $t_7 \leq t_5 \leq t_4$, *so we have the corresponding ordering on the function values (not reported for conciseness). Moreover, we have these arrival constraints:*

- $F^{(1)}_{(1,x)}(t_k) - F^{(1)}_{(1,x)}(t_\ell) \leq \alpha_{(1,x)}(t_k - t_\ell)$, $x \in \{1,2\}$, $(k,\ell) \in \{(4,5), (4,6), (4,7), (5,7), (6,7)\}$;
- $F^{(2)}_{(2,2)}(t_2) - F^{(2)}_{(2,2)}(t_3) \leq \alpha_{(2,2)}(t_2 - t_3)$.

The LP with these constraints yields an upper bound on the WCD. To obtain the exact WCD, one has to generate two LPs: one with $t_5 \leq t_6$ *and another with* $t_6 \leq t_5$. *For the program with* $t_5 \leq t_6$, *one has to add the date constraints and the corresponding function value constraints:*

- $F^{(1)}_{(1,x)}(t_5) \leq F^{(1)}_{(1,x)}(t_6)$, $x \in \{1,2\}$;
- $F^{(1)}_{(1,x)}(t_5) - F^{(1)}_{(1,x)}(t_6) \leq \alpha_{(1,x)}(t_5 - t_6)$, $x \in \{1,2\}$.

Similar constraints are required for $t_6 \leq t_5$. *The exact WCD is the maximum of the optimal solutions of the two LPs. For the lower bound, the linear program is obtained by imposing that* $t_5 = t_6$. *More in detail, this entails the following:*

- *suppress* t_5 *and* $F^{(1)}_{(1,x)}(t_5)$, $x \in \{1,2\}$ *and the constraints where this variable appears;*
- *add the service constraint* $F^{(2)}_{(1,2)}(t_2) + F^{(2)}_{(1,1)}(t_2) \geq F^{(1)}_{(1,2)}(t_7) + F^{(1)}_{(2,2)}(t_7) + \beta_2(t_2 - t_7)$;

Putting numbers into the scheme, we obtain:

- *the WCD is equal to* $max\{12.82, 8\}$, *i.e. 12.82.;*
- *the upper bound on the WCD is* $V_{LP} = 12.82$, *i.e. it is the WCD itself.*
- *the lower bound on the WCD is* $v_{LP} = 8$.

V. Properties of a worst-case scenario

Computing the WCD through linear programming can be complex, due to the explosion of the size and number of LPs. For this reason, it is important to explore properties of worst-case scenarios. In a long term, these might enable us to compute the WCD algorithmically, rather than via optimization, i.e. to find the CAFs and node behaviors that yield the WCD as a function of the topology and arrival/service curves. Moreover, knowing more about the worst-case scenario might allow us to simplify the computations (e.g., to reduce the number of dates in the LP). In this section, we review a conjecture on the worst-case scenario first appeared in [1] and refined in [3], valid for tandems with leaky-bucket shaped flows and rate-latency service curves. Then, we report two new properties that describe worst-case scenarios under more general hypotheses (those of our system model), which are compatible with that conjecture. Our properties are non-conclusive, meaning that they cannot identify *the* worst-case scenario. However, we believe that they are useful for future research.

Conjecture 1. *In a FIFO tandem of rate-latency nodes traversed by leaky-bucket shaped flows the WCD for the tagged flow takes place under the following hypotheses:*

(a) All nodes are lazy *(or* exact*), i.e. equality holds in input-output relationships at each node.*

(b) The tagged flow $(1, N)$ sends its whole burst $\sigma_{1,N}$ at time $t = 0$ and then stops. Therefore, the $\sigma_{(1,N)}$-th bit of the tagged flow experiences a larger delay than the others.

(c) Every cross flow (i, j) is saturated at the time when the $\sigma_{(i,j)}$-th bit of the tagged flow has arrived on node i. In other words, it sends "as much traffic as possible" to delay the $\sigma_{(1,N)}$-th bit of the tagged flow.

(d) At node 1, cross flows $(1, j)$ send their burst before *the tagged flow's burst (i.e., at time 0^-). In the overall CAF at node 1, bursts are sorted by j (i.e., the flow leaving the tandem first has its burst at the front of the queue) [3].*

Note that there are infinite CAFs that verify hyp. (c). For instance, one is the *greedy* CAF, i.e. the maximal CAF allowed by the arrival curve. Another one is the *delayed-greedy* CAF. A flow F is said to be delayed-greedy at t if $F(t) = \alpha(t)$ and $F(x) = (F(t) - \alpha(t - x))_+$ for $x < t$, i.e. if it has the minimum CAF that allows it to reach the same quota as the greedy flow at time t.

We now prove two properties that hold with general arrival and service curves. The first one (Lemma 4) shows that hypothesis (a) is not restrictive in *nested* tandems. A tandem is said to be nested if there are no two flows (i, j) and (h, k) such that $i < h \leq j < k$. A non nested tandem is shown instead in Figure 10. The second property (Lemma 5) constrains the CAF of the tagged flow, in a way which is compatible with hypothesis (b).

Lemma 4. *Assume that a nested tandem of N FIFO nodes is traversed by a set of flows, and fix the CAF of each flow (i, j) at its entry node, $F^{(i)}_{(i,j)}$. Then, the WCD for flow $(1, N)$ is achieved in a scenario where all nodes are lazy (or exact).*

Proof: We proceed by induction and show that $F^{(N+1)}_{(1,N)}$ is minimized when the servers are exact. Let $(F^{(h)}_{(i,j)})$ be a scenario for a nested tandem network and let $(\tilde{F}^{(h)}_{(i,j)})$ be the scenario where $\forall i, j,\ \tilde{F}^{(i)}_{(i,j)} = F^{(i)}_{(i,j)}$ and where nodes are exact. Notation $\tilde{.}$ is used for variables of this new scenario. Let x be a quantity of data arriving or departing from server h. We denote by $x^{(h)}_{(i,j)}$ the quantity of data among it belonging to flow (i, j). Note that, because of the FIFO assumption, this quantity is the same for the arrival and departure flow. We show by induction that $\sum_{(i,j):\ j\geq k} \tilde{x}^{(h)}_{(i,j)} \leq \sum_{(i,j):\ j\geq k} x^{(h)}_{(i,j)}$, i.e., the contribution of flows departing late in the network is reduced when the service is exact, and that the date of departure of the x-th bit from node h is delayed: $\tilde{t}^{(h)}_x \geq t^{(h)}_x$ (this is equivalent to $\tilde{D}^{(h)} \leq D^{(h)}$).

For $h = 1$, this result holds as the departure is delayed and the proportions of data are unchanged (arrivals are unchanged). Suppose that this is true for node $h - 1$. The departures at node $h - 1$ are delayed. Consider the arrivals at node h. The CAFs of flows (h, j), $j \geq h$ are unchanged, but as the departures at node $h - 1$ are delayed and $\sum_{(i,j):\ j\geq h} \tilde{x}^{(h)}_{(i,j)} \leq \sum_{(i,j):\ j\geq h} x^{(h)}_{(i,j)}$, the arrivals at h of flows (i, j), $i < h \leq j$, are delayed. Then, $\tilde{A}^{(h)} \leq A^{(h)}$. As the service is exact and convolution is isotonic, $\tilde{D}^{(h)} \leq D^{(h)}$, and the departures are delayed (for any x, $\tilde{t}^{(h)}_x \geq t^{(h)}_x$). Fix a quantity of data x. We know that the contribution for flows (i, j), $j \geq h$ is decreased: with $y = \sum_{(i,j):\ j\geq h} x^{(h)}_{(i,j)}$ and $\tilde{y} = \sum_{(i,j):\ j\geq h} \tilde{x}^{(h)}_{(i,j)}$, $\tilde{y} \leq y$. But then, as the contributions are increasing with x, it follows that for any $k \geq h$, $\sum_{(i,j):\ j\geq k;\ i<h} \tilde{x}^{(h)}_{(i,j)} \leq \sum_{(i,j):\ j\geq k;\ i<h} x^{(h)}_{(i,j)}$. As flows are nested, for each node k, either i) only *old* flows, or ii) every old flow and some new flows, exit at nodes $j \geq k$. In the first case the previous inequality holds. In the second case, the contribution of any subset of new flows increases, hence the complement decreases. To conclude, $\tilde{D}^{(n)} \leq D^{(h)}$ and $\tilde{x}^{(N)}_{(1,N)} \leq x^{(N)}_{(1,N)}$, so $\tilde{F}^{(N+1)}_{(1,N)} \leq F^{(N+1)}_{(1,N)}$. ■

Lemma 5. *Suppose that $F^{(1)}_{(1,N)}$ is α-constrained and α is subadditive. Fix x and t_0 such that $F^{(1)}_{(1,N)}(t_0) = x$. The worst-case delay for bit x of the tagged flow is experienced when the flow is delayed-greedy at time t_0, i.e. $F^{(1)}_{(1,N)}(t) = (x - \alpha(t_0 - t))_+$ if $t < t_0$.*

Proof: We show the lemma by induction. Let $(F^{(h)}_{(i,j)})$ be a scenario and transform the scenario by replacing $F^{(1)}_{(1,N)}$ by $\tilde{F}^{(1)}_{(1,N)} \leq F^{(1)}_{(1,N)}$ such that $\tilde{F}^{(1)}_{(1,N)}(t_0) = F^{(1)}_{(1,N)}(t_0) = x$. The $\tilde{.}$ notation is used for this new scenario and the other CAF are unchanged. $\tilde{A}^{(1)} \leq A^{(1)}$ and $\tilde{A}^{(1)} \otimes \beta_1 \leq A^{(1)} \otimes \beta_1$. Among all possible (and feasible) outputs, one can choose $\tilde{D}^{(1)} = \min(D^{(1)}, \tilde{A}^{(1)})$. As the arrival time of the x-th bit of data remained the same, as well as the cross-traffic, the departure of the x-th bit can only be delayed (as the output process has decreased). Note that, for flows $(i, j) \neq (1, N)$, the output process can only increase: the arrival process of these flows is unchanged, but any bit sees fewer data from flow $(1, N)$: $\forall (i, j) \neq (1, N)$, $\tilde{F}^{(2)}_{(i,j)} \geq F^{(2)}_{(i,j)}$ and $\tilde{A}^{(2)} \leq A^{(2)}$.

Suppose that the result is true for any tandem network with $N - 1$ servers and that after the $N - 1$-th server, $\tilde{F}^{(N)}_{(1,N)} \leq F^{(N)}_{(1,N)}$ and $\tilde{F}^{(N)}_{(i,j)} \geq F^{(N)}_{(i,j)}$ $\forall (i, j) \neq (1, N)$, and that $\tilde{A}^{(N)} \leq A^{(N)}$. One can still define $\tilde{D}^{(N)} = \min(\tilde{D}^{(N)}, A^{(N)})$, and with this departure process, $\tilde{F}^{(N+1)}_{(1,N)} \leq F^{(N+1)}_{(1,N)}$ and $\tilde{F}^{(N+1)}_{(i,j)} \geq F^{(N+1)}_{(i,j)}$ $\forall (i, j) \neq (1, N)$. ■

VI. Numerical evaluation

In this section, we evaluate the upper and lower bounds computed using our LP method, also comparing them to DEBORAH's bounds. To allow the comparison, we restrict the evaluation to tandems whose nodes have rate-latency service curves and flows are shaped by affine (i.e., leaky-bucket) arrival curves. We remark that our work allows more general hypotheses. Three case studies are reported. The first two are related to *nested* tandems, and the third one to a *non-nested* tandem. While the above distinction is irrelevant for our algorithms, it matters for DEBORAH, which employs different algorithms to solve nested and non nested tandems. [3] reports that delay bounds for non-nested tandems should not be expected to be tight. DEBORAH's upper bound, called

LUDB and denoted with V_{LUDB}, is computed by progressively removing cross-flows from the path of the tagged flow, using *equivalent* service curves. Each time a cross flow is removed, a free parameter is added, hence a delay bound can be formulated as a (piecewise linear) function of several parameters. The LUDB is the minimum of the above function, computed by solving an exponentially large number of simplexes. For non nested tandems, mutual dependencies between two cross flows exist, that make it impossible to remove either, hence a pre-processing of the tandem is required to sort out these dependencies. This requires making independent (and possibly mutually incompatible) worst-case scenarios for different sections of the tandem, which is likely to yield loose bounds. The current implementation of DEBORAH computes lower bounds on the WCD based on hypotheses (a)- (c) of Conjecture 1, neglecting hypothesis (d). Both greedy and delayed-greedy CAFs are tested in the cross flows, in all possible combinations (which makes the algorithm exponential). We denote with $v_{\{a,c\}}$ DEBORAH's lower bound, and with $v_{\{a,d\}}$ the one obtained by exploiting all the four hypotheses of Conjecture 1.

A. CS1: one-hop persistent cross traffic

Consider a tandem of N nodes traversed by the tagged flow $(1, N)$ and by cross-flows (i, i), $i \in [1, N]$, as depicted in Figure 5.

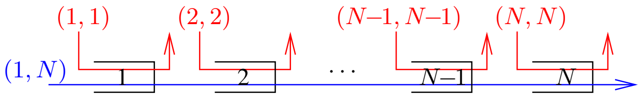

Fig. 5. Tandem network with one-hop cross-traffic.

Assume that all flows have the same leaky-bucket arrival curve $\alpha : t \mapsto \sigma + \rho t$ and all nodes have the same rate-latency service curve $\beta : t \mapsto R(t-\theta)_+$, with $R = 2\rho/U$, $0 < U \leq 1$. U is a utilization factor, so that $U = 0.2$ means that nodes rate is five times the required minimum. In the above hypothesis, both V_{LUDB} and $v_{\{a,d\}}$ can be computed in a closed form [3].

$$V_{LUDB} = N\theta + \frac{U\sigma}{\rho}\left[\frac{N}{2} + \frac{1}{2-U}\right].$$

$$v_{\{a,d\}} = N\theta + \frac{U\sigma}{\rho}\left[\frac{N}{2} + (1-(U/2)^N)/(2-U)\right].$$

Figure 6 reports V_{LUDB}, V_{LP}, $v_{\{a,c\}}$ and $v_{\{a,d\}}$ and v_{LP}. We choose $N = 6$, $\sigma = 2$, $R = 10$, $\theta = 1$ and U ranging from 0.1 to 1, hence ρ ranging from 0.5 to 5. For this scenario, it is $V_{LP} = v_{LP} = v_{\{a,d\}}$. This proves that $v_{\{a,d\}}$ is the WCD (confirming Conjecture 1), and that both the LP bounds are equal to the WCD. V_{LUDB} is nearly tight, overrated by around 0.01, whereas $v_{\{a,c\}}$ loses something to $v_{\{a,d\}}$ because hypothesis (d) is important in this case.

We repeat the above comparison with non-uniform cross-traffic, i.e., $\rho_1 = UR/2$ for odd nodes and $\rho_2 = UR/4$ for even nodes (hence U represents the utilization of the odd nodes), and show the results in Figure 7. In this case, instead, V_{LP} is not tight, as it yields a scenario where some functions are not monotonic. Note, however, that the V_{LP} is i) smaller than the V_{LUDB}, the gap growing as U approaches one, and ii) very close to v_{LP}, certainly much closer than V_{LUDB} is to $v_{\{a,c\}}$.

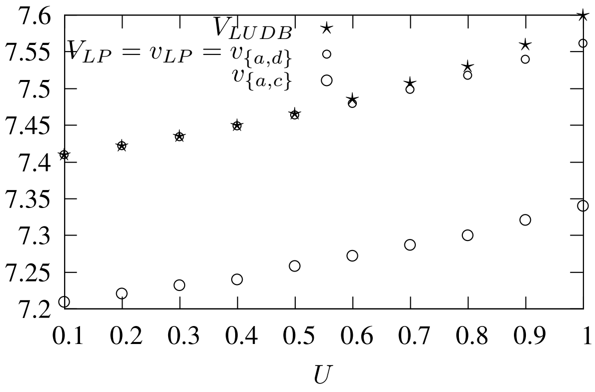

Fig. 6. Delay bounds with uniform one-hop cross-traffic.

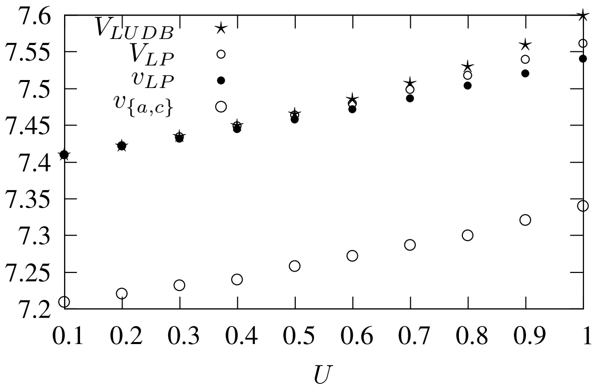

Fig. 7. Delay bounds with non-uniform one-hop cross-traffic.

B. CS2: source-tree tandem

The case-study N-node tandem, shown in Figure 8, is traversed by the tagged flow $(1, N)$ and by cross-flows $(1, i)$, $1 \leq i \leq N$. This is called a *source-tree* tandem, for symmetry with the sink-tree case (see [12]).

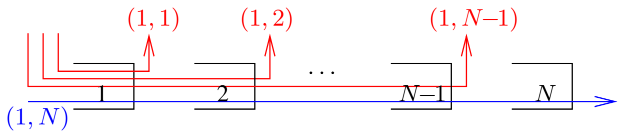

Fig. 8. Source-tree tandem network

This is a case that can be easily solved using a LP approach because the path of every flow starts at node 1. Then, the FIFO order of data can be defined at node 1, and the service order at the other nodes just follows. Consider a scenario $(F^{(h)}_{(1,j)})$, and define the dates as described in Section IV. This scenario is transformed as follows, from node 1 to node N: for any node h, functions $F^{(h+1)}_{(1,j)}$ is unchanged between dates $t_{2^{N-j+2}-1}$

and $t_{2^{N-j+1}-1}$ and is set to $F^{(h)}_{(1,j)}$ otherwise. Note that the functions can only be increased, and the FIFO order of data is preserved, so this new scenario is still feasible. Moreover, it has the same delay. Now, the number of dates of interest can be reduced: for each node h we only need to consider the interval of time where the service is not infinite ($F^{(h+1)}_{(1,j)} \neq F^{(h)}_{(1,j)}$) and the amount of data served by this node.

- *dates:* $t_1 \geq t_3 \geq t_7 \geq \cdots \geq t_{2^{n+1}-1}$, and $t_{2^n} \geq t_{2^n+2^{n-1}} \geq \cdots \geq t_{2^{n+1}-1}$.

The constraints are the same as in Section IV. The number of dates is $O(N)$, so the number of linear constraints is $O(N^2)$. Note that, in this scenario, the WCD is computed in the LP program generated for the lower bound problem, as the latter contains all the constraints of interest.

The LUDB can be computed in a closed form in source-tree tandems, by considering that all *fully nested* tandems (i.e., those where the path of a flow is entirely included into that of another) are tree-equivalent [13] and using the formulas for sink trees [12]. Assume all flows have uniform arrival curves, with $\sigma_{(1,i)} = \sigma$ and $\rho_{(1,i)} = \rho$, and all nodes have uniform service curves, with $\theta_i = \theta$ and $R_i = (N+1-i)\rho/U, 0 < U \leq 1$. In this case we have ([3]) $V_{LUDB} = N\theta + UH_N\frac{\sigma}{\rho}$, where H_N is the N-th harmonic number, and $v_{\{a,d\}} = V_{LUDB}$, hence V_{LUDB} is the WCD. Note that this is not a property of source-tree tandems in themselves. In fact, it is enough to change the nodes' rates to obtain a different situation. Assume, in fact, that the rate of all nodes is constant, i.e. $R_i = N\rho/U$ (again, in this case U represents the utilization of the first node). In this case, we have (see again [3]):

$$V_{LUDB} = N\theta + \frac{\sigma}{\rho}\left[\left(1+\frac{U}{N}\right)^N - 1\right] \text{ and } v_{\{a,d\}} = N\theta + \frac{\sigma}{\rho}U.$$

Figure 9 compares V_{LUDB}, $v_{\{a,d\}}$ and $v_{\{a,c\}}$ with the LP bounds V_{LP}, v_{LP}. We choose $N = 10$, $\sigma = 2$, $R = 20$, $\theta = 1$ and $0 < U \leq 1$, hence ρ ranges from 0.2 to 2. It is always $V_{LP} = v_{LP}$=$v_{\{a,d\}}$, again confirming Conjecture 1 and attaining the exact WCD, whereas V_{LUDB} is increasingly overrated as U approaches 1. Moreover, $v_{\{a,c\}}$ is underestimated.

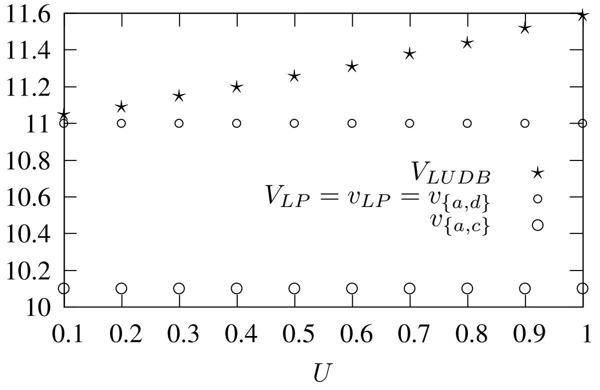

Fig. 9. Source-tree tandem.

C. CS3: non-nested tandem

The tandem shown in Figure 10 is analyzed. Nodes have identical latencies $\theta = 1$, and $R = 10$; all flows have the same arrival curve with $\sigma = 1$ and $\rho = RU/3$ with $0 < U \leq 1$. Figure 11 shows the results of the above scenario. Except for $U \in [0.8, 0.99]$, it is $V_{LP} = v_{LP}$, hence they are equal to the WCD. On the other hand, as expected, V_{LUDB} is overrated, all the more as U approaches one, and $v_{\{a,c\}}$ is slightly underrated.

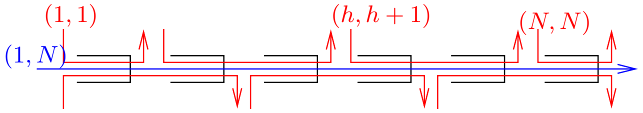

Fig. 10. Non-nested tandem.

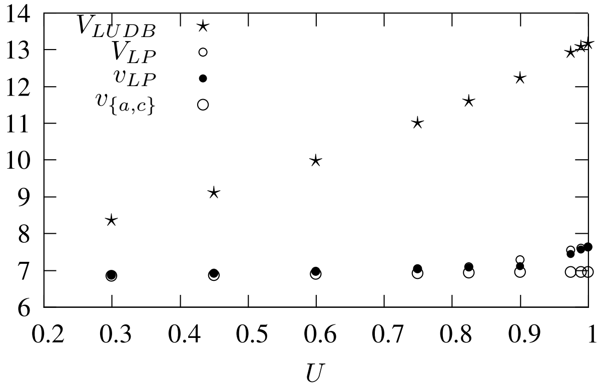

Fig. 11. Delay bounds in a 6-node non-nested tandem.

VII. Conclusions and Future Work

In this paper we have presented a method to compute Worst-case Delays (WCD) for flows in FIFO tandems. The method works under the assumption that nodes exhibit convex, piecewise affine service curves, and flows are regulated by concave, piecewise affine arrival curves. This is the first work that achieves this objective, since previous works only managed to compute (often loose) upper bounds on the WCD, under less general assumptions. The method proposed in this paper relies on Linear Programming. More specifically, it entails solving a number of LP problems. Both the number and the size of these LP problems grow exponentially with the tandem length. We have proposed approximate algorithms to compute both upper and lower bounds on the WCD. Both require only one LP: the one for the upper bound has still an exponential size, whereas the LP for the lower bound has a polynomial (quadratic) size. We have compared both our bounds to those computed using DEBORAH, the only existing tool that computes bounds for FIFO tandems, in some sample scenario. The results show that our bounds are always tighter, and that in many cases they coincide, allowing us to compute the exact WCD. Finally, we have reported some properties related to worst-case scenarios in FIFO tandems. Although they do not allow one to identify such scenario univocally, they provide some insight into the problem. Other works have conjectured what the worst-case scenario for a FIFO tandem should be (under less general hypotheses than ours). Our properties are compatible with that conjecture, which thus rests undisproved as yet.

The work described in this paper can be extended in several directions. A first direction is that of code optimization. This paper gives a proof of concept, but does not attempt to build an optimized tool. One direction to pursue is to derive properties of worst-case scenarios that allow the number of LPs (or their size) to be reduced. Furthermore, we want to explore using non-linear optimization: monotonicity constraint can probably be written as quadratic constraints, which would eliminate the explosion in the number of LPs, though at the price of using non-linear solvers. A second direction is a thorough evaluation of the exact and approximate algorithms in both the dimensions of *accuracy* and *efficiency*. We want to know whether our algorithms scale to acceptably large tandem dimensions (e.g., 10-15 nodes), and, in case, whether exact or approximate WCD computations can be done in real time up to some (possibly smaller) dimension. Finally, although this paper deals with tandem networks, the extension to general feed-forward networks appears not to pose conceptual difficulties. It only requires that more dates and variables be put in the model, with per-path ordering among dates.

References

[1] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea. Estimating the worst-case delay in FIFO tandems using network calculus. In *Proceedings of Valuetools'2008*, 2008.

[2] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea. Deborah: A tool for worst-case analysis of FIFO tandems. In *Proceedings of ISoLA 2010, Special Track on Worst-case Traversal Time*, 2010.

[3] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea. Numerical analysis of worst-case end-to-end delay bounds in FIFO tandem networks. *Springer Real-Time Systems Journal*, 2012.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services*. IETF, 1998.

[5] A. Bouillard, L. Jouhet, and E. Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. In *Proceedings of Infocom'2010*, 2010.

[6] A. Bouillard and A. Junier. Worst-case delay bounds with fixed priorities using network calculus,. In *Proceedings of Valuetools'11*, 2011.

[7] S. Chakraborty, S. Kuenzli, L. Thiele, A. Herkersdorf, and P. Sagmeister. Performance evaluation of network processor architectures: Combining simulation with analytical estimation. *Computer Networks*, 42(5):641–665, 2003.

[8] R. L. Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, 1991.

[9] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, 1991.

[10] A. Koubaa, M. Alves, and E. Tovar. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *Proceedings of IEEE RTSSS'06*, pages 412–421, 2006.

[11] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume LNCS 2050. Springer-Verlag, revised version 4, may 10, 2004 edition, 2001.

[12] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea. Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks. *Performance Evaluation*, 63(9-10):956–987, 2006.

[13] L. Lenzini, E. Mingozzi, and G. Stea. A methodology for computing end-to-end delay bounds in FIFO-multiplexing tandems. *Performance Evaluation*, 65(11-12):922–943, 2008.

[14] E. Rosen, A. Viswanathan, and R. Callon. *Multiprotocol Label Switching Architecture*. IETF, 2001.

[15] J. B. Schmitt and U. Roedig. Sensor network calculus: A framework for worst case analysis. In *Proceedings of 1st International Conference on Distributed Computing in Sensor Systems*, pages 141–154, 2005.

[16] J. B. Schmitt, F. A. Zdarsky, and M. Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch ... In *Proceedings of INFOCOM'2008*, 2008.

[17] T. Skeie, S. Johannessen, and O. Holmeide. Timeliness of real-time IP communication in switched industrial ethernet networks. *IEEE Transactions on Industrial Informatics*, 2:25–39, 2006.