

Infocom 2008

Stability and Delay Bounds in Heterogeneous Networks of Aggregate Schedulers

Gianluca Rizzo

EPFL, CH-1015 Lausanne, Switzerland

Email: gianluca.rizzo@epfl.ch

Jean-Yves Le Boudec

EPFL, CH-1015 Lausanne, Switzerland

Email: jean-yves.leboudec@epfl.ch

Abstract—Aggregate scheduling is one of the most promising solutions to the issue of scalability in networks, like DiffServ networks and high speed switches, where hard QoS guarantees are required. For networks of FIFO aggregate schedulers, the main existing sufficient conditions for stability (the possibility to derive bounds to delay and backlog at each node) are of little practical utility, as they are either relative to specific topologies, or based on strong ATM-like assumptions on the network (the so-called "RIN" result), or they imply an extremely low node utilization. We use a deterministic approach to this problem. We identify a nonlinear operator on a vector space of finite (but large) dimension, and we derive a first sufficient condition for stability, based on the super-additive closure of this operator. Second, we use different upper bounds of this operator to obtain practical results. We find new sufficient conditions for stability, valid in an heterogeneous environment and without any of the restrictions of existing results. We present a polynomial time algorithm to test our sufficient conditions for stability. We show that with leaky bucket constrained flows the inner bound to the stability region derived with our algorithm is always larger than the one determined by all existing results. We prove that all the main existing results can be derived as special cases of our results. We also present a method to compute delay bounds in practical cases.

Keywords: Differentiated Services; QoS; Stability; Network Calculus

I. INTRODUCTION

We study networks of FIFO nodes, where flows are constrained by arrival curves. A crucial issue in these networks is: Can we derive a bound to the maximum delay that a packet can experience when traversing the network, and to the maximum queue size at each node? For a generic FIFO network these are still open questions: instability in FIFO networks is an old problem (see [1], [2] and references therein for a review). A recent result by Andrews [2] shows that, contrary to common sense, no matter how low the maximum node utilization is in the network, it is possible to build an example of an unstable FIFO network. An open issue in these networks is therefore the determination of sufficient conditions for *stability*, defined as the possibility to derive bounds to delay and backlog at all nodes.

We tackle this problem using a deterministic approach based on worst-case behavior, because in order to enable QoS guarantees on networks we need to derive hard bounds on packet delay and queue size.

The focus of this work is on networks of aggregate schedulers. In these networks, the scheduling decision at each node does not take into account to which flow the packets belong. As this implies that nodes do not need to store per-flow information, aggregate scheduling allows for a better scalability of the scheduling policy. Its applications can be found in high speed switches, in Differentiated Services networks, in network-on-chip systems, and in all cases in which the need for QoS guarantees has to comply with a large network size. Some existing positive results regarding stability are related to specific network topologies: It is the case of unidirectional ring networks [3] and of feed-forward networks, which are stable for any value of node utilization inferior to one.

The determination of good sufficient conditions for stability in FIFO networks with a generic topology and in heterogeneous settings appears still to be an open issue. A result by Charny and Le Boudec [4], states that a network with leaky bucket constrained flows is stable if the maximum node utilization in the network is inferior to $(h - 1)^{-1}$, where h is the maximum flow hop count in that network. The main limit of such a result is that in realistic scenarios it leads to very low values of node utilization, as in practical cases h can take quite large values (more than 20 [5]). However, as of today this constitutes the best available result for networks with leaky bucket constrained flows, with packets of different size, and nodes of different service rates. In the rest of the paper, we will refer to this result as the "DiffServ bound".

A second result that does not depend on network topology is the one by Chlamtac et al. [6], [7], which is based on strong, ATM-like assumptions on the network: Flows are constrained by staircase arrival curves, packets are all equal-sized, nodes have all the same service rate, all packet arrival and departure times are synchronous at all nodes (we call this model the *homogeneous network model*). This work introduces a sufficient condition for stability, in terms of lower bounds to the time between the emission of two consecutive packets at each flow's source: For each flow, this time must be not smaller than $RIN + 1$, where RIN is the *route interference number* for that flow (that is, the total number of flows that the considered flow meets along its path, counted with multiplicity if a flow interferes more than once). It also derives simple formulas for backlog and delay bounds at all nodes. The main drawback of

this result is its non-applicability to a generic, heterogeneous network, because its derivation relies on assumptions that make it not useful in many practical cases.

Our aim is to derive good sufficient conditions for stability in heterogeneous network settings and for a generic topology, which are as general as possible, without making any simplifying assumption on the network model. We consider flows constrained by generic arrival curves, with packets of different sizes, and nodes with different service rates. We define two types of variables: the maximum packet delay at each node, and the maximum number of bytes present in a *super chain*, relative to a given flow and to a given sequence of nodes traversed by that flow. The concept of *super chain*, introduced in [6], is used to model packet interactions and their effect on packet delay. We assume the variation over time of these variables to be clocked by events (packet departures). At first, we derive an operator Π (given in detail in Section III) that upper bounds the value of the variables at a given time t_p , in function of their values at t_{p-1} . The first contribution of this work consists in showing how the problem of stability can be reduced to properties of the super-additive closure of this operator. Our second main contribution consists in the derivation of a polynomial time algorithm for testing some sufficient conditions on stability, based on a linear operator that upper bounds Π . Using an example, we show that when flows are leaky bucket constrained, the inner bound to the stability region derived with our algorithm is much larger than those derived with all existing results, thus allowing for radically better performances than before. We demonstrate how the two main existing results (the “RIN result” and the “DiffServ bound”) can be derived as special cases from our approach. Finally, we show how an extension of the algorithm can be used to derive bounds to delay at all nodes.

The paper is organized as follows: In Section II we introduce the network model and the main concepts used in the present paper, and we define the variables in the system. In Section III we describe the operator Π , and we show how stability properties of the network are associated to the super-additive closure of this operator.

In Section IV we describe an algorithm that can be used to test sufficient conditions for stability. In Section V we discuss the tightness of the new sufficient conditions for stability, and in Section VI we relate them to other existing results. Finally, in Section VII we numerically assess the algorithm and compare the performance of the new sufficient stability conditions to existing results.

II. MODEL AND ASSUMPTIONS

A. Network Model

We model a network as a directed graph, where each vertex (hereafter called also “node”) n models a buffer at the input of a physical link. We assume the traffic in the network is organized in flows: Every flow is represented by an integer f , and to each flow it is associated an ordered sequence of traversed nodes. For any couple of nodes n_1 and n_2 in the graph, we assume a directed edge is present from n_1 to n_2 if at least one flow traverses the two nodes in this order. We assume no losses are present at buffers in

the network (buffers of infinite capacity). Using a standard terminology from graph theory, we assume that in general the graph associated to the network can be partitioned in a set of *strongly connected components*. We assume each flow f is constrained by an arrival curve $\alpha_f(t)$, and in general it has packets of different size. We assume for each flow there exists a finite set of possible packet sizes. We consider a network whose nodes are store-and-forward FIFO schedulers that perform aggregate scheduling. Each node n offers to the aggregate of flows a service curve of the rate-latency type¹ $\beta_{r_n, T_n}(t) = r_n(t - T_n)^+$ with service rate r_n and latency T_n , generally different for each node. We assume service curves are *strict* (i.e. during a busy period of duration u , the output of the system is at least $\beta_{r, T}(u)$) [8]. This is a very general node model, encompassing many scheduling disciplines (e.g. priority schedulers, or FIFO constant rate schedulers).

With Δ_n we denote the propagation delay of the physical link at the output of the buffer at node n , and r_n is also the capacity of that link.

We define a *relevant network event* as the dequeuing of a packet at a node. Starting from $t = 0$, we consider the (ordered) succession of time instants t_p , $p \in \mathbb{N}$ associated to relevant network events in the considered network: therefore t_p denotes the time instant of the p -th network event. When two or more network event take place at the same time, we label them in an arbitrary order.

We assume that at time 0 at each node n and for each flow f passing from that node there are $a_f^n \geq 0$ packets from flow f in the queue. Table I describes our notation.

B. Definition of Stability

The definition of stability we use is the following:

Definition 2.1: Consider a network, with each fresh flow f constrained by $\alpha_f(t)$, where at $t = 0$ in the buffer at each node n are present a_f^n bytes from flow f . We say that this network is *totally stable* if $\exists \Gamma > 0$ such that, for any array of input sequences $\mathbf{R}(t) = (R_1(t), \dots, R_F(t))$ relative to fresh flows, and compatible with the given arrival curve constraints,

$$\sup_{t>0} S(t) < \Gamma$$

where $S(t)$ is the total number of bytes in the network at time t , with $S(0) = \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}_n} a_f^n$.

Note that this definition of stability is stronger than the following, commonly adopted in the literature:

Definition 2.2: With the same assumptions as in the previous definition, a network is *stable* when for any array of input sequences $\mathbf{R}(t) = (R_1(t), \dots, R_F(t))$ relative to fresh flows, and compatible with the given arrival curve constraints, $\exists \Gamma > 0$ such that,

$$\sup_{t>0} S(t) < \Gamma$$

We can see that the total stability of a network implies its stability, but the converse is not true. In this paper we focus on total stability.

¹In what follows we use the notation x^+ for $\max(x, 0)$.

C. Definition of the Variables

Our choice of variables is based on the concept of *super chain*, first defined in [6]. Before describing in detail our choice, we recall here some definitions from [6], [9]:

Definition 2.3: Given two packets c and d , and a node n , we say that $c \preceq_n d$ if c and d are in the same busy period at n , and c leaves n before d .

Definition 2.4: Consider a sequence of packets $\mathbf{c} = (c_0, \dots, c_K)$ and a sequence of nodes $\mathbf{n} = (n_1, \dots, n_K)$ (all different). We say that (\mathbf{c}, \mathbf{n}) is a **super chain** if

- nodes n_1, \dots, n_K are all on the path of packet c_0 ;
- $c_{j-1} \preceq_{n_j} c_j$, $j = 1 \div K$;
- the path of packet c_j from n_j to n_{j+1} is a subpath of the path of c_0 .

We call the path of packet c_0 from n_1 to n_K **the path of the super chain**.

We now introduce some more definitions: We define a super chain (\mathbf{c}, \mathbf{n}) in which the first packet c_0 belongs to flow f as **a super chain relative to flow f** .

Definition 2.5: We say that **a packet c is included in a super chain (\mathbf{c}, \mathbf{n})** if either $c = c_j$, $j = 0 \div K$, or there exists an index $l = 1 \div K$, for which it holds $c_{l-1} \preceq_{n_l} c \preceq_{n_l} c_l$.

For any couple of ordered node sequences \mathbf{n} , \mathbf{n}' , we say that \mathbf{n}' is *included in* \mathbf{n} , and we indicate it with $\mathbf{n}' \subseteq \mathbf{n}$, when there exist two sequences of nodes \mathbf{n}_1 , \mathbf{n}_2 such that $\mathbf{n} = (\mathbf{n}_1, \mathbf{n}', \mathbf{n}_2)$.

The choice of variables is the following:

- for any flow f in the network, and for any subpath \mathbf{n} of the path of flow f , for any time instant t_p , $m_f^n[p]$ is the maximum number of bytes belonging to f included in any super chain $(\mathbf{c}, \mathbf{n}')$ relative to flow f , up to time t_p , and with $\mathbf{n}' \subseteq \mathbf{n}$;
- for any node n , for any time instant t_p , the variable $d^n[p]$ is the maximum packet delay at node n up to time t_p .

With B we indicate the total number of variables in the network.

III. THE STABILITY OF THE NETWORK IS RELATED TO THE SUPER-ADDITIVE CLOSURE OF THE UPPER BOUNDING OPERATOR

A. Derivation of the Upper Bounding Operator

Our choice of variables is such that, if it exists a finite bound to their value then the network is stable. The first step is therefore the derivation of an operator that upper bounds the value of the variables at a given time t_p , in function of the values of the variables at time t_{p-1} :

Theorem 3.1: For any integer $p > 0$, we have:

$$\begin{cases} (\mathbf{m}[p], \mathbf{d}[p]) \leq \Pi(\mathbf{m}[p-1], \mathbf{d}[p-1]) \\ \mathbf{m}[0] \leq \mathbf{m}_0 \\ \mathbf{d}[0] \leq \mathbf{d}_0 \end{cases} \quad (1)$$

where $\Pi : (\mathbb{R}^+ \cup +\infty)^B \rightarrow (\mathbb{R}^+ \cup +\infty)^B$ is the operator

TABLE I
NOTATION USED IN THE PAPER.

Symbol	Definition
e	Index of the e -th strongly connected component
$p \in \mathbb{N}$	Index of the p -th relevant network event
\mathcal{F}^n, F^n	Set of all flows passing through node n , of cardinality F^n
$\mathcal{F}(\mathcal{F}^e), F(F^e)$	Set of all flows in the network (resp. in the e -th strongly connected component), of cardinality $F(F^e)$
$\mathcal{N}(\mathcal{N}^e), N(N^e)$	Set of all nodes in the network, (resp. in the e -th strongly connected component), of cardinality $N(N^e)$
\mathcal{U}_f^n	Set of those nodes belonging to $\text{path}(f)$ that precede node n on the path of flow f
$\mathcal{Q}_f^n(Q_f^n)$	Set of all the flows that arrive at node n from the same node as flow f , of cardinality Q_f^n
$\text{path}(f)$	Ordered sequence of nodes, constituted by all the nodes traversed by flow f
\mathcal{P}_f	Set of all the possible subsequences of $\text{path}(f)$
$d^n[p]$	Maximum delay at node n up to time t_p
$\mathbf{d}[p]$	Array of $d^n[p]$ over all n
$m_f^n[p]$	Maximum amount of bytes included in any super chain $(\mathbf{c}, \mathbf{n}')$ relative to flow f , up to time t_p , and with $\mathbf{n}' \subseteq \mathbf{n}$;
$\mathbf{m}[p]$	Array of $m_f^n[p]$ over all f and $\mathbf{n} \in \mathcal{P}_f$
$\mathcal{I}(\mathbf{n})$	Set of all the nodes in the ordered sequence of nodes \mathbf{n} ;
α_f	Arrival curve for flow f at the input to the network
a_f^n	Amount of flow f 's bytes present in the queue at node n at time 0.
r_n	Service rate of node n
T_n	Latency of node n
Δ_n	Propagation delay of the physical link at the output of node n
$\text{prec}(n, f)$	Node that precedes node n on the path of flow f
L_f	Maximum packet size for flow f
B	Total number of variables
We make dependency over the index p of relevant network events visible through square brackets. For instance, $d^n[p]$ is the maximum packet delay at node n , up to time t_p .	

defined by $\Pi(\mathbf{m}, \mathbf{d}) = (\mathbf{m}', \mathbf{d}')$, with $\forall f \in \mathcal{F}, \forall \mathbf{n} \in \mathcal{P}_f$,

$$\begin{aligned} m_f'^{\mathbf{n}} = & \sum_{n \in \mathcal{I}(\text{path}(f))} a_f^n + \alpha_f \left(\sum_{n \in \mathcal{U}_f^{n_1}} (b^n \wedge d^n) + \right. \\ & \left. + \sum_{f'} \sum_{\mathbf{n}' \in \mathcal{G}(f, f', \mathbf{n})} m_{f'}^{\mathbf{n}'} S(\mathbf{n}') + \right) \\ & + \sum_{n \in \mathcal{I}(\mathbf{n})} \left\{ \frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{F}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right\} \end{aligned} \quad (2)$$

$$d'^n = \sum_{f' \in \mathcal{F}^n} \frac{a_{f'}^n}{r_n} + T_n + \Delta_n + \min_{f \in \mathcal{F}^n} \left\{ \sum_{f' \in \mathcal{F}^n \setminus \mathcal{Q}_f^n} \frac{m_{f'}^n}{r_n} + \frac{\max_{f' \in \mathcal{F}^n} L_{f'}}{r_n} + \sum_{f' \in \mathcal{Q}_f^n} m_{f'}^n \left(\frac{1}{r_n} - \frac{1}{r_{\text{prec}(n,f)}} \right)^+ \right\} \wedge \left[\frac{\sum_{f' \in \mathcal{F}^n} \alpha_{f'} \left(\sum_{n'' \in \mathcal{U}_{f'}^n} d^{n''} + \tau \right)}{r_n} - \tau \right] \quad (3)$$

with

$$b^n = \min_{f' \in \mathcal{F}^n \setminus \mathcal{H}(f,n,n_1)} \left\{ \sum_{f'' \in \mathcal{F}^n \setminus (\mathcal{Q}_{f'}^n \cup \mathcal{H}(f,n,n_1))} \frac{m_{f''}^n}{r_n} + \frac{\max_{f'' \in \mathcal{F}^n} L_{f''}}{r_n} + \sum_{f'' \in \mathcal{Q}_{f'}^n} m_{f''}^n \left(\frac{1}{r_n} - \frac{1}{r_{\text{prec}(n,f')}} \right)^+ \right\} \wedge \left[\frac{\sum_{f' \in \mathcal{F}^n \setminus \mathcal{H}(f,n,n_1)} \alpha_{f'} \left(\sum_{n'' \in \mathcal{U}_{f'}^n} d^{n''} + \tau \right)}{r_n} - \tau \right] + \sum_{f' \in \mathcal{F}^n} \frac{a_{f'}^n}{r_n} + T_n + \Delta_n \quad (4)$$

where:

- \mathbf{m}_0 is an array such that $\forall f \in \mathcal{F}, \forall \mathbf{n} \in \mathcal{P}_f, (\mathbf{m}_0)_f^n = L_f$, with L_f the maximum packet size for flow f ;
- \mathbf{d}_0 is such that $\forall n \in \mathcal{N}, (\mathbf{d}_0)^n = \frac{\max_{j \in \mathcal{F}^n} L_j}{r_n} + T_n + \Delta_n$;
- $\forall f \in \mathcal{F}, \forall \mathbf{n} = (n_1, \dots, n_K) \in \mathcal{P}_f, S(\mathbf{n}) = \frac{1}{r_{n_1}} + \sum_{j=2}^K \left(\frac{1}{r_{n_j}} - \frac{1}{r_{n_{j-1}}} \right)^+$
- $\mathcal{H}(f, n, n')$ is the set of those flows $\neq f$ that join the path of flow f at node n , and follow the path of flow f up to (at least) node n' , with n, n' belonging to the same strongly connected component;
- $\forall f, f' \in \mathcal{F}$ and $\forall \mathbf{n} \in \mathcal{P}_f$, consider the set $\{\mathbf{n}' = (n'_0, \dots, n'_k) : \mathbf{n}' \in \mathcal{P}_f \cap \mathcal{P}_{f'}, \exists e : \mathcal{I}(\mathbf{n}') \subseteq \mathcal{N}^e, n'_k \in \mathcal{I}(\mathbf{n})\}$. Then $\mathcal{G}(f, f', \mathbf{n})$ is the subset of all the maximal elements of that set.

Proof: see Appendix A.

B. A First Sufficient Condition for Stability

In this section we show how from the properties of the operator Π defined by the upper bounds to variables in Theorem 3.1 we can derive sufficient conditions for the total stability of a network. These conditions are based on the concept of *super-additive closure*:

Definition 3.1 (Super-additive Closure): Let \mathcal{E} be a partially ordered set such that, for any enumerable subset of it, the supremum is well defined, and let Φ be an operator $\mathcal{E} \rightarrow \mathcal{E}$. Denote with $\Phi^{(l)}$ the operator $\mathcal{E} \rightarrow \mathcal{E}$ obtained by composing l times the operator Φ with itself. By convention, $\forall x \in \mathcal{E}, \Phi^{(0)}(x) = x$. Then the *super-additive closure* of Φ , indicated

with Φ^* , is defined by

$$\forall x \in \mathcal{E}, \Phi^*(x) = \sup_{l \in \mathbb{N}} \Phi^{(l)}(x)$$

By applying this definition to the operator Π defined in Theorem 3.1, we derive a first sufficient condition for total stability:

Theorem 3.2: Let $(\mathbf{m}^*, \mathbf{d}^*) = \Pi^*(\mathbf{m}_0, \mathbf{d}_0)$, where Π^* is the super-additive closure of Π . If either \mathbf{m}^* or \mathbf{d}^* are finite, then

- the network is stable, and
- delay bounds at all nodes can be derived from \mathbf{d}^* and \mathbf{m}^* .

Proof: see Appendix B.

IV. CHECKING STABILITY CONDITIONS IN POLYNOMIAL TIME FOR LEAKY BUCKET CONSTRAINED FLOWS

A. A Bounding Methodology

A negative aspect of Theorem 3.2 is that in general we do not know how to compute the super-additive closure of Π . We also note that Π has a nonlinear expression. Then in order to derive practical results we use the following procedure: We first derive an operator Π' that upper bounds Π for any $\mathbf{x} \in (\mathbb{R}^+ \cup +\infty)^B$, and which is linear and monotonic. Then we show that if the fixed point problem $\Pi'(\mathbf{x}) = \mathbf{x}$ admits a finite solution larger than $(\mathbf{m}_0, \mathbf{d}_0)$, this solution is a finite upper bounds the variable values, and therefore the network is totally stable. Therefore, sufficient conditions for the existence of a finite solution for this fixed point problem are sufficient conditions for the total stability of the network.

The result at the basis of our procedure is the following:

Theorem 4.1: If Φ is a monotonic operator $\Phi : (\mathbb{R}^+ \cup +\infty)^B \rightarrow (\mathbb{R}^+ \cup +\infty)^B$ such that $\forall x \in (\mathbb{R}^+ \cup +\infty)^B, \Phi(x) \geq \Pi(x)$ (where Π is the operator described in Theorem 3.1), if the fixed point problem $\Phi(\mathbf{m}, \mathbf{d}) = (\mathbf{m}, \mathbf{d})$ admits a solution $(\mathbf{m}_s, \mathbf{d}_s)$ larger than $(\mathbf{m}_0, \mathbf{d}_0)$, and if either \mathbf{m}_s or \mathbf{d}_s are finite, then the network is stable.

Proof: Theorem 4.1 derives from the following two lemmas:

Lemma 4.1: If Φ_1 and Φ_2 are two operators $\mathcal{E} \rightarrow \mathcal{E}$ such that $\forall x \in \mathcal{E}, \Phi_1(x) \geq \Phi_2(x)$, and if Φ_1 is monotone, then we have that $\Phi_1^*(x) \geq \Phi_2^*(x)$.

Proof (Lemma 4.1): By the definition of super-additive closure, for any integer $l \geq 0$, and $\forall x \in \mathcal{E}, \Phi_1^*(x) \geq \Phi_1^{(l)}(x) \geq \Phi_2^{(l)}(x)$, and by definition of supremum, this implies that $\Phi_1^*(x) \geq \Phi_2^*(x)$. \square

Lemma 4.1 implies that in Theorem 3.2 we can use the super-additive closure of an operator that upper bounds Π .

Lemma 4.2: If Φ is a monotonic operator $\mathcal{E} \rightarrow \mathcal{E}$, and $x_s \in \mathcal{E}$ is a solution (finite or not) of the fixed point problem $x = \Phi(x)$ such that $x_s \geq x_0$ with $x_0 \in \mathcal{E}$, then we have that $x_s \geq \Phi^*(x_0)$.

Proof (Lemma 4.2): As $x_s \geq x_0$, we have by the monotonicity of Φ , that for any integer $l \geq 0, \Phi^{(l)}(x_s) \geq \Phi^{(l)}(x_0)$. Again, as $\Phi(x_s) = x_s$, we get that $\sup_{l \geq 0} \Phi^{(l)}(x_s) = x_s \geq \sup_{l \geq 0} \Phi^{(l)}(x_0) = \Phi^*(x_0)$. \square

By carefully choosing the operator that upper bounds Π , we can derive different results in terms of sufficient conditions for total stability of the network. The operator that is used to derive the main practical results in this paper is described in the following lemma:

Lemma 4.3: For any $(\mathbf{m}, \mathbf{d}) \in (\mathbb{R}^+ \cup +\infty)^B$, we have that $\Psi(\mathbf{m}, \mathbf{d}) \geq \Pi(\mathbf{m}, \mathbf{d})$, where

$\Psi : (\mathbb{R}^+ \cup +\infty)^B \rightarrow (\mathbb{R}^+ \cup +\infty)^B$ is a monotone operator defined by $\Psi(\mathbf{m}, \mathbf{d}) = (\mathbf{m}^*, \mathbf{d}^*)$, where:

- for any flow f and any subpath \mathbf{n} of $\text{path}(f)$, for $m_f^{\mathbf{n}}$ it holds the upper bound obtained from the one in (2), by replacing for any interfering flow f' , any variable $m_{f'}^{\mathbf{n}'}$ with the variable $m_{f'}^{\mathbf{n}''}$ (with \mathbf{n}' included in \mathbf{n}''), where \mathbf{n}'' is the maximal path of flow f' in the strongly connected subnet to which \mathbf{n}' belongs, and by replacing at any node n , $b^n \wedge d^n$ with d^n .
- For any node n , for d^{*n} it holds the upper bound obtained from the one in (3), by using only variables \mathbf{d} .

Proof: By the same definition of the variables in the network, for any flow f and any two sequences of node \mathbf{n} , \mathbf{n}' such that $\mathbf{n} \subseteq \mathbf{n}'$, for any $p > 0$ it holds $m_f^{\mathbf{n}}[p] \leq m_f^{\mathbf{n}'}[p]$. We then observe that in (2) and (3) $(\mathbf{m}', \mathbf{d}')$ is a function of variables which are each relative to a single strongly connected subnet. Then Lemma 4.3 is derived from Theorem 3.1, by using this property, and taking into account that any arrival curve $\alpha(t)$ is a nondecreasing function of t . \square

We consider in what follows the case in which flows are leaky bucket constrained. Leaky bucket constrained traffic represents a case of particular interest in applications: for example, the network model on which DiffServ framework is based assumes leaky bucket constrained flows. We can see that in this case, the operator Ψ defined in Lemma 4.3 is linear.

B. A Polynomial Time Algorithm

We introduce here one of our main results, which consists in an algorithm that allows us to decide in polynomial time if a network, with leaky bucket constrained flows and with nodes performing aggregate scheduling, satisfies some sufficient conditions for total stability.

The basis of the algorithm is in the following result, which defines a new set of sufficient conditions for total stability:

Theorem 4.2: For any strongly connected component e , let's consider the matrix V_1^e , of dimension $\leq F^2$,

$$\forall f, f' \in \mathcal{F}^e, (V_1^e)_{f,f'} = \rho_f \sum_{\mathbf{n} \in \mathcal{G}(f,f', \text{path}(f)), I(\mathbf{n}) \in \mathcal{N}^e} S(\mathbf{n})$$

and the matrix V_2^e , of dimension $\leq N^2$, whose elements are

$$\forall n, n' \in \mathcal{N}^e \quad (V_2^e)_{n,n'} = \begin{cases} \sum_{f \in \mathcal{M}(n',n)} \frac{\rho_f}{r_n} & n \neq n' \\ 0 & n = n' \end{cases} \quad (5)$$

where $\mathcal{M}(n', n)$ is the set of flows passing through node n' and n (in the order). If for all e the spectral radius of either V_1^e or V_2^e is inferior to one, then the network is stable.

Proof: see Appendix C.

We note that neither the burstiness of arrival curves nor the amount of buffer content at nodes at $t = 0$ are involved in the sufficient condition for total stability in Theorem 4.2.

Based on Theorem 4.2, we outline an algorithm that can be used to test the new sufficient conditions for total stability. We note that Algorithm 1 can be also applied to flows with staircase arrival curves, because we can always find a leaky bucket arrival curve that constrains the flow, derived from its staircase arrival curve, as described in [8].

Algorithm 1 Total Stability in networks with leaky bucket constrained flows

INPUT:

- for any node n : r_n ; and
- for any flow f : $\text{path}(f)$, σ_f and ρ_f .

```

1: compute the strongly connected components of the network;
2: for any strongly connected component  $e$  do
3:   for any flow  $f$  in the  $e$ -th strongly connected component do
4:     for any flow  $f'$  in the  $e$ -th strongly connected component do
       compute  $\mathcal{G}(f, f', \text{path}(f))$ 
       compute
       
$$(V_1^e)_{f,f'} = \rho_f \sum_{\mathbf{n} \in \mathcal{G}(f,f', \text{path}(f)), I(\mathbf{n}) \in \mathcal{N}^e} S(\mathbf{n})$$

5:   end for
6: end for
7: for any node  $n$  in the  $e$ -th strongly connected component do
8:   for any node  $n'$  in the  $e$ -th strongly connected component do
       compute  $\mathcal{M}(n', n)$ 
       compute  $(V_2^e)_{n,n'} = \sum_{f \in \mathcal{M}(n',n)} \frac{\rho_f}{r_n}$ 
9:   end for
10: end for
11: compute spectral radius of  $V_1^e$  and  $V_2^e$ 
12: if spectral radius of  $V_1^e$  and of  $V_2^e$  is  $\geq 1$  then
       return the network may be not totally stable
13: end if
14: end for
       return network is stable

```

Let us analyze now the worst-case time complexity of the various parts that compose the algorithm:

- computation of the strongly connected components: $O(FN + N^2)$;
- computation of matrices V_1 and V_2 : $O(N(F^2N^2 + FN^2))$; and
- spectral radius computation (with a relative error bound of 2^{-b} [10]): $O(N[N^3 + \log(b)N \log^2 N + F^3 + \log(b)F \log^2 F])$.

We have therefore a polynomial time complexity. We finally observe that when a network is stable according to Algorithm 1, an extension of the same algorithm ([11]) allows to derive

delay bounds at all nodes.

C. Comparison to the “DiffServ bound” and to the “Generalized RIN Result”

We observe how the “DiffServ bound” [4] can be derived from Theorem 4.2:

Theorem 4.3: In a network with leaky bucket constrained flows, if the maximum node utilization in the network is inferior to $(h - 1)^{-1}$, where h is the maximum hop count for a flow in the network, then the network is stable.

Proof: If h is the maximum hop count in the network, $\forall n$ the sum of the elements in the row relative to node n of matrix V_2^e , $\forall e$ in Theorem 4.2 is upper bounded by $(h - 1)(\sum_{f \in \mathcal{F}^n} \rho_f)/r_n$, that is by $h - 1$ times the node utilization of node n . By imposing $\max_{n \in \mathcal{N}}(\sum_{f \in \mathcal{F}^n} \rho_f)/r_n < (h - 1)^{-1}$ we have that the network is stable. \square

Moreover, we can verify how the “Generalized RIN result” in [9] can be derived as a special case from Theorem 4.2:

Theorem 4.4 (Generalized Source Rate Condition [9]):

With the given assumptions on the network, when flows are leaky bucket constrained, if for any flow f , indicating with $(n_1^e, \dots, n_{K^e}^e)$ its path in the e -th strongly connected subnet, its rate ρ_f satisfies the condition $\rho_f < \min_e h^e$, with:

$$h^e = \left[\frac{F_{n_1^e}^e}{r_{n_1^e}^e} + \sum_{j=2}^{K^e} \frac{F_{n_j^e}^e - Q_{n_j^e}^e}{r_{n_j^e}^e} + Q_{n_j^e}^e \left(\frac{1}{r_{n_j^e}^e} - \frac{1}{r_{n_{j-1}^e}^e} \right) \right]^+{}^{-1} \quad (6)$$

then the network is stable.

Proof: The result derives from imposing that for all e the row sums of all matrices V_1^e in Theorem 4.2 be smaller than one. \square

V. DISCUSSION OF TIGHTNESS

For a generic network, determining the tightness of the sufficient conditions for total stability, tested by Algorithm 1, is an open issue. However, we can get an idea of how those sufficient conditions perform by applying them to some network examples for which we know their stability behavior. One example is given by feed-forward networks, which are known to be always stable, provided that the *natural condition* (sum of flow rates is inferior to node service rate) is satisfied [12]. For those networks, each node represents a strongly connected component of the network: In this case, by imposing that the spectral radius of each of the matrices in Theorem 4.2 be smaller than one we can easily see that we derive the natural condition.

Another example of network that is known to be always stable is the ring [3], provided that node utilization is strictly smaller than one. In these networks, Algorithm 1 performs differently according to the number of flows in the ring and to their path, and in general its outcome is that the ring is not totally stable for any value of flow rates satisfying the natural condition at all nodes. As an example, consider the network in Fig. 1: with $\rho_b = 0$, the maximum value of ρ_a for which the network is stable is $\rho_a \leq 0.0909$, that leads to a maximum node utilization of 0.5454.

VI. DISCUSSION ON STATE OF THE ART

In [13] Otel derives a set of sufficient conditions for total stability which extend the “RIN result” to heterogeneous networks, but are in the form of a minimum packet inter-arrival time for each flow. This way of shaping input traffic in a network is not compatible with constraints given in the form of arrival curves, which are the most commonly used in the majority of network models. That is, this sufficient condition cannot be mapped into an arrival curve constraint. This makes the result not useful in practical cases. Moreover, in these sufficient conditions the minimum packet inter-arrival time for each flow scales linearly with the maximum packet size (or burst size) among all flows in the network, leading to a very poor performance in terms of node utilization. As an example, in a network where all flows satisfy the sufficient condition in [13] with a maximum packet size of 60 bytes, increasing the maximum packet size to 1500 bytes would decrease the maximum node utilization by a factor of 25.

In [9], following the same approach as in the “RIN result” and using the concept of super chain, sufficient conditions for total stability are derived (called “Generalized RIN result” or GRIN), which extend the “RIN result” to leaky bucket constrained flows and to heterogeneous networks. The lower bound to the period of staircase arrival curve of the “RIN result” becomes, in this work, an upper bound to the rate for each flow, which is a function of an extension of the concept of “route interference number”. The sufficient conditions derived in the present paper are less tight than those in the GRIN result: indeed, as shown in Section IV-C, GRIN can be derived from the new sufficient conditions through an approximation that brings to more conservative sufficient conditions. Numerically, we can verify in Fig. 2 that the inner bound to the stability region obtained by Algorithm 1 in Section IV-B is much larger than the one derived with the GRIN conditions, as well as the ones derived with any of the other existing results.

VII. NUMERICAL ASSESSMENT OF THE RESULTS

In order to evaluate the performance of the algorithm in Section IV-B, we applied it over the network example in Fig. 1. The network is composed of $N = 6$ nodes in a ring structure, labeled from 1 to N . For simplicity, we assume all flows are of two types, “a” and “b”, and that all flows of the same type have the same rate (respectively ρ_a and ρ_b) and the same burstiness $\sigma_a = \sigma_b$. At any node $n \in [1, N]$ a fresh flow of “type a” enters the network, traverses clockwise all nodes on the ring, and exits the network at node $(n + N - 1)_{\text{mod}(N)}$. At any node n we also have a fresh flow of “type b”, that traverses nodes n and $(n + N/2)_{\text{mod}(N)}$ and then exits the network.

Definition 7.1: The *stability region* of a network with leaky bucket constrained flows, is the closure of the set of all those vectors of flow rates for which the network is totally stable. By using Algorithm 1, we derived an inner bound to the stability region of the network in Fig. 1, with all service rates equal to 1, with flow burstiness equal to 1, with link capacities equal to 1, and with all buffers in the network empty at $t = 0$. In Fig. 2, the dash-dotted line represents the inner bound to the capacity region obtained by Algorithm 1 in Section IV-B;

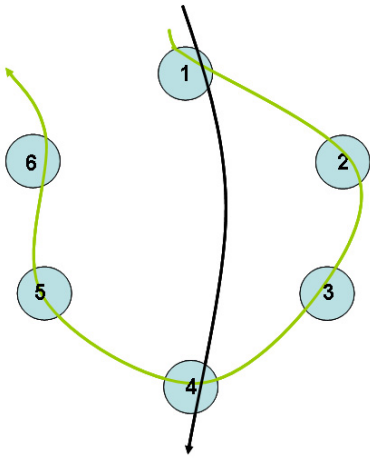


Fig. 1. An example of the symmetric network on which Algorithm 1 in Section IV-B was run, with $N = 6$ nodes. The nodes are disposed on a ring. At each node n ($n \in [1, N]$) there is a "type a" fresh flow, with rate ρ_a , that traverses clockwise all nodes on the ring, and exits the network at node $(n + N - 1)_{\text{mod}(N)}$, and a "type b" flow, that traverses nodes n and $(n + N/2)_{\text{mod}(N)}$, and then exits the network. Only the paths of fresh flows at node 1 are shown in the figure.

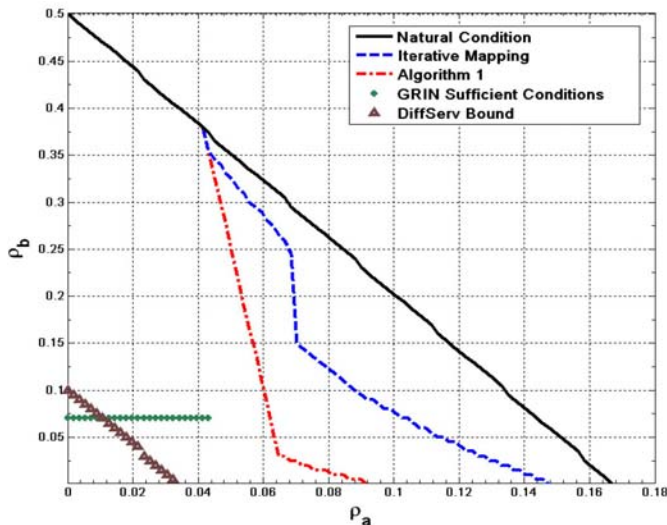


Fig. 2. Inner and outer bounds to the stability region of the network in Fig. 1, with $N = 6$ nodes, with all service rates equal to 1, $\sigma_a = \sigma_b = 1$, link capacities equal to 1, and buffers at all nodes empty at $t = 0$. The straight line is the border of the region in which the natural condition (sum of flow rates is inferior to node service rate) is satisfied, and therefore it represents an outer bound to the stability region of the network. The dashed line is the inner bound to the stability region obtained by iterative application of the upper bound in Theorem 3.1, with a maximum number of iterations equal to 100, and considering the iterations to have converged when the increase in the upper bounds between steps n and $n + 1$ is inferior to 1%. The dash-dotted line represents the inner bound derived by using Algorithm 1 in Section IV-B; the dotted line and the one with triangles represent the inner bounds derived, respectively, through the the Generalized RIN condition [9] and through the "DiffServ bound" [4].

the dotted line and the one with triangles represent the inner bounds derived through the the generalized RIN condition in [9], and the "DiffServ bound" [4] respectively. The blue dashed line is the inner bound obtained as follows: We

considered the quantity $\sup_{0 \leq l \leq n} \Pi^{(l)}(\mathbf{x}_0)$ for increasing values of n , stopping when the increase in the value of this quantity between two consecutive values of n is inferior to 1%. We limited the maximum value of n to 100, and we computed the region of values for which the iterations stop. The resulting region gives an idea of what we could gain if we were able to derive $\Pi^*(\mathbf{x}_0)$. Finally the straight line limits the region in which the natural condition is satisfied, and it represents therefore an outer bound to the stability region of the considered network.

We can observe how with Algorithm 1 we have an inner bound that is much larger than the one derived by applying any of the previously available sufficient conditions.

VIII. CONCLUSION

In this paper we consider the problem of deriving good sufficient conditions for stability in networks of FIFO aggregate schedulers. We use an algebraic approach. First, we identify a nonlinear operator on a vector space of finite (but large) dimension and show that the problem can be reduced to properties of the super-additive closure of this operator. Second, we use different upper bounds of this operator to obtain practical results. We find new sufficient conditions for stability, which are valid without any of the restrictions of the "RIN result". We derive a polynomial time algorithm to test our sufficient conditions for stability. We show that in the case when flows are leaky bucket constrained, the inner bound to the stability region derived with our algorithm is always larger than the one determined by all existing results. We prove that both the "DiffServ bound" and the "RIN result" can be derived as special cases of our result. We also present a method to compute delay bounds in practical cases.

REFERENCES

- [1] T. Seidman, "First Come, First Served" can be unstable!" *IEEE Transactions on Automatic Control*, vol. 39, no. 10, October 1994.
- [2] M. Andrews, "Instability of FIFO in the Permanent Sessions Model at Arbitrarily Small Network Loads," in *ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, New Orleans, USA, January 2007.
- [3] L. Tassiulas and L. Georgiadis, "Any work-conserving policy stabilizes the ring with spatial re-use," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 205–208, August 1996.
- [4] A. Charny and J.-Y. L. Boudec, "Delay Bounds in a Network with Aggregate Scheduling," in *QoSIS*, September 2000, pp. 1–13.
- [5] F. Begtasevi and C. V. Miegheem, "Measurements of the Hopcount in Internet," in *Proceedings of Passive and Active Measurement (PAM2001)*, Amsterdam, April 2001.
- [6] I. Chlamtac, A. Faragó, H. Zhang, and A. Fumagalli, "A Deterministic Approach to the End-to-end Analysis of Packet Flows in Connection Oriented Networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 422–431, August 1998.
- [7] J.-Y. Le Boudec and G. Hebuterne, "Comment on A Deterministic Approach to the End-to-end Analysis of Packet Flows in Connection Oriented Networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 121–124, February 2000.
- [8] J.-Y. Le Boudec and P. Thiran, *Network Calculus. A Theory of Deterministic Queueing Systems for the Internet*. Springer Verlag LNCS, vol. 2050, July 2001.
- [9] G. Rizzo and J. L. Boudec, "Generalization of the RIN Result to Heterogeneous Networks and to leaky bucket constrained Flows," *ICON 2007, Adelaide*, Nov 2007.
- [10] Z. C. V.Y. Pan and A. Zheng, "The Complexity of the Algebraic Eigenproblem," December 1998.
- [11] G. Rizzo, "Stability in networks with aggregate scheduling," February 2008, Ph.D. Dissertation, EPFL.

- [12] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu, "Universal Stability Results for Greedy Contention Resolution Protocols," in *37th Annual IEEE Symposium on Foundations of Computer Science (FOCS'96)*, Burlington VT, October 1996.
- [13] F.-D. Otel and J.-Y. L. Boudec, "Deterministic End-to-End Delay Guarantees in a Heterogeneous Route Interference Environment," in *QoSIS*, October 2003, pp. 21–30.

APPENDIX

A. Proof of Theorem 3.1

Definition A.1: The **super chain time** is the time at which the last packet in the super chain is served at the last node of the super chain.

Also, with reference to Definition 2.4, at each node n_j on the path of the super chain we call the packets c_{j-1} and c_j , respectively, *open packet* and *close packet* for that node.

Lemma A.1: The delay of a packet of flow f that arrives at a node n with service rate r_n and latency T_n on a link with rate r' is upper bounded by the quantity

$$\frac{\Theta}{r_n} + \Gamma \left(\frac{1}{r_n} - \frac{1}{r'} \right)^+ + \frac{A + \max_{f' \in Q_f^n} L_{f'}}{r_n} + T_n + \Delta_n \quad (7)$$

where Γ , Θ and A represent the sum of packet sizes for packets served in the same busy period as the considered packet and before this packet. Γ and Θ refer to packets arrived at the node, respectively, on the same link as the considered packet and from other nodes or sources, while A refers to packets present at the node at $t = 0$.

Proof: The delay of the considered packet is the same it would experience in the case in which no packet arrives at the node before the beginning of the considered busy period and after the arrival of the considered packet: hence we consider this case. Then the bound in (7) derives from computing the maximum horizontal distance between an aggregate arrival curve for all input flows of the form $\Theta + \max_{f' \in Q_f^n} L_{f'} + \min(\Gamma, r't)$, and the service curve $r_n(t - T_n - A/r_n)$, and including the constant delay Δ_n . \square

Proof (Theorem 3.1): We proceed by induction on the index p of relevant network events.

Base case: $p=1$. In order to upper bound the quantities $m_f^n[1]$, for any flow f , the worst case to consider is the one in which the first two relevant network events in the network are relative, respectively, to the first and the second packet served in the same busy period at a given node $n \in \mathcal{I}(\mathbf{n})$.

Therefore we put in this case, and we consider the busy period that starts at node n at $t = 0$, in which the first and the second packets served belong respectively to flows f and f' . We have in this case a super chain relative to f formed by two packets and a single node, n . t_1 is the time at which the second packet in the busy period is served: As it must be in the same busy period as the one served at t_0 , then it must have arrived at the node by time t_0 (t_0 is the time after which the node starts serving the second packet, which should then be already at the node). As a consequence, also taking into account those packets present at the node at $t = 0$, we have for flow f that $m_f^n[1] \leq \alpha_f(t_0) + a_f^n$. Now we have that $t_0 \leq (m_f[0] + m_{f'}[0] + L_f)/r_n + \Delta_n + T_n$, as the first packet of the considered super chain belongs to flow f , so inequality in (2) holds.

For packet delay at node n , as the node has a strict service curve, we have that $d^n[1] \leq (m_f[0] + m_{f'}[0] + L_f)/r_n + \Delta_n + T_n$. For all the variables $m_f^n[1]$ for which $n \notin \mathcal{I}(\mathbf{n}')$, as well as for all variables $d^{n'}[1]$ with $n' \neq n$ the bound we derived here is clearly conservative.

Induction step. $\forall n \in \mathcal{N}$, the upper bound to $d^n[p]$ derives from Lemma A.1, where for each flow f' traversing node n we use $m_{f'}^n[p-1]$ to upper bound the contribution of this flow to delay, and from computing the horizontal distance between the aggregate arrival curve at node n , and the service curve at the node.

We derive now an upper bound to the variable $m_f^n[p]$. Let us define the *duration* of a super chain with time t as the time interval from the emission time of the first packet of the super chain up to time t .

Let us consider a given super chain (\mathbf{c}, \mathbf{n}) relative to flow f and with time t_q , $q \leq p$. The maximum number of flow f 's bytes that can be included in this super chain is upper bounded by all the flow f 's packets present at $t = 0$ in the network, plus the maximum number of bytes from flow f that can be emitted from the emission time of packet c_0 (the first packet of the super chain) up to time t_{q-1} (and not up to time t_q , as in order to be served by time t_q or before at the last node of the super chain, a packet must arrive at that node by time t_{q-1}). This time interval is upper bounded by the maximal duration of a super chain relative to flow f and to the sequence of nodes \mathbf{n} , with time $\leq t_{p-1}$. In what follows we derive an upper bound to this maximal duration. If n_1 is the first node in \mathbf{n} , we indicate with \mathbf{n}^* the sequence of nodes on the path of flow f that precede n_1 . For each node n'_k , $k = 1, \dots, K$ in the sequence $(\mathbf{n}^*, \mathbf{n})$ we indicate with $\Delta t_{int,k}$ the delay experienced at node n'_k by the open packet at that node (for nodes $\in \mathcal{I}(\mathbf{n}^*)$ we consider the open packet to coincide with the close packet). We denote with Γ^k and Θ^k the sum of packet sizes for packets served in the same busy period as the open packet and before it, and coming, respectively, from node n'_{k-1} (or from the same source as flow f) and from other nodes (or from fresh flows different than f). Then using Lemma A.1 we have for $k \geq 1$

$$\Delta t_{int,k} \leq \frac{\Theta^k}{r_{n'_k}} + \Gamma^k \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{prec(n'_k, f)}} \right)^+ + \frac{A'_k + \max_{f' \in Q_f^{n'_k}} L_{f'}}{r_{n'_k}} + 2T_{n'_k} + \Delta_{n'_k} = B_k + 2T_{n'_k} + \Delta_{n'_k} \quad (8)$$

where A'_k is relative to packets present at the node at $t = 0$. If Δt_k is the time interval between the departure of the open packet and of the close packet, using the definition of strict service curve, we write

$$\Delta t_{int,k} + \Delta t_k \leq B_k + T_{n'_k} + \Delta_{n'_k} + \frac{\gamma^k + \theta^k + A_k}{r_{n'_k}} \quad (9)$$

γ^k and θ^k represent the sum of packet sizes for packets included in the considered super chain and arrived at node n'_k , respectively, from node n'_{k-1} (or from the same source as

flow f) and from other nodes (or from fresh flows different than f). A_k represent the sum of packet sizes for packets present at the node at $t = 0$ and served between the open and the close packet at the node. Therefore the duration of a generic super chain relative to flow f , to the sequence of nodes \mathbf{n} and with time $\leq p - 1$ is upper bounded by the sum of bounds (9) over all the nodes in the path of flow f . At each node n'_k , each of the terms γ^k , θ^k , Γ^k , Θ^k , A_k , A'_k can be written as a sum of the contributions from all the input flows at the node: for instance, for Θ^k we can write $\Theta^k = \sum_{f' \in \mathcal{N}'_k} \Theta^k_{f'}$. For any flow f' (not necessarily distinct from f), $\forall \mathbf{n}' = (n'_1, \dots, n'_{l+K'}) \in \mathcal{G}(f, f', \mathbf{n})$, the contribution to the upper bound to the super chain duration is

$$\frac{\Theta^k_{f'} + \theta^k_{f'}}{r_{n'_l}} + \sum_{k=l}^{l+K'} \left[\Gamma^k_{f'} \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{n'_{k-1}}} \right)^+ + \frac{\gamma^k_{f'} + a^k_{f'}}{r_{n'_k}} \right] \quad (10)$$

By definition of the variables $m^{\mathbf{n}'}_f[p]$, we have that $((\dots(\Theta^k_{f'} + \theta^k_{f'} + \Gamma^k_{f'} + \gamma^k_{f'}) \vee \Gamma^{l+1}_{f'} + \gamma^{l+1}_{f'}) \vee \Gamma^{l+2}_{f'} + \gamma^{l+2}_{f'}) \vee \dots \vee \Gamma^{l+K'}_{f'} + \gamma^{l+K'}_{f'} \leq m^{\mathbf{n}'}_f[p - 1]$. Therefore, the quantity in (10) is upper bounded by

$$m^{\mathbf{n}'}_f[p - 1]S(\mathbf{n}') + \sum_{\mathbf{n}' \in \mathcal{I}(\mathbf{n}')} \frac{a^{\mathbf{n}'}_{f'}}{r_{n'}} \quad (11)$$

For any node $n'_k \in \mathcal{U}_f^{n_1}$, $\Delta t_k = 0$, and $\Delta t_{int,k}$ is upper bounded by the minimum between $d^{n'_k}[p - 1]$, and a bound to delay derived in the same way as the bound to $d^{n'_k}[p]$: In this last bound, the contributions from any flow $f' \in \mathcal{H}(f, n'_k, n_1)$ are absent, as they are already taken into account by a term of the form $m^{\mathbf{n}'}_f[p - 1]S(\mathbf{n}')$, with $\mathbf{n}' = (n'_k, \dots, n_1, \dots)$. For the contributions relative to all the other flows, we note that $\Gamma^k_{f'} + \gamma^k_{f'} + \Theta^k_{f'} + \theta^k_{f'} \leq m^{\mathbf{n}'_k}_f[p - 1]$: we finally get the upper bound in (4). Then for the considered super chain (as well as for any super chain (\mathbf{c}, \mathbf{n}) relative to flow f and with time $\leq t_{p-1}$), putting together the contribution from all flows, we finally get the following upper bound to the duration:

$$\begin{aligned} & \sum_{n \in \mathcal{U}_f^{n_1}} (b^n[p - 1] \wedge d^n[p - 1]) + \\ & + \sum_{f'} \sum_{\mathbf{n}' \in \mathcal{G}(f, f', \mathbf{n})} m^{\mathbf{n}'}_f[p - 1]S(\mathbf{n}') + \\ & + \sum_{\mathbf{n} \in \mathcal{I}(\mathbf{n})} \left(\frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a^{\mathbf{n}'}_{f'}}{r_n} + 2T_n + \Delta_n \right) \end{aligned}$$

Then using the expression of flow f 's arrival curve, and taking into account those packets from flow f present in the buffers in the network at $t = 0$, we finally derive the upper bound in (2). \square

B. Proof of Theorem 3.2

In order to prove Theorem 3.2, we first need the following lemma:

Lemma A.2: If Π is the operator defined in Theorem 3.1, we have that $\forall p \in \mathbb{N}$, $(\mathbf{m}[p], \mathbf{d}[p]) \leq \Pi^*(\mathbf{m}_0, \mathbf{d}_0)$

Proof (Lemma A.2): We have from Theorem 3.1 that $\forall p > 0$

$$(\mathbf{m}[p], \mathbf{d}[p]) \leq \Pi(\mathbf{m}[p - 1], \mathbf{d}[p - 1]) \leq$$

We observe that the operator Π is monotonic. Then we get

$$\leq \Pi^{(2)}(\mathbf{m}[p - 2], \mathbf{d}[p - 2]) \leq \Pi^{(p)}(\mathbf{m}[0], \mathbf{d}[0]) \leq$$

As by Theorem 3.1, $(\mathbf{m}[0], \mathbf{d}[0]) \leq (\mathbf{m}_0, \mathbf{d}_0)$,

$$\leq \Pi^{(p)}(\mathbf{m}_0, \mathbf{d}_0) \leq \sup_{p' \geq 0} \Pi^{(p')}(\mathbf{m}_0, \mathbf{d}_0) = \Pi^*(\mathbf{m}_0, \mathbf{d}_0) \quad \square$$

Proof (Theorem 3.2): This theorem derives from Lemma A.2 and the definition of total stability: Indeed, if \mathbf{m}^* is finite, then for any node n in the network, for any flow f traversing the node, we have an upper bound to the maximum number of bytes of that flow that can be present in a busy period at node n . Therefore from \mathbf{m}^* and \mathbf{d}^* we can derive finite bounds to the maximum queue size and maximum delay at each node. \square

C. Proof of Theorem 4.2

Proof : We consider separately the case of networks composed by a single strongly connected component, and the general case.

Networks with a single strongly connected component. From Lemma 4.3, we note that for any flow the only variables associated to super chains that come into play are those relative to the whole path of the flow in the network. Therefore, if we indicate with \mathbf{m}^{path} the array of all those variables in the network (one per flow), we can rewrite part of the set of inequalities in Lemma 4.3 in the form

$$\begin{cases} \mathbf{m}^{path}[p] \leq \Psi_1(\mathbf{m}^{path}[p - 1]) & \forall p \geq 1 \\ \mathbf{d}[p] \leq \Psi_2(\mathbf{d}[p - 1]) & \forall p \geq 1 \\ \mathbf{m}^{path}[0] \leq \mathbf{m}'_0 \\ \mathbf{d}[0] \leq \mathbf{d}_0 \end{cases} \quad (12)$$

\mathbf{m}'_0 is such that $\forall f$, $m'_{0,f} = L_f$. We observe that $\forall (\mathbf{m}, \mathbf{d}) \in (\mathbb{R}^+ \cup +\infty)^B$, $\Psi(\mathbf{m}, \mathbf{d}) \geq (\mathbf{m}_0, \mathbf{d}_0)$: therefore any solution of the fixed point $(\mathbf{m}, \mathbf{d}) = \Psi(\mathbf{m}, \mathbf{d})$ is larger than $(\mathbf{m}_0, \mathbf{d}_0)$. We have therefore two upper bounding operators, one for each class of variables. For each of them we can apply Lemma 4.2: if at least one of the fixed point problems $\mathbf{m}^{path} = \Psi_1(\mathbf{m}^{path})$, $\mathbf{d} = \Psi_2(\mathbf{d})$ admits a finite solution, then the network is stable, as in this way we have finite bounds to the values of all the variables belonging to one class, at least. As the two operators are linear, each of the two fixed points admits a finite solution if the spectral radius of the associated matrix is inferior to one.

General case. We note that neither the flow burstiness nor the initial buffer content play any role in the sufficient conditions for total stability in Theorem 4.2: then if a single strongly connected component network satisfies the sufficient conditions with a given setting of these parameters, it satisfies them for any setting of these parameters, and in particular, for any (finite) value of the flow burstiness at the input to the network. Let's consider then the case of a network composed by two strongly connected components, connected between them: this implies that some flows traverse first one component (we call

it component 1) and then the component 2 (no flow traverses them in the reverse order, otherwise we would have only one strongly connected component). Let us assume that each component, considered as a separate network, satisfies the sufficient conditions in Theorem 4.2 with a given allocation of flow burstiness: then as observed before, it satisfies them for any finite value of flow burstiness at the input to the component. All flows getting into component 2 and coming

from component 1 have a finite burstiness, as component 1 is totally stable: then component 2 is totally stable also in this configuration. Hence the network is totally stable. In the general case, the graph of the strongly connected components of a network has a feed forward structure, to which it can be easily extended the proof shown for the case of two components. \square